

TECNOLÓGICO
NACIONAL DE MÉXICO



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

**TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE ORIZABA
División de Estudios de Postgrado e Investigación**

DOCTORADO EN CIENCIAS DE LA INGENIERÍA

TESIS

**Desarrollo de software multi-dominio a partir de patrones de diseño
de interfaces de usuario usando técnicas de inteligencia artificial**

PARA OBTENER EL GRADO DE:
Doctor en ciencias de la ingeniería

PRESENTADO POR:
M.S.C Laura Nely Sánchez Morales

DIRECTOR DE TESIS:
Dr. Giner Alor Hernández

CODIRECTOR DE TESIS:
Dr. Jezreel Mejía Miranda

Índice

Índice de figuras	5
Índice de tablas	8
Abstract	10
Resumen	11
Introducción	12
Capítulo 1 Antecedentes	15
1.1 Marco teórico.....	15
1.1.1 Técnicas de Inteligencia Artificial para generar software	15
1.1.2 Aprendizaje Automático	15
1.1.2.1 Procesamiento Digital de Imágenes	15
1.1.2.2 Algoritmos de procesamiento de imágenes.....	15
1.1.3 Aprendizaje Profundo	17
1.1.3.1 Red Neuronal Artificial.....	17
1.1.3.2 Redes Neuronales Convolucionales	18
1.1.3.3 APIs de aprendizaje profundo	19
1.1.4 Momentos invariantes de Hu	20
1.1.5 Patrones de Diseño de Interfaz de Usuario	20
1.1.6 Software Multi-dominio, Aplicaciones Multi-dispositivo y Software Multi-plataforma	21
1.1.7 Aplicaciones educativas.....	21
1.1.8 Aplicaciones de tipo social media	22
1.1.9 Aplicaciones e-commerce	22
1.1.10 Enfoques de desarrollo de software.....	22
1.2 Planteamiento del problema	23
1.3 Objetivos	25
1.3.1 Objetivo general	25
1.3.2 Objetivos específicos	25
1.4 Hipótesis.....	25
1.5 Justificación	26
1.6 Contribución al conocimiento.....	27

Capítulo 2 Estado del arte	28
2.1 Enfoques de desarrollo multi-dispositivo	28
2.2 Herramientas de desarrollo de aplicaciones multi-dispositivo	30
2.3 Desarrollo de software multi-dominio	32
2.4 Análisis comparativo de trabajos relacionados.....	36
Capítulo 3 Metodología.....	39
3.1 Etapa de Análisis.....	40
3.1.1 Análisis de patrones para generar aplicaciones multi-dominio.	41
3.2 Etapa de Diseño	54
3.2.1 Proceso de desarrollo de aplicaciones multi-dominio, multi, dispositivo y multi-plataforma.	55
3.2.2 Diseño de la arquitectura de software.....	59
3.2.3 Diseño de la estructura del documento de configuración basado en XML.....	68
3.2.4 Conjunto de <i>mockups</i> de la herramienta generadora de aplicaciones..	69
3.2.5 Conjunto de iconos de patrones de diseño de interfaz de usuario.....	80
3.2.6 Mecanismo para el diseño de la estructura del contenido para aplicaciones móviles	94
3.2.6.1 Plantillas para aplicaciones multi-dominio	94
3.2.7 Representación en RuleML del proceso de identificación de elementos en imágenes generadas a mano alzada	96
3.2.7.1 Árbol parcial de las representaciones en RuleML	109
3.2.8 Características de la red neuronal usando Matlab.....	122
3.2.9 Características de la red neuronal convolucional usando Inception_V3.....	124
3.3 Etapa de Desarrollo.....	125
3.3.1 Repositorio de imágenes	125
3.3.2 Módulo inicial de procesamiento de imágenes con Matlab.....	128
3.3.2.1 Red neuronal.....	130
3.3.3 Módulo de procesamiento de imágenes con Deep Learning	133
3.3.3.1 Proceso de transformación de imágenes	133

3.3.3.2	Pruebas con red neuronal	134
3.3.4	Modulo generador de XML.....	155
3.3.5	Generador de código	157
3.3.6	Herramienta generadora de aplicaciones multi-dominio	159
Capítulo 4	Resultados.....	162
4.1	Caso de estudio 1: Generación de una aplicación educativa tipo Wiki a mano alzada.....	162
4.1.1	Evaluación	170
4.1.1.1	Diseño de la evaluación.....	172
4.1.1.2	Resultados.....	177
4.1.1.3	Discusión	181
4.2	Caso de estudio 2: Generación de una aplicación educativa tipo MOOC a través del Wizard.....	182
4.2.1	Evaluación.....	188
4.2.1.1	Diseño de la evaluación.....	189
4.2.1.2	Discusión	201
4.3	Caso de estudio 3: Generación de una aplicación e-commerce tipo B2C a mano alzada.....	202
4.3.1	Evaluación.....	207
4.3.1.1	Diseño de la evaluación.....	209
4.3.1.2	Resultados.....	213
4.3.1.3	Discusión	214
4.4	Caso de estudio 4: Generación de una aplicación tipo Social Networks a mano alzada.....	216
Capítulo 5	Conclusiones.....	223
5.1	Conclusiones.....	223
5.2	Trabajo a futuro.....	224
Productos Académicos	226
Referencias	229

Índice de figuras

Figura 3.1 Etapas de la metodología.....	39
Figura 3.2 Etapas del proceso de desarrollo de software.....	55
Figura 3.3 Diseño de la arquitectura.....	59
Figura 3.4 Estructura general del documento de configuración basado en XML..	69
Figura 3.5 Mockup de la pantalla de inicio de sesión	70
Figura 3.6 Mockup de la página principal.	71
Figura 3.7 Mockup de la página principal del proceso ideal: selección del dominio.	72
Figura 3.8 Mockup del proceso ideal: a) selección del tipo de aplicación educativa, b) selección del tipo de aplicación e-commerce y c) selección del tipo de aplicación social media.	73
Figura 3.9 Mockup del proceso ideal: Selección de una plantilla	74
Figura 3.10 Mockup del proceso ideal: Selección de patrones.....	74
Figura 3.11 Mockup del proceso ideal: Selección de dispositivos/plataformas. ...	75
Figura 3.12 Mockup del proceso ideal: Configuración.....	76
Figura 3.13 Mockup del proceso ideal: Confirmar configuración.....	76
Figura 3.14 Mockup del proceso ideal: a) Generación de aplicación y b) Descargar código.....	77
Figura 3.15 Mockup del proceso a mano alzada: Cargar imagen.	77
Figura 3.16 Mockup del proceso a mano alzada: Identificación de patrones.	78
Figura 3.17 <i>Mockups</i> del proceso a mano alzada: a) Selección de patrones, b) Selección del dominio de la aplicación y c) Selección de una plantilla.....	79
Figura 3.18 <i>Mockups</i> del proceso a mano alzada: a) Selección del tipo de aplicación educativa, b) Selección del tipo de aplicación e-commerce y c) Selección del tipo de aplicación social media.	80
Figura 3.19 Plantillas para generar aplicaciones multi-dominio.....	95
Figura 3.20 Plantillas para aplicaciones educativas: a) Blog, b) MOOC y c) Wiki.95	
Figura 3.21 Arquitectura de la red neuronal convolucional Inception_v3.	125
Figura 3.22 UIDPs de muestra: a) Video, b) Carousel, c) Wizard, d) Notification, e) List, f) Login, g) Progress bar, h) Dashboard, i) Contextual menu, j) Accordion y k) Splashscreen.....	131
Figura 3.23 Proceso de transformación: (a) Imagen original, (b) eliminación de sombras, (c) umbral optimizado.	134
Figura 3.24 Estructura del archivo de validación JSON: a) Posiciones validas de cada UIDP, b) Posiciones activas por plantilla y c) Patrones de diseño permitidos por tipo de aplicación.	157
Figura 3.25 Estructura de carpetas generada.	158
Figura 3.26 Estructura de carpetas para los proyectos de las aplicaciones de Android™	159
Figura 3.27 Estructura de carpetas para los proyectos de las aplicaciones de iOS®	159

Figura 3.28 Pantallas de MultiGene: a) Login y b) Menú principal.	160
Figura 3.29 Interfaz del inicio para el proceso ideal.	161
Figura 3.30 Pantalla de inicio del proceso a mano alzada.	161
Figura 4.1 Interfaz generada a mano alzada para crear una Wiki.	163
Figura 4.2 Pantalla 1: a) Subir la interfaz, b) UIDPs identificados.	164
Figura 4.3 Pantalla 2: Selección de nuevos UIDPs.	164
Figura 4.4 Pantalla 3: Selección del dominio de la aplicación.	165
Figura 4.5 Pantalla 4: Selección de plantilla.	165
Figura 4.6 Pantalla 5: Selección del tipo de aplicación educativa.	166
Figura 4.7 Pantalla 6: Selección de dispositivos y plataformas de desarrollo. ...	167
Figura 4.8 Pantalla 7: Configuración.	167
Figura 4.9 Pantalla 8: a) Confirmación de la configuración y b) Generación del código fuente de la aplicación y c) descarga de la aplicación.	168
Figura 4.10 Despliegue de la aplicación en Android™ Studio: a) Splashscreen, b) Carousel y List, c) List y Search, d) Video.	169
Figura 4.11 Despliegue de la aplicación en XCode: a) Splashscreen, b) Carousel y List, c) List y Search, d) Video.	169
Figura 4.12 Resultados gráficos de la evaluación.	181
Figura 4.13 Pantalla 1 del proceso ideal: Selección del dominio de la aplicación.	183
Figura 4.14 Pantalla 2 del proceso ideal: Selección del tipo de aplicación educativa.	184
Figura 4.15 Pantalla 3 del proceso ideal: Selección de una platilla.	184
Figura 4.16 Pantalla 4 del proceso ideal: Selección de patrones.	185
Figura 4.17 Pantalla 5 del proceso ideal: Selección de dispositivos y plataformas.	185
Figura 4.18 Pantalla 6 del proceso ideal: Configuración.	186
Figura 4.19 Pantalla 7 del proceso ideal: a) Confirmación de la configuración y b) Generación del código fuente de la aplicación y c) descarga de la aplicación. ...	187
Figura 4.20 Despliegue de la aplicación desde Android™ Studio: a) Splashscreen; b) Login; c) Dashboard; d) List; e) Video.	187
Figura 4.21 Diseño de la evaluación de <i>MultiGene</i>	191
Figura 4.22 Gráfica de resultados de la evaluación cualitativa.	195
Figura 4.23 Gráfica de resultados de evaluación cuantitativa.	201
Figura 4.24 Interfaz generada a mano alzada para crear una aplicación B2C... ..	203
Figura 4.25 Paso 1: a) Elementos identificados en una imagen generada a mano alzada; y b) Selección de UIDPs.	204
Figura 4.26 Paso 2: a) Selección de una plantilla de diseño; y b) selección de un tipo de aplicación de e-commerce.	205
Figura 4.27 Paso 3: a) Selección del tipo de software multiplataforma; y b) configuración de la aplicación.	206
Figura 4.28 Paso 4 UIDPs de la aplicación e-commerce: a) Product page, b) Menu, c) Comment, y d) Search y Shopping Cart.	207

Figura 4.29 Grafica de resultados de evaluación.	214
Figura 4.30 Interfaz de usuario para una aplicación tipo Social networks.	217
Figura 4.31 Paso 1: a) Elementos identificados, y b) Composición seleccionada.	218
Figura 4.32 Paso 2: a) Selección del diseño de la plantilla, y b) selección del tipo de aplicación social media.	219
Figura 4.33 Paso 3: a) Selección del diseño de la plantilla, y b) selección del tipo de aplicación social media.	220
Figura 4.34 Paso 4 UIDPs de la aplicación: a) Gallery, b) Search, c) Menu, y d) Reaction y Follow.	222

Índice de tablas

Tabla 2.1 Análisis de trabajos relacionados.....	36
Tabla 3.1 Descripción y beneficios de las aplicaciones educativas.....	42
Tabla 3.2 Relación de patrones de diseño de interfaces de usuario por plataforma.....	42
Tabla 3.3 Relación de patrones de diseño de interfaces de usuario por contexto de uso y plataforma.....	44
Tabla 3.4 Relación entre los patrones de diseño de IU y las aplicaciones educativas.....	44
Tabla 3.5 Relación de aplicaciones, UIDPs y su posición en la interfaz de usuario.....	46
Tabla 3. 6 UIDPs identificados en las aplicaciones de e-commerce.....	49
Tabla 3.7 Clasificación de aplicaciones social media.....	51
Tabla 3.8 UIDPs identificados en aplicaciones móviles en el ámbito entretenimiento: Social media.....	53
Tabla 3.9 Iconografía de patrones de diseño de interfaz de usuario en su representación individual.....	81
Tabla 3.10 Iconografía de interfaces de usuario compuestas para el dominio educativo.....	88
Tabla 3.11 Iconografía de interfaces de usuario compuestas para el dominio e-commerce.....	89
Tabla 3.12 Iconografía de interfaces de usuario compuestas para el dominio social media.....	92
Tabla 3.13 Reglas RuleML para representaciones de patrones simples e individuales.....	97
Tabla 3.14 Reglas RuleML para la representación de patrones individuales en composición.....	103
Tabla 3.15 Árboles de reglas para patrones simples.....	109
Tabla 3.16 Árboles de reglas para patrones en composición.....	116
Tabla 3.17 Imágenes ideales y no ideales generadas para patrones de diseño simples.....	126
Tabla 3.18 Imágenes reales en blanco y negro, así como reales a color generadas para composiciones.....	127
Tabla 3.19 Resultados de entrenamiento y evaluación con 1 capa oculta.....	131
Tabla 3.20 Resultados de entrenamiento y evaluación con 2 capas ocultas.....	132
Tabla 3. 21 Resultados de la prueba con TensorFlow y el modelo binario para: galería y reproductor de música.....	135
Tabla 3.22 Resultados de TensorFlow y modelo binario para: Gallery, Horizontal menu, Vertical menu y Panel.....	136
Tabla 3.23 Resultados de TensorFlow y el modelo AlexNet para los patrones: Gallery, Horizontal menu, Vertical menu, Music player, Panel, Picker y Ranking.....	137

Tabla 3.24 Resultados de entrenamiento usando AlexNet con 9 patrones y 200 instancias por UIDP.....	138
Tabla 3.25 Resultados de entrenamiento usando AlexNet con 9 patrones y 400 instancias por UIDP.....	139
Tabla 3.26 Resultados de entrenamiento usando AlexNet con 17 UIDPs y 400 instancias por UIDP (Parte 1).....	140
Tabla 3.27 Resultados de entrenamiento usando AlexNet con 27 UIDPs y 400 instancias por UIDP (Parte 1).....	141
Tabla 3.28 Resultados de entrenamiento usando Inception_V3 con 9 patrones y 200 instancias por UIDP.....	147
Tabla 3.29 Resultados de entrenamiento usando Inception_V3 con 9 UIDPs y 400 instancias por UIDP.....	148
Tabla 3.30 Resultados de entrenamiento usando Inception_V3 con 17 UIDPs y 400 instancias por UIDP (Parte 1).....	149
Tabla 3.31 Resultados de entrenamiento usando Inception_V3 con 27 UIDPs y 400 instancias por UIDP (Parte 1).....	150
Tabla 4.1 Cuestionario de la evaluación.	176
Tabla 4.2 Resultados de la evaluación.	178
Tabla 4.3 Resultados de evaluación cualitativa.	194
Tabla 4.4 Medida de la eficiencia.....	199
Tabla 4.5 Medida de efectividad.	200
Tabla 4.6 Medida del soporte.....	200
Tabla 4.7 Cuestionario de evaluación cualitativa.	211

Abstract

Over the years, several approaches to software development have been proposed. Some approaches are based on models such as MDA (Model-Driven Architecture) or MDD (Model-Driven Development), others are oriented agile development as RAD (Rapid Application Development) or FDD (Feature-Driven Development) to mention but a few. However, the techniques addressed by these approaches lack the use of artificial intelligence techniques, neural networks, deep learning, and patterns recognition. Due to this, new scientific knowledge and computational methods on multi-domain applications development process are needed by incorporating artificial intelligence techniques for an agile, easy and intuitive software development process. The aim of this document is to present the results of this thesis. The thesis consists in multi-domain software development by using user interface design patterns and artificial intelligence techniques. In other words, it is to generate source code of an application from an image generated freehand which represents a user interface. The application generation process consists of three phases: (1) Identification of elements using artificial intelligence techniques such as image processing, neural networks, and deep learning. In this phase, the artificial intelligence techniques allow identifying the design patterns in an image generated freehand. (2) Configuration: in this phase, the necessary data for the development and deployment of the application are added. The devices (Desktop, Tablet / Smartphone and TV), development platforms (Web, Android™, iOS®, and Firefox® OS), domain of the application (Educational, e-commerce or social media), type of application according to domain, template to present the content in the user interface, name of application, and data about author of the application are the main configuration information. The data are stored in an XML-based file. (3) Source code generation: in this phase, the XML-based file is interpreted and the source code of a multi-domain application is generated.

Resumen

A lo largo del tiempo se han propuesto diversos enfoques de desarrollo de software. Algunos de los enfoques se basan en modelos como MDA (*Model-Driven Architecture*) o MDD (*Model-Driven Development*), otros son orientados al desarrollo ágil como RAD (*Rapid Application Development*) o FDD (*Feature-Driven Development*) por mencionar algunos. Las técnicas adoptadas por dichos enfoques carecen del uso de técnicas de inteligencia artificial, redes neuronales, aprendizaje profundo y reconocimiento de patrones. Debido a esto, se necesitan nuevos conocimientos científicos y métodos computacionales en el proceso de desarrollo de aplicaciones multi-dominio incluyendo técnicas de inteligencia artificial para un proceso de desarrollo de software ágil, fácil e intuitivo. El objetivo de este documento es presentar los resultados del desarrollo de este tema de tesis. El trabajo de tesis consiste en el desarrollo de software multi-dominio a partir de patrones de diseño de interfaz de usuario y técnicas de inteligencia artificial. En otras palabras, consiste en generar código fuente de aplicaciones a partir de una imagen generada a mano alzada que represente una interfaz de usuario. El proceso de generación de aplicaciones consta de tres fases: (1) Identificación de elementos usando técnicas de inteligencia artificial como el procesamiento de imágenes, redes neuronales y aprendizaje profundo. En esta fase las técnicas de inteligencia artificial permiten identificar los patrones de diseño en la imagen generada a mano alzada. (2) Configuración: en esta fase se agregan los datos necesarios para el desarrollo y despliegue de la aplicación. Los principales datos de configuración son los dispositivos (Escritorio, Tablet/Smartphone y TV) y plataformas de desarrollo (Web, Android™, iOS®, y Firefox® OS), dominio de la aplicación (Educativo, e-commerce o social media), tipo de aplicación de acuerdo al dominio de desarrollo, plantilla para presentar el contenido en la interfaz de usuario, nombre de la aplicación y datos sobre el autor de la aplicación. Los datos se almacenan en un archivo de configuración basado en XML (*Extensible Markup Language*). (3) Generación de código fuente: en esta fase se interpretan el documento basado en XML y se genera el código fuente de una aplicación multi-dominio.

Introducción

El desarrollo de software es un área de la ingeniería de software fundamental en constante crecimiento y evolución. Como parte de la evolución, en las técnicas de desarrollo de software se implementan diversos enfoques de desarrollo como el desarrollo basado en características (FDD) con énfasis en las fases de diseño y construcción, así como iteraciones cortas que generan rápidamente entregas de software, sin embargo, la falta de documentación hace que sea imposible la reutilización de código. Otro enfoque es la arquitectura basada en modelos (MDA) centrada en el desarrollo de software a través de la generación y transformación de modelos, los tres modelos principales de este enfoque son: modelo independiente de cálculo (CIM), el modelo independiente de la plataforma (PIM) y la plataforma de modelo específico (SMP); sin embargo, este enfoque requiere tener un conocimiento adecuado o acceso a las herramientas correctas del proceso de transformación. Por otra parte, se tiene el enfoque de desarrollo rápido de aplicaciones (RAD) con un desarrollo iterativo y creación de prototipos de interfaces a través de herramientas de ingeniería de software asistida por computadora (CASE), sin embargo, la desventaja de una aplicación desarrollada bajo este enfoque es que necesita interactuar con sistemas existentes o archivos con errores debido a la codificación deficiente causada por una entrega rápida.

Además de los enfoques de desarrollo, se reportan herramientas para el desarrollo de aplicaciones móviles que aplican diversas técnicas de desarrollo como el uso de marcos de trabajo (Freitas & Maia, 2015), (Grønli et al., 2014), (Rogai et al., 2015) y técnicas basadas en diversos enfoques de desarrollo (Marin et al., 2015), (Umuhoza et al., 2015), (Costa et al., 2014), (Panunzio & Vardanega, 2014) por mencionar algunas. En este sentido, todas las herramientas, técnicas y enfoques de desarrollo tienen un objetivo en común: agilizar el proceso de desarrollo de software.

Por otra parte, el uso de patrones de diseño de interfaz de usuario como técnica de desarrollo de software requiere en la mayoría de ocasiones conocimientos avanzados, sin embargo, es una técnica que bien aplicada permite solventar los principales problemas de diseño en el desarrollo de software como una adecuada

navegabilidad, interfaces amigables, con adecuada visualización del contenido independiente del dispositivo gracias al diseño responsivo.

Partiendo de lo anterior, este trabajo de investigación plantea el desarrollo de un conjunto de nuevos métodos computacionales para la generación automática de software multi-dominio y multi-dispositivo a partir de patrones de diseño de interfaces de usuario usando técnicas de inteligencia artificial. Lo anterior implica una contribución directa en el área de las ciencias computacionales, concretamente en la Ingeniería de Software estableciendo un nuevo enfoque y/o paradigma para el proceso de desarrollo automático de software de una forma más ágil, intuitiva y novedosa. Dicho lo anterior, en este documento se presentan los avances obtenidos hasta el momento, los cuales se centran en la generación automática de aplicaciones móviles del dominio médico.

El contenido de este documento se estructura en cuatro capítulos, en el capítulo 1 se presenta el marco teórico que incluye los conceptos más relevantes relacionados con este tema de tesis como las técnicas de inteligencia artificial (procesamiento de imágenes digitales, redes neuronales artificiales, momentos invariantes de *Hu*), patrones de diseño de interfaz de usuario, software multi-dominio, multi-dispositivo y multi-plataforma, enfoques de desarrollo de software. También se incluye el planteamiento del problema, el objetivo general y objetivos específicos, hipótesis, justificación, y contribución al conocimiento.

El capítulo 2 incluye el estado del arte con los trabajos relacionados divididos en tres subsecciones: enfoques de desarrollo multi-dispositivo, herramientas de desarrollo de aplicaciones multi-dispositivo, y desarrollo de software multi-dominio. Al final de este capítulo 2 se incluye el análisis comparativo de los trabajos revisados y las principales características identificadas.

Respecto al capítulo 3, el contenido se enfoca en la descripción de las actividades realizadas en cada etapa de la metodología de desarrollo de este trabajo de tesis. Las etapas descritas a lo largo de este capítulo son cinco: 1) Análisis: el análisis comprende los patrones de diseño identificados para el dominio educativo; 2) diseño: sobre el proceso de desarrollo, diseño de la arquitectura, diseño de la estructura del documento basado en XML para la generación automática de

aplicaciones, diseño del conjunto de *mockups* para el desarrollo de una herramienta, el conjunto de iconos de patrones de interfaz de usuario, el diseño del mecanismo para el diseño de la estructura del contenido de las aplicaciones, el diseño de la representación en RuleML del proceso de identificación de elementos en imágenes generadas a mano alzada, finalmente el diseño de las características de la red neuronal artificial para la identificación de elementos en imágenes. 3) Desarrollo: donde se incluyen los módulos de procesamiento de imágenes, módulo generador del documento basado en XML, módulo generador de código y la herramienta generadora de aplicaciones.

El capítulo 4 contiene los resultados obtenidos a través de diferentes casos de estudio. El primer caso de estudio describe la generación de una aplicación educativa tipo Wiki a través de una imagen generada a mano alzada. Además, se describen los resultados obtenidos en la evaluación cualitativa de las aplicaciones generadas a través de nuestra herramienta. El segundo caso de estudio consiste en la generación de una aplicación educativa de tipo MOOC a través de nuestra herramienta. Al final del caso de estudio, se describen los resultados de una evaluación enfocada a nuestra herramienta como generadora de aplicaciones educativas. El tercer caso de estudio presenta la generación de una aplicación e-commerce tipo B2C usando una imagen generada a mano alzada, así como la evaluación de la interfaz generada a través de nuestra herramienta. Finalmente, el cuarto caso de estudio describe la generación de una aplicación social media de tipo red social usando una imagen real generada a mano alzada.

Finalmente, el capítulo 5 incluye las conclusiones y trabajo a futuro de este trabajo de investigación.

Capítulo 1 Antecedentes

En este capítulo se presenta el conjunto de conceptos básicos relacionados con el tema de tesis. Se incluye el planteamiento del problema, los objetivos, hipótesis, justificación y contribución al conocimiento de este trabajo de tesis.

1.1 Marco teórico

1.1.1 Técnicas de Inteligencia Artificial para generar software

1.1.2 Aprendizaje Automático

Machine Learning o Aprendizaje Automático es derivado como subconjunto de la I.A. (The Mathworks Inc., 2018). El aprendizaje automático ofrece técnicas y modelos que se seleccionan en función de la aplicación, dependiendo del tamaño de datos para procesar y el tipo de problema a resolver.

1.1.2.1 Procesamiento Digital de Imágenes

El procesamiento digital de imágenes es el conjunto de técnicas que se aplican a las imágenes digitales con el objetivo de mejorar su calidad o facilitar la búsqueda de información inmersa en ellas. En este sentido el tratamiento digital de imágenes contempla el procesamiento y el análisis de imágenes (*Digital Image Processing Using Matlab - Gonzalez Woods & Eddins.pdf*, n.d.). Este procesamiento se refiere a la realización de transformaciones y a la restauración y mejoramiento de las imágenes. El análisis consiste en la extracción de propiedades y características de las imágenes, así como la clasificación, identificación y el reconocimiento de patrones (De la Rosa, 2007).

1.1.2.2 Algoritmos de procesamiento de imágenes

Dentro de los algoritmos para procesamiento de imágenes, existen dos clases de transformaciones de imágenes:

Transformaciones radiométricas, en las cuales los valores de nivel de gris de los píxeles son alterados sin modificar la geometría de la imagen (contrastes, filtrados, clasificación, texturas, cocientes). Aquí se distinguen dos grupos de algoritmos de procesamiento de imágenes: puntual y espacial.

1. **Procesamiento Puntual:** las operaciones sobre un píxel de la imagen se realizan sin tener en cuenta los píxeles vecinos (ensanche de contraste, umbralización, pseudocolor, operaciones algebraicas entre imágenes).
2. **Procesamiento espacial:** la operación para un píxel de la imagen de salida tiene en cuenta tanto el píxel correspondiente en la imagen de entrada como una cantidad arbitraria de vecinos de éste (operaciones de filtrado, gradientes, realce de bordes).

Transformaciones geométricas, en las que se altera la geometría de la imagen, es decir, la ubicación de los píxeles dentro de esta.

A continuación, se presenta una pequeña descripción de algunos algoritmos empleados para el procesamiento de imágenes, es importante mencionar que estos algoritmos son de manera general puesto que existen distintos aplicados para cada uno.

Filtrado (filtro de mediana): Un filtro de mediana es un filtro digital no lineal, que es capaz de preservar los cambios de señal agudos y es muy eficaz en la eliminación de ruido de impulso. Este algoritmo sustituye el valor de un píxel por el valor de la media de los píxeles vecinos (*Digital Image Processing Using Matlab - Gonzalez Woods & Eddins.pdf*, n.d.). Es capaz de mejorar ciertas características de una imagen que posibiliten efectuar operaciones del procesado sobre ella.

Operaciones morfológicas: Las operaciones morfológicas se utilizan en binario y las imágenes en escala de grises, y es útil en muchas áreas de procesamiento de imágenes, tales como esqueletización, restauración y análisis de la textura. Un operador morfológico utiliza un elemento de estructuración para procesar una imagen.

Operaciones de convolución: La convolución es una operación matemática simple que es fundamental para muchos operadores comunes de procesamiento de imágenes. Convolución es una forma de multiplicar juntos dos conjuntos de

números de diferentes tamaños para producir una tercera matriz de números. La convolución pertenece a una clase de algoritmos denominados filtros espaciales.

Detección de bordes: La detección de bordes se utiliza ampliamente en la segmentación de la imagen cuando se quiere dividir la imagen en las zonas correspondientes a los diferentes objetos, llamadas zonas de interés.

1.1.3 Aprendizaje Profundo

Derivado como un subconjunto de la I.A y del aprendizaje automático que utiliza redes neuronales artificiales, el aprendizaje profundo permite modelos computacionales que son compuestos de múltiples capas procesadas para aprender representaciones de datos con contenido de varias capas, que permiten el reconocimiento del lenguaje, la detección de algún objeto e inclusive la traducción. El aprendizaje profundo descubre una estructura intrincada sobre un conjunto de datos, usando el algoritmo de retro propagación para indicar como la máquina cambia sus parámetros internos para la representación en cada capa desde la representación en una capa previa. Los métodos mantienen un mejoramiento en el reconocimiento del lenguaje, reconocimiento de objetos visuales, detección de objetos y muchos otros dominios tales como descubrimiento de medicamentos y genómicas (Lecun et al., 2015).

Gracias a las grandes cantidades de datos etiquetados y la potencia de cálculo de las GPU (*Graphics Processing Unit*, Unidad de Procesamiento Gráfico) se logra una eficiencia para cubrir el aprendizaje profundo. Algunas técnicas de aprendizaje actualmente aplicadas son:

- Conducción autónoma: detección de objetos y personas, contribuye a reducir accidentes.
- Sector aeroespacial y de defensa: identificar objetos desde los satélites en ciertas áreas seguras o no seguras.
- Investigación médica: detección de células cancerígenas.
- Electrónica (CES): audición automatizada y la traducción del habla.

1.1.3.1 Red Neuronal Artificial

Las redes neuronales artificiales son modelos matemáticos que permiten modelar el comportamiento del cerebro humano. En (Flórez & Fernández, 2008) se menciona que, las redes neuronales artificiales son sistemas complejos que procesan información, y tienen la capacidad de responder dinámicamente.

Existen diferentes modelos de redes neuronales artificiales algunos de ellos y sus características principales, de acuerdo con (Flórez & Fernández, 2008) son los siguientes:

- 1. Perceptrón simple:** red unidimensional, con dos capas de neuronas es decir; N celdas de entrada y M neuronas de salida. Modelo útil en tareas de clasificación como para la representación de funciones booleanas, permite discriminar entre dos clases linealmente separables.
- 2. Adaline (*AD*aptive *LINE*ar *E*lement) y Madaline (*M*ultiple *AD*ALINE):** modelos muy similares al Perceptrón simple, Adaline utiliza funciones de transferencia lineales y Madaline permite conectarse de forma sucesiva, en ambos casos el mecanismo de aprendizaje es la regla de *Widrow-Hoff*.
- 3. Perceptrón Multicapa:** modelo con propagación hacia adelante, se caracteriza por su organización en capas de celdas disjuntas, esto es, ninguna salida neuronal constituye una entrada para las neuronas de la misma capa o de capas previas. De acuerdo con (Díez et al., 2001) la arquitectura del perceptrón multicapa tiene al menos tres capas, una de entrada, una de salida y una o más intermedias. Es el modelo más utilizado en implementaciones comerciales.

1.1.3.2 Redes Neuronales Convolucionales

Las ConvNet o CNN (Redes Neuronales Convolucionales) son diseños para procesar datos que llegan en forma de múltiples arreglos (Lecun et al., 2015), por ejemplo, una imagen a color compuesta de tres arreglos 2D que contiene intensidades de píxeles en los tres canales de color. Las modalidades de los datos están en la forma de múltiples arreglos: 1D para señales y secuencias agregando el lenguaje; 2D para imágenes o espectrogramas de audio y 3D para video o imágenes volumétricas. Se mantiene cuatro ideas claves detrás de las ConvNets

que toman ventaja de las propiedades de señales locales: conexiones locales, pesos compartidos, agrupación y el uso de muchas capas.

1.1.3.3 APIs de aprendizaje profundo

A continuación, se explican algunas APIs (Interfaz de Programación de Aplicaciones) que soportan algoritmos de aprendizaje profundo, el uso de bibliotecas o lenguajes de programación que se utilizan para las diferentes herramientas de desarrollo.

TensorFlow: Es una biblioteca de código abierto para el cálculo numérico utilizando gráficos de flujo de datos. Desarrollado por Google® en la organización de Inteligencia Artificial para el uso del aprendizaje automático y la investigación de redes neuronales profundas, pero se usa de forma general permitiendo una gran variedad de dominios (*Tensor Flow*, 2018).

Theano: es una biblioteca para Python que permite definir, optimizar y evaluar expresiones matemáticas, especialmente aquellas con matrices multidimensionales. El uso de Theano permite alcanzar velocidades que rivalizan con las implementaciones en C, que involucran problemas con grandes cantidades de datos. Además, combina aspectos de un sistema de álgebra computacional con aspectos de un compilador de optimización (*Theano at a Glance*, 2018).

Caffe: Es un *framework* de aprendizaje profundo hecho con expresiones, velocidad y modularidad. La arquitectura expresiva fomenta la aplicación y la innovación. Los modelos y la optimización se definen mediante la configuración sin codificación rígida, solamente configurando un solo entrenamiento en una GPU e implementarla en clúster de productos básicos y dispositivos móviles. Mantiene un gran desarrollo activo, y cuenta con una gran implementación en la industria y con la comunidad de investigación (*Caffe*, 2018).

DeepLearning4j: Es una biblioteca de aprendizaje profundo escrita en Java y la JVM (Máquina Virtual de Java). Esta biblioteca de funciones permite componer redes neuronales profundas de varias redes poco profundas, de las cuales forman una llamada 'capa'. Esta flexibilidad le permite combinar redes convolucionales, y los autocodificadores de secuencia a secuencia, autocodificadores variacionales o redes recurrentes usando las CPUs y las GPUs, su principal lenguaje es Java, pero

también está disponible para Python, C++, Scala, Kotlin, entre otros (*Deep Learning for Java*, 2018).

1.1.4 Momentos invariantes de Hu

Son parámetros estadísticos invariantes a la traslación, tamaño del objeto de análisis y rotación de acuerdo con (Gutiérrez et al., 2012). Los momentos invariantes son uno de los descriptores de forma más usados desde que se introdujeron en los 60's por *Hu* (Srihari & Franke, 2008).

1.1.5 Patrones de Diseño de Interfaz de Usuario

Los patrones de diseño de interfaz de usuario son modelos que permiten resolver problemas de diseño con características comunes. Se reportan diferentes clasificaciones, entre las que se encuentra la categorizada por plataforma: Móvil (Neil, 2014; Sheibley, 2013; Tidwell, 2010), Smart TV (Developer, 2013; Android TV, 2015; Apple TV, 2016), y Web (Tidwell, 2010; Toxboe, 2016). A continuación, se presenta una pequeña lista de patrones de diseño aplicados en distintas plataformas de software:

- **Splash screen:** este patrón permite mostrar al usuario una pantalla personalizada o no al iniciar una aplicación (Neil, 2014).
- **Login:** patrón que permite controlar el acceso de usuarios a un área restringida (Neil, 2014).
- **Video:** Es un patrón de diseño ideal para incorporar contenido multimedia en una aplicación (Neil, 2014).
- **Dashboard:** Permite al usuario visualizar de forma rápida el contenido más importante de una aplicación, sin la necesidad de navegar en otra pantalla (Toxboe, 2016).
- **List:** en este patrón se muestra al usuario un conjunto de opciones para navegar dentro de la aplicación (Toxboe, 2016)
- **First walktrought:** Patrón de diseño que permite incorporar ayuda acerca del funcionamiento de la aplicación. Es ideal cuando las aplicaciones

diseñadas son poco intuitivas o bien contienen procesos complejos (Toxboe, 2016).

1.1.6 Software Multi-dominio, Aplicaciones Multi-dispositivo y Software Multi-plataforma

El software multi-dominio permite el desarrollo de aplicaciones para dos o más dominios de aplicación. Para cada dominio las aplicaciones se adaptan de acuerdo a las necesidades del contexto.

Las aplicaciones multi-dispositivo como su nombre lo indica, permiten el desarrollo en un lenguaje o conjunto de lenguajes y su posterior ejecución en diferentes dispositivos de hardware. Una característica importante de las aplicaciones multi-dispositivo es que en cada dispositivo su presentación es distinta, pero sin afectar su funcionalidad esencial.

El software multi-plataforma se refiere a aplicaciones que se ejecutan en dos o más plataformas. Las plataformas de ejecución se refieren al sistema o sistemas operativos. El desarrollo de estas aplicaciones multi-plataforma implica la generación de versiones para cada plataforma o bien el desarrollo de una misma aplicación con la capacidad de ejecutarse en al menos dos plataformas distintas.

1.1.7 Aplicaciones educativas

Las aplicaciones educativas difieren sustancialmente de las aplicaciones tradicionales, debido a que sus principales objetivos son capturar la atención del usuario, aplicar una teoría del aprendizaje y, en última instancia, compartir el conocimiento. Gracias al uso de las tecnologías de la información, actualmente se cuenta con un gran número de aplicaciones educativas que están disponibles para todas las personas sin importar su edad, género o intereses. Esto debido a las comunidades de aprendizaje que surgen alrededor de todo el mundo, por ejemplo, los blogs, videoconferencias, juegos, MOOCs, simuladores, WebQuests y wikis, por mencionar algunos (Zhan & Mei, 2013).

1.1.8 Aplicaciones de tipo social media

Las aplicaciones de tipo social media o medios sociales permiten la creación e intercambio de contenido generado por el usuario y el diseño de una gama de aplicaciones basadas en Internet (Tsikerdekis & Zeadally, 2014). Algunos tipos de aplicaciones social media son *microbloggin* como Twitter®[®], *streaming* como Spotify®[®], y redes sociales como Facebook®[®].

1.1.9 Aplicaciones e-commerce

Las aplicaciones de e-Commerce (Comercio Electrónico) son soluciones de comercio electrónico o comercio por Internet, las cuales son seguras y permiten satisfacer las necesidades de los clientes y empresas. Permiten interactuar con los clientes a través de productos y ofertas personalizados, procesar información de forma rápida y segura, para enfocarse en el cumplimiento y en dar servicio al cliente. Además, tienen soporte para dar soluciones por medio de redes sociales o páginas Web (*Comercio Electrónico*, 2018). De acuerdo con su función, se encuentran diferentes tipos de aplicaciones e-commerce como B2C, C2C o B2B.

1.1.10 Enfoques de desarrollo de software

Para agilizar el proceso de desarrollo de software se han propuesto diversas metodologías y enfoques de desarrollo, algunos de los más relevantes se describen a continuación.

MDA (Model-Driven Architecture): es un enfoque para el desarrollo de software propuesto por OMG (*Object Management Group*), MDA usa lenguajes de modelado como UML (Lenguaje de Modelado Unificado), el *Meta Object Facility* (MOF) y el *Common Warehouse Metamodel* (CWM). MDA permite abordar el ciclo completo de desarrollo, que abarca el análisis, diseño, programación, pruebas, ensamblaje de componentes, así como el despliegue y mantenimiento (Truyen, 2006).

MDD (Model-Driven Development): Es un enfoque para el desarrollo de software donde se crean modelos extensos antes de escribir el código fuente. Al considerar

los modelos como entidades de primera clase, MDD apunta a reducir la complejidad de la producción de software (Reda et al., 2005).

MDE (Model-Driven Engineering): es una metodología de desarrollo de software que se centra en la creación y explotación de modelos de dominio, que son modelos conceptuales de todos los temas relacionados con un problema específico. Por lo tanto, destaca y apunta a representaciones abstractas del conocimiento y las actividades que rigen un dominio de aplicación particular (Schmidt, 2006)

FDD (Feature-Driven Development): es un proceso de desarrollo de software iterativo e incremental. Es uno de varios métodos ágiles para desarrollar software creado para trabajar con equipos grandes de desarrollo (Stephens, 2015). FDD combina una serie de mejores prácticas reconocidas por la industria en un todo cohesivo. Todas estas prácticas se basan en una perspectiva de funcionalidad (*feature*) valorada por el cliente (Coad et al., 1999; Palmer & Felsing, 2001).

RDD (Responsibility-Driven Design): es una técnica de diseño en programación orientada a objetos, que mejora la encapsulación mediante el uso del modelo cliente-servidor. Fue propuesta por Rebecca Wirfs-Brock y Brian Wilkerson (Wirfs-Brock & McKean, 2003; Wirfs-Brock & Wilkerson, 1989).

1.2 Planteamiento del problema

En la ingeniería de software, de manera específica en el desarrollo de software se han propuesto diversos enfoques de desarrollo tales como FDD (*Feature-Driven Development*), MDA (*Model Driven Architecture*), RAD (*Rapid Application Development*), por mencionar algunos, los cuales proponen la implementación de técnicas de desarrollo muy similares. Sin embargo, estas técnicas están enfocadas principalmente a desarrolladores experimentados, ya que se requiere el conocimiento de modelos abstractos, análisis de requisitos, experiencia en el desarrollo de prototipos y experiencia en herramientas de Ingeniería de Software Asistido por Computadora (CASE). Otros enfoques de desarrollo de software novedosos, parten de técnicas como el reconocimiento de voz (Arnold et al., 2000; Ayub & Saleem, 2013; Islam et al., 2018; Sasikumar, 2008), reconocimiento o procesamiento de texto (El-Kassas et al., 2016; Oyarzo et al., 2014; Thanh-diane et

al., 2016; Wesson et al., 2017), y procesamiento de imágenes (Halbe & Joshi, 2015; Nedzved et al., 2013; Rosales-Morales et al., 2019).

Por otra parte, el uso de técnicas de inteligencia artificial como el procesamiento de imágenes y redes de aprendizaje profundo en conjunción con Patrones de Diseño de Interfaces de Usuario, representan una alternativa novedosa para generar un nuevo enfoque que integre los aspectos antes mencionados. En este sentido, el uso de técnicas de inteligencia artificial aplicadas en el desarrollo de software ya se aplicaron en (Rosales-Morales et al., 2019), una investigación previa que consistió en proponer un nuevo enfoque de desarrollo de software multi-dispositivo y multi-plataforma. Sin embargo, aunque en (Rosales-Morales et al., 2019) se soporta la generación de software usando imágenes que representan una interfaz de usuario a través de representaciones abstractas de UIDPs, existen limitaciones. Los UIDPs soportados son de uso general, no se definen para la generación de aplicaciones de dominios específicos como el educativo, e-commerce o social media. Tampoco se tiene soporte para la composición de UIDPs y carece del soporte para aplicaciones de Smart TV. Además, el tipo de red neuronal que se aplicó no fue una red neuronal de aprendizaje profundo, lo que limita el tiempo de respuesta y el tipo de imágenes soportadas.

Desde esta perspectiva, surge el siguiente planteamiento: ¿Cómo mejorar las técnicas de los enfoques existentes para tener un proceso de desarrollo de software más ágil e intuitivo? Por lo anterior, existe la necesidad de realizar investigación de nuevas técnicas de desarrollo de software, de manera específica en la generación automática de código. Estas nuevas técnicas de desarrollo de software además de integrar procesamiento de imágenes, redes de aprendizaje profundo y composición de patrones de diseño; tienen que considerar el desarrollo de software multi-dominio, multi-dispositivo y multi-plataforma a través de la representación de interfaces de usuario en imágenes reales a color, reales en blanco y negro, ideales y no ideales. Todo integrado para dar origen a un nuevo paradigma de desarrollo de software.

1.3 Objetivos

1.3.1 Objetivo general

Desarrollar un conjunto de nuevos métodos computacionales que permitan la generación de código fuente de aplicaciones multi-dominio, a través de técnicas de inteligencia artificial y composición de patrones de diseño de interfaces de usuario.

1.3.2 Objetivos específicos

- Analizar las técnicas de inteligencia artificial (procesamiento de imágenes, reconocimiento de patrones, redes neuronales, aprendizaje profundo) para seleccionar las adecuadas aplicables al reconocimiento de elementos en imágenes que representen interfaces de usuario.
- Realizar un análisis comparativo de los algoritmos identificados en el estado del arte para el procesamiento y detección de patrones en imágenes.
- Diseñar un conjunto de prestaciones de servicio de los nuevos métodos computacionales para la generación de software multi-dominio y multi-dispositivo.
- Desarrollar los nuevos métodos computacionales para el reconocimiento de patrones en imágenes para la generación de software multi-dominio y multi-dispositivo a partir del uso de técnicas de inteligencia artificial.
- Integrar el conjunto de métodos desarrollados a través de una herramienta como prueba de concepto de los métodos.
- Validar y evaluar los métodos computacionales utilizando diversos casos de estudio.

1.4 Hipótesis

El desarrollo un conjunto de nuevos métodos computacionales para la generación de código a partir del procesamiento de imágenes, reconocimiento de patrones,

aprendizaje profundo y redes neuronales permitirá ofrecer mecanismos para agilizar y facilitar el desarrollo de aplicaciones multi-dominio, multi-dispositivo y multi-plataforma.

1.5 Justificación

La motivación para la realización de este trabajo surge con la finalidad de proveer una novedosa forma de desarrollar software a través del uso de técnicas de inteligencia artificial, las cuales tienen poca aplicación en el campo de la ingeniería de software. El uso de técnicas y/o métodos que permitan el desarrollo de aplicaciones multi-dominio, multi-dispositivo y multi-plataforma a partir del uso de técnicas de inteligencia artificial como el procesamiento de imágenes, reconocimiento de patrones, aprendizaje profundo y el uso de redes neuronales en imágenes digitalizadas (previamente delimitada ya sea una interfaz gráfica de usuario dibujada a mano alzada o un diagrama UML por mencionar algunos ejemplos), asegura un avance importante en la manera de desarrollar sistemas computacionales. Al revisar las técnicas y herramientas para el desarrollo de software, se observa claramente que sus características son limitadas, es decir algunos permiten la generación de código a partir de la transformación de esquemas (modelos de casos de uso, meta-datos por mencionar algunos), otros más permiten que el código generado se implemente en dispositivos móviles y en algunos otros casos se genera software orientado a un solo dominio en específico. Sin embargo, no se ofrece algo innovador, de ahí surge lo novedoso de esta propuesta ya que, hasta el momento, se desconoce la existencia de algún enfoque, técnica u herramienta de aplicaciones multi-dominio que proporcione toda la funcionalidad que esta propuesta plantea, tal como la generación de código a partir de una imagen digitalizada y composición de patrones solo por mencionar una de las funcionalidades más importantes, lo que conlleva a la obtención de los siguientes beneficios:

- ✓ Agilizar y facilitar el desarrollo de aplicaciones multi-dominio.

- ✓ Generar únicamente el código fuente de los componentes que se consideren necesarios y que previamente se identificaron dentro de alguna imagen debidamente delimitada.
- ✓ Generar el código fuente de una interfaz gráfica de usuario completa a partir de una imagen debidamente delimitada a través del uso de composición de patrones de diseño de interfaces de usuario.

1.6 Contribución al conocimiento

Las principales contribuciones de esta investigación son las siguientes:

- Un nuevo enfoque para la generación de aplicaciones multi-dominio basado en técnicas de inteligencia artificial y composición de patrones de diseño de interfaces de usuario.
- Diseño de una arquitectura de software para el desarrollo de un sistema de generación automática de aplicaciones multi-dominio.

Capítulo 2 Estado del arte

En esta sección, se presentan los trabajos relevantes en el área de generación automática de código para el desarrollo de software multi-dominio. Estos trabajos de investigación se clasificaron en enfoques de desarrollo multi-dispositivo, herramientas de desarrollo de aplicaciones multi-dispositivo, y desarrollo de software multi-dominio.

2.1 Enfoques de desarrollo multi-dispositivo

Diferentes técnicas y métodos se reportan en la literatura para el desarrollo de software, el objetivo de las diferentes técnicas es facilitar y agilizar el proceso de desarrollo. En este sentido, (Francese et al., 2016) propuso una metodología junto con un método de notación gráfica para el desarrollo de aplicaciones móviles llamado MicroApps. Los usuarios de MicroApps construyen aplicaciones directamente desde sus teléfonos móviles mediante la selección táctil y siguiendo un enfoque de flujo de datos. Mientras en (El-Kassas et al., 2016), se propuso un nuevo enfoque de conversión de códigos utilizando XSLT y expresiones regulares. El enfoque se llama Integrated Cross-Platform Mobile Applications Development (ICPMD), con soporte para convertir el código de la aplicación Windows® Phone 8 (WP8) en código de la aplicación Android y viceversa. En (Thomas et al., 2017), se presentó un análisis de diversos enfoques de desarrollo de aplicaciones para dispositivos móviles. Se encontró que la necesidad de tratar con diversos estándares, protocolos y tecnologías de red; las capacidades limitadas de los dispositivos; las restricciones de tiempo impuestas por un mercado altamente dinámico y la existencia de distintas plataformas de hardware y software hace necesario el desarrollo de nuevas prácticas y metodologías específicas para el desarrollo de aplicaciones para dispositivos móviles. Desde otra perspectiva, en

(Molina Ríos et al., 2017) se presentó un análisis de metodologías para el desarrollo de aplicaciones Web, entre las metodologías analizadas se encuentran Metodología de Diseño de Escenarios Orientado a Objetos en Hipermedia (SOHDM), Método de Diseño para Sitios web (WSDM), UML basado en Ingeniería Web (UWE), y Método de Diseño e Hipermedia Orientado a Objetos (OOHDM). Los autores concluyen que la metodología más utilizada es la metodología ágil, debido a que extiende la productividad, reduce la sobrecarga de procesos y mejora la gestión de riesgos. Desde otra perspectiva, en (Narkawicz et al., 2017) presentó MINERVA un proceso de desarrollo de software para los componentes de software de seguridad crítica de los sistemas ciber físicos. El proceso se basa en métodos formales para validar rigurosamente el código frente a sus requisitos a través de un lenguaje de especificación formal para describir los algoritmos. Mientras en (Koc & Aydos, 2017), se propuso un modelo de procesos de desarrollo de software de Scrum de seguridad mejorada (Scrum de confianza) aprovechando tanto las actividades de seguridad como el marco de Scrum que tiene una rápida adaptación y un ciclo iterativo.

En (Carrizo & Alfaro, 2018) se presentó un método de aseguramiento de la calidad, como estrategia y actividad en el desarrollo de un proyecto de software, que a través de sus diferentes actividades, tareas y conceptos, se adaptan a la metodología de desarrollo o en diferentes etapas del desarrollo del proyecto. Por su parte, (Ileana & Cabrera, 2018) presentó un *framework* de desarrollo de Proyectos Sociotecnológicos basado en la notación de metamodelos de procesos de ingeniería de software (SPEM 2.0) adaptación del modelo de proceso MeRinde (Metodología de la Red Nacional de Integración y Desarrollo de Software Libre) para la disciplina Requisitos, Análisis, y Diseño. Por su parte, (Mills et al., 2018) propuso un nuevo paradigma de desarrollo de software ágil, centrado en la arquitectura y un marco de herramientas complementarias que facilita el desarrollo de software libre de estrés y errores en todas las etapas de desarrollo de software que es altamente agnóstico a las reglas de negocios específicas de una organización. Desde otra perspectiva, en (Montero Molina et al., 2018) se presentó un análisis comparativo metodologías ágiles y tradicionales. Se concluyó que las metodologías de

desarrollo tradicionales tienen una disciplina de trabajo fundamentada en la documentación sobre el proceso de desarrollo de software, se hace hincapié en la planificación global y total de todo el trabajo, y una vez que esté detallado, comienza el ciclo de desarrollo de software; caso contrario a lo que respecta a las metodologías de desarrollo ágiles que muchas veces obvian la documentación y se centran en el trabajo, buscan el equilibrio entre proceso/esfuerzo.

En (Delia et al., 2019), se discutió diferentes alternativas para el desarrollo de aplicaciones móviles, se identificó y analizó diferentes características que afectan la elección del enfoque de desarrollo a utilizar. Los autores recomiendan seleccionar un enfoque de desarrollo de acuerdo a las diferentes plataformas: aplicaciones Web móviles, aplicaciones híbridas, aplicaciones interpretadas y aplicaciones de compilación cruzada. Por su parte, (Von Thienen et al., 2019) presento InnoDev un enfoque que combina *Design Thinking*, *Lean Startup* y *Scrum* para crear un proceso de desarrollo de software ágil que ofrece los productos y servicios innovadores orientados al cliente que requieren las empresas.

Los trabajos revisados proponen enfoques o métodos de desarrollo de software basados en técnicas tradicionales como diagramas de clases, casos de uso, diagramas de estados por mencionar algunos. Estas técnicas implican un nivel de conocimiento en tecnologías y herramientas especializadas por lo que los usuarios de los enfoques propuestos hasta el momento en la literatura son desarrolladores expertos. Por otra parte, se encontró que los enfoques de desarrollo de aplicaciones móviles deben considerar características específicas de diseño dada la gran variedad de dispositivos móviles en el mercado. Por lo anterior, se concluye que es necesario establecer técnicas de desarrollo de software intuitivas, fácil de implementar que agilicen el proceso de desarrollo de aplicaciones para dispositivos móviles como el uso de patrones de diseño de interfaz de usuario, procesamiento digital de imágenes, y redes de aprendizaje profundo.

2.2 Herramientas de desarrollo de aplicaciones multi-dispositivo

En la literatura, se reportan diferentes herramientas para la generación automática de código. En cuanto al desarrollo de aplicaciones móviles, (Freitas & Maia, 2015)

presentó el *Framework* denominado *JustBusiness*. Este *Framework* se basa en una arquitectura de *Naked Objects*. El objetivo principal de *JustBusiness* es respaldar el rápido desarrollo de aplicaciones comerciales basadas en Android™ a través de la generación automática de código, a partir de la asignación de clases de negocios. Por otro lado, (Rogai et al., 2015) propuso una solución para generar software para administrar contenido editorial para diferentes plataformas móviles. Se utilizó un Sistema de gestión de contenido para la administración de contenido a través de un sitio web. Desde una perspectiva diferente, en (Umuhoza et al., 2015) discutieron las mejores opciones para generar código de aplicaciones móviles a partir de la implementación de enfoques y marcos de desarrollo. De acuerdo con (Acerbis et al., 2015) se propuso una suite de herramientas: *Web Ratio Mobile Platform*, una plataforma de desarrollo basada en modelos y bajo código, basada en Eclipse que genera aplicaciones móviles multiplataforma para el desarrollo basado en modelos de aplicaciones móviles. La herramienta cuenta con verificación de modelo y generación de código completo que produce aplicaciones móviles multiplataforma listas para su publicación que explota el marco de trabajo Apache Cordova™. De forma similar, (Majchrzak et al., 2015) presentó MD2 como una solución para desarrollar aplicaciones comerciales multiplataforma. MD2 se basa en un enfoque de desarrollo dirigido por modelos, MD2 toma los requisitos de las aplicaciones de negocios para crear modelos y luego transformar esos modelos en aplicaciones multiplataforma. De acuerdo con (Marin et al., 2015) se presentó un *Framework* llamado *LIZARD*, que permite desarrollar aplicaciones para Windows® Phone OS y dos tipos de dispositivos Android™ (tabletas y teléfonos inteligentes). Las aplicaciones generadas utilizando *LIZARD* siguen las pautas del diseño de la interfaz del usuario y los patrones arquitectónicos y el diseño especificado por el fabricante del sistema operativo correspondiente. Mientras, (Christopher et al., 2016) presentaron el diseño y desarrollo de una aplicación móvil fácil de usar que usa el software *MIT App Inventor 2* para la adquisición de datos a través de sensores inalámbricos y la visualización de biopotenciales en dispositivos biomédicos. Desde una perspectiva diferente, (Núñez-Valdez et al., 2016) mostró un enfoque basado en modelos para generar y desplegar videojuegos multiplataforma de forma

automática a través de un meta-modelo y un DSL gráfico. Para la generación del código fuente, los autores propusieron una plataforma para el desarrollo ágil de juegos multi-plataforma utilizando modelos de abstracción de alto nivel. Como resultado, se obtuvo un enfoque apropiado para mejorar el proceso de desarrollo del juego a través del uso de MDE. Desde un enfoque similar, (Rieger, 2017) propuso Munster App Modeling Language (MAML), un marco de trabajo desarrollado bajo un enfoque basado en modelos. MAML utiliza un lenguaje específico del dominio para modelar los datos, las vistas, la lógica del negocio e interacciones de los usuarios. Finalmente, (Cortes-Camarillo et al., 2019) diseñó EduGene, un generador de aplicaciones educativas basado en UIDPs. Las aplicaciones de EduGene son compatibles tanto con la web como con cuatro sistemas operativos: Android™, Firefox® OS, macOS® y Windows® Phone.

La mayoría de las herramientas de desarrollo que se encuentran en la literatura reportan diferentes técnicas de desarrollo de software, en su mayoría basada en modelos abstractos. Además, se encontró que las herramientas no siempre soportan la generación automática de código fuente de la aplicación. Desde esta perspectiva, se concluye que existe la oportunidad de proponer herramientas que soporten la generación automática de código fuente de aplicaciones para dispositivos móviles que consideren características específicas de diseño de acuerdo a la plataforma y dispositivo, y que soporten la generación de código fuente.

2.3 Desarrollo de software multi-dominio

El desarrollo de software tiene aplicación en diversos dominios, algunas herramientas consideran técnicas para solucionar los principales problemas de diseño como el tamaño de pantalla, la navegación, o la correcta presentación de contenido. Desde esta perspectiva, en (Delgado et al., 2016) se presentó *WAINÉ*, una MBUIDE que admite la reutilización de los activos de IU definidos en sus modelos y explicó cómo sus técnicas de reutilización, parametrización y sub especificaciones, inclusión o repositorios compartidos, son capaces de utilizarse junto con el proceso de desarrollo de la interfaz de usuario. Desde una perspectiva similar, (Morgado & Paiva, 2016) presentó la herramienta *iMPACT*, *iMPACT* utiliza

ingeniería inversa aplicada a aplicaciones móviles para identificar patrones de interfaces de usuario presentes en una aplicación de prueba para verificar si se aplican correctamente. Por otro lado, (Dwivedi et al., 2016) introdujo un estudio sobre la técnica de minería de patrones de diseño, que se basa en técnicas de aprendizaje supervisado y en métricas de software. Durante el proceso de minería de patrones, se desarrolla un conjunto de datos basado en métricas. Posteriormente, se aplican las técnicas de aprendizaje automático, tales como *Layer Recurrent Neural Network* y *Random Forest*, para el proceso de minería de patrones. Mientras que (Thanh-diane et al., 2016) proporcionó un *Framework* y una metodología para definir y usar patrones MUI (interfaz de usuario multi-contexto) basados en UIPLML (Lenguaje de marcado de lenguaje de patrones de interfaz de usuario), un lenguaje de marcado XML que define explícitamente una referencia a cualquier texto de uso aplicable, como para qué usuario, para qué tarea, para qué dispositivo o plataforma, y en qué entorno. Para este propósito, se usa un meta-modelo que define la semántica de este lenguaje de patrones, un esquema XML se deriva de este meta-modelo para especificar la sintaxis de UIPLML y luego genera código para sistemas de información multi-plataforma.

Las capacidades de los UIDPs en el dominio médico se han explotado con fines de prevención y control de la salud. En este sentido, en (Mejía-figueroa et al., 2016) se presentó un estudio para tratar de determinar un modelo de usuario y modelo de interacción para interfaces adaptables para personas con trastorno del espectro autista basado en mediciones de la función ejecutiva. Lo anterior para resolver la usabilidad en aplicaciones de software hechas para usuarios con Autismo en mente, utilizables por unos pocos con el Trastorno del Espectro Autista. Por otra parte, (Cortes-Camarillo et al., 2017) presentó *Atila*, un generador de aplicaciones educativas basado en UIDPs. Las aplicaciones desarrolladas son compatibles con cuatro sistemas operativos: Android™, Firefox® OS, macOS®, Windows® Phone y la Web. El generador de aplicaciones crea proyectos según los dispositivos disponibles para cada plataforma. *Atila* usa UIDPs orientados al entorno educativo, y genera proyectos nativos para cada plataforma.

En otro contexto, se reportan diferentes aplicaciones y herramientas desarrolladas para el dominio educativo, e-commerce, o social media por mencionar algunos. En este sentido, (Alghabban et al., 2017) presentó una interfaz multimodal, una herramienta de *m-learning* basada en la nube con la que los estudiantes interactúan naturalmente en función de sus estilos de aprendizaje preferidos. La herramienta propuesta se adapta al perfil de cada alumno y al estilo de aprendizaje preferido a través del uso de una función multimodal que se proporciona en diferentes formatos (imágenes, audio y texto). La herramienta promueve las motivaciones e intereses de aprendizaje de los estudiantes en el aprendizaje de la lectura al proporcionar algunos mecanismos útiles para ayudar a su capacidad de lectura. Por su parte, en (Hasselbring & Steinacker, 2017) se propuso arquitecturas de microservicios para facilitar la escalabilidad, agilidad y fiabilidad en las plataformas de comercio electrónico. Otras investigaciones como en (Sohaib & Naderpour, 2017), se analizó el uso de servicios basados en la nube como SaaS, PaaS e IaaS para negocios de comercio electrónico. Los autores concluyeron que el SaaS es el mejor software de servicio para los negocios de comercio electrónico. Mientras, en (Jagtap & Hanchate, 2017) se propuso desarrollar aplicaciones de comercio electrónico móvil para Android™ a partir de una aplicación web existente. La aplicación para Android™ se integra con el servidor web para compartir la base de datos y los recursos.

El autor (Chuang, 2017) presentó MEMIS (Sistema de Instrucción Medio-móvil soportado por móvil), que incluye una aplicación móvil T & S (Maestros y Estudiantes) y un método pedagógico EMI para abordar las limitaciones de las tecnologías interactivas, fomentar un aprendizaje más colaborativo, facilitar una mayor interacción en clase en los cursos de conferencias grandes, y resolver la mala comprensión de las clases de los estudiantes en los cursos de EMI (inglés como medio de instrucción). MEMIS rastrea el progreso de aprendizaje de los estudiantes, de modo que el instructor y los estudiantes utilizan la aplicación para lograr una mayor participación en el aula y consultar los resultados de aprendizaje en clase y fuera de la clase; la aplicación multiplataforma funciona para cualquier dispositivo móvil. Por otra parte, en (Salahli et al., 2018) se discutió una aplicación

móvil para el aprendizaje de programación *Scratch*. La aplicación fue desarrollada para apoyar la educación en las escuelas secundarias para desarrollar las habilidades de programación de los estudiantes. La aplicación móvil se desarrolló para resolver dos problemas relacionados con la enseñanza del lenguaje de

programación *Scratch*. El primer problema está relacionado con los aspectos educativos de la programación *Scratch* comúnmente utilizada en las escuelas secundarias. El segundo problema es que *Scratch* no tiene aplicaciones móviles para su aprendizaje.

El desarrollo de aplicaciones multi-dominio revisado en esta sección permite identificar el uso de UIDPs para el diseño de interfaces de usuario, así como la generación parcial o total de una aplicación. Además, se encontró una variedad de técnicas aplicadas para el desarrollo de software que incluyen ingeniería inversa, aprendizaje supervisado, y lenguajes de marcado por mencionar algunas.

2.4 Análisis comparativo de trabajos relacionados

La presente sección contiene la descripción del análisis comparativo de los trabajos de investigación presentados a lo largo de este capítulo, donde las principales características relacionadas con esta propuesta son: 1) Dominio, 2) Generación de código fuente, 3) Soporte para patrones de diseño de interfaz de usuario y 4) multi-plataforma / Multi-dispositivo.

Tabla 2.1 Análisis de trabajos relacionados.

Autor	Dominio	Generación de código fuente	Soporta UIDPs	Multi-plataforma / Multi-dispositivo
(Freitas & Maia, 2015)	Negocios	Sí (Automático)	No	No/Sí
(Rogai et al., 2015)	Indefinido	Sí (Automático)	No	Sí/Sí
(Umuhoza et al., 2015)	Ingeniería de Software	Sí (Automático)	No	Sí/Sí
(Acerbis et al., 2015)	Desarrollo móvil	Sí	No	Sí/Sí
(Majchrzak et al., 2015)	Desarrollo de aplicaciones empresariales	Sí	No	Sí/Sí
(Marin et al., 2015)	Ingeniería Dirigida por Modelos	Sí	Sí	Sí/Sí
(Francese et al., 2016)	Indefinido	Sí (Automático)	No	No específico /Sí (Parcial)

Autor	Dominio	Generación de código fuente	Soporta UIDPs	Multi-plataforma / Multi-dispositivo
(El-Kassas et al., 2016)	Desarrollo móvil	Sí (Automático)	No	Sí/Sí (Parcial)
(Christopher et al., 2016)	Salud	No	No	No/Sí
(Núñez-Valdez et al., 2016)	Entretenimiento	Sí	No	Sí/No
(Delgado et al., 2016)	Ingeniería de software	No	No (Elementos de IU)	No/No
(Morgado & Paiva, 2016)	Ingeniería inversa	No	Sí	No/No
(Dwivedi et al., 2016)	Minería de patrones de diseño	No	Sí	No/No
(Thanh-diane et al., 2016)	Ingeniería de sistemas de información	No	Sí	Sí / Sí
(Mejía-figueroa et al., 2016)	Salud	No	Sí	No/ Indefinido
(Thomas et al., 2017)	Desarrollo móvil	No	No	Sí/Sí
(Molina Ríos et al., 2017)	Desarrollo Web	No	No	No/No
(Narkawicz et al., 2017)	Ingeniería de software	No	No	No/No
(Koc & Aydos, 2017)	Ingeniería de software	No	No	No/No
(Rieger, 2017)	Negocios	Sí (Automático)	No	Sí/Sí
(Cortes-Camarillo et al., 2017)	Educativo	Sí (automático)	Sí	Sí / Sí
(Alghabban et al., 2017)	Educativo	No	No específico	No específico / Móvil
(Hasselbring & Steinacker, 2017)	E-commerce	No	No	No/No
(Sohaib & Naderpour, 2017)	E-commerce	No	No	No/No
(Jagtap & Hanchate, 2017)	E-commerce	No	No	No/Móvil
(Chuang, 2017)	Educativo	No	No específico	Sí / Sí
(Carrizo & Alfaro, 2018)	Ingeniería de software	No	No	No/No
(Ileana & Cabrera, 2018)	Ingeniería de software	Si	No	No específico/No específico
(Mills et al., 2018)	Ingeniería de software	Sí	No	No específico/No específico
(Montero Molina et al., 2018)	Ingeniería de software	No	No	No específico/No específico
(Salahli et al., 2018)	Educativo	No	No específico	No específico / Móvil

Autor	Dominio	Generación de código fuente	Soporta UIDPs	Multi-plataforma / Multi-dispositivo
(Delia et al., 2019)	Desarrollo móvil	Sí	No	Si/Móvil
(Von Thienen et al., 2019)	Ingeniería de software	Sí	No	No específico/No específico
(Cortes-Camarillo et al., 2019)	Educativo	Sí	Sí	Si/Sí

De acuerdo con la literatura revisada en esta sección se concluye que las técnicas de desarrollo de software son diversas. En este sentido, se encontró que los enfoques de desarrollo ofrecen técnicas y herramientas para agilizar el proceso de desarrollo, sin embargo, los usuarios deben tener conocimientos previos. Lo anterior deja fuera a usuarios no expertos en desarrollo de software. Por otra parte, se encontró que algunas investigaciones de desarrollo móvil destacan la importancia de considerar características específicas del diseño como la plataforma y el dispositivo. Desde esta perspectiva, se concluye que técnicas de desarrollo como aprendizaje profundo, redes neuronales, procesamiento de imágenes y patrones de diseño de interfaz de usuario usadas en la ingeniería de software para la generación automática de software multi-dominio tienen poca aplicación. Por lo anterior, el planteamiento de un nuevo enfoque de desarrollo de software multi-dominio, multi-dispositivo y multi-plataforma basado en estas técnicas, que además de agilizar el proceso de desarrollo sea fácil e intuitivo es una propuesta factible.

Capítulo 3 Metodología

El contenido del presente capítulo se enfoca en describir las etapas que conforman la metodología empleada para desarrollar el presente trabajo de tesis. La metodología se basa en el método científico inductivo que comprende cuatro etapas: análisis, diseño, desarrollo y pruebas. El seguimiento de las cuatro etapas permite alcanzar la solución del problema propuesto.

A continuación, se describen las etapas de la metodología, en la figura 3.1 se presenta el esquema general de las principales actividades por etapa de la metodología.



Figura 3.1 Etapas de la metodología

3.1 Etapa de Análisis

Dentro de esta etapa, se consideró el análisis del conjunto de técnicas y metodologías reportadas en la literatura para el desarrollo de software. Sin embargo, este análisis se incluye en el capítulo 2 de este documento por lo que en esta sección no se describe.

Por otra parte, se realizó el análisis para definir los dominios de generación de aplicaciones y los lenguajes de programación soportados. Con respecto a la selección de los dominios, se encontró que las aplicaciones para los dominios Educativo, *e-commerce* y social media representan un crecimiento significativo en su uso diario. En el dominio educativo, es cada vez más frecuente adoptar nuevas estrategias para facilitar el proceso de enseñanza aprendizaje dentro y fuera del entorno educativo. Por lo que el uso de aplicaciones educativas disponibles en diferentes dispositivos son una alternativa importante, en conjunción con el uso de UIDPs que dada sus características tienen soporte para la implementación de actividades educativas de acuerdo a su contexto de uso. Respecto al dominio *e-commerce*, se encontró que el uso de este tipo de aplicaciones son una estrategia de marketing adoptada por diferentes empresas y negocios, por lo que requieren un diseño acorde que garantice el soporte para la implementación de características importantes como la utilidad, usabilidad y fiabilidad del contenido importantes en aplicaciones *e-commerce*. Por lo que el uso de UIDPs para el diseño de interfaces que garanticen una adecuada experiencia de usuario son una alternativa idónea. Finalmente, el dominio *social media* representa una parte importante en las actividades diarias del ser humano, entonces el desarrollo de aplicaciones *social media* toma relevancia dentro de la ingeniería de software. Por lo que es necesario contar con mecanismos que faciliten su desarrollo tomando en cuenta características propias de los medios sociales.

Para la selección de las plataformas y lenguajes de programación, se consideró aquellas plataformas con mayor uso, destacando los dispositivos móviles (Smartphone y Tablet), seguido por el navegador Web y recientemente las Smart TV. Tomando en cuenta los dispositivos identificados, los lenguajes de desarrollo

se seleccionaron de acuerdo a su aplicación y uso, por lo que Android™, iOS® y en un menor porcentaje Firefox® OS están dentro de los más usados en dispositivos móviles así como el soporte para Smart TV; mientras que para Web, HTML5 es el más común.

Por otra parte, se realizó el análisis de las técnicas de procesamiento de imágenes y redes neuronales para seleccionar las adecuadas capaces de complementar el proceso de desarrollo de aplicaciones multi-dominio, multi-dispositivo y multi-plataforma.

En esta etapa también se realizó el análisis de los patrones de diseño adecuados para generar aplicaciones móviles de acuerdo a cada dominio: Educativo, e-commerce y social media.

Por otra parte, se delimitó el formato, calidad y contenido de las imágenes empleadas durante el procesamiento, evaluación y entrenamiento de la red neuronal artificial.

3.1.1 Análisis de patrones para generar aplicaciones multi-dominio.

En esta sección se presenta el análisis de patrones de diseño para generar aplicaciones móviles. Primero se describen los patrones para el dominio educativo, seguido del análisis para el dominio *e-commerce*, finalizando con el análisis del dominio *social media*.

3.1.1.1.1 Análisis de Patrones en el dominio educativo

Antes de definir el conjunto de patrones de diseño adecuados para generar una aplicación educativa y de cualquier dominio es necesario definir el tipo de aplicaciones existentes e identificar sus principales características. Partiendo de lo anterior, en el dominio educativo los tres tipos de aplicaciones que fueron seleccionados son: Blog, Wiki y MOOC. Para definir estos tres tipos de aplicaciones se consideró el uso de teorías de aprendizaje, donde los Blogs y los MOOC promueven teorías de aprendizaje colaborativo y el aprendizaje distribuido, respectivamente; mientras que el constructivismo social y el aprendizaje comunitario pueden ser promovidos a través de las Wikis. En la tabla 3.1 se

describen las principales características de las aplicaciones seleccionadas en el dominio educativo.

Tabla 3.1 Descripción y beneficios de las aplicaciones educativas.

Aplicaciones educativas	Descripción y beneficios
Blog	Utilizado para construir conocimiento y motivar la resolución de problemas, la reflexión y la comunicación de las emociones (Chu et al., 2012). Los blogs permiten a los usuarios publicar contenido, comentar y hacer observaciones sobre los mensajes de otros usuarios (Tess, 2013).
MOOC	Los <i>MOOCs</i> (<i>Massive Open Online Courses</i> , Cursos en Línea Abiertos Masivos) son populares en la educación superior porque permiten que la enseñanza sea más accesible a una audiencia global a través de una combinación de recursos educativos abiertos y aprendizaje en línea (Watson et al., 2016).
Wiki	Las <i>wikis</i> son una herramienta didáctica en línea que permite desarrollar procesos de gestión del conocimiento en la educación (Biasutti & El-Deghaidy, 2012). Una de las características principales de las <i>wikis</i> es que son beneficiosas en el desarrollo profesional (Hutchison & Colwell, 2012).

De acuerdo al tipo de aplicaciones descritas en la Tabla 3.1, es importante identificar los dispositivos móviles y asociarlos a cada tipo de aplicación para definir el conjunto de patrones validos en cada dispositivo y tipo de aplicación. En este sentido, para definir el conjunto de patrones de diseño que serían usados se realizó una revisión de la literatura y se identificaron diferentes categorías de patrones de diseño de interfaz de usuario propuestos por diversos autores, para los dispositivos móviles se encuentran los planteados por (Neil, 2014), (Tidwell, 2010), (Sheibley, 2013), (Toxboe, 2016) y (UNITiD, 2016), en el caso de las televisiones inteligentes se encuentra (Developer, 2013), (Android TV, 2015) y (Apple TV, 2016).

Partiendo de la revisión en la Tabla 3.2 se presenta el análisis realizado para relacionar las plataformas con los patrones de diseño de interfaz de usuario que tienen en común y los autores que proponen los patrones de diseño.

Tabla 3.2 Relación de patrones de diseño de interfaces de usuario por plataforma.

Plataforma	Patrones de diseño de interfaz de usuario	Autores
Móvil	Accordion, Wizard, Tag Bar, Tool Bar, Progress Bar, Search, Carousel, Dashboard, Dialog, Form, Gallery, Load Indicator,	Jennifer Tidwell, Mari Sheibley,

Plataforma	Patrones de diseño de interfaz de usuario	Autores
	List, Login, Tag Navigation, Notifications, Contextual Menu, Control Segments, Tables.	Anders Toxboe y UNITiD
TV	Accordion, Wizard, Tag Bar, Tag Navigation, Progress Bar, Button, Search, Collections, Divisor, Form, Load Indicators, List, Notification, Page Control, Expandable Selector, Tables, Detail View, Split View.	LG® Developer, Android™ TV y Apple® TV
Web	Accordion, Wizard, Search, Carousel, Dashboard, Dialog, Form, Gallery, Load Indicators, List, Login, Tag navigation, Notification, Contextual Menu, Control Segments, Tables	Jennifer Tidwell y Anders Toxboe

Una vez identificados los patrones por plataforma y dispositivos es importante asociar un contexto de uso a cada patrón de acuerdo con su funcionalidad, en este sentido los contextos de uso identificados en el ámbito educativo son los siguientes:

- 1) **Asistente:** Guía al usuario a través de un conjunto de reglas para configurar o realizar una tarea.
- 2) **Cuestionario:** Permite la generación de un cuestionario, encuesta, examen, entre otros.
- 3) **Curso:** Representa la estructura de un curso o materia, así como su descripción y contenido.
- 4) **Lista de contenido:** Hace referencia al conjunto de cursos o materias que contiene la aplicación.
- 5) **Login:** Se utiliza para administrar la sesión de un usuario.
- 6) **Menú:** Se utiliza para ver las opciones que se tienen sobre un elemento de la aplicación.
- 7) **Multimedia:** Se refiere a las imágenes, video y audios de la aplicación.
- 8) **Notificación:** Son aquellos mensajes de advertencia o error que emite la aplicación.

Ya que se identificaron los principales contextos de uso para el dominio educativo es posible asociar los patrones de diseño adecuados aprovechando la funcionalidad que cada patrón ofrece.

En la Tabla 3.3 se muestra la relación entre los contextos de uso, las plataformas y los patrones de interfaz de usuario. Esta clasificación ayudó a determinar los patrones que serían seleccionados para generar cada tipo de aplicación educativa.

Tabla 3.3 Relación de patrones de diseño de interfaces de usuario por contexto de uso y plataforma.

Contexto de uso	Plataformas	Patrones de diseño de interfaz de usuario relacionados
Asistente	Móvil, TV y Web	<i>Wizard</i>
Cuestionario	Móvil y Web	<i>Form</i>
	TV	<i>Wizard, Progress bar</i>
Curso	Móvil	<i>Accordion</i>
	TV	<i>Vista detalle</i>
	Web	<i>Accordion</i>
Lista de contenido	Móvil y Web	<i>List, Accordion</i>
	TV	<i>List, Colecciones</i>
Login	Móvil, TV y Web	<i>Login</i>
Menú	Móvil	<i>Contextual menu, pop-up menu</i>
	TV	<i>Expandable Selector, Tag Bar</i>
	Web	<i>Contextual menu</i>
Multimedia	Móvil	<i>Carousel, Gallery, Image grid</i>
	TV	<i>Gallery, Image grid</i>
	Web	<i>Carousel</i>
Notificación	Móvil y Web	<i>Dialog</i>
	TV	<i>Notification</i>

Como se mencionó anteriormente, asociar el contexto de uso con un patrón de diseño ayudo a identificar el conjunto de patrones de diseño válidos para generar cada tipo de aplicación educativa. En la Tabla 3.4 se describen los catorce patrones de diseño comúnmente empleados para desarrollar los tres tipos de aplicaciones educativas: Blog, Wiki y MOOC.

Tabla 3.4 Relación entre los patrones de diseño de IU y las aplicaciones educativas.

Patrones de diseño de IU	Descripción	Blog	MOOC	Wiki
Accordion	Expande y contrae información vertical u horizontalmente al hacer clic, mostrando las secciones más importantes y visualizando su contenido.	Sí	Sí	Sí
Wizard	Permite al usuario alcanzar una meta a través de una serie de pasos. El <i>Wizard</i> tiene como objetivo facilitar y racionalizar un proceso.	No	Sí	No

Patrones de diseño de IU	Descripción	Blog	MOOC	Wiki
Progress Bar	Permite al usuario ver el estado de una acción o una petición que realizará el sistema.	Sí	Sí	No
Carousel	Organiza un conjunto de elementos en una línea horizontal, proporciona controles de navegación y permite a los usuarios seleccionar una opción a través de una imagen.	No	Sí	No
Dashboard	Facilita la toma de decisiones utilizando un conjunto de operaciones que están a la vista y fáciles de percibir.	Sí	Sí	Sí
Dialog	Recibe una respuesta explícita del usuario. Aunque un Diálogo no ocupa toda la pantalla, requiere la respuesta del usuario para continuar.	Sí	Sí	No
Form	Recibe información del usuario, la cual es interpretada por la aplicación para realizar alguna tarea específica.	Sí	Sí	No
List	Guía al usuario a través de un conjunto de información que sea relevante.	Sí	Sí	Sí
Login	Proporciona al usuario un nivel de seguridad, porque impide el acceso a la información de la aplicación hasta que se ingrese un usuario y una contraseña válidos.	Sí	Sí	Sí
Contextual Menu	Ofrece la posibilidad de mostrar un conjunto de opciones al seleccionar un elemento, lo que permite al usuario interactuar con las capacidades de un elemento.	No	Sí	No
Notification	Despliega un mensaje enviado por la aplicación para el usuario, con información relevante sobre el estado o alguna acción realizada en la aplicación.	Sí	Sí	No
Splashscreen	Muestra una pantalla de bienvenida con un texto o una imagen.	Sí	Sí	Sí
Table	Permite al usuario presentar información compleja a través de filas y columnas.	No	Sí	Sí
Video	Componente gráfico que permite al usuario reproducir un videoclip.	No	Sí	Sí

Sin embargo, es importante identificar además las posiciones que cada patrón de diseño tiene permitido tomar dentro de la interfaz de usuario. Para esto, las posiciones validas dentro de una interfaz de usuario son: *body*, *right*, *left*, *top* y *foot*. Entonces, para los 14 patrones de diseño se identificaron las posiciones validas

dentro de una interfaz de usuario, dicha relación considera también el tipo de aplicación educativa tal como se observa en la Tabla 3.5.

Tabla 3.5 Relación de aplicaciones, UIDPs y su posición en la interfaz de usuario.

No.	UIDPs	Aplicación educativa			Posiciones en la interfaz de usuario				
		Blog	MOOC	Wiki	Head	left	body	Right	Foot
1	<i>Accordion</i>	✓	✓	✓	x	✓	✓	x	x
2	<i>Carousel</i>	x	✓	x	x	x	✓	x	x
3	<i>Contextual Menu</i>	x	✓	x	x	x	✓	x	x
4	<i>Dashboard</i>	✓	✓	✓	x	x	✓	x	x
5	<i>Dialog</i>	✓	✓	x	x	x	✓	x	x
6	<i>Form</i>	✓	✓	x	x	x	✓	x	x
7	<i>List</i>	✓	✓	✓	x	✓	✓	✓	x
8	<i>Login</i>	✓	✓	✓	✓	x	✓	x	x
9	<i>Notification</i>	✓	✓	x	x	x	✓	x	x
10	<i>Progress Bar</i>	✓	✓	x	x	x	✓	x	x
11	<i>Table</i>	x	✓	✓	x	x	✓	x	x
12	<i>Splashscreen</i>	✓	✓	✓	x	x	✓	x	x
13	<i>Video</i>	x	✓	✓	x	x	✓	x	x
14	<i>Wizard</i>	x	✓	x	x	x	✓	x	x

3.1.1.1.2 Análisis de patrones para el dominio de e-commerce

El uso de aplicaciones *e-commerce* se ha incrementado notablemente, muchas empresas están implementándolas como parte de sus estrategias de ventas aprovechando su disponibilidad a través de diversas plataformas y dispositivos móviles. Desde esta perspectiva, dada la variedad de dispositivos móviles existentes en el mercado es necesario considerar dentro del diseño de las aplicaciones *e-commerce* las características del dispositivo y plataforma. Por lo anterior, el uso de patrones de diseño de interfaz de usuario para el diseño y

desarrollo de aplicaciones *e-commerce* son una alternativa de solución por lo que en esta sección se presenta el análisis de UIDPs válidos.

Las aplicaciones *e-commerce* permiten realizar operaciones de comercialización de productos o servicios a través de medios electrónicos, con lo que es innecesaria la presencia física del comprador o vendedor. Existen diversos tipos de aplicaciones de comercio electrónico, sin embargo, para fines de este estudio se seleccionó las siguientes:

- 1) **B2C (Business to consumer):** En este tipo de aplicaciones el comercio es entre empresas y consumidores finales.
- 2) **C2C (Consumer to consumer):** Para este tipo de aplicaciones, el comercio se realiza entre consumidores finales. Algunos de los consumidores toman el papel como vendedores y otros como compradores.

Partiendo de los tipos de aplicaciones *e-commerce*, por cada tipo de aplicación se realizó una clasificación y se identificó el conjunto de patrones de diseño más usados en cada clasificación. Para lograr lo anterior fue necesario analizar un conjunto de aplicaciones existentes considerando el número de usuarios en cada aplicación y las plataformas para las cuales tiene disponibilidad. Las aplicaciones seleccionadas en *e-commerce* son las que se describen a continuación.

Amazon® Shop: permite a los usuarios de buscar, comparar precios, leer opiniones y realizar pedidos en Amazon® directamente desde su móvil. Gracias a ella, ya es posible acceder a cualquiera de las páginas de Amazon® desde una sola aplicación. Esta aplicación también permite al cliente escanear códigos de barras para conseguir información inmediata sobre el precio y la disponibilidad de los productos. Los clientes de Amazon® tienen acceso integral a su cesta, listas de deseos, opciones de pago y envío, historial de pedidos, Configuración 1-Clic (r) y a los beneficios de suscripción de Premium (Amazon compras, 2018).

Wish®: esta aplicación permite la compra de una amplia variedad de productos y con distintos proveedores. Dentro de sus principales características se encuentran la búsqueda y consulta de productos, leer las opiniones de otros clientes, ver fotos

y videos de sus compras, rastrear sus compras, servicio de atención a clientes, y cupones de descuentos (Wish, 2018).

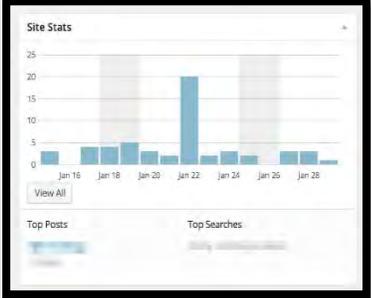
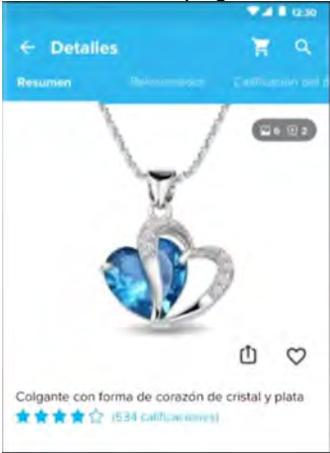
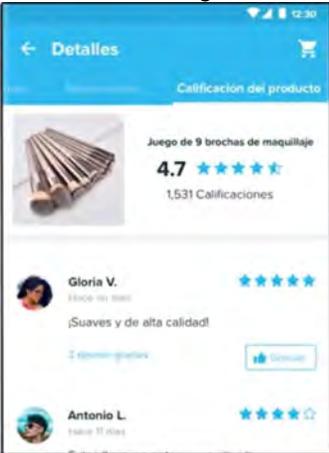
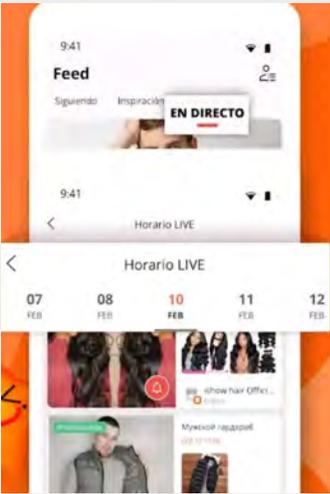
AliExpress™: esta aplicación ofrece la compra de diversos productos en línea, dentro de las características principales destacan la búsqueda fácil de productos a través de imágenes. El usuario tiene la opción de tomar una foto del artículo que quiere o subir una foto para encontrar una coincidencia exacta o artículos similares a lo que está buscando. Cuenta con cupones, descuentos, flash y ofertas. Además, permite la gestión de pedidos y notificaciones automáticas de envío (AliExpress, 2018).

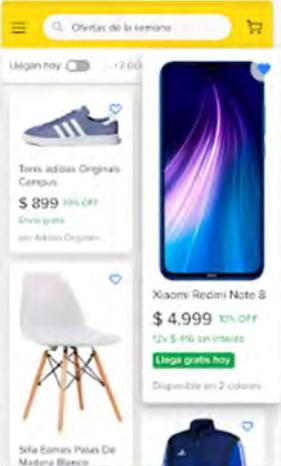
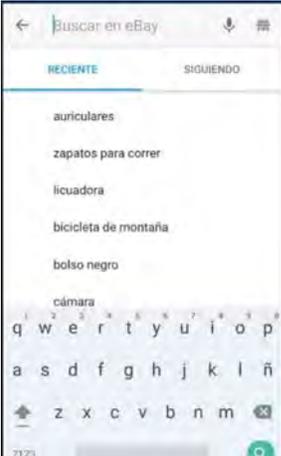
Mercado libre®: es una aplicación que permite a los usuarios la compra y venta de productos en línea. Dentro de las principales características ofrece a los usuarios guardar productos en una sección de favoritos para consultar después. También permite buscar productos con filtros específicos para el precio, ubicación, envío gratis, entre otros. Permite además comparar precios y leer opiniones ®de otros compradores. Para la venta de productos ofrece a los usuarios la opción de vender productos nuevos y usados, publicar anuncios gratis. Los productos se publican gratis en 4 sencillos pasos. Permite gestionar compras y ventas desde una cuenta de usuario. Además permite establecer comunicación con los compradores o vendedores a través de la mensajería instantánea (Mercado Libre, 2018).

eBay®: es una aplicación que permite a los usuarios la compra y venta de diversos artículos. Los usuarios tienen la oportunidad de buscar productos, encontrar ofertas y ahorros. Además, permite realizar ofertas en una subasta sobre la marcha, si el usuario quiere vender productos el proceso es sencillo a través de escanear el código de barras y la lista de productos (eBay, 2018).

Después de revisar el conjunto de interfaces de cada una de las aplicaciones de e-commerce, se identificaron los principales UIDPs que se presentan en cada interfaz. En la Tabla 3.6 se muestra el resultado de los UIDPs identificados por aplicación.

Tabla 3. 6 UIDPs identificados en las aplicaciones de e-commerce.

Tipo de aplicación	Aplicación	Patrones de diseño	Ejemplo de interfaces	
B2C	Amazon® Compras	Login Ranking Panel Chat Gallery Carousel Targets Comments Stats	<p style="text-align: center;">Comments</p> 	<p style="text-align: center;">Stats</p> 
	Wish®	Login Ranking Targets Video Carousel Shopping cart Form Search Coupon Menu Product page List Tip a friend	<p style="text-align: center;">Product page</p> 	<p style="text-align: center;">Ranking</p> 
	AliExpress™	Login Targets Coupon Search List Video Pricing table Menu Master detail Product page Shopping cart Panel	<p style="text-align: center;">Video</p> 	<p style="text-align: center;">List</p> 

Tipo de aplicación	Aplicación	Patrones de diseño	Ejemplo de interfaces	
C2C	Mercado Libre®	Login Ranking Gallery Targets Carousel Chat Master detail	<p style="text-align: center;"><i>Target</i></p> 	<p style="text-align: center;"><i>Carousel</i></p> 
	eBay®	Search Menu Shopping cart Targets Login Coupon Panel	<p style="text-align: center;"><i>Search</i></p> 	<p style="text-align: center;"><i>Panel</i></p> 

3.1.1.1.3 Análisis de patrones para el dominio social media

El uso de aplicaciones de tipo social media es cada vez más común entre usuarios de diversas edades, géneros y gustos, gracias a la amplia variedad de dispositivos móviles que las soportan. Por otra parte, el diseño y desarrollo de este tipo de aplicaciones se agiliza si se consideran las características y funcionalidades que se desean implementar. Adicionalmente, si se aprovechan las características de los patrones de diseño de interfaz de usuario el proceso de desarrollo sería reducido en tiempo y costo, pero se conservaría la calidad del producto final.

Dicho lo anterior, se realizó el análisis patrones usados en aplicaciones social media para lo cual se delimito el conjunto de aplicaciones que serían analizadas en el dominio social media. De acuerdo con la definición, las aplicaciones de tipo social media, el contenido se crea por los propios usuarios mediante el uso de las tecnologías de la Web que facilitan la edición, la publicación y el intercambio de información. Dentro de este tipo de aplicaciones destacan las siguientes:

Redes sociales: Son aplicaciones para comunicarse informalmente con otras personas, compartir intereses similares y encontrar personas.

Microblogging: Estas aplicaciones se caracterizan por permitir publicar entradas o actualizaciones muy breves en un sitio de redes sociales.

Streaming: Compartir contenido como videos, ya sea en tiempo real o a través de un catálogo.

A continuación, se describen en la Tabla 3.7 las aplicaciones seleccionadas en social media.

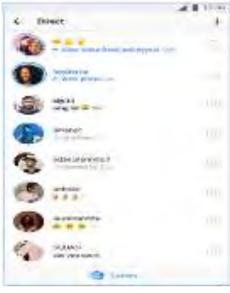
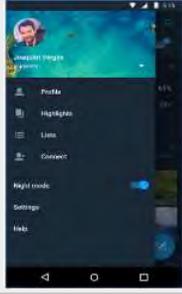
Tabla 3.7 Clasificación de aplicaciones social media.

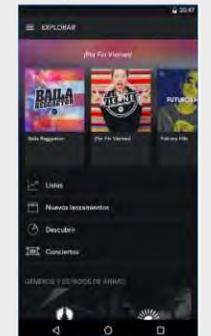
Tipo de aplicación	Aplicación	Descripción
Redes sociales	Facebook®	<i>Facebook®</i> es un sitio web y servicio de redes sociales donde los usuarios publican comentarios, comparten fotografías y enlaces a noticias u otros contenidos interesantes en la Web, jugar, chatear en vivo y transmitir videos en vivo. Incluso pedir comida con <i>Facebook®</i> si es lo que quiere hacer. El contenido compartido se hace público o se comparte solo entre un grupo selecto de amigos o familiares, o con una sola persona (Nations, n.d.).
	Instagram®	<i>Instagram®</i> es una aplicación de red social creada para compartir fotos y videos desde un teléfono inteligente. De forma similar a <i>Facebook®</i> o <i>Twitter®</i> , todos los que crean una cuenta de Instagram® tienen un perfil y una fuente de noticias (Moreau, n.d.).
Microbloggin	Twitter®	<i>Twitter®</i> es el sitio de redes sociales y noticias en línea donde las personas se comunican con mensajes cortos llamados <i>tweets</i> . <i>Twittear</i> es

Tipo de aplicación	Aplicación	Descripción
		enviar mensajes cortos a cualquiera que lo siga en <i>Twitter</i> [®] , con la esperanza de que sus mensajes sean útiles e interesantes para alguien de su audiencia. Otra descripción de <i>Twitter</i> [®] y <i>twittear</i> es el <i>microblogging</i> (Gil, n.d.).
	<i>Tumblr</i> [®]	<i>Tumblr</i> [®] debutó en febrero de 2007 como parte de la herramienta de <i>blogging</i> , herramienta <i>microblogging</i> y comunidad social. Es extremadamente fácil de usar y funciona en todos los sistemas operativos móviles (Gunelius, n.d.).
Streaming	<i>YouTube</i> [®]	Es fundado en 2005, <i>YouTube</i> [®] es hoy uno de los sitios de videos más populares en la Web. Se suben y comparten millones de videos aquí, desde avances de películas hasta videos de aficionados a los gatos, y todo lo demás (Collins, n.d.).
	<i>Netflix</i> [®]	<i>Netflix</i> [®] es un servicio de Video <i>On Demand</i> (VOD). Lo que significa que, a diferencia de los canales de televisión y las salas de cine, permite ver la película / programa de televisión siempre que lo desee (dado que el título está disponible con <i>Netflix</i> [®] o con el servicio de VOD que esté utilizando) (Anoop, 2019).
	<i>Spotify</i> [®]	<i>Spotify</i> [®] es un servicio de transmisión de música digital que le brinda acceso a millones de canciones, podcasts y videos de artistas de todo el mundo (Willings, 2020).
	<i>Deezer</i> [®]	<i>Deezer</i> [®] es una compañía pequeña de transmisión de música, pero funciona igual que las grandes, con solo algunas cosas diferentes, como cualquier otra empresa de transmisión por secuencias (De Wet, 2019).

En la Tabla 3.8 se presentan los principales patrones de diseño identificados en las aplicaciones de social media descritas en la Tabla 3.7.

Tabla 3.8 UIDPs identificados en aplicaciones móviles en el ámbito entretenimiento: Social media.

Tipo de aplicación	Aplicación	Patrones	Ejemplo de Interfaces	
Redes sociales	Facebook®	<p>Login</p> <p>Video</p> <p>Gallery</p> <p>Menu</p> <p>(horizontal and vertical)</p> <p>Comments</p>	<p>Horizontal Menu</p> 	<p>Gallery</p> 
	Instagram®	<p>Login</p> <p>Video</p> <p>Gallery</p> <p>Ranking</p> <p>List</p> <p>Menu</p>	<p>Video</p> 	<p>List</p> 
Microblogging	Twitter®	<p>Login</p> <p>Video</p> <p>Menu</p> <p>Gallery</p> <p>List</p>	<p>Video</p> 	<p>Vertical Menu</p> 
	Tumblr®	<p>Video</p> <p>Search</p> <p>Login</p> <p>Menú</p> <p>Panel</p> <p>Gallery</p> <p>Comments</p>	<p>Search</p> 	<p>Comments</p> 

Tipo de aplicación	Aplicación	Patrones	Ejemplo de Interfaces	
Streaming	YouTube®	<i>Video</i> <i>Ranking</i> <i>Gallery</i> <i>List</i> <i>Chat</i> <i>Comments</i> <i>Login</i>	<i>Video</i> 	<i>List</i> 
	Netflix®	<i>Login</i> <i>Menú</i> <i>Picker</i> <i>Video</i> <i>Gallery</i> <i>Carousel</i> <i>Ranking</i>	<i>Carousel</i> 	<i>Ranking</i> 
	Spotify®	<i>Gallery</i> <i>List</i> <i>Login</i> <i>Menu</i> <i>Carousel</i> <i>Music player</i>	<i>Music player</i> 	<i>Carousel</i> 
	Deezer®	<i>List</i> <i>Login</i> <i>Menu</i> <i>Gallery</i> <i>Music player</i>	<i>Gallery</i> 	<i>Music player</i> 

3.2 Etapa de Diseño

Las principales actividades durante esta etapa incluyen el diseño del proceso de generación de aplicaciones, el diseño de la arquitectura de software, la estructura del documento de configuración basado en XML. Adicionalmente, se incluye el

conjunto de *mockups* para el desarrollo de la herramienta generadora de aplicaciones móviles educativas. También, incluye la iconografía para representar patrones de diseño simples de aplicaciones educativas, e-commerce y social media; así como un conjunto de iconos para representar composiciones de patrones de diseño para estos tres tipos de aplicaciones. La iconografía de patrones simples y composiciones se usan para generar interfaces de usuario a mano alzada.

En esta etapa, se incluyen diseños de un conjunto de plantillas que permitirán distribuir los patrones de diseño dentro de la interfaz de usuario. Estos diseños parten del tipo de aplicación que se desarrollará y las posiciones validas de cada patrón de diseño analizado en la etapa de análisis.

3.2.1 Proceso de desarrollo de aplicaciones multi-dominio, multi, dispositivo y multi-plataforma.

El proceso de desarrollo de aplicaciones consta de tres etapas: (1) Análisis de la imagen, (2) Configuración y (3) Generación de código. Cada etapa está conformada por un conjunto de módulos tal como se aprecia en la figura 3.2

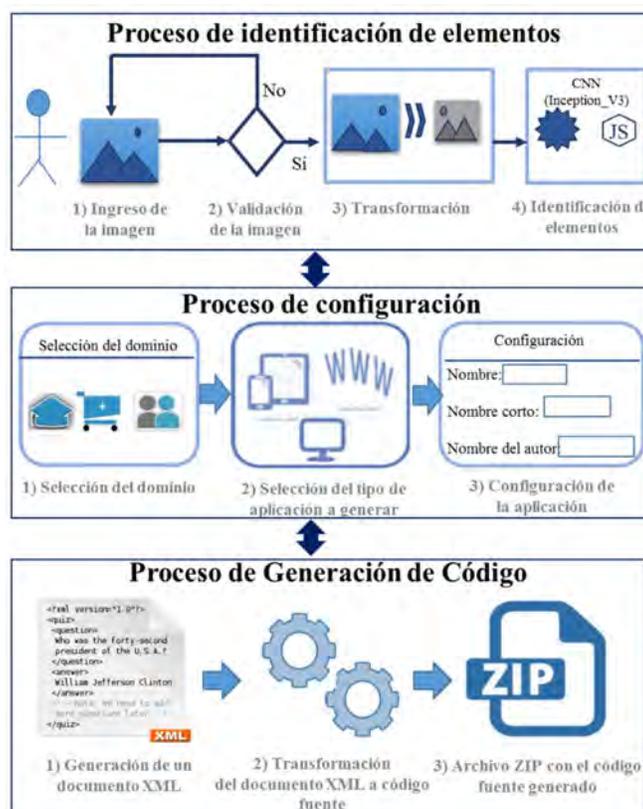


Figura 3.2 Etapas del proceso de desarrollo de software.

Cada etapa del proceso de desarrollo de software se describe a continuación:

Análisis de la imagen: esta etapa tiene como objetivo validar las imágenes que se soportan, así como identificar los UIDPs utilizando técnicas de Aprendizaje Profundo para posteriormente generar el código fuente de la aplicación. Para lograr el análisis se consideran tres pasos importantes los cuales se explican con más detalle a continuación.

- 1) **Entrada de la imagen a procesar:** las imágenes son interfaces de usuario generadas a mano alzada en hojas de papel o desde una herramienta de dibujo. En el caso de las imágenes generadas a mano alzada en hojas de papel, las imágenes se digitalizan a través de una cámara o un escáner, y se cargan en memoria. El formato soportado es PNG debido que tienen una mayor calidad de los datos. Cada imagen tiene dimensiones de 1050 x 650 píxeles. Los algoritmos no se limitan a esta única resolución; es posible adaptarlos a otras resoluciones automáticamente. Sin embargo, para esta investigación, se aplicó una única resolución.
- 2) **Validación de la imagen de entrada:** La validación de la imagen consiste en verificar que el formato de la imagen es PNG. Si el formato es correcto, el siguiente paso es la transformación.
- 3) **Transformación:** Este paso consiste en la implementación de un conjunto de algoritmos de procesamiento de imágenes para facilitar la identificación de los elementos de la imagen. Los algoritmos forman parte de un modelo de red neuronal convolucional; estos algoritmos permiten eliminar las sombras y el ruido de la imagen.
- 4) **Identificación de elementos:** mediante algoritmos de aprendizaje profundo a través de la red neural convolucional Inception_V3, se obtienen los diferentes componentes de la imagen. Los componentes de la imagen deben ser representaciones válidas de UIDPs para el desarrollo de aplicaciones educativas, de comercio electrónico y social media. El proceso de la red neural convolucional consiste en tomar un núcleo de la imagen y clasificarlo de capa a capa, y así consecutivamente hasta la clasificación completa del patrón en la capa de salida. El modelo Inception_v3 muestra los elementos identificados

ordenados por porcentaje de mayor a menor. Los cuatro porcentajes más cercanos a cada UIDP se guardan en un formato JSON. Los UIDPs soportados por la red neural convolucional para el desarrollo de las aplicaciones son Chat, Comment, Menu, Pricing table, Product Page, Ranking, Search, Shopping cart, Gallery, Video, Tip a friend, Coupon, Stats, y Panel. Además, la red neural convolucional soporta un conjunto de interfaces predefinidas llamadas composiciones.

Configuración: el objetivo principal de esta etapa es permitir la configuración del tipo de aplicación a generar (Web, móvil o TV), después de la identificación de los elementos en la imagen durante la etapa anterior. Esta etapa comprende los siguientes pasos.

- 1) **Selección del dominio:** de acuerdo con los UIDP identificados en la etapa anterior, ya que no todos los UIDP serán válidos para una aplicación educativa, de comercio electrónico o social media. El tipo de aplicaciones educativas soportadas son Blog, Wiki y MOOC, para el dominio e-commerce son B2C y C2C; para el dominio social media son *Microbloggin*, *Streaming*, y Redes sociales.
- 2) **Selección del tipo de aplicación a generar:** En este caso, se permite generar aplicaciones Web, aplicaciones para dispositivos móviles e inclusive de TV. Esta decisión se toma con la finalidad de que la generación de código se realice según las características de cada una de estas aplicaciones. Para la selección del tipo de aplicación se toma en cuenta los UIDPs identificados, no todos los UIDPs son válidos para cualquier dispositivo y/o plataforma.
- 3) **Configuración de la aplicación:** de acuerdo con el dominio y tipo de aplicación seleccionada en los pasos previos, se selecciona la configuración general de la aplicación. Algunos de los datos de configuración incluyen título principal, una cadena compuesta para 50 caracteres; lenguaje de desarrollo una cadena de 20 caracteres, los lenguajes soportados son para Web con HTML5, Android® con Java y iOS® con Objective –C. Otro dato de configuración es la versión de la aplicación, donde se aceptan solo números, puntos y guiones. Finalmente, datos

del autor de la aplicación como nombre, apellidos, e-mail, empresa y e-mail de la empresa.

Generación de código: durante esta etapa se realiza la generación del código fuente de la aplicación para lo cual se consideró un conjunto de pasos que permitan realizar la transformación de la información obtenida durante la etapa de análisis y la configuración. A continuación, se describen los pasos considerados para esta etapa.

- 1) **Generación de un documento basado en XML:** se genera un documento XML que contenga etiquetas para almacenar el conjunto de datos obtenidos durante la configuración, dichos datos serán: nombre de la aplicación, autor (s), etiquetas con la configuración de acuerdo a cada tipo de aplicación y lenguaje seleccionado, resolución, orientación y elementos de plantilla (header, body, footer, left y right). Finalmente, un conjunto de etiquetas con la representación de cada uno de los elementos identificados en la imagen.
- 2) **Transformación del documento XML a código fuente:** la transformación consiste primero en procesar el documento XML por medio de un documento XML Schema con el objetivo de validar la estructura de y garantizar la correcta generación del código fuente de la aplicación. Si la estructura es correcta se genera el equivalente en código por plataforma de los UIDPs descritos en el documento XML. La estructura generada se agrupa en clases y se almacenan en una carpeta por plataforma de acuerdo con el lenguaje de programación seleccionado por el usuario.
- 3) **Archivo ZIP con el código fuente generado:** con el código fuente generado se empaquetan las carpetas con los proyectos generados por plataforma y se crea un único archivo ZIP, el usuario descarga el código fuente para modificar o adecuar conforme a sus requerimientos y como mejor le convenga. En este sentido, los proyectos generados tienen la ventaja de exportarse a entornos de desarrollo de acuerdo a la plataforma de desarrollo. Los entornos de desarrollo válidos para exportar los proyectos son Dreamweaver u otro entorno de desarrollo similar en HTML5; para iOS® el entorno de Xcode™ y por último Android Studio™ para el proyecto Android.

3.2.2 Diseño de la arquitectura de software

A continuación, se presenta el diseño de la arquitectura propuesta para la herramienta generadora de aplicaciones multi-dominio. El diseño de la arquitectura se estableció en capas, este diseño permite la escalabilidad y el mantenimiento adecuado. Por otro lado, las tareas y funciones se distribuyen de manera equilibrada. La figura 3.5 muestra el diseño propuesto de la arquitectura.

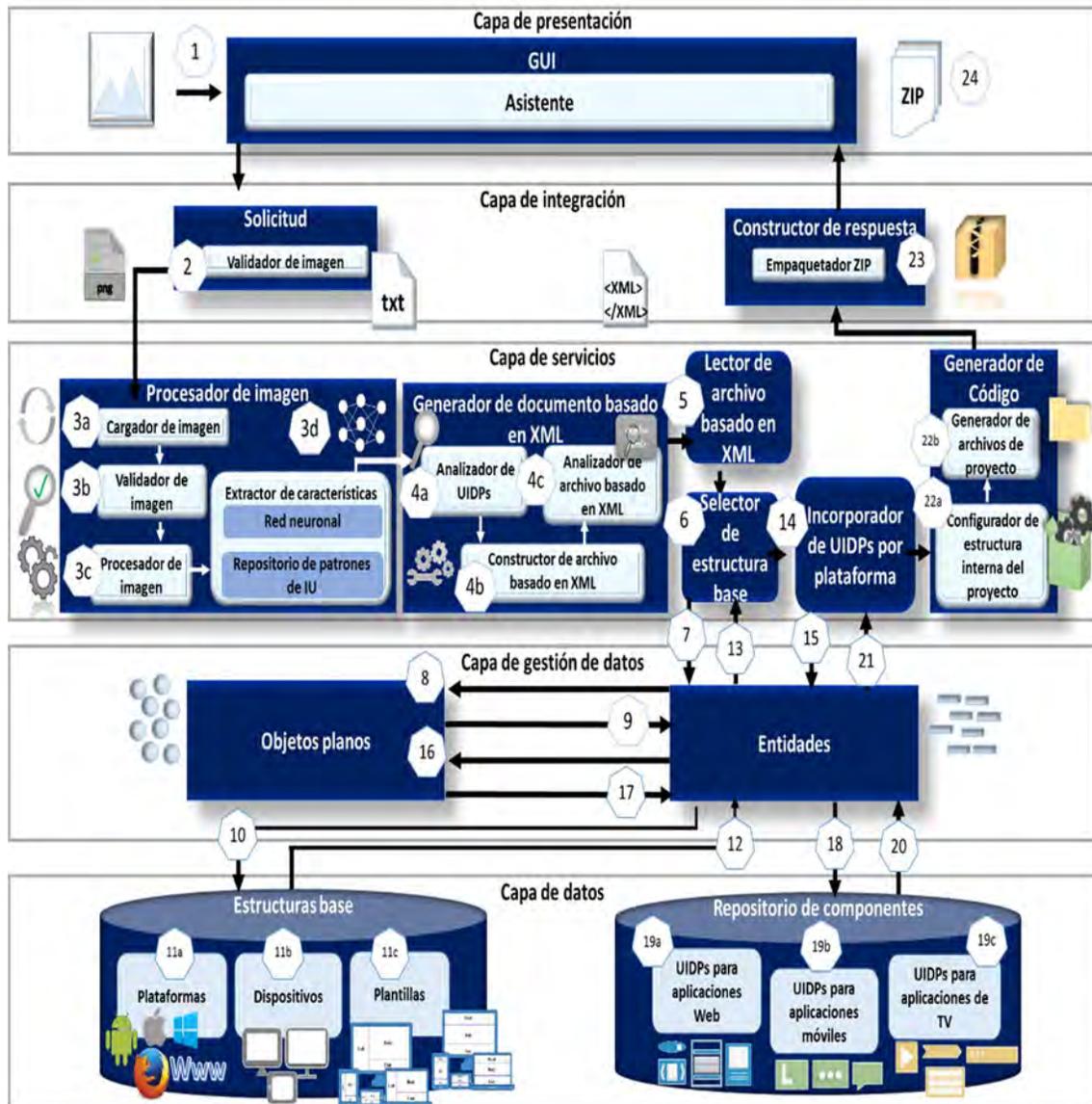


Figura 3.3 Diseño de la arquitectura.

A continuación, se describen las capas de la arquitectura de software propuesta en la figura 3.3.

Capa de datos. Esta capa contiene la estructura básica de cada plataforma y dispositivo, más un conjunto de UIDP que permiten al usuario generar la aplicación.

Capa de gestión de datos. Esta capa permite la comunicación entre la capa de servicios y la capa de datos. La capa de gestión de datos contiene un conjunto de objetos que representan de forma abstracta el conjunto de componentes necesarios para generar el código fuente de una aplicación educativa, e-commerce o social media. Por otro lado, la capa de gestión de datos proporciona una serie de entidades con operaciones de lectura / escritura que permiten el acceso a los componentes en la capa de datos.

Capa de servicios. Esta capa proporciona el conjunto de servicios (módulos) ofrecidos por la herramienta. Estos servicios representan diferentes funciones, como (1) procesamiento de imágenes y (2) generación de código fuente. La capa de servicios tiene acceso a la capa de integración, que recibe una imagen de entrada y devuelve un conjunto de archivos generados.

Capa de integración. Esta capa contiene los componentes necesarios para ubicar los servicios solicitados a través de la GUI. La capa de integración facilita el desarrollo de aplicaciones accediendo a todas las interfaces públicas proporcionadas por la herramienta generadora de aplicaciones.

Capa de presentación. En esta capa, la herramienta generadora de aplicaciones muestra el contenido usando HTML5. La capa de presentación no sabe qué eventos están teniendo lugar dentro de la herramienta y cómo se brindan los servicios. Solo se usa para mostrar la interfaz gráfica al usuario final.

Componentes

- **GUI:** Componente que representa la interfaz de usuario a través de la cual el usuario interactúa con la herramienta. El componente tiene un módulo llamado Asistente que se describe a continuación.
 - *Asistente:* este módulo permite al usuario seleccionar, capturar y generar el código fuente de una aplicación educativa, e-commerce o social media a través de una serie de pasos.

- **Solicitud:** componente que recibe las solicitudes de la capa de presentación, valida la información recibida y, si es correcta, envía la información a la capa de servicios al módulo correspondiente. El componente tiene un módulo llamado Validador de imagen descrito a continuación.
 - *Validador de imagen:* la función principal de este módulo es validar el formato y el tamaño de la imagen que el usuario envía a través de la GUI. En caso de identificar cualquier error, devuelve el mensaje de error a la capa de presentación.
- **Constructor de respuesta:** componente que recibe la respuesta de la capa de servicios, empaqueta la respuesta y la envía a la capa de presentación. El componente de solicitud tiene un módulo llamado paquete ZIP descrito a continuación.
 - *Empaquetador ZIP:* la función de este módulo es crear un archivo ZIP para agrupar el conjunto de archivos generados por el componente Generador de código.
- **Procesador de imagen:** este componente se desarrolló utilizando Inception_V3 y es responsable de realizar cuatro tareas principales. (1) Cargar la imagen recibida de la capa de integración, (2) Validar que la imagen represente una interfaz de usuario válida, (3) aplicar un conjunto de algoritmos de procesamiento de imágenes, (4) extraiga un conjunto de características de la imagen e identificar el conjunto de patrones que componen la interfaz. Para realizar las tareas antes mencionadas, el componente del procesador de imagen tiene los siguientes módulos.
 - *Cargador de imagen:* este módulo carga la imagen y la prepara para su procesamiento.
 - *Validador de imagen:* verifica a través de un conjunto de algoritmos si la imagen representa una interfaz de usuario válida.
 - *Procesador de imagen:* este módulo aplica un conjunto de algoritmos y métodos de procesamiento de imágenes para facilitar la extracción de las características de la imagen.

- *Extractor de características*: utiliza técnicas y métodos para obtener un conjunto de características que permiten identificar UIDPs en una imagen a través de una red neuronal convolucional. El módulo Extractor de características tiene dos sub módulos que se describen a continuación.
 - a) *Red neuronal*: determina a través del conjunto de características obtenidas de la imagen si alguno de los objetos pertenece a un UIDP (por ejemplo, *video*, *list*, *login* o *carousel*).
 - b) *Repositorio de patrones de IU*: contiene un conjunto de características que sirven como margen de referencia para la red neuronal para identificar cada patrón dentro de una imagen.
- **Generador de documento basado en XML**: Este componente crea el documento basado en XML, el documento basado en XML contiene la configuración de implementación para cada aplicación que se genera de acuerdo con el dispositivo y la plataforma de desarrollo. Los módulos de este componente se describen de la siguiente manera.
 - *Analizador de UIDPs*: la función principal de este módulo es leer los patrones de diseño identificados en la imagen desde un archivo JSON para generar el conjunto de etiquetas XML que representan cada patrón de diseño.
 - *Constructor de archivo basado en XML*: este módulo crea, estructura e integra el conjunto de etiquetas XML necesarias para crear una aplicación. El documento basado en XML es un documento de configuración que incluye información como el nombre de la aplicación, los autores, plataformas, dispositivos y patrones de diseño.
 - *Analizador de archivo basado en XML*: este módulo es responsable de validar la estructura y el contenido del documento basado en XML generado por el módulo de creación de archivos basado en XML mediante el uso de un esquema XML.
 - *Lector de archivo basado en XML*: la tarea principal de este módulo es procesar el documento basado en XML. El lector de archivos

basado en XML toma la información del documento basado en XML y genera los objetos necesarios para facilitar la lectura de los datos en la capa de gestión de datos.

- **Selector de estructura base:** este componente obtiene las estructuras base de las plataformas, dispositivos y plantillas de acuerdo con la información proporcionada por el archivo XML para construir la aplicación.
- **Incorporador de UIDPs por plataforma:** recupera el conjunto de estructuras de cada UIDP necesario para iniciar la aplicación en cada plataforma (web, móvil o TV) según los dispositivos seleccionados en el documento de configuración basado en XML.
- **Generador de código:** la tarea de este componente es configurar y crear las estructuras de la carpeta por aplicación y plataforma. El generador de códigos requiere dos módulos que se describen a continuación.
- **Configurador de estructura interna del proyecto:** el objetivo de este módulo es configurar los aspectos de implementación para garantizar el correcto funcionamiento de la aplicación. Para cada tipo de dispositivo y plataforma de desarrollo, el módulo configurador de estructura de proyecto interno aplica una configuración personalizada. Los principales aspectos de la configuración son la navegación, el nombre de la aplicación, los datos de los autores, la versión, entre otros.
- **Generador de archivos de proyecto:** este módulo genera todos los archivos necesarios de la aplicación que se generará. El generador de archivos del proyecto genera los archivos en función de la configuración realizada en el configurador de estructura del proyecto interno del módulo. Finalmente, el generador de archivos de proyecto separa los archivos de los diferentes proyectos según las plataformas de desarrollo y los organiza por carpetas.
- **Objetos planos:** este módulo representa los datos en objetos que son utilizados por la herramienta.
- **Entidades:** este componente contiene el conjunto de entidades que representan las operaciones de acceso a los datos solicitados por *MultiGene*.

- **Estructuras base:** este componente almacena un conjunto de estructuras necesarias para construir una aplicación educativa, las estructuras están separadas por plataforma y dispositivos. Los módulos que componen este módulo son los siguientes.
 - *Plataformas:* para crear una aplicación, cada plataforma de desarrollo requiere un conjunto de archivos de configuración, por lo que la tarea de este módulo es guardar la información requerida por cada plataforma.
 - *Dispositivos:* este módulo contiene los archivos de configuración necesarios para mostrar las aplicaciones de acuerdo con el tipo de dispositivo, como la resolución de pantalla.
 - *Plantillas:* este módulo contiene el conjunto de plantillas (estructuras) que permiten presentar de forma ordenada cada UIDP en la interfaz de usuario. Las plantillas se agrupan según el tipo de aplicación educativa, e-commerce o social media. Las posiciones de un UIDP dentro de la plantilla son *head*, *body*, *left*, *right* y *foot*. Las posiciones de los UIDPs dependen del tipo de aplicación que se está desarrollando.
- **Repositorio de componentes:** este componente contiene los UIDP separados por plataforma para facilitar su incorporación en los proyectos generados. Los UIDP soportados por *MultiGene* para generar aplicaciones educativas se almacenan en los siguientes módulos.
 - *UIDPs para aplicaciones Web.*
 - *UIDPs para aplicaciones móviles.*
 - *UIDPs para aplicaciones de TV.*

A continuación, se describe la interacción entre los componentes de cada capa del diseño de la arquitectura de la figura 3.5.

(1) La capa de presentación recibe la solicitud del usuario, es decir, la Imagen que representa una interfaz de usuario y la envía a la Capa de integración. (2) En la *Capa de integración*, el componente Solicitud a través de su módulo Validador de imagen verifica que la imagen tenga un formato y tamaño válidos. El formato de

imagen compatible con nuestra herramienta es PNG. El tamaño de imagen recomendado es inferior a 1 MB, mientras que el tamaño en píxeles recomendado es de 1050 x 650 píxeles. Si la imagen cumple con los requisitos, se envía a la *Capa de servicios*.

(3) En la Capa de servicios, el componente Procesador de imagen recibe la imagen para ser procesada usando Inception_V3. (3a) El módulo Cargador de imagen sube la imagen a la memoria para iniciar el procesamiento de la imagen. (3b) Una vez que se carga la imagen en memoria, el módulo Validador de imagen identifica que la imagen represente una interfaz de usuario válida. El módulo Validador de imagen aplica un conjunto de algoritmos tales como filtros, operaciones morfológicas y operaciones de convolución para limpiar la imagen y facilitar el proceso de identificación de elementos. Si al menos uno de los elementos en la imagen es un patrón de diseño, la imagen es válida y continúa con el proceso de identificación de elementos. (3c) El módulo Procesador de imagen aplica un conjunto de algoritmos tales como detección de bordes: horizontal y vertical. La imagen resultante se envía a la red neuronal convolucional para identificar cada patrón en la imagen, el módulo Extractor de características crea un archivo JSON con un conjunto de elementos identificados.

(4) Después de identificar cada elemento en la imagen y generar el archivo JSON con cada patrón, el componente Generador de documento basado en XML es responsable de construir el archivo basado en XML. (4a) Para construir el archivo basado en XML, el módulo Analizador de UIDPs realiza la lectura del JSON y extrae los patrones identificados, prepara el conjunto de etiquetas XML equivalentes para representar cada patrón y también verifica el orden de precedencia de cada patrón para establecer la navegación que tendrá la aplicación. (4b) Una vez establecidas las etiquetas XML de cada patrón, el módulo Constructor de archivo basado en XML genera el conjunto de etiquetas de configuración adecuadas para generar la aplicación. Las principales etiquetas contienen información como el nombre de la aplicación, los autores, las plataformas de desarrollo, dispositivos, navegación y las etiquetas de los patrones creados en el módulo Analizador de UIDPs. Todas las etiquetas se integran para generar el archivo de configuración basado en XML. (4c)

Una vez que se genera el archivo basado en XML, el módulo Analizador de archivo basado en XML recibe el documento basado en XML y valida la estructura y el contenido del documento utilizando un esquema XML.

(5) El módulo Lector de archivo basado en XML extrae la información del archivo de configuración generado en el módulo Constructor de archivo basado en XML. El módulo Lector de archivo basado en XML genera un conjunto de objetos a partir de los datos principales necesarios para generar una aplicación: Nombre de la aplicación, Autores de la aplicación, tipo de aplicación educativa, e-commerce o social media, plataformas de desarrollo, dispositivos, plantilla y patrones de diseño. Los objetos generados facilitan el acceso en la capa de gestión de datos desde el componente selector de estructura base.

(6) El componente Selector de estructura base solicita a la capa de gestión de datos las estructuras de código para cada plataforma de desarrollo, dispositivos y plantilla.

(7) Luego, el componente Entidades de la capa de gestión de datos prepara las entidades correspondientes de las plataformas, dispositivos y plantillas y (8) solicita al componente Objetos planos los objetos de acceso correspondientes para cada entidad. (9) El componente Objetos planos devuelve los objetos al componente Entidades. (10) El componente Entidades a través de operaciones de acceso solicita la información requerida al componente Estructuras base de la capa de datos. (11a) Luego, el módulo devuelve las estructuras correspondientes a las plataformas solicitadas, (11b) el módulo Dispositivos devuelve las estructuras de configuración del dispositivo y (11c) el módulo Plantillas devuelve las estructuras para presentar de forma ordenada en la interfaz de usuario cada patrón de diseño. (12) Las estructuras son recibidas por el componente Entidades quien las encapsula. (13) Posteriormente, el componente Entidades devuelve el conjunto de estructuras al componente Selector de estructura *base* en la capa de servicios.

(14) En el siguiente paso, el componente Incorporador de UIDPs *por plataforma* envía la solicitud a la capa de gestión de datos para obtener las estructuras de código de los patrones de diseño. (15) En la Capa de gestión de datos, el componente Entidades recibe la solicitud de la capa de servicios y prepara las entidades correspondientes a los patrones de diseño solicitados. (16)

Posteriormente, el componente Entidades solicita los objetos al componente Objetos planos, (17) el componente Objetos planos devuelve los objetos correspondientes al componente de Entidades. (18) Los objetos permiten a las entidades recuperar información de los patrones de diseño en la capa de datos. (19) En la *Capa de datos*, el componente Repositorio de componentes devuelve las estructuras solicitadas. (19a) El módulo UIDPs para aplicaciones Web devuelve las estructuras de componentes para una aplicación Web, (19b) el módulo UIDPs para aplicaciones móviles devuelve las estructuras componentes para una aplicación móvil y (19c) el módulo UIDPs para aplicaciones de TV devuelve las estructuras para componentes de una aplicación de TV. (20) El Repositorio de componentes devuelve las estructuras a la capa de gestión de datos, (21) el componente Entidades encapsula las estructuras y las devuelve a la capa de servicio.

(22) El componente Generador de código recibe las estructuras de datos necesarias para construir la aplicación educativa. (22a) Luego, el módulo Configurador de estructura interna del proyecto prepara la configuración para el despliegue de la aplicación educativa. Para cada dispositivo y plataforma, este módulo establece los parámetros para garantizar un inicio correcto de la aplicación educativa. Además, agrega los aspectos necesarios para garantizar la navegación de la aplicación de acuerdo con los patrones de diseño. Establece los aspectos finales con respecto a la distribución de los patrones en la interfaz de usuario de acuerdo con la información de las plantillas. (22b) Una vez establecidos los aspectos de implementación de la aplicación, el módulo Generador de archivos de proyecto crea los archivos necesarios con el código fuente de la aplicación educativa. Para cada plataforma de desarrollo el módulo agrupa los archivos de código en diferentes carpetas. Finalmente, la carpeta del proyecto con la aplicación funcional construida está disponible para la capa de integración.

(23) El componente Constructor de respuesta de la Capa de integración recibe la ruta a la carpeta del proyecto generado, y a través de su módulo llamado empaquetador ZIP se crea el archivo ZIP con el código fuente de la aplicación educativa. Finalmente, el componente Constructor de respuesta regresa como

respuesta un archivo ZIP a la Capa de presentación, donde el usuario descarga el archivo ZIP.

3.2.3 Diseño de la estructura del documento de configuración basado en XML

Durante el proceso de generación de aplicaciones descrito en la sección 3.1.2.1 se describe el uso de un archivo de configuración basado en XML, el cual es parte fundamental para la generación de las aplicaciones. Partiendo de lo anterior, en esta sección se describe la estructura básica del documento basado en XML necesaria para construir aplicaciones educativas.

EL documento consta de tres secciones principales, datos de configuración, configuración de despliegue y navegación. Cada sección se compone de un conjunto de etiquetas que almacenan los datos necesarios para construir una aplicación. Cada sección se describe a continuación.

Datos de configuración: esta sección incluye las etiquetas que almacenan el nombre de la aplicación, la información de los autores, la información de la organización y la versión de la aplicación.

Configuración de despliegue: En esta sección se agrupan las etiquetas relacionadas con los dispositivos, las plataformas, la orientación y las plantillas que se utilizan en el proyecto.

Navegación: en esta sección se incluye la lógica de navegación separada por páginas que contienen la información de los patrones de diseño con los cuales se genera la aplicación.

En la figura 3.4 se muestra el esquema de la estructura de etiquetas en cada sección descrita del documento de configuración basado en XML.

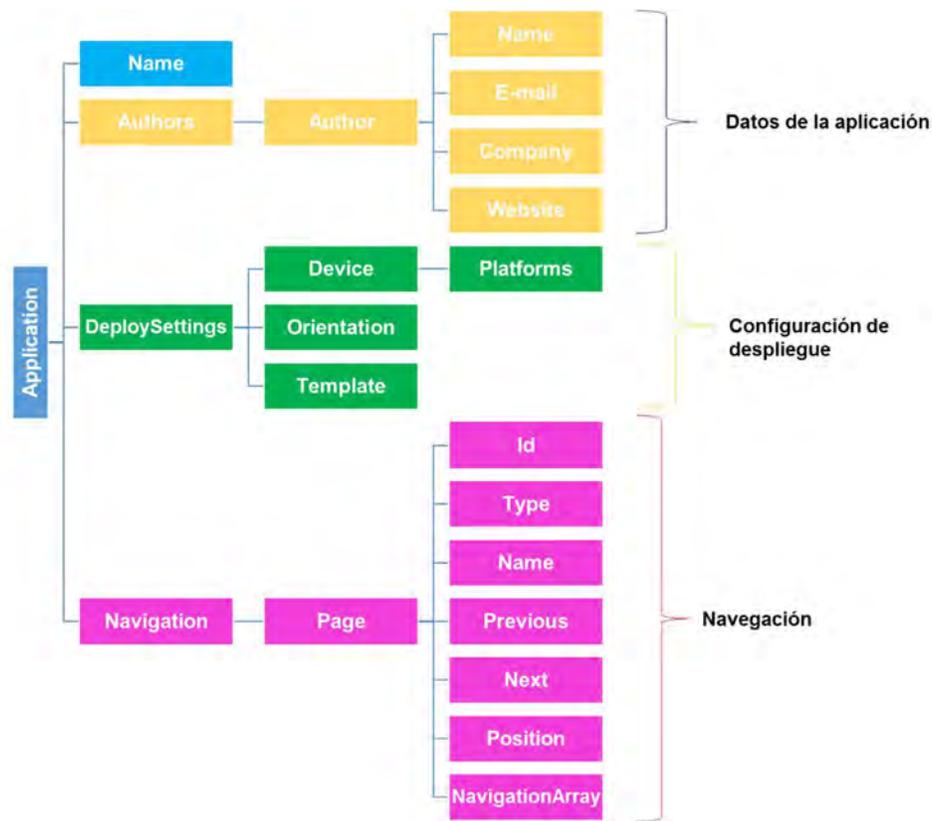


Figura 3.4 Estructura general del documento de configuración basado en XML.

3.2.4 Conjunto de *mockups* de la herramienta generadora de aplicaciones

Para integrar el conjunto de nuevos métodos computacionales para el desarrollo automático de aplicaciones es necesario plantear el diseño de una herramienta, el diseño de la herramienta se basa en *mockups* los cuales sirven de apoyo durante el proceso de desarrollo de cualquier aplicación. Los *mockups* permiten a diseñadores y desarrolladores mantener una adecuada retroalimentación sobre los cambios antes de finalizar el desarrollo de un producto de software. En este sentido, los *mockups* de la herramienta generadora de aplicaciones se diseñaron usando *Balsamiq Mockups 3* que es una herramienta disponible en una versión Web, así como para instalar en los sistemas operativos MacOS®, Windows™ y Linux®. Balsamiq cuenta con una amplia biblioteca de símbolos adecuados para diseñar una interfaz gráfica de usuario. Balsamiq permite exportar los *mockups* a los formatos XML, PDF, PNG (balsamiq.com, 2018).

El diseño de la herramienta tiene como objetivo implementar el proceso de desarrollo de aplicaciones multi-dominio, multi-dispositivo y multi-plataforma así como el diseño de la arquitectura propuesta en este documento. Partiendo de lo anterior se describe a continuación los *mockups* diseñados para las interfaces de la herramienta.

Diseño de la pantalla de inicio de sesión: Se planteó que los usuarios accedan a la herramienta de manera controlada a través de un inicio de sesión. La figura 3.5 muestra el diseño de la interfaz de acceso a la herramienta.

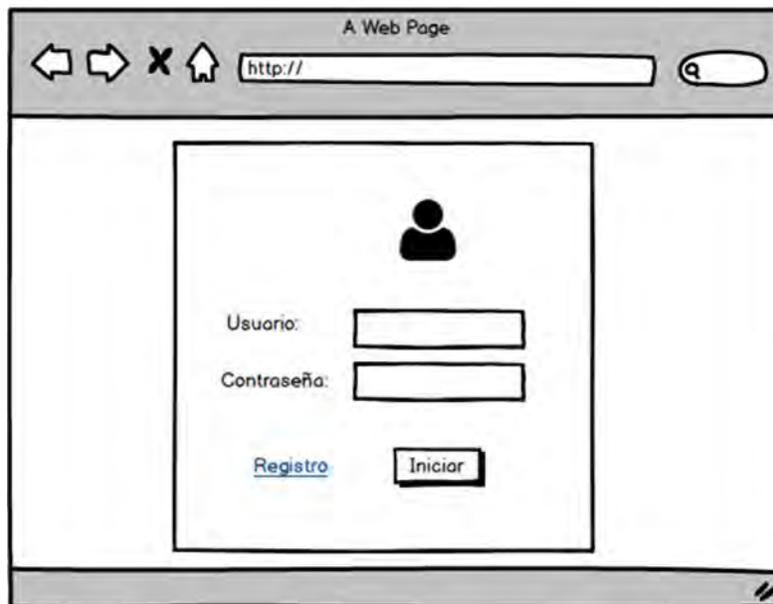


Figura 3.5 Mockup de la pantalla de inicio de sesión

Diseño de la página principal: Después de que los usuarios inicien sesión la pantalla principal incluye un menú de opciones y un carrusel con imágenes relacionadas al proceso de generación. Las opciones del menú principal son para la generación de aplicaciones desde dos enfoques: Ideal y a mano alzada. El *mockup* diseñado para esta interfaz se observa en la figura 3.6.

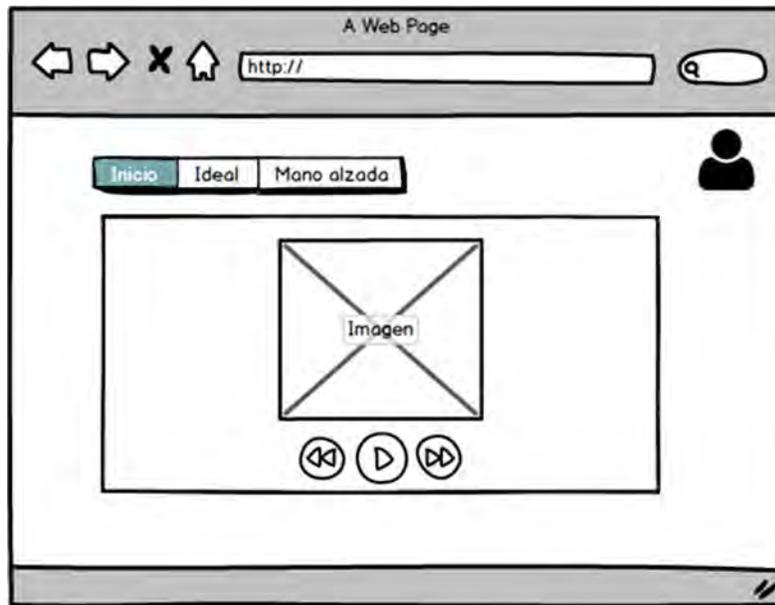


Figura 3.6 Mockup de la página principal.

Diseño de la página principal para el proceso ideal: Cuando el usuario seleccione la opción Ideal para generar una aplicación, en esta página se muestra un *Wizard*. El *Wizard* contiene un conjunto de pasos que facilitan el proceso de configuración de la aplicación. El inicio del proceso ideal desde el *Wizard* se observa en la figura 3.7, donde se presenta al usuario seleccionar el dominio de la aplicación que desea generar. Los dominios soportados como se mencionó son Educativo, e-Commerce y Social Media.

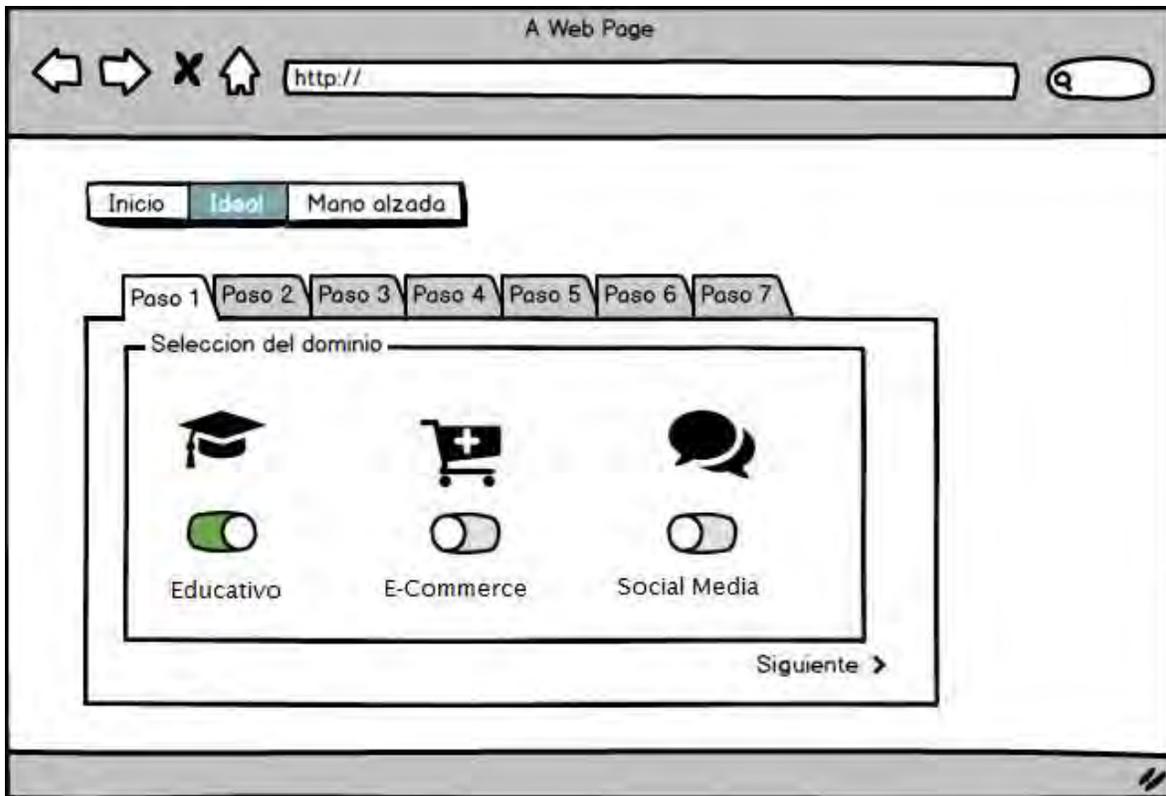


Figura 3.7 Mockup de la página principal del proceso ideal: selección del dominio.

Diseño de la página para la selección del tipo de aplicación en el proceso ideal:

El siguiente paso en el proceso ideal es la selección del tipo de aplicación dependiendo del dominio. En este punto, la figura 3.8 (a) muestra el tipo de aplicaciones para el dominio educativo: Blog, Wiki y MOOC, la figura 3.8 (b) el tipo de aplicaciones para el dominio e-commerce: B2C y C2C. Finalmente, la figura 3.8 (c) muestra los tipos de aplicaciones tentativas para el dominio Social Media: *social networks*, *microbloggin* y *streaming*.

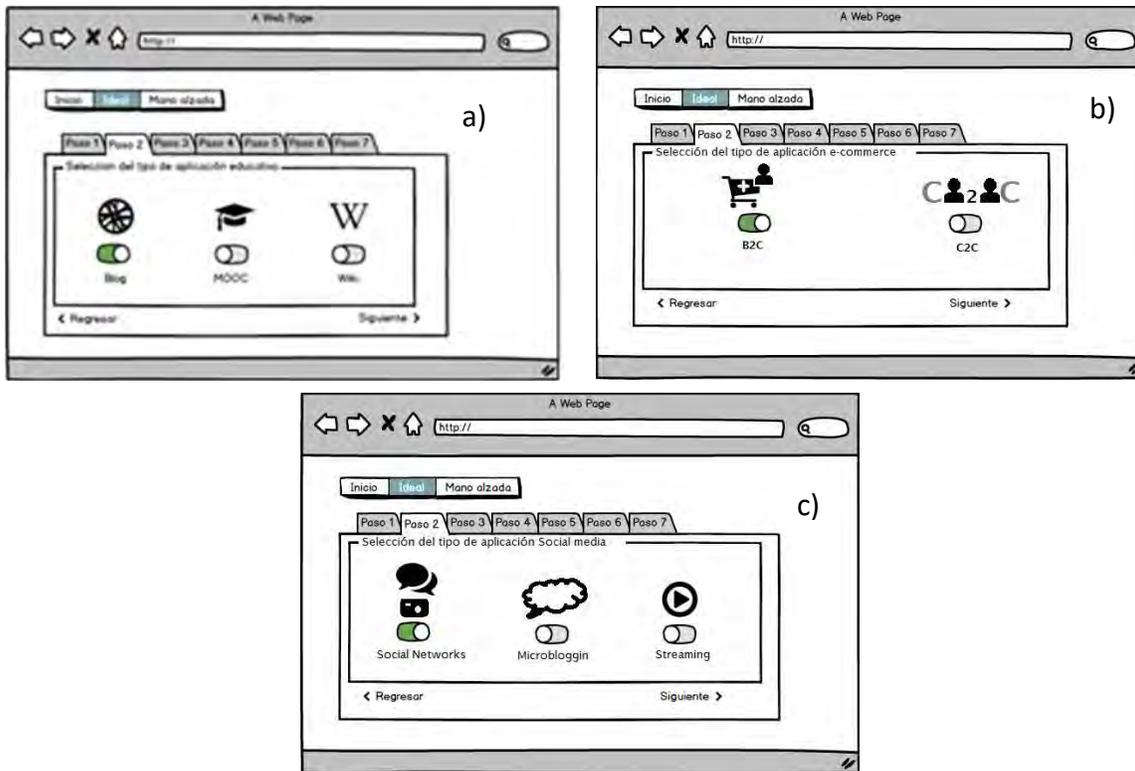


Figura 3.8 Mockup del proceso ideal: a) selección del tipo de aplicación educativa, b) selección del tipo de aplicación e-commerce y c) selección del tipo de aplicación social media.

Diseño de la página para la selección de una plantilla en el proceso ideal: una vez seleccionado el tipo de aplicación de acuerdo con el dominio el siguiente paso es la selección de una plantilla para presentar el contenido de la aplicación en la interfaz de usuario. En este sentido, se diseñó una página donde a través de un carrusel de imágenes se presenta a los usuarios los diferentes tipos de plantillas disponibles. Tal como se observa en la figura 3.9, el usuario visualiza el conjunto de

plantillas de las cuales selecciona la que mejor se adapte al tipo de aplicación que quiere generar.

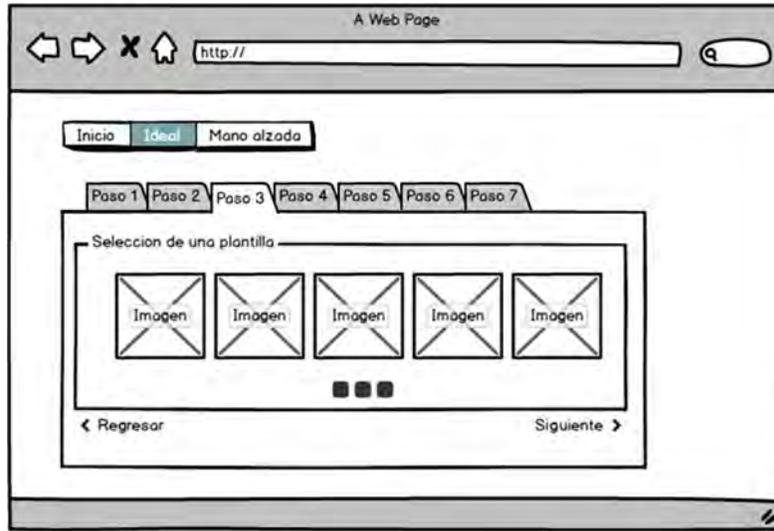


Figura 3.9 Mockup del proceso ideal: Selección de una plantilla

Diseño de la página para la selección de patrones en el proceso ideal: Para la selección de patrones el *mockup* diseñado presenta al usuario de manera ordenada el conjunto de iconos representativos de cada patrón y un *checkbox* para seleccionar el patrón. El diseño propuesto se observa en la figura 3.10.

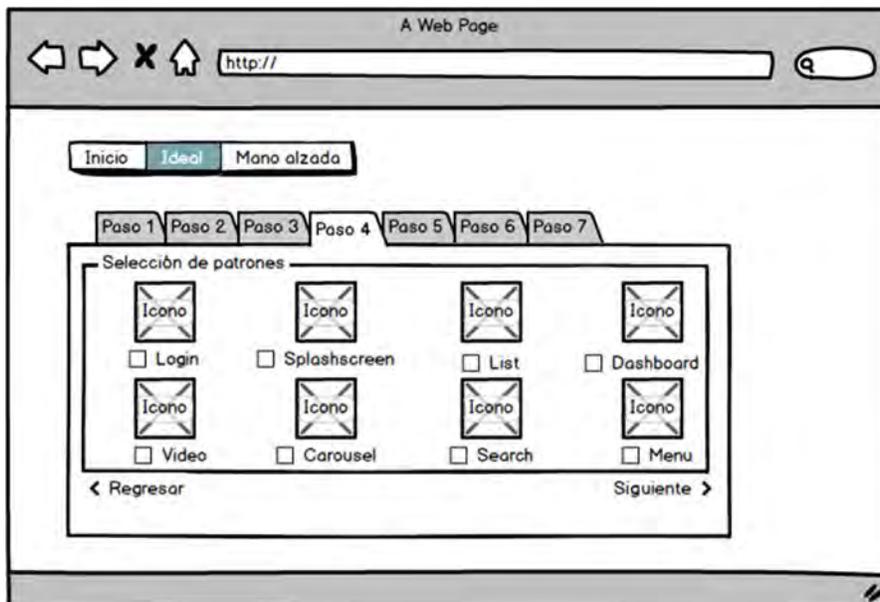


Figura 3.10 Mockup del proceso ideal: Selección de patrones.

Diseño de la página para seleccionar dispositivos y plataformas en el proceso ideal: Después de seleccionar el conjunto de patrones el siguiente paso es la selección del dispositivo y plataformas de desarrollo de la aplicación. En este caso el *mockup* diseñado presenta los tres tipos de dispositivos soportados y las plataformas de desarrollo de acuerdo a los dispositivos. En este caso para escritorio la plataforma de desarrollo considerada es Web, para Tableta/Smartphone son Web, Firefox® OS, Android™, Windows® Phone y iOS®. En TV son Web, Firefox® OS, Android™ y WebOS. La figura 3.11 muestra el *mockup* diseñado para este paso.

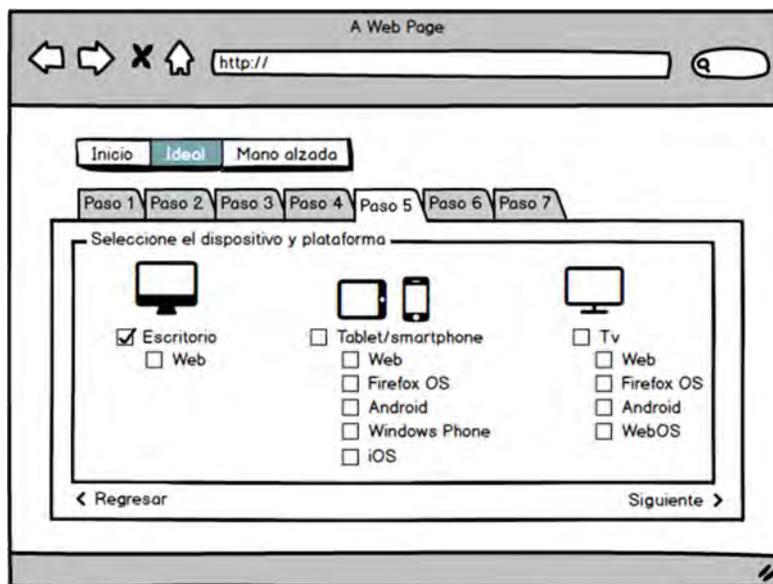


Figura 3.11 *Mockup* del proceso ideal: Selección de dispositivos/plataformas.

Diseño de página para la configuración en el proceso ideal: el *mockup* diseñado para el paso de configuración incluye los campos para capturar información necesaria para el correcto despliegue de las aplicaciones generadas. La información incluida se refiere al nombre de la aplicación, nombre corto, versión y datos sobre el autor de la aplicación. La figura 3.12 muestra el diseño planteado para la captura de la información.

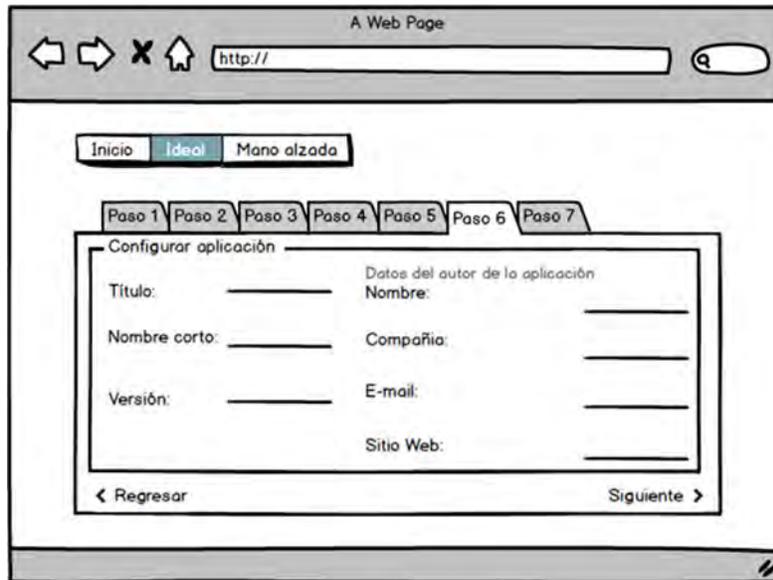


Figura 3.12 Mockup del proceso ideal: Configuración.

Antes de generar la aplicación y después de capturar los datos de configuración es importante que el usuario confirme que la configuración seleccionada es la definitiva. La figura 3.13 se muestra una ventana que notifica al usuario la configuración seleccionada para generar la aplicación, con la opción de regresar para cambiarla o aceptar la configuración e iniciar con el proceso de generación del código fuente de la aplicación.



Figura 3.13 Mockup del proceso ideal: Confirmar configuración.

Diseño de la página para descargar el código: para el paso final, se diseñó un mockup que primero presenta al usuario una imagen mientras de manera interna se generan los archivos necesarios de la aplicación tal como se observa en la figura 3.14 (a), y después presenta al usuario el enlace para descargar un archivo ZIP con el código fuente de la aplicación generada como en la figura 3.14 (b).

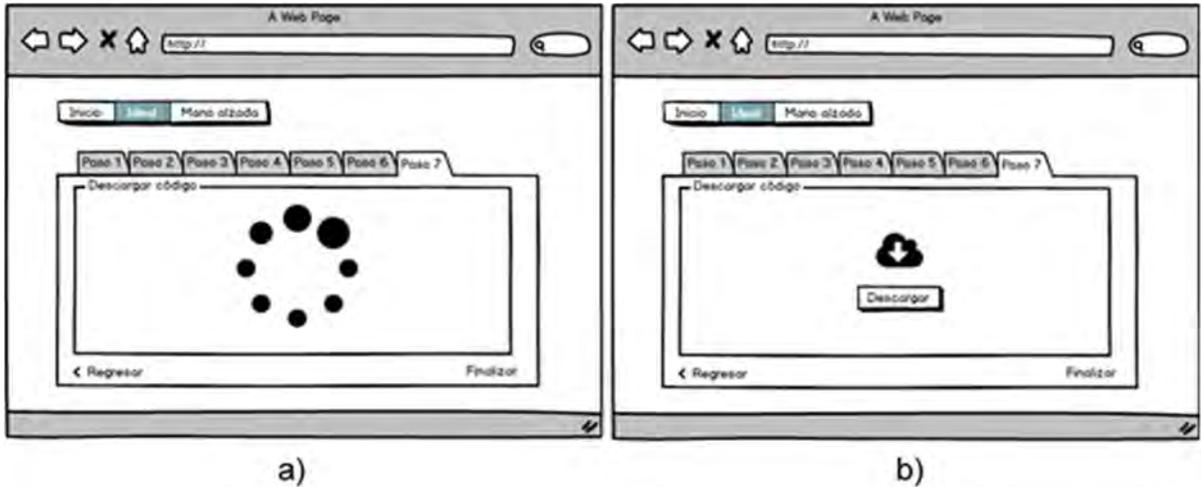


Figura 3.14 Mockup del proceso ideal: a) Generación de aplicación y b) Descargar código.

Diseño de la página principal del proceso a mano alzada: en este caso el *mockup* diseñado parte de la carga de una imagen, tal como se observa en la figura 3.15. Cabe mencionar que en este proceso los pasos para concluir la generación de la aplicación son 8, los cuales se distribuyen a través de un *Wizard* como en el caso del proceso ideal solo que cambia el orden de secuencia.

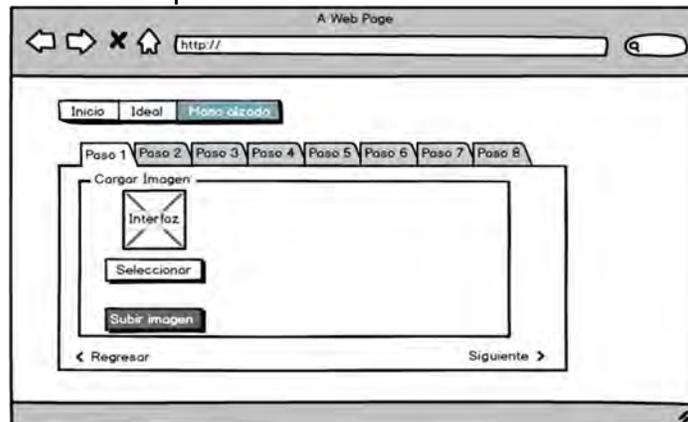


Figura 3.15 Mockup del proceso a mano alzada: Cargar imagen.

Una vez que el usuario selecciona una imagen, en el diseño se propone que a través de un botón se active el proceso de análisis e identificación de los patrones de diseño contenidos en la imagen. Al finalizar el proceso de identificación de elementos, se muestra al usuario la lista de patrones identificados en la imagen, y la opción de cambiar los patrones identificados o bien continuar con el siguiente paso que es la selección del dominio de la aplicación a generar. En la figura 3.16 se muestra como se presentaría al usuario la lista de patrones identificados.

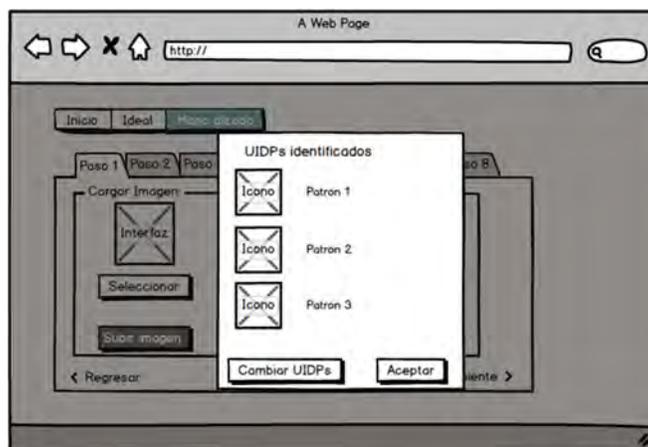


Figura 3.16 *Mockup* del proceso a mano alzada: Identificación de patrones.

Diseño de las interfaces para la selección de patrones, dominio y plantilla en el proceso a mano alzada: los diseños de los pasos siguientes después de cargar la imagen son similares a los pasos del proceso ideal, lo único que hace la diferencia es el orden de secuencia de los pasos. En la figura 3.17 a) se observa el diseño para la selección de patrones, en la figura 3.17 b) el *mockup* para seleccionar el dominio de la aplicación a generar y en la figura 3.17 c) el diseño para seleccionar una plantilla.

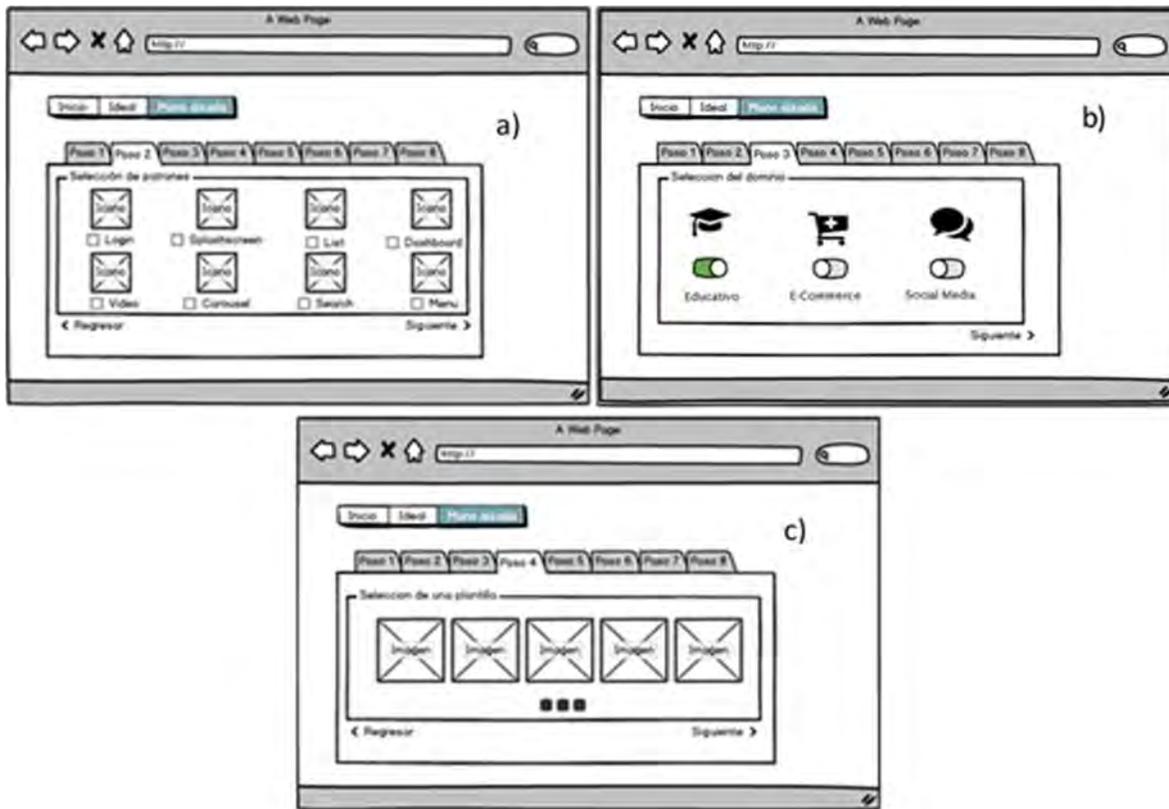


Figura 3.17 *Mockups* del proceso a mano alzada: a) Selección de patrones, b) Selección del dominio de la aplicación y c) Selección de una plantilla.

Diseño de las páginas para seleccionar el tipo de aplicación en el proceso a mano alzada: los *mockups* diseñados para este paso del proceso a mano alzada retoman el diseño de los *mockups* que se presentan en las figuras 3.8 (a), 3.8 (b) y 3.8 (c). Únicamente cambia el número de pasos tal como se observa en la figura 3.18 (a), 3.18 (b) y 3.18 (c).

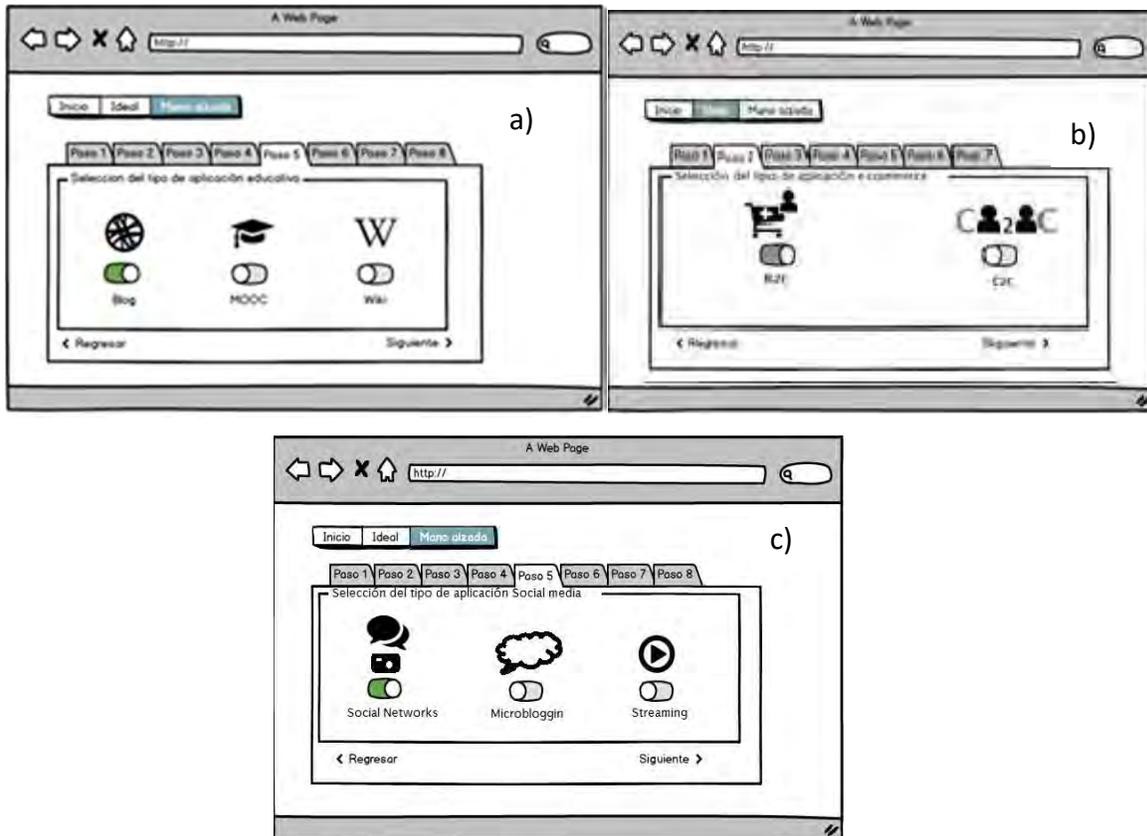


Figura 3.18 *Mockups* del proceso a mano alzada: a) Selección del tipo de aplicación educativa, b) Selección del tipo de aplicación e-commerce y c) Selección del tipo de aplicación social media.

Diseño de las páginas para la selección de dispositivos/plataformas, configuración y descarga del código: estos diseños también retoman el diseño de los *mockups* presentados en la figura 3.11 para la selección de dispositivos/plataformas, 3.13 para la configuración, 3.14 para confirmar la configuración seleccionada, 3.15 (a) para generar la aplicación y 3.15 (b) para descargar el código de la aplicación.

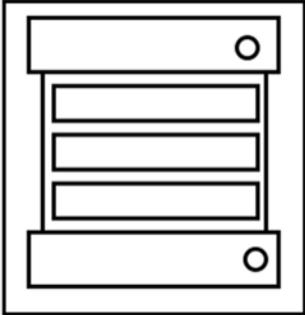
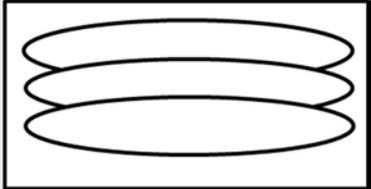
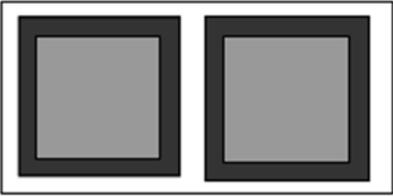
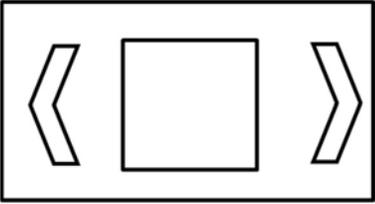
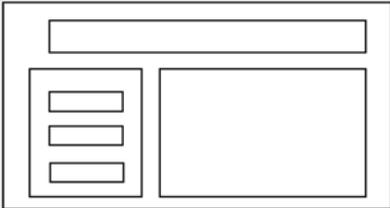
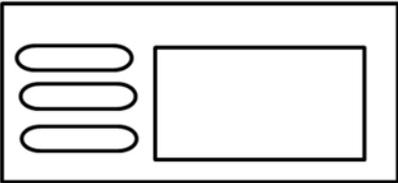
3.2.5 Conjunto de iconos de patrones de diseño de interfaz de usuario

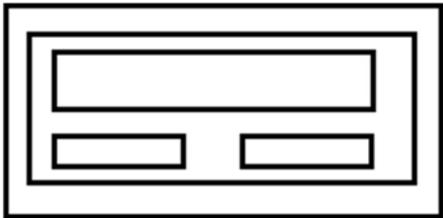
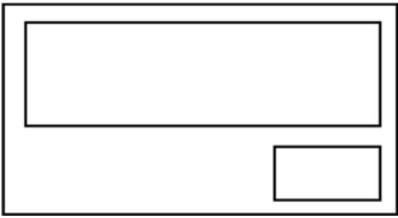
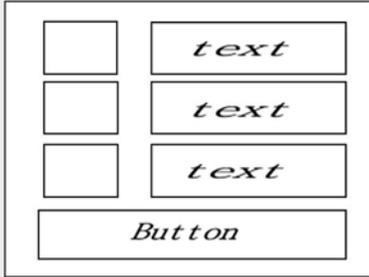
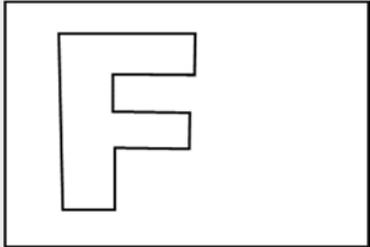
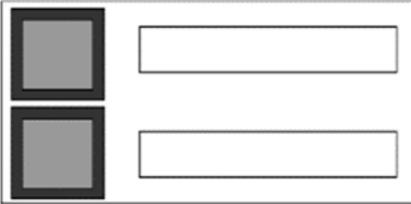
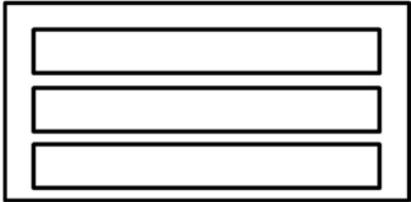
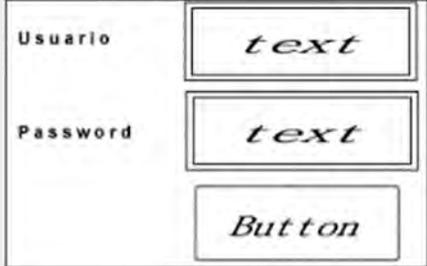
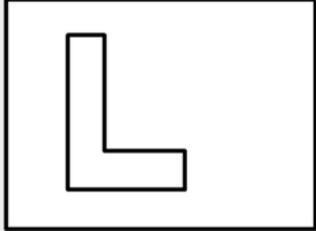
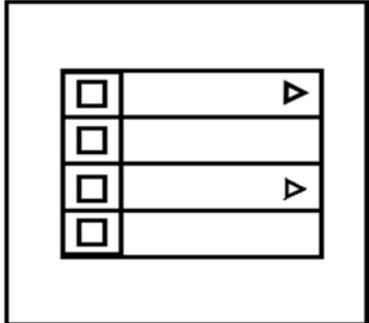
Para el proceso de generación de aplicaciones se utilizó un conjunto de representaciones graficas de patrones de diseño individuales y composiciones, estas representaciones graficas permiten la generación de imágenes a mano alzada

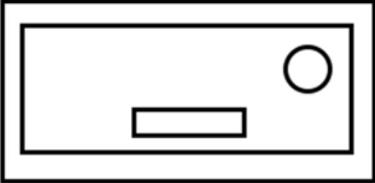
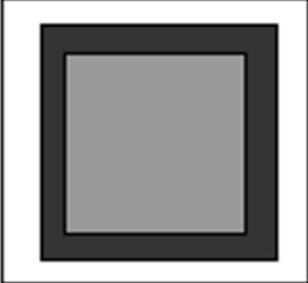
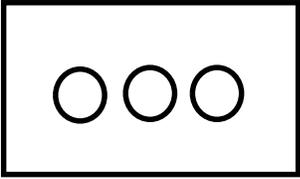
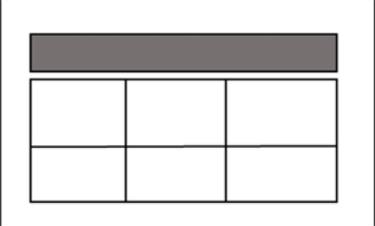
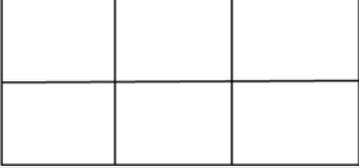
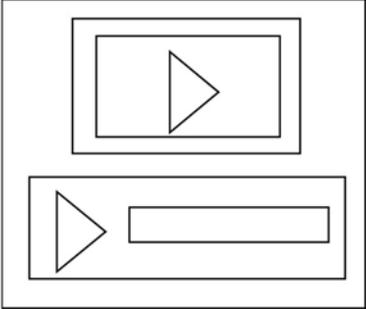
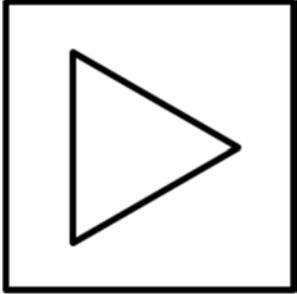
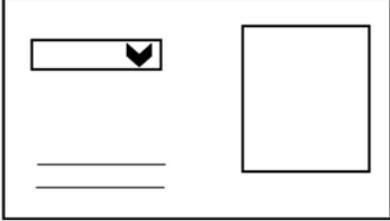
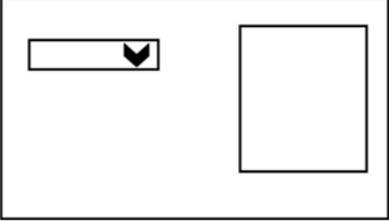
que representan interfaces de usuario válidas para el sistema y así agilizar el proceso de generación de aplicaciones.

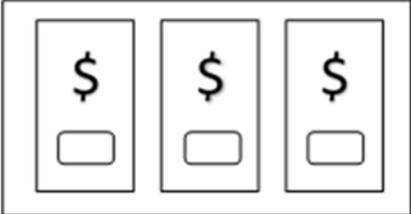
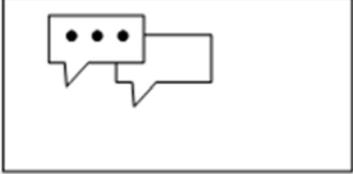
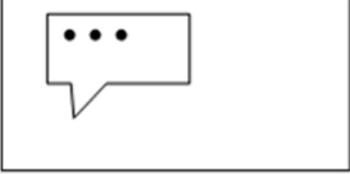
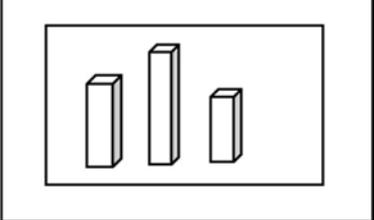
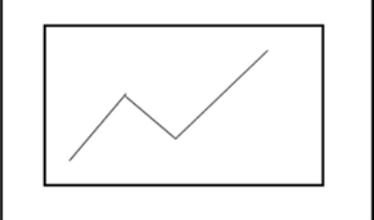
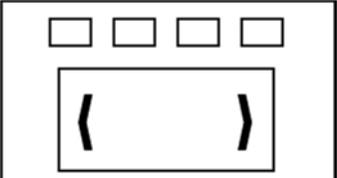
Los patrones de diseño individuales utilizan dos representaciones, el primer icono se utiliza para identificar patrones individuales y el segundo icono es para la relación con los patrones compuestos. En la Tabla 3.9 se muestran los patrones de diseño individuales con su iconografía correspondiente.

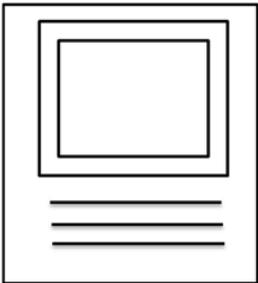
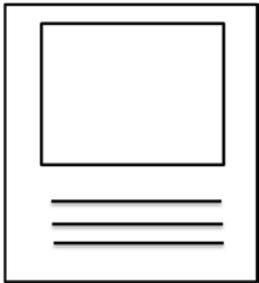
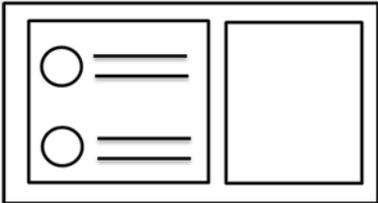
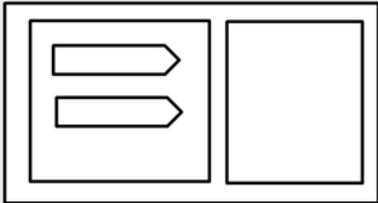
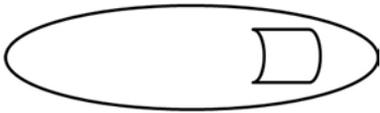
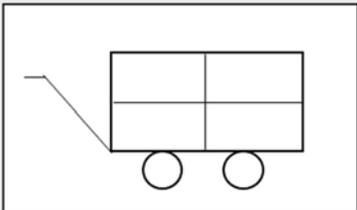
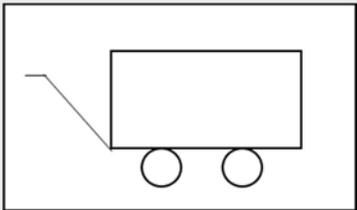
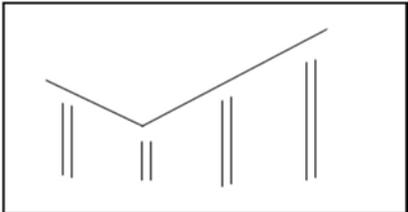
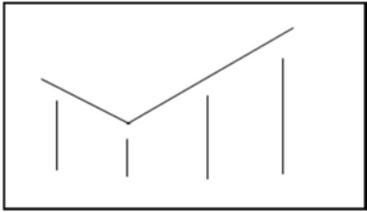
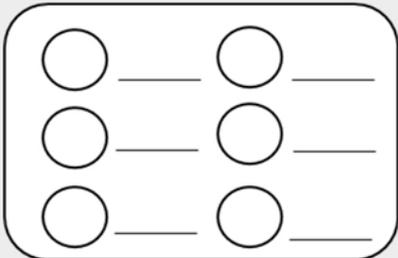
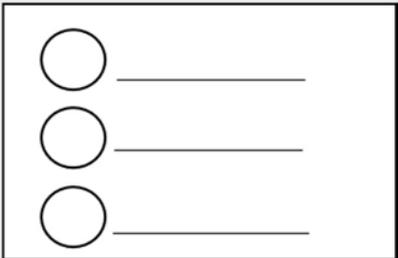
Tabla 3.9 Iconografía de patrones de diseño de interfaz de usuario en su representación individual.

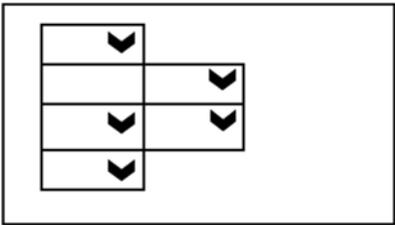
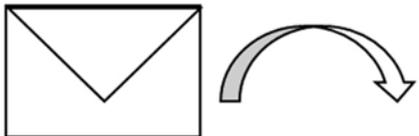
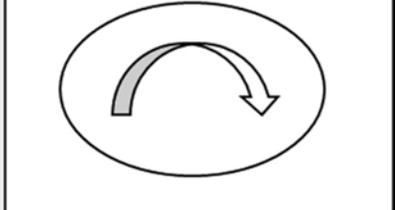
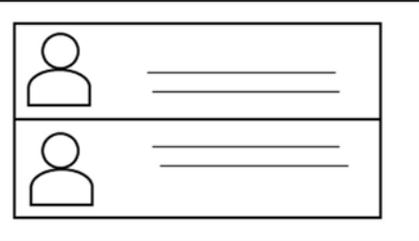
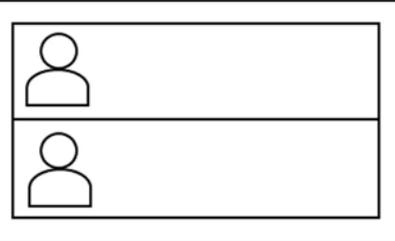
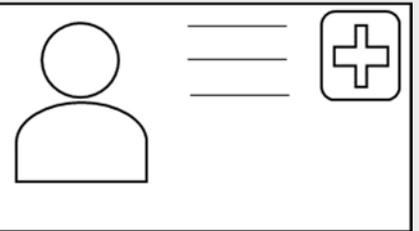
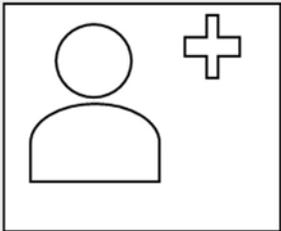
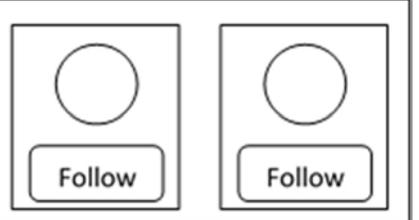
Patrones individuales de diseño de IU	Icono para patrones individuales	Icono para patrones compuestos
<i>Accordion</i>		
<i>Wizard</i>		
<i>Progress Bar</i>		
<i>Carousel</i>		
<i>Dashboard</i>		

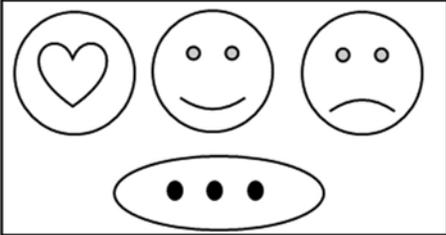
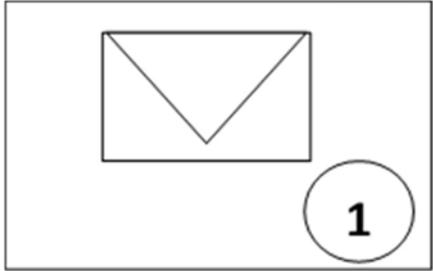
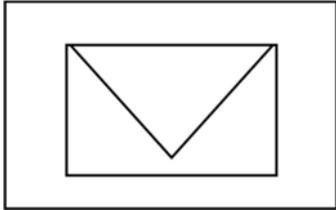
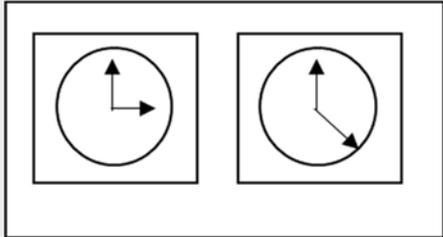
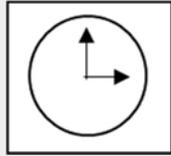
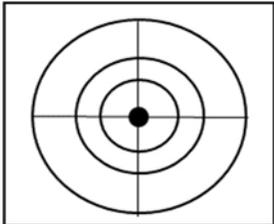
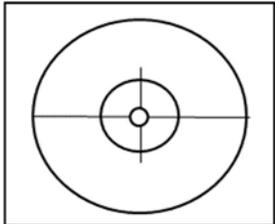
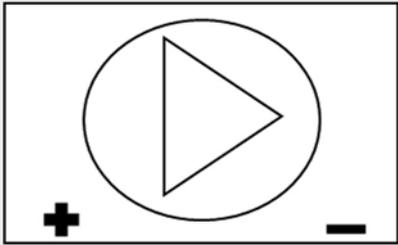
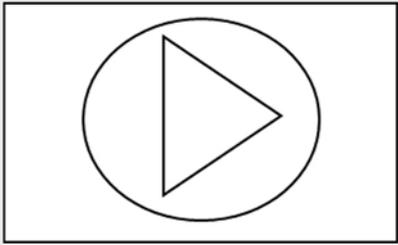
Patrones individuales de diseño de IU	Icono para patrones individuales	Icono para patrones compuestos
Dialog		
Form		
List		
Login		
Contextual Menu		

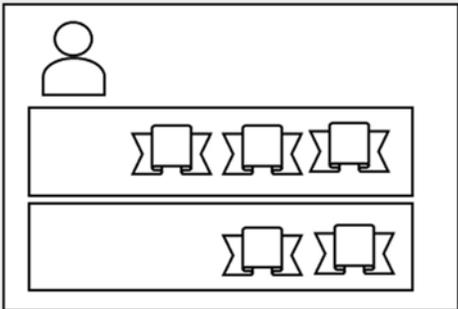
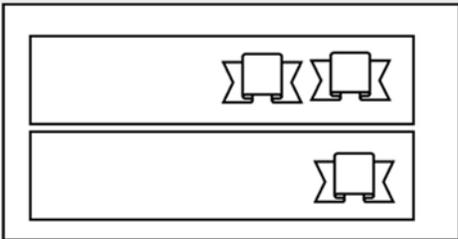
Patrones individuales de diseño de IU	Icono para patrones individuales	Icono para patrones compuestos
Notification		
Splashscreen		
Table		
Video		
Product page		

Patrones individuales de diseño de IU	Icono para patrones individuales	Icono para patrones compuestos
Pricing table		
Coupon		
Comments		
Chat		
Stats		
Gallery		

Patrones individuales de diseño de IU	Icono para patrones individuales	Icono para patrones compuestos
Targets		
Master-Detail		
Search		
Shopping cart		
Ranking		
Panel		

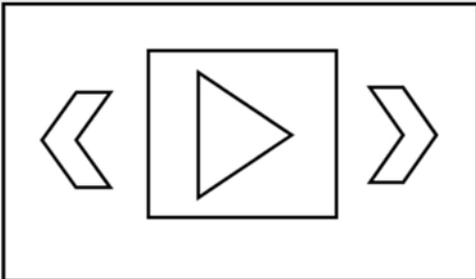
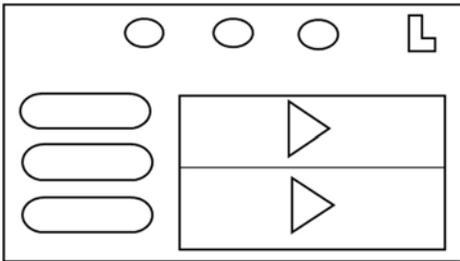
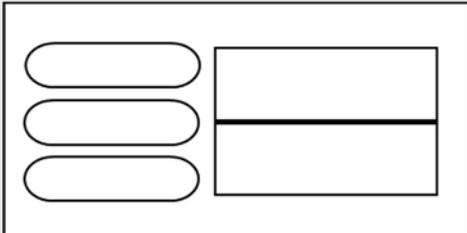
Patrones individuales de diseño de IU	Icono para patrones individuales	Icono para patrones compuestos
Menu		
Tip a friend		
Friend list		
Friend		
Follow		

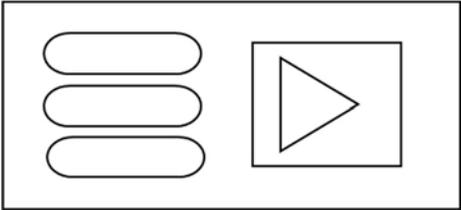
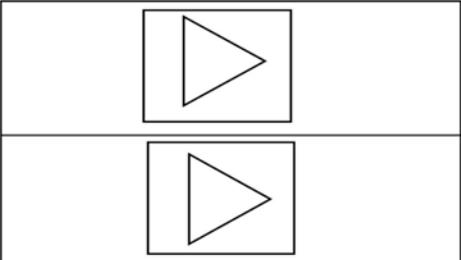
Patrones individuales de diseño de IU	Icono para patrones individuales	Icono para patrones compuestos
Reaction		
Invited Friends		
Activity stream		
Picker		
Music player		
Archievements		

Patrones individuales de diseño de IU	Icono para patrones individuales	Icono para patrones compuestos
Loaderboards		

La iconografía de los patrones de diseño de interfaz de usuario compuestos predefinidos para el dominio educativo se presenta en la Tabla 3.10.

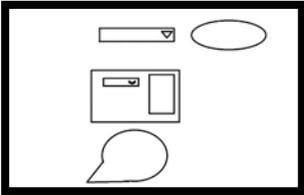
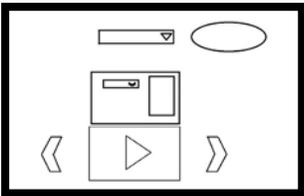
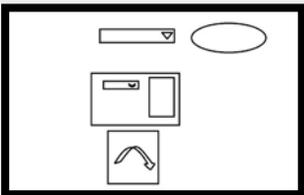
Tabla 3.10 Iconografía de interfaces de usuario compuestas para el dominio educativo.

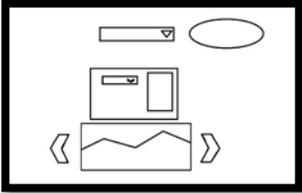
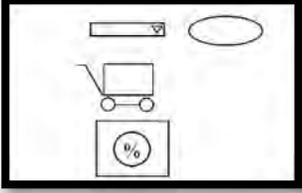
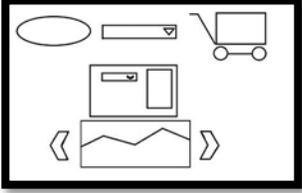
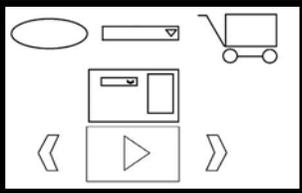
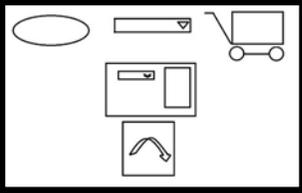
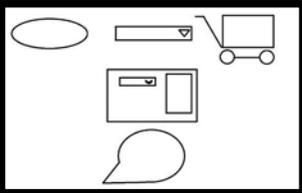
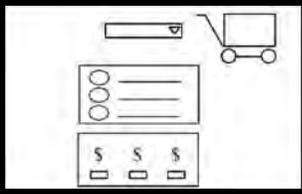
Composición	Icono
Carrusel de vídeos	
Navegación básica	
Dashboard con lista	

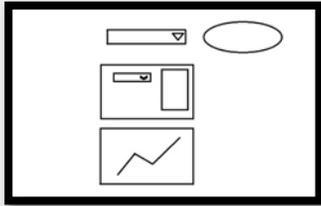
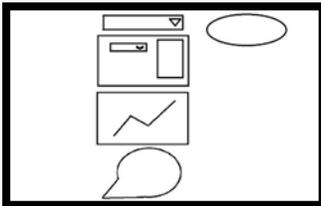
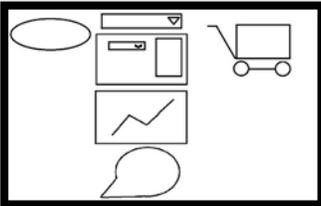
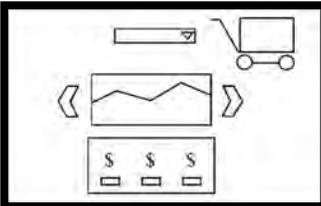
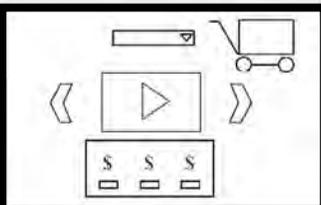
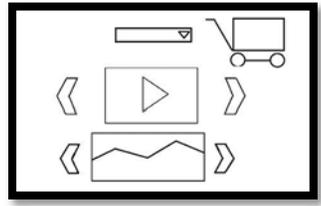
Composición	Icono
Dashboard con video	
Lista de videos	

Respecto a las composiciones definidas para el dominio e-commerce se muestran en la tabla 3.11.

Tabla 3.11 Iconografía de interfaces de usuario compuestas para el dominio e-commerce.

No.	UIDPs de la composición	Icono
1	<i>Search</i> <i>Menu</i> <i>Product page</i> <i>Comment</i>	
2	<i>Search</i> <i>Menu</i> <i>Product page</i> <i>Player</i>	
3	<i>Search</i> <i>Menu</i> <i>Product page</i> <i>Tip a friend</i>	

No.	UIDPs de la composición	Icono
4	<p><i>Search</i></p> <p><i>Menu</i></p> <p><i>Product page</i></p> <p><i>Gallery</i></p>	
5	<p><i>Search</i></p> <p><i>Menu</i></p> <p><i>Shopping Cart</i></p> <p><i>Coupon</i></p>	
6	<p><i>Search</i></p> <p><i>Menu</i></p> <p><i>Shopping Cart</i></p> <p><i>Product page</i></p> <p><i>Gallery</i></p>	
7	<p><i>Search</i></p> <p><i>Menu</i></p> <p><i>Shopping Cart</i></p> <p><i>Product page</i></p> <p><i>Player</i></p>	
8	<p><i>Search</i></p> <p><i>Menu</i></p> <p><i>Shopping Cart</i></p> <p><i>Product page</i></p> <p><i>Tip a friend</i></p>	
9	<p><i>Search</i></p> <p><i>Menu</i></p> <p><i>Shopping Cart</i></p> <p><i>Product page</i></p> <p><i>Comment</i></p>	
10	<p><i>Search</i></p> <p><i>Menu</i></p> <p><i>Panel</i></p> <p><i>Pricing Table</i></p>	

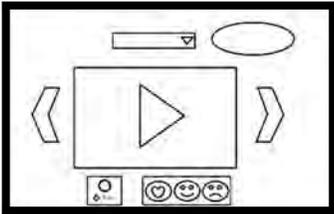
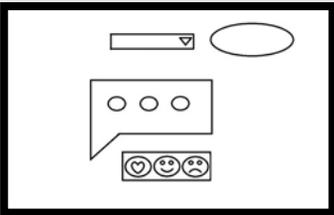
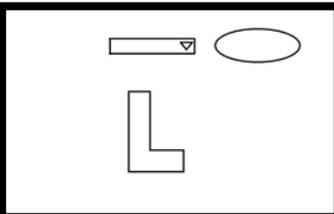
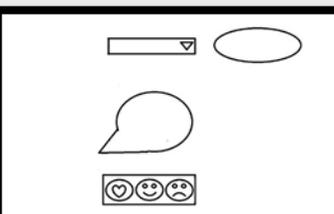
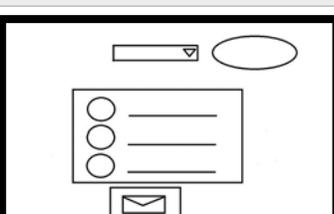
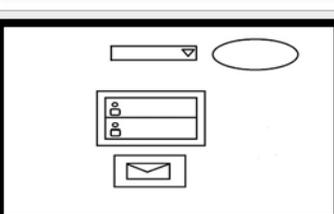
No.	UIDPs de la composición	Icono
11	<p><i>Search</i></p> <p><i>Menu</i></p> <p><i>Product page</i></p> <p><i>Stats</i></p>	
12	<p><i>Search</i></p> <p><i>Menu</i></p> <p><i>Product page</i></p> <p><i>Stats</i></p> <p><i>Comment</i></p>	
13	<p><i>Search</i></p> <p><i>Menu</i></p> <p><i>Shopping Cart</i></p> <p><i>Product page</i></p> <p><i>Stats</i></p> <p><i>Comment</i></p>	
14	<p><i>Menu</i></p> <p><i>Shopping Cart</i></p> <p><i>Gallery</i></p> <p><i>Pricing Table</i></p>	
15	<p><i>Menu</i></p> <p><i>Shopping Cart</i></p> <p><i>Video</i></p> <p><i>Pricing Table</i></p>	
16	<p><i>Menu</i></p> <p><i>Shopping Cart</i></p> <p><i>Video</i></p> <p><i>Gallery</i></p>	

No.	UIDPs de la composición	Icono
17	<p><i>Menu</i></p> <p><i>Shopping Cart</i></p> <p><i>Pricing Table</i></p> <p><i>Stats</i></p>	
18	<p><i>Search</i></p> <p><i>Menu</i></p> <p><i>Pricing Table</i></p> <p><i>Stats</i></p>	
19	<p><i>Search</i></p> <p><i>Menu</i></p> <p><i>Gallery</i></p> <p><i>Video</i></p>	
20	<p><i>Search</i></p> <p><i>Menu</i></p> <p><i>Video</i></p> <p><i>Pricing Table</i></p>	

Para el dominio social media, fueron propuestas 10 composiciones que se observan en la Tabla 3.12.

Tabla 3.12 Iconografía de interfaces de usuario compuestas para el dominio social media.

No.	UIDPs de la composición	Icono
1	<p><i>Search</i></p> <p><i>Menu</i></p> <p><i>Gallery</i></p> <p><i>Follow</i></p> <p><i>Reaction</i></p>	

No.	UIDPs de la composición	Icono
2	<p><i>Search</i></p> <p><i>Menu</i></p> <p><i>Player</i></p> <p><i>Follow</i></p> <p><i>Reaction</i></p>	
3	<p><i>Search</i></p> <p><i>Menu</i></p> <p><i>Chat</i></p> <p><i>Reaction</i></p>	
4	<p><i>Search</i></p> <p><i>Menu</i></p> <p><i>Logging</i></p>	
5	<p><i>Search</i></p> <p><i>Menu</i></p> <p><i>Comment</i></p> <p><i>Reaction</i></p>	
6	<p><i>Search</i></p> <p><i>Menu</i></p> <p><i>Panel</i></p> <p><i>Friend Invite</i></p>	
7	<p><i>Search</i></p> <p><i>Menu</i></p> <p><i>Friend Invite</i></p> <p><i>Friend List</i></p>	

No.	UIDPs de la composición	Icono
8	<p><i>Search</i></p> <p><i>Menu</i></p> <p><i>Friend Invite</i></p> <p><i>Friend</i></p>	
9	<p><i>Search</i></p> <p><i>Menu</i></p> <p><i>Activity Stream</i></p>	
10	<p><i>Search</i></p> <p><i>Menu</i></p> <p><i>Ranking</i></p>	

3.2.6 Mecanismo para el diseño de la estructura del contenido para aplicaciones móviles

Es importante identificar las posiciones validas que cada patrón de diseño tiene dentro de la interfaz de usuario para garantizar la correcta generación de aplicaciones móviles, pero sobre todo la correcta visualización y funcionamiento de cada patrón en dispositivos móviles.

Cabe mencionar, que hasta el momento las plantillas diseñadas consideran el conjunto de patrones de diseño identificados para la generación de aplicaciones móviles educativas.

3.2.6.1 Plantillas para aplicaciones multi-dominio

De acuerdo con la Tabla 3.5, cada tipo de aplicación tiene un conjunto de patrones de IU válidos y estos patrones tienen una posición válida dentro de la interfaz de usuario. Partiendo de lo anterior en la figura 3.19 se presentan un conjunto de plantillas que permiten distribuir los patrones en la interfaz de usuario, estas

plantillas son apropiadas para todos los tipos de aplicaciones soportados de los tres dominios.

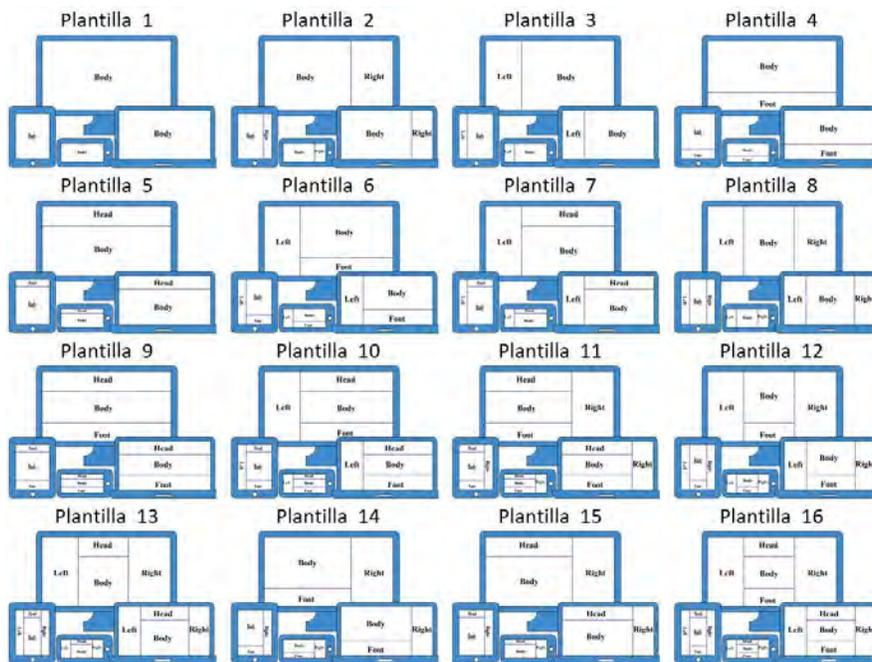


Figura 3.19 Plantillas para generar aplicaciones multi-dominio.

Las plantillas 7, 13 y 15 de la figura 3.19 son las que más se relacionan con cada tipo de aplicación educativa abordada en este dominio. La figura 3.20 (a) representa la plantilla para un blog, la figura 3.20 (b) representa la plantilla de un MOOC y, finalmente, la figura 3.20 (c) representa la plantilla para una wiki.

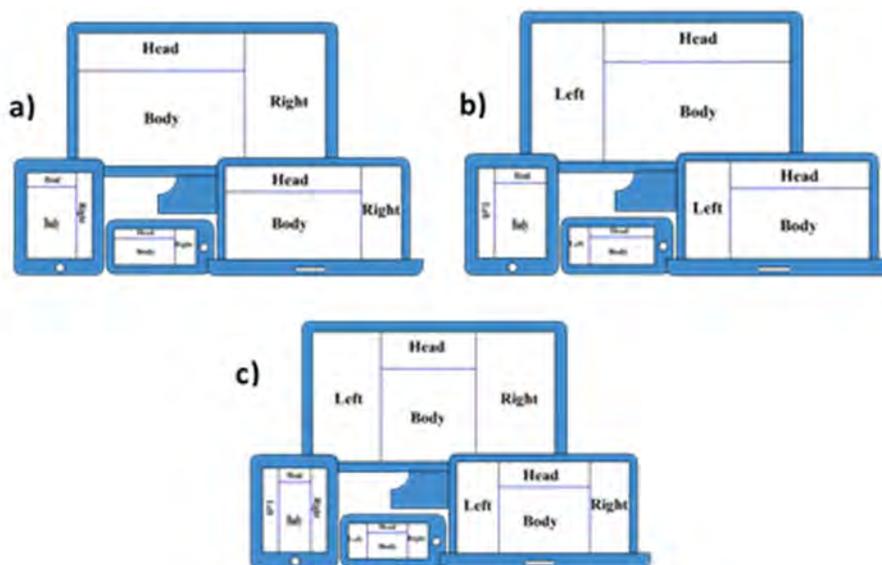


Figura 3.20 Plantillas para aplicaciones educativas: a) Blog, b) MOOC y c) Wiki.

Las plantillas propuestas son la base para la generación de aplicaciones multi-dominio, el usuario elige desde la interfaz de MultiGene una de acuerdo al tipo de aplicación a generar.

3.2.7 Representación en RuleML del proceso de identificación de elementos en imágenes generadas a mano alzada

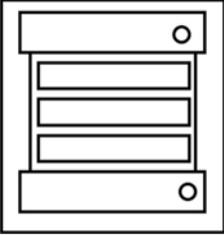
Durante el proceso de identificación de elementos en imágenes generadas a mano alzada o bien en imágenes generadas a través de una herramienta de dibujo, es necesario establecer un conjunto de reglas que permitan establecer los mecanismos que faciliten la identificación de los patrones de diseño en imágenes. En este caso estas reglas se aplicaron en un módulo de procesamiento e identificación de elementos usando Matlab y se evaluó a través de elementos de aplicaciones educativas.

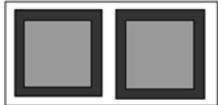
Dicho lo anterior, el objetivo de definir el conjunto de reglas parte del hecho de tener un grupo de especificaciones para cada elemento que sirvan como identificadores únicos al momento de realizar el procesamiento de la imagen, dichas reglas se representan en un árbol el cual emplea el lenguaje RuleML en su versión 1.0 (Boley et al., 2010). RuleML es una iniciativa abierta en la que se busca establecer un sistema de reglas de inferencia lógica a partir de ontologías y documentos RDF con su propio lenguaje de especificación y ejecución.

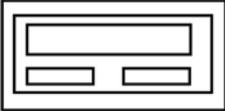
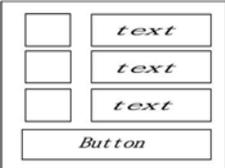
Debido a que los iconos diseñados para representar a cada patrón de diseño y su composición están representados en su mayoría por figuras geométricas, las reglas establecidas son basadas en la combinación de las figuras de cada patrón de diseño.

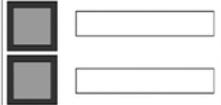
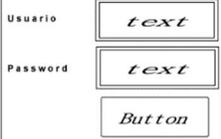
En total se realizó un conjunto de 28 reglas para identificar 14 patrones de diseño de interfaz de usuario individuales y su representación para composición que se presentan en la Tabla 3.9. Las reglas para las representaciones de patrones simples e individuales se presentan en la Tabla 3.13.

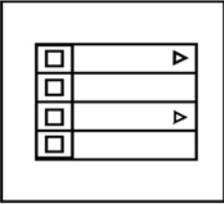
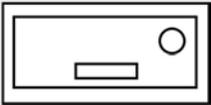
Tabla 3.13 Reglas RuleML para representaciones de patrones simples e individuales.

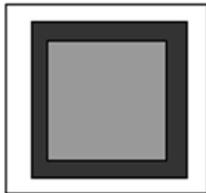
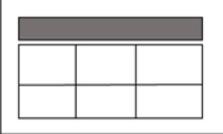
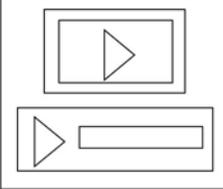
Patrones de diseño de IU	Icono simple	RuleML
<p>Accordion</p>		<pre> <Implies> <If> <And> <Atom> <Var>Forma externa</var> <Rel>Rectángulo</Rel> </Atom> <Atom> <Var>Figuras internas</var> <Rel>Rectángulos</Rel> <Ind>Cinco</Ind> <Ind>Dos grandes</Ind> <Ind>Tres chicos</Ind> <Rel>Cuadrado</Rel> <Rel>Círculo</Rel> <Ind>Dos</Ind> </Atom> </And> </If> <Then> <Atom> <Var>Forma externa</Var> <Var>Figuras internas</Var> <Rel>Acordeón</Rel> </Atom> </Then> </Implies> </pre>
<p>Wizard</p>		<pre> <Implies> <If> <And> <Atom> <Var>Forma externa</var> <Rel>Rectángulo </Rel> </Atom> <Atom> <Var>Figuras internas</var> <Rel>Rectángulos</Rel> <Ind>Tres</Ind> <Ind>Dos chicos</Ind> <Ind>Uno grande</Ind> <Rel>Triángulos</Rel> <Ind>Dos</Ind> </Atom> </And> </If> <Then> <Atom> <Var>Forma externa</Var> <Var>Figuras internas</Var> <Rel>Asistente de configuración</Rel> </Atom> </Then> </pre>

Patrones de diseño de IU	Icono simple	RuleML
Progress bar		<pre> </Implies> <Implies> <If> <And> <Atom> <Var>Forma externa</var> <Rel>Rectángulo</Rel> </Atom> <Atom> <Var>Figuras internas</var> <Rel>Rectángulos</Rel> <Ind>Cuatro</Ind> <Ind>Uno grande</Ind> <Ind>Tres chicos verticales</Ind> </Atom> </And> </If> <Then> <Atom> <Var>Forma externa</Var> <Var>Figuras internas</var> <Rel>Barra de progreso</Rel> </Atom> </Then> </Implies> </pre>
Carousel		<pre> <Implies> <If> <And> <Atom> <Var>Forma externa</var> <Rel>Rectángulo</Rel> </Atom> <Atom> <Var>Figuras internas</var> <Rel>Cuadrado</Rel> <Ind>Cuatro</Ind> </Atom> </And> </If> <Then> <Atom> <Var>Forma externa</Var> <Var>Figuras internas</var> <Rel>Carrusel</Rel> </Atom> </Then> </Implies> </pre>
Dashboard		<pre> <Implies> <If> <And> <Atom> <Var>Forma externa</var> <Rel>Rectángulo</Rel> </Atom> </And> </If> <Atom> </pre>

Patrones de diseño de IU	Icono simple	RuleML
		<pre> <Var>Figuras internas</var> <Rel>Rectángulos</Rel> <Ind>Seis</Ind> <Ind>Tres chicos</Ind> <Ind>Dos grandes horizontales</Ind> <Ind>Uno grande vertical</Ind> </Atom> </And> </If> <Then> <Atom> <Var>Forma externa</Var> <Var>Figuras internas</Var> <Rel>Dashboard</Rel> </Atom> </Then> </Implies> </pre>
Dialog		<pre> <Implies> <If> <And> <Atom> <Var>Forma externa</var> <Rel>Rectángulo</Rel> </Atom> <Atom> <Var>Figuras internas</var> <Rel>Rectángulos</Rel> <Ind>Cuatro</Ind> <Ind>Uno grande</Ind> <Ind>Uno mediano</Ind> <Ind>Dos chicos</Ind> </Atom> </And> </If> <Then> <Atom> <Var>Forma externa</Var> <Var>Figuras internas</Var> <Rel>Diálogo</Rel> </Atom> </Then> </Implies> </pre>
Form		<pre> <Implies> <If> <And> <Atom> <Var>Forma externa</var> <Rel>Rectángulo</Rel> </Atom> <Atom> <Var>Figuras internas</var> <Rel>Cuadrados</Rel> <Ind>Tres</Ind> <Rel>Rectángulos</Rel> </pre>

Patrones de diseño de IU	Icono simple	RuleML
		<pre> <Ind>Cuatro</Ind> <Rel>Texto</Rel> <Ind>Cuatro</Ind> </Atom> </And> </If> <Then> <Atom> <Var>Forma externa</Var> <Var>Figuras internas</Var> <Rel>Formulario</Rel> </Atom> </Then> </Implies> </pre>
List		<pre> <Implies> <If> <And> <Atom> <Var>Forma externa</var> <Rel>Rectángulo</Rel> </Atom> <Atom> <Var>Figuras internas</var> <Rel>Cuadrados</Rel> <Ind>Cuatro</Ind> <Rel>Rectángulos</Rel> <Ind>Dos</Ind> </Atom> </And> </If> <Then> <Atom> <Var>Forma externa</Var> <Var>Figuras internas</Var> <Rel>Lista</Rel> </Atom> </Then> </Implies> </pre>
Login		<pre> <Implies> <If> <And> <Atom> <Var>Forma externa</var> <Rel>Rectángulo</Rel> </Atom> <Atom> <Var>Figuras internas</var> <Rel>Rectángulos</Rel> <Ind>Cinco</Ind> <Rel>Textos</Rel> <Ind>Cinco</Ind> </Atom> </And> </If> <Then> </pre>

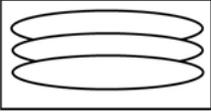
Patrones de diseño de IU	Icono simple	RuleML
		<pre> <Atom> <Var>Forma externa</Var> <Var>Figuras internas</Var> <Rel>Login</Rel> </Atom> </Then> </Implies> </pre>
Contextual menu		<pre> <Implies> <If> <And> <Atom> <Var>Forma externa</var> <Rel>Rectángulo</Rel> </Atom> <Atom> <Var>Figuras internas</var> <Rel>Cuadrados</Rel> <Ind>Ocho</Ind> <Rel>Rectángulos</Rel> <Ind>Cuatro</Ind> <Rel>Triángulos</Rel> <Ind>Dos</Ind> </Atom> </And> </If> <Then> <Atom> <Var>Forma externa</Var> <Var>Figuras internas</Var> <Rel>Menú contextual</Rel> </Atom> </Then> </Implies> </pre>
Notification		<pre> <Implies> <If> <And> <Atom> <Var>Forma externa</var> <Rel>Globo de mensaje</Rel> </Atom> </And> </If> <Then> <Atom> <Var>Forma externa</Var> <Rel>Rectángulos</Rel> <Ind>Uno grande</Ind> <Ind>Uno chico</Ind> <Rel>Círculo</Rel> </Atom> </Then> </Implies> </pre>

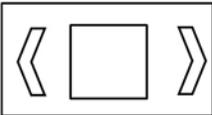
Patrones de diseño de IU	Icono simple	RuleML
Splashscreen		<pre> <Implies> <If> <And> <Atom> <Var>Forma externa</var> <Rel>Rectángulo</Rel> </Atom> <Atom> <Var>Figuras internas</var> <Rel>Cuadrados</Rel> <Ind>Dos</Ind> </Atom> </And> </If> <Then> <Atom> <Var>Forma externa</Var> <Var>Figuras internas</Var> <Rel>Splashscreen</Rel> </Atom> </Then> </Implies> </pre>
Table		<pre> <Implies> <If> <And> <Atom> <Var>Forma externa</var> <Rel>Rectángulo</Rel> </Atom> <Atom> <Var>Figuras internas</var> <Rel>Rectángulo</Rel> <Rel>Cuadrado</Rel> <Ind>Seis</Ind> </Atom> </And> </If> <Then> <Atom> <Var>Forma externa</Var> <Var>Figuras internas</var> <Rel>Tabla</Rel> </Atom> </Then> </Implies> </pre>
Video		<pre> <Implies> <If> <And> <Atom> <Var>Forma externa</var> <Rel>Cuadrado</Rel> </Atom> <Atom> <Var>Figuras internas</var> <Rel>Triángulo</Rel> </Atom> </And> </If> <Then> <Atom> <Var>Forma externa</Var> <Var>Figuras internas</var> <Rel>Video</Rel> </Atom> </Then> </Implies> </pre>

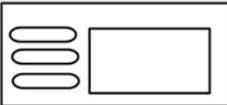
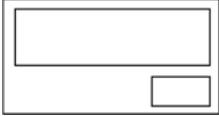
Patrones de diseño de IU	Icono simple	RuleML
		<pre> <Rel>Dos</Rel> <Rel>Rectángulo</Rel> <Rel>Cuatro</Rel> </Atom> </And> </If> <Then> <Atom> <Var>Forma externa</Var> <Var>Figuras internas</Var> <Rel>Video</Rel> </Atom> </Then> </Implies> </pre>

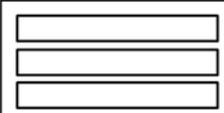
En la tabla 3.14 se presentan las reglas RuleML para las representaciones de patrones individuales en composiciones.

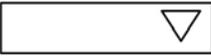
Tabla 3.14 Reglas RuleML para la representación de patrones individuales en composición.

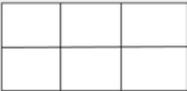
Patrones de diseño de IU	Icono para composición	RuleML
Accordion		<pre> <Implies> <If> <And> <Atom> <Var>Forma externa</var> <Rel>Rectángulo</Rel> </Atom> <Atom> <Var>Figuras internas</var> <Rel>Óvalos</Rel> <Ind>Tres</Ind> <Ind>Óvalos encimados hacia arriba</Ind> </Atom> </And> </If> <Then> <Atom> <Var>Forma externa</Var> <Var>Figuras internas</Var> <Rel>Acordeón</Rel> </Atom> </Then> </Implies> </pre>
Wizard		<pre> <Implies> <If> <And> <Atom> </pre>

Patrones de diseño de IU	Icono para composición	RuleML
		<pre> <Var>Forma externa</var> <Rel>Pestaña </Rel> </Atom> </And> </If> <Then> <Atom> <Var>Forma externa</Var> <Rel>Asistente de configuración</Rel> </Atom> </Then> </Implies> </pre>
Progress bar		<pre> <Implies> <If> <And> <Atom> <Var>Forma externa</var> <Rel>Rectángulo</Rel> <Ind>Dos</Ind> <Ind>Uno relleno</Ind> <Ind>Uno vacío</Ind> </Atom> </And> </If> <Then> <Atom> <Var>Forma externa</Var> <Rel>Barra de progreso</Rel> </Atom> </Then> </Implies> </pre>
Carousel		<pre> <Implies> <If> <And> <Atom> <Var>Forma externa</var> <Rel>Rectángulo</Rel> </Atom> <Atom> <Var>Figuras internas</var> <Rel>Cuadrado</Rel> <Ind>Uno</Ind> <Rel>Flecha abierta</Rel> <Ind>Dos</Ind> </Atom> </And> </If> <Then> <Atom> <Var>Forma externa</Var> <Var>Figuras internas</Var> <Rel>Carrusel</Rel> </Atom> </pre>

Patrones de diseño de IU	Icono para composición	RuleML
Dashboard		<pre> </Then> </Implies> <Implies> <If> <And> <Atom> <Var>Forma externa</var> <Rel>Rectángulo</Rel> </Atom> <Atom> <Var>Figuras internas</var> <Rel>Rectángulos</Rel> <Ind>Cuatro</Ind> <Ind>Tres con esquinas redondeadas</Ind> <Ind>Uno en posición horizontal</Ind> </Atom> </And> </If> <Then> <Atom> <Var>Forma externa</Var> <Var>Figuras internas</Var> <Rel>Dashboard</Rel> </Atom> </Then> </Implies> </pre>
Dialog		<pre> <Implies> <If> <And> <Atom> <Var>Forma externa</var> <Rel>Rectángulo</Rel> </Atom> <Atom> <Var>Figuras internas</var> <Rel>Rectángulos</Rel> <Ind>Dos</Ind> <Ind>Uno grande</Ind> <Ind>Uno chico</Ind> </Atom> </And> </If> <Then> <Atom> <Var>Forma externa</Var> <Var>Figuras internas</Var> <Rel>Diálogo</Rel> </Atom> </Then> </Implies> </pre>

Patrones de diseño de IU	Icono para composición	RuleML
Form		<pre> <Implies> <If> <And> <Atom> <Var>Forma externa</var> <Rel>Rectángulo</Rel> </Atom> <Atom> <Var>Figuras internas</var> <Rel>Letra F</Rel> </Atom> </And> </If> <Then> <Atom> <Var>Forma externa</Var> <Var>Figuras internas</Var> <Rel>Formulario</Rel> </Atom> </Then> </Implies> </pre>
List		<pre> <Implies> <If> <And> <Atom> <Var>Forma externa</var> <Rel>Rectángulo</Rel> </Atom> <Atom> <Var>Figuras internas</var> <Rel>Rectángulos</Rel> <Ind>Tres</Ind> </Atom> </And> </If> <Then> <Atom> <Var>Forma externa</Var> <Var>Figuras internas</Var> <Rel>Lista</Rel> </Atom> </Then> </Implies> </pre>
Login		<pre> <Implies> <If> <And> <Atom> <Var>Forma externa</var> <Rel>Rectángulo</Rel> </Atom> <Atom> <Var>Figuras internas</var> <Rel>Letra L</Rel> </Atom> </And> </If> <Then> <Atom> <Var>Forma externa</Var> <Var>Figuras internas</Var> <Rel>Formulario</Rel> </Atom> </Then> </Implies> </pre>

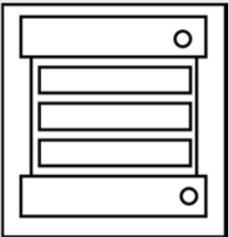
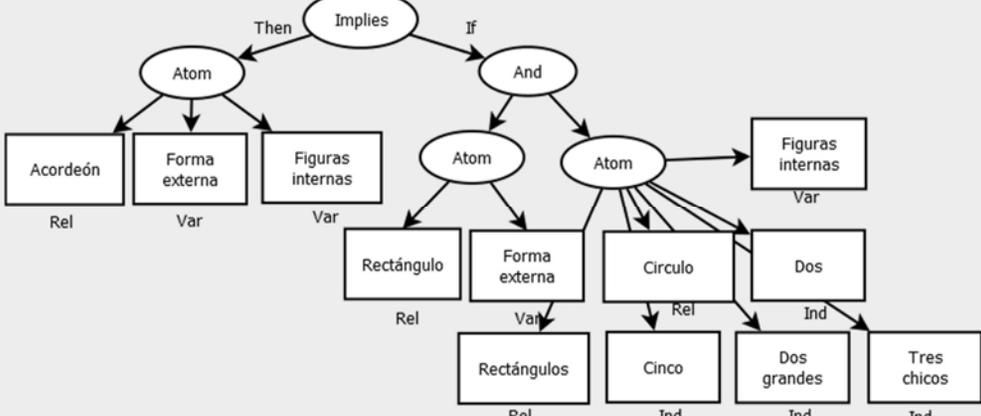
Patrones de diseño de IU	Icono para composición	RuleML
		<pre> </And> </If> <Then> <Atom> <Var>Forma externa</Var> <Var>Figuras internas</Var> <Rel>Login</Rel> </Atom> </Then> </Implies> </pre>
Contextual menu		<pre> <Implies> <If> <And> <Atom> <Var>Forma externa</var> <Rel>Rectángulo</Rel> </Atom> <Atom> <Var>Figuras internas</var> <Rel>Triángulo</Rel> <Ind>Uno hacia abajo</Ind> </Atom> </And> </If> <Then> <Atom> <Var>Forma externa</Var> <Var>Figuras internas</Var> <Rel>Menú contextual</Rel> </Atom> </Then> </Implies> </pre>
Notification		<pre> <Implies> <If> <And> <Atom> <Var>Forma externa</var> <Rel>Globo de mensaje</Rel> </Atom> </And> </If> <Then> <Atom> <Var>Forma externa</Var> <Rel>Notificación</Rel> </Atom> </Then> </Implies> </pre>
Splashscreen		<pre> <Implies> <If> <And> <Atom> <Var>Forma externa</var> <Rel>Rectángulo</Rel> </pre>

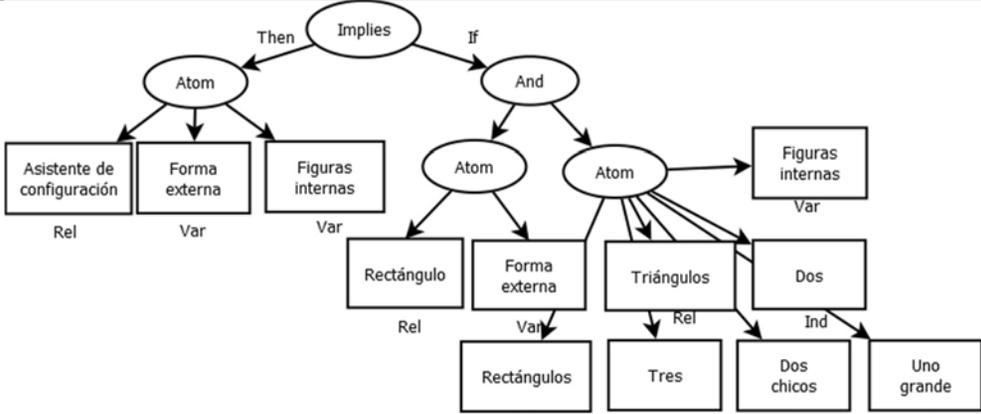
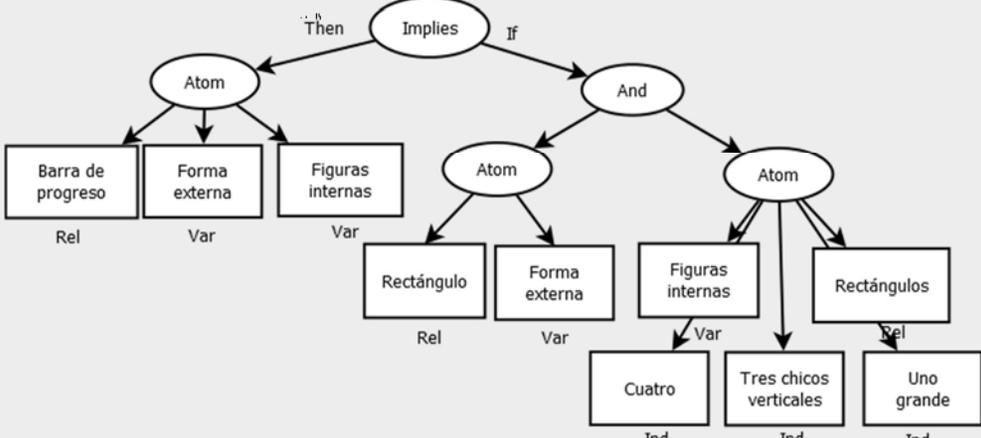
Patrones de diseño de IU	Icono para composición	RuleML
		<pre> </Atom> <Atom> <Var>Figuras internas</var> <Rel>Círculos</Rel> <Ind>Tres</Ind> </Atom> </And> </If> <Then> <Atom> <Var>Forma externa</Var> <Var>Figuras internas</Var> <Rel>Splashscreen</Rel> </Atom> </Then> </Implies> </pre>
Table		<pre> <Implies> <If> <And> <Atom> <Var>Forma externa</var> <Rel>Rectángulos</Rel> <Ind>Seis juntos</Ind> </Atom> </And> </If> <Then> <Atom> <Var>Forma externa</Var> <Rel>Tabla</Rel> </Atom> </Then> </Implies> </pre>
Video		<pre> <Implies> <If> <And> <Atom> <Var>Forma externa</var> <Rel>Cuadrado</Rel> </Atom> <Atom> <Var>Figuras internas</var> <Rel>Triángulo</Rel> </Atom> </And> </If> <Then> <Atom> <Var>Forma externa</Var> <Var>Figuras internas</Var> <Rel>Video</Rel> </Atom> </Then> </Implies> </pre>

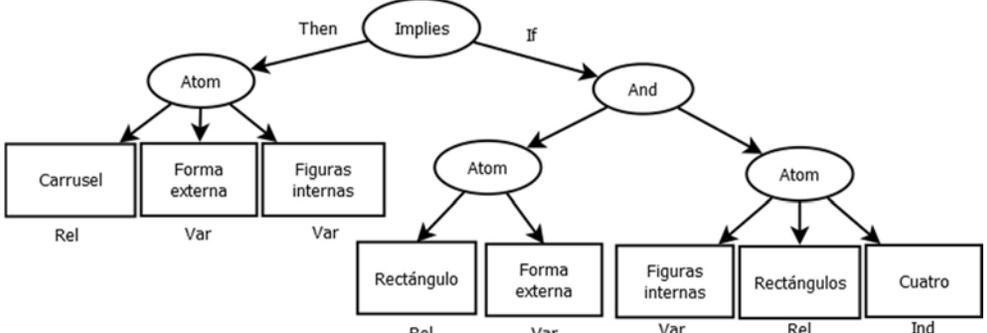
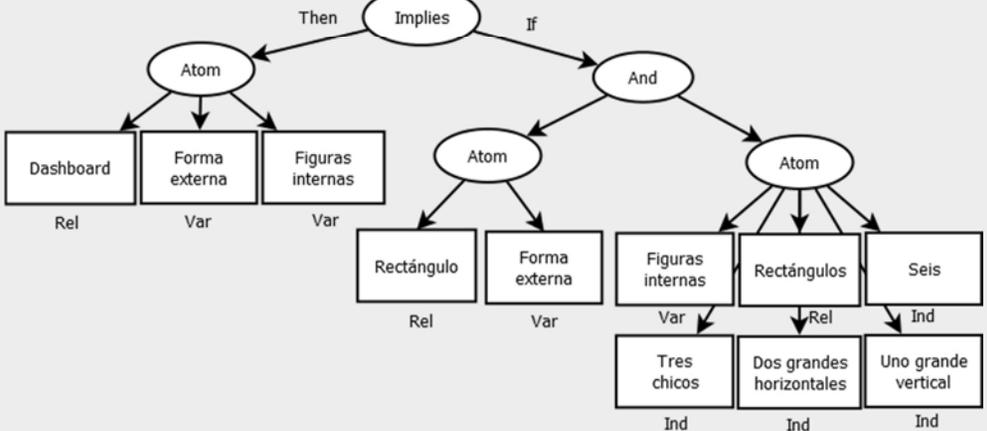
3.2.7.1 Árbol parcial de las representaciones en RuleML

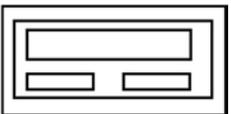
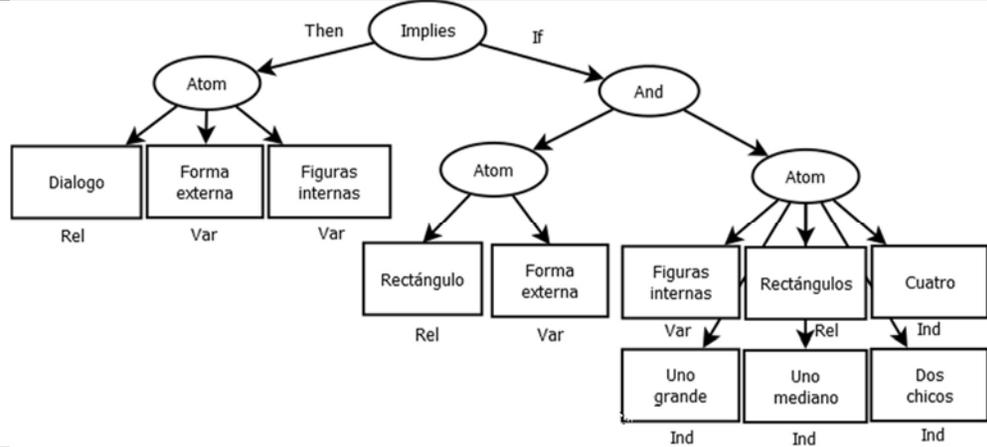
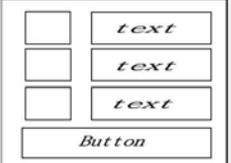
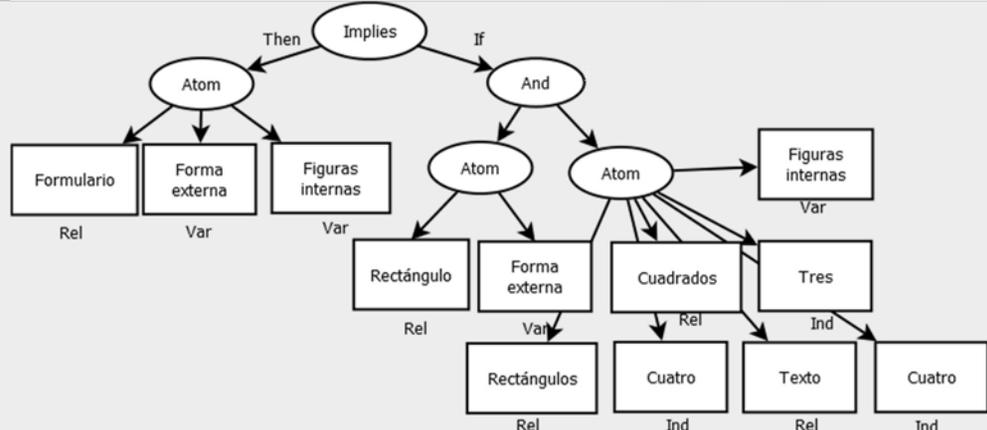
El conjunto de reglas creadas en el lenguaje RuleML se representan en un árbol desde su versión 1.0 (Boley et al., 2010). Con base en las reglas descritas anteriormente en RuleML, en la **¡Error! No se encuentra el origen de la referencia.3.15** se presentan los diagramas de árbol para los patrones individuales simples y en la **¡Error! No se encuentra el origen de la referencia.3.16** se encuentran los diagramas de árbol para los patrones individuales en composición.

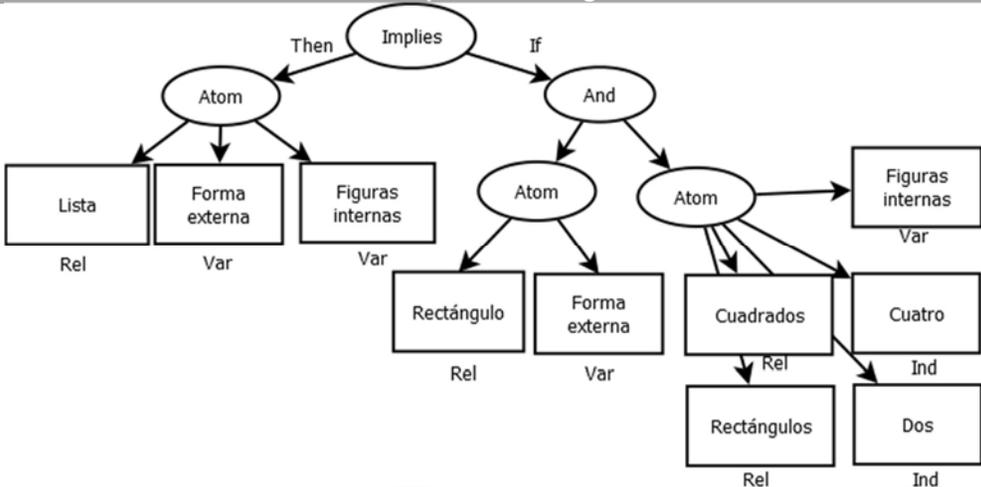
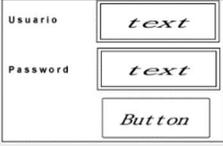
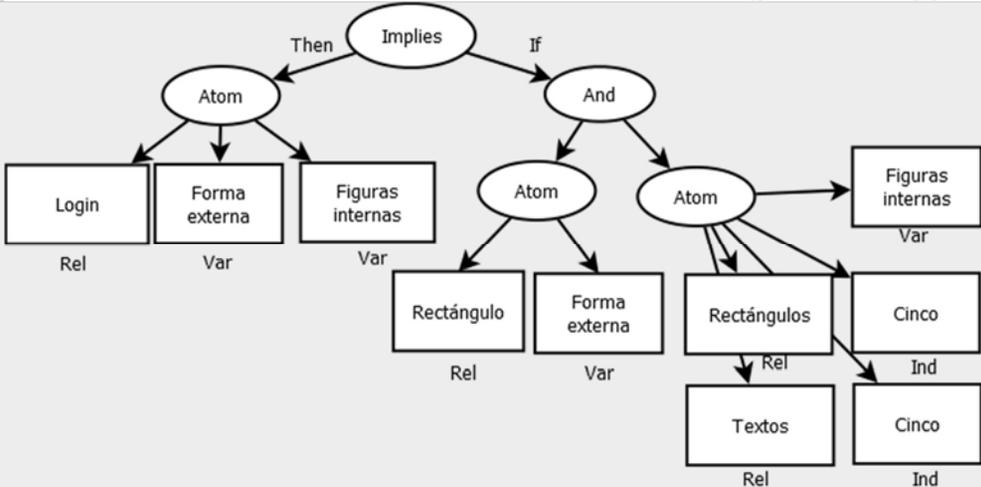
Tabla 3.15 Árboles de reglas para patrones simples.

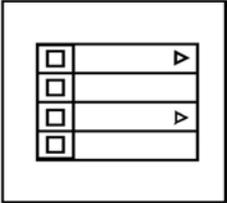
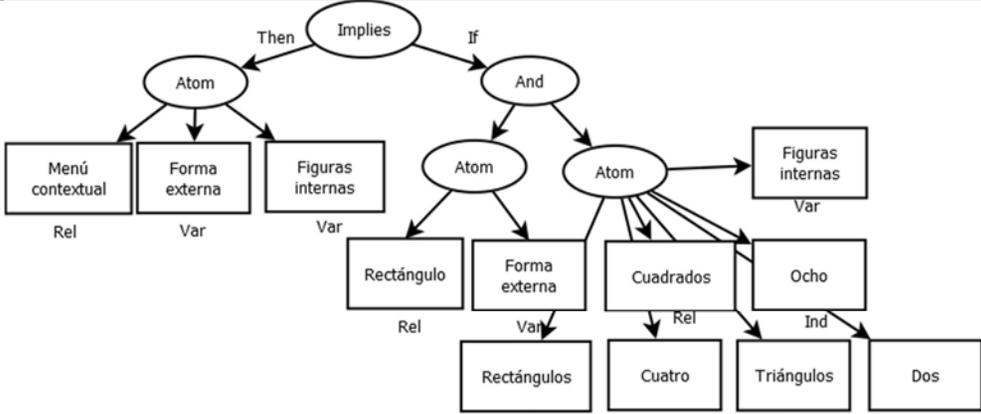
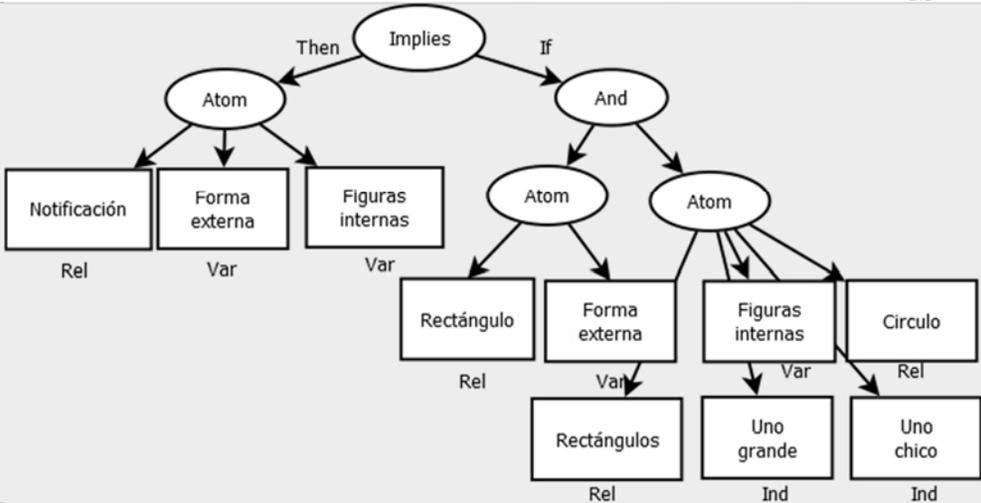
Patrones de diseño de IU	Icono simple	Árbol parcial de reglas
<p>Accordion</p>		

Patrones de diseño de IU	Icono simple	Árbol parcial de reglas
<p>Wizard</p>		
<p>Progress bar</p>		

Patrones de diseño de IU	Icono simple	Árbol parcial de reglas
Carousel		
Dashboard		

Patrones de diseño de IU		Icono simple	Árbol parcial de reglas
Dialog			
Form			

Patrones de diseño de IU		Icono simple	Árbol parcial de reglas
List			
Login			

Patrones de diseño de IU		
	Icono simple	Árbol parcial de reglas
Contextual menu		
Notification		

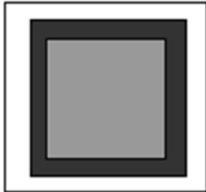
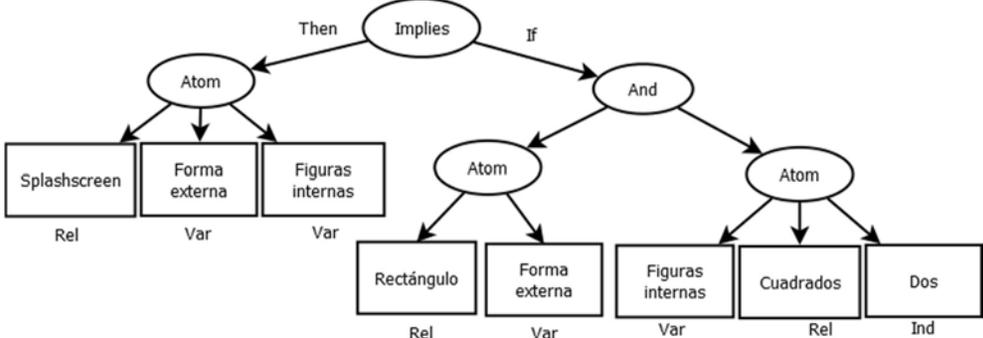
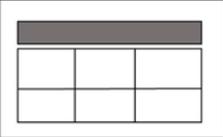
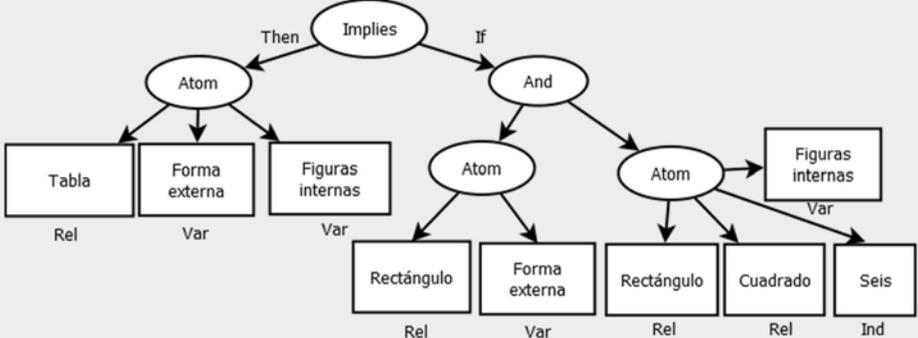
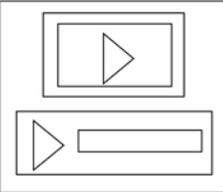
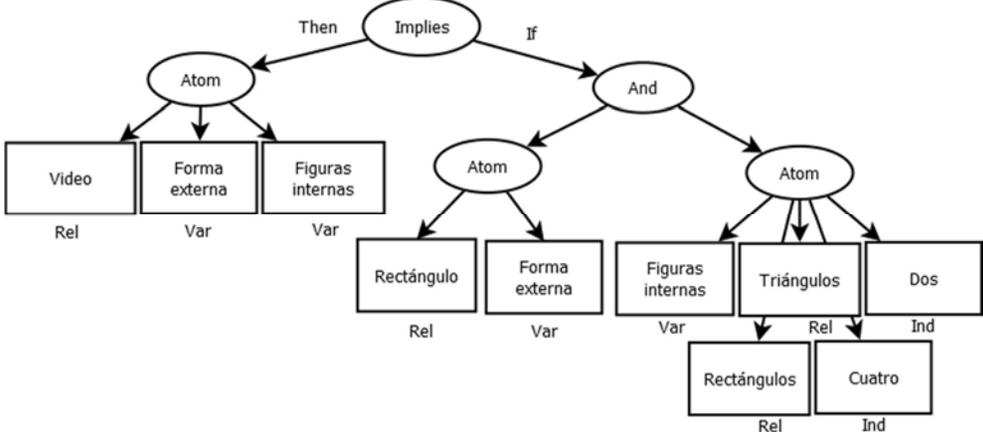
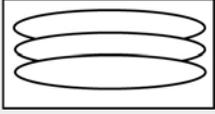
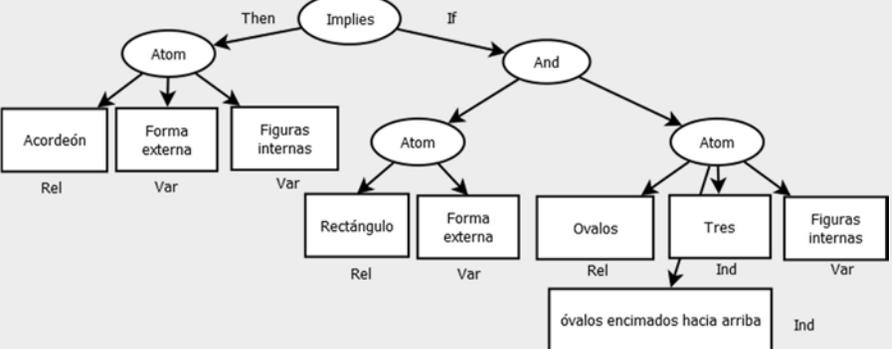
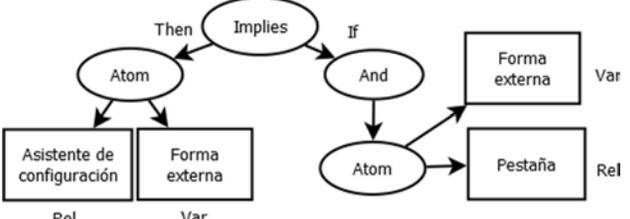
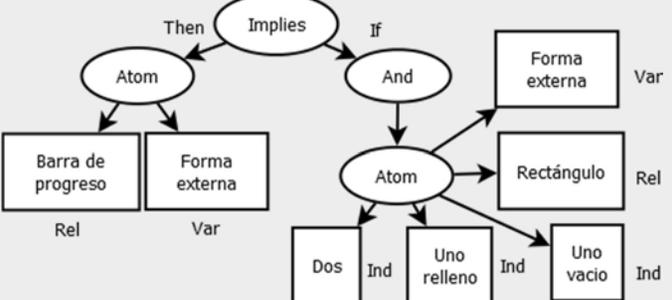
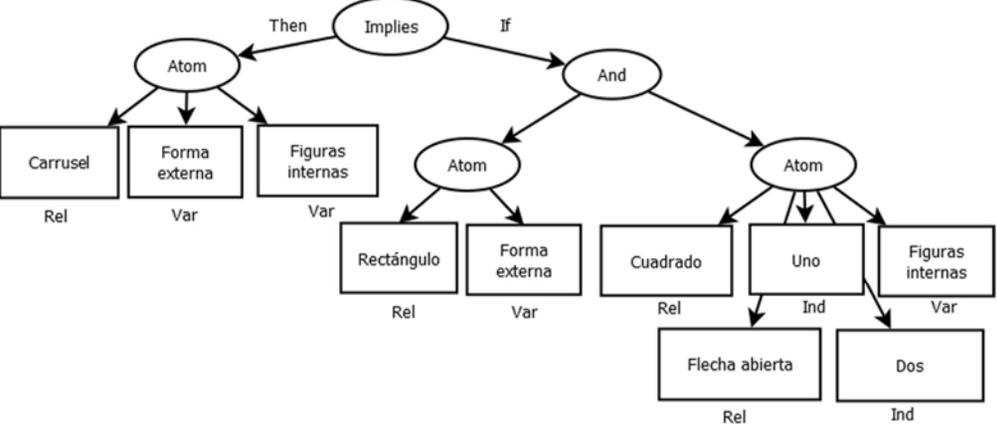
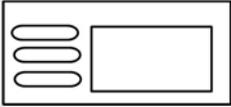
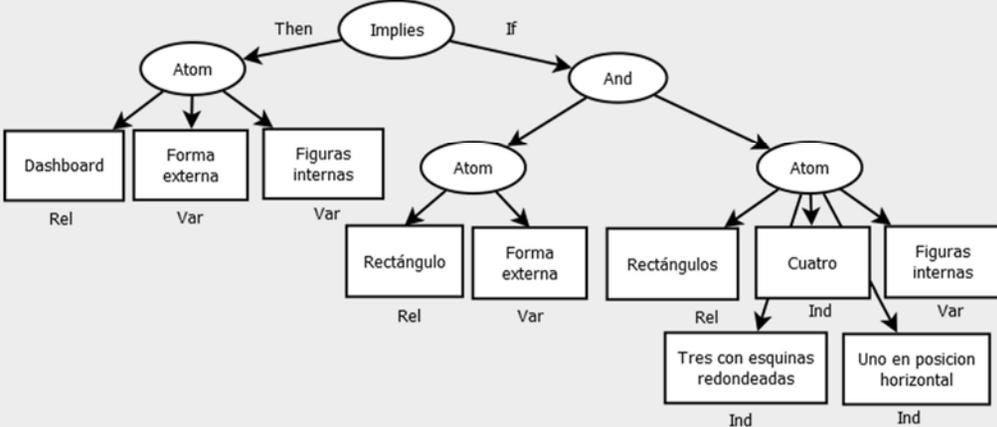
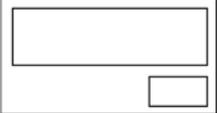
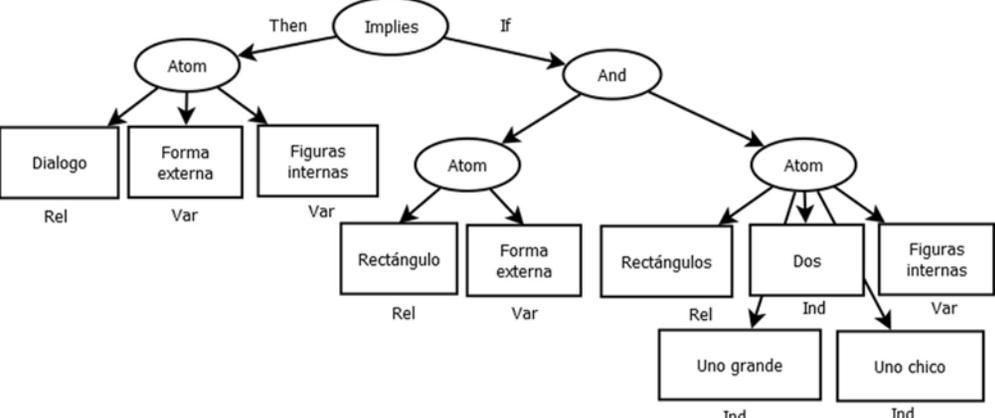
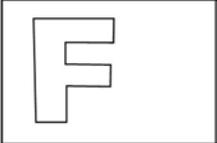
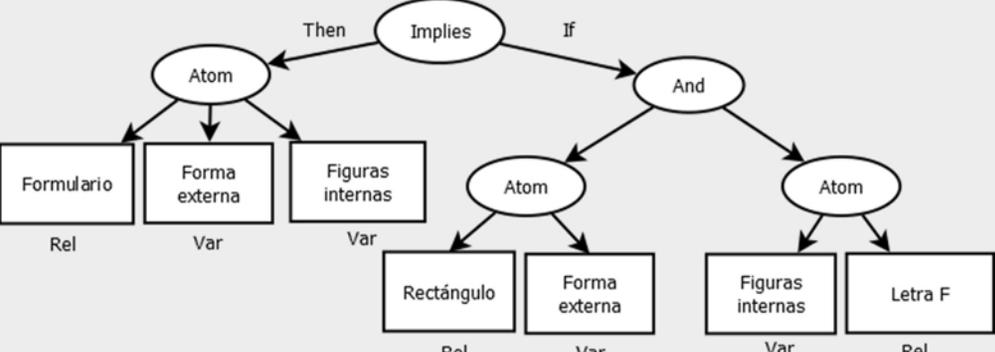
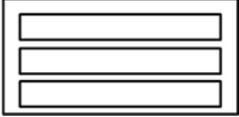
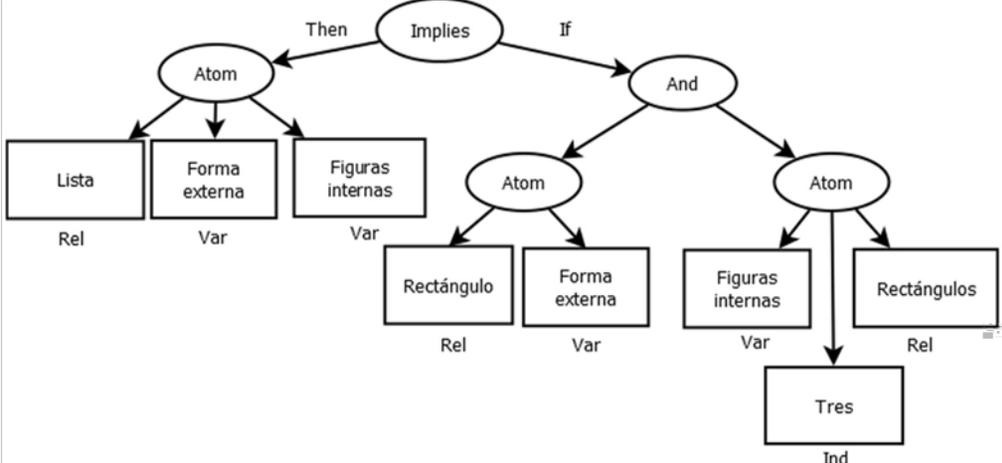
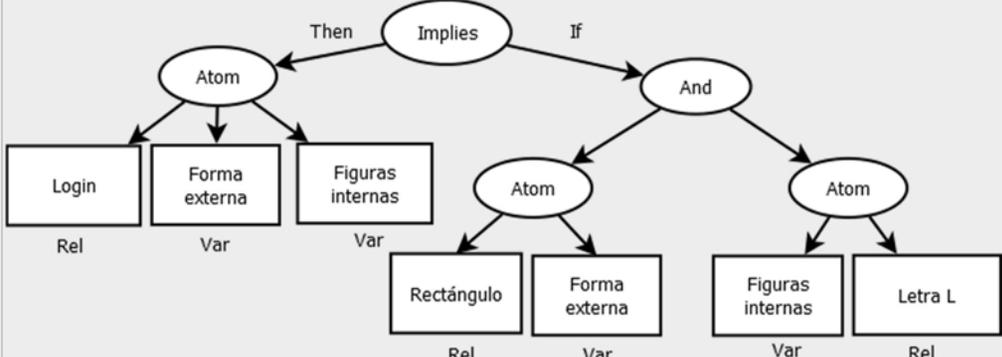
Patrones de diseño de IU	Icono simple	Árbol parcial de reglas
Splashscreen		
Table		
Video		

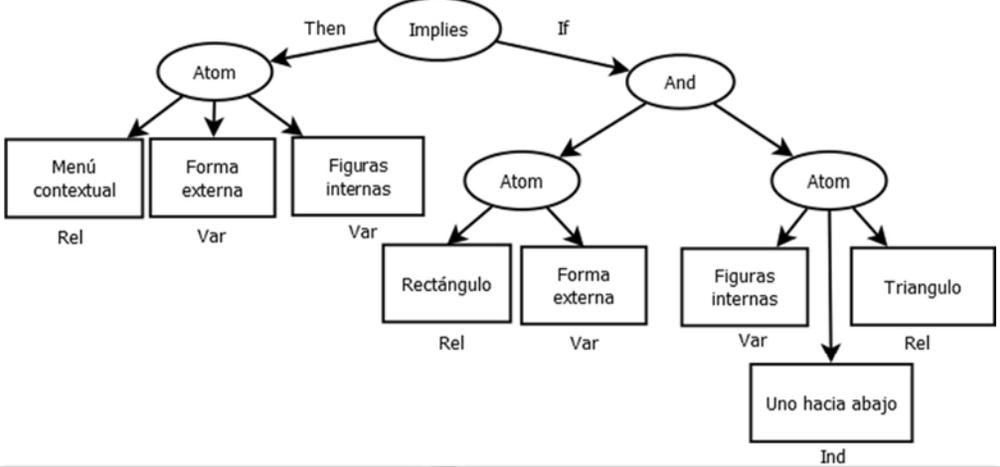
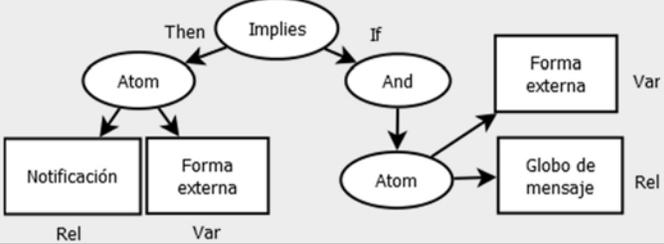
Tabla 3.16 Árboles de reglas para patrones en composición.

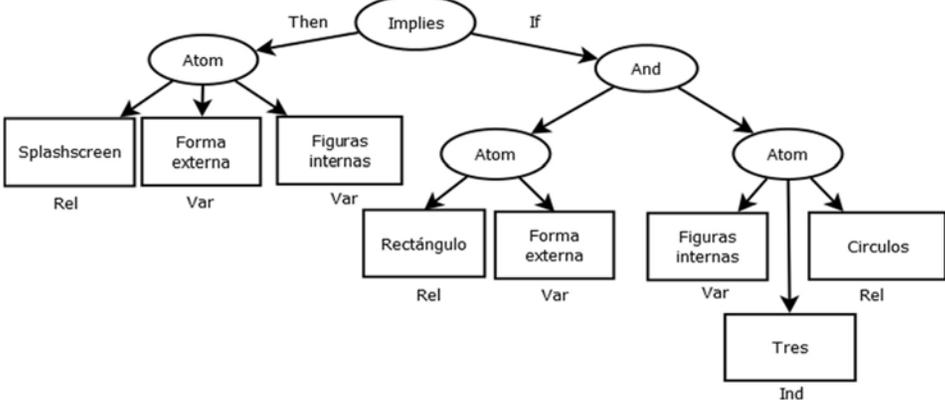
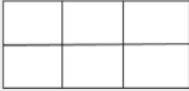
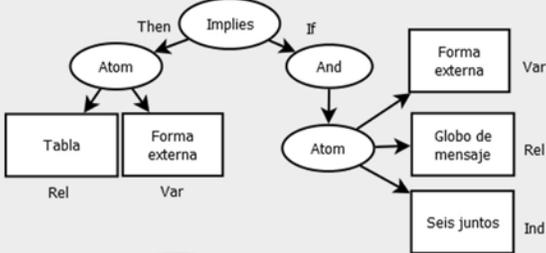
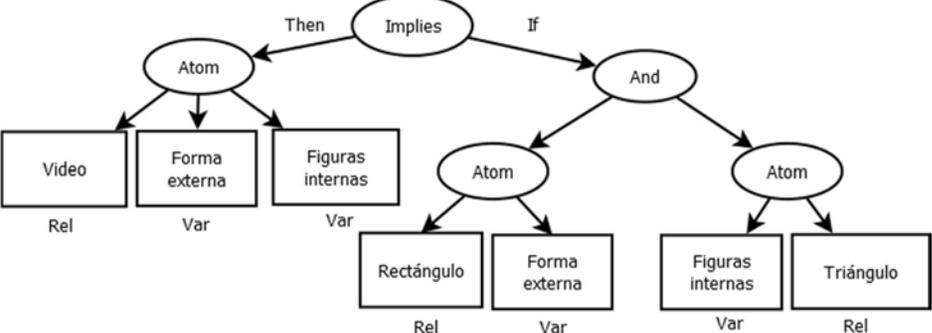
Patrones de diseño de IU	Icono para composición	Árbol parcial de reglas
<p>Accordion</p>		
<p>Wizard</p>		
<p>Progress bar</p>		

Patrones de diseño de IU	Icono para composición	Árbol parcial de reglas
Carousel		
Dashboard		

Patrones de diseño de IU	Icono para composición	Árbol parcial de reglas
Dialog		
Form		

Patrones de diseño de IU	Icono para composición	Árbol parcial de reglas
<p>List</p>		
<p>Login</p>		

Patrones de diseño de IU	Icono para composición	Árbol parcial de reglas
Contextual menu		
Notification		

Patrones de diseño de IU	Icono para composición	Árbol parcial de reglas
Splashscreen		
Table		
Video		

3.2.8 Características de la red neuronal usando Matlab

Dentro del proceso de desarrollo de aplicaciones multi-dominio se consideró la generación de aplicaciones a partir del reconocimiento de patrones de diseño de interfaz de usuario en imágenes generadas a mano alzada. Partiendo de lo anterior, las técnicas aplicadas para la identificación de elementos en imágenes parten de técnicas de inteligencia artificial como el uso de redes neuronales artificiales. En este sentido, el objetivo de implementar redes neuronales durante el proceso de identificación es, garantizar una mayor eficiencia en la identificación de elementos en imágenes. Es por ello que se seleccionaron un conjunto de características que servirían de apoyo en la identificación de elementos a través del uso de una red neuronal. Desde esta perspectiva, las características de configuración de la red neuronal y los parámetros de entrada empleados se describen a continuación.

El tipo de red neuronal empleada es un Perceptrón Multicapa que utiliza el algoritmo denominado activación de propagación hacia atrás (*Back Propagation*). Para los parámetros de configuración de la red neuronal, se realizaron un conjunto de pruebas para obtener tasas aceptables de entrenamiento y evaluación, es decir, valores superiores al 90%. Los parámetros utilizados durante el entrenamiento y la evaluación de la red neuronal se describen a continuación.

- 1) *Cuatro momentos Hu*: estos son parámetros estadísticos, invariables al tamaño y la posición de una figura dentro de una imagen.
- 2) *Número de objetos por patrón*: este parámetro considera el número de objetos encontrados en la interfaz descartando las posibles letras.
- 3) *El número de objetos, incluidas las letras*: este número incluye el total de figuras por cada patrón más las letras.
- 4) *Número de firmas de la figura del círculo*: representa el número de objetos identificados con el elemento de forma circular.
- 5) *Número de firmas de la figura del rectángulo*: Representa la cantidad de objetos encontrados dentro del UIDP que es un rectángulo.
- 6) *Número de firmas de la figura cuadrado*: representa el número de objetos dentro del UIDP que es un elemento cuadrado.

7) *Número de firmas de la figura del triángulo*: representa el número de objetos contenidos en el UIDP que es un elemento triangular.

8) *Número de firmas del contorno de una línea*: representa la cantidad de objetos dentro del UIDP que es una línea.

9) *Número de firmas del contorno del elemento acordeón*: representa el número de contornos encontrados dentro del UIDP igual al contorno del elemento acordeón.

10) *Número de firmas del contorno del elemento Menú contextual*: representa el número de objetos con contorno igual al elemento Menú contextual.

11) *Número de firmas del contorno del elemento Lista*: Representa el número de objetos identificados dentro del UIDP igual al contorno del elemento de lista.

12) *Número de firmas de la figura Óvalo*: Representa el número de objetos con una estructura igual a la figura Óvalo.

13) *Número de firmas de la figura flecha hacia la derecha*: Representa la cantidad de objetos encontrados dentro del UIDP a identificar que tiene el mismo contorno que la flecha hacia la derecha.

14) *Número de firmas de la figura flecha hacia la izquierda*: representa el número de objetos encontrados dentro del UIDP a identificar que tienen contorno igual al de una flecha hacia la izquierda.

15) *Número de firmas del contorno de una letra L*: representa el número de objetos dentro del UIDP que tienen contorno igual al de una letra L.

El número de firmas de figuras como círculos, rectángulos, triángulos y cuadrados considerados como parámetros de entrada se debe a que cada UIDP U está formado por las figuras geométricas mencionadas. Además, se agregó las firmas de los elementos más complejos. Cabe mencionar que, para seleccionar los parámetros descritos en el párrafo anterior, se realizó un conjunto de pruebas las cuales consistieron en entrenar y evaluar con diferentes parámetros de entrada a la red neuronal hasta obtener una configuración donde los porcentajes de eficiencia en la identificación de elementos fueran superiores al 90 % durante el entrenamiento y evaluación.

3.2.9 Características de la red neuronal convolucional usando Inception_V3

Como parte de las técnicas aplicadas para la identificación de elementos en imágenes generadas a mano alzada, se consideró el uso de técnicas de Deep Learning con el objetivo de mejorar el tiempo y eficiencia en la identificación de elementos en imágenes reales a color y con diferentes texturas en el lienzo del papel del dibujo. Por lo anterior, el nuevo modelo después de un conjunto de pruebas comparativas con otros modelos de redes convolucionales se implementó con Inception_V3.

Inception_V3 es un tipo de red neuronal convolucional modular, para su implementación se usó la API TensorFlow 1.14 y la versión de Keras 2.2.5 con la versión de Anaconda 1.9.6, así como la versión de Spyder 3.6. La red es ganadora del reto ImageNet 2014, esta red dio mejores resultados con respecto a otros modelos como AlexNet. Además, la red cuenta con 22 capas profundas a las 8 de AlexNet, tiene mejor eficiencia con respecto al poder computacional, casi dos veces menor al proceso en AlexNet, una mayor precisión, menor uso de la memoria, contiene menos parámetros, y es muy usada para dispositivos móviles. Además de ser modular, Inception_V3 permite escoger automáticamente el kernel convolucional mejorando el poder computacional. La red usa filtros de reducción de dimensiones de capas convolucionales 1x1 para convertir automáticamente una imagen RGB de 256x256x3 a 256x256x1. Además, la red tiene capas convolucionales de 3x3, 5x5 y de 7x7 dependiendo la elección del kernel convolucional. Al final, la red aplica Max-Pooling para mejorar los filtros de los algoritmos (ver figura 3.21).

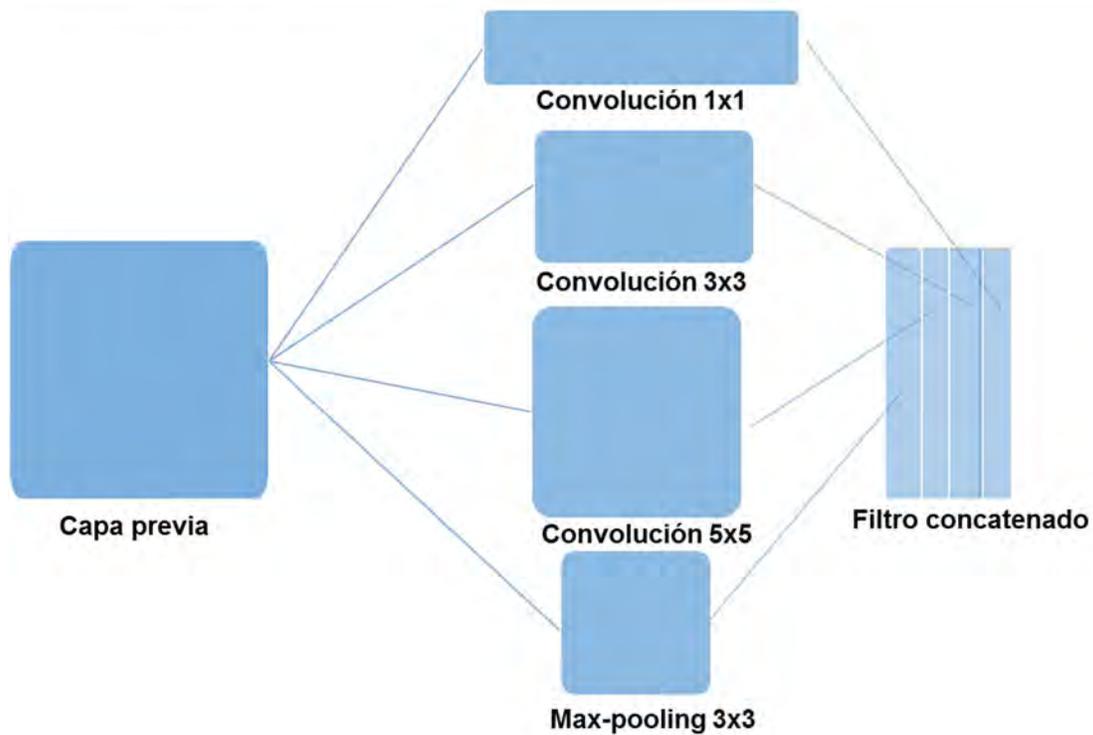


Figura 3.21 Arquitectura de la red neuronal convolucional Inception_v3.

3.3 Etapa de Desarrollo

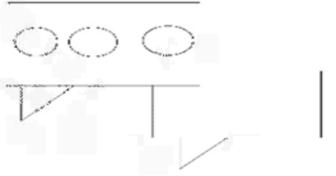
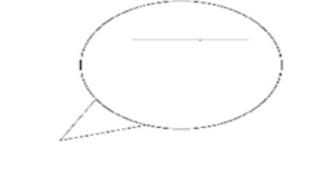
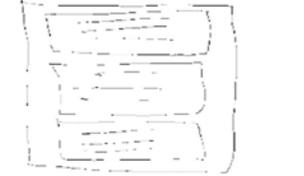
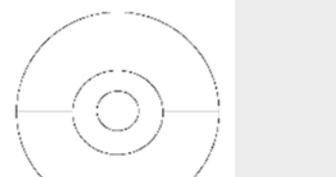
Durante esta etapa se utilizan los diferentes modelos descritos en la etapa anterior para la implementación del conjunto de módulos de software necesarios para la generación de aplicaciones multi-dominio. Dicho lo anterior, durante esta etapa se describen las características de los componentes de código propuestos para la generación automática de código. También, la configuración y resultados de entrenamiento de la red neuronal usada para la identificación de elementos en imágenes generadas a mano alzada.

3.3.1 Repositorio de imágenes

Para realizar las pruebas de entrenamiento y evaluación de los modelos analizados, implementados con redes neuronales se usó un conjunto de imágenes clasificadas en 4 tipos: Ideales, No ideales, Reales en blanco y negro, y Reales a color. Todas estas imágenes se encuentran almacenadas en un repositorio que cuenta con más de 10,000 muestras. Por cada patrón de diseño se tienen 400 imágenes de las

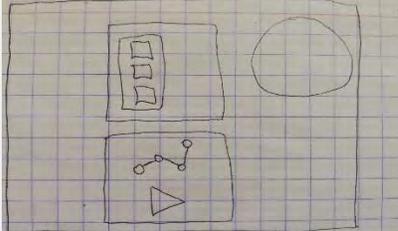
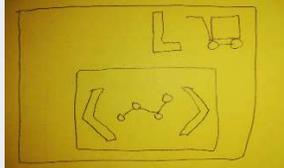
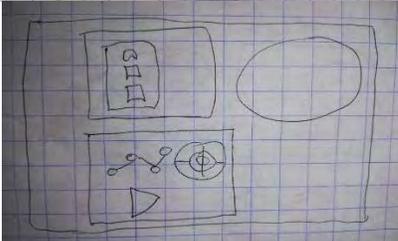
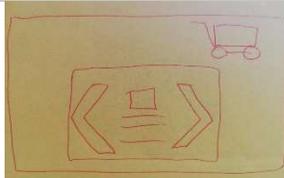
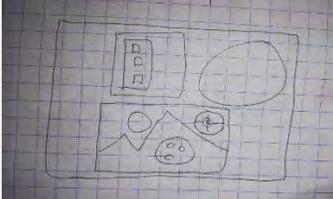
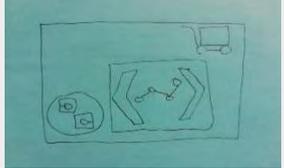
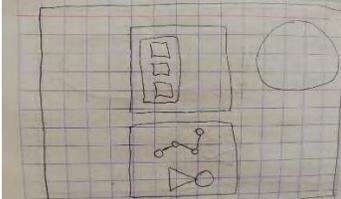
cuales 100 corresponden a muestras Ideales, 100 a muestras No ideales, 100 Reales en blanco y negro, y las 100 restantes son muestras de imágenes Reales a color. Las imágenes del repositorio pertenecen a los tres dominios de aplicaciones soportados: educativo, e-commerce y social media. La resolución de las imágenes es de 1050 x 650 pixeles, todas en formato PNG. En la Tabla 3.17 se muestran algunos ejemplos de las imágenes ideales y no ideales generadas para UIDPs simples.

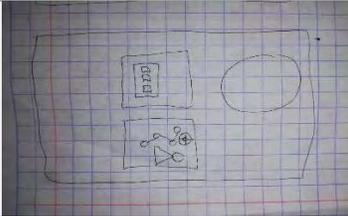
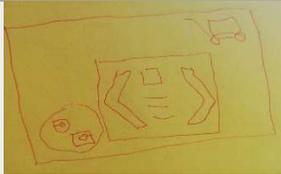
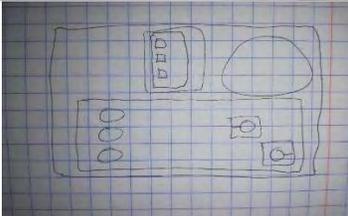
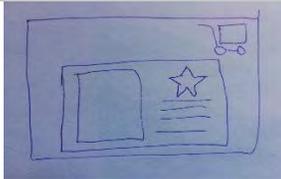
Tabla 3.17 Imágenes ideales y no ideales generadas para patrones de diseño simples.

UIDP	Imágenes ideales	UIDP	Imágenes no ideales
Chat		Horizontal menu	
Comments		Vertical menu	
Gallery		Music player	
Panel		Shopping cart	
Picker		Ranking	

A continuación, en la Tabla 3.18 se muestran ejemplos de las imágenes reales en blanco y negro, y las imágenes reales a color usando composiciones. En este caso, las composiciones de la Tabla 3.18 llamadas Composición 1-social media hasta Composición 8-social media pertenecen a las primeras 8 composiciones del dominio social media presentadas en la Tabla 3.12. respecto a las composiciones llamadas Composición 1-e-commerce hasta Composición 8-e-commerce pertenecen a las primeras 8 composiciones propuestas para el dominio e-commerce que se presentaron en la Tabla 3.11.

Tabla 3.18 Imágenes reales en blanco y negro, así como reales a color generadas para composiciones.

Composición	Imagen Real en blanco y negro	Composición	Imagen real a color
Composición 1-social media		Composición 1-e-commerce	
Composición 2-social media		Composición 2-e-commerce	
Composición 5-social media		Composición 3-e-commerce	
Composición 6-social media		Composición 4-e-commerce	

Composición	Imagen Real en blanco y negro	Composición	Imagen real a color
Composición 7-social media		Composición 5-e-commerce	
Composición 8-social media		Composición 6-e-commerce	

3.3.2 Módulo inicial de procesamiento de imágenes con Matlab

Tal como se describe en la etapa de diseño, para el reconocimiento de elementos en imágenes generadas a mano alzada se diseñó un conjunto de algoritmos a partir del procesamiento de imágenes y se complementó con el uso de redes neuronales artificiales. Partiendo de lo anterior, se desarrolló un primer módulo de procesamiento de imágenes implementado en MatLab (*Matrix Laboratory*). Matlab es un software matemático que proporciona una biblioteca de funciones: *Image Processing Toolbox*, para procesamiento de imágenes (The Mathworks Inc., 2016). La *Image Processing Toolbox* de Matlab proporciona un conjunto bastante amplio de algoritmos y herramientas gráficas para el procesamiento, análisis y visualización de imágenes digitales. El módulo de Matlab identifica a través de una red neuronal un conjunto de patrones de diseño de interfaz de usuario. Los patrones de diseño admitidos por este módulo hasta el momento, son los patrones seleccionados para el dominio educativo de los cuales el módulo soporta: *Accordion*, Asistente, Barra de progreso, *Carousel*, *Dashboard*, *List*, *Login*, Menú contextual, *Notification*, *Splashscreen* y Video. Por otra parte, soporta algunas composiciones: Carrusel de videos, Navegación básica, *Dashboard* con lista, *Dashboard* con video y lista de videos.

El módulo de Matlab tiene un conjunto de funciones que permiten extraer un conjunto de características que para la identificación de cada patrón de diseño en la imagen. El conjunto de características se obtiene al segmentar la imagen de

entrada para obtener la posición de cada patrón de diseño de IU en la imagen; otra función obtiene un conjunto de firmas por patrón de diseño de IU, dependiendo de las figuras que conforman cada patrón mencionado en la etapa de diseño de este documento. Otra función obtiene el número de objetos encontrados de acuerdo con las firmas obtenidas por la función anterior. Con las características obtenidas, la red neuronal identifica cada patrón en la imagen de entrada.

Las principales funciones y algoritmos utilizados para el procesamiento de imágenes con Matlab son:

- *bwlabel*: se utiliza para etiquetar los píxeles de cada objeto en la imagen de manera única lo que es útil en el análisis. Así también algunas de sus propiedades, como obtener la imagen del objeto.
- *bwmorph*: realiza algunas operaciones morfológicas en el objeto de la imagen.
- *imerode*: se usa para erosionar la imagen y obtener los bordes.
- *rgb2gray*: se usa para convertir una imagen en color a escala de grises.

El esquema de funcionamiento del módulo comienza con primero (1) cargar la imagen en la memoria. (2) Una vez que se carga la imagen, se aplica un conjunto de operaciones morfológicas para obtener una imagen con solo el contorno. (3) Luego, un proceso de segmentación obtiene para cada elemento las coordenadas de su posición en la imagen, las coordenadas se almacenan en una matriz. (4) El siguiente paso es obtener un conjunto de firmas. De acuerdo con las coordenadas almacenadas de la imagen original, se obtiene un recorte del elemento, el recorte se envía a otra función para obtener un conjunto de firmas. La técnica de la firma implica obtener la distancia del centroide de la figura a un pixel límite de la misma figura, de 1 a 360 grados. Una vez que se han obtenido las firmas de todas las figuras que componen cada elemento, se almacenan todas en una matriz.

(5) Posteriormente, con la matriz de firmas, se obtiene en otra función el número de figuras que conforman al cada elemento; esto se hace a través de la correlación cruzada entre la firma de cada figura a identificar y las firmas de todas las formas ideales: círculos, rectángulos, cuadrados, triángulos, líneas. La figura se asocia con el valor más alto que se aproxima a la firma de cada figura ideal respectivamente.

(6) Finalmente, se obtienen otras características como: los primeros 4 momentos de *Hu* y el número de objetos, incluidas las letras. Todas las características son utilizadas por la red neuronal para determinar qué patrones de diseño es.

Una vez que el elemento es identificado, algunas de sus características se almacenan en una matriz, como el color en formato RGB, el ancho, el alto, las posiciones (X, Y) y el nombre del elemento identificado. Si el elemento no es identificado, no se almacenan datos de él. Después de los pasos anteriores, se genera un archivo .txt con datos de todos los elementos identificados. Este archivo contiene en cada línea separada por comas las características de cada elemento. Una vez hecho esto, el módulo finaliza el procesamiento de la imagen.

3.3.2.1 Red neuronal

Para complementar el proceso de identificación de elementos se aplicó la configuración de la red neuronal con las características descritas en la etapa de diseño. Partiendo de lo anterior, se desarrolló el conjunto de pruebas de entrenamiento y evaluación de la red neuronal, los resultados se presentan en esta sección.

Las características de las imágenes que se utilizaron para el entrenamiento y evaluación de la red neuronal son imágenes en formato .png con dimensiones de 1050 x 650 píxeles. Todos los bocetos utilizados en las pruebas se generaron con una tableta digitalizadora Wacom Intuos. La tableta incluye su propio editor de dibujo, sin embargo, las imágenes se generaron con la herramienta de dibujo "*Paint*". Las imágenes se generaron teniendo en cuenta el formato y las dimensiones establecidos. Para cada boceto se extrajeron las características descritas en la etapa de diseño: los 4 momentos de *Hu*, el número de objetos encontrados basado en el número de firmas, el número de objetos encontrado considerando letras, el número de círculos, el número de rectángulos, el número de cuadrados, el número de triángulos, el número de líneas, número de elementos acordeón, número de elementos menú contextual, número de elementos lista, número de óvalos, número de flechas hacia la derecha y número de flechas hacia la izquierda. En la figura 3.22, se muestran algunos ejemplos de las imágenes utilizadas durante el entrenamiento y la evaluación de la red neuronal.

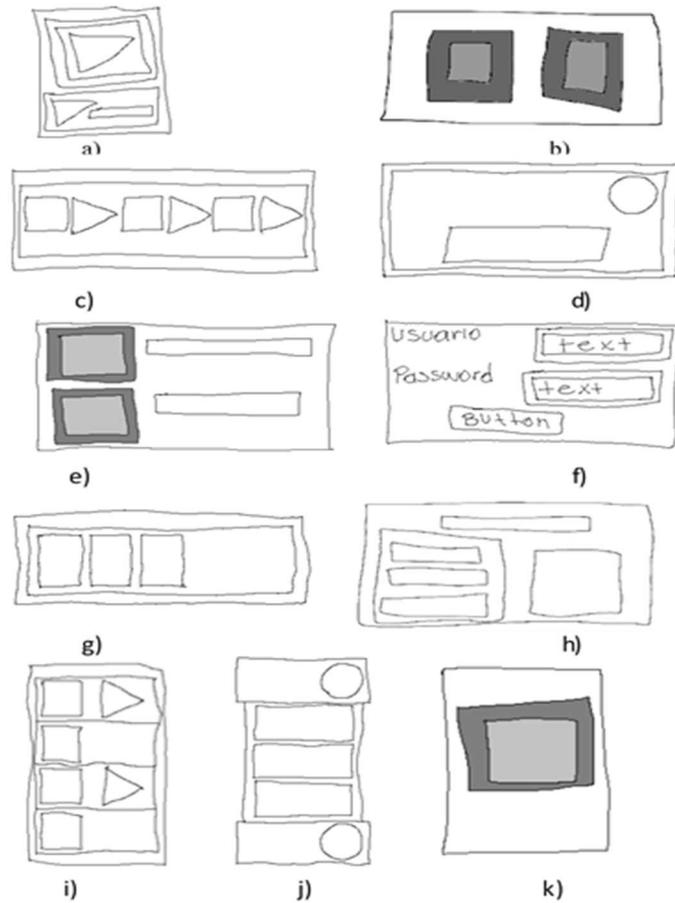


Figura 3.22 UIDPs de muestra: a) *Video*, b) *Carousel*, c) *Wizard*, d) *Notification*, e) *List*, f) *Login*, g) *Progress bar*, h) *Dashboard*, i) *Contextual menu*, j) *Accordion* y k) *Splashscreen*.

Prueba 1: para realizar esta prueba, la red neuronal se configuró utilizando una capa oculta y diferente número de neuronas. La Tabla 3.19, presenta el número de prueba, el número de neuronas, porcentaje de eficiencia obtenido durante el entrenamiento, y el porcentaje de eficiencia obtenido durante la evaluación.

Tabla 3.19 Resultados de entrenamiento y evaluación con 1 capa oculta.

No. Prueba	No. de neuronas	% Eficiencia en entrenamiento	% Eficiencia en evaluación
1.1	8	68.1%	No aplica
1.2	9	67.8%	No aplica
1.3	10	78.0%	No aplica
1.4	11	98.1%	90.6%

No. Prueba	No. de neuronas	% Eficiencia en entrenamiento	% Eficiencia en evaluación
1.5	12	80.4%	No aplica
1.6	13	86.9%	No aplica
1.7	14	98.3%	91.1%
1.8	15	98.8%	90.3%
1.9	16	99.1%	91.1%
1.10	17	98.9%	87.5%
1.11	18	98.9%	90.5%
1.12	19	99.8%	88.3%
1.13	20	99.4%	91.9%
1.14	21	99.6%	92.3%
1.15	22	99.8%	90.9%

Prueba 2: en esta prueba, se utilizó una configuración de dos capas ocultas en la red neuronal con diferente número de neuronas en cada capa. La Tabla 3.20 muestra los porcentajes de eficiencia en entrenamiento y evaluación de la red neuronal con diferentes configuraciones.

Tabla 3.20 Resultados de entrenamiento y evaluación con 2 capas ocultas.

No. Prueba	No. de neuronas Capa 1 / Capa 2	% Eficiencia en entrenamiento	% Eficiencia en evaluación
2.1	8/8	68.8%	No aplica
2.2	8/9	68.8%	No aplica
2.3	8/10	75.0%	No aplica
2.4	8/11	93.8%	83.9%
2.5	8/12	81.1%	No aplica
2.6	8/13	87.5%	No aplica
2.7	8/14	93.8%	84.8%
2.8	8/15	99.6%	92.8%
2.9	8/16	99.6%	91.4%
2.10	9/10	68.8%	No aplica
2.11	9/11	93.5%	86.0%
2.12	9/12	80.8%	No aplica
2.13	9/15	99.9%	91.8%

Cuando el porcentaje de eficiencia en el entrenamiento es inferior al 90%, la evaluación no se realizó, por lo que algunas celdas en la columna de evaluación en ambas tablas (Tabla 3.22 y 3.23) indican no aplica.

Como se observa en ambos casos, se obtienen porcentajes de entrenamiento y evaluación superiores al 90%, por lo que el reconocimiento de Patrones de Diseño de IU a través de redes neuronales es totalmente factible.

Debido a que se obtuvo más de una red neuronal con porcentajes de eficiencia superiores al 90%, el criterio para seleccionar la red neuronal se basó en el porcentaje de eficiencia en la evaluación. La configuración final fue la red neuronal obtenida en la prueba 2, con 18 parámetros de entrada, dos capas ocultas con 8 y 15 neuronas, así como un 99.6% de eficiencia en el entrenamiento y un 92.8% de eficiencia en la evaluación.

3.3.3 Módulo de procesamiento de imágenes con Deep Learning

El módulo final se obtuvo después de un conjunto de pruebas con diferentes redes de convolucionales, por lo que en esta sección se describen a detalle todas las pruebas realizadas.

Antes de describir las pruebas se presenta un proceso de transformación de las imágenes o pre-procesamiento que fue implementado en las pruebas del modelo con *AlexNet* e *Inception_V3*.

3.3.3.1 Proceso de transformación de imágenes

Una parte importante de la etapa de análisis en nuestro proceso de desarrollo propuesto es el proceso de transformación o pre-procesamiento de la imagen. El objetivo de este proceso es mejorar la calidad de las imágenes para facilitar la identificación de elementos en los diferentes tipos de imágenes. En este caso, los tipos de imágenes soportados por este proceso son las descritas en la sección 3.1.1, desde esta perspectiva, los estándares óptimos para alimentar de una red neuronal como *AlexNet* que cuenta con una resolución de la imagen de 256 x 256 píxeles. Mientras que *Inception_v3* tiene una resolución de 255 x 255 píxeles para cada

imagen. Por lo tanto, es necesario reorientar la resolución de las imágenes para el proceso neuronal convolucional mejorando la calidad de precisión de las imágenes. En este sentido, para obtener imágenes con un estándar específico y limpio es necesario pasar las imágenes de color a imágenes en blanco y negro, esto ayuda completamente a disminuir el consumo computacional y por lo tanto a mejorar la eficiencia en el proceso de identificación de elementos.

El proceso de transformación para el modelo con Inception_v3 consiste en a partir de la imagen original (ver Figura 3.23-a) con una dimensión total de 1050 x 650 píxeles, redimensionarla a 256 x 256 píxeles. Posteriormente, se elimina el ruido por medio de métodos y funciones de algoritmos ya implementados, las cuales son: (a) *mediaBlur*, una función que suaviza la imagen; (b) dilatación que permite el aumento de los objetos a primer plano; y (c) la normalización, que se encarga de optimizar los valores altos y bajos de una matriz (ver Figura 3.23-b). Finalmente, se aplica la función (d) *Local Otsu*, encargada de optimizar el umbral del pixel aclarando y maximizando los píxeles en dos clases, primer plano y fondo (ver Figura 3.23-c).

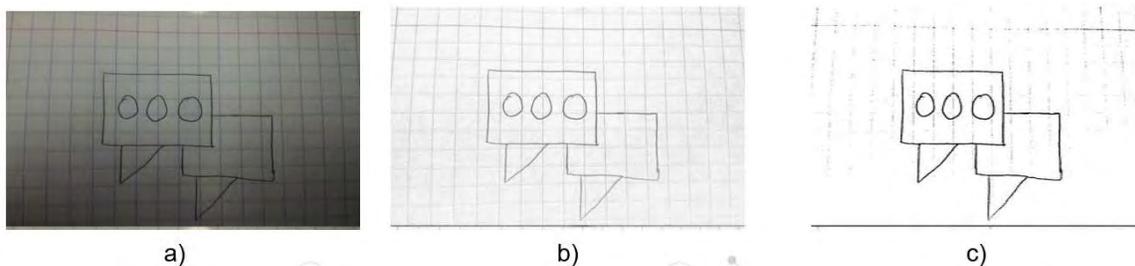


Figura 3.23 Proceso de transformación: (a) Imagen original, (b) eliminación de sombras, (c) umbral optimizado.

Estos métodos y funciones utilizan las APIs *scikit-image* versión 0.15.0 y *opencv* versión 2.4.13.1, que son bibliotecas especiales y esenciales que contienen diversos algoritmos para el procesamiento de imágenes.

3.3.3.2 Pruebas con red neuronal

Antes de iniciar con la descripción y resultados de cada prueba, es importante mencionar que para realizar cada prueba se usaron imágenes generadas previamente descritas en la sección 3.1.1 con dimensiones de 1050 x 650 píxeles en formato PNG.

Las siguientes pruebas generadas permitieron conocer ampliamente las características que conforman las redes convolucionales tales como: los atributos, métodos y funciones que se requieren para madurar una red convolucional.

Prueba 1: En esta primera prueba se trabajó la API DeepLearning4j y el modelo LeNet. La configuración de la red de convolucional conto con 5 capas de convolución. Los patrones usados en esta prueba fueron 5 (*accordion*, *Navigation Bar*, *Login*, *Master-detail* y *Target*). Se usaron 50 imágenes reales en blanco y negro por patrón, haciendo un total de 250 imágenes. Se tuvieron 3 épocas con 256 batch por época. Los resultados obtenidos fueron por debajo de 45% de precisión. El tiempo estimado de entrenamiento fue de 32 horas. Mientras que el tiempo de respuesta del modelo en la identificación de un elemento fue de 21.32 segundos en promedio.

Prueba 2: esta prueba se realizó con la API de *TensorFlow* con un modelo binario de edición libre de *TensorFlow*. La configuración de la red de convolución consta de tres capas; una capa *flattening*, una capa *fully-connected* y la capa de salida. Para esta prueba se usaron 2 patrones de diseño, galería y reproductor de música. De cada patrón se usaron 300 imágenes de las cuales 100 fueron imágenes ideales, otras 100 imágenes no ideales y 100 más fueron imágenes reales a blanco y negro; en total 600 imágenes para el entrenamiento de la red neuronal. Se contó con una configuración de 100 épocas, obteniendo un batch de 1699 por época, con una duración de aproximada de 6 días. En esta prueba se pretendía ver la capacidad de tiempo de entrenamiento de la red de convolucional y ver la precisión que obtendría dicha red neuronal, el resultado contó con una precisión con un 100% de forma positiva para la detección del patrón. Los resultados se observan en la Tabla 3.21.

Tabla 3. 21 Resultados de la prueba con *TensorFlow* y el modelo binario para: galería y reproductor de música.

UIDP	Porcentaje obtenido	Tiempo (segundos)
Gallery	1.0000000	30.01
Music player	3.1575022e-13	45.0

Prueba 3: en esta prueba se usó la API de *TensorFlow* con el modelo binario de edición libre, con tres capas de convolución: una capa flattening, una capa fully-connected y la capa de salida. Se contó con una configuración de 10 épocas, obteniendo un batch de 1401 por época, con duración de 2 días. Esta prueba resolvería ver la precisión que genera el modelo de clasificación para 4 patrones (*Gallery, Horizontal menu, Vertical menu, y Panel*). Para cada patrón se usaron 200 imágenes de las cuales 100 fueron imágenes ideales y otras 100 imágenes no ideales, en total 800 imágenes para el entrenamiento. Los resultados no fueron muy prometedores, ya que la arquitectura de la red neuronal no lo permitía. Tal como se observa en la Tabla 3.22:

Tabla 3.22 Resultados de TensorFlow y modelo binario para: *Gallery, Horizontal menu, Vertical menu y Panel*.

UIDP	Porcentaje obtenido	Tiempo (segundos)
Gallery	0.4339219	23.407269
Horizontal menu	0.422323233	23.407269
Vertical menu	0.123212333232	23.407269
Panel	0.0221412	23.407269

Prueba 4: en esta prueba se usó la API de *TesorFlow* y el modelo *AlexNet* con cinco capas de convolución, una capa para aplanar el nivel, una capa completamente conectada y una capa de resultados. La elección de *AlexNet* es por su forma simple, además mantiene una tasa de error que comprende del 26% a 15.3%, soporta en su configuración algoritmos como *max-pooling, dropuot, data argumentation* y *ReLU* que permiten una estandarización de los datos. Además, no hay reglas para el tamaño o dimensiones para cada capa de convolución. Los patrones usados en esta prueba fueron: *Gallery, Horizontal menu, Vertical menu, Music-player, Panel, Picker* y *Ranking*. Se contó con una configuración de 10 épocas, obteniendo un batch de 699 por cada época, con una duración de aproximada de 8 horas de entrenamiento. Para esta prueba se optó por aplicar un conjunto de algoritmos de procesamiento de imágenes, cuyo objetivo es pre-procesar las imágenes para eliminar el ruido lo que permita mejorar los porcentajes de eficiencia en el reconocimiento de elementos

en imágenes. Este proceso se describe en la siguiente sección con mayor detalle, mientras, en la Tabla 3.23 se observan los resultados obtenidos en esta prueba.

Tabla 3.23 Resultados de *TesorFlow* y el modelo *AlexNet* para los patrones: Gallery, Horizontal menu, Vertical menu, Music player, Panel, Picker y Ranking.

UIDP	Porcentaje obtenido	Tiempo (segundos)
Gallery	0.99983644	8.24
Horizontal menu	0.56006614	8.32
Vertical menú	0.99995962	8.40
Music player	0.9624471	8.18
Panel	0.9980167	7.98
Picker	0.8734556	7.91
Ranking	0.98580235	7.85

Los resultados ponen a consideración que los datos obtenidos son bastante óptimos en estas primeras pruebas obteniendo casi el 100% de eficiencia en la identificación de cada patrón, y un tiempo estándar de 8 segundos. Con tal respuesta se ve claramente el impacto que genera una red neuronal convolucional para la clasificación de imágenes que representan patrones de diseño de interfaz de usuario. Sin embargo, se realizaron otras pruebas con otros como AlexNet e Inception_V3. Las pruebas se describen a continuación.

Entrenamiento con AlexNet

Prueba 1: La importancia de hacer esta prueba es observar los resultados y validar el alcance que tiene el modelo, otro punto a resaltar es la implicación del proceso de transformación que ayuda a mejorar y concentrar las imágenes necesarias para el entrenamiento. Con 9 patrones y 200 instancias, el modelo AlexNet cumple con los resultados de precisión, debido a que se tomaron los cuatro primeros resultados obtenidos. En la Tabla 3.24 se observan valores donde 1, se toma como un correcto acercamiento de los cuatro primeros lugares cercanos al 100% y 0 si no se encuentra en los cuatro primeros lugares de clasificación cercano al 100%. La configuración de la red para esta prueba consistió en 1199 batch con un total de 10 épocas y un tiempo de entrenamiento de 8.7 horas, dando una precisión de 88%.

Tabla 3.24 Resultados de entrenamiento usando AlexNet con 9 patrones y 200 instancias por UIDP.

	Chat	Gallery	Picker	Panel	Comment	Horizontal menu	Vertical menu	Music player	Ranking
Chat	1	0.00016250 6	0.19352973	0.00047347 1	0.06813639 4	0.002334441	1.16E-05	0.03520408	0.00334989 1
Gallery	0.01306084 6	1	0.00120266 9	4.30E-05	0.00774126 6	0.001050286	5.16E-05	0.006903645	0.01054452 5
Picker	0.00045928 6	8.85E-05	1	4.91E-05	0.10553073	0.000117258	0.003349891	0.002373413	0.00052385 3
Panel	0.00055751	3.29E-05	0.00564839 5	1	0.02310633 7	0.301504659	0.000206908	0.000674044	0.00044756 4
Comment	0.00051446 3	0.00166992 7	0.14482954	0.00012649 2	1	0.450005678	4.71E-06	0.26613018	0.000487
Horizontal menu	0.00228702 1	0.00092562 4	0.00191804 3	0.00102294 9	0.01277389 9	0	0.000415436	0.000210459	0.00214533 7
Vertical menu	0.0000544	0.0004578	0.00053249 6	0.00056593 1	0.00123855 9	0.24860461	1		0.00132029 3
Music player	0.00020929 4	0.00062694 4	0.03488034 4	2.61E-05	0.01874780 7	0.00584537	2.07E-05	1	0.00144582 7
Ranking	0.00018263 3	0.00226597 7	.0044776	.001279	0.00106866 7	0.003140203	4.57E-05	0.001355291	1

Prueba 2: En esta segunda prueba se utilizaron los mismos 9 UIDPs, sin embargo, el número de instancias incremento a 400 imágenes por UIDP. Por lo tanto, el entrenamiento se realizó con 3600 imágenes el doble de la prueba anterior, esto fue gracias a una nueva versión del proceso de transformación de la imagen. La configuración de la red fue de 1199 batch, 10 épocas, y un tiempo de entrenamiento de 11.6 horas, dando una precisión del 100%. El objetivo de esta prueba fue ver el comportamiento de la red con el doble de instancias. En la Tabla 3.25 se registran los resultados del entrenamiento.

Tabla 3.25 Resultados de entrenamiento usando AlexNet con 9 patrones y 400 instancias por UIDP.

	Chat	Gallery	Picker	Panel	Comment	Horizontal menu	Chat	Gallery	Picker
Chat	1	0.24458238	0.09221611	0.00931159	0.07781014	0.19702998	0.01267048	0.00077627	0.00737012
				7	6		5	9	7
Gallery	0.03856194	1	0.05090722	0.00241594	0.00509466	0.041085303	0.00674395	3.76E-05	0.00132175
	8		4	2	4		9		6
Picker	0.018224	0.00054664	4	0.00681989	0.0458856	0.012618925	0.00014516	0.00017796	0.00582297
		8		7			4	5	4
Panel	0.00683337	0.02781803	0.13985293	1	0.27250868	0.061365534	0.00085999	0.00022490	0.00905620
	8	2					5	5	8
Comment	0.00513589	0.01412527	0.05282312	0.0510871	1	0.019245375	0.00055173	1.04E-05	0.00609590
	1	9					9		1
Horizontal menu	0.19797277	0.06933659	0.00417035	0.00107846	0.01261253	1	0.00241951	0.00010604	0.00109108
				8			1	5	5
Vertical menu	0.00688794	0.05686648	0.05500244	0.00328556	0.00587123	0.06844314	1	0.00015222	0.16353118
	8		3	2	4			2	
Music player	0.00503937	0.00241547	0.10943875	0.00045557	0.00086624	0.010981618	0.01885675	1	0.00131632
	4	3		2	7		8		6
Ranking	0.00076302	0.00236688	0.00432572	0.00067317	0.00108938	0.004886695	0.00044957	7.39E-05	1
	8	2	7	3	8		1		

Prueba 3: Para esta prueba se utilizaron 17 UIDPs, 9 UIDPs simples y las primeras 8 composiciones del dominio social media descritos en la Tabla 3.12. Para cada instancia se usó un conjunto de 400 imágenes, esto es 6800 instancias en total. La configuración de la red fue de 1499 batch, 10 épocas, y un tiempo de entrenamiento de 19.4 horas; dando una precisión del 52% en la identificación de elementos. El motivo de aumentar 10 UIDPs en esta prueba fue saber el comportamiento de la red neuronal, y como se observa la

precisión se redujo considerablemente. En la Tabla 3.26 se registran los resultados obtenidos del entrenamiento, donde los términos C1-s hasta C6-s, hacen referencia a las primeras 6 composiciones del dominio social media (ver Tabla 3.12).

Tabla 3.26 Resultados de entrenamiento usando AlexNet con 17 UIDPs y 400 instancias por UIDP (Parte 1).

	C1-s	C2-s	C3-s	C4-s	C5-s	C6-s	Chat
C1-s	1	0.24458238	0.09221611	0.009311597	0.077810146	0.19702998	0.012670485
C2-s	0.038561948	0	0.050907224	0.002415942	0.005094664	0.041085303	0.006743959
C3-s	0.018224	0.000546648	0	0.006819897	0.0458856	0.012618925	0.000145164
C4-s	0.006833378	0.027818032	0.13985293	1	0.27250868	0.061365534	0.000859995
C5-s	0.005135891	0.014125279	0.05282312	0.0510871	0	0.019245375	0.000551739
C6-s	0.19797277	0.06933659	0.00417035	0.001078468	0.01261253	0	0.002419511
Chat	0.006887948	0.05686648	0.055002443	0.003285562	0.005871234	0.06844314	1

Tabla 3.26 Resultados de entrenamiento usando AlexNet con 17 UIDPs y 400 instancias por UIDP (Parte 2).

	Gallery	Picker	Panel	Comment	Horizontal menu	Vertical menu	Music player	Ranking
Gallery	1	0.001316326	6.29E-05	0.009131086	0.00077163	6.96E-05	0.00872514	0.012969824
Picker	7.39E-05	1	5.29E-05	0.11054075	5.36E-05	.000547877	0.00272435	0.000436317
Panel	2.20E-05	0.004038893	1	0.021036336	0.000931349	0.000236409	0.000579792	0.000409668
Comment	0.001374114	0.13027096	0.000151663	1	0.027818032	4.98E-06	0.24293001	0.0001934
Horizontal menu	0.000637018	0.001296212	0.001056242	0.00960097	0	0.00058558	0.000174559	0.002006637
Vertical menu		0.000409019	0.000904508	0.001131709	0.16476302	1		0.001281688
Music player	0.000615163	0.03418452	3.42E-05	0.019978378	0.000356791	2.25E-05	1	0.001332199
Ranking	0.002439373	.0024757	3.20E-05	0.001259505	0.002316048	6.23E-05	0.001457633	1

Prueba 4: En esta prueba se utilizaron 27 UIDPs con 400 imágenes cada uno, esto es 10,800 imágenes. De los 27 UIDPs, 8 son composiciones de social media, 8 son composiciones de e-commerce, y 11 son UIDPs simples. La configuración para esta prueba fue de 2993 batch, 10 épocas y un tiempo de entrenamiento de 37.3 horas; así como una precisión de 43.3%. En este caso, la precisión

sigue disminuyendo mientras más UIDPs se clasifiquen, por lo tanto, *AlexNet* en esta última prueba no logra un resultado satisfactorio para reconocer los patrones. En la Tabla 3.27 se registraron los resultados obtenidos del entrenamiento donde, los términos C1-s hasta C8-s pertenecen a las primeras 8 composiciones del dominio social media (Ver Tabla 3.12). Los terminos C1-e hasta C8-e pertenecen a las primeras 8 composiciones del dominio e-commerce listadas en la Tabla 3.11.

Tabla 3.27 Resultados de entrenamiento usando AlexNet con 27 UIDPs y 400 instancias por UIDP (Parte 1).

	C1-s	C2-s	C3-s	C4-s	C5-s	C6-s	C7-s	C8-s	Chat	Galle ry	Picke r	Panel	Com ment	Horizon tal menu	Vertic al menu	Music playe r	Ranki ng
C1-s	1 32829	0.092	0.069 03471	0.003 90426	0.024 25053	0.031 27537	0.014 12037	0.006 89028	0.011 27594	0.000 73639	0.003 29104	0.000 33662	0.000 55417	0.13392 355	0.0003 17341	9.42E -05	0.000 57598
C2-s	0.017 77823	0	0.004 90830	0.002 89152	0.003 95139	0.015 00138	0.092 46051	0.000 65035	0.008 19108	0.000 22932	0.001 75655	2.32E -05	0.011 90351	0.01022 4656	5.36E- 05	0.000 13029	0.002 87941
C3-s	0.013 37651	0.001 02191	0 90830	0.006 87921	0.029 11335	0.004 71696	0.001 29313	0.006 50771	0.000 37061	0.000 48995	0.003 73450	3.41E -05	0.002 17446	0.01996 3393	8.12E- 05	0.000 19933	0.002 45533
C4-s	0.004 20760	0.018 85186	0.016 63662	0 87921	0.054 50768	0.021 16119	0.048 47296	0.003 39090	0.000 95049	0.000 43940	0.007 94825	3.99E -05	0.013 54297	0.00153 0593	2.97E- 06	0.000 25445	0.000 61601
C5-s	0.003 31273	0.008 78274	0.002 09723	0.020 01338	1 69273	0.006 69273	0.077 12532	0.003 03118	0.000 56714	2.70E -05	0.005 45062	1.89E -05	0.003 66088	0.00026 8517	2.35E- 06	9.87E -05	0.000 24071
C6-s	0.055 77554	0.049 47731	0.003 46896	0.001 19103	0.007 47697	0 57481	0.047 57481	0.000 36018	0.002 43066	0.000 16646	0.001 53350	2.03E -05	0.000 67068	0.00092 0208	1.31E- 05	0.000 41630	0.001 77474

	C1-s	C2-s	C3-s	C4-s	C5-s	C6-s	C7-s	C8-s	Chat	Galle ry	Picke r	Panel	Com ment	Horizon tal menu	Vertic al menu	Music playe r	Ranki ng
C7-s	0.002 30558 9	0.018 84681	0.000 36990 9	0.000 16281 5	0.000 35469 2	0.031 54679 8	1	0.000 24924 2	0.000 70661 9	3.90E -06 1	0.000 29311 1	5.82E -07 5	0.001 20136	0.00162 6083	0.0081 01461	1.71E -05	0.000 29562
C8-s	0.007 36999 3	0.003 29495 1	0.071 92188	0.015 74792 9	0.081 96105	0.030 80185 3	0.001 16141 5	1	0.000 46712 3	1.84E -05 3	7.97E -05 5	0.000 54651 5	0.000 19598 9	0.14984 247	7.72E- 05	2.05E -05	0.000 63175
Chat	0.004 36719 9	0.041 29565	0.006 61078 2	0.001 88419 1	0.001 43435 4	0.029 46979 2	0.001 98691 2	5.42E -05	1	0.000 43793 3	0.058 37088 5	0.000 16346 3	0.030 44183 2	0.00102 5596	1.11E- 05	0.019 35448 1	0.004 85952
Gallery	0.001 71210 2	0.001 43546 9	0.013 05518 2	0.000 31032 6	0.000 23844 6	0.004 19297 5	7.75E -05	1.27E -06	0.010 95861 7	1	0.000 43886 7	3.10E -05 3	0.002 74330	0.00043 4709	5.87E- 05	0.004 62401 4	0.017 64701 9
Picker	0.000 80096 3	0.002 74907	0.000 24815 5	0.000 80885 5	0.000 57484 4	0.003 72854 6	0.002 07176 3	1.38E -05	0.000 59818 4	0.000 21542 2	0	2.98E -05	0.058 69282	5.17E- 05	1.42E- 06	0.002 63075 8	0.000 49505 8
Panel	0.000 42930 4	0.001 05682 3	0.008 00842 5	0.002 60976 7	0.002 34859 1	0.007 96173 1	0.000 23127 7	5.08E -05	0.000 88143 4	7.61E -05 1	0.002 51184 1	1	0.010 64465 6	0.00171 4562	0.0007 17219	0.000 46736 4	0.000 53758 5
Comme nt	0.000 42355 8	0.001 51988 1	0.000 95145 6	0.000 34116	0.000 33254 1	0.003 16445	0.000 68312	4.10E -06	0.000 82917 9	0.004 33993	0.106 60472	9.57E -05	0	4.35E- 05	6.26E- 06	0.251 25784	0.000 28112 2
Horizon tal menu	0.000 33423 5	0.000 19434 5	0.017 23277	0.000 72923 5	0.005 79948 4	0.003 52423 6	7.69E -05	0.000 69619 6	0.001 54992 7	0.000 78608 6	0.000 45175 8	0.000 57105 9	0.002 89762 6	0	0.0012 57752	8.30E -05	0.001 83819 2

	C1-s	C2-s	C3-s	C4-s	C5-s	C6-s	C7-s	C8-s	Chat	Galle ry	Picke r	Panel	Com ment	Horizon tal menu	Vertic al menu	Music playe r	Ranki ng
Vertical menu	0.000 34236 6	0.000 22047 9	0.005 33479 2	0.001 19497 3	0.002 09385 6	0.001 19673 3	9.66E -05 9	4.50E -05 9	0.000 11512 9	2.51E -05 3	0.000 26622 3	0.001 62256 8	0.000 73907 8	0.17851 332	1	4.13E -05	0.001 78224 2
Music player	0.001 79305 6	0.000 89866 7	0.002 56202 7	0.001 45740 2	0.000 84503 5	0.003 10036 1	0.000 43427	1.07E -05	0.000 42065 1	0.002 03752 2	0.026 73712	2.69E -05	0.013 85125 5	0.00025 1632	2.24E- 05	0	0.001 60067 8
Rankin g	0.000 13363 2	7.31E -05	0.000 93439 8	6.46E -05	5.50E -05	0.000 47462	4.68E -04	1.19E -06	0.000 60332 9	0.008 72071 6	0.000 25821 9	2.80E -05	0.000 81125 2	0.00229 3031	9.50E- 05	0.001 42691 3	1
C1-e	0.108 81881 4	0.079 71123 6	0.067 04838	0.003 70420 6	0.001 71124 1	0.152 0008	0.017 12563 6	5.10E -05	0.009 11392 2	0.004 36325 9	0.000 43777 7	8.18E -06	0.000 30396 6	0.01674 3302	0.0001 18941	0.000 11738 5	0.001 96425
C2-e	0.068 83622	0.027 32702 5	0.010 10791 8	0.004 28653 1	0.000 41699 6	0.022 53540 8	0.007 01835 4	3.92E -05	0.002 72021	0.001 66476 4	0.001 46421 7	4.64E -05	0.010 79172 6	0.01185 0477	0.0003 92441	5.20E -05	0.006 36875
C3-e	0.006 51681 1	0.000 51992 4	0.051 42821	0.003 38299 1	0.013 17794 7	0.008 13607 2	0.000 62055 4	0.003 18605 8	0.000 23326 2	0.000 51094 4	0.002 66419 3	1.79E -05	0.001 41259 2	0.01718 0104	4.86E- 05	0.000 14048	0.002 97176 5
C4-e	0.004 24569 9	0.014 58559 7	0.019 06332 2	0.175 93181	0.048 69090 4	0.017 89418 8	0.047 64857	0.003 33644 1	0.000 80298 3	0.000 34261 2	0.009 01257 3	3.78E -05	0.016 30851 4	0.00174 9163	3.36E- 06	0.000 27155 9	0.000 50779 8
C5-e	0.002 38524 5	0.008 07921 5	0.001 91578 6	0.016 67592 5	0.071 4309	0.006 43720 7	0.053 04356 7	0.002 54646	0.000 52150 9	2.70E -05	0.004 86443	1.52E -05	0.002 77509 2	0.00020 0427	1.83E- 06	8.48E -05	0.000 18738

	C1-s	C2-s	C3-s	C4-s	C5-s	C6-s	C7-s	C8-s	Chat	Galle ry	Picke r	Panel	Com ment	Horizon tal menu	Vertic al menu	Music playe r	Ranki ng
C6-e	0.098 21650 4	0.081 56671 4	0.005 84473 5	0.001 79352	0.009 10125 5	0.116 31474	0.064 16419	0.000 29998 3	0.002 77656	0.000 34260 6	0.001 77082	1.25E -05 7	0.000 71038 7	0.00115 3213	1.90E- 05	0.000 54681	0.002 12367 4
C7-e	0.013 08052 3	0.010 81210 9	0.004 04028 9	0.007 26324 2	0.002 62687 4	0.018 38058	0.000 86151 4	0.000 12692	0.000 85350 3	0.000 62449 6	0.014 25195 9	0.055 9938 1	0.007 70779 1	0.00577 3085		0.000 25223 9	0.000 20537 6
C8-e	0.016 08332 6	0.039 68190 4	0.000 96411 5	0.003 17336 5	0.003 59487 1	0.012 70717 2	0.015 61844 4	0.000 26641 7	0.000 28170 4	8.20E -05 6	0.001 22285 6	1.02E -05 6	0.001 74287 6	0.00077 055 1449	0.0003 1449	0.000 14434 2	0.000 87668 5
Coupo n	0.021 72192 7	0.048 93888 5	0.094 56854	0.161 31629	0.024 47903 9	0.081 24992	0.013 55337	0.000 10599 8	0.013 28016 9	0.000 30854 3	0.007 29345 5	6.00E -05 48088	0.082 48088	0.00085 8983	9.73E- 05	0.003 09893 4	0.001 93271
Pricing Table	0.006 58993 3	0.002 50041 9	0.079 6556	0.075 6525	0.106 81879 5	0.026 28711 2	0.001 10982 9	0.001 91084 2	0.000 84656 8	0.001 76792 7	0.001 48006 5	3.39E -05 79679	0.004 79679 3	0.01440 5541	0.0003 3979	0.000 30567 7	0.002 97906 3
Produc t Page	0.034 51763 5	0.024 45214 4	0.095 2573	0.014 95513	0.008 27961 6	0.082 11774	0.005 22711 8	0.000 23215 2	0.056 35922 8	0.003 16525 3	0.003 06722 8	9.05E -05 50268	0.009 50268 8	0.16148 17	0.0023 05698	0.000 43611 8	0.026 66924 9
Shoppi ng Cart	0.001 00847 1	0.004 66453 3	0.016 53029 4	0.005 67667 9	0.012 97587 2	0.049 10123	0.000 75217 1	0.000 36475 6	0.007 16058 9	0.002 06044 9	0.061 17518	0.003 80176 7	0.005 80262 4	0.00151 6702	6.12E- 05	0.007 40864 4	0.002 87440 4
Tip a Tip	0.020 16737 7	0.021 95081 9	0.004 26350 5	0.001 29190 2	0.001 46177 4	0.022 46218 4	0.005 52359 4	8.95E -06	0.058 72440 3	0.071 40732 6	0.288 54692	5.32E -05 1359	0.314 1359	0.00197 9704	3.98E- 05	0.135 96803	0.003 28211 7

Tabla 3.27 Resultados de entrenamiento usando AlexNet con 27 UIDPs y 400 instancias por UIDP (Parte 2).

	C1-e	C2-e	C3-e	C4-e	C5-e	C6-e	C7-e	C8-e	Coupo n	Pricing Table	Product Page	Shopping Cart	Tip a Tip
C1-s	0.00122 2212	0.00849 8555	0.00021 5067	0.00283 6069	0.00022 9817	0.04884 9013	0.21250 871	0.00239 7288	0.0003 237	5.25E-05	0.011576 437	0.001370 623	0.00036 4582
C2-s	0.00845 4538	0.00711 6523	0.00288 0617	0.01723 857	0.00602 7459	0.00470 0975	0.00195 9268	0.00078 1094	0.0094 663	0.000325 914	0.003758 69	0.000270 088	0.00030 5973
C3-s	0.00350 2943	0.02410 948	0.00251 941	0.01284 2772	0.00464 4138	0.00096 4578	0.01327 5475	0.00738 9479	0.0002 795	0.003510 809	0.000336 711	0.000224 094	0.00026 4911
C4-s	0.00678 6767	0.00132 4369	0.01028 0787	0.00463 7126	0.01809 6015	0.00016 4309	0.00502 6802	0.00359 7574	0.2109 172	0.003747 971	2.33E-05	0.000543 934	4.69E- 05
C5-s	0.00892 0413	0.00219 9738	0.04634 3703	0.00616 3945	0.01336 6538	0.00014 3894	0.00296 5571	0.00397 718	0.1847 462	0.001238 715	0.011577 657	6.07E-05	9.80E- 06
C6-s	0.00111 1792	0.00359 2488	0.01187 2754	0.00086 2901	0.00118 6524	0.00374 9183	0.01968 7437	0.00021 9391	0.0007 428	6.69E-05	0.000167 043	0.001341 972	1.14E- 05
C7-s	0.00026 3427	0.00033 2229	0.02197 8298	0.00309 6926	0.00054 7146	0.04890 9252	0.00037 99	1.90E- 05	0.0295 478	9.97E-05	1.10E-05	0.000252 415	9.47182 -07
C8-s	0.00128 9974	0.00877 2186	0.00024 078	0.00047 3908	0.00031 7529	0.00113 867	0.00105 5921	0.00038 3838	0.0004 692	0.000194 933	0.002555 163	0.000269 062	8.86E- 06
Chat	0.16868 266	0.06935 984	0.12407 381	0.01999 2307	0.01780 9797	0.00223 6657	0.03075 584	0.00062 9054	0.0082 497	0.004791 313	0.003621 182	0.001252 51	0.00617 0376
Gallery	0.03107 5047	0.00646 7854	0.00355 9057	0.00667 2331	0.00590 452	0.00512 0111	0.00034 8792	0.00010 5513	9.50E- 05	0.009002 4	0.175296 13	0.000420 665	0.19699 919
Picker	0.00255 2322	0.00153 1312	0.01884 6426	0.00070 5575	0.00502 7881	0.00010 1329	0.00066 649	8.02E- 05	0.0417 174	0.000744 131	1.01E-05	0.000490 869	8.93E- 05
Panel	0.00056 2693	0.00417 4547	0.00052 7666	0.00107 1659	0.00083 9345	0.00758 0689	0.01061 434	0.00016 6724	0.0005 838	0.001433 217	0.001397 139	0.001098 01	4.67E- 05
Comment	0.00420 3074	0.00507 7907	0.02365 4275	0.00094 883	0.01013 6013	0.00141 5114	0.00237 0329	0.00013 232	0.0079 052	0.000229 172	7.76E-05	0.002243 017	0.00254 0476

	C1-e	C2-e	C3-e	C4-e	C5-e	C6-e	C7-e	C8-e	Coupo n	Pricing Table	Product Page	Shopping Cart	Tip a Tip
Horizontal menu	0.00062 6785	0.00909 5392	0.00019 937	0.01173 5376	0.00333 8289	0.00745 432	0.00242 4878	0.00516 0174	0.0001 423	0.042052 638	0.005427 169	0.001212 45	0.00023 6373
Vertical menu	0.00014 3034	0.00444 3634	5.02E- 05	0.00278 95	0.00023 0323	0.02512 6405	0.02092 0135	0.00458 4726	9.37E- 05	0.000428 463	0.002489 994	0.000136 83	1.89E- 05
Music player	0.00235 1448	0.00339 9683	0.00151 6045	0.00106 3561	0.00454 646	0.00056 8046	0.00214 2363	4.03E- 05	4.232E -05	0.000469 517	0.000309 979	0.000900 845	0.00107 112
Ranking	0.00066 0585	0.00233 7842	0.00053 3077	0.00119 4976	0.00028 9639	0.00064 6346	0.00019 3118	1.14E- 05	0.0004 701	0.008386 015	0.005001 901	0.000370 232	0.00182 1808
1-e	0	0.08202 814	0.01079 4097	0.05539 7544	0.01186 005	0.05525 4135	0.01737 3513	0.00075 1836	0.0001 199	0.001565 359	0.028398 156	0.000493 001	0.00392 3169
C2-e	0.01120 3741	0	0.00438 4674	0.01322 0945	0.02722 5675	0.00793 0527	0.00593 4994	0.00014 7061	0.0035 094	0.002674 006	0.029865 11	0.000111 4	0.00104 6189
C3-e	0.00193 9175	0.01278 5941	0	0.00860 987	0.00300 8344	0.00054 6343	0.00627 8422	0.00394 3317	0.0001 442	0.002984 697	0.000268 68	0.000193 247	0.00018 2618
C4-e	0.00617 0199	0.00146 8711	0.00977 3295	0	0.01848 7513	0.00017 3705	0.00589 0452	0.00366 0766	0.1006 177	0.003963 298	2.31E-05	0.000534 539	5.05E- 05
C5-e	0.00618 9026	0.00154 7917	0.02748 1943	0.00444 096	0	9.05E- 05	0.00156 5817	0.00301 4107	0.1422 796	0.001210 387	1.71E-06	0.001424 923	7.06E- 06
C6-e	0.00182 0942	0.00577 3742	0.01917 6602	0.00148 9398	0.00169 5659	1	0.02865 2415	0.00030 191	0.0009 944	0.000107 239	0.000242 739	0.001273 446	2.00E- 05
C7-e	0.00230 7526	0.01128 3241	0.00120 4547	0.03987 8763	0.00516 7937	0.26872 11	0	0.00049 5149	0.0007 083	0.000284 01	0.012815 164	0.003300 189	0.00021 9851
C8-e	0.03498 1206	0.02021 6295	0.02193 2047	0.26377 65	0.15138 28	0.00264 9222	0.03144 8323	1	0.0012 535	0.001240 397	1.60E-05	0.000161 003	5.43E- 05
Coupon	0.11141 771	0.06242 381	0.18111 774	0.01644 7896	0.16463 314	0.00378 7947	0.00125 288	0.00038 7939	0	0.015185 01	0.002384 813	0.000577 009	0.00103 9606

	C1-e	C2-e	C3-e	C4-e	C5-e	C6-e	C7-e	C8-e	Coupo n	Pricing Table	Product Page	Shopping Cart	Tip a Tip
Pricing Table	0.01390 6231	0.04128 359	0.00327 6981	0.10608 325	0.00981 0957	0.00029 8047	0.00519 3083	0.19870 315	0.0010 569	0	6.36E-05	0.002928 054	6.34E- 05
Product Page	0.03100 0808	0.08600 34	0.00494 0166	0.00444 4527	0.08930 9745	0.00825 8119	0.00954 0436	0.00287 7597	0.0008 145	0.040694 542	0	0.002605 956	0.00267 4726
Shopping Cart	0.00341 6761	0.00441 7317	0.00624 7829	0.00126 6395	0.00299 8278	0.00298 3385	0.21376 601	0.00283 8019	0.0005 201	0.001849 188	0.001198 713	1	0.00071 1557
Tip a Tip	0.08044 1624	0.04718 3674	0.00129 5769	0.08085 692	0.06581 27	0.04638 522	0.03645 029	0.00185 8326	0.0107 401	0.007196 794	0.016118 055	0.004834 933	1

A continuación, se describen las pruebas con el modelo Inception_V3 que sirvieron para comparar los resultados obtenidos con todos los modelos anteriormente descritos, y así determinar cuál sería la mejor opción para el procesamiento e identificación de elementos en imágenes generadas a mano alzada.

Pruebas con Inception_V3

Prueba 1: En esta prueba se utilizaron 9 UIDPs con 200 imágenes cada uno, esto es 1800 instancias. La configuración fue de 500 batch, 100 épocas, 6.3 horas de entrenamiento, y se obtuvo una precisión de 88%. El objetivo de esta prueba fue observar el comportamiento con respecto a AlexNet, y los resultados obtenidos verifican que esta prueba tiene la misma tasa de error que la prueba 1 de AlexNet. En la Tabla 3.28, se observan los resultados obtenidos del entrenamiento.

Tabla 3.28 Resultados de entrenamiento usando Inception_V3 con 9 patrones y 200 instancias por UIDP.

	Chat	Comment	Gallery	Horizontal menú	Vertical menu	Music player	Panel	Picker	Ranking
Chat	1	0.16358	0.04076	0.10806	0.06382	0.01578	0.15495	0.14627	0.08844
Comment	0.03356	1	0.03502	0.01721	0.03612	0.04643	0.13155	0.12357	0.07479
Gallery	0.03578	0.11136	1	0.07059	0.03841	0.02072	0.02647	0.11297	0.22975

	Chat	Comment	Gallery	Horizontal menú	Vertical menu	Music player	Panel	Picker	Ranking
Horizontal menu	0.05365	0.0202	0.0357	1	0.0262	0.00527	0.04673	0.04739	0.50558
Vertical menu	0.02378	0.07978	0.12727	0.09396	0	0.02127	0.13903	0.11171	0.30199
Music player	0.00811	0.62126	0.04349	0.00424	0.0149	1	0.09181	0.07292	0.03582
Panel	0.07911	0.32926	0.05673	0.08863	0.1117	0.02002	1	0.11249	0.08259
Picker	0.0317	0.3057	0.09098	0.04878	0.02888	0.01698	0.0281	1	0.04972
Ranking	0.05903	0.12544	0.0694	0.0685	0.06121	0.01539	0.05159	0.15177	1

Prueba 2: En esta prueba se utilizaron 9 UIDPs con 400 imágenes cada uno, 3800 imágenes en total. La configuración para esta prueba fue de 500 batch, 100 épocas, 8.7 horas de entrenamiento, y una precisión de 100%. Para esta prueba se usó el proceso de transformación de la imagen, por lo que los resultados mejoraron si se comparan con la prueba 2 de *AlexNet*. En la Tabla 3.29 se muestran los resultados obtenidos del entrenamiento.

Tabla 3.29 Resultados de entrenamiento usando Inception_V3 con 9 UIDPs y 400 instancias por UIDP.

	Chat	Comment	Gallery	Horizontal menú	Vertical menu	Music player	Panel	Picker	Ranking
Chat	1.0000	0.1099	0.0154	0.0741	0.1613	0.0248	0.3284	0.0428	0.1255
Comment	0.0110	1.0000	0.0202	0.0168	0.1982	0.0830	0.1502	0.0620	0.0327
Gallery	0.0209	0.0703	1.0000	0.0183	0.0493	0.0156	0.0414	0.0239	0.6514
Menu- horizontal	0.0379	0.0077	0.0124	1.0000	0.0734	0.0155	0.0566	0.0130	0.4272
Menu- vertical	0.0128	0.0435	0.0368	0.0382	1.0000	0.0198	0.3943	0.0196	0.2611
Music player	0.0037	0.5016	0.0189	0.0021	0.0456	1.0000	0.1216	0.0299	0.0260

	Chat	Comment	Gallery	Horizontal menú	Vertical menu	Music player	Panel	Picker	Ranking
Panel	0.0222	0.0710	0.0090	0.0266	0.1887	0.0149	1.0000	0.0115	0.0317
Picker	0.0067	0.3957	0.0251	0.0357	0.1394	0.0459	0.0441	1.0000	0.0219
Ranking	0.0463	0.0519	0.0267	0.0168	0.0712	0.0144	0.1262	0.0308	1.0000

Prueba 3: En esta prueba se utilizaron 17 patrones con 400 imágenes cada uno, 6800 instancias en total. Para esta prueba, la configuración de la red neuronal fue de 500 batch, 100 épocas, 17.3 horas de entrenamiento, y se obtuvo una precisión de 83%. Con respecto a la prueba 3 de AlexNet y se observa que a pesar de que se aumentaron los UIDPs el porcentaje de precisión es superior al 80%, esto se debe a la disminución de errores que existe con la red neuronal de Inception_v3. Por lo tanto, se concluye que con Inception_V3 la precisión mejoro respecto a la prueba 3 de *AlexNet*. En la Tabla 3.30 se observan los resultados obtenidos del entrenamiento.

Tabla 3.30 Resultados de entrenamiento usando Inception_V3 con 17 UIDPs y 400 instancias por UIDP (Parte 1).

	Chat	Comment	Gallery	Horizontal menu	Vertical menu	Music player	Panel	Picker
Chat	0.00000	0.08880	0.01870	0.05369	0.10847	0.02138	0.18422	0.03406
Comment	0.01160	1.00000	0.02795	0.01246	0.10337	0.07369	0.11080	0.05399
Gallery	0.02634	0.07821	1.00000	0.02130	0.04466	0.02238	0.04868	0.02750
Horizontal menu	0.04045	0.01183	0.01511	1.00000	0.06968	0.01361	0.05984	0.01513
Vertical menu	0.01634	0.04984	0.03953	0.03876	1.00000	0.02447	0.29296	0.02329
Music player	0.00430	0.46879	0.03094	0.00195	0.03220	1.00000	0.10284	0.03282
Panel	0.01786	0.06664	0.01285	0.02054	0.11705	0.01557	1.00000	0.01199
Picker	0.00648	0.25056	0.03011	0.01811	0.07232	0.04055	0.03278	1.00000

Tabla 3.30 Resultados de entrenamiento usando Inception_V3 con 17 UIDPs y 400 instancias por UIDP (Parte 2).

	Ranking	C1-s	C2-s	C3-s	C4-s	C5-s	C6-s	C7-s	C8-s
Ranking	1.00000	0.03395	0.01339	0.00869	0.00784	0.01237	0.03815	0.01903	0.01710
C1-s	0.21545	1.00000	0.04459	0.02661	0.01480	0.01628	0.04332	0.02547	0.02157
C2-s	0.07749	0.09150	0.00000	0.00996	0.01944	0.05044	0.08482	0.08136	0.04509
C3-s	0.02678	0.07313	0.07401	1.00000	0.02536	0.03803	0.04841	0.06709	0.02168
C4-s	0.03589	0.03395	0.01339	0.00869	1.00000	0.01237	0.03815	0.01903	0.01710
C5-s	0.21545	0.01911	0.04459	0.02661	0.01480	1.00000	0.04332	0.02547	0.02157
C6-s	0.07749	0.09150	0.02168	0.00996	0.01944	0.05044	0.00000	0.08136	0.04509
C7-s	0.49122	0.02814	0.01520	0.01650	0.00578	0.00793	0.02172	1.00000	0.00848
C8-s	0.34856	0.02146	0.02168	0.04673	0.01455	0.00659	0.00672	0.00755	1.00000

Prueba 4: En esta prueba se utilizaron 27 patrones con 400 imágenes cada uno, 10,800 imágenes en total. La configuración de la red neuronal consistió en 500 batch, 100 épocas, 34.3 horas de entrenamiento, y una precisión del 74%. Para esta prueba se observó, una mejora con respecto a la prueba 4 de *AlexNet* a pesar de tener la misma cantidad de imágenes. Con respecto a la tercera prueba generada en Inception_v3, se observa una reducción en la precisión. Por lo tanto, este modelo tiene un mayor rendimiento que el modelo de AlexNet. En la Tabla 3.31 se presentan los resultados obtenidos.

Tabla 3.31 Resultados de entrenamiento usando Inception_V3 con 27 UIDPs y 400 instancias por UIDP (Parte 1).

	Chat	Comm ent	Galle ry	Horizontal menu	Vertical menu	Music player	Panel	Pick er	Ranki ng	C1-s	C2-s	C3-s	C4-s	C5-s
Chat	0.000 00	0.0888 0	0.018 70	0.05369	0.10847	0.02138	0.184 22	0.03 406	0.109 25	0.035 90	0.022 42	0.017 40	0.016 64	0.029 40
Comment	0.011 60	1.0000 0	0.027 95	0.01246	0.10337	0.07369	0.110 80	0.05 399	0.037 68	0.037 61	0.057 48	0.017 38	0.020 65	0.036 76

	Chat	Comm ent	Galle ry	Horizontal menu	Vertical menu	Music player	Panel	Pick er	Ranki ng	C1-s	C2-s	C3-s	C4-s	C5-s
Gallery	0.026 34	0.0782 1	1.000 00	0.02130	0.04466	0.02238	0.048 68	0.02 750	0.491 22	0.028 14	0.015 20	0.016 50	0.005 78	0.007 93
Horizontal menu	0.040 45	0.0118 3	0.015 11	1.00000	0.06968	0.01361	0.059 84	0.01 513	0.348 56	0.021 46	0.021 68	0.046 73	0.014 55	0.006 59
Vertical menu	0.016 34	0.0498 4	0.039 53	0.03876	1.00000	0.02447	0.292 96	0.02 329	0.219 72	0.041 72	0.022 14	0.012 02	0.005 59	0.006 96
Music player	0.004 30	0.4687 9	0.030 94	0.00195	0.03220	1.00000	0.102 84	0.03 282	0.035 89	0.019 11	0.012 68	0.010 18	0.003 70	0.005 28
Panel	0.017 86	0.0666 4	0.012 85	0.02054	0.11705	0.01557	1.000 00	0.01 199	0.038 60	0.087 17	0.025 80	0.009 22	0.011 93	0.018 10
Picker	0.006 48	0.2505 6	0.030 11	0.01811	0.07232	0.04055	0.032 78	1.00 000	0.026 78	0.073 13	0.074 01	0.014 05	0.025 36	0.038 03
Ranking	0.044 18	0.0624 7	0.035 36	0.01912	0.06569	0.01837	0.111 53	0.03 441	1.000 00	0.033 95	0.013 39	0.008 69	0.007 84	0.012 37
C1-s	0.017 27	0.0973 1	0.043 75	0.04408	0.11661	0.04019	0.079 89	0.02 728	0.215 45	1.000 00	0.044 59	0.026 61	0.014 80	0.016 28
C2-s	0.014 15	0.0197 1	0.006 89	0.04201	0.31074	0.00408	0.075 59	0.01 574	0.077 49	0.091 50	0.000 00	0.009 96	0.019 44	0.050 44
C3-s	0.006 48	0.2505 6	0.030 11	0.01811	0.07232	0.04055	0.032 78	0.03 282	0.026 78	0.073 13	0.074 01	1.000 00	0.025 36	0.038 03
C4-s	0.044 18	0.0624 7	0.035 36	0.01912	0.06569	0.01837	0.111 53	0.03 441	0.035 89	0.033 95	0.013 39	0.008 69	1.000 00	0.012 37
C5-s	0.017 27	0.0973 1	0.043 75	0.04408	0.11661	0.04019	0.079 89	0.02 728	0.215 45	0.019 11	0.044 59	0.026 61	0.014 80	1.000 00
C6-s	0.014 15	0.0197 1	0.006 89	0.04201	0.31074	0.00408	0.075 59	0.01 574	0.077 49	0.091 50	0.021 68	0.009 96	0.019 44	0.050 44

	Chat	Comm ent	Galle ry	Horizontal menu	Vertical menu	Music player	Panel	Pick er	Ranki ng	C1-s	C2-s	C3-s	C4-s	C5-s
C7-s	0.026 34	0.0782 1	0.019 71	0.02130	0.04466	0.02238	0.048 68	0.02 750	0.491 22	0.028 14	0.015 20	0.016 50	0.005 78	0.007 93
C8-s	0.040 45	0.0118 3	0.015 11	0.05984	0.06968	0.01361	0.059 84	0.01 513	0.348 56	0.021 46	0.021 68	0.046 73	0.014 55	0.006 59
C1-e	0.032 12	0.0115 4	0.112 34	0.07680	0.04670	0.04798	0.076 19	0.04 758	0.076 86	0.276 43	0.088 40	0.058 69	0.376 75	0.039 75
C2-e	0.063 22	0.0946 2	0.016 58	0.00645	0.06165	0.14647	0.012 77	0.01 640	0.075 60	0.016 47	0.041 66	0.046 57	0.332 54	0.056 27
C3-e	0.015 49	0.1245 7	0.091 58	0.02430	0.02468	0.04968	0.046 16	0.01 649	0.045 73	0.032 66	0.065 80	0.113 41	0.009 94	0.058 46
C4-e	0.005 64	0.0991 5	0.065 14	0.03417	0.07816	0.06779	0.016 54	0.01 976	0.076 78	0.064 34	0.054 00	0.086 01	0.057 91	0.021 80
C5-e	0.012 25	0.0466 5	0.006 46	0.04457	0.09679	0.14673	0.017 76	0.01 550	0.031 64	0.043 57	0.047 86	0.027 12	0.070 24	0.086 68
C6-e	0.048 87	0.0878 5	0.089 10	0.01677	0.07461	0.04647	0.079 14	0.05 476	0.016 47	0.046 77	0.056 39	0.013 35	0.056 51	0.086 36
C7-e	0.007 81	0.1154 8	0.013 45	0.01247	0.07691	0.04677	0.064 77	0.04 973	0.016 79	0.046 77	0.038 33	0.007 51	0.087 64	0.088 94
C8-e	0.000 46	0.0487 9	0.046 76	0.07799	0.04645	0.01675	0.019 67	0.09 183	0.015 76	0.067 56	0.001 07	0.064 74	0.046 30	0.064 51
C9-e	0.009 55	0.0808 9	0.016 54	0.05167	0.03497	0.04796	0.016 73	0.01 640	0.046 42	0.053 97	0.030 68	0.070 57	0.062 76	0.039 66
C10-e	0.023 35	0.0078 0	0.032 55	0.03467	0.01546	0.04675	0.014 60	0.01 676	0.016 46	0.032 02	0.003 05	0.016 91	0.049 40	0.064 32

Tabla 3.31 Resultados de entrenamiento usando Inception_V3 con 27 UIDPs y 400 instancias por UIDP (Parte 2).

	C6-s	C7-s	C8-s	C1-e	C2-e	C3-e	C4-e	C5-e	C6-e	C7-e	C8-e	C9-e	C10-e
Chat	0.05851	0.0601	0.0534	0.0876	0.0012	0.0268	0.0567	0.0713	0.0596	0.0258	0.0493	0.0395	0.0760
		9	8	5	1	7	9	2	4	5	1	2	4
Comment	0.02832	0.0568	0.0174	0.0057	0.0930	0.0028	0.0178	0.0562	0.0619	0.0412	0.0078	0.0213	0.0953
		9	9	8	6	4	9	8	5	5	9	2	6
Gallery	0.02172	0.0120	0.0084	0.0460	0.0673	0.0099	0.0448	0.0203	0.0366	0.0687	0.0773	0.0397	0.0920
		5	8	3	4	7	8	6	5	5	4	7	3
Horizontal menú	0.00672	0.0075	0.0620	0.0885	0.0864	0.0277	0.0170	0.0024	0.0700	0.0306	0.0746	0.0313	0.0027
		5	5	0	7	1	0	6	3	7	8	2	6
Vertical menu	0.03450	0.0204	0.0117	0.0630	0.0648	0.0394	0.0730	0.0834	0.0034	0.0094	0.0773	0.0232	0.0758
		4	4	0	4	7	6	3	1	6	2	1	9
Music player	0.01160	0.0106	0.0072	0.0943	0.0426	0.0069	0.0038	0.0167	0.0156	0.0063	0.0226	0.0263	0.0471
		0	5	7	2	9	9	3	2	2	3	6	8
Panel	0.14658	0.0734	0.0305	0.0980	0.0106	0.0564	0.0262	0.0636	0.0276	0.0973	0.0304	0.0508	0.0388
		4	0	1	1	4	2	8	9	7	0	2	6
Picker	0.04841	0.0670	0.0216	0.0238	0.0144	0.0066	0.0192	0.0599	0.0990	0.0514	0.0647	0.0941	0.0460
		9	8	4	0	1	4	5	4	6	7	4	9
Ranking	0.03815	0.0190	0.0171	0.0734	0.0887	0.0065	0.0960	0.0657	0.0146	0.0271	0.0691	0.0776	0.0067
		3	0	4	0	8	8	1	2	3	3	8	9
C1-s	0.04332	0.0254	0.0215	0.0249	0.0163	0.0748	0.0129	0.0506	0.0674	0.0542	0.0638	0.0049	0.0304
		7	7	6	2	3	8	5	9	0	8	4	0
C2-s	0.08482	0.0813	0.0450	0.0876	0.0585	0.0678	0.0622	0.0060	0.0955	0.0605	0.0244	0.0283	0.0137
		6	9	5	4	7	2	1	6	2	5	5	6
C3-s	0.04841	0.0670	0.0216	0.0465	0.0823	0.0379	0.0263	0.0741	0.0487	0.0842	0.0418	0.0688	0.0154
		9	8	4	8	9	1	4	3	9	2	9	0
C4-s	0.03815	0.0190	0.0171	0.0315	0.0172	0.0494	0.0336	0.0838	0.0525	0.0168	0.0787	0.0302	0.0701
		3	0	0	6	0	4	4	0	9	1	2	9

	C6-s	C7-s	C8-s	C1-e	C2-e	C3-e	C4-e	C5-e	C6-e	C7-e	C8-e	C9-e	C10-e
C5-s	0.04332	0.0254	0.0215	0.0433	0.0465	0.0309	0.0080	0.0851	0.0450	0.0776	0.0025	0.0801	0.0571
		7	7	3	1	6	9	1	0	5	5	4	3
C6-s	0.00000	0.0813	0.0450	0.0958	0.0607	0.0734	0.0557	0.0132	0.0343	0.0759	0.1404	0.0317	0.0140
		6	9	2	9	4	6	2	7	9	0	0	7
C7-s	0.02172	1.0000	0.0084	0.0453	0.0714	0.0120	0.0301	0.0815	0.0701	0.0231	0.0579	0.0692	0.0772
		0	8	9	4	0	3	8	5	4	1	3	5
C8-s	0.00672	0.0075	0.0000	0.0839	0.0489	0.0601	0.0150	0.0906	0.0823	0.0583	0.0788	0.0824	0.0044
		5	0	9	1	6	9	1	6	9	8	3	8
C1-e	0.01743	0.0502	0.0878	1.0000	0.0275	0.0028	0.0147	0.0392	0.0744	0.0721	0.0386	0.0442	0.0959
		2	8	0	1	4	0	5	5	0	4	6	4
C2-e	0.00486	0.0589	0.0516	0.0764	1.0000	0.0914	0.0835	0.0355	0.0115	0.0313	0.0931	0.0667	0.0998
		3	8	6	0	1	6	7	8	1	3	8	1
C3-e	0.02661	0.0507	0.0394	0.0596	0.0570	1.0000	0.0931	0.0907	0.0586	0.0353	0.0976	0.0295	0.0263
		8	7	1	8	0	0	7	8	0	6	2	5
C4-e	0.43463	0.0364	0.1143	0.0923	0.0227	0.0477	1.0000	0.0180	0.0092	0.0053	0.0461	0.0469	0.0920
		1	9	6	8	5	0	8	4	7	2	1	1
C5-e	0.06957	0.0522	0.0049	0.0821	0.0681	0.0235	0.0458	1.0000	0.0808	0.0709	0.0912	0.2673	0.0504
		1	1	9	0	2	0	0	4	0	9	1	6
C6-e	0.09025	0.0613	0.0638	0.0938	0.0319	0.0079	0.0442	0.0900	0.0000	0.0635	0.0109	0.0251	0.0807
		8	8	5	2	9	2	7	0	0	6	9	4
C7-e	0.04777	0.0650	0.0428	0.0450	0.0381	0.0988	0.0089	0.0666	0.0824	1.0000	0.0856	0.0739	0.0815
		1	8	2	0	4	8	9	1	0	0	6	0
C8-e	0.01230	0.0255	0.0652	0.0741	0.0946	0.0640	0.0859	0.0619	0.0958	0.0938	0.0000	0.0378	0.0791
		2	8	7	9	2	4	5	3	3	0	7	4
C9-e	0.02272	0.2166	0.0078	0.0631	0.0055	0.0604	0.0096	0.0026	0.0808	0.1442	0.0089	1.0000	0.0074
		2	8	1	4	6	4	5	9	1	1	0	7
C10-e	0.06531	0.0215	0.0399	0.0243	0.0878	0.0943	0.0294	0.0063	0.0165	0.0141	0.0894	0.0685	0.0000
		7	1	2	6	0	3	8	8	9	7	4	0

Como conclusión, se observa que el resultado obtenido con el modelo de Inception_V3 con respecto al modelo de AlexNet para la clasificación de imágenes es 30.7% mejor. Por lo tanto, Inception_v3 resulta un excelente modelo para clasificar e identificar elementos en imágenes. Es importante mencionar, que a diferencia del modelo generado con Matlab estos nuevos modelos permiten agregar un mayor abanico de imágenes y mejorar el tiempo de procesamiento e identificación de elementos en imágenes generadas a mano alzada. Con el modelo de Matlab el tiempo de identificación se extendía hasta los 3 minutos, con el modelo de AlexNet el tiempo de respuesta es de 7 a 9 segundos, mientras que con Inceptio_V3 el tiempo se reduce de 7 hasta 4 segundos. Además, el margen de error es significativamente menor con AlexNet e Inception_V3 a pesar de contar con menor porcentaje de eficiencia en el entrenamiento. Por lo tanto, el modelo generado con Inception_V3 es el adecuado para la identificación de elementos en imágenes generadas a mano alzada con lienzos de diferentes texturas y color en los bordes de cada UIDP dibujado.

3.3.4 Modulo generador de XML

Dentro del proceso de desarrollo de aplicaciones móviles multi-dominio y multi-plataforma se desarrolló un módulo generador del documento de configuración basado en XML, este módulo se desarrolló en PHP. EL módulo funciona desde dos perspectivas: a partir de un archivo .txt o bien desde los datos de configuración obtenidos a través de una interfaz gráfica. El funcionamiento a partir del archivo .txt generado por el módulo de procesamiento de imágenes, consiste en a través de un servicio Web leer el archivo, del archivo txt se extraen los patrones de diseño y se genera un objeto JSON (*JavaScript Object Notation*). En el objeto JSON, a cada patrón de diseño se le asigna una clave que facilita la manipulación de la información. Por ejemplo, para el patrón *Video* la clave es s1, para el patrón *Login* es s2, y así sucesivamente con el resto de patrones simples. En el caso de las composiciones el formato de la clave es muy similar, por ejemplo, para la composición Carrusel de videos la clave es c1. Una vez que se tienen los patrones identificados, a través de la interfaz gráfica se obtiene el resto de datos de configuración necesarios para construir el archivo basado en XML. En ambos

enfoques, el módulo generador del documento basado en XML usa un archivo JSON que permite validar la configuración de la aplicación para generar el documento basado en XML. Las principales secciones del archivo JSON se presentan a continuación, en la figura 3.24 (a) se muestra la estructura para validar por patrón de diseño las posiciones activas dentro de una interfaz de usuario son: *body*, *right*, *left*, *top* y *foot*, para lo que se consideró el análisis de la Tabla 3.5.

La figura 3.24 (b) se muestra la sección del JSON que permite validar las posiciones activas de cada plantilla de la figura 3.19, mientras que la figura 3.24 (c) muestra el segmento del JSON para validar de acuerdo al tipo de aplicación los patrones de diseño válidos y para cada patrón el o los patrones relacionados de navegación. Por ejemplo, en la figura se observa la validación para una aplicación tipo MOOC del dominio educativo, los patrones de diseño usados son: *s6 (SplashScreen)*, *s2 (Login)*, *s3 (Dashboard)*, *s4 (List)*, *s26 (Video)*, y *s16 (Search)*. En la etiqueta *relation* se incluyen las dependencias de los patrones de diseño respecto a otros patrones, después del patrón *s4 (List)* el patrón siguiente de navegación es *s26 (Video)*, esto se incluye en la etiqueta *nextUIDP*. Finalmente, la etiqueta *containUIDP* sirve para indicar los patrones de diseño contenidos dentro del patrón de diseño o con los que aparece en una misma página de la interfaz de usuario, por ejemplo, el patrón *s4 (List)* aparece en una misma página de la interfaz de usuario con el patrón *s16 (Search)*.

Una vez que se valida la configuración, a través de otro servicio se crea la estructura del documento basado en XML con los datos de configuración previamente seleccionados, el documento basado en XML se valida a través de un documento XML Schema para finalizar el proceso de generación del XML.

```

"UIDPs": [
  {
    "id": "s2",
    "nombre": "Login",
    "name": "Login",
    "position": {
      "head": true,
      "left": false,
      "body": true,
      "right": false,
      "foot": false
    }
  }, {
    "id": "s3",
    "nombre": "Dashboard",
    "name": "Dashboard",
    "position": {
      "head": false,
      "left": false,
      "body": true,
      "right": false,
      "foot": false
    }
  }
],
"templates": [
  {
    "id": "t1",
    "position": {
      "head": false,
      "left": false,
      "body": true,
      "right": false,
      "foot": false
    }
  }, {
    "id": "t2",
    "position": {
      "head": true,
      "left": false,
      "body": true,
      "right": false,
      "foot": false
    }
  }
],
"apps": {
  "a6": {
    "nombre": "MOOC",
    "name": "mooc",
    "relation": [
      {
        "uidp": "s6",
        "nextUIDP": ["s2", "s3"],
        "containUIDP": []
      }, {
        "uidp": "s2",
        "nextUIDP": ["s3", "s4"],
        "containUIDP": []
      }, {
        "uidp": "s3",
        "nextUIDP": ["s4"],
        "containUIDP": []
      }, {
        "uidp": "s4",
        "nextUIDP": ["s26"],
        "containUIDP": ["s16"]
      }
    ],
    "patterns": ["s6", "s2", "s3", "s4", "s26", "s16"]
  }
}

```

a)

b)

c)

Figura 3.24 Estructura del archivo de validación JSON: a) Posiciones válidas de cada UIDP, b) Posiciones activas por plantilla y c) Patrones de diseño permitidos por tipo de aplicación.

3.3.5 Generador de código

El último módulo de software tiene como objetivo generar de manera automática el código fuente de una aplicación multi-dominio, multi-plataforma y multi-dispositivo. Este módulo también se desarrolló con PHP. Para lograr su objetivo, este módulo usa el documento de configuración XML creado por el generador de XML descrito en la sección 3.1.4 de este documento y transforma las etiquetas de cada patrón de diseño en código fuente de una aplicación, para lo cual usa un servicio Web. El servicio Web permite leer el documento basado en XML y obtener las estructuras de código necesarias para generar la aplicación de acuerdo con la configuración del documento basado en XML. Finalmente, se genera un archivo ZIP que empaqueta el o los proyectos generados, este archivo ZIP es puesto a disposición del usuario para su descarga.

El módulo generador de código, crea una carpeta principal que contiene una carpeta con el nombre del proyecto, dentro de esta carpeta se encuentran por dispositivo las carpetas de cada proyecto generado para cada plataforma de desarrollo:

Desktop solamente tiene disponible Web; *Mobile* soporta los proyectos de Android™, Firefox® OS, iOS®, y Web; TV tiene soporte para los proyectos de Android™, Firefox® OS, Web y WebOS. La estructura de carpetas se observa en la figura 3.25.

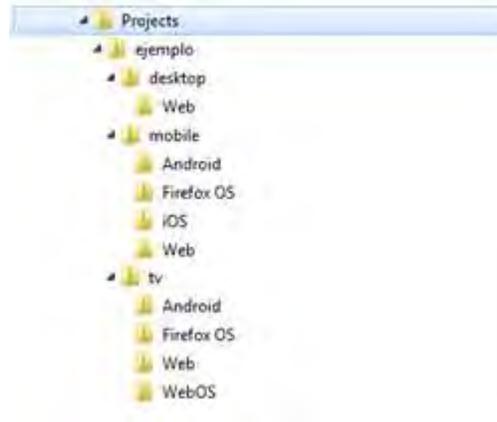


Figura 3.25 Estructura de carpetas generada.

Cada plataforma utiliza un conjunto de carpetas que permite la ejecución y compilación del proyecto en su entorno de desarrollo. Los entornos de desarrollo que contemplados son: 1) para las aplicaciones de Android™ se utilizó Android™ Studio; y 2) para las aplicaciones en iOS el entorno es Xcode®. En el caso de las aplicaciones educativas basadas en un entorno de desarrollo Web se contempla una organización de acuerdo a las bibliotecas implementadas y los recursos multimedia que utiliza.

En la figura 3.26 se presenta la estructura de carpetas de los proyectos en Android™ para Android™ Studio 2.3.

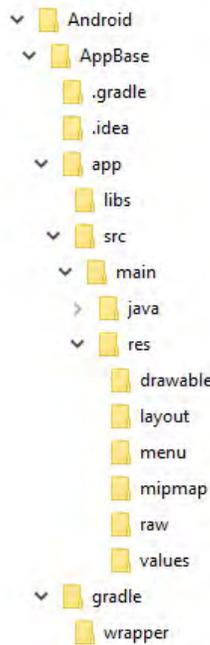


Figura 3.26 Estructura de carpetas para los proyectos de las aplicaciones de Android™

Las aplicaciones desarrolladas en iOS® se encuentra en la carpeta de Mobile, en la figura 3.27 se presenta su estructura de carpetas.

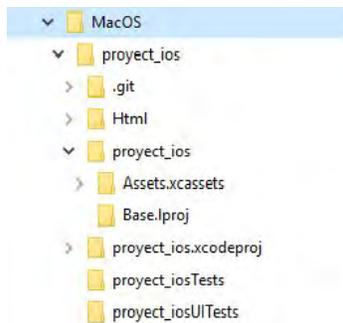


Figura 3.27 Estructura de carpetas para los proyectos de las aplicaciones de iOS®

3.3.6 Herramienta generadora de aplicaciones multi-dominio

En esta sección se presenta *MultiGene*, una herramienta Web para la generación de aplicaciones multi-dominio, de manera específica para los dominios educativo, e-commerce y social media. *MultiGene* fue desarrollada en PHP, HTML5 y Frameworks como *AngularJS* y *Bootstrap*, esta herramienta permite a través de un conjunto de servicios generar aplicaciones móviles. Cabe mencionar que *MultiGene* es una herramienta que admite la generación de aplicaciones multi-dispositivo y

multi-plataforma. *MultiGene*, ofrece dos formas de generar aplicaciones, la primera permite la selección de un dominio de aplicación; el segundo es a través de una imagen que representa una interfaz de usuario donde se aplican técnicas de inteligencia artificial.

Las principales interfaces de la herramienta a través de las cuales se realiza la generación de aplicaciones desde los dos métodos proporcionados se presentan a continuación. La primera interfaz de *MultiGene* es un *Login* tal como se muestra en la figura 3.28 (a), una vez que el usuario ingresa sus datos de acceso se muestran las dos opciones de generación de aplicaciones tal como se observa en la figura 3.28 (b).

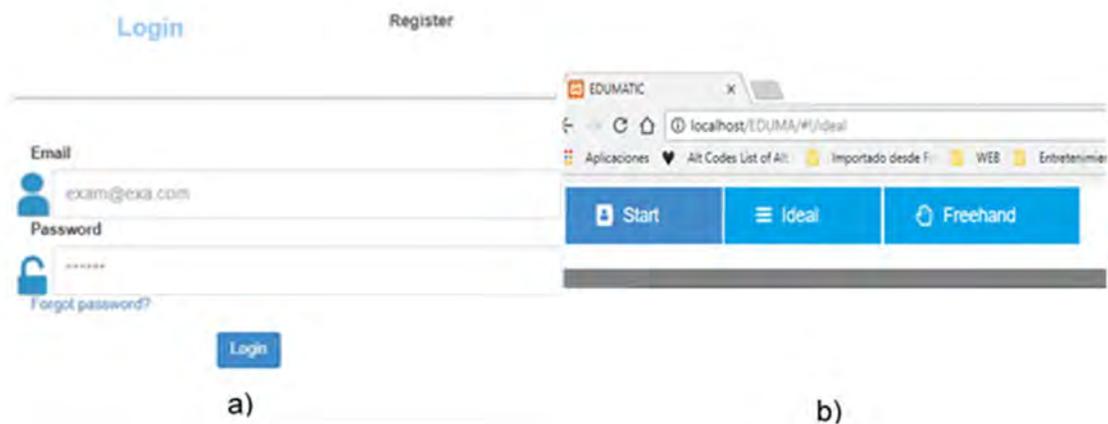


Figura 3.28 Pantallas de *MultiGene*: a) *Login* y b) Menú principal.

Para los procesos de generación de aplicaciones: a mano alzada e ideal los pasos a seguir son similares, sin embargo, el orden de secuencia cambia dependiendo del tipo de proceso. Desde esta perspectiva a continuación se presentan las dos interfaces de acuerdo al proceso de generación de aplicaciones: ideal y mano alzada. La figura 3.29 es la interfaz para iniciar la generación de una aplicación usando el proceso ideal, el cual parte de la selección del dominio: educativo, e-commerce y social media de la aplicación que desea generar.

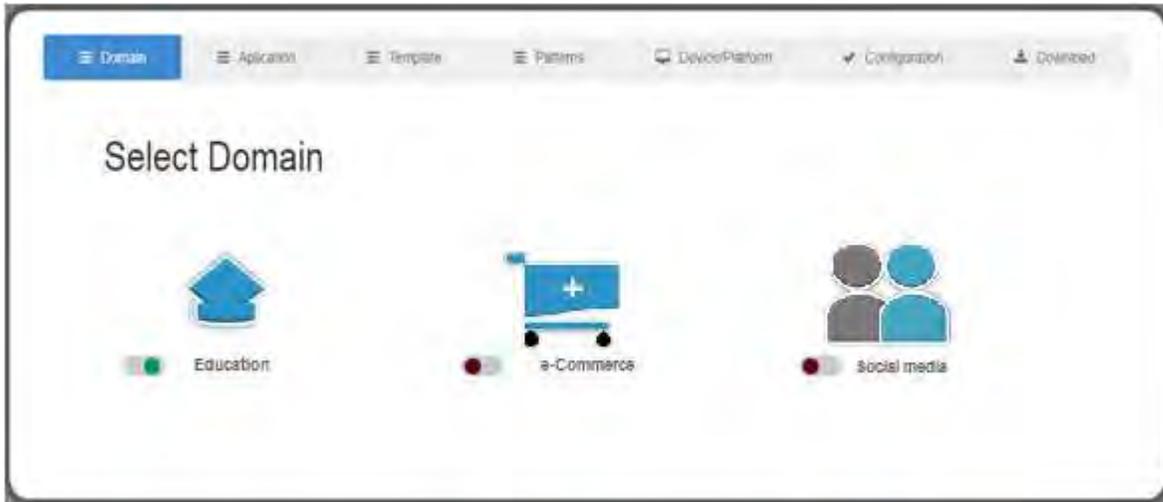


Figura 3.29 Interfaz del inicio para el proceso ideal.

Por otra parte, la primera interfaz del proceso a mano alzada es la que se observa en la figura 3.30, donde a diferencia del proceso ideal, se inicia con la carga de una imagen que representa una interfaz de usuario generada a mano alzada por el usuario desde una herramienta de dibujo respetando las características descritas en la etapa de diseño.

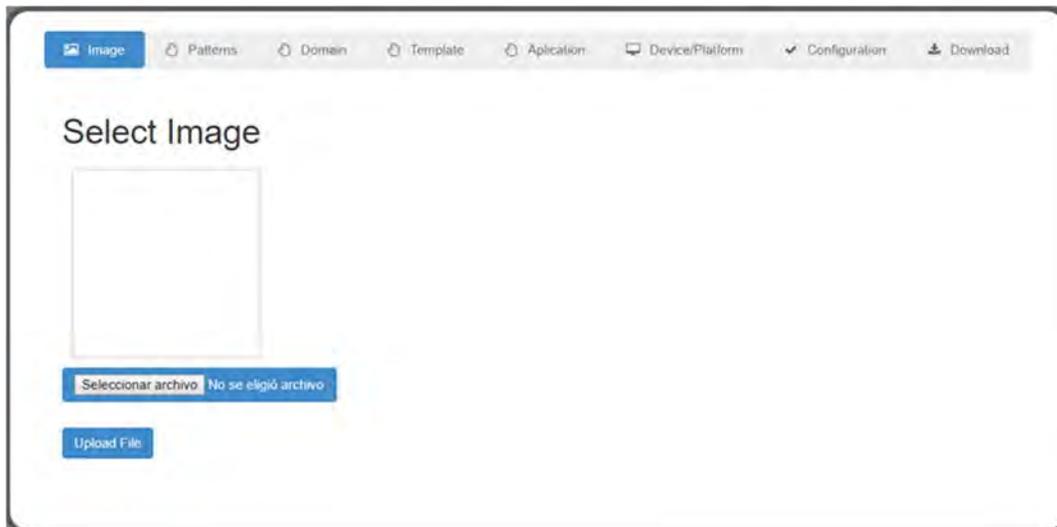


Figura 3.30 Pantalla de inicio del proceso a mano alzada.

Capítulo 4 Resultados

En esta sección se presentan cuatro casos de estudio como prueba de concepto de los resultados obtenidos hasta el momento durante este trabajo de tesis.

El primer caso de estudio ilustra la generación de una aplicación educativa de tipo Wiki a partir del proceso a mano alzada. El segundo caso de estudio ilustra el proceso de generación de aplicaciones a través del proceso ideal para generar una aplicación educativa tipo MOOC. El tercer caso de estudio se centra en la generación de una aplicación e-commerce de tipo B2C. Finalmente, se describe un cuarto caso de estudio sobre una aplicación social media tipo social network.

4.1 Caso de estudio 1: Generación de una aplicación educativa tipo Wiki a mano alzada

El objetivo del caso de estudio es presentar el proceso de generación de aplicaciones educativas tipo wiki a través de *MultiGene*. La aplicación se genera para las plataformas Android™ y iOS®. La aplicación tiene como objetivo reforzar el conocimiento en el área de la informática. Los requerimientos funcionales se describen de la siguiente manera:

- Mostrar el logo de la aplicación antes de ejecutar la aplicación.
- Mostrar temas relevantes a través de un carrusel de imágenes.
- Permitir al usuario realizar búsquedas sobre conceptos o temas de interés.
- Mostrar una lista de resultados de búsqueda.
- Reproducir video clips.
- Notificar al usuario de posibles errores al capturar datos o si la aplicación deja de funcionar.

Para generar la aplicación de informática, se consideraron las características mencionadas y el conjunto de UIDPs de la Tabla 3.5, donde se presentan los UIDPs válidos para una Wiki. Tomando en cuenta las características de la aplicación, los patrones de diseño para generar la Wiki educativa son los siguientes:

- *Splashscreen*: presenta el logotipo de la aplicación.

- *Carousel*: muestra recursos educativos relevantes
- *Search*: buscar recursos educativos
- *List*: muestra los resultados de una búsqueda
- *Video*: reproduce video clips

A continuación, se describe paso a paso las acciones necesarias para generar la aplicación Wiki sobre informática con las características descritas.

Es necesario autenticarse a través de *MultiGene* como se muestra en la figura 3.28 (a), una vez que el usuario se autentica es posible seleccionar desde el menú principal la opción a través de la cual se realizará la generación de la aplicación. En este caso, la opción será a través del proceso a mano alzada. Por lo anterior, la interfaz generada para crear las aplicaciones en este caso de estudio se presenta en la figura 4.1, donde se observan cuatro de los patrones de diseño necesarios para crear la wiki, esto con el objetivo de mostrar la mayor funcionalidad de *MultiGene*.

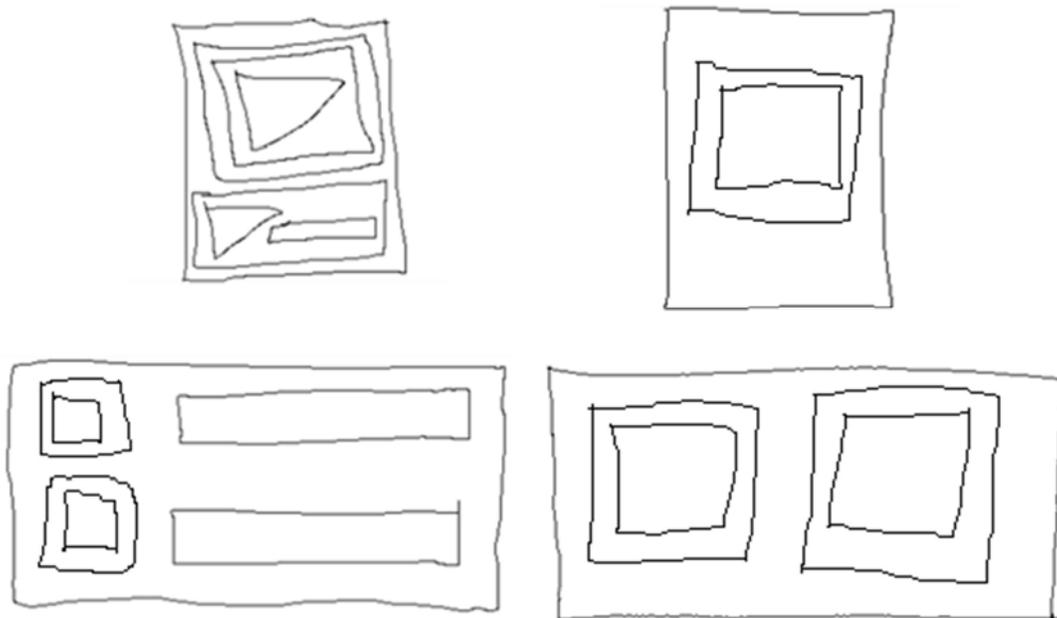


Figura 4.1 Interfaz generada a mano alzada para crear una Wiki.

Una vez creada la interfaz, la figura 4.2 (a) ilustra la carga de la interfaz a través de *MultiGene*. Una vez que se carga la interfaz, *MultiGene* valida el formato de la imagen e inicia el proceso para identificar el conjunto de UIDPs en la imagen. A

continuación, la figura 4.2 (a) muestra cuando la interfaz se envía a *MultiGene* y en la figura 4.2 (b) muestra la lista de UIDPs identificados y la opción de cambiar o agregar más UIDPs, en este caso se agregó uno más.

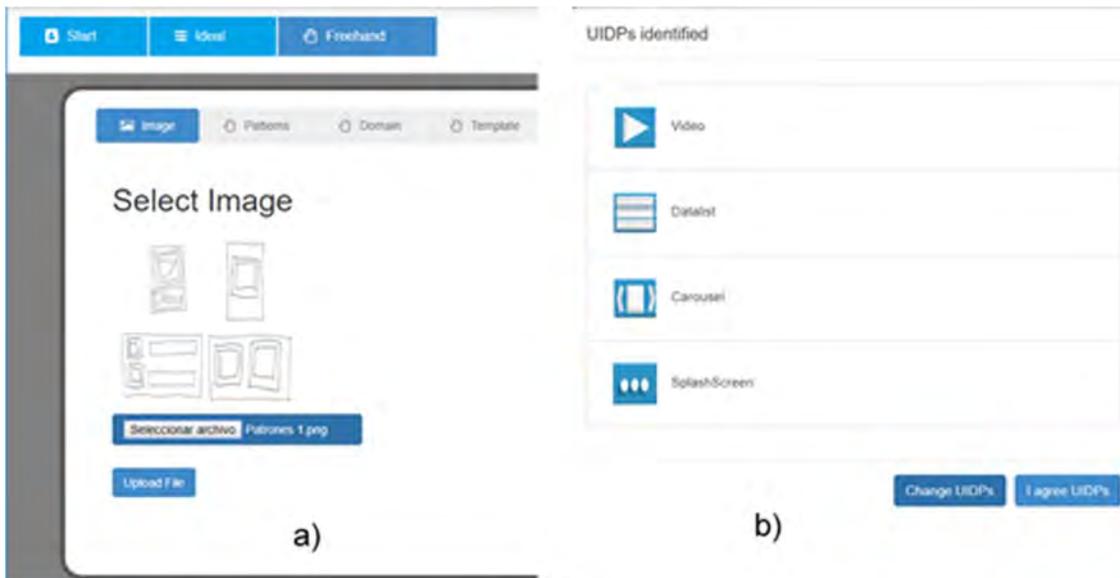


Figura 4.2 Pantalla 1: a) Envío de la interfaz, b) UIDPs identificados.

Ahora, el siguiente paso es seleccionar el patrón de diseño *Search* para agregarlo antes de continuar con el proceso de generación. En ese sentido la figura 4.3, muestra la lista de patrones identificados además del conjunto de patrones adicionales para que el usuario cambie los patrones con los cuales quiere generar la aplicación.

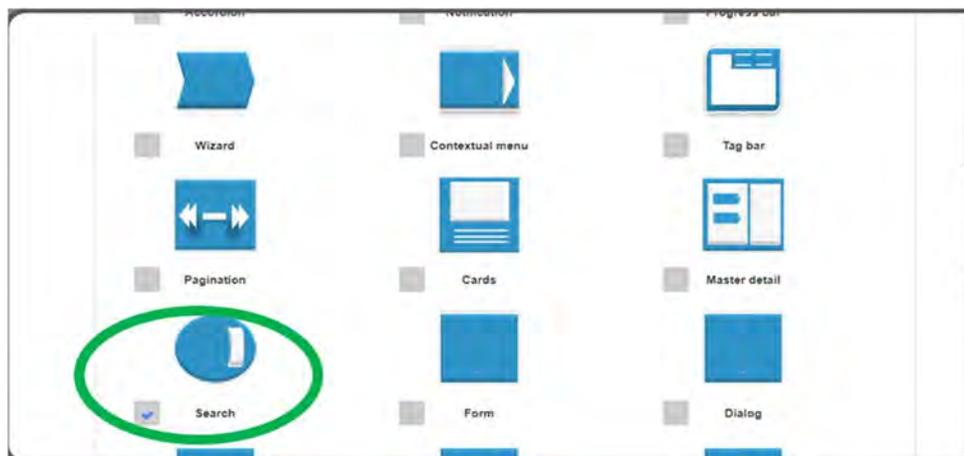


Figura 4.3 Pantalla 2: Selección de nuevos UIDPs.

En el siguiente paso es necesario seleccionar el dominio para el que se generará la aplicación. Como se mencionó en secciones anteriores, *MultiGene* apoya la generación de aplicaciones para los dominios educativo, médico y entretenimiento. En este caso, la aplicación pertenece al dominio educativo, por lo que se seleccionó tal como se muestra en la figura 4.4.



Figura 4.4 Pantalla 3: Selección del dominio de la aplicación.

Después de seleccionar el dominio, es necesario seleccionar una plantilla que permita organizar los UIDPs en la interfaz de usuario. La selección de la plantilla depende de los UIDPs debido a que cada patrón tiene posiciones válidas dentro de la interfaz, tal como se señaló en la tabla 3.5. Partiendo de lo anterior, para esta aplicación la plantilla seleccionada se basó en los criterios de la tabla 3.5, y tal como se muestra en la figura 4.5.

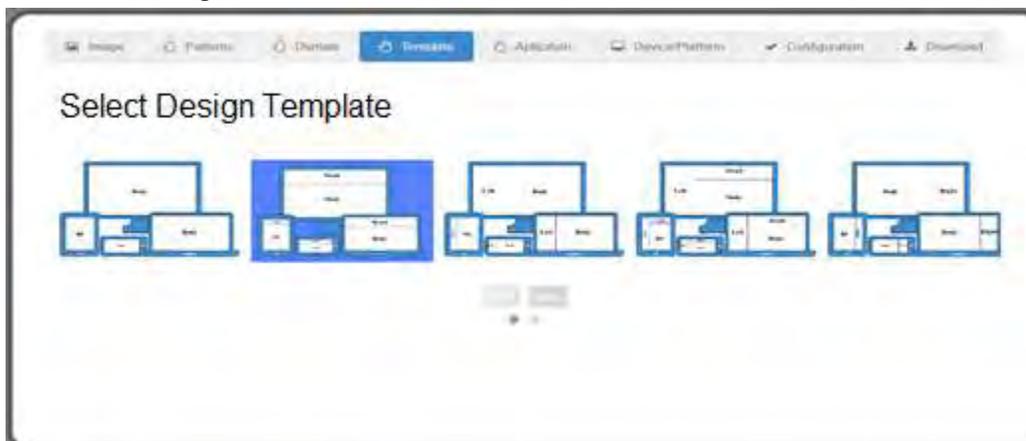


Figura 4.5 Pantalla 4: Selección de plantilla.

Una vez que se seleccionó la plantilla, se selecciona el tipo de aplicación educativa a generar: Blog, Wiki o MOOC. En este paso, *MultiGene* valida los patrones y la

plantilla seleccionada para permitir la selección del tipo de aplicación. De acuerdo con el caso de estudio descrito, el tipo de aplicación educativa a generar es una Wiki tal como se muestra en la Figura 4.6.

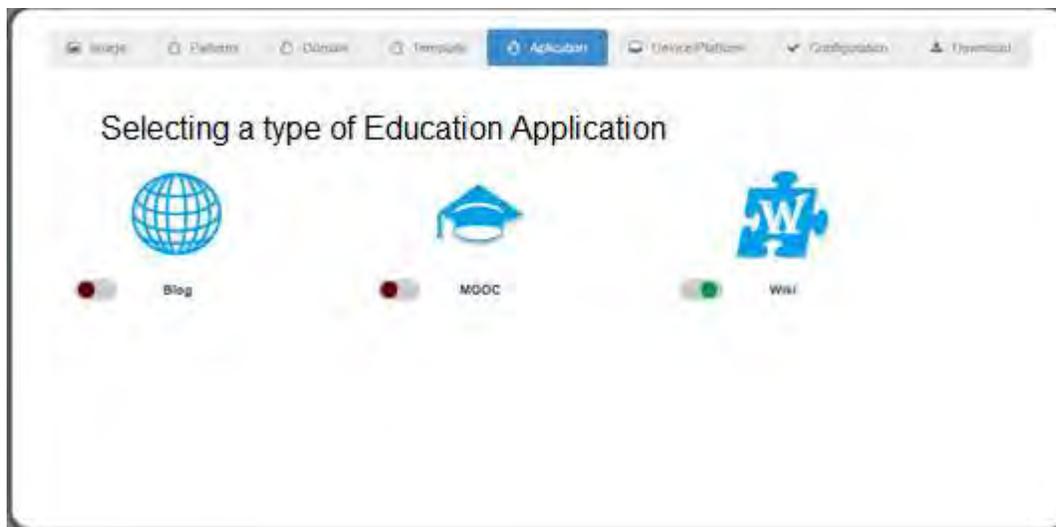


Figura 4.6 Pantalla 5: Selección del tipo de aplicación educativa.

Ahora, viene la selección de las plataformas de desarrollo de la aplicación y los dispositivos móviles en los que se implementará la aplicación. Como se menciona en la descripción inicial del caso de estudio, las plataformas de desarrollo son Android™ y iOS®, los dispositivos móviles son tabletas y teléfonos inteligentes. En la Figura 4.7 se presenta la selección de la configuración de dichos dispositivos seleccionada para este caso de estudio.

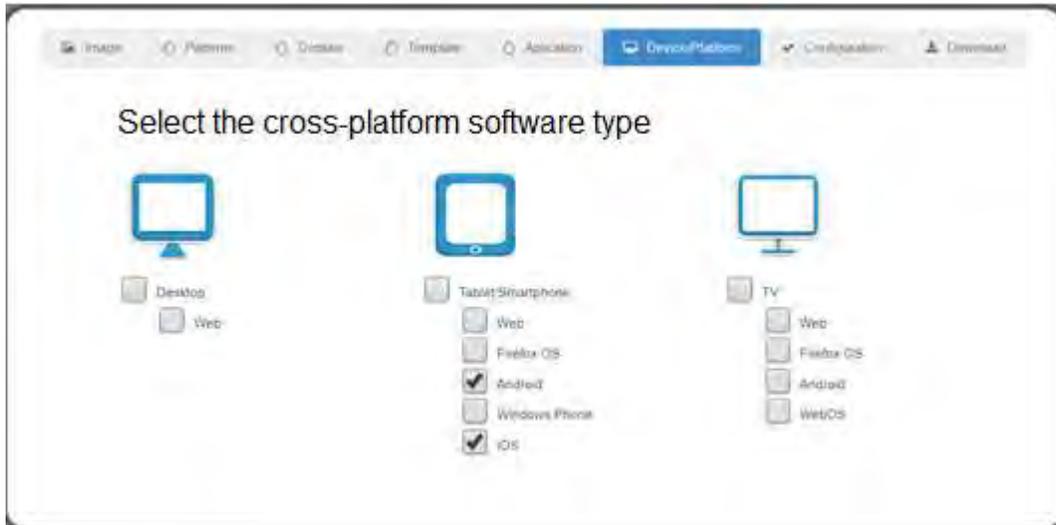


Figura 4.7 Pantalla 6: Selección de dispositivos y plataformas de desarrollo.

Antes de generar la aplicación, es necesario agregar los datos de despliegue de la aplicación, como nombre, nombre corto, versión. Además de datos sobre el autor de la aplicación, como su nombre, empresa, correo electrónico y sitio web. En este caso, el nombre de la aplicación es *Wiki-comp* y el nombre corto es *Comp* como se muestra en la Figura 4.8.

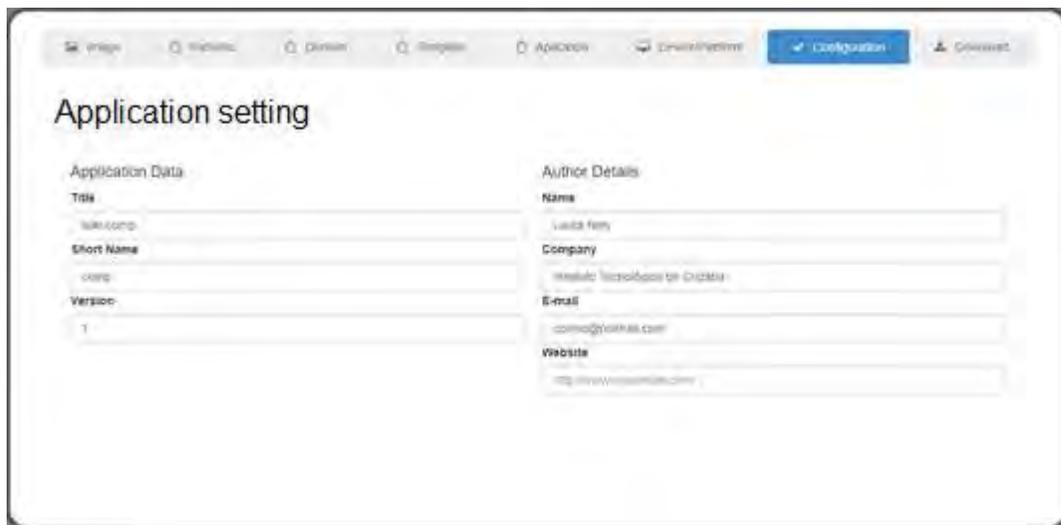


Figura 4.8 Pantalla 7: Configuración.

Antes de construir la aplicación, *MultiGene* muestra la configuración seleccionada para que el usuario verifique los datos y confirme la generación de la aplicación, tal como se muestra en la figura 4.9 (a). Una vez que se confirman los datos, se construyen los archivos necesarios para crear la aplicación de manera interna (figura 4.9-b). La aplicación se construye de acuerdo a la configuración seleccionada a lo largo de los pasos anteriores. Al finalizar, *MultiGene* empaqueta el código fuente de la aplicación en un archivo ZIP y proporciona el enlace de descarga tal como se muestra en la figura 4.9 (c).



Figura 4.9 Pantalla 8: a) Confirmación de la configuración y b) Generación del código fuente de la aplicación y c) descarga de la aplicación.

Una vez que el código se descarga, a través de los entornos de desarrollo Android™ Studio y Xcode® las aplicaciones generadas se despliegan a través de sus emuladores. Cabe mencionar que las aplicaciones generadas están listas para que el usuario agregue el contenido que mejor le convenga. Partiendo de esto, se agregaron los recursos multimedia necesarios para dar el enfoque de una Wiki para reforzar temas de computación.

Una vez que la aplicación se actualizó con los recursos multimedia, desde el emulador de Android™ Studio se despliega la aplicación generada, y tal como se observa en la figura 4.10 Wiki-comp usa los patrones de diseño definidos para el caso de estudio: *Splashscreen, Search, List, Carousel* y *Video*.

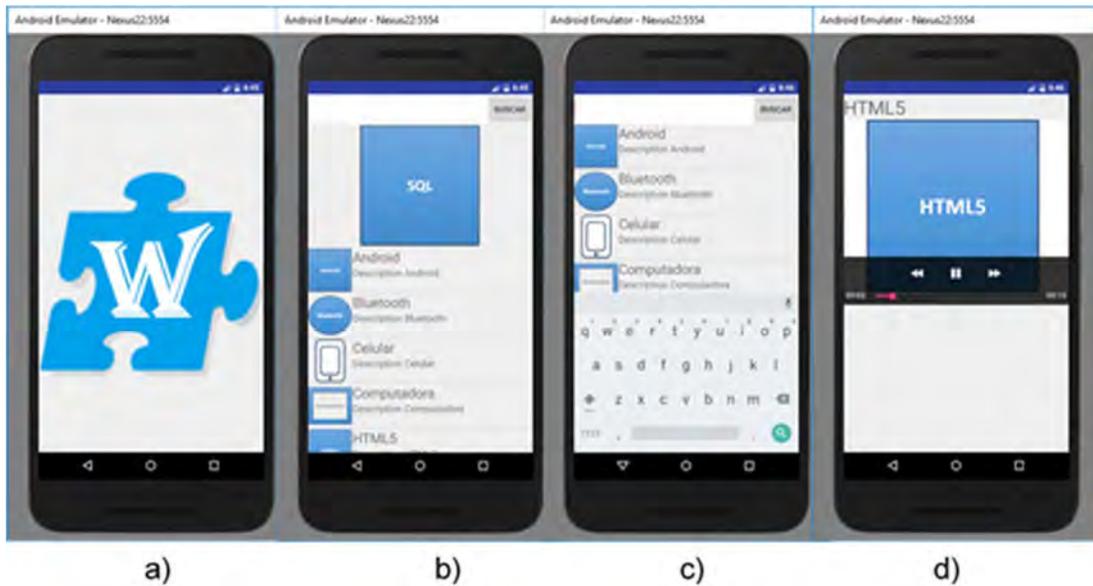


Figura 4.10 Despliegue de la aplicación en Android™ Studio: a) Splashscreen, b) Carousel y List, c) List y Search, d) Video.

El despliegue de la aplicación desde Xcode® se observa en la figura 4.11 y de manera similar al despliegue de la aplicación desde Android™ Studio, en este caso también se observan los patrones de diseño seleccionados para el caso de estudio, los cuales mantienen el diseño propio de la plataforma de desarrollo.



Figura 4.11 Despliegue de la aplicación en XCode: a) Splashscreen, b) Carousel y List, c) List y Search, d) Video.

Como se observa, *MultiGene* genera aplicaciones con interfaces limpias e intuitivas, sin embargo, para validar los aspectos anteriores, es necesario evaluar a través de la participación de expertos y usuarios.

4.1.1 Evaluación

En la literatura, se informan varios métodos para evaluar software con el fin de validar los métodos / herramientas propuestos, para realizar la evaluación hay tres enfoques: evaluaciones cualitativas que son subjetivas; evaluaciones cuantitativas que son objetivas; y evaluaciones híbridas, que es cualitativa-cuantitativa. En este sentido, (Kitchenham, 1996) señala que las evaluaciones cuantitativas identifican en términos mensurables los efectos del uso de métodos / herramientas de Ingeniería de Software. Por otro lado, las evaluaciones cualitativas consisten en establecer la idoneidad de los métodos / herramientas en términos de las características proporcionadas por los mismos métodos / herramientas. Dicho esto, existen estándares como (ISO/IEC 9126, 2001) y (ISO/IEC 14598, 1998) para evaluar la calidad del producto de software. Por ejemplo, (Colombo-Mendoza et al., 2014) presentó *MobiCloUP!*, una plataforma como servicio para aplicaciones móviles basadas en servicios en la nube. En este artículo, se realizó una evaluación cualitativa y una evaluación cuantitativa, en términos cualitativos, se propusieron diversos criterios basados en el tipo de aplicación a evaluar y en términos cuantitativos se basó en el modelo de calidad ISO / IEC9126.

Por otro lado, evaluar una aplicación educativa móvil requiere considerar, además de aspectos como la calidad del software, también las características educativas. La literatura reporta varios métodos para evaluar desde el enfoque cualitativo una aplicación educativa móvil, desde este enfoque según (Baran et al., 2016) se deben considerar aspectos pedagógicos, usabilidad técnica, contenido, conectividad y contextualidad. De manera similar (Jareño & Martínez, 2016) planteó una rúbrica para evaluar las aplicaciones educativas, la rúbrica se divide en tres aspectos principales: criterios educativos, criterios técnicos y criterios económicos. Esta rúbrica fue validada por juicios de expertos, para obtener requisitos mínimos de fiabilidad y validez. Mientras tanto, (Soad et al., 2016) propuso un método para

evaluar la calidad de las aplicaciones de aprendizaje móvil. El método de calidad propuesto comprende: (i) un modelo de calidad, que establece un conjunto bien definido de criterios y requisitos; (ii) métricas, es decir, los requisitos de calidad que se asignan en atributos que se miden. Para llevar a cabo las mediciones, desarrollaron una lista de verificación, que comprende 80 preguntas, clasificadas en ocho grupos diferentes responsables del establecimiento de los criterios de calidad; y (iii) criterio de criterios, es decir, para cada lista de verificación definida, se asigna una puntuación que permite que la pregunta reciba un valor que caracterice su nivel de calidad. Otra propuesta es presentada por (Chen, 2016), donde los autores realizaron una investigación y evaluación sobre las posibilidades de las aplicaciones móviles de aprendizaje del idioma inglés para estudiantes adultos. La evaluación consta de tres pasos: diseño de una rúbrica basada en la teoría, selección de aplicaciones y evaluación de aplicaciones.

En otro contexto, de acuerdo con (Isa et al., 2016) para evaluar una aplicación educativa se deben considerar los factores que impulsa a los estudiantes a usar el aprendizaje móvil. En este sentido, los autores señalan siete factores: (i) facilidad de uso, (ii) uso de nueva tecnología, (iii) movilidad, (iv) estímulo de la universidad, (v) mejorar la calidad de vida, (vi) costos y tiempo y (vii) fácil acceso. Desde una perspectiva similar, (Hamidi & Chavoshi, 2017) enfatizaron siete factores relacionados con la adopción del aprendizaje móvil en la educación superior, las categorías principales son: facilidad de uso, confianza, características personales y cualidades, contexto, utilidad percibida de uso, intención de comportamiento y cultura de usar un modelo de investigación.

Finalmente, existen herramientas para evaluar la calidad del e-learning, y en este contexto (Betrián et al., 2016) señala que la calidad del e-learning abarca varios aspectos, como la usabilidad, la relevancia educativa, la interactividad, la profundidad del contenido, la reutilización y la modularidad o accesibilidad. Del mismo modo, (Shoukry et al., 2015) presentó *Pre-MEGa*, un marco de trabajo que tiene como objetivo facilitar el proceso de traducir la investigación en características concretas y mensurables para diseñar y evaluar software educativo móvil. *Pre-*

MEGa se basa en un conjunto detallado de atributos que constituyen un juego de aprendizaje divertido, útil, beneficioso y, sobre todo, exitoso, dirigido a preescolares. Como se observa en la literatura, se proponen varias métricas en el tema *e-learning* y *m-learning* para evaluar las aplicaciones educativas móviles desde el enfoque cualitativo, algunas métricas se usan con diferentes nombres, pero en el mismo sentido.

4.1.1.1 Diseño de la evaluación

Para evaluar las aplicaciones generadas, se seleccionaron tres aplicaciones educativas móviles tipo Wiki existentes. Las aplicaciones se seleccionaron de la Google Play Store, se consideró la calificación que los usuarios proporcionaron sobre la aplicación y la cantidad de usuarios. Las aplicaciones seleccionadas son *Khan Academy*, *Wiki enciclopedia* y *Kiwix Wikipedia*. (*AcademyKhan*, 2018) es una aplicación móvil que ofrece acceso a un conjunto de ejercicios de práctica, videos y un panel de aprendizaje personalizado que permite a los estudiantes aprender a su propio ritmo, dentro y fuera del aula. La aplicación está disponible para las plataformas macOS® y Android™. (*Kiwix*, 2018) es una aplicación que permite el acceso al contenido educativo incluso sin conexión a Internet gracias a la capacidad de descargar contenido para ver más tarde. También, funciona en una amplia variedad de sistemas operativos, desde Android™, iOS® hasta Microsoft Windows, macOS® y distribuciones GNU / Linux. Por otro lado, (*Enciclopedia*, 2018) es una aplicación que permite a los usuarios buscar diversos artículos y leerlos en más de 30 idiomas. Además de seleccionar elementos aleatorios que se presentarán al usuario como recomendación de lectura.

De acuerdo con la literatura revisada presentada en las secciones anteriores de este documento, y con el apoyo de un grupo de expertos, así como la participación de un grupo de estudiantes, se diseñó el instrumento de evaluación y luego se realizó la evaluación. El panel se integró por 10 expertos: tres expertos en desarrollo de software, dos diseñadores de interfaz de usuario y cinco expertos en educación.

Por otro lado, se integró un grupo de 60 estudiantes, los estudiantes interactuaron con las diferentes aplicaciones, y cada estudiante revisó las principales

funcionalidades de las aplicaciones. Luego, los estudiantes recibieron a través de un correo electrónico el conjunto de preguntas para evaluar las aplicaciones revisadas. Los estudiantes tenían entre 18 y 22 años, todos con experiencia en el uso de dispositivos móviles. El 50% de los estudiantes eran mujeres y el otro 50% hombres. Los participantes son estudiantes del Instituto Tecnológico de Orizaba de las siguientes carreras: Sistemas Computacionales, Ingeniería Industrial, Ingeniería Química, Ingeniería Eléctrica, Ingeniería Mecánica e Ingeniería en Gestión Empresarial.

La evaluación se lleva a cabo en el siguiente orden a través de un conjunto de instrucciones enviadas a los participantes en un correo electrónico que se describen a continuación: (1) Se solicitó a los estudiantes participantes que instalaran las cuatro aplicaciones para ser evaluadas en un dispositivo móvil (Academy Khan, Wiki enciclopedia, Kiwix Wikipedia y Wiki-Comp). Los estudiantes no conocían a ninguno de los otros participantes, por lo que se garantiza que los resultados obtenidos no se influenciaron por la experiencia del resto de los participantes. Una vez que se instalaron las aplicaciones, (2) cada usuario exploró las funciones principales de cada aplicación, como (a) búsqueda de contenido, (b) guardar contenido, (c) explorar resultados de búsqueda, (d) perfil de usuario y (e) mecanismos de ayuda. (3) Al final de la revisión de cada aplicación, el usuario respondió un breve cuestionario de acuerdo con su experiencia luego de interactuar con las aplicaciones. El cuestionario incluye 20 preguntas para abordar todos los aspectos cualitativos de la evaluación. Los aspectos cualitativos de la evaluación son: Funcionalidad, Pedagogía, Rendimiento, Soporte y usabilidad.

Estos aspectos son propuestos por la literatura para validar el aprendizaje y la enseñanza a través de las aplicaciones educativas generadas por *MultiGene*. Es importante mencionar que los evaluadores no sabían que una de las aplicaciones fue desarrollada por nosotros. Los aspectos cualitativos de la evaluación se describen a continuación.

Necesidad 1 (N1): Funcionalidad. De acuerdo con (Soad et al., 2016), la funcionalidad se refiere a la disponibilidad de funciones para satisfacer adecuadamente las necesidades explícitas e implícitas de la aplicación. El conjunto

de características recomendadas para evaluar esta necesidad se describe a continuación.

Característica 1 (F1): Eficacia de la ejecución de la tarea. Se refiere a la posibilidad de facilitar la ejecución de tareas, minimizando el tiempo, el costo de conexión y la cantidad de datos transmitidos.

Característica 2 (F2): Pantallas por tarea. Número de pantallas utilizadas durante la ejecución de una tarea.

Característica 3 (F3): Precisión. Posibilidad de proporcionar resultados precisos.

Característica 4 (F4): Recursos multimedia. Capacidad para admitir diferentes tipos de medios como video, texto, audio, entre otros.

Necesidad 2 (N2): Pedagógico. (Soad et al., 2016) señala que este aspecto aborda los problemas relacionados con la enseñanza y el aprendizaje. En este sentido, (Baran et al., 2016) mencionan que la aplicación debe presentar contenidos o actividades que mantengan el interés del usuario. En este sentido, (Chen, 2016) señala que las habilidades proporcionadas en la aplicación deben ser coherentes con el objetivo de aprendizaje específico.

Característica 1 (F1): Contenido coherente. El contenido tiene que estar fuertemente conectado con las habilidades o el concepto objetivo.

Característica 2 (F2): Complejidad multimedia. La aplicación debe mantener un balance con el número de elementos multimedia, ya que serían puntos de distracción para el usuario si son demasiados.

Característica 3 (F3): Vistas separadas. Se refiere a la separación de las vistas de alumno y del tutor. Es decir, acuerdo con cada perfil de usuario, la aplicación debe proporcionar diferentes vistas.

Característica 4 (F4): Reutilización del contenido de aprendizaje. Se refiere a la capacidad de la aplicación para proporcionar reutilización de contenido.

Característica 5 (F5): Complejidad del contenido. El contenido no debe tener demasiada información sobre las lecciones y ofrecerse gradualmente, evitando que un contenido complejo afecte la comprensión del usuario.

Necesidad 3 (N3): rendimiento. Según (Soad et al., 2016), se refiere al rendimiento de la aplicación con respecto al consumo de los recursos necesarios hasta que se muestre el contenido.

Característica 1 (F1): Tiempo de carga. Se refiere al tiempo requerido para mostrar la información solicitada por el usuario.

Característica 2 (F2): Eficiencia en el consumo de energía. Esta función se refiere al consumo de energía de los dispositivos al implementar y usar la aplicación.

Característica 3 (F3): Limpieza de memoria. Se refiere al almacenamiento innecesario de archivos que la aplicación ya no usa.

Necesito 4 (N4): Soporte. Se refiere a los aspectos de soporte provistos por la aplicación una vez instalada, y los que el usuario tiene acceso.

Característica 1 (F1): Actualización y configuración. Se refiere al proceso de actualización y configuración de la aplicación. De acuerdo con (Soad et al., 2016), una aplicación debe actualizarse y configurarse automática y rápidamente.

Característica 2 (F2): Resolución de errores. En este sentido, una aplicación debe proporcionar mecanismos que permitan al usuario resolver errores en la aplicación.

Necesidad 5 (N5): Usabilidad. De acuerdo con (Soad et al., 2016) la usabilidad de una aplicación se refiere a la capacidad de una aplicación para ser utilizada por diferentes usuarios. (Balayan et al., 2014) afirma que la usabilidad de una aplicación considera la experiencia del usuario al interactuar con la aplicación, es decir, la complejidad para aprender a usar la aplicación. Los criterios incluyen diseño visual, visibilidad del contenido y control del usuario, entre otros. (Soad et al., 2016) establece las siguientes características para evaluar la usabilidad en aplicaciones de *m-learning*.

Característica 1 (F1): Tiempo de aprendizaje. Se refiere al tiempo requerido por el usuario para aprender a usar la aplicación.

Característica 2 (F2): Homogeneidad del diseño. Se refiere a mantener un diseño homogéneo en todas las interfaces de la aplicación.

Característica 3 (F3): Presentación de la información. Esto se refiere a la capacidad de ver correctamente el contenido de la aplicación en varios dispositivos móviles.

Característica 4 (F4): Diversidad de dispositivos. Se refiere al correcto funcionamiento de la aplicación en varios dispositivos.

La Tabla 4.1 presenta el conjunto de preguntas aplicadas para realizar la evaluación cualitativa descrita.

Tabla 4.1 Cuestionario de la evaluación.

Característica	Preguntas
Funcionalidad	1) ¿Crees que, al usar la aplicación, el tiempo de respuesta y el uso de datos son adecuados? 2) Al realizar una actividad, ¿el número de pantallas es apropiado? 3) Al realizar una actividad, ¿las respuestas de la aplicación son consistentes con la solicitud realizada? 4) ¿La aplicación admite el uso de diferentes recursos multimedia sin afectar el correcto funcionamiento de la aplicación?
Pedagogía	5) ¿El contenido es coherente con cada concepto objetivo o actividad? 6) ¿El contenido multimedia no desvía excesivamente la atención del usuario cuando realiza una actividad? 7) ¿La aplicación admite contenido separado por perfiles: profesor o alumno? 8) ¿La aplicación le permite al usuario adaptar el contenido y usarlo en diversas actividades? 9) 9) ¿El contenido de la aplicación es claro, limpio y concreto?
Rendimiento	10) ¿El contenido de la aplicación se muestra en un tiempo satisfactorio? 11) Cuando el usuario interactúa con la aplicación, ¿el consumo de energía del dispositivo móvil es adecuado? 12) ¿La aplicación elimina el conjunto de archivos que no son necesarios después de un tiempo?
Soporte	13) ¿La aplicación proporciona una guía para resolver problemas comunes? 14) ¿La aplicación proporciona mecanismos de validación para resolver errores básicos de captura de información? 15) ¿Las actualizaciones de la aplicación se instalan automáticamente?

Característica	Preguntas
	16) ¿El proceso de configuración durante una actualización es rápido y fácil de completar?
Usabilidad	17) ¿Es realmente fácil y rápido aprender a usar la aplicación? 18) ¿El diseño en todas las pantallas de la aplicación es homogéneo? 19) ¿El contenido de la aplicación tiene una visibilidad adecuada en varios dispositivos móviles? 20) ¿La aplicación funciona correctamente en varios dispositivos móviles?

Para medir los niveles de los valores obtenidos en la evaluación, se usa la escala de Likert, de acuerdo con (Lubiano et al., 2017) y (Norman, 2010) permite medir de forma efectiva los niveles de varios factores. En esta evaluación se usó una escala Likert de 5 puntos. El valor más bajo (1) representa el puntaje más bajo, y el valor más alto (5) representa el mejor puntaje como se representa en la siguiente escala:

- 5 puntos: muy de acuerdo
- 4 puntos: de acuerdo
- 3 puntos: neutral
- 2 puntos: no estoy de acuerdo
- 1 punto: totalmente en desacuerdo

El puntaje general de cada aplicación fue la suma de todos los puntajes secundarios de los aspectos descritos anteriormente.

4.1.1.2 Resultados

En esta sección, se describen de acuerdo con lo informado por los estudiantes participantes los resultados de la evaluación.

La Tabla 4.2 muestra los resultados obtenidos por aplicación usando la escala de *Liker*. Los resultados en la Tabla 4.2 son el puntaje normalizado promedio de cada característica evaluada por los 60 estudiantes.

Tabla 4.2 Resultados de la evaluación.

Aspect	Khan Academy	Wiki enciclopedia	Kiwix Wikipedia offline	Wiki-Comp
N1				
F1	5	5	3	5
F2	4	5	4	5
F3	3	4	4	5
F4	3	3	3	5
N2				
F1	5	5	4	5
F2	5	5	5	5
F3	5	1	1	1
F4	1	1	1	1
F5	5	4	4	5
N3				
F1	5	5	3	5
F2	4	5	5	5
F3	4	4	4	3
N4				
F1	4	2	3	3
F2	5	5	5	5
F3	5	5	5	4
F4	5	5	5	4
N5				
F1	5	4	3	5
F2	5	4	4	5
F3	5	5	3	5
F4	5	4	4	5
Total	88	81	73	86

Para N1-F1 *Khan Academy*, *Wiki enciclopedia* y *Wiki-Comp* tienen un tiempo de respuesta y uso adecuado de los datos, sin embargo, *Kiwix Wikipedia offline* requiere un mayor consumo de recursos para responder a la solicitud del usuario. En cuanto a N1-F2 *Wiki enciclopedia* y *Wiki-Comp* de acuerdo con los usuarios, el número de pantallas al presentar las respuestas de las solicitudes es adecuado,

mientras que *Kiwix Wikipedia offline* y *Khan Academy* requieren mejorar ligeramente este aspecto. Para N1-F3 *Wiki enciclopedia*, *Kiwix Wikipedia offline* y *Wiki-Comp* los resultados son consistentes con lo que los usuarios solicitaron, sin embargo, es posible mejorar, por otro lado, *Khan Academy*, a pesar de ser eficiente los usuarios recomiendan mejorar este aspecto. Para N1-F4, *Wiki enciclopedia*, *Kiwix Wikipedia offline* y *Khan Academy* se identificó el uso de al menos un tipo de recurso multimedia, pero no combinan diferentes recursos multimedia, aunque en ningún caso los recursos multimedios afectan el adecuado funcionamiento de las aplicaciones. En este aspecto, *Wiki-Comp* presenta a los usuarios recursos multimedia combinados, pero sobre todo dinámicos para el usuario.

En cuanto a N2-F1, *Wiki enciclopedia*, *Khan Academy* y *Wiki-Comp* proporcionan al usuario contenido consistente con los conceptos, con las búsquedas y actividades. Sin embargo, *Kiwix Wikipedia offline* requiere mejorar algunas de sus características para mejorar la eficiencia presentando contenido de acuerdo con los conceptos de búsqueda que el usuario solicite. Mientras que en N2-F2 el contenido multimedia utilizado por *Wiki enciclopedia* y *Kiwix Wikipedia offline* incluye imágenes simples y apropiadas, *Khan Academy* utiliza videos cuyo contenido se centra únicamente en el objetivo de aprendizaje, finalmente *Wiki-Comp* muestra imágenes y videos con diseño amigable sin perder el objetivo de enseñanza. Por otro lado, en N2-F3, la *Kiwix Wikipedia offline*, *Wiki enciclopedia* y *Wiki-Comp* no admiten la gestión de contenido separado por perfiles, porque el objetivo de las aplicaciones es solo apoyar el proceso de aprendizaje de los estudiantes. Sin embargo, *Khan Academy* difiere del resto de las aplicaciones, pues si admite el uso de perfiles. Respecto a N2-F4, las aplicaciones mantienen un contenido fijo, los usuarios no agregan o editan el contenido, por lo que es una característica que en todas las aplicaciones debería mejorarse. En N2-F5 las aplicaciones *Kiwix Wikipedia offline*, *Wiki enciclopedia* presentan texto en la mayoría del contenido, de ahí que es posible mejorar el diseño y por ende la presentación del contenido. En el caso de *Khan Academy* y *Wiki-Comp*, la presentación del contenido es más específica debido al tipo de recursos multimedia utilizados para presentar el contenido.

Para N3-F1, *Wiki enciclopedia*, *Khan Academy* y *Wiki-Comp* muestran el contenido en pantalla de manera eficiente, se debe mencionar que la velocidad de implementación también depende de la calidad de la conexión a Internet. Incluso en el caso de *Wiki enciclopedia* ofrece a los usuarios la opción de guardar el contenido y verlo sin una conexión a Internet. Por otro lado, *Kiwix Wikipedia offline* presenta una desventaja sobre las otras aplicaciones porque el usuario primero tiene que descargar un conjunto de paquetes antes de ver cualquier contenido. N3-F2 *Wiki enciclopedia*, *Kiwix Wikipedia offline* y *Wiki-Comp* mantienen un consumo de energía normal durante su uso, sin embargo, *Khan Academy* requieren consumir un poco más de energía debido a que el contenido que los usuarios visualizan son videos. Con respecto a N3-F3, *Khan Academy*, *Wiki enciclopedia*, *Kiwix Wikipedia offline* permite a los usuarios almacenar contenido temporalmente, sin embargo, el contenido almacenado temporalmente solo se elimina de manera directa por el usuario. En cuanto a *Wiki-Comp*, es necesario revisar e implementar mecanismos que permitan a los usuarios eliminar contenido que el usuario no necesita o usa.

Respecto a N4-F1, *Khan Academy* ofrece una sección de preguntas frecuentes desde un enlace externo a la aplicación. *Wiki enciclopedia*, no ofrece una guía para los usuarios. *Kiwix Wikipedia offline* presenta una breve explicación del uso de las características principales, pero no una guía para la solución de problemas frecuentes. *Wiki-Comp* a pesar de no presentar una guía de solución a problemas frecuentes presenta una interfaz intuitiva que facilita a los usuarios aprender a usar la aplicación. Acerca de N4-F2, las cuatro aplicaciones proporcionan mecanismos de validación adecuados cuando se trata de capturar información. De manera similar, en N4-F4 *Khan Academy*, *Wiki enciclopedia* y *Kiwix Wikipedia offline* los usuarios describen el proceso de configuración de las actualizaciones como adecuado, mientras que para *Wiki-Comp* por el momento no contiene actualizaciones.

Por otro lado, N5-F1 *Khan Academy* y *Wiki-Comp* presentan interfaces de usuario intuitivas y fáciles de entender que simplifican el proceso para aprender a usarlas. Sin embargo, en *Wiki enciclopedia* y *Kiwix Wikipedia offline* es posible mejorar el diseño para que sea más fácil y rápido entender cómo usar la aplicación. Con

respecto a N5-F2, las aplicaciones mantienen un diseño homogéneo en todas sus pantallas, sin embargo, los usuarios consideran que es posible mejorar el diseño en la *Wiki enciclopedia* y *Kiwix Wikipedia offline*. Para N5-F3 *Khan Academy*, *Wiki enciclopedia* y *Wiki-Comp* permiten una visualización correcta del contenido en diferentes tamaños de pantalla según cada dispositivo. Sin embargo, *Kiwix Wikipedia offline* tiene errores en algunos tamaños de dispositivos móviles.

La Figura 4.12 muestra gráficamente los resultados obtenidos de acuerdo con los puntajes de la Tabla 4.2.

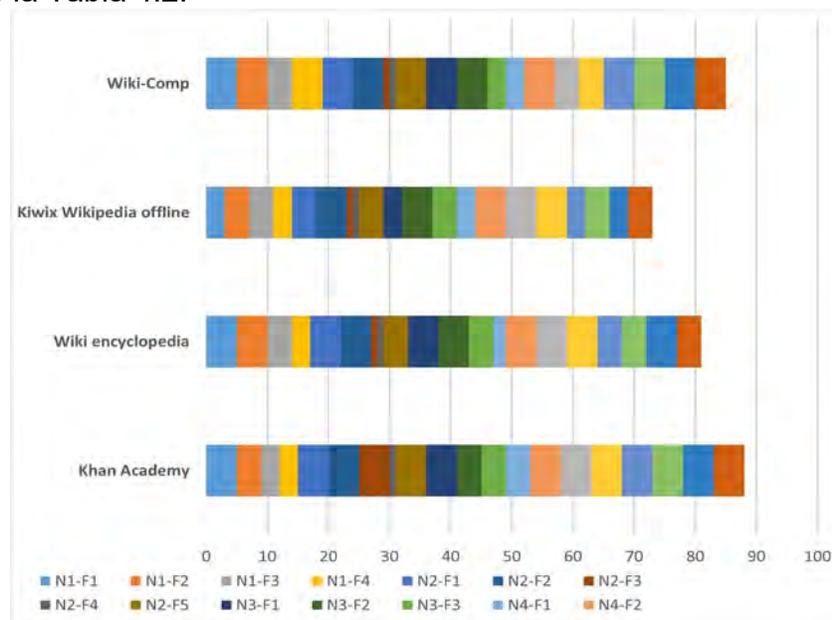


Figura 4.12 Resultados gráficos de la evaluación.

4.1.1.3 Discusión

En aspectos de funcionalidad y usabilidad, *Wiki-comp* sobresale respecto a las tres aplicaciones restantes. Respecto a funcionalidad, *Wiki-comp* mantiene un diseño limpio lo que agiliza el tiempo de respuesta, por otra parte, la interfaz se basa en patrones de diseño lo que permite administrar el número de interfaces usadas por la aplicación. De forma similar, los aspectos de usabilidad en *Wiki-comp* están relacionados con el diseño basado en UIDPs, *Wiki-comp* aprovecha sus características para presentar una interfaz amigable e intuitiva, con adecuada navegabilidad y correcta visualización del contenido independiente del tamaño de pantalla del dispositivo.

En aspectos de soporte *Wiki-comp* requiere implementar mecanismos que ayuden a los usuarios sobre la solución a problemas frecuentes, aunque este aspecto se compensa con la presentación de una interfaz intuitiva. En este mismo aspecto, *Wiki-comp* requiere considerar la instalación y configuración de actualizaciones en futuras versiones. Es importante mencionar que *Wiki enciclopedia*, *Kiwix Wikipedia offline* y *Khan Academy* son aplicaciones disponibles para los usuarios con mayor tiempo que *Wiki-comp*. Sin embargo, la estructura generada de *Wiki-comp* está preparada para agregar los mecanismos necesarios que garanticen instalar actualizaciones. Finalmente, *Wiki-comp* presenta menor puntaje en la característica N3-F3 al no considerar mecanismos que permitan a los usuarios eliminar archivos o contenido que no es requerido por el usuario. En este sentido, es importante mencionar que a pesar de no considerar estos mecanismos, en tiempo y consumo de energía *Wiki-comp* mantiene un balance adecuado mejor que *Khan Academy*.

4.2 Caso de estudio 2: Generación de una aplicación educativa tipo MOOC a través del Wizard

Para ejemplificar la funcionalidad de *MultiGene* desde el proceso de generación ideal, a continuación, se describen las características para una aplicación educativa tipo MOOC que se desea construir.

Se plantea la generación de un MOOC para Android™, la aplicación es para generar la estructura de un curso a partir de la selección directa de los patrones de diseño en *MultiGene*. La aplicación educativa a desarrollar pretende cumplir con los siguientes requisitos funcionales:

- Mostrar el logo de la aplicación antes de ejecutar la aplicación.
- Proporcionar un sistema de autenticación para otorgar acceso solo a los estudiantes registrados.
- Mostrar cursos relacionados de Física.
- Mostrar el plan de estudios de cada curso de manera ordenada.
- Advertir al usuario en caso de problemas en la aplicación.
- Reproducir video clips.

De acuerdo a las características y con la finalidad de cumplir con las necesidades solicitadas, se utilizarán los siguientes patrones de diseño de interfaz de usuario:

- *Splashscreen*: Para presentar el logotipo de la aplicación.
- *Login*: Proporciona un control de acceso a la aplicación.
- *Dashboard*: Muestra los cursos disponibles.
- *List*: Muestra el programa de cada curso en forma de una lista.
- *Video*: Reproducir videoclips.

El primer paso para crear la aplicación, es acceder a *MultiGene* a través del ingreso del usuario y contraseña como en la figura 3.28 (a). Una vez que se inicia sesión, se selecciona el proceso ideal del menú principal, donde el primer paso permite seleccionar el dominio de la aplicación a generar que en este caso es el educativo como se muestra en la figura 4.13.

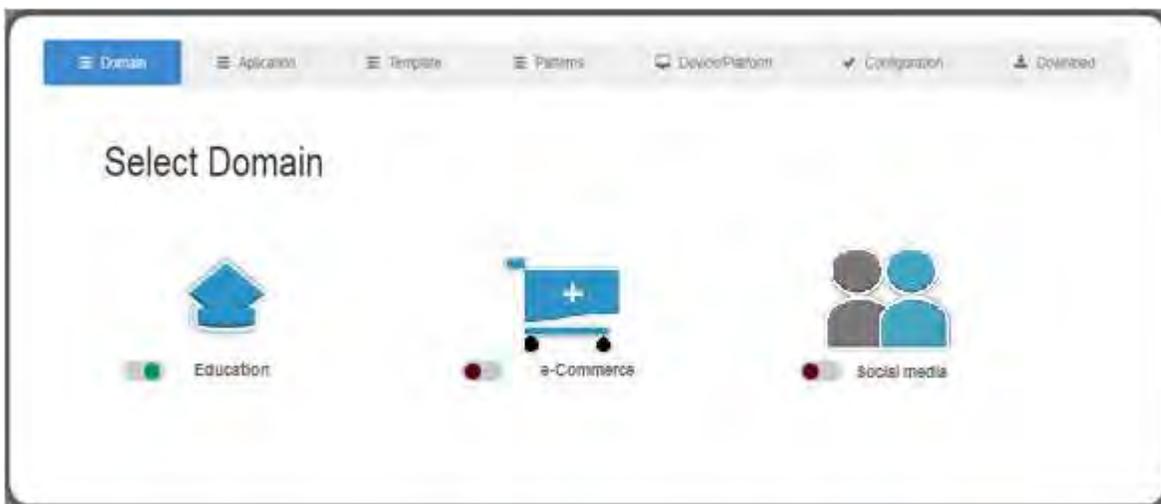


Figura 4.13 Pantalla 1 del proceso ideal: Selección del dominio de la aplicación.

Después de seleccionar el dominio se procede a la selección del tipo de aplicación educativa que en este caso es un MOOC. La figura 4.14 ilustra la selección de una aplicación tipo MOOC.

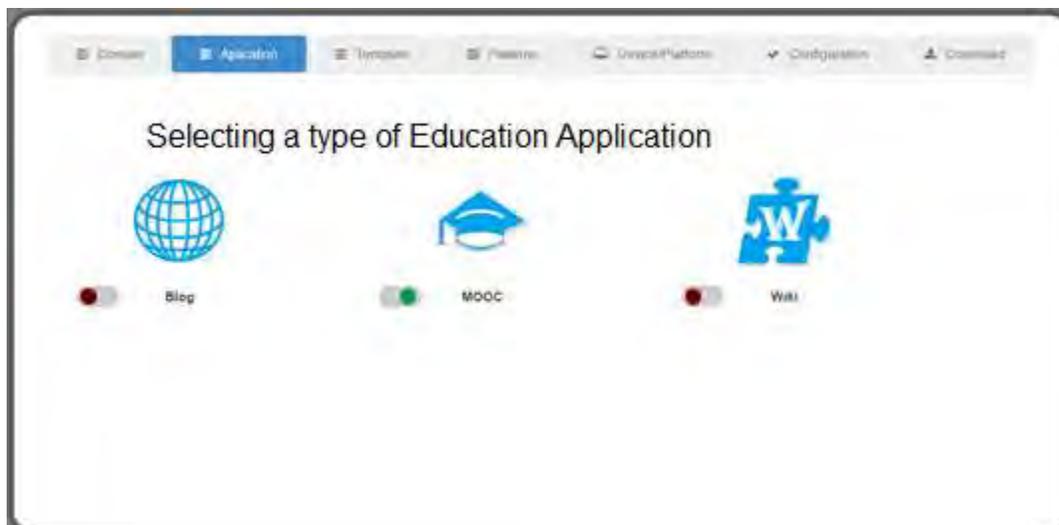


Figura 4.14 Pantalla 2 del proceso ideal: Selección del tipo de aplicación educativa.

A diferencia del proceso a mano alzada, el siguiente paso es seleccionar una plantilla para organizar el contenido dentro de la interfaz de usuario. En este paso, la selección de la plantilla está basado en el tipo de aplicación educativa que se generara, *MultiGene* valida de manera interna que la plantilla sea compatible con un MOOC. En la figura 4.15 se muestra la selección de la plantilla.

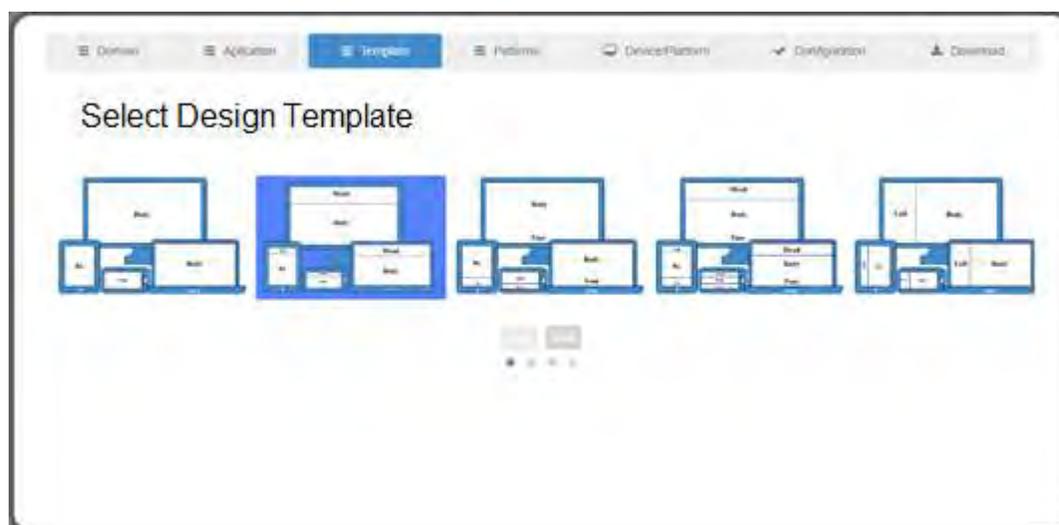


Figura 4.15 Pantalla 3 del proceso ideal: Selección de una plantilla.

Después de seleccionar la plantilla viene la selección de los patrones necesarios para generar la aplicación, en este punto *MultiGene* valida que los patrones sean compatibles con las posiciones disponibles en la plantilla, en caso de no ser

patrones compatibles con la plantilla *MultiGene* muestra un mensaje al usuario con el error. En la figura 4.16 se muestra la selección de los patrones definidos para la generación del MOOC que se planteó para este caso de estudio: *Video*, *Login*, *Dashboard*, *List* y *Splashscreen*.

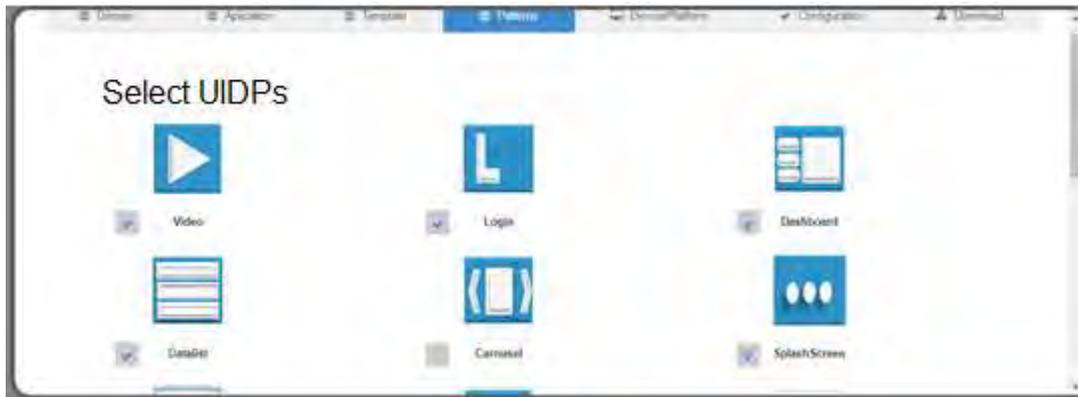


Figura 4.16 Pantalla 4 del proceso ideal: Selección de patrones.

Una vez que se seleccionan los patrones de diseño, el siguiente paso es la selección de dispositivos y plataformas de desarrollo, en este punto la plataforma de desarrollo es Android™ y los dispositivos son *Tablet/Smartphone* tal como se observa en la figura 4.17.

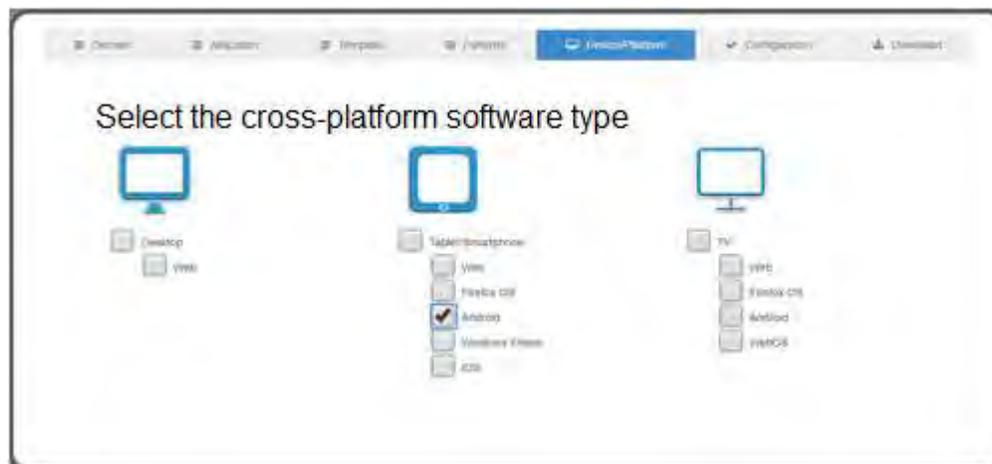


Figura 4.17 Pantalla 5 del proceso ideal: Selección de dispositivos y plataformas.

El siguiente paso consiste en capturar los principales datos para el despliegue de la aplicación a generar, y de manera similar al proceso ideal los datos de configuración son: título de la aplicación, nombre corto, versión y datos del autor de la aplicación.

En la figura 4.18 se muestra la captura de los datos para generar este MOOC educativo.

The screenshot shows a web interface for configuring an application. At the top, there is a navigation bar with several menu items: 'Home', 'Aplicación', 'Herramientas', 'Plantas', 'Servicio al Cliente', 'Configuración', and 'Cuenta'. The main heading is 'Application setting'. Below this, there are two columns of form fields. The left column, titled 'Application Data', includes 'Title' (with the value 'mooctool'), 'Short Name' (with the value 'mooctool'), and 'Version' (with the value '1'). The right column, titled 'Author Details', includes 'Name' (with the value 'Laura Pardo'), 'Company' (with the value 'Instituto Tecnológico de Cetzaco'), 'E-mail' (with the value 'laura@tecnol.com', which is highlighted with a blue border), and 'Website' (with the value 'http://www.itecnol.com/').

Figura 4.18 Pantalla 6 del proceso ideal: Configuración.

Antes de continuar con la generación de la aplicación, *MultiGene* muestra al usuario los datos de configuración seleccionados desde el dominio y tipo de aplicación a generar, hasta la plantilla, patrones de diseño, dispositivos y plataformas de desarrollo. El mensaje de confirmación se observa en la figura 4.19 (a). Al confirmar los datos de la aplicación el siguiente paso es la descarga del código fuente de la aplicación, para lo cual el usuario espera mientras de manera interna se generan los archivos necesarios de la aplicación como se observa en la figura 4.19 (b). Después de generar los archivos de la aplicación *MultiGene* crea la estructura de carpetas con el código fuente, empaqueta en un archivo ZIP y muestra al usuario el enlace de descarga que se observa en la figura 4.19 (c).

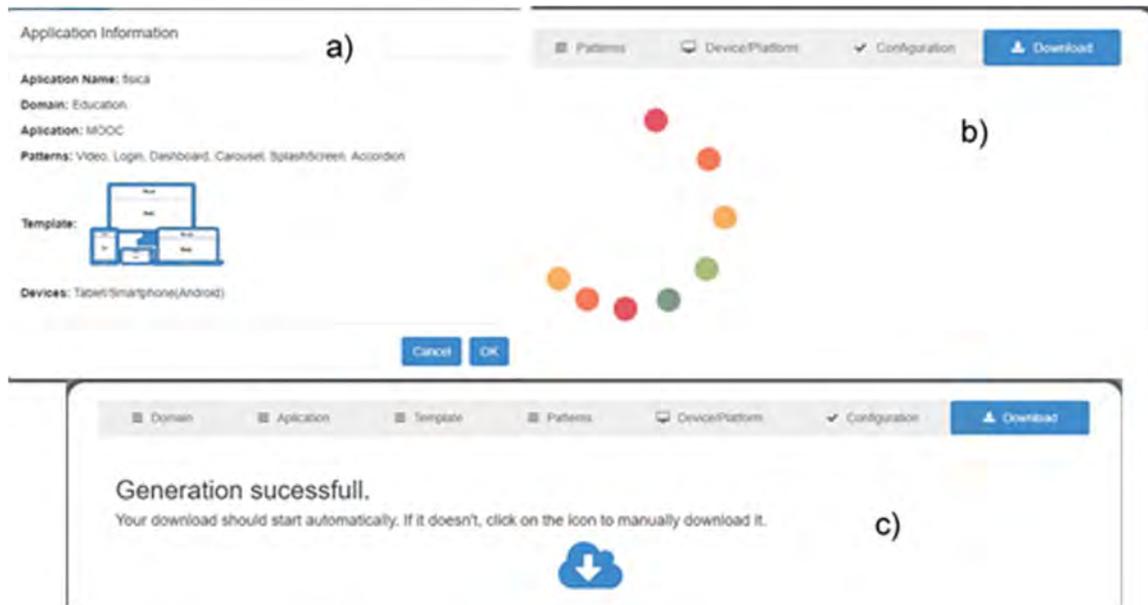


Figura 4.19 Pantalla 7 del proceso ideal: a) Confirmación de la configuración y b) Generación del código fuente de la aplicación y c) descarga de la aplicación.

Después de descargar el código de la aplicación, desde el entorno de desarrollo Android™ Studio se despliega la aplicación a través del emulador. Las interfaces generadas del MOOC se observan en la figura 4.20.

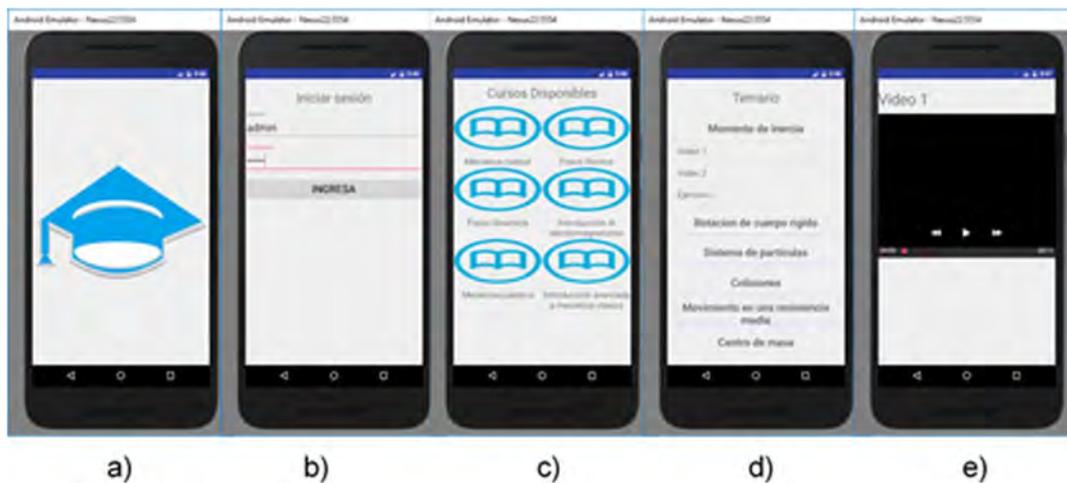


Figura 4.20 Despliegue de la aplicación desde Android™ Studio: a) Splashscreen; b) Login; c) Dashboard; d) List; e) Video.

4.2.1 Evaluación

De acuerdo con (Twining et al., 2017), existe un desequilibrio entre los artículos cuantitativos y cualitativos publicados en revistas de investigación altamente calificadas. Este reclamo es respaldado por numerosos hallazgos. Por ejemplo, (Chiu et al., 2016) revelaron que durante los últimos 31 años, el 37.9% de los estudios han seguido un enfoque cualitativo, mientras que el 17.2% han utilizado un enfoque cuantitativo. Además, el 19,8% más de estudios han adoptado un enfoque mixto (es decir, cualitativo y cuantitativo), otro 19,8% tiene datos cuantitativos cuantificados en revistas internacionales, y el 17,5% de los trabajos de investigación han realizado estudios no empíricos para métodos de codificación. Asimismo, los autores encontraron una tasa creciente de publicaciones que enfatizan en el análisis de datos cualitativos, del 0,3% en el primer período (1982-1987) al 11,5% en el último período (2008-2012). Por esta razón, la evaluación de software tiene que relacionar apropiadamente cada función de software con su método de evaluación correspondiente (es decir, cuantitativo o cualitativo).

Algunos estudios han estudiado y propuesto las características esenciales de las aplicaciones de *e-learning*. Por ejemplo, (Joo et al., 2014) presentaron un estudio sobre el uso de las TIC y destacaron el papel de la interfaz de usuario, desde la perspectiva de los estudiantes, como un factor que afecta la utilidad y facilidad de una aplicación. Del mismo modo, (Mohammadi, 2015) implementó un modelo integrado de sistemas de información y modelo de aceptación de tecnología (TAM) en el contexto del *e-learning* y analizó las características de calidad que afectan las intenciones del usuario y la satisfacción con el *e-learning*. Además, (McGill et al., 2014) presentaron los elementos utilizados para evaluar los factores asociados con la continuidad de las iniciativas de *e-learning*. Los autores encontraron que tener una iniciativa de *e-learning* que funcione para estudiantes y profesores no es suficiente para distinguir entre iniciativas sostenibles e insostenibles.

(Aparicio et al., 2016) proporcionó una comprensión más profunda del impacto de las características culturales de los estudiantes, para el individualismo / colectivismo, en los resultados percibidos del uso de sistemas de aprendizaje electrónico. El estudio propuso un modelo exitoso de sistemas de aprendizaje

electrónico que incluía una construcción cultural en términos de individualismo / colectivismo. Del mismo modo, (Porter et al., 2016) estudiaron los efectos de las estrategias institucionales, la estructura y las medidas de apoyo en la adopción del aprendizaje combinado (LB) entre los docentes de educación superior. Finalmente, (Abdullah & Ward, 2016) identificaron los factores externos más utilizados del TAM en el contexto de la adopción del *e-learning*. Los investigadores llevaron a cabo un meta-análisis cuantitativo de 107 artículos que abarcaron los últimos diez años.

4.2.1.1 Diseño de la evaluación

La evaluación tiene como objetivo evaluar las capacidades de interacción de *MultiGene* y las características educativas con respecto a herramientas similares. Las herramientas de *e-learning* seleccionadas para realizar la evaluación comparativa de *MultiGene* son: Moodle, Sakai y ATutor. Es importante mencionar que el objetivo de la evaluación fue solo resaltar los beneficios de *MultiGene*.

La evaluación comparativa siguió un enfoque cualitativo para evaluar la usabilidad de *MultiGene* y un enfoque cuantitativo para evaluar la herramienta desde una perspectiva de ingeniería a través de un conjunto de métricas de calidad de software.

Con la ayuda de dos paneles de expertos multidisciplinarios y un grupo de estudiantes, se diseñó el instrumento de evaluación para realizar la evaluación. La parte cualitativa de la evaluación involucró un cuestionario cualitativo que fue desarrollado por el primer panel de expertos después de revisar cuidadosamente la literatura y seleccionar los parámetros más apropiados para evaluar la usabilidad de *MultiGene*. Es importante mencionar que el primer panel de expertos fue formado por diez especialistas: tres expertos en desarrollo de software, dos diseñadores de interfaz de usuario y cinco expertos en educación.

Por otro lado, la evaluación cuantitativa involucró un cuestionario de calidad de software y lo diseñó el segundo panel de expertos, es decir, cinco expertos en educación (diferentes de los del panel anterior): un psicólogo, dos pedagogos y dos profesores. El segundo panel de expertos revisó la literatura para identificar y

seleccionar las métricas de calidad para evaluar mejor a *MultiGene* desde el punto de vista de la ingeniería.

Finalmente, se solicitó al grupo de estudiantes que respondieran a la evaluación cualitativa y cuantitativa de *MultiGene*. El grupo incluyó a 40 estudiantes de 18-22 años de seis programas académicos: Sistemas de Información Computacional, Ingeniería Industrial, Ingeniería Química, Ingeniería Eléctrica, Ingeniería Mecánica e Ingeniería de Gestión Comercial. Veinte estudiantes eran hombres y 20 mujeres. La participación de los estudiantes fue un componente clave de la evaluación, ya que los estudiantes dieron su opinión sobre la usabilidad de *MultiGene* y la de las otras herramientas de aprendizaje electrónico (Moodle, Sakai y ATutor).

Para evaluar *MultiGene*, los estudiantes primero exploraron las principales funcionalidades de todas las herramientas. Dichas funcionalidades son iniciar sesión, crear una aplicación educativa, agregar contenido educativo, agregar material multimedia y publicar contenido educativo / compilar el código fuente de la aplicación. El cuestionario fue respondido en línea a través de un enlace de acceso enviado previamente por correo electrónico. El cuestionario incluyó 20 preguntas o puntos que abordaron todos los aspectos cualitativos y cuantitativos de la evaluación. Los aspectos cualitativos son retroalimentación, presentación-claridad, usabilidad, estabilidad y efectividad percibida, mientras que los aspectos cuantitativos incluyeron eficiencia, efectividad y soporte.

En la figura 4.21 muestra un esquema sobre el diseño de la evaluación.

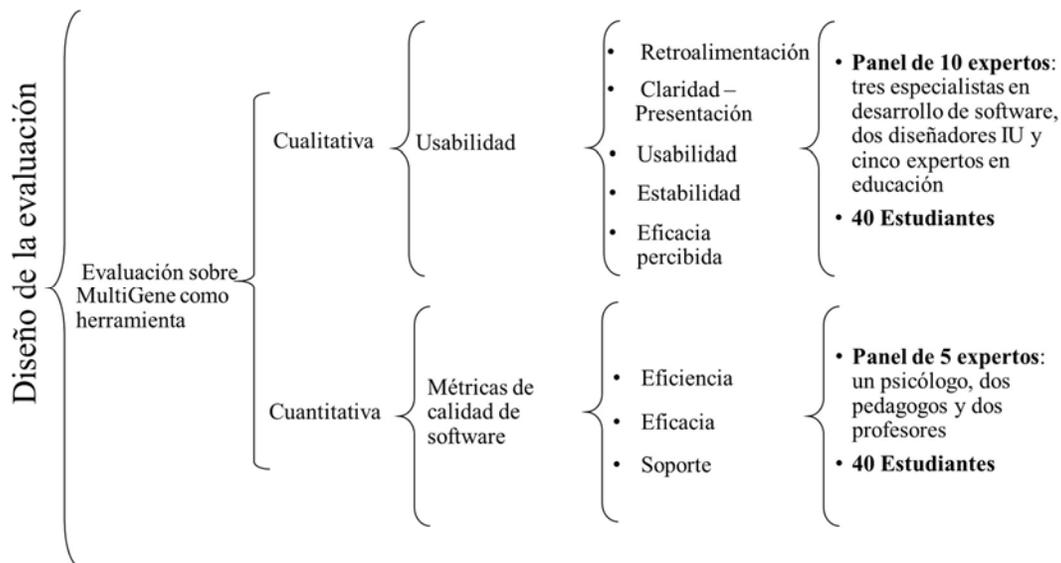


Figura 4.21 Diseño de la evaluación de *MultiGene*.

Como se mencionó anteriormente, el cuestionario fue adaptado de la literatura. Para los aspectos de retroalimentación, se revisó el trabajo de (Davids et al., 2014), mientras que para los aspectos de claridad, se tomó en cuenta (Poulova & Simonova, 2014), (Garcia-Cabot et al., 2015). Los trabajos de (Alshammari et al., 2015) y (Joo et al., 2014) se consultó los ítems del cuestionario que evaluaron la usabilidad de los sistemas de *e-learning*, mientras que la investigación de (McGill et al., 2014) se revisó para aspectos de estabilidad. De manera similar, se consultó la investigación de (Vásquez-Ramírez et al., 2014) para aspectos de efectividad percibida, y se revisó la eficiencia y efectividad en ISO / IEC 25000 (2015). Finalmente, (Porter et al., 2016) y (Poulova & Simonova, 2014) se tuvieron en cuenta para los aspectos de soporte.

Además, la evaluación comparativa de *MultiGene* fue doble ciego. Los estudiantes no se conocían y el cuestionario en línea fue respondido individualmente. El propósito de esta comparación doble ciego fue evitar que los evaluadores intercambiaran sus experiencias durante la evaluación e influir en los resultados. Además, se verificó que los estudiantes no contaran con conocimiento sobre las plataformas evaluadas antes del proceso de evaluación, pero con conocimientos en el uso de dispositivos móviles y sistemas de información de uso general. Las siguientes secciones describen las etapas del análisis cualitativo y cuantitativo.

4.2.1.1.1 Evaluación cualitativa

Para diseñar la evaluación, se consideraron cinco aspectos clave de los sistemas de *e-learning*. Estos aspectos son propuestos por la literatura sobre sistemas de *e-learning* y se dice que aseguran el aprendizaje y la enseñanza adecuado. El objetivo de los aspectos seleccionados fue evaluar la usabilidad de *MultiGene* como una herramienta para el desarrollo de aplicaciones educativas. De acuerdo con (Harrati et al., 2016), la usabilidad es un aspecto importante que tiene que evaluarse cuando se trata de adoptar un sistema de *e-learning*.

Necesidad 1 (N1): Retroalimentación.

De acuerdo con (Davids et al., 2014), la retroalimentación en los sistemas de *e-learning* informa a los usuarios qué acciones tiene permitido realizar. La retroalimentación del sistema se proporciona a través de elementos visuales y mensajes de texto oportunos, así como la adecuada captura de las respuestas de los usuarios.

Característica 1 (F1): Permitir la captura de la respuesta del usuario.

- ¿La herramienta te permite capturar respuestas apropiadamente durante una tarea?

Característica 2 (F2): Ayuda visual.

- ¿La herramienta implementa elementos visuales que dan soporte cuando se realiza una tarea?

Necesidad 2 (N2): Claridad - Presentación.

(Poulova & Simonova, 2014) afirman que cualquier sistema de *e-learning* tiene que ser comprensible para sus usuarios. Esta característica se logra al combinar interfaces gráficas de usuario con contenido apropiado y diseño consistente. Además, (García-Cabot et al., 2015) argumentan que el contenido presentado tiene que ser presentado de manera eficiente.

Característica 1 (F1): diseño ordenado.

- ¿La interfaz de usuario muestra el contenido de forma clara y ordenada?

Característica 2 (F2): consistencia.

- ¿El diseño de la interfaz te ayuda a utilizar la herramienta de forma intuitiva?

Característica 3 (F3): presentación de contenido eficiente.

- ¿El contenido mantiene el orden y la presentación adecuados?

Necesidad 3 (N3): usabilidad

Según (Alshammari et al., 2015), para que un sistema de *e-learning* sea utilizable, su interfaz gráfica tiene que adaptarse a diferentes tamaños de pantalla y fácil de aprender. Del mismo modo, (Joo et al., 2014) argumentan que un diseño eficiente de la aplicación implica necesariamente una interfaz fácil de usar.

Característica 1 (F1): diseño adaptativo.

- ¿Las interfaces se adaptan a diferentes tamaños de pantalla sin perder calidad de diseño?

Característica 2 (F2): facilidad de uso.

- ¿Es la herramienta fácil de usar?

Característica 3 (F3): Facilidad de aprendizaje.

- ¿Es fácil aprender a usar la herramienta?

Necesidad 4 (N4): Estabilidad.

De acuerdo con (McGill et al., 2014), un sistema estable de *e-learning* hace uso de tecnologías novedosas y no obsoletas.

Característica 1 (F1): tecnologías innovadoras.

- ¿La herramienta usa tecnología innovadora?

Característica 2 (F2): funciona con un propósito bien definido.

- ¿Está claro el propósito de la herramienta?

Necesidad 5 (N5): eficacia percibida.

(Vásquez-Ramírez et al., 2014) argumentan que un sistema de *e-learning* tiene que reflejar la utilidad en el proceso de enseñanza-aprendizaje. Aunque muchos sistemas de *e-learning* tienen un gran potencial, no significa que crean entornos de aprendizaje apropiados. Para evaluar este aspecto, se usaron dos características.

Característica 1 (F1): Soporte para el proceso de enseñanza-aprendizaje.

- ¿El sistema apoya el proceso de enseñanza-aprendizaje?

Característica 2 (F2): contenido educativo.

- ¿El sistema le permite, a través de sus funcionalidades, agregar contenido educativo?

De forma similar, en esta evaluación se usó una escala de *Liker* donde el valor (1) representa el puntaje más bajo, y el valor más alto se representa por (5) como se representa en la siguiente escala:

- 5 puntos: muy de acuerdo
- 4 puntos: de acuerdo
- 3 puntos: Neutral
- 2 puntos: no estoy de acuerdo
- 1 punto: Muy en desacuerdo

El puntaje general de cada herramienta fue la suma de todos los puntajes secundarios de los aspectos mencionados anteriormente.

4.2.1.1.1.1 Resultados

Según lo propuesto por (Malganova & Rahkimova, 2015), (Queirós et al., 2016), (Abdullah & Ward, 2016), en las plataformas (Moodle, 2017), (Sakai, 2014) y (ATutor, 2015) para realizar la evaluación comparativa de *MultiGene*, Moodle es un sistema de gestión de aprendizaje en línea (LMS) gratuito que permite a los educadores crear su propio sitio privado lleno de cursos dinámicos. Sakai es una plataforma abierta, flexible y rica en características para el aprendizaje, la enseñanza y la colaboración. Finalmente, ATutor es un LMS de código abierto gratuito diseñado para desarrollar cursos en línea y crear contenido de *e-learning*. La Tabla 4.3 y la figura 4.22 muestran los resultados obtenidos después de evaluar las características de la herramienta con la escala Likert mencionada anteriormente. Por un lado, la tabla 4.3 presenta el puntaje promedio normalizado de cada característica evaluada por los 40 estudiantes.

Tabla 4.3 Resultados de evaluación cualitativa.

Aspecto	Moodle Score	ATutor Score	Sakai Score	MultiGene Score
N1				
F1	5	5	4	4
F2	4	4	4	4

Aspecto	Moodle Score	ATutor Score	Sakai Score	MultiGene Score
N2				
F1	5	4	4	5
F2	5	4	4	5
F3	4	3	3	5
N3				
F1	4	3	3	5
F2	4	3	3	5
F3	4	3	3	5
N4				
F1	4	3	3	5
F2	5	5	5	5
N5				
F1	5	4	4	5
F2	4	2	4	3
Total	53	43	44	56

Por otro lado, la figura 4.22 presenta un gráfico de los resultados de la Tabla 4.3. El gráfico ilustra el puntaje promedio de cada característica para cada herramienta y facilita la comparación entre las características y herramientas.

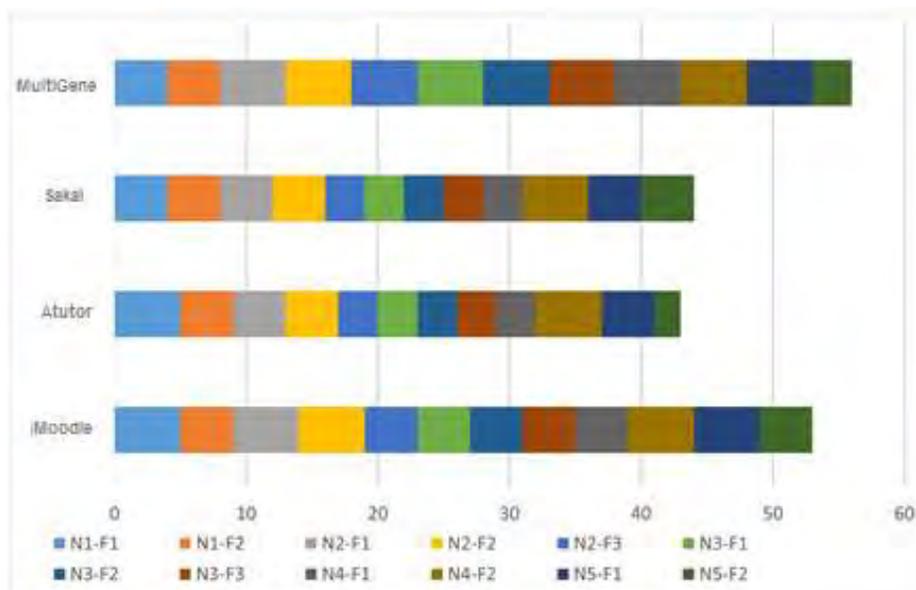


Figura 4.22 Gráfica de resultados de la evaluación cualitativa.

En cuanto a N1-F1, Moodle y ATutor capturan apropiadamente la respuesta del usuario. Sakai y *MultiGene* también logran capturar las respuestas de los usuarios, pero con menor capacidad. En cuanto al aspecto N1-F2, se encontró que todas las herramientas tienen soporte visual, pero no son accesibles para todos los usuarios. En lo que respecta a N2-F1, los puntajes revelan que tanto Moodle como *MultiGene* tienen un diseño ordenado, pero ATutor y Sakai requieren mejorarlo.

Los puntajes N2-F2 demuestran que el diseño basado en UIDPs de *MultiGene* ayuda a los usuarios a usar la herramienta de manera intuitiva, mientras que ATutor y Sakai no parecen tener interfaces altamente intuitivas. Del mismo modo, para el aspecto N2-F3, gracias al uso de UIDP, el contenido en *MultiGene* se presenta de manera eficiente. Sin embargo, el diseño de las otras plataformas no es muy claro y comprensible. En cuanto al aspecto N3-F1, el análisis revela que *MultiGene* ajusta su contenido a diferentes tamaños de pantalla. Moodle también tiene esta capacidad, pero se encontraron defectos en su diseño receptivo. En cuanto a ATutor y Sakai, no usan todos los diseños receptivos. De forma similar, para los aspectos N3-F2 y N3-F3, los diseños de *MultiGene* y Moodle facilitan la interacción del usuario, mientras que ATutor y Sakai no parecen tener interfaces fáciles de usar.

En cuanto al aspecto N4-F1, los puntajes muestran que *MultiGene* utiliza un amplio repertorio de plataformas y dispositivos, para lo cual depende de las tecnologías actuales. Por otro lado, las otras herramientas solo se visualizan a través de un browser Web, aunque Moodle usa Android™ y macOS®. En cuanto al aspecto N4-F2, todas las herramientas presentan claramente su objetivo a los usuarios. En conclusión, *MultiGene* sobresale entre los otros sistemas de aprendizaje electrónico y es muy similar a Moodle; sin embargo, Moodle carece de algunas de las características de *MultiGene*. Además, parece que ATutor y Sakai necesitan mejorar el diseño y la implementación de contenido para llamar la atención de los estudiantes y profesores.

Finalmente, en cuanto a los aspectos N5-F1 y N5-F2, *MultiGene* y Moodle apoyan el proceso de enseñanza-aprendizaje, mientras que Sakai y ATutor parecen tener un menor impacto en los usuarios. Por otro lado, *MultiGene*, Moodle y Sakai permiten a los usuarios agregar contenido educativo a través de tres aplicaciones

educativas diferentes, mientras que en ATutor solo se desarrolla un curso. Además, los resultados revelan que *MultiGene* tiene algunas desventajas si se compara con Moodle y Sakai, ya que no permite a los usuarios almacenar información.

4.2.1.1.2 Evaluación cuantitativa

ISO (Organización Internacional para la Estandarización) e IEC (Comisión Electrotécnica Internacional) propusieron la serie de normas (ISO/IEC, 2015), también conocida como SQuaRE (Requisitos y Evaluación de la Calidad del Software). El objetivo de SQuaRE es proporcionar un *framework* para la evaluación de la calidad del producto de software. Del mismo modo, SQuaRE tiene cinco categorías o divisiones: Gestión de calidad (2500n), Modelo de calidad (2501n), Medición de calidad (2502n), Requisitos de calidad (2503n) y Evaluación de calidad (2504n) (Bevan et al., 2016; Garcés et al., 2017; ISO/IEC, 2015). Dicho esto, para evaluar la calidad del software de *MultiGene*, se mapean cuatro aspectos esenciales de los sistemas de *e-learning* con las sub-características de la división de medición de calidad ISO / IEC 2502n. Los aspectos de calidad evaluados y sus correspondientes métricas de software se describen a continuación.

Necesidad 1 (N1): Eficiencia

Según (ISO/IEC, 2015), la eficiencia es la capacidad del sistema de software para alcanzar los objetivos del usuario con los recursos mínimos. Es decir, la eficiencia tiene que ver con el rendimiento del software y la cantidad de recursos utilizados para realizar una tarea. En este sentido, las características medidas en la evaluación incluyen el tiempo requerido para realizar una tarea y el número de acciones innecesarias que se toman para realizar esta tarea.

Característica 1 (F1): tiempo de la tarea

- ¿Cuánto tiempo lleva crear una aplicación a través de la herramienta?

Característica 2 (F2): acciones innecesarias

- ¿Cuántas acciones innecesarias identificas?

Necesidad 2 (N2): Efectividad.

Según (ISO/IEC, 2015), la efectividad es la capacidad del sistema de software para permitir al usuario alcanzar sus objetivos con precisión e integridad. En este sentido,

se evaluaron dos características de efectividad: la integridad de la tarea y la efectividad de la tarea. Además, la tarea evaluada fue la generación de una aplicación educativa, por lo que la efectividad de las herramientas se basó en esta tarea.

Característica 1 (F1): Corrección de la tarea. Se refiere a la cantidad de tareas completadas correctamente por el usuario y la cantidad de tareas intentadas. Tenga en cuenta que la tarea evaluada fue la generación de una aplicación educativa.

- ¿Se completó con éxito la generación de una aplicación educativa?

Característica 2 (F2): Completitud de la tarea. Se refiere a los objetivos que se logran por una tarea. En este sentido, los 15 objetivos de generar una aplicación educativa son: (1) aumentar el acceso a la información, (2) aumentar la participación de estudiantes y docentes, (3) otorgar acceso a información a usuarios jóvenes y adultos, (4) garantizar el acceso a la herramienta y la permanencia de los miembros, (5) ofrecen mecanismos bilingües y multiculturales, (6) apoyan la inclusión de contenido educativo, (7) aumentan la adquisición de conocimiento, (8) transfieren conocimiento apropiadamente, (9) aumentan el acceso al uso del aprendizaje-tecnologías orientadas, (10) incluyen un sistema de evaluación, (11) incluyen un sistema de retroalimentación, (12) mejoran y adaptan el diseño de contenido educativo, (13) permiten el mantenimiento continuo del contenido educativo, (14) proporcionan una red para aumentar miembros colaboración, y (15) fortalecer la investigación científica y tecnológica.

- ¿Cuántos objetivos se lograron?

Necesidad 3 (N3): Soporte.

(Porter et al., 2014) y (Poulova & Simonova, 2014) afirman que un sistema de aprendizaje electrónico eficaz ofrece ayuda a los usuarios mientras utilizan el sistema, incluso cuando realizan modificaciones.

Característica 1 (F1): Soporte para usar la herramienta.

- ¿Cuántos mecanismos de soporte te ayudan a usar la herramienta?

Característica 2 (F2): Soporte para modificar la herramienta.

- ¿Cuántos mecanismos de soporte te ayudan a modificar la herramienta?

Característica 3 (F3): plataformas compatibles.

- ¿Cuántas plataformas admite la herramienta?

Característica 4 (F4): dispositivos compatibles.

- ¿Cuántos dispositivos admite la herramienta?

4.2.1.1.2.1 Resultados

Para la evaluación cuantitativa, se parte de un conjunto de métricas de calidad de software para analizar la calidad de las herramientas como generadores de aplicaciones educativas.

Tenga en cuenta que los puntajes de eficiencia de las herramientas se muestran en la Tabla 4.4 El tiempo de desarrollo en las cuatro herramientas es muy similar; oscila entre cinco y siete minutos. Además, en cuanto a las acciones innecesarias, ninguno de los cinco pasos de desarrollo (iniciar sesión, crear una aplicación educativa, agregar contenido educativo, agregar contenido multimedia y publicar contenido educativo) se calificó como innecesario en cualquier herramienta. Sin embargo, *MultiGene* no soporta el quinto paso; en cambio, genera el código fuente y los proyectos independientes en función de las plataformas previamente seleccionadas. La ventaja de generar proyectos independientes en un proceso de desarrollo nativo es que estos proyectos se ejecutan de manera eficiente en diferentes dispositivos.

Tabla 4.4 Medida de la eficiencia.

Aspecto	Moodle	ATutor	Sakai	MultiGene
Tiempo de la tarea	5 min.	7 min.	6 min.	5 min.
Acciones innecesarias	0	0	0	0

Como se mencionó anteriormente, la efectividad de las herramientas se midió en términos de tareas correctas y tareas completas. En este sentido, la Tabla 4.5 muestra que 35 de los 40 estudiantes generaron exitosamente una aplicación educativa, mientras que para Sakai y ATutor, solo 32 y 31 estudiantes,

respectivamente, generaron exitosamente una aplicación educativa. En cuanto a los 15 objetivos, todas las herramientas llegaron a más de la mitad; sin embargo, *MultiGene* se destacó con 13 objetivos logrados. La Tabla 4.5 muestra estos resultados.

Tabla 4.5 Medida de efectividad.

Aspect	Moodle	ATutor	Sakai	MultiGene
Tareas correctas	35	31	32	35
Tareas completas	12	8	10	13

También se analizó el soporte técnico que las herramientas ofrecen a los usuarios, llamado, soporte para usar la herramienta y soporte para modificar la herramienta. Además, se evaluó la cantidad de dispositivos y plataformas compatibles con las herramientas. La Tabla 4.6 muestra que Moodle, ATutor y Sakai ofrecen más soporte técnico en comparación con *MultiGene*. Del mismo modo, Moodle ofrece más oportunidades para la configuración de la plataforma que cualquier otra herramienta en evaluación. En cuanto a plataformas y dispositivos, se encontró que *MultiGene* es compatible con un mayor número de plataformas y dispositivos, y por lo tanto, fomenta la interacción de estudiantes / docentes con diversas tecnologías de la información.

Tabla 4.6 Medida del soporte.

Aspecto	Moodle	ATutor	Sakai	MultiGene
Soporte para usar la herramienta	2	2	2	1
Soporte para modificar la herramienta	1	0	0	0
Plataformas soportadas	3	1	1	5
Dispositivos soportados	2	2	2	3

Para apreciar mejor los resultados cuantitativos de la evaluación, la Figura 4.23 muestra gráficamente el puntaje promedio de cada herramienta en cada parámetro cuantitativo. Como se observa, *MultiGene* destaca en varias métricas de calidad, pero carece de otras características que los sistemas restantes poseen, como el soporte técnico.

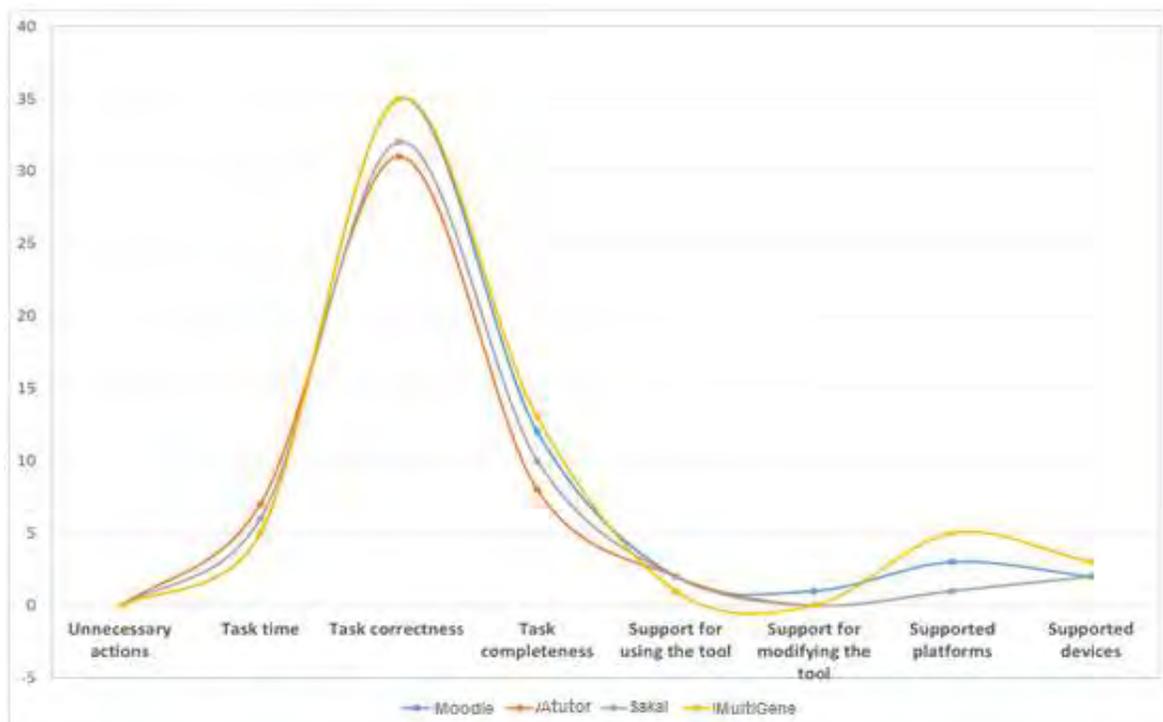


Figura 4.23 Gráfica de resultados de evaluación cuantitativa.

4.2.1.2 Discusión

Los resultados de la evaluación muestran cómo *MultiGene*, si se compara con el resto de las herramientas evaluadas, logra una puntuación más alta en diseño, presentación de contenido y facilidad de uso. Esto se debe principalmente al uso de UIDP en el diseño tanto de *MultiGene* como de sus aplicaciones generadas, ya que los UIDP mejoran las características clave de cualquier aplicación cuyo propósito sea facilitar el proceso de enseñanza-aprendizaje. Además, los resultados demuestran que *MultiGene* logra objetivos educativos tales como aumentar el acceso a la información, alentar la participación de estudiantes y docentes, ampliar la educación a adultos y jóvenes, mejorar la adquisición de conocimientos y mejorar

el diseño del contenido educativo. Sin embargo, en cuanto a las debilidades de *MultiGene*, el sistema proporciona una experiencia menos satisfactoria en términos de soporte. Dicho esto, la asistencia técnica adecuada garantiza un mayor nivel de usabilidad y aumenta la tasa de aplicaciones que se publican con éxito. Por lo tanto, se concluye que *MultiGene* tiene que hacer ajustes sustanciales para mejorar la interacción del usuario y la experiencia general del usuario.

4.3 Caso de estudio 3: Generación de una aplicación e-commerce tipo B2C a mano alzada

El caso de estudio tiene como objetivo presentar la generación de una aplicación de e-commerce. El usuario necesita diseñar una aplicación que facilite la venta de productos en línea a través de un catálogo dinámico de productos disponibles, la búsqueda de productos mediante una palabra clave, la gestión de productos por medio de un carrito de compras. Además, la aplicación debe mejorar el seguimiento de los productos más vendidos o más vistos por los clientes. El caso de estudio plantea la generación de una aplicación e-commerce para dispositivos móviles en Android™. De acuerdo con el diseño de la aplicación, el caso de estudio es una aplicación B2C. El proceso de generación de la aplicación se llevará a cabo a través de la herramienta *MultiGene*. En la herramienta se integró un conjunto de módulos para facilitar la generación de aplicaciones de comercio electrónico. Los módulos integrados tienen funciones específicas: 1) un módulo para identificar los elementos de la imagen mediante técnicas de aprendizaje en profundidad; 2) a partir de los elementos identificados en la imagen, un módulo para generar un documento de configuración basado en XML con la estructura de la aplicación que se va a generar; y 3) un módulo para la generación automática del código fuente de la aplicación a partir de la estructura del documento de configuración basado en XML. Dicho lo anterior, a continuación, se describen las funcionalidades necesarias de la aplicación y los UIDPs requeridos:

- *Search*: Esta función se encuentra en la parte superior derecha de la pantalla y permite al usuario buscar productos.

- *Menu*: Esta función permite acceder a otras vistas de la aplicación para motivar al usuario a visitar un producto de interés, comprarlo o comentarlo. Además, facilita la navegación en la aplicación.
- *Product page*: Permite al usuario ver las características de un producto seleccionado como imagen del producto e información de ventas.
- *Shopping cart*: Esta función permite al usuario consultar los productos seleccionados, comprobar el total de su compra, modificar el número de productos o eliminar un producto de la lista.
- *Stats*: Esta función muestra las estadísticas de los productos más visitados; que sirven a los propietarios para hacer un seguimiento de los productos preferidos de sus clientes.
- *Comment*: Esta función permite a los usuarios añadir opiniones sobre un producto, y también ayuda como medio de interacción entre el comprador y el propietario.

Para generar la aplicación utilizando la herramienta, se utilizó una imagen generada a mano alzada para representar todos los elementos necesarios para construir el código fuente de la aplicación. En este sentido, la imagen representa una composición de UIDP con los siguientes elementos: *Search*, *Menu*, *Product page*, *Shopping cart*, *Stats* y *Comment*. El tipo de imagen fue no ideal, con una resolución de 1050 x 650 píxeles, el formato del archivo fue PNG, y se generó en Paint un editor de dibujos. En la figura 4.24 se muestra la imagen generada para este caso de estudio.

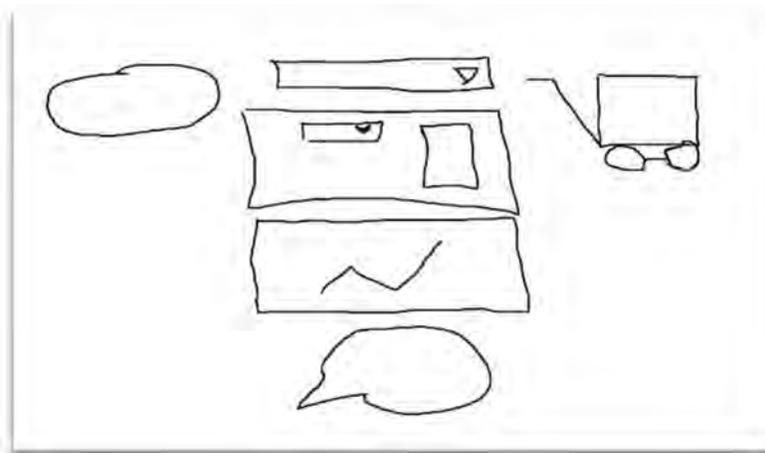


Figura 4.24 Interfaz generada a mano alzada para crear una aplicación B2C.

A continuación, se describe el proceso de generación de la aplicación a través de nuestra herramienta.

1) El primer paso después de iniciar sesión, consiste en subir la imagen para iniciar el proceso de identificación de elementos a través del módulo de Aprendizaje Profundo. La interfaz de la herramienta muestra una vista previa de la imagen cargada. Al hacer clic en el botón Subir Archivo se activa el proceso de identificación de elementos. El resultado de la Red Neural de Aprendizaje Profundo muestra los cuatro resultados más cercanos al componente buscado tal como se ilustra en la Figura 4.25 (a). El tiempo promedio para identificar los elementos de la imagen es de cinco segundos. A partir de los resultados, el usuario selecciona el UIDP correcto o cambiarlo por otro diferente a los sugeridos por el módulo de Aprendizaje Profundo (Ver Figura 4.25-b).

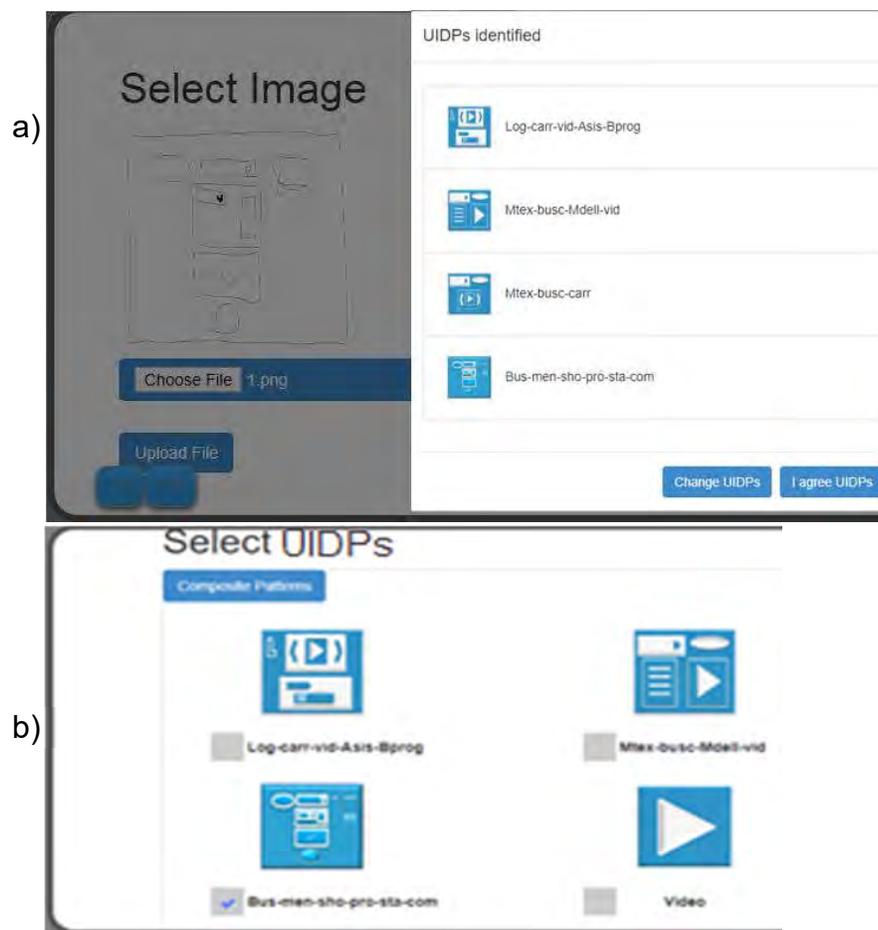


Figura 4.25 Paso 1: a) Elementos identificados en una imagen generada a mano alzada; y b) Selección de UIDPs.

2) El segundo paso consiste en seleccionar una plantilla y el tipo de aplicación de e-commerce. Las plantillas son estructuras predefinidas para organizar el contenido en la interfaz de usuario. Las plantillas propuestas consideran cinco posiciones básicas de una aplicación (Cuerpo, Izquierda, Derecha, Pie y Parte Superior). De acuerdo con los UIDP identificados, se presenta un conjunto de diseños de plantillas a partir de las cuales el usuario selecciona una opción (Ver Figura 4.26-a). En cuanto a la selección del tipo de aplicación de e-commerce, la herramienta válida que los elementos identificados y la plantilla seleccionada en los pasos anteriores son adecuados para generar una aplicación B2C o C2C. Algunos UIDPs son válidos para ambos tipos de aplicaciones, como los UIDPS de *Comment*, *Search* o *Shopping cart*. Si los elementos identificados son válidos para ambos tipos de aplicaciones, la herramienta permite al usuario decidir el tipo de aplicación a generar. En caso contrario, la herramienta muestra sólo el tipo de aplicación válida. En este caso, los elementos identificados son válidos para ambos tipos de aplicaciones; sin embargo, para lograr los objetivos del caso de uso, se seleccionó una aplicación B2C tal como se muestra en la figura 4.26 (b).

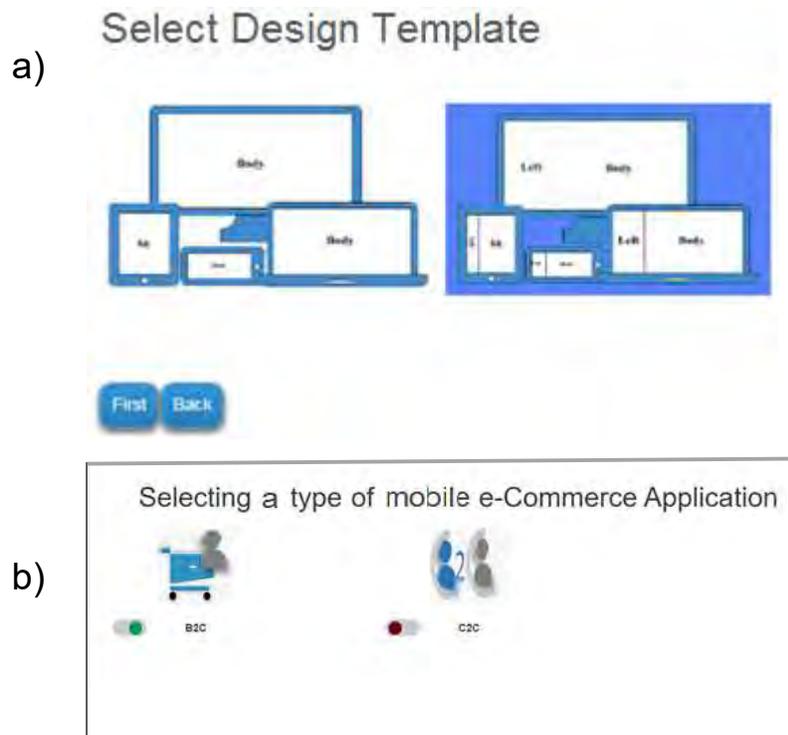


Figura 4.26 Paso 2: a) Selección de una plantilla de diseño; y b) selección de un tipo de aplicación de e-commerce.

3) Este paso consiste en seleccionar la plataforma, el dispositivo y los datos de despliegue para generar la aplicación. En este caso, la aplicación requerida es para un dispositivo móvil Android™, como se selecciona en la Figura 4.27 (a). Los datos de configuración como el nombre de la aplicación, la versión y el autor de la aplicación se agregan a través del formulario tal como se ilustra en la figura 4.27 (b). Al enviar los datos, la herramienta pide al usuario que confirme la información a través de un modal con un resumen de los datos de configuración seleccionados. Si el usuario acepta la configuración, la herramienta genera el código fuente de la aplicación. Una vez generado el código fuente, la herramienta encapsula el proyecto en una carpeta, y se devuelve un archivo ZIP al usuario. La herramienta estructura el código de la aplicación mediante un conjunto de carpetas por plataforma: Móvil, Escritorio y TV. Cada plataforma contiene los proyectos generados separados por el lenguaje de programación. En nuestro caso de estudio, el código del proyecto para Android™ está organizado para permitir al usuario editarlo y configurarlo desde el entorno de desarrollo de Android™ Studio.

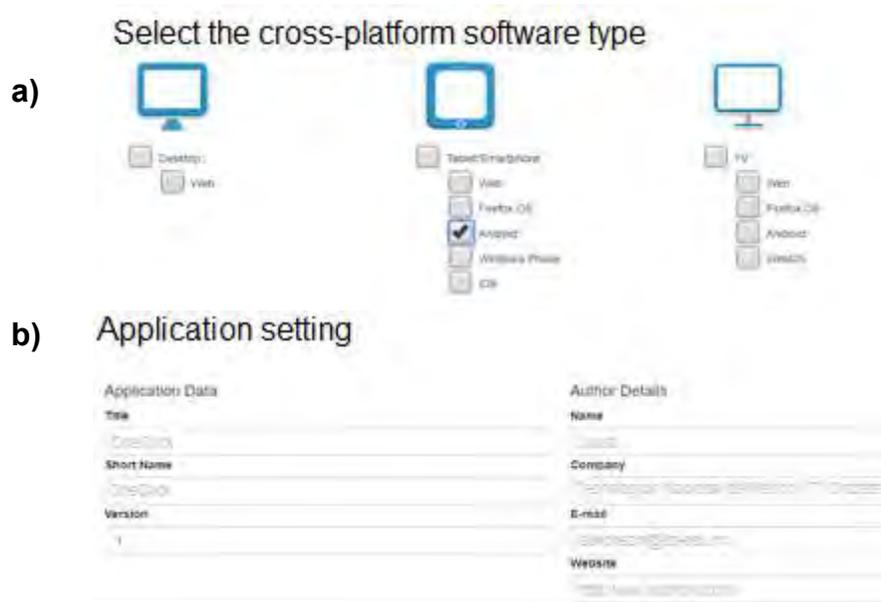


Figura 4.27 Paso 3: a) Selección del tipo de software multiplataforma; y b) configuración de la aplicación.

4) Por último, el usuario despliega la aplicación generada en un entorno de desarrollo de acuerdo con el lenguaje de desarrollo para verificar que las clases y

La configuración del proyecto están libres de errores y listas para un despliegue exitoso. La aplicación generada en nuestro caso de estudio se despliega desde el entorno de desarrollo de Android™ Studio. La figura 4.28 muestra las pantallas de los componentes generados.

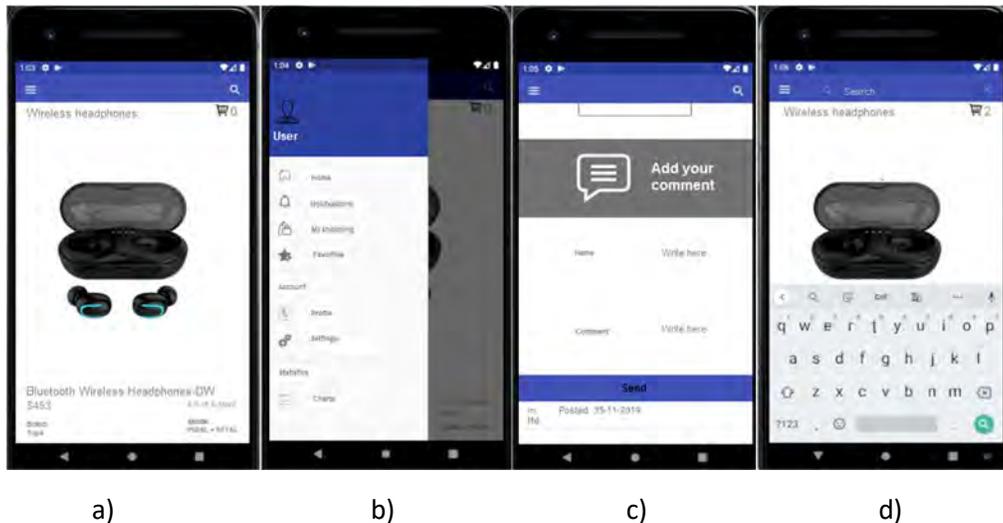


Figura 4.28 Paso 4 UIDPs de la aplicación e-commerce: a) *Product page*, b) *Menu*, c) *Comment*, y d) *Search y Shopping Cart*.

Tal como se observa en la figura 4.28, el código fuente de la aplicación fue generado completamente y está listo para su uso. En este caso, se generó una aplicación de e-commerce para agregar productos a la lista de compras, escribir y revisar los comentarios de los productos buscar y revisar las estadísticas de los productos. Si el usuario lo requiere, desde Android™ Studio se edita el código para añadir funcionalidades más específicas o bien agregar más contenido.

4.3.1 Evaluación

Hay tres enfoques para la evaluación de software: 1) la evaluación cualitativa, que establece la idoneidad de un instrumento o método informático por medio de sus características; 2) la evaluación cuantitativa, que ayuda a determinar en términos mensurables los efectos de la utilización de un instrumento o método informático; y 3) la evaluación híbrida, que es la combinación de la evaluación cualitativa y cuantitativa (Kitchenham, 1996). Desde esta perspectiva, en el diseño de cualquier

software deberían considerarse aspectos de la interacción entre el hombre y la computadora. Según (Scott, Mackenzie, 2012), la interacción se produce cuando un humano realiza una tarea utilizando algún tipo de tecnología de la información. "Realizar una tarea" forma la interacción. Las tareas de interacción están generalmente orientadas a objetivos, como el envío de un correo electrónico o la introducción de un destino en un sistema de posicionamiento global (GPS). (Scott, Mackenzie, 2012) también señala que a veces no hay un objetivo específico, pero navegar por la web o chatear en una red social también se consideran tareas de interacción.

En (Chin et al., 1988), se informó que hay diferentes maneras de evaluar la interfaz humano-computadora, incluyendo tareas de velocidad y precisión, todas asociadas con medidas de rendimiento. Por otra parte, el tiempo necesario para aprender a utilizar un sistema y los conocimientos retenidos adquiridos a lo largo del tiempo son características relacionadas con la eficiencia al utilizar un sistema. Por último, la aceptación de un sistema o la satisfacción subjetiva es la medida que define el éxito de un sistema. Un sistema no se utiliza debido a la insatisfacción del usuario con el sistema o su interfaz de usuario. Desde esta perspectiva, (Sánchez, 2011) estableció que la usabilidad en las aplicaciones de e-commerce debe ser una condición necesaria para la supervivencia. Esto significa que las aplicaciones de e-commerce deben establecer a los usuarios lo que la empresa ofrece, proporcionar información fácil de leer, y el usuario debe encontrar lo que busca y no perderse. De lo contrario, el usuario abandonará la aplicación.

El uso de los dispositivos móviles para las diferentes actividades diarias está en constante crecimiento; por esta razón, es importante considerar el análisis de usabilidad de las interfaces de usuario de las aplicaciones móviles (Ji et al., 2010). Del mismo modo, (Ahmad & Ibrahim, 2017) propusieron siete elementos de diseño que una interfaz de usuario de e-commerce debería tener en cuenta: contexto, contenido, elementos comunitarios, personalización, comunicación, conexión y comercio. Mientras tanto, (Saetang, 2017) subrayó que el diseño de una interfaz de usuario de una aplicación de e-commerce debería proporcionar una experiencia positiva para el usuario basada en tres cualidades: utilidad, facilidad de uso y

conveniencia. En conclusión, las características identificadas en la literatura revisada se centran en el establecimiento de un diseño adecuado de la interfaz de usuario para facilitar la interacción con la aplicación y mejorar la experiencia del usuario. Esto garantiza el éxito de las aplicaciones de e-commerce. Desde esta perspectiva, una evaluación cualitativa es apropiada para destacar las características de las aplicaciones de e-commerce generadas a través del proceso de desarrollo de software propuesto.

4.3.1.1 Diseño de la evaluación

Evaluar las interfaces de usuario es el objetivo de esta sección, por lo que el diseño de la evaluación estuvo a cargo de un grupo de diez expertos (cuatro desarrolladores de software, cuatro diseñadores de interfaces de usuario y dos expertos en marketing). También participaron un grupo de 60 clientes potenciales mexicanos (30 mujeres y 30 hombres); todos los clientes con edades comprendidas entre 18 y 22 años. Para participar en nuestra evaluación, los clientes debían conocer el uso de los dispositivos móviles y tener experiencia en el uso de aplicaciones de e-commerce. Para evitar sesgos en los resultados, ninguno de los clientes se conocía.

El proceso de la evaluación consistió en lo siguiente: se envió por correo electrónico un cuestionario diseñado por el panel de expertos, y un enlace para descargar el apk de nuestra aplicación generada llamada OneClick. En el correo electrónico se describían las instrucciones necesarias para que los estudiantes instalaran OneClick en sus teléfonos móviles, así como una serie de tareas necesarias para evaluar la interfaz de usuario de OneClick. El objetivo de la evaluación de la interfaz de OneClick fue destacar las características de su interfaz de usuario. La verificación de la aceptación de las interfaces de usuario basadas en patrones para el dominio e-commerce también fue importante en esta investigación. Por lo tanto, la experiencia del usuario a través de las tareas de interacción es el parámetro adecuado para validar la idoneidad de las interfaces de usuario generadas a través de nuestro proceso de desarrollo de software propuesto. Las tareas de interacción en cada aplicación fueron las siguientes:

- Buscar al menos tres productos en la aplicación.
- Consultar la información descriptiva de al menos tres productos.
- Añadir al menos tres productos al carrito de la compra.
- Modificar la lista de productos: eliminar un producto de la lista y aumentar el número de productos.
- Añadir un comentario a los productos consultados.

Una vez completadas las tareas de interacción, los clientes respondieron al cuestionario para evaluar las características de la aplicación. Es importante mencionar que la evaluación de la interfaz de OneClick se centra únicamente en el perfil de los posibles usuarios, es decir, los clientes que utilizarían la aplicación para adquirir productos. La evaluación del perfil como propietario o administrador de la empresa queda fuera del alcance de esta evaluación.

En el cuestionario para la evaluación se consideraron cuatro aspectos fundamentales de un diseño basado en la experiencia del usuario: Utilidad, Usabilidad, deseabilidad y fiabilidad del contenido.

Necesidad 1 (N1): Utilidad.

El sistema da valor a los usuarios (Saetang, 2017). Esta característica es la cualidad más importante para que un usuario acepte utilizar cualquier sistema. Según (Knežević & Delić, 2017), las aplicaciones de e-commerce deben ofrecer contenidos útiles o temas interesantes para los usuarios. Esta característica se refiere al grado en que una persona considera que la realización de transacciones en línea a través del comercio móvil mejoraría su rendimiento, puntualizó (Knežević & Delić, 2017). Esto significa que la aplicación de e-commerce debe proporcionar a los usuarios servicios y beneficios adicionales en comparación con las aplicaciones de e-commerce tradicionales.

Necesidad 2 (N2): Usabilidad.

El sistema es fácil de utilizar. Según (Saetang, 2017), la interfaz de usuario de una aplicación e-commerce tiene que ser amigable, intuitiva y fácil de usar. Mientras tanto, (Knežević & Delić, 2017) concluyó que la usabilidad es importante para la satisfacción del usuario; esto es, el grado en que el usuario espera que el sistema sea sin esfuerzo. También se refiere al grado en que los usuarios perciben la

facilidad de interacción con la aplicación, y esta facilidad de interacción les permite obtener la información apropiada y útil que solicitan. Además, en (Indriana & Adzani, 2017) se mencionó la facilidad de uso de una aplicación cuando el usuario interactúa con la aplicación aprovechando todo su potencial.

Necesidad 3 (N3): Deseabilidad.

La aplicación da una experiencia agradable a los usuarios. Según (Saetang, 2017), una aplicación de e-commerce debe tener una buena estrategia de marketing de contenido. Esta estrategia permite que los usuarios tengan la sensación de utilidad y los motiva a usar la aplicación. Además, el autor recomienda añadir una plataforma de red social y una comunidad de práctica en línea como parte de la aplicación e-commerce. (Saetang, 2017) concluyo que, si los usuarios perciben la utilidad y la facilidad de uso de la aplicación, es más probable que tengan una buena actitud y acepten utilizar la aplicación. Por último, el diseño de la aplicación de comercio electrónico requiere que se tengan en cuenta las características culturales de los posibles usuarios para que se sientan cómodos con la aplicación.

Necesidad 4 (N4): Fiabilidad del contenido.

El contenido de la aplicación debe ser accesible y estar correctamente presentado en la aplicación. De acuerdo con (Knežević & Delić, 2017), la fiabilidad del contenido es una característica importante que mejora la experiencia del usuario en las aplicaciones de e-commerce para dispositivos móviles. En este sentido, la aplicación debe mostrar suficiente contenido de pre-información en forma de texto debido al tamaño de la pantalla del dispositivo. Si se presenta el contenido de la pre-información, los usuarios deben tener acceso sin solicitar el pago. Además, la aplicación debe permitir al usuario obtener postscripts o notas después de utilizar el servicio.

Teniendo en cuenta las características seleccionadas, el cuestionario de evaluación se presenta en la Tabla 4.7.

Tabla 4.7 Cuestionario de evaluación cualitativa.

Característica	Pregunta
Utilidad	Q1) ¿Encuentra útil el contenido de la aplicación?

Característica	Pregunta
	<p>Q2) ¿Encuentra interesantes los servicios de la aplicación?</p> <p>Q3) ¿Cree que la aplicación facilita la compra de productos en línea?</p> <p>Q4) ¿La aplicación ofrece algún beneficio nuevo o atractivo para usted?</p> <p>Q5) ¿Encuentra la aplicación atractiva y le gustaría usarla a menudo?</p>
Usabilidad	<p>Q1) ¿Cree que la aplicación es fácil de aprender a usar?</p> <p>Q2) ¿Cree que el soporte técnico es necesario para utilizar la aplicación?</p> <p>Q3) ¿Se siente cómodo usando la aplicación?</p> <p>Q4) ¿Cree que la aplicación no es compleja y proporciona información clara y concisa?</p> <p>Q5) Cuando interactúas con la interfaz de la aplicación, ¿encuentras las diferentes funcionalidades de la aplicación intuitivas?</p> <p>Q6) ¿Cree que es necesario realizar muchos pasos para terminar una tarea en la interfaz de usuario?</p> <p>Q7) ¿Cree que otras personas aprenderían a utilizar la aplicación fácilmente?</p>
Deseabilidad	<p>Q1) ¿Cree que la presentación de productos en la interfaz de usuario es atractiva?</p> <p>Q2) ¿Cree que el diseño de la interfaz de usuario ofrece anuncios innecesarios?</p> <p>Q3) ¿Permite el diseño de la interfaz de usuario una adecuada retroalimentación a través de medios sociales?</p> <p>Q4) ¿Cree que el diseño de la aplicación es congruente con el contenido y apropiado?</p> <p>Q5) ¿Permite el diseño de la interfaz de usuario personalizar la compra de los productos deseados?</p>
Fiabilidad del contenido	<p>Q1) ¿El contenido se presenta de forma ordenada y se muestra correctamente en el dispositivo móvil?</p> <p>Q2) ¿La descripción del producto es suficiente y clara para motivarlo a comprar un producto?</p>

Característica	Pregunta
	Q3) ¿El contenido es consistente con el producto presentado?
	P4) ¿La aplicación le permite ver la descripción completa del producto sin restricciones de acceso?
	P5) ¿La aplicación le permite obtener retroalimentación después de realizar una compra?

4.3.1.2 Resultados

De acuerdo con (Scott, Mackenzie, 2012), la escala de Likert es una herramienta aplicada en los cuestionarios en los que se solicita una escala lineal. Por lo tanto, se diseñó una escala de Likert de cinco puntos para responder al cuestionario. En este caso, el valor más bajo (1) representa la peor puntuación, y el valor más alto (5) representa la mejor puntuación, como se describe a continuación. 5 puntos: Totalmente de acuerdo; 4 puntos: De acuerdo; 3 puntos: Neutral; 2 puntos: En desacuerdo; y 1 punto: Totalmente en desacuerdo.

A continuación, se presentan los resultados obtenidos después de responder al cuestionario y se lleva a cabo una discusión para analizar los resultados. La figura 4.29 muestra los resultados obtenidos, los valores de cada pregunta representan los valores promedio estandarizados.

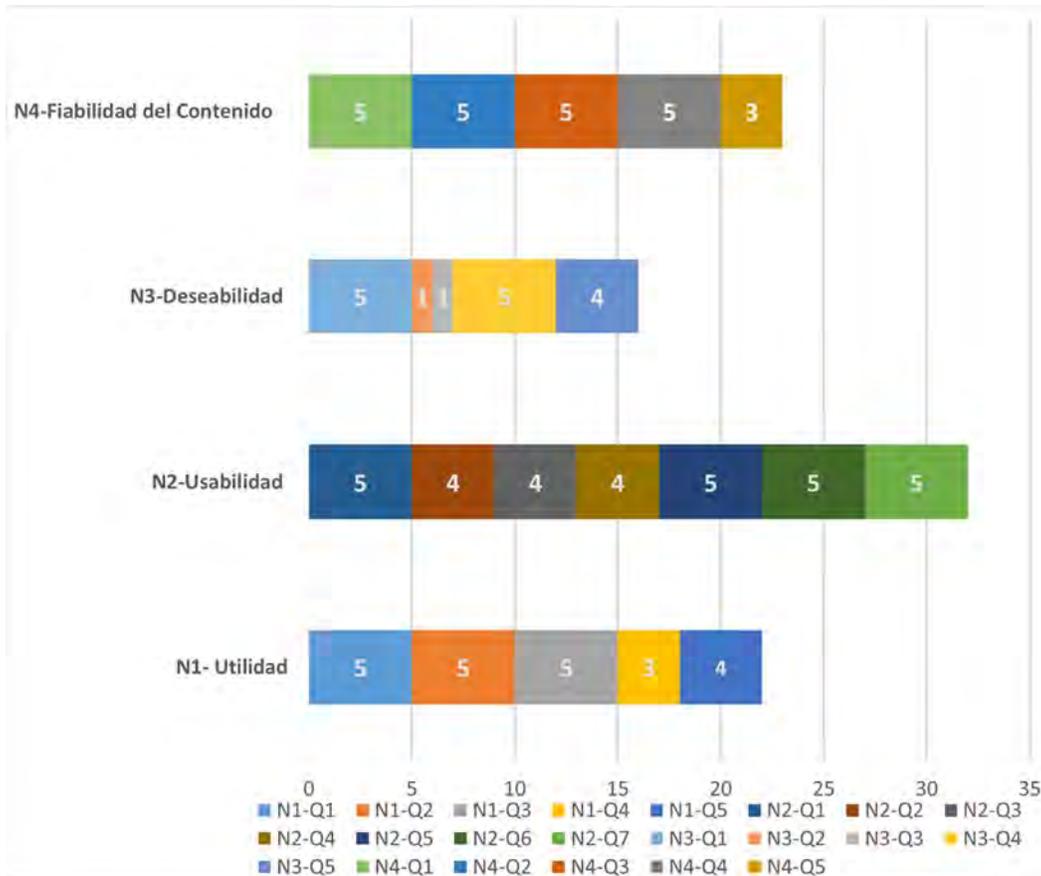


Figura 4.29 Grafica de resultados de evaluación.

4.3.1.3 Discusión

En lo que respecta a N1-Q1, N1-Q2 y N1-Q3, los usuarios están de acuerdo en que el contenido de OneClick es útil, teniendo en cuenta los resultados de las tareas de búsqueda. Encuentran interesante el servicio de compras en línea y, por lo tanto, facilitan la compra de productos. Sin embargo, en N1-Q4, los clientes recomiendan añadir características adicionales como cupones de descuento o puntos adicionales para hacer más atractiva la compra de productos a través de OneClick. Finalmente, en N1-Q5, los clientes están de acuerdo en utilizar la aplicación, pero recomiendan mejorar aspectos relacionados con los productos ofrecidos; hasta ahora, los productos son accesorios para computadoras.

Los resultados de N2-Q1 indican que la aplicación es fácil de aprender a usar. En N2-Q2, N2-Q3 y N2-Q4, los clientes están de acuerdo en que no es vital tener soporte técnico, pero, si se agregan nuevas características, en algún momento se

requerirá asistencia. También están de acuerdo en que la aplicación no es compleja porque presenta un contenido limpio y claro. Finalmente, para N2-Q5, N2-Q6 y N2-Q7, los clientes están de acuerdo en que las funcionalidades exploradas a través de las tareas de interacción fueron intuitivas, lo que no representó una dificultad para completar las tareas. En cada tarea, el número de pasos para completarla era reducido; un máximo de cinco pasos sencillos en el caso de una búsqueda de productos. La mayoría de los clientes estuvieron de acuerdo en que otros usuarios aprenderían fácilmente a utilizar OneClick.

En el caso de N3-Q1, los clientes estuvieron de acuerdo en que la presentación de los productos es atractiva. Los elementos multimedia utilizados en cada producto atraen la atención de los clientes. Para N3-Q2, los clientes están de acuerdo en que el contenido presentado es claro y limpio, y no se utiliza publicidad excesiva. En N3-Q3, los clientes señalan que la interfaz de usuario no presenta opciones de retroalimentación a través de una red social. En N3-Q4, los clientes mencionan que el diseño es congruente con el contenido de la aplicación. Las imágenes y el contenido son apropiados para la aplicación de e-commerce. Finalmente, en N3-Q5, los clientes describen que es posible personalizar la compra, pero recomiendan agregar más características.

En lo que respecta a la fiabilidad del contenido, en los casos N4-Q1, N4-Q2, N4-Q3 y N4-Q4, los clientes califican la visualización del contenido en su dispositivo móvil como adecuada; la descripción del producto es suficiente para que el cliente se interese en adquirirlo. Los clientes reiteran que el contenido (descripción e imágenes) son coherentes con el producto. Una característica importante para los clientes es el acceso gratuito a la descripción completa de un producto sin restricciones. En el caso de N4-Q5, los clientes señalan que es necesario ampliar los mecanismos para obtener retroalimentación después de que se complete el pago de los productos. Algunas sugerencias para mejorar esta última característica son: añadir un mecanismo de seguimiento como la consulta del estado del producto enviado, y añadir restricciones a los cambios o devoluciones.

En conclusión, los datos reunidos mediante la evaluación permiten observar que la interfaz de usuario de OneClick es adecuada, garantizando una experiencia de

interacción con el usuario adecuada. El uso de patrones de diseño de la interfaz de usuario resuelve diferentes problemas de diseño, como las restricciones de tamaño de la pantalla de los dispositivos móviles. Los aspectos mejor valorados son la utilidad, la facilidad de uso y el contenido fiable; sin embargo, es importante reforzar algunas características para mejorar la experiencia del usuario. Según la literatura revisada, los tres aspectos importantes para garantizar la adecuada satisfacción del usuario al interactuar con una aplicación e-commerce son la utilidad, la facilidad de uso y la deseabilidad. Los resultados indican que OneClick cumple con dos de estos aspectos: Utilidad y Usabilidad. Sin embargo, encontramos que es importante mejorar las estrategias de marketing para asegurar la supervivencia de OneClick como plataforma de e-commerce. Aunque estos resultados sirven como margen de referencia, y nos indican que las interfaces generadas a través de nuestra propuesta tienen potencial. Por otra parte, los resultados nos permiten encontrar oportunidades para fortalecer el diseño de las interfaces generadas a través de nuestro enfoque.

4.4 Caso de estudio 4: Generación de una aplicación tipo Social Networks a mano alzada

El objetivo de este caso de estudio consiste en generar una aplicación que permita a los usuarios revisar imágenes de la comunidad, buscar imágenes, reaccionar a una imagen y ver las últimas imágenes actualizadas por el usuario. Además, se busca que el contenido de la aplicación sea desplegado desde una Smart TV. Desde esta perspectiva, se describen a continuación los principales requerimientos funcionales asociados a un conjunto de UIDPs.

- *Search*: permite buscar imágenes de diferentes usuarios dentro de la red social. Se encuentra en la parte superior derecha de la interfaz de usuario.
- *Menu*: permite acceder a otras funcionalidades que motivan al usuario a visitar otras secciones de su interés, tales como revisar imágenes, calificar y seguir a más usuarios.

- *Gallery*: permite ver las imágenes del usuario y de todos los integrantes de la red social.
- *Follow*: permite seguir las actividades de otros usuarios y suscribirse a las actualizaciones de sus últimas imágenes publicadas.
- *Reaction*: permite reaccionar a las imágenes disponibles de los usuarios.

Para generar la aplicación requerida a través de nuestra herramienta es necesario dibujar a mano alzada la representación de los UIDPs, en este caso la representación de la interfaz se realizó en una hoja de papel blanco y un lápiz de color azul. Posteriormente, se digitalizo a través de un dispositivo móvil con un lente de 10 megapíxeles. La imagen resultante tiene una resolución de 1050 x 650 píxeles tomando como referencia el diseño de la composición propuesta con los mismos UIDPs descritos para este caso de estudio. La imagen de la interfaz de usuario generada se observa en la figura 4.30.

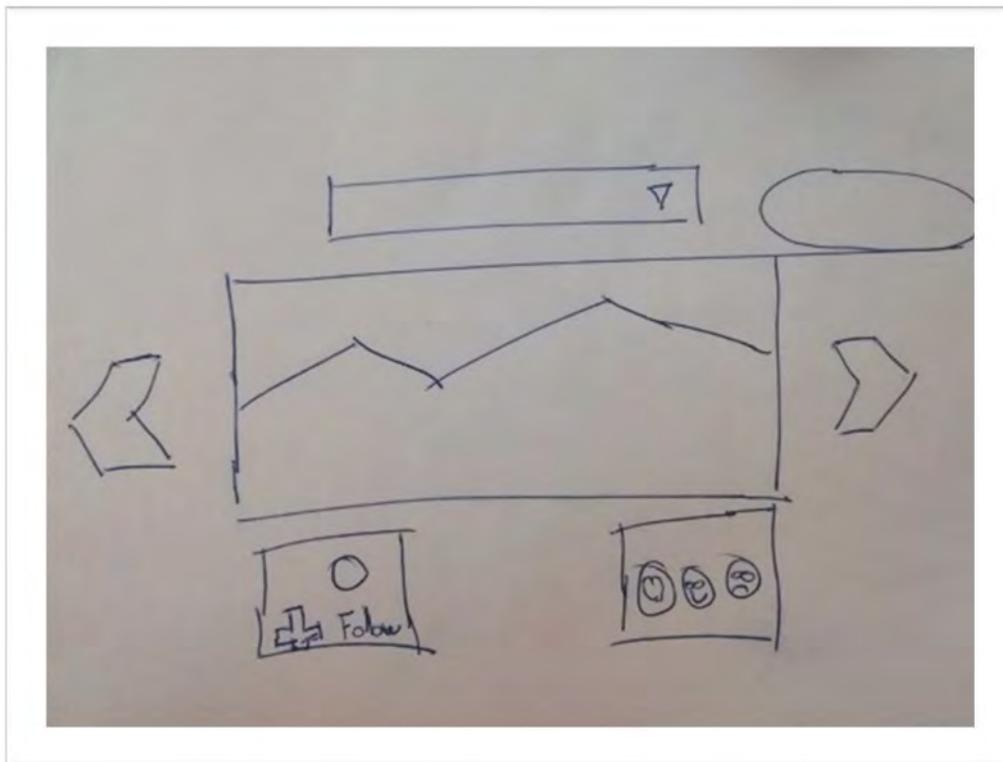


Figura 4.30 Interfaz de usuario para una aplicación tipo social network.

Una vez generada la imagen, el proceso de generación automática del código fuente de la aplicación es lo siguiente, para lo cual a continuación se describen los principales pasos ilustrados a través de las capturas de pantalla de la herramienta.

1) En este caso de estudio el primer paso consiste en cargar la imagen de la interfaz generada, después de la carga de la imagen el módulo de aprendizaje profundo a través de la red neuronal de convolución se encarga del procesamiento e identificación de los elementos en la imagen. Al finalizar el proceso de identificación de elementos en la imagen, la herramienta muestra los cuatro resultados más cercanos al elemento real. En este caso, los resultados obtenidos tal como se observa en la figura 4.31-a son de composiciones, debido a que la imagen usada también es una composición. La herramienta permite al usuario decidir entre los cuatro resultados obtenidos cual será la composición que debe ser tomada en cuenta para la generación del código fuente. En este caso, la composición seleccionada es la que contiene los UIDPs *Search*, *Menu*, *Gallery*, *Follow*, y *Reaction* tal como se observa en la figura 4.31-b.

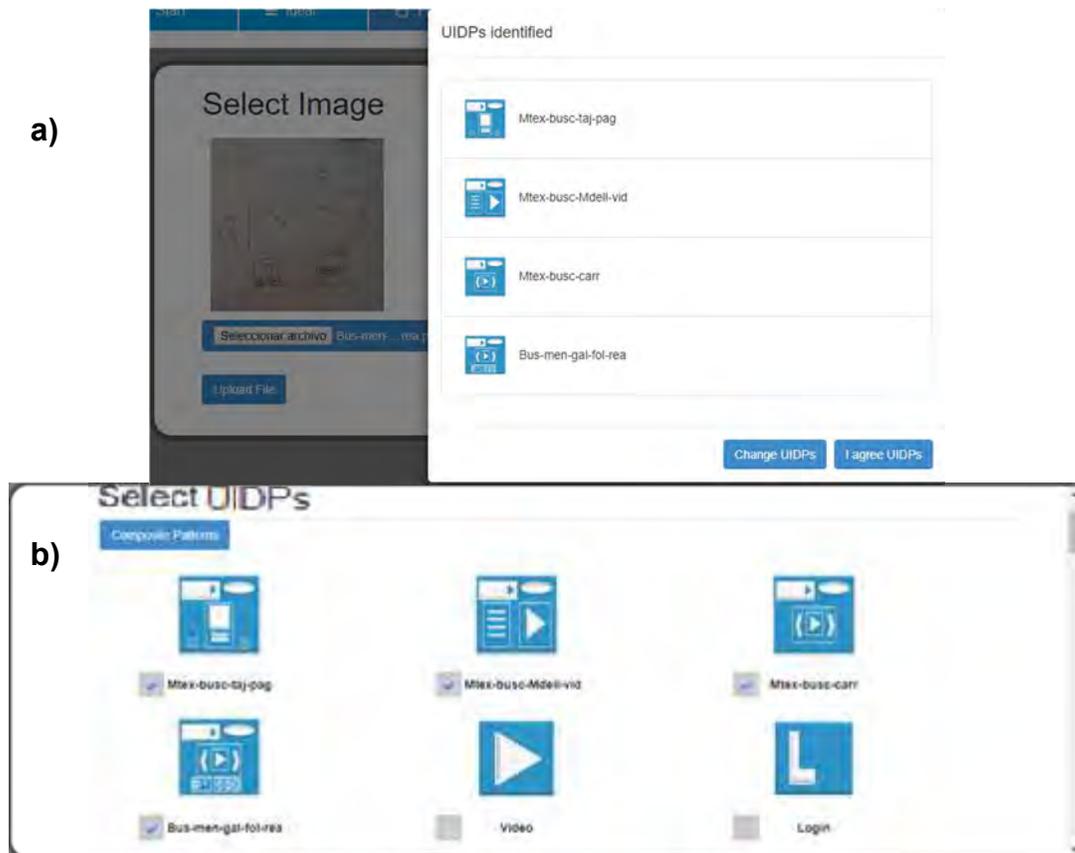


Figura 4.31 Paso 1: a) Elementos identificados, y b) Composición seleccionada.

2) El siguiente paso consiste en seleccionar el dominio de la aplicación a generar, en este caso tal como se describe, la aplicación a generar es una aplicación de Social media, por lo que pasamos a la selección de una plantilla que de acuerdo con los UIDPs que integran a la interfaz de usuario serán mostradas en la interfaz de la herramienta tal como se muestra en la figura 4.32-a. Una vez definida la plantilla para presentar el contenido en la interfaz de la aplicación, es necesario definir el tipo de aplicación Social media a generar. En la figura 4.32-b se aprecia que la herramienta solo muestra la aplicación tipo social networks, debido a que los UIDPs identificados en la imagen son únicamente válidos para este tipo de aplicación.

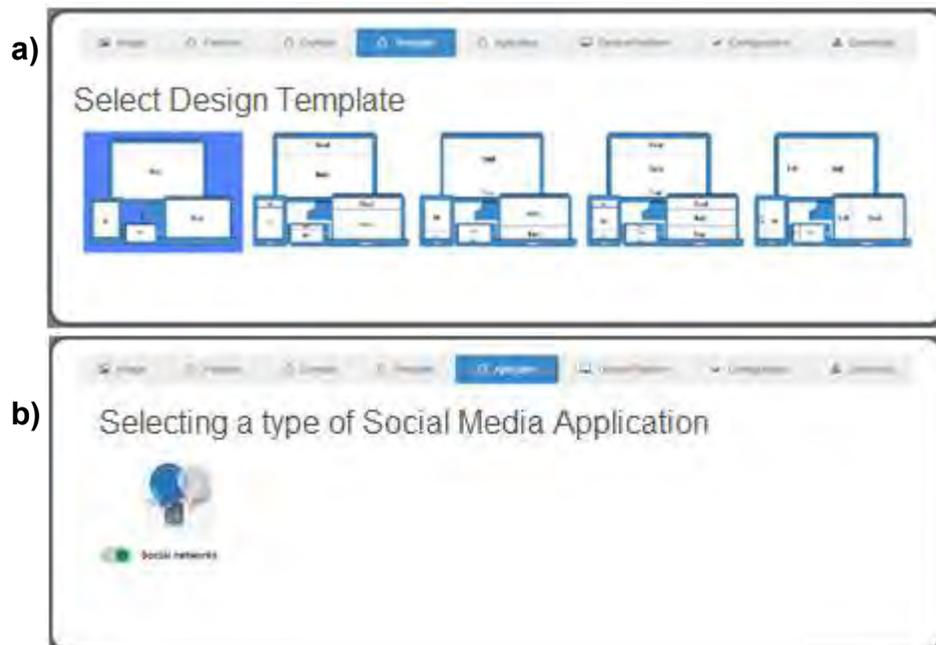


Figura 4.32 Paso 2: a) Selección del diseño de la plantilla, y b) selección del tipo de aplicación social media.

3) Ahora, es importante seleccionar la plataforma y dispositivo para el cual estará disponible la aplicación. En este caso, en la definición del caso de estudio se planteó una aplicación para Android™ y como dispositivo una Smart TV, por lo que en la herramienta así se seleccionaron (Ver figura 4.33-a). Además, es necesario ingresar datos adicionales como el nombre de la aplicación, los datos del autor e institución o empresa a la que pertenece. En la figura 4.33-b) se aprecian los datos ingresados a través del formulario de la herramienta.

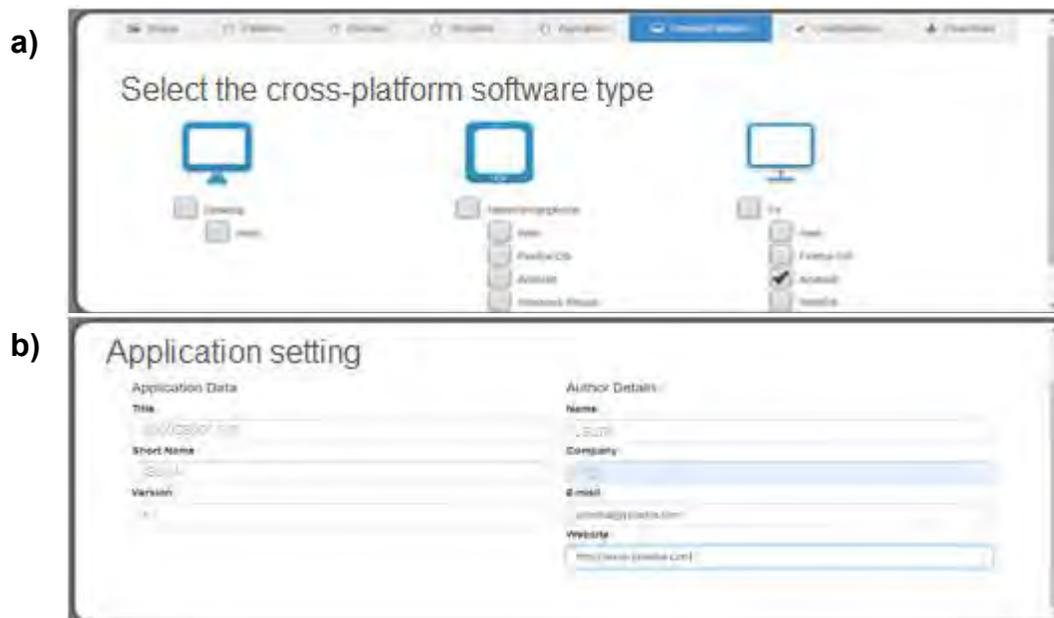


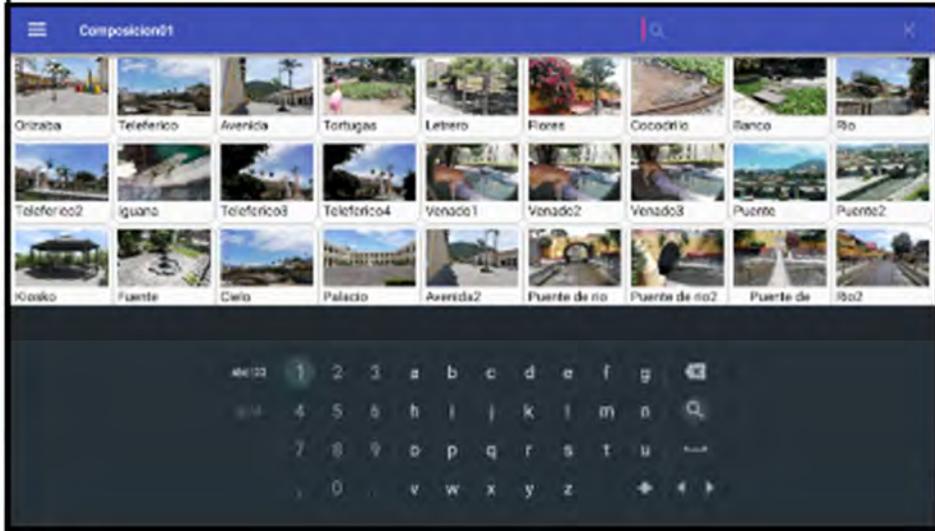
Figura 4.33 Paso 3: a) Selección del diseño de la plantilla, y b) selección del tipo de aplicación social media.

4) Finalmente, previo a la generación automática del código fuente de la aplicación, la herramienta muestra los datos generales con el objetivo de permitirle al usuario continuar con el proceso o regresar y cambiar la configuración. En este caso, los datos de configuración no se modifican por lo que se procede al despliegue de la aplicación desde el emulador de Android™ Studio. Con el despliegue de la aplicación en el emulador, también se valida que el código fuente generado esté libre de errores. Por otra parte, el usuario tiene la oportunidad de editar el código y modificar la funcionalidad de acuerdo a sus necesidades o bien, agregar, actualizar y eliminar contenido personalizado antes de su distribución. La aplicación generada con nuestra herramienta se aprecia en la figura 4.34, donde se observan cada uno de los UIDPs que la conforman.

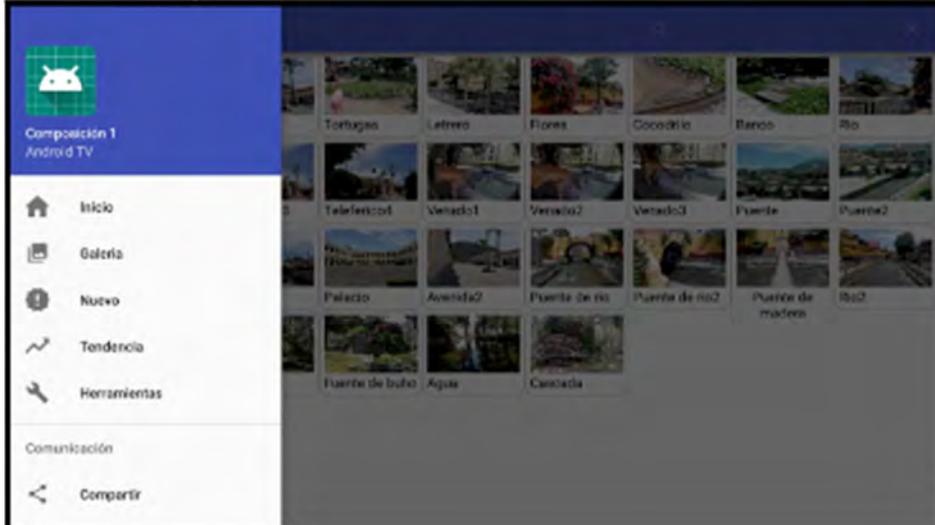
a)



b)



c)



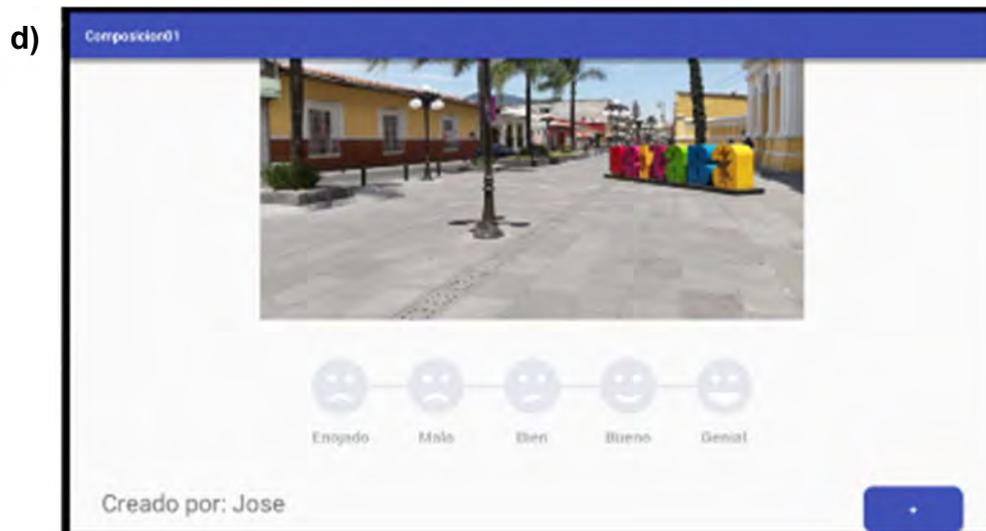


Figura 4.34 Paso 4 UIDPs de la aplicación: a) *Gallery*, b) *Search*, c) *Menu*, y d) *Reaction y Follow*.

Como conclusión, este caso de estudio permite validar que la generación para dispositivos Smart Tv es totalmente factible, el código generado mantiene la estructura de un proyecto en Android™ Studio lo que facilita su compilación y ejecución. Por lo tanto, generar una aplicación Social Media para cubrir las necesidades de aplicaciones para dispositivos Smart TV es satisfactoria. De manera particular, este caso de estudio tiene por default funciones de búsqueda, consulta, reacción a contenido y seguir imágenes de uno o más usuarios todas dirigidas al usuario final. Para el desarrollador, dada la estructura del proyecto permite una rápida manipulación del código generado desde el entorno de desarrollo.

Capítulo 5 Conclusiones

5.1 Conclusiones

Los resultados de esta investigación aportan un nuevo enfoque de desarrollo de software ágil e intuitivo que ofrece la oportunidad de automatizar y reducir el tiempo de desarrollo. En este sentido, las técnicas como el Deep Learning, procesamiento de imágenes y el uso de Patrones de Diseño de Interfaces de Usuario son parte fundamental de este nuevo enfoque de desarrollo. Las características de estas técnicas permiten hacer más rápida la generación automática de código fuente de aplicaciones multi-dominio. A diferencia de otros enfoques, esta propuesta parte del diseño de una interfaz de usuario generada a mano alzada, una técnica novedosa que reduce el nivel de expertise de los usuarios. Con lo anterior, se amplía el abanico de usuarios potenciales dando así la oportunidad a que nuevos usuarios interesados en desarrollar sus propias aplicaciones tengan esa posibilidad. Para usuarios expertos, este nuevo enfoque representa la oportunidad de agilizar las tareas de diseño y codificación durante el ciclo de desarrollo de software.

Dentro de las principales contribuciones de esta investigación están: (1) la generación de software multi-dominio basado en imágenes que son una representación abstracta de una interfaz de usuario; (2) un proceso de desarrollo de software intuitivo basado en técnicas de inteligencia artificial, lo que aporta ventajas significativas en la reducción del tiempo y costo de desarrollo al aprovechar las características de aprendizaje y clasificación de información contenida en imágenes de las redes neuronales de aprendizaje profundo; (3) el desarrollo de aplicaciones con interfaces de usuario que garanticen una adecuada interacción para los usuarios finales, esto gracias a las características de los UIDPs.

Como contribuciones a la ingeniería de software, se tiene: 1) el diseño de un modelo de aprendizaje automático y un conjunto de reglas que permiten la generación automática de código; 2) una adecuada precisión en la clasificación de imágenes que mejore el ciclo de desarrollo de software; 3) el enriquecimiento de las aplicaciones que forman parte de las actividades cotidianas como la educación, el e-commerce, y los medios sociales; y por tanto, la motivación del desarrollo de

software multi-dispositivo y multi-plataforma. Finalmente, 4) una herramienta de desarrollo de software que a diferencia de otras herramientas integra técnicas de inteligencia artificial y los UIDPs para la generación automática de código fuente.

Los resultados obtenidos de los modelos de clasificación para la identificación de UIDPs en imágenes generadas a mano alzada, representan un alto impacto en el tiempo requerido para generar código para aplicaciones multi-dominio, multi-dispositivo y multi-plataforma. Se observó que las técnicas de aprendizaje profundo mejoran significativamente la precisión y el tiempo de respuesta en la clasificación e identificación de elementos en imágenes, lo que permite concluir que este último modelo de clasificación es mejor.

En cuanto a la herramienta, integra los principales módulos para la generación de aplicaciones multi-dominio. Los módulos se basan en técnicas de inteligencia artificial que incluyen el procesamiento de imágenes y el uso de redes neuronales artificiales y de aprendizaje profundo, así como el reconocimiento de composición de patrones de diseño de interfaz de usuario. La herramienta soporta la generación de aplicaciones educativas, e-commerce y social media a través del uso de una imagen que representa una interfaz de usuario perfectamente bien delimitada. En este sentido, el porcentaje de eficiencia en el reconocimiento de elementos es superior al 70 %, sin embargo, el tiempo de respuesta y eficiencia en la identificación de elementos en cuatro tipos de imágenes generadas a mano alzada es adecuado. Otro aspecto importante, es el diseño de la arquitectura de software que es parte medular en este proyecto de investigación pues es una de las aportaciones del trabajo. Dicho lo anterior, el diseño presentado es una propuesta a partir de la cual *MultiGene* fue desarrollada y por tanto se integraron los módulos y servicios necesarios para la generación automática de aplicaciones multi-dominio.

5.2 Trabajo a futuro

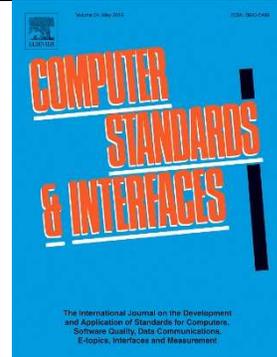
Como trabajo a futuro, es importante ampliar los dominios soportados para la generación de aplicaciones e incluir dominios como el medico. Además, explorar otras técnicas novedosas de desarrollo de software a partir del reconocimiento de

voz o texto, serán oportunidades que permitirán evaluar y comparar la factibilidad nuestro proceso de desarrollo de software y su facilidad para adaptarse a otros datos de entrada.

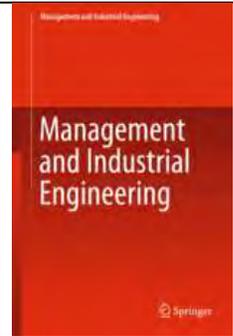
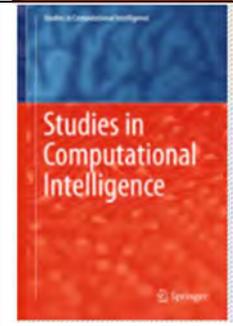
Respecto a la precisión para la identificación de elementos en imágenes, será necesario incrementar el número de muestras para mejora de la eficiencia del módulo de procesamiento e identificación de elementos en imágenes. Así también, es importante ampliar el número de componentes de UIDPs, lo que permitirá reforzar las características de diseño, funcionalidades y por tanto generar aplicaciones mas complejas. Finalmente, el proceso de desarrollo propuesto servirá como referencia para iniciar una nueva línea de investigación de desarrollo de software basado en técnicas de inteligencia artificial.

Productos Académicos

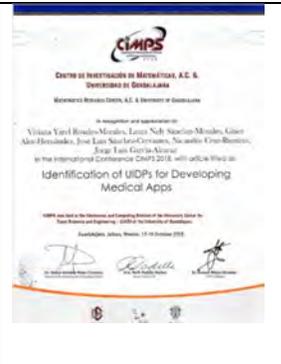
Revistas indizadas en el Journal Citation Reports

	<p>Sánchez-Morales, Laura N.; Alor-Hernández, Giner; Rosales-Morales, Viviana Y.; Cortes-Camarillo, Cesar A. Generation of educational mobile applications based on UIDPs identified by artificial intelligence techniques. Computer Standards & Interfaces. Editorial: Elsevier. ISSN: 0920-5489. Doi: https://doi.org/10.1016/j.csi.2019.103407 I.F. 2.447 (Q2). Estatus: Publicado</p>
	<p>Cortes-Camarillo, Cesar A.; Alor-Hernández, Giner; Sánchez-Morales, Laura N.; Rosales-Morales, Viviana Y.; Rodríguez-Mazahua, Lisbeth; Sánchez-Cervantes, José L.; EduGene: A UIDPs-based educational application generator for multiple devices and platforms. International Journal of Human-Computer Interaction. Editorial: Taylor & Francis. ISSN: 1532-7590. doi: https://doi.org/10.1080/10447318.2018.1459347 I.F.: 1.905 (Q2) Estatus: Publicado</p>
	<p>Rosales-Morales, Viviana Y.; Sánchez-Morales, Laura N.; Alor-Hernández, Giner; Garcia-Alcaraz, Jorge L.; Sánchez-Cervantes, José L.; Rodríguez-Mazahua, Lisbeth; ImagIngDev: A New Approach for Developing Automatic Cross-Platform Mobile Applications Using Image Processing Techniques. The Computer Journal. Editorial: Oxford University Press. ISSN 1460-2067. DOI: https://doi.org/10.1093/comjnl/bxz029 . I.F: 0.915 (Q2). Estatus: Publicado.</p>
	<p>Sánchez-Morales, Laura N.; Alor-Hernández, Giner; Colmenares-Morales, Aaron; Mejía-Miranda, Jezreel. A New Software Development Process for Mobile E-Commerce applications by using Deep Learning Techniques. Journal of Universal Computer Science (J.UCS). Editorial: J.UCS Consortium. ISSN: 0948-695X. I.F. 1.066 (Q2). Estatus: Primera revisión.</p>

Capítulos de libro

	<p>Sánchez-Morales, Laura N.; Alor-Hernández, Giner; Miranda-Luna, Rosebet; Rosales-Morales, Viviana Y.; Cortes-Camarillo, Cesar A.; Generation of User Interfaces for mobile applications using neuronal networks. Management and Industrial Engineering. Springer Verlag. Vol. 1(11). ISSN: 2365-0532. ISBN 978-3-319-56871-3 (eBook). DOI: https://doi.org/10.1007/978-3-319-56871-3_11 . Estatus: Publicado</p>
	<p>Rosales-Morales V.Y., Cruz-Ramírez N., Sánchez-Morales L.N., Alor-Hernández G., Quiroz-Castellanos M., Mezura-Montes E.; Towards a quantitative identification of mobile social media UIDPs' visual features using a combination of digital image processing and machine learning techniques. Studies in Computational Intelligence. Springer Verlag. ISSN: 1860-949X. DOI: https://doi.org/10.1007/978-3-030-35445-9_45 Estatus: Publicado.</p>

Artículos en congreso y seminarios

	<p>Cortes-Camarillo, Cesar A.; Rosales-Morales, Viviana Y.; Sanchez-Morales, Laura N.; Alor-Hernández, Giner; Rodríguez-Mazahua, Lisbeth; Atila: A UIDPs-based educational application generator for mobile devices. In Electronics, Communications and Computers (CONIELECOMP), 2017 International Conference on (pp. 1-7). IEEE. ISBN: 978-1-5090-3621-9. DOI: https://doi.org/10.1109/CONIELECOMP.2017.7891810 . Estatus: Publicado.</p>
	<p>Rosales-Morales V.Y., Sánchez-Morales L.N., Alor-Hernández G., Sánchez-Cervantes J.L., Cruz-Ramírez N., García-Alcaraz J.L. (2019) Identification of UIDPs for Developing Medical Apps. In: Mejia J., Muñoz M., Rocha Á., Peña A., Pérez-Cisneros M. (eds) Trends and Applications in Software Engineering. CIMPS 2018. Advances in Intelligent Systems and Computing. Vol. 865. Springer, Cham. ISBN: 978-3-030-01170-3. DOI: https://doi.org/10.1007/978-3-030-01171-0_16 . Estatus: publicado.</p>

	<p>Rosales-Morales V.Y., Sánchez-Morales L.N., Alor-Hernández G., Cruz-Ramírez N., Quiroz-Castellanos M., Mezura-Montes E. Identification of UIDPs for Developing Social Media Apps. In International Seminar on Computational Intelligence (ISCI) 2019.</p>
---	--

Registros de derechos de autor

	<p>Alor-Hernández, G., Sánchez-Morales, L. N., Sánchez-Cervantes J. L. & Sánchez-Ramírez, C. EDUMATIC: Generador de Aplicaciones Educativas Usando Técnicas de Inteligencia Artificial. Estatus: Registrado. Numero de registro INDAUTOR: 03-2017-090413515100-01.</p>
---	--

Proyecto de investigación como colaborador

	<p>Desarrollo de una herramienta para la generación de aplicaciones educativas multi-plataforma y multi-dispositivo usando patrones de diseño de interfaces de usuario y técnicas de inteligencia artificial. Convocatoria de Apoyo a Proyectos de Investigación Científica, Aplicada, Desarrollo Tecnológico e Innovación 2017. No. De registro: 6155.17-P. TecNM. Estatus: Concluido.</p>
---	---

Estancias de Investigación

	<p>CIMAT (Centro de Investigación en Matemáticas), Zacatecas. Dr. Jezreel Mejia Miranda Objetivo: Definir un proceso formal de desarrollo de software multi-dominio. Resultados: Representación técnica de la formalización del proceso de desarrollo propuesto. Periodo: 01 de Septiembre-31 de Noviembre de 2019 Estatus: Concluida</p>
---	--

Referencias

- Abdullah, F., & Ward, R. (2016). Developing a General Extended Technology Acceptance Model for E-Learning (GETAMEL) by analysing commonly used external factors. *Computers in Human Behavior*, 56, 238–256. <https://doi.org/10.1016/j.chb.2015.11.036>
- AcademyKhan. (2018). *khan academy*. <https://es.khanacademy.org/about>
- Acerbis, R., Bongio, A., Butti, S., Brambilla, M., Milano, P., & Milano, D. (2015). Model-Driven Development of Cross-platform Mobile Applications with WebRatio and IFML. *2nd ACM International Conference on Mobile Software Engineering and Systems Model-Driven*, 170–171. <https://doi.org/10.1109/MobileSoft.2015.49>
- Ahmad, Z., & Ibrahim, R. (2017). Mobile Commerce (M-Commerce) Interface Design: A Review of Literature. *IOSR Journal of Computer Engineering*, 19(3), 66–70. <https://doi.org/10.9790/0661-1903046670>
- Alghabban, W. G., Salama, R. M., & Altalhi, A. H. (2017). Mobile cloud computing: An effective multimodal interface tool for students with dyslexia. *Computers in Human Behavior*, 75, 160–166. <https://doi.org/10.1016/j.chb.2017.05.014>
- AliExpress. (2018). *AliExpress*. <https://play.google.com/store/apps/details?id=com.alibaba.aliexpresshd&hl=en>
- Alshammari, M., Anane, R., & Hendley, R. J. (2015). Design and Usability Evaluation of Adaptive e-learning Systems Based on Learner Knowledge and Learning Style [Inbook]. In J. Abascal, S. Barbosa, M. Fetter, T. Gross, P. Palanque, & M. Winckler (Eds.), *Human-Computer Interaction -- INTERACT 2015: 15th IFIP TC 13 International Conference, Bamberg, Germany, September 14-18, 2015, Proceedings, Part II* (pp. 584–591). Springer International Publishing. https://doi.org/10.1007/978-3-319-22668-2_45
- Amazon-Mobile-LLC. (2020). *Amazon Compras*. <https://play.google.com/store/apps/details?id=com.amazon.mShop.android.shopping>
- Anoop, S. (2019). *What exactly is Netflix?* <https://www.quora.com/What-exactly-is->

Netflix.

Aparicio, M., Bacao, F., & Oliveira, T. (2016). Cultural impacts on e-learning systems' success [Article]. *Internet and Higher Education*, 31, 58–70. <https://doi.org/10.1016/j.iheduc.2016.06.003>

Arnold, S. C., Mark, L., & Goldthwaite, J. (2000). Programming by voice, vocalprogramming. *Annual ACM Conference on Assistive Technologies, Proceedings*, 149–155. <https://doi.org/10.1145/354324.354362>

ATutor. (2015). *ATutor Learning Management System: Information:*

Ayub, M., & Saleem, M. A. (2013). A Speech Recognition based Approach for Development in C ++. *IJCSI International Journal of Computer Science*, 10(5), 52–56.

Balayan, M. P. A., Conoza, V. V. B., Tolentino, J. M. M., Solamo, R. C., & Feria, R. P. (2014). On Evaluating SkillVille : An Educational Mobile Game on Visual Perception Skills. *IISA 2014 - 5th International Conference on Information, Intelligence, Systems and Applications*, 69–74. <https://doi.org/10.1109/IISA.2014.6878828>

balsamiq.com. (n.d.). *balsamiq.com*. Retrieved May 20, 2018, from <https://balsamiq.com/>

Baran, E., Uygun, E., & Altan, T. (2016). Examining Preservice Teachers ' Criteria for Evaluating Educational Mobile Apps. *Journal of Educational Computing Research*, 58(4), 1117–1141. <https://doi.org/10.1177/0735633116649376>

Betrián, A. P., Ingeniero, D., & Informática, P. De. (2016). *Herramientas para la evaluación de calidad en el e - learning*. 332–336.

Bevan, N., Carter, J., Earthy, J., Geis, T., & Harker, S. (2016). New ISO Standards for Usability, Usability Reports and Usability Measures [Inbook]. In M. Kurosu (Ed.), *Human-Computer Interaction. Theory, Design, Development and Practice : 18th International Conference, HCI International 2016, Toronto, ON, Canada, July 17-22, 2016. Proceedings, Part I* (pp. 268–278). Springer International Publishing. https://doi.org/10.1007/978-3-319-39510-4_25

Biasutti, M., & El-Deghaidy, H. (2012). Using Wiki in teacher education: Impact on

- knowledge management processes and student satisfaction [Article]. *Computers and Education*, 59(3), 861–872. <https://doi.org/10.1016/j.compedu.2012.04.009>
- Boley, H., Paschke, A., & Shafiq, O. (2010). RuleML 1.0: The overarching specification of web rules. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6403 LNCS, 162–178. <https://doi.org/10.1007/978-3-642-16289-3-15>
- Caffe. (2018). <http://caffe.berkeleyvision.org/>
- Carrizo, D., & Alfaro, A. (2018). Método de aseguramiento de la calidad en una metodología de desarrollo de software: Un enfoque práctico. *Ingeniare*, 26(1), 114–129. <https://doi.org/10.4067/S0718-33052018000100114>
- Chen, X. (2016). *Evaluating Language-learning Mobile Apps for Second-language Learners*. 9(2). <https://doi.org/10.18785/jetde.0902.03>
- Chin, J. P., Diehl, V. A., & Norman, K. L. (1988). Development of an instrument measuring user satisfaction of the human-computer interface. *Conference on Human Factors in Computing Systems - Proceedings, Part F1302*(January 2016), 213–218. <https://doi.org/10.1145/57167.57203>
- Chiu, M.-H., Lin, J.-W., & Chou, C.-C. (2016). Content Analysis of Conceptual Change Research and Practice in Science Education: From Localization to Globalization [Inbook]. In M.-H. Chiu (Ed.), *Science Education Research and Practices in Taiwan: Challenges and Opportunities* (pp. 89–131). Springer Singapore. https://doi.org/10.1007/978-981-287-472-6_5
- Christopher, A., Ghamari, M., & Nazeran, H. (2016). Development of Mobile Apps for Wireless Sensor Data Acquisition and Visualization of Biopotentials. *32nd Southern Biomedical Engineering Conference Development*, 5090. <https://doi.org/10.1109/SBEC.2016.23>
- Chu, S. K. W., Chan, C. K. K., & Tiwari, A. F. Y. (2012). Using blogs to support learning during internship [Article]. *Computers and Education*, 58(3), 989–1000. <https://doi.org/10.1016/j.compedu.2011.08.027>

- Chuang, Y. (2017). MEMIS: A Mobile-Supported English-Medium Instruction System. *Telematics and Informatics*, 34(2), 640–656. <https://doi.org/10.1016/j.tele.2016.10.007>
- Coad, P., Lefebvre, E., & Luca, J. D. (1999). *Java Modeling In Color With UML: Enterprise Components and Process* (L. A. H.-G. César Ricardo Alducin-Ruiz, Jorge Octavio Ocharán-Hernández (Ed.)) [JOUR].
- Collins, J. (n.d.). *What Is YouTube? How Do I Use It?* Retrieved May 20, 2018, from <https://www.lifewire.com/youtube-101-3481847>
- Colombo-Mendoza, L. O., Alor-Hernández, G., Rodríguez-gonzález, A., & Valencia-garcía, R. (2014). MobiCloUP!: a PaaS for cloud services-based mobile applications [Article]. *Automated Software Engineering*, 21(3), 391–437. <https://doi.org/10.1007/s10515-014-0143-5>
- Comercio Electrónico*. (2018). <https://azure.microsoft.com/es-mx/solutions/e-commerce/>
- Cortes-Camarillo, C. A., Alor-Hernández, G., Sánchez-Morales, L. N., Rosales-Morales, V. Y., Rodríguez-Mazahua, L., & Sánchez-Cervantes, J. L. (2019). EduGene: A UIDP-Based Educational App Generator for Multiple Devices and Platforms. *International Journal of Human-Computer Interaction*, 35(3), 274–296. <https://doi.org/10.1080/10447318.2018.1459347>
- Cortes-Camarillo, C. A., Rosales-Morales, V. Y., Sánchez-Morales, L. N., Alor-Hernández, G., & Rodríguez-Mazahua, L. (2017). Atila: A UIDPs-based educational application generator for mobile devices. *2017 International Conference on Electronics, Communications and Computers (CONIELECOMP)*, 5–11. <https://doi.org/10.1109/CONIELECOMP.2017.7891810>
- Costa, S. L. Da, Neto, V. V. G., & Oliveira, J. L. De. (2014). A user interface stereotype to build Web portals. *Proceedings - 9th Latin American Web Congress, LA-WEB 2014*, 10–18. <https://doi.org/10.1109/LAWeb.2014.8>
- Davids, M. R., Chikte, U., Grimmer-somers, K., & L Halperin, M. (2014). Usability testing of a multimedia e-learning resource for electrolyte and acid-base

- disorders. *British Journal of Educational Technology*, 45(2), 367–381. <https://doi.org/10.1111/bjet.12042>
- De la Rosa, R. (2007). Procesamiento de Imágenes Digitales. *X Congreso Nacional En Informática Y Computación Del Instituto Tecnológico de Puebla*.
- De Wet, J. (2019). *What is Deezer's business model?* <https://www.quora.com/What-is-Deezers-business-model>
- Deep Learning for Java*. (2018). <https://deeplearning4j.org/>
- Delgado, A., Estepa, A., Troyano, J. A., & Estepa, R. (2016). Reusing UI elements with Model-Based User Interface Development. *Journal of Human Computer Studies*, 86, 48–62. <https://doi.org/10.1016/j.ijhcs.2015.09.003>
- Delia, L., Thomas, P., Corbalan, L., Sosa, J. F., & Cuiti, A. (2019). Development Approaches for Mobile Applications : Comparative Analysis of Features. In A. K., K. S., & B. R. (Eds.), *Advances in Intelligent Systems and Computing* (Vol. 857, pp. 470–484). Springer, Cham.
- Developer, L. G. (2013). *LG Developer* [Misc]. <https://developer.lge.com/webOSTV/design/>
- Díez, R. P., Gómez, A. G., & de Abajo Martínez, N. (2001). *Introducción a la inteligencia artificial: sistemas expertos, redes neuronales artificiales y computación evolutiva* [BOOK]. Universidad de Oviedo.
- Digital Image Processing Using Matlab - Gonzalez Woods & Eddins.pdf*. (n.d.).
- Dwivedi, A. K., Tirkey, A., & Rath, S. K. (2016). Applying Software Metrics for the Mining of Design Pattern. *Uttar Pradesh Section International Conference on Electrical, Computer and Electronics Engineering (UPCON)*, 9–11.
- eBay. (2018). *eBay*. <https://play.google.com/store/apps/details?id=com.ebay.mobile&hl=en>
- El-Kassas, W. S., Abdullah, B. A., Yousef, A. H., & Wahba, A. M. (2016). Enhanced Code Conversion Approach for the Integrated Cross-Platform Mobile Development (ICPMD). *IEEE Transactions on Software Engineering*, 42(11), 1036–1053. <https://doi.org/10.1109/TSE.2016.2543223>
- Enciclopedia, W. (2018). *Wiki Enciclopedia*.

- <https://play.google.com/store/apps/details?id=uk.co.appsunlimited.wikiapp>
- Flórez, R., & Fernández, J. M. F. (2008). Las redes neuronales artificiales: Fundamentos teóricos y aplicaciones prácticas [JOUR]. *España, Netbiblo*.
- Francese, R., Risi, M., Tortora, G., & Tucci, M. (2016). Visual Mobile Computing for Mobile End-Users. *IEEE Transactions on Mobile Computing*, *15*(4), 1033–1046. <https://doi.org/10.1109/TMC.2015.2422295>
- Freitas, F., & Maia, P. H. M. (2015). Just Business: A Framework for Developing Android Applications Using Naked Objects. *Proceedings - 2015 9th Brazilian Symposium on Software Components, Architectures and Reuse, SBCARS 2015*, 11–20. <https://doi.org/10.1109/SBCARS.2015.12>
- Garcés, L., Ampatzoglou, A., Avgeriou, P., & Nakagawa, E. Y. (2017). Quality attributes and quality models for ambient assisted living software systems: A systematic mapping [Article]. *Information and Software Technology*, *82*, 121–138. <https://doi.org/10.1016/j.infsof.2016.10.005>
- Garcia-Cabot, A., De-Marcos, L., & Garcia-Lopez, E. (2015). An empirical study on m-learning adaptation: Learning performance and learning contexts [Article]. *Computers & Education*, *82*, 450–459. <https://doi.org/10.1016/j.compedu.2014.12.007>
- Gil, P. (n.d.). *What Is Twitter & How Does It Work?* Retrieved May 20, 2018, from ?, <https://www.lifewire.com/what-exactly-is-twitter-2483331>.
- Grønli, T.-M., Ghinea, G., & Younas, M. (2014). Context-aware and automatic configuration of mobile devices in cloud-enabled ubiquitous computing [Article]. *Personal and Ubiquitous Computing*, *18*(4), 883–894. <https://doi.org/10.1007/s00779-013-0698-3>
- Gunelius, S. (n.d.). *Is Tumblr the Right Blogging Tool for You?* Retrieved May 20, 2018, from <https://www.lifewire.com/tumblr-overview-for-bloggers-3476387>.
- Gutiérrez, A. I. B., Díaz, S. M., & Torres, J. L. G. (2012). *VISIÓN ESTEREOSCÓPICA POR COMPUTADORA CON MATLAB Y OPENCV* [JOUR].
- Halbe, A., & Joshi, A. R. (2015). A novel approach to HTML page creation using

- neural network [JOUR]. *Procedia Computer Science*, 45, 197–204. <https://doi.org/10.1016/j.procs.2015.03.122>
- Hamidi, H., & Chavoshi, A. (2017). Learning in Higher Education : A Case Study of Students of the Analysis of the Essential Factors for the Adoption of Mobile Learning in Higher Education : A Case Study of Students of the University of Technology. *Telematics and Informatics*. <https://doi.org/10.1016/j.tele.2017.09.016>
- Harrati, N., Bouchrika, I., Tari, A., & Ladjailia, A. (2016). Exploring user satisfaction for e-learning systems via usage-based metrics and system usability scale analysis [Article]. *Computers in Human Behavior*, 61, 463–471. <https://doi.org/10.1016/j.chb.2016.03.051>
- Hasselbring, W., & Steinacker, G. (2017). Microservice architectures for scalability, agility and reliability in e-commerce. *Proceedings - 2017 IEEE International Conference on Software Architecture Workshops, ICSAW 2017: Side Track Proceedings*, 243–246. <https://doi.org/10.1109/ICSAW.2017.11>
- Hutchison, A., & Colwell, J. (2012). Using a wiki to facilitate an online professional learning community for induction and mentoring teachers [Article]. *Education and Information Technologies*, 17(3), 273–289. <https://doi.org/10.1007/s10639-011-9159-7>
- Ileana, D., & Cabrera, M. (2018). Framerwork de desarrollo de Proyectos Sociotecnológicos basado en la notación de Metamodelos de procesos de Ingeniería de Software (spem 2.0). *Revista Arbitrada Interdisciplinaria de Ciencias de La Educación, Turismo, Ciencias Sociales Y Económica, Ciencias Del Agro Y Mar Y Ciencias Exactas Y Aplicadas*, 3(15), 180–193.
- Indriana, M., & Adzani, M. L. (2017). UI/UX analysis & design for mobile e-commerce application prototype on Gramedia.com. *Proceedings of 2017 4th International Conference on New Media Studies, CONMEDIA 2017, 2018–Janua*, 170–173. <https://doi.org/10.1109/CONMEDIA.2017.8266051>
- Isa, M., Wan, W. A. R., Lokman, A. M., Adnan, M., Ridwan, M., Sah, M., Noratika, I., Hamdan, A. R., & Luaran, J. E. (2016). Factors of using m-learning in blended

- learning environment: A qualitative inquiry [JOUR]. *Advanced Science Letters*, 22(10), 3155–3158.
- Islam, M. D., Mobarak, H., & Islam, M. D. (2018). *Voice command based android java code generator* [BRAC University]. <http://dSPACE.bracu.ac.bd/xmlui/handle/10361/10848>
- ISO/IEC, 25000. (2015). *The ISO/IEC 25000 series of standardss* [Misc]. <http://iso25000.com/index.php/en/iso-25000-standards/>
- ISO/IEC 14598, I. 14598. (1998). *Information Technology - Evaluation of Software Products*. ISO/IEC.
- ISO/IEC 9126. (2001). *Software engineering – Product quality*. ISO/IEC.
- Jagtap, S. S., & Hanchate, D. B. (2017). Development of Android Based Mobile App for PrestaShop eCommerce Shopping Cart (ALC). *International Research Journal of Engineering and Technology(IRJET)*, 4(7), 2248–2254. <https://irjet.net/archives/V4/i7/IRJET-V4I7460.pdf>
- Jareño, A., & Martínez, F. (2016). *Design and validation of an instrument to evaluate educational apps and creation of a digital repository*.
- Ji, Y. G., Park, J. H., Lee, C., & Yun, M. H. (2010). A Usability Checklist for the Usability Evaluation of Mobile Phone User Interface. *International Journal of Human-Computer Interaction*, 20(September 2011), 37–41. <https://doi.org/10.1207/s15327590ijhc2003>
- Joo, Y. J., Lee, H. W., & Ham, Y. (2014). Integrating user interface and personal innovativeness into the TAM for mobile learning in Cyber University [Article]. *Journal of Computing in Higher Education*, 26(2), 143–158. <https://doi.org/10.1007/s12528-014-9081-2>
- Kitchenham, B. A. (1996). Evaluating Software Engineering Methods and Tool Part 1 : The Evaluation Context and Evaluation Methods. *ACM SIGSOFT Software Engineering Notes*, 21(1), 11–14. <https://doi.org/https://doi.org/10.1145/381790.381795>
- Kiwix, W. O. (2018). *Kiwix, Wikipedia offline*. <http://www.kiwix.org/es/>
- Knežević, B., & Delić, M. (2017). Young consumers' perception of problems and

- usefulness of mobile shopping applications. *Entrepreneurial Business and Economics Review*, 5(1), 43–58. <https://doi.org/10.15678/EBER.2017.050103>
- Koc, G., & Aydos, M. (2017). Trustworthy scrum: Development of secure software with scrum. *2017 International Conference on Computer Science and Engineering (UBMK)*, 244–249. <https://doi.org/10.1109/ubmk.2017.8093383>
- Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>
- Lubiano, M. A., Salas, A., De Saa, S. de da R., Montenegro, M., & Gil, M. Á. (2017). An empirical analysis of the coherence between fuzzy rating scale-and likert scale-based responses to questionnaires. *Advances in Intelligent Systems and Computing*, 456, 329–337. https://doi.org/10.1007/978-3-319-42972-4_41
- Majchrzak, T. A., Ernsting, J., & Kuchen, H. (2015). Model-Driven cross-platform apps: Towards business practicability. *CEUR Workshop Proceedings*, 1367(2), 129–136.
- Malganova, I., & Rahkimova, A. (2015). The Usage of Distance Education Practice of the World Leading Universities in Russian Multiethnic Region [Article]. *Procedia - Social and Behavioral Sciences*, 191, 2622–2625. <https://doi.org/http://dx.doi.org/10.1016/j.sbspro.2015.04.392>
- Marin, I., Ortin, F., Pedrosa, G., & Rodriguez, J. (2015). Generating native user interfaces for multiple devices by means of model transformation [Article]. *Frontiers of Information Technology {&} Electronic Engineering*, 16(12), 995–1017. <https://doi.org/10.1631/FITEE.1500083>
- McGill, T. J., Klobas, J. E., & Renzi, S. (2014). Critical success factors for the continuation of e-learning initiatives [Article]. *Internet and Higher Education*, 22, 24–36. <https://doi.org/10.1016/j.iheduc.2014.04.001>
- Mejía-figueroa, A., Quezada-Cisnero, M. D. L. Á., & Juárez-Ramírez, J. . R. (2016). Developing Usable Software Applications for Users with Autism : User Analisis , User Interface Design Patterns and Interface Components. *IEEE International Conference in Software Engineering Research and Innovation*, 196–204. <https://doi.org/10.1109/CONISOFT.2016.36>

- Mercado-Libre. (2020). *Mercado Libre: Encuentra tus marcas favoritas*.
<https://play.google.com/store/apps/details?id=com.mercadolibre>
- Mills, J., Turin, R., Mangas, J., & Ranville, S. (2018). Innovation Welcome: An Agile Approach to Model-Based Development of Safety-Critical Embedded Systems. *2018 IEEE International Conference on Software Architecture Companion (ICSA-C)*, 152–152. <https://doi.org/10.1109/icsa-c.2018.00045>
- Mohammadi, H. (2015). Investigating users' perspectives on e-learning: An integration of TAM and IS success model [Article]. *Computers in Human Behavior*, 45, 359–374. <https://doi.org/10.1016/j.chb.2014.07.044>
- Molina Ríos, J. R., Zea Ordóñez, M. P., Contento Segarra, M. J., & García Zerda, F. G. (2017). Estado Del Arte: Metodologías De Desarrollo En Aplicaciones Web. *3C Tecnología_Glosas de Innovación Aplicadas a La Pyme*, 6(3), 54–71. <https://doi.org/10.17993/3ctecno.2017.v6n3e23.54-71>
- Montero Molina, B., Cevallos Vite, H., & Dávila Cuesta, J. (2018). Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software. *Espirales Revista Multidisciplinaria de Investigación*, 2(17), 114–121. <https://doi.org/https://doi.org/10.1007/BF03194495>
- Moodle, L. M. S. (2017). *Moodle - Open-source learning platform | Moodle.org* [Misc]. <https://moodle.org/>
- Moreau, E. (n.d.). *What Is Instagram, Anyway?* Retrieved May 20, 2018, from <https://www.lifewire.com/what-is-instagram-3486316>.
- Morgado, I. C., & Paiva, A. C. R. (2016). The iMPAcT tool: Testing UI patterns on mobile applications. *Proceedings - 2015 30th IEEE/ACM International Conference on Automated Software Engineering, ASE 2015*, 876–881. <https://doi.org/10.1109/ASE.2015.96>
- Narkawicz, A., Munoz, C. A., & Dutle., A. M. (2017). The MINERVA software development process. *9th NASA Formal Methods Symposium (NFM)*, 1–8. <https://ntrs.nasa.gov/search.jsp?R=20170005461>
- Nations, D. (n.d.). *What Is Facebook?* Retrieved May 20, 2018, from <https://www.lifewire.com/what-is-facebook-3486391>

- Nedzved, A., Gurevich, I., Trusova, Y., & Ablameyko, S. (2013). Software development technology with automatic configuration to classes of image processing problems [JOUR]. *Pattern Recognition and Image Analysis*, 23(2), 269–277.
- Neil, T. (2014). *Mobile design pattern gallery: UI patterns for smartphone apps* [Book]. “O’Reilly Media, Inc.”
- Norman, G. (2010). Likert scales, levels of measurement and the “laws” of statistics [Article]. *Advances in Health Sciences Education*, 15(5), 625–632. <https://doi.org/10.1007/s10459-010-9222-y>
- Núñez-Valdez, E. R., García-Díaz, V., Cueva Lovelle, J. M., Sáez Achaerandio, Y., & González-Crespo, R. (2016). A model-driven approach to generate and deploy videogames on multiple platforms. *Journal of Ambient Intelligence and Humanized Computing*, 1–13. <https://doi.org/10.1007/s12652-016-0404-1>
- Oyarzo, F., Herrera, F., & Casas, S. (2014). API TVD, a wizard for interactive applications for Digital TV [CONF]. *Computing Conference (CLEI), 2014 XL Latin American*, 1–8.
- Palmer, S. R., & Felsing, M. (2001). *A practical guide to feature-driven development* [BOOK]. Pearson Education.
- Panunzio, M., & Vardanega, T. (2014). The Journal of Systems and Software A component-based process with separation of concerns for the development of embedded real-time software systems. *The Journal of Systems & Software*, 96, 105–121. <https://doi.org/10.1016/j.jss.2014.05.076>
- Porter, W. W., Graham, C. R., Bodily, R. G., & Sandberg, D. S. (2016). A qualitative analysis of institutional drivers and barriers to blended learning adoption in higher education [Misc]. In *The Internet and Higher Education* (Vol. 28, pp. 17–27). <https://doi.org/10.1016/j.iheduc.2015.08.003>
- Porter, W. W., Graham, C. R., Spring, K. A., & Welch, K. R. (2014). Blended learning in higher education: Institutional adoption and implementation [Article]. *Computers and Education*, 75, 185–195. <https://doi.org/10.1016/j.compedu.2014.02.011>

- Poulova, P., & Simonova, I. (2014). E-learning Reflected in Research Studies in Czech Republic: Comparative Analyses [Article]. *Procedia - Social and Behavioral Sciences*, 116, 1298–1304. <https://doi.org/10.1016/j.sbspro.2014.01.386>
- Queirós, R., Leal, J. P., & Paiva, J. C. (2016). Integrating Rich Learning Applications in LMS [Inbook]. In Y. Li, M. Chang, M. Kravcik, E. Popescu, R. Huang, Kinshuk, & N.-S. Chen (Eds.), *State-of-the-Art and Future Directions of Smart Learning* (pp. 381–386). Springer Singapore. https://doi.org/10.1007/978-981-287-868-7_46
- Reda, B., Philippe, D., & Marie-Pierre, G. (2005). MDA Components: A flexible way for implementing the MDA approach. In H. Alan & K. David (Eds.), *Model Driven Architecture - Foundations and Applications: First European Conference, ECMDA-FA 2005* (ECMDA-FA 2, pp. 59–73). Springer Berlin Heidelberg.
- Rieger, C. (2017). Business apps with MAML: A model-driven approach to process-oriented mobile app development. *Proceedings of the ACM Symposium on Applied Computing, Part F1280*, 1599–1606. <https://doi.org/10.1145/3019612.3019746>
- Rogai, D., Bisconti, C., & Faye, S. O. (2015). Content-Based Multi-platform App Forge. *Proceedings - 2nd ACM International Conference on Mobile Software Engineering and Systems, MOBILESoft 2015*, 166–167. <https://doi.org/10.1109/MobileSoft.2015.47>
- Rosales-Morales, V. Y., Sánchez-Morales, L. N., Alor-Hernández, G., Garcia-Alcaraz, J. L., Sánchez-Cervantes, J. L., & Rodriguez-Mazahua, L. (2019). ImagingDev: A New Approach for Developing Automatic Cross-Platform Mobile Applications Using Image Processing Techniques. *The Computer Journal*. <https://doi.org/10.1093/comjnl/bxz029>
- Saetang, S. (2017). The E-Commerce strategies responding to the UX design. *Ubi-Media 2017 - Proceedings of the 10th International Conference on Ubi-Media Computing and Workshops with the 4th International Workshop on Advanced E-Learning and the 1st International Workshop on Multimedia and IoT:*

- Networks, Systems and Applications*, 1–6.
<https://doi.org/10.1109/UMEDIA.2017.8074086>
- Sakai, L. M. S. (2014). *Introducing Sakai 11 | Sakai* [Misc]. <https://sakaiproject.org/>
- Salahli, M. A., Yildirim, E., Gasimzadeh, T., Alasgarova, F., & Guliyev, A. (2018). One mobile application for the development of programming skills of secondary school students. *Procedia Computer Science*, 120, 502–508.
<https://doi.org/10.1016/j.procs.2017.11.271>
- Sánchez, W. (2011). La usabilidad en Ingeniería de Software: definición y características. *Ing-Novación. Reporte de Investigación*, 2, 7–21.
<http://www.redicces.org.sv/jspui/bitstream/10972/1937/1/2>. La usabilidad en Ingeniería de Software- definicion y características.pdf
- Sasikumar, M. (2008). Automatic Generation of Speech Interface for GUI Tools / Applications using Accessibility Framework. *Techshare India*, 1–7.
<http://naveenk.wdfiles.com/local--files/publications/AccessibleSpeech.pdf>
- Schmidt, D. C. (2006). Model-driven engineering [JOUR]. *COMPUTER-IEEE COMPUTER SOCIETY-*, 39(2), 25.
<https://doi.org/http://doi.org/10.1109/MC.2006.58>
- Scott, Mackenzie, I. (2012). *Human Computer Interaction: An Empirical Research Perspective*. Newnes.
- Sheibley, M. (2013). *Mobile Patterns* [Misc]. <http://www.mobile-patterns.com/>
- Shoukry, L., Sturm, C., & Galal-edeen, G. H. (2015). *Pre-MEGa: A Proposed Framework for the Design and Evaluation of Preschoolers' Mobile Educational Games*. <https://doi.org/10.1007/978-3-319-06773-5>
- Soad, G. W., Barbosa, E. F., & Filho, N. F. D. (2016). Quality Evaluation of Mobile Learning Applications. *Frontiers in Education Conference (FIE)*, 1–8.
<https://doi.org/http://doi.org/10.1109/FIE.2016.7757540>
- Sohaib, O., & Naderpour, M. (2017). Decision making on adoption of cloud computing in e-commerce using fuzzy TOPSIS. *IEEE International Conference on Fuzzy Systems*, 0–5. <https://doi.org/10.1109/FUZZ-IEEE.2017.8015404>
- Srihari, S. N., & Franke, K. (2008). Computational Forensics: Second International

- Workshop [Proceedings]. *IWCF 2008*, 5158.
- Stephens, R. (2015). *Beginning Software Engineering* (J. W. & Sons (Ed.); reimpresia).
- Tensor Flow*. (2018). <https://www.tensorflow.org/?hl=es>
- Tess, P. A. (2013). The role of social media in higher education classes (real and virtual)-A literature review [Misc]. In *Computers in Human Behavior* (Vol. 29, Issue 5). <https://doi.org/10.1016/j.chb.2012.12.032>
- Thanh-diane, N., Vanderdonckt, J., & Seffah, A. (2016). UIPLML : Pattern-based Engineering of User Interfaces of Multi-Platform Systems. *2016 IEEE Tenth International Conference on Research Challenges in Information Science (RCIS)*, 1–12. <https://ieeexplore.ieee.org/abstract/document/7549348>
- The Mathworks Inc. (2016). *MATLAB - MathWorks*. [Www.mathworks.com/products/matlab](http://www.mathworks.com/products/matlab).
<http://www.mathworks.com/products/matlab/>
- The Mathworks Inc. (2018). *Deep Learning (Aprendizaje Profundo) Tres cosas que es necesario saber*. Deep Learning. <https://la.mathworks.com/discovery/deep-learning.html?requestedDomain=>
- Theano at a Glance*. (2018). <http://deeplearning.net/software/theano/introduction.html>
- Thomas, P., Delia, L., Corbalan, L., Cáseres, G., Cuitiño, A., Sosa, J. F., & Pesado, P. (2017). Análisis de Enfoques de Aplicaciones para Dispositivos Móviles. *Workshop de Investigadores En Ciencias de La Computación*, 428–431.
- Tidwell, J. (2010). *Designing interfaces: Patterns for effective interaction design*. “O’Reilly Media, Inc.”
- Toxboe, A. (2016). *UI Patterns User Interface Design pattern Library*. <http://ui-patterns.com/>
- Truyen, F. (2006). The fast guide to model driven architecture the basics of model driven architecture [JOUR]. *Cephas Consulting Corp*.
- Tsikerdekis, M., & Zeadally, S. (2014). Online deception in social media [JOUR]. *Communications of the ACM*, 57(9), 72–80.

- TV, A. (2015). *Android TV Patterns* [Misc]. <http://www.google.com/design/spec-tv/patterns>
- TV, A. (2016). *Human Interface Guidelines* [Misc]. <https://developer.apple.com/tvos/human-interface-guidelines/ui-elements/>
- Twining, P., Heller, R. S., Nussbaum, M., & Tsai, C.-C. (2017). Some guidance on conducting and reporting qualitative studies [Article]. *Computers {&} Education*, 106, A1-- A9. <https://doi.org/http://dx.doi.org/10.1016/j.compedu.2016.12.002>
- Umuhaza, E., Ed-Douibi, H., Brambilla, M., Cabot, J., & Bongio, A. (2015). Automatic code generation for cross-platform, multi-device mobile apps: Some reflections from an industrial experience. *MobileDeLi 2015 - Proceedings of the 3rd International Workshop on Mobile Development Lifecycle, October*, 37–44. <https://doi.org/10.1145/2846661.2846666>
- UNITiD. (2016). *Android Patterns* [Misc]. <http://unitid.nl/androidpatterns/>
- Vásquez-Ramírez, R., Alor-Hernández, G., & Rodríguez-González, A. (2014). Athena: A Hybrid Management System for Multi-device Educational Content [Article]. *Comput. Appl. Eng. Educ.*, 22(4), 750–763. <https://doi.org/10.1002/cae.21567>
- Von Thienen, J., Clancey, W., & Meinel, C. (2019). InnoDev: a software development methodology integrating design thinking, scrum and lean startup. In M. C. & L. L. (Eds.), *Design Thinking Research* (pp. 199–227). Springer Nature Switzerland. <https://doi.org/10.1007/978-3-319-97082-0>
- Watson, S. L., Loizzo, J., Watson, W. R., Mueller, C., Lim, J., & Ertmer, P. A. (2016). Instructional design, facilitation, and perceived learning outcomes: an exploratory case study of a human trafficking MOOC for attitudinal change [Article]. *Educational Technology Research and Development*, 64(6), 1273–1300. <https://doi.org/10.1007/s11423-016-9457-2>
- Wesson, J. L., Cowley, N. L. O., & Brooks, C. E. (2017). Extending a mobile prototyping tool to support user interface design patterns and reusability. *Proceedings of the South African Institute of Computer Scientists and Information Technologists on - SAICSIT '17*, 1–9.

<https://doi.org/10.1145/3129416.3129444>

Willings, A. (2020). *What is Spotify and how does it work?* <https://www.pocket-lint.com/apps/news/spotify/139236-what-is-spotify-and-how-does-it-work>.

Wirfs-Brock, R., & McKean, A. (2003). *Object design: roles, responsibilities, and collaborations* [BOOK]. Addison-Wesley Professional.

Wirfs-Brock, R., & Wilkerson, B. (1989). Object-oriented design: a responsibility-driven approach [CONF]. *ACM SIGPLAN Notices*, 24(10), 71–75.

Wish. (2018). *Wish*.
<https://play.google.com/store/apps/details?id=com.contextlogic.wish&hl=en>

Zhan, Z., & Mei, H. (2013). Academic self-concept and social presence in face-to-face and online learning: Perceptions and effects on students' learning achievement and satisfaction across environments [Article]. *Computers and Education*, 69, 131–138. <https://doi.org/10.1016/j.compedu.2013.07.002>