

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

OPCIÓN I.- TESIS

TRABAJO PROFESIONAL

“Uso de patrones de diseño de interfaz de usuario para la generación de aplicaciones educativas multi-dispositivo a partir del procesamiento de imágenes”.

QUE PARA OBTENER EL GRADO DE:

Maestro en Sistemas Computacionales

PRESENTA:

I.S.C. Cesar Augusto Cortes Camarillo

DIRECTOR DE TESIS:

Dr. Giner Alor Hernández





"Año del Centenario de la Promulgación de la Constitución Política de los Estados Unidos Mexicanos"

FECHA: 22/11/2017
DEPENDENCIA: POSGRADO
ASUNTO: Autorización de Impresión
OPCIÓN: I

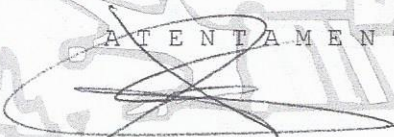
C. CESAR AUGUSTO CORTES CAMARILLO
CANDIDATO A GRADO DE MAESTRO EN:
SISTEMAS COMPUTACIONALES

De acuerdo con el Reglamento de Titulación vigente de los Centros de Enseñanza Técnica Superior, dependiente de la Dirección General de Institutos Tecnológicos de la Secretaría de Educación Pública y habiendo cumplido con todas las indicaciones que la Comisión Revisora le hizo respecto a su Trabajo Profesional titulado:

**"USO DE PATRONES DE DISEÑO DE INTERFAZ DE USUARIO PARA LA
GENERACION DE APLICACIONES EDUCATIVAS MULTI-DISPOSITIVO A PARTIR
DEL PROCESAMIENTO DE IMAGENES".**

Comunico a Usted que este Departamento concede su autorización para que proceda a la impresión del mismo.

ATENTAMENTE


RUBEN POSADA GOMEZ
JEFE DE LA DIV. DE ESTUDIOS DE POSGRADO

C.A. TITULACIÓN



SECRETARIA DE
EDUCACIÓN PÚBLICA
INSTITUTO
TECNOLÓGICO
DE ORIZABA

ggc





"Año del Centenario de la Promulgación de la Constitución Política de los Estados Unidos Mexicanos"

FECHA : 04/10/2017

ASUNTO: Revisión de Trabajo Escrito

C. RUBEN POSADA GOMEZ
JEFE DE LA DIVISION DE ESTUDIOS
DE POSGRADO E INVESTIGACION.
P R E S E N T E

Los que suscriben, miembros del jurado, han realizado la revisión de la Tesis del (la) C. :

CESAR AUGUSTO CORTES CAMARILLO

la cual lleva el título de:

"USO DE PATRONES DE DISEÑO DE INTERFAZ DE USUARIO PARA LA GENERACION DE APLICACIONES EDUCATIVAS MULTI-DISPOSITIVO A PARTIR DEL PROCESAMIENTO DE IMAGENES".

Y concluyen que se acepta.

A T E N T A M E N T E

PRESIDENTE : DR. GINER ALOR HERNANDEZ


FIRMA

SECRETARIO : MCE BEATRIZ ALEJANDRA OLIVARES


FIRMA

VOCAL : DRA. LISBETH RODRIGUEZ MAZAHUA


FIRMA

VOCAL SUP. : M.C. SILVESTRE GUSTAVO SERGIO PELAEZ


FIRMA

EGRESADO (A) DE LA MAESTRIA EN SISTEMAS COMPUTACIONALES

OPCION: I Tesis



Agradecimientos

A Dios por darme la oportunidad de vivir y por ayudarme todos los días, porque con él todo ha sido posible.

A mi madre María Estela Camarillo Ascención por apoyarme en todo, por ser un gran ejemplo a seguir y por todos sus sacrificios que ha realizado para que alcance mis metas.

A mi novia Josefina Zavala de la Trinidad por comprenderme y por todo el tiempo que ha estado a mi lado en todas las adversidades.

A mi Director de Tesis el Dr. Giner Alor Hernández por compartir su conocimiento y apoyarme incondicionalmente para el desarrollo de este proyecto.

A mis hermanas y sobrinos por ofrecerme su apoyo por ser una gran fuente de mis alegrías.

A mis amigos por todos aquellos momentos divertidos que pasamos juntos y porque a pesar de la distancia los siento cerca de mí.

A todos los maestros que compartieron sus conocimientos para hacer de mi aprendizaje más valioso y satisfactorio.

Índice general

Índice general	2
Índice de tablas	6
Índice de figuras	7
Índice de listados	8
Resumen	9
Abstract	10
Introducción	11
Capítulo 1. Antecedentes	12
1.1. Marco teórico.....	12
1.1.1. Procesamiento digital de imágenes	12
1.1.2. Software	12
1.1.3. Aplicaciones Multi-dispositivo	15
1.1.4. Aplicaciones Multiplataforma.....	16
1.1.5. Patrones de diseño de interfaz de usuario	18
1.2. Planteamiento del problema	20
1.3. Situación de la empresa	21
1.4. Objetivo general y específico	21
1.4.1. Objetivo general	21
1.4.2. Objetivos específicos.....	21
1.5. Justificación	22
Capítulo 2. Estado de la práctica.....	24
2.1. Trabajos relacionados	24
2.1.1. Desarrollo de aplicaciones para móviles	24

2.1.2.	Patrones de diseño de interfaz de usuario en el desarrollo de software	27
2.2.	Análisis comparativo	28
2.3.	Propuesta de solución	33
2.3.1.	Análisis de las tecnologías de información (TI).....	33
Capítulo 3.	Aplicación de la metodología.....	37
3.1.	Introducción.....	37
3.2.	Arquitectura y funcionalidad del componente generador de código.....	38
3.2.1.	Descripción de la arquitectura.....	39
3.2.2.	Componentes de software utilizados en EduGene	43
3.2.3.	Descripción del flujo de trabajo usado en la arquitectura	46
3.3.	Identificación de actividades de aprendizaje en interfaces gráficas de usuario	47
3.4.	Patrones de interfaz de usuario soportados.....	49
3.4.1.	Clasificación de patrones de diseño de interfaz de usuario.....	50
3.4.2.	Patrones de diseño individuales de interfaz de usuario.....	55
3.4.3.	Patrones de diseño compuestos de interfaz de usuario	58
3.5.	Proceso de identificación de patrones de diseño de interfaz de usuario.....	59
3.5.1.	Iconografía de los patrones de diseño de interfaz de usuario.....	60
3.5.2.	Reconocimiento de patrones de diseño en imágenes usando un proceso ideal... 64	
3.5.3.	Reconocimiento de patrones de diseño en imágenes usando un proceso no ideal	
	92	
3.6.	Generador de aplicaciones.....	92
3.6.1.	Descripción del generador de aplicaciones	92
3.6.2.	Archivo basado en XML	92
3.6.3.	Estructura de carpetas de los proyectos generados	93

3.6.4. Componentes.....	96
Capítulo 4. Resultados	101
4.1. Caso de estudio 1: Generación de MOOC utilizando un archivo XML.....	101
4.1.1. Diseño de la aplicación	101
4.1.2. Construyendo la aplicación	102
4.1.3. Probando la aplicación	104
4.2. Caso de estudio 2: Generación de un MOOC a través de una interfaz gráfica de usuario	106
4.2.1. Diseño de la aplicación	106
4.2.2. Construyendo la aplicación	107
4.2.3. Probando la aplicación	112
4.3. Caso de estudio 3: Creación de un MOOC por medio de una imagen.....	113
4.3.1. Diseño de la aplicación	114
4.3.2. Construyendo la aplicación	114
4.3.3. Probando la aplicación	116
4.4. Evaluación	117
4.4.1. Evaluación cualitativa	118
4.4.2. Evaluación cuantitativa	123
Capítulo 5. Conclusiones y recomendaciones.....	128
5.1. Conclusiones.....	128
5.2. Recomendaciones	128
Productos académicos	130
Artículos JCR	130
Artículos en otros índices	130

Conferencias internacionales.....	130
Capítulos de libro	131
Registros de derechos de autor.....	131
Proyectos de Investigación.....	131
Estancias Académicas	131
Terminología involucrada	132
Bibliografía.....	133

Índice de tablas

Tabla 2.1 Comparación de enfoques para el desarrollo de aplicaciones móviles.	30
Tabla 2.2 Análisis comparativo de las aplicaciones que utilizan patrones de diseño de interfaz de usuario	32
Tabla 2.3 Tabla comparativa de bibliotecas para el procesamiento de imágenes.....	34
Tabla 3.1 Descripción y beneficios de las aplicaciones educativas	48
Tabla 3.2 Categorías de patrones de diseño de interfaz de usuario presentado por autor.....	49
Tabla 3.3 Relación entre patrones de diseño de interfaz de usuario	50
Tabla 3.4 Contextos de usos detectados al utilizar los patrones	52
Tabla 3.5 Relación entre los patrones de diseño de IU y las aplicaciones educativas	53
Tabla 3.6 Relación de patrones de diseño de interfaz de usuario con las posiciones	56
Tabla 3.7 Patrones de diseño compuestos de interfaces de usuario.....	58
Tabla 3.8 Tabla de la iconografía de los patrones individuales de diseño de interfaz de usuario	60
Tabla 3.9 Tabla de la iconografía de los patrones compuestos de diseño de interfaz de usuario	63
Tabla 3.10 Conjunto de reglas en RuleML para los patrones individuales de diseño de interfaz de usuario	65
Tabla 3.11 Conjunto de reglas en RuleML para los patrones individuales para patrones compuestos de diseño de interfaz de usuario	71
Tabla 3.12 Conjunto de diagramas de árbol para los patrones individuales de diseño de interfaz de usuario	78
Tabla 3.13 Conjunto de diagramas de árbol para los patrones individuales para patrones compuestos de diseño de interfaz de usuario	85
Tabla 4.1 Resultados de la evaluación cualitativa.....	121
Tabla 4.2 Medición del aspecto educativo	125
Tabla 4.3 Medición del aspecto de efectividad.....	126
Tabla 4.4 Medición del aspecto del soporte.....	126

Índice de figuras

Figura 3.1 Metodología dividida en etapas y actividades	37
Figura 3.2 Arquitectura en capas de EduGene	41
Figura 3.3 Flujo de trabajo de la arquitectura	47
Figura 3.4 Vista de elementos multi-dispositivo.....	55
Figura 3.5 Plantillas generadas a partir de las cinco posiciones	56
Figura 3.6 Estructura del archivo XML	93
Figura 3.7 Estructura de las carpetas.....	94
Figura 3.8 Estructura de carpetas para los proyectos de las aplicaciones de Android™	95
Figura 3.9 Estructura de carpetas para los proyectos de las aplicaciones de macOS®	95
Figura 3.10 Estructura de carpetas para los proyectos de las aplicaciones de Windows Phone®	96
Figura 3.11 Página de inicio de las funcionalidades	96
Figura 3.12 Leer texto basado en XML	97
Figura 3.13 Añadir componente de un archivo ZIP	98
Figura 3.14 Administrar componentes	98
Figura 3.15 Propiedades de un patrón de diseño de IU.....	99
Figura 3.16 Configuraciones de un patrón de diseño de IU.....	100
Figura 3.17 Configuraciones para cada patrón de diseño	100
Figura 4.1 La interfaz gráfica del generador de aplicaciones para una aplicación en macOS® y Windows Phone®.....	104
Figura 4.2 Caso de estudio 1: Ejecución de una aplicación educativa con Visual Studio: a) SplashScreen; b) Inicio de sesión; c) Dashboard; d) Menú; e) Lista; f) Video.....	105
Figura 4.3 Caso de estudio 1: Ejecución de la aplicación educativa mediante Xcode: a) SplashScreen; b) Inicio de sesión; c) Dashboard; d) Menú; e) Lista; f) Video.....	105
Figura 4.4 Plataforma EduGene	107
Figura 4.5 Desarrollo de aplicaciones educativas con EduGene	108
Figura 4.6 Edición de contenido de aplicaciones educativas	109
Figura 4.7 Caso de estudio 2: Ejecución de una aplicación educativa con Visual Studio: a) SplashScreen; B) Inicio de sesión; C) Dashboard; D) Menú; E) Lista; F) Video.....	112

Figura 4.8 Caso de estudio 2: Ejecución de la aplicación educativa mediante Xcode: a) SplashScreen; B) Inicio de sesión; C) Dashboard; D) Menú; E) Lista; F) Video.....	113
Figura 4.9 Caso de estudio 3: Bosquejos de los patrones de diseño de interfaz de usuario....	115
Figura 4.10 Caso de estudio 3: Carga de imagen de los patrones de diseño de interfaz de usuario	115
Figura 4.11 Caso de estudio 3: Patrones de diseño de interfaz de usuario identificados	116
Figura 4.12 Caso de estudio 3: Ejecución de una aplicación educativa en un dispositivo Android™: a) SplashScreen; B) Inicio de sesión; C) Dashboard; D) Carrusel y Lista; E) Patrón de video	117
Figura 4.13 Resultados de la evaluación cualitativa	122
Figura 4.14 Resultados de la evaluación cuantitativa	127

Índice de listados

Listado 4.1 Caso de estudio 1: Elementos principales de la etiqueta deploySettings en un archivo XML	102
Listado 4.2 Caso de estudio 1: Ejemplo de una capa	103
Listado 4.3 Caso de estudio 2: Información relevante de la aplicación en el archivo XML ..	109
Listado 4.4 Caso de estudio 2: Elementos principales de la etiqueta deploySettings en un archivo XML	110
Listado 4.5 Caso de estudio 2: Contenido y navegación de la aplicación en el archivo XML	111

Resumen

Resumen

La educación es clave para una sociedad justa y progresista, se proporciona especialmente por las escuelas y se refleja en diversas áreas de nuestras vidas. La importancia de la educación conduce a nuevos esquemas en el desarrollo de aplicaciones educativas. Sin embargo, desarrollar aplicaciones que proporcionen un diseño de interfaz de usuario eficiente y agradable, y además permita al usuario familiarizarse fácilmente con el software es una tarea complicada, porque generalmente es necesario tener buenos conocimientos de diseño y experiencia de interfaz de usuario para lograrlo. En la actualidad, el desarrollo de software se enfoca en disminuir el tiempo y el costo de la creación del software, sin afectar la calidad del mismo.

Como objetivo principal del proyecto de tesis, se propone desarrollar un componente de software que permita la generación de código fuente de aplicaciones educativas multi-dispositivo, a través de técnicas de reconocimiento de imágenes y patrones de diseño de interfaces de usuario para facilitar la laboriosa y tediosa tarea de desarrollo de aplicaciones al intentar obtener software de calidad en el menor tiempo posible. Por tal razón, se pretende utilizar las ventajas de los patrones de diseño de interfaz de usuario para simplificar la interacción usuario-aplicación y desarrollar un generador de aplicaciones para simplificar el desarrollo de software y permitir el desarrollo de aplicaciones multiplataforma.

Los beneficios de este proyecto de tesis ofrecen desarrollar un componente que genera aplicaciones educativas compatibles con cuatro sistemas operativos: Android™, Firefox® OS, macOS® y Windows Phone®, y la Web. Además, permite la generación de aplicaciones para tres dispositivos: móviles (teléfonos inteligentes y tabletas), escritorio y televisión.

Abstract

Abstract

Education is key to a just and progressive society, and it is specially provided by schools and reflects on various areas of our lives. The importance of education has led to new schemes in the development of educational applications. However, developing applications that provide an efficient UI design while allowing the user to easily become familiar with the software is a complicated task. Currently, software development focuses on reducing the time and cost of creating the software, without affecting the quality of the software.

This thesis project aims to develop a software component that allows the generation of source code of educational multi-device applications, through image recognition techniques and a set of User Interface Design Patterns (UIDPs) to facilitate the laborious, time-consuming task of applications development when trying to obtain quality software in the shortest possible time. On one hand, UIDPs simplify user-application interaction to create high-quality software. On the other hand, application generators simplify software development in general and allow for cross-platform applications development.

The benefits of this thesis offer to develop a component that generates educational applications compatible with four operating systems: Android™, Firefox® OS, macOS® and Windows Phone®, and the Web. In addition, it allows the generation of applications for three devices: mobiles (smartphones and tablets), desk and television.

Introducción

Introducción

El desarrollo de software y la educación están en constante evolución. La tecnología forma parte de la vida cotidiana, y cada vez es más fácil interactuar con ella. En la mayoría de las veces, la tecnología disponible en los dispositivos móviles y la televisión inteligente se utilizan para el entretenimiento en lugar del contexto educativo, esto provoca que se desaproveche su uso para incrementar el aprendizaje y la enseñanza.

Por tal razón, lograr que una aplicación proporcione un diseño de interfaz de usuario eficiente y agradable, y además permita al usuario familiarizarse fácilmente con el software es una tarea complicada, porque generalmente es necesario tener buenos conocimientos de diseño y experiencia para lograrlo. Una aplicación multi-dispositivo tiene la ventaja de visualizarse en diferentes dispositivos, pero de una manera óptima que aproveche los recursos del dispositivo donde se encuentra instalada. Al tener diferentes dispositivos provoca un problema en la interfaz de usuario, generalmente por el tamaño de la pantalla del dispositivo.

El contexto educativo tiene gran importancia porque permite el intercambio de conocimientos e incrementa el aprendizaje. Los niños, jóvenes y adultos aprenden constantemente, a causa de la enseñanza de diferentes temas. Por ejemplo, una receta de cocina, la compostura de un dispositivo electrónico, una asignatura de la escuela, entre otros.

La razón de utilizar contextos de uso es porque permiten establecer qué patrones de diseño de interfaz de usuario son recomendables para un dispositivo, facilitando el desarrollo de aplicaciones móviles multi-dispositivos sin perjudicar la calidad, el tiempo y el costo del producto de software.

Este trabajo se estructura de la siguiente forma: En el capítulo 1 se presenta el marco teórico, planteamiento del problema, objetivos y justificación de la tesis. En el capítulo 2 se mencionan algunos artículos que están relacionados con la tesis y una tabla comparativa de los mismos artículos. En el capítulo 3 se encuentra la aplicación de la metodología para el tema de tesis. En el capítulo 4 se describen los resultados obtenidos de la tesis, a través de tres casos de estudio. Finalmente, en el capítulo 5 se presentan las conclusiones obtenidas y las recomendaciones sugeridas de la tesis.

Capítulo 1. Antecedentes

Capítulo 1. Antecedentes

En este capítulo, se dan a conocer los conceptos que se utilizan a lo largo de la tesis, se presenta el problema a resolver, los objetivos, así como la justificación del mismo.

1.1. Marco teórico

A continuación, se presentan algunos conceptos relacionados con la tesis, los cuales son esenciales para tener un mejor entendimiento del proyecto propuesto y permita comprender los temas relacionados al procesamiento de imágenes, el software utilizado y el ámbito educativo.

1.1.1. Procesamiento digital de imágenes

El procesamiento digital de imágenes está presente en aplicaciones industriales, ciencias médicas, biometría e identificación, agricultura, ganadería, satélites de observación terrestre, como en tantas otras tareas de distinta índole en la sociedad. Es por ello que se enfatiza mucho en desarrollar nuevas y mejores técnicas para llevar a cabo estos tratamientos en las diversas actividades donde se requieran.

Una imagen se considera como el conjunto de puntos de colores, es decir, una sucesión coherente de puntos que conforman una matriz de información para el uso digital. Estos puntos se denominan '*píxeles*' (*picture element*). Un *píxel* es la menor unidad homogénea en color que forma parte de una imagen digital. Un filtro sobre una imagen digital es precisamente la operación que se le aplica a los píxeles de la misma para optimizarla, remarcar cierta información u obtener un efecto especial en ella.

Los objetivos principales que se buscan son:

- Suavizar la imagen: reducir las variaciones de intensidad entre *píxeles* vecinos.
- Eliminar ruido: eliminar aquellos *píxeles* cuyo nivel de intensidad es muy diferente al de sus vecinos [1].

1.1.2. Software

De acuerdo al glosario estándar de terminología de ingeniería de software de la IEEE [2] *software* es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y

Capítulo 1. Antecedentes

datos asociados, que forman parte de las operaciones de un sistema de computación. Y el software de una aplicación es aquel software diseñado para satisfacer las necesidades específicas de un usuario; por ejemplo, software para navegación, nómina, o control de procesos.

Para la realización de la tesis se utilizaron las herramientas que provee Matlab para el procesamiento de imágenes y el uso de redes neuronales.

1.1.2.1. MATLAB

MATLAB es un lenguaje de alto rendimiento que integra la computación, la visualización y la programación en un entorno fácil de usar donde los problemas y soluciones se expresan en notación matemática. Se usa generalmente en:

- Matemáticas y computación.
- El desarrollo de algoritmos.
- Modelado, simulación y creación de prototipos.
- Análisis, exploración y visualización de los datos.
- Gráficos científicos y de ingeniería.
- Desarrollo de aplicaciones, incluyendo la construcción de interfaces gráficas de usuario.

MATLAB es un sistema interactivo cuyo elemento básico de datos es una matriz que no requiere dimensionamiento. Esto le permite resolver problemas de computación técnica, especialmente aquellos con formulaciones matriciales y vectoriales, en una fracción del tiempo que tomaría escribir un programa en un lenguaje no interactivo escalar como C o Fortran.

MATLAB ofrece una familia de soluciones para aplicaciones específicas llamadas cajas de herramientas. Las cajas de herramientas son colecciones de las funciones de MATLAB que amplían el entorno de MATLAB para resolver clases particulares de problemas. Las cajas de herramientas están disponibles en áreas como el procesamiento de señales, sistemas de control, redes neuronales, lógica difusa, ondas, simulación, entre otros [3].

1.1.2.2. Image Processing Toolbox

Image Processing Toolbox proporciona un conjunto completo de algoritmos, funciones y aplicaciones para el procesamiento, análisis y visualización de imágenes, así como para el

Capítulo 1. Antecedentes

desarrollo de algoritmos. *Image Processing Toolbox* soporta un conjunto diverso de tipos de imágenes, tales como las de alto rango dinámico, las de resolución de *gigapíxeles*, las de perfiles *ICC* (*International Color Consortium*, Consorcio Internacional del Color) embebidos y las tomográficas. Las características y aplicaciones de visualización permiten explorar imágenes y vídeos, examinar una región de píxeles, ajustar el color y el contraste, crear contornos o histogramas y manipular regiones de interés (*ROI, regions of interest*). Esta caja de herramientas soporta flujos de trabajo para procesar y mostrar imágenes de gran tamaño, así como para navegar por ellas.

Características principales:

- Análisis de imágenes, incluyendo la segmentación, la morfología, la estadística y la medición.
- Mejora, filtrado y desenfoco de la imagen.
- Transformaciones geométricas y métodos de registro de imágenes basado en intensidad.
- Transformaciones de imagen, incluyendo *FFT* (*Fast Fourier transform*, Transformada rápida de Fourier), *DCT* (*Discrete Cosine Transform*, Transformada de Coseno Discreta), radón y la proyección de haz en abanico.
- Flujos de trabajo de imagen de gran tamaño, incluyendo el procesamiento de bloques, baldosas, y la visualización de varias resoluciones.
- Aplicaciones de visualización, incluyendo Visor de imágenes y Visor de Video.
- Funciones Multi-core y GPU habilitadas, y generación de código C [4].

1.1.2.3. Neural Network Toolbox

Neural Network Toolbox ofrece funciones y aplicaciones para el modelado de sistemas complejos no lineales que no son fácilmente modelados con una ecuación de forma cerrada. *Neural Network Toolbox* apoya el aprendizaje supervisado con alimentación directa, base radial y redes dinámicas. También es compatible con aprendizaje no supervisado con los mapas de auto-organización y capas competitivas. Con esta caja de herramientas se permite diseñar, entrenar, visualizar y simular redes neuronales. *Neural Network Toolbox* se utiliza para

Capítulo 1. Antecedentes

aplicaciones tales como ajuste de datos, reconocimiento de patrones, agrupación, predicción de series de tiempo, y modelado de sistemas dinámicos y de control.

Características principales:

- Redes supervisadas.
- Redes no supervisadas.
- Aplicaciones para el ajuste de datos, reconocimiento de patrones y agrupación.
- La computación paralela y el apoyo de la GPU para acelerar el entrenamiento (usando *Parallel Computing Toolbox*).
- Preprocesamiento y postprocesamiento para mejorar la eficiencia del entrenamiento de la red y la evaluación de rendimiento de la red.
- Representación de red modular para la gestión y la visualización de las redes de tamaño arbitrario.
- Bloques *Simulink* para la construcción y evaluación de redes neuronales y para aplicaciones de sistemas de control [5].

1.1.3. Aplicaciones Multi-dispositivo

Una aplicación multi-dispositivo es aquella que se ejecuta en diferentes dispositivos (teléfono inteligente, tableta, televisión inteligente, reloj, carro), donde la aplicación creada se ajusta a los diferentes tamaños de pantalla, y el código que generó la aplicación es capaz de funcionar correctamente en todos los dispositivos que se mencionan a continuación, incluyendo las limitaciones del hardware.

1.1.3.1. Dispositivos móviles

Un dispositivo móvil es aquel aparato electrónico de pequeño tamaño, con capacidades de procesamiento similares a las de una computadora. En la tesis se contemplaron dos dispositivos móviles, que son los teléfonos inteligentes y las tabletas (PC).

Un teléfono inteligente es el dispositivo móvil más utilizado y conocido por las personas [6], es un dispositivo móvil y portátil que permite realizar llamadas a través de una red telefónica, cuenta con un sistema operativo de alto nivel, una pantalla táctil, un navegador Web, permite

Capítulo 1. Antecedentes

realizar tareas múltiples al mismo tiempo y se usa con fines tanto personales como comerciales. De acuerdo con el reporte de *Cell Internet Use 2013* [7], aproximadamente el 63% de los adultos en América que cuentan con un teléfono inteligente, lo utilizan para conectarse a Internet.

Una tableta (PC) es un dispositivo que tiene prestaciones muy similares a las de una computadora, se presenta en un diseño plano, fino, ligero y compacto, con pantalla táctil. Este dispositivo móvil revolucionó el concepto de movilidad por tener un peso ligero y ser fácil de transportar. Otra de las facilidades que presenta es su capacidad para estar conectado a Internet en prácticamente todo momento de forma permanente desde cualquier lugar, además de disponer de un sinnúmero de aplicaciones que ayudan al usuario en una labor concreta, ya sea de carácter personal, profesional, de ocio o entretenimiento. Estas aplicaciones con las que cuenta la tableta, dependerán del sistema operativo con el que trabaje. Actualmente hay que destacar que los sistemas operativos que dominan el mercado de las tabletas son Android™ e iOS [8].

1.1.3.2. Televisión inteligente

Una televisión inteligente es un nuevo concepto de televisión en el que se integran la imagen, el sonido, la navegación Web y la sincronización con otros dispositivos. Una nueva generación de televisiones se caracteriza por su diseño innovador, por permitir al usuario acceder a una plataforma desde donde es posible acceder a Internet y ver contenidos o aplicaciones similares a las de un teléfono inteligente o una tableta, además de permitir el uso y la sincronización de todo tipo de dispositivos: PC, tableta, teléfonos inteligentes, videoconsola, entre otros. Para ello, los dispositivos utilizan los mismos protocolos. Los televisores inteligentes cuentan con una interfaz agradable por la cual es posible navegar de una forma intuitiva y cómoda, y a la vez permiten que el usuario cuente con un alto grado de interacción con el dispositivo. La gama alta de los últimos modelos está introduciendo también el control por voz y gestos [9].

1.1.4. Aplicaciones Multiplataforma

Una aplicación multiplataforma es aquella aplicación que se ejecuta en diferentes plataformas o en su defecto, permite la compilación del código fuente a diversas plataformas. A nivel tecnológico requiere un mayor esfuerzo de programación, para crear diferentes versiones de la aplicación y se ejecuten en cada uno de los entornos que tiene contemplado [10]. Por tal razón,

Capítulo 1. Antecedentes

se contemplaron principalmente cuatro sistemas operativos, los cuales se presentan a continuación.

1.1.4.1. Android™

Android™ es un sistema operativo basado en Linux para dispositivos móviles, como teléfonos inteligentes y tabletas, que se desarrolló inicialmente por Android™ Inc., una firma comprada por Google® en el año 2005. El anuncio del sistema Android™ se realizó el 5 de noviembre de 2007 junto con la creación de la *Open Handset Alliance*, un consorcio de 78 compañías de hardware, software y telecomunicaciones dedicadas al desarrollo de estándares abiertos para dispositivos móviles. Google® liberó la mayoría del código de Android™ bajo la licencia Apache, una licencia libre y de código abierto. Las unidades vendidas de teléfonos inteligentes con Android™ se situaron en el primer puesto en los Estados Unidos, en el segundo y tercer trimestre de 2010, con una cuota de mercado de 43,6% en el tercer trimestre. Android™ tiene una gran comunidad de desarrolladores de aplicaciones que extienden la funcionalidad básica de los dispositivos. Hasta la fecha, existen cerca de 200.000 aplicaciones disponibles para Android™. La mayoría de los programas están escritos en el lenguaje de programación Java. La estructura del sistema operativo Android™ se compone de aplicaciones que se ejecutan en un *framework* Java de aplicaciones orientadas a objetos sobre el núcleo de las bibliotecas de Java en una máquina virtual Dalvik con compilación en tiempo de ejecución [11].

1.1.4.2. Firefox® OS

Firefox® OS es un nuevo sistema operativo desarrollado por Mozilla denominado “*Boot to Gecko*” (B2G). Usa un núcleo Linux y parte de un motor de tiempo de ejecución basado en *Gecko*, que permite a los usuarios ejecutar aplicaciones desarrolladas íntegramente en HTML5, JavaScript, CSS y otras APIs de aplicaciones abiertas.

La principal característica del sistema operativo Firefox® OS es que todo es Web y está construido de acuerdo a los estándares abiertos de Internet como HTML5, CSS3 y JavaScript, así como otro tipo de aplicaciones más complejas en otros lenguajes que se empaquetan para emplearse en los dispositivos que utilizan Firefox® OS [12].

Capítulo 1. Antecedentes

1.1.4.3. IOS

IOS (anteriormente denominado iPhone OS) es un sistema operativo móvil de Apple desarrollado originalmente para el iPhone, después usado en los dispositivos *iPod Touch* e *iPad*. Es un derivado de Mac OS X, que a su vez se basa en el núcleo Darwin BSD. El iOS tiene cuatro capas de abstracción: la capa del núcleo del sistema operativo, la capa de "Servicios Principales", la capa de "Medios de comunicación" y la capa de "Cocoa Touch". Todo el sistema se encuentra en la partición "/root" del dispositivo, ocupa poco menos de 500 megabytes. El sistema operativo ocupa menos de medio Gigabyte del total del dispositivo, de 8 GB o 16 GB, esto se realizó para soportar futuras aplicaciones de Apple, así como aplicaciones de terceros publicadas en la iTunes Store o la App Store [11].

1.1.4.4. Windows Phone®

Windows Phone®, anteriormente llamado Windows Mobile es un sistema operativo móvil compacto desarrollado por Microsoft, y diseñado para su uso en teléfonos inteligentes y otros dispositivos móviles. Windows Phone® se basa en el núcleo del sistema operativo Windows CE y cuenta con un conjunto de aplicaciones básicas que utilizan las API de Microsoft Windows. Windows Phone® se diseñó para ser similar estéticamente a las versiones de escritorio de Windows. Además, existe una gran oferta de software de terceros disponible para Windows Mobile, la cual se adquiere a través de la plataforma *Windows Marketplace for Mobile*. Originalmente apareció bajo el nombre de Pocket PC, como una ramificación de desarrollo de Windows CE para equipos móviles con capacidades limitadas. En la actualidad, la mayoría de los teléfonos con Windows Mobile vienen con un estilete digital, que se utiliza para introducir órdenes pulsando en la pantalla. Windows Mobile evolucionó y cambió de nombre varias veces durante su desarrollo [11].

1.1.5. Patrones de diseño de interfaz de usuario

Una interfaz de usuario es la manera en la que un usuario utiliza e interactúa con un programa de software y permite a los usuarios ver representaciones gráficas de las opciones del programa. Los usuarios esperan controles fáciles de usar, que contengan etiquetas descriptivas y un formato claro. Por lo cual una interfaz de usuario bien desarrollada es esencial para la creación de una aplicación de software exitosa.

Capítulo 1. Antecedentes

Jennifer Tidwell declaró que "un patrón describe las posibles soluciones a un problema de diseño común dentro de un contexto determinado, mediante la descripción de las cualidades invariables de todas esas soluciones" cuando ella comenzó a estudiar los patrones de diseño en el MIT (*Massachusetts Institute of Technology*).

El padre del patrón de diseño, Christopher Alexander definió su concepto de patrón de diseño original como "una regla de tres partes, que expresa una relación entre cierto contexto, un problema y una solución." Él afirma que "Cada patrón describe un problema que se produce una y otra vez en nuestro entorno y, a continuación, se describe el núcleo de la solución a ese problema." Alexander utilizó estas creencias como primer paso para escribir sus propios patrones de diseño para la arquitectura [13].

1.1.5.1. Patrones de diseño de IU para dispositivos móviles

Jenifer Tidwell [14] mencionó que aunque cada persona es única, en general todos se comportan de manera predecible. Esto permite que los patrones de diseño ayuden a los usuarios a conseguir sus objetivos de una manera más eficaz, que permita una exploración segura, satisfacción, modificaciones en el diseño y una mejor interacción con el usuario. Además de presentar una agrupación de los patrones de diseño, hace un énfasis en el comportamiento de las personas con el software, porque la interfaz de usuario depende del humano que interactúe con el software.

Theresa Neil [15] presentó una clasificación similar a la de Jenifer Tidwell, porque se enfoca principalmente en la navegación, la organización, las redes sociales y en cómo presentar la información. Aunque utilizan diferentes nombres, cada una de ellas hizo que su clasificación de patrones de diseño de interfaz de usuario abarcara temas similares.

1.1.5.2. Patrones de diseño de IU para televisión inteligente

A diferencia de los dispositivos móviles, no hay una mucha información sobre patrones de diseño de IU para la televisión inteligente, aunque cada plataforma tiene las guías necesarias para desarrollar la interfaz de usuario.

Los patrones de diseño establecidos para la televisión inteligente dan mayor énfasis a la interacción del usuario con el dispositivo. Tal es el caso de Android™ y Google® que hacen

Capítulo 1. Antecedentes

referencia al uso del control remoto, el tamaño de la letra y a desactivar ciertos sensores (pantalla táctil, sensores de movimiento) que se permiten en una aplicación Android™.

1.1.5.3. Interfaz de Usuario 10-pies (3 metros)

El contexto que se dificulta al momento de diseñar aplicaciones para una televisión inteligente se conoce como “*lean-back*” o la experiencia de 10 pies. Conseguir esta experiencia no es necesariamente difícil, pero orientar una aplicación a la experiencia de 10 pies, después de ser diseñada para otros equipos de sobremesa o portátiles, lo hace complicado.

La principal clave para esta experiencia es tener en cuenta que la aplicación se usará en un dispositivo utilizado en una sala de estar o un entorno equivalente, su interfaz adapta la mentalidad de las personas a relajarse a 10 pies de distancia de una pantalla grande [16].

El principal patrón de diseño de Interfaz de Usuario que se menciona es el de 10-pies (3 metros), hace referencia a la distancia en la que se encuentra el usuario de la televisión. Por ejemplo, Intel [17] mencionó cómo aplicar el patrón 10-pies en un teatro en casa, utilizando una computadora con la plataforma Linux Mint usando XBMC. Donde fue necesario adaptar el uso del control remoto, a la computadora para utilizar el patrón 10-pies.

1.2. Planteamiento del problema

La tecnología forma parte de nuestra vida cotidiana, y cada vez es más fácil interactuar con ella. Además, la mayoría de las veces la tecnología que se encuentra a nuestro alcance, por ejemplo, los dispositivos móviles y la televisión inteligente se utilizan para el entretenimiento en lugar del contexto educativo. Generando que se desaproveche su recurso para incrementar el aprendizaje y la enseñanza.

Sin embargo, desarrollar aplicaciones que proporcionen un diseño de interfaz de usuario eficiente y agradable, y además permita al usuario familiarizarse fácilmente con el software es una tarea complicada, porque generalmente es necesario tener buenos conocimientos de diseño y experiencia de interfaz de usuario para lograrlo. En la actualidad el desarrollo de software se enfoca en disminuir el tiempo y el costo de la creación del software, sin afectar la calidad del mismo.

Capítulo 1. Antecedentes

Para que un docente realice una aplicación es necesario tener un método que facilite el desarrollo de la estructura de la aplicación. Y teniendo en cuenta que la visión es el sentido más utilizado por el ser humano para detectar detalles e interpretar los patrones que reciben los ojos. El procesamiento de imágenes permite a una computadora reconocer estos patrones y utilizarlos para un proceso de desarrollo.

Se pretende utilizar los patrones de diseño de interfaz de usuario con el procesamiento de imágenes en el ámbito educativo, para lograr que un docente desarrolle una aplicación para un dispositivo móvil o una televisión inteligente, dicha aplicación se visualizará correctamente en cada uno de ellos con el fin de aumentar la enseñanza y el aprendizaje.

1.3. Situación de la empresa

El Instituto Tecnológico de Orizaba es un centro educativo que pertenece al Tecnológico Nacional de México y se localiza en la avenida Oriente 9, entre las calles norte 14 y norte 22, del municipio de Orizaba en el estado de Veracruz. El instituto cuenta con un programa nacional de postgrados de calidad y tiene como oferta académica ocho licenciaturas, cinco maestrías y un doctorado.

1.4. Objetivo general y específico

En esta sección se presenta el objetivo general y los objetivos específicos de la tesis.

1.4.1. Objetivo general

Desarrollar un componente de software que permita la generación de código fuente de aplicaciones educativas multi-dispositivo, a través de técnicas de reconocimiento de imágenes y patrones de diseño de interfaces de usuario.

1.4.2. Objetivos específicos

- Estudiar y analizar los patrones de diseño de interfaz de usuario que se ocupan en cada dispositivo para después usar los de mayor relevancia en el componente.
- Establecer los contextos de uso adecuados para cada patrón de diseño que sean adecuados para una aplicación orientada a la educación.

Capítulo 1. Antecedentes

- Estudiar y analizar herramientas que soporten desarrollo de aplicaciones multi-dispositivo para seleccionar la más conveniente en el componente.
- Definir la arquitectura y los niveles de prestación del componente a desarrollar para delimitar los alcances del componente.
- Desarrollar técnicas para la interpretación y reconocimiento de imágenes para la transformación a componentes de una aplicación de software que permitan generar el código fuente de la aplicación multi-dispositivo, con los elementos de la imagen.
- Desarrollar un componente generador de código fuente que permita la construcción de aplicaciones educativas para diferentes dispositivos y plataformas.
- Definir un caso de estudio basado en los resultados obtenidos para evaluar el alcance y las limitaciones del componente desarrollado.

1.5. Justificación

La idea central de esta propuesta consiste en generar y/o desarrollar aplicaciones educativas multi-dispositivo que implique la generación del código fuente mediante un esquema específico de imágenes delimitadas, que se manipule utilizando técnicas de procesamiento de imágenes. En la actualidad los sistemas operativos móviles más populares son *Android*[™] (tableta, teléfono inteligente, televisión inteligente), *iOS*, *BlackBerry*[®] y *Windows Phone*[®]. Cada sistema operativo tiene su propia identidad, la cual se refleja en la apariencia y comportamiento de cada uno de los elementos que componen su interfaz. Sin embargo, todos comparten algunos puntos de vista fundamentales que se manifiestan en el diseño de sus interfaces, como lo es la navegación, los cuadros de diálogo, notificaciones, entre otros, que todos los sistemas operativos contienen, pero cada uno de ellos los representa en diferente lugar. Los patrones de diseño de interfaz son una solución a los problemas más comunes de diseño, y permiten al usuario sentirse más cómodo y seguro al utilizar la aplicación.

Raquel Vázquez et al. [18] presentaron un análisis de cinco diversos tipos de patrones de diseño de interfaces para dispositivos móviles, donde detectaron que varios patrones de diseño eran similares, ellos compilaron una colección de los patrones de diseño que cuenta con siete categorías, donde se muestran 17 patrones de diseño que se utilizan para la creación de aplicaciones móviles educativas y/o aplicaciones de TV.

Capítulo 1. Antecedentes

La aplicación propuesta tiene como finalidad utilizar una imagen digitalizada (previamente delimitada por una interfaz gráfica de usuario o un diagrama UML, por mencionar algunos ejemplos), para desarrollar una aplicación educativa para dispositivos móviles y/o TV, creada a partir del uso de patrones de diseño de interfaces para dispositivos móviles. Entre los beneficios esperados se encuentran:

- Generar únicamente el código fuente de los componentes que se consideren necesarios dentro de alguna imagen debidamente delimitada.
- La posibilidad de que el código fuente se ajuste a la experiencia del usuario, permitiendo al usuario interactuar fácilmente con la aplicación dependiendo del sistema operativo elegido.

Capítulo 2. Estado de la práctica

Capítulo 2. Estado de la práctica

A continuación, se presenta una recopilación de trabajos relacionados con este proyecto de tesis que permitirán conocer el estado actual de la investigación y analizar el trabajo realizado por otros autores para resolver problemas similares. De acuerdo a esto, la revisión del estado de la práctica describe las tecnologías, técnicas, métodos y arquitecturas desarrolladas para resolver los problemas que cada uno de los autores encontró durante su investigación.

2.1. Trabajos relacionados

En esta sección se encuentran trabajos relacionados con el uso de generadores de aplicaciones basados en los patrones de diseño de interfaz de usuario y las aplicaciones educativas. En la subsección 2.1.1 se destaca el alcance de algunos generadores de software y sus beneficios dentro del entorno de desarrollo de software. Por otro lado, en la subsección 2.1.2 se analizaron el uso de los patrones de diseño de interfaz de usuario en aplicaciones móviles y en la Web.

2.1.1. Desarrollo de aplicaciones para móviles

El primer paso en el desarrollo de aplicaciones móviles es determinar el tipo de aplicación que se va a construir. Heitkötter et al. [19] y, Brucker y Herzberg [20] dividieron las aplicaciones móviles en tres enfoques: 1) aplicaciones nativas, 2) aplicaciones Web y 3) aplicaciones híbridas. Las aplicaciones nativas son aquellas que se instalan en un dispositivo móvil y se crean utilizando tecnologías específicas de plataformas (por ejemplo, Java para Android™ o Swift para iOS) que permiten que las aplicaciones se ejecuten más rápido y eficientemente. Por otro lado, las aplicaciones Web utilizan un visualizador de contenido Web y su contenido se desarrolla a través de tecnologías Web estándar (por ejemplo, HTML5, CSS3 y JavaScript). Por último, las aplicaciones híbridas combinan las ventajas de las aplicaciones nativas y las aplicaciones Web, utilizando las características de las aplicaciones híbridas mediante el uso de un componente WebView para cada plataforma en la que se instalará y ocupa el hardware a través de un contenedor nativo que realiza llamadas a la API de cada plataforma.

El rápido crecimiento en el número de aplicaciones y usuarios provocó que el desarrollo de aplicaciones móviles sea uno de los campos más prometedores de la ingeniería de software. En

Capítulo 2. Estado de la práctica

el contexto de los generadores de aplicaciones móviles nativos, se encontraron los siguientes trabajos relevantes:

Pokress y Veiga [21] introdujeron una herramienta de programación visual llamada MIT App Inventor, que consiste en arrastrar y soltar una colección de componentes para diseñar y construir aplicaciones móviles completamente funcionales para la plataforma de Android™. Actualmente, MIT App Inventor es muy popular entre estudiantes, profesores, desarrolladores, aficionados, entre otros. Asimismo, Grønli et al. [22] propusieron la idea de una arquitectura de software basada en la nube, que permite una adaptación contextual de las interfaces de usuario. La arquitectura se basa en el uso de un cliente Android™, una aplicación de servidor en la nube y los servicios remotos de Google®. Los resultados de la evaluación de la arquitectura mostraron su capacidad de integrar y adaptar la información contextual para proporcionar al usuario un comportamiento y un contenido de aplicación oportuno, relevante y adaptado.

En el mismo contexto de los generadores de aplicaciones móviles nativas, Marin et al. [23] presentaron un marco llamado LIZARD, que permite crear aplicaciones para Windows Phone® y dos tipos de dispositivos Android™ (tabletas y teléfonos inteligentes). Las aplicaciones generadas con LIZARD siguen las pautas de diseño de interfaz de usuario y patrones arquitectónicos y diseño especificados por el fabricante del sistema operativo correspondiente. Asimismo, Vaupel et al. [24] mostraron un lenguaje de modelado y una infraestructura para construir aplicaciones nativas para Android™ e iOS, permitiendo así el desarrollo de aplicaciones flexibles en diferentes niveles de abstracción. Como resultado, los autores obtuvieron un enfoque prometedor para asegurar el rápido desarrollo de varias plataformas móviles. Desde una perspectiva diferente, Núñez Valdez et al. [25] mostraron un enfoque basado en modelos para generar y desplegar videojuegos multiplataforma de forma automática a través de un meta-modelo y un *DSL (Digital Subscriber Line, línea de suscripción digital)* gráfico. Para la generación de código fuente, los autores propusieron una plataforma para el desarrollo de juegos multi-plataforma ágil 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 (*Model-Driven Engineering, Ingeniería dirigida por modelos*).

Capítulo 2. Estado de la práctica

En el contexto de los generadores de aplicaciones web se encuentra el trabajo de Fogli et al. [26], donde los autores presentaron un lenguaje de patrones de diseño para la accesibilidad basado en las últimas especificaciones del W3C (*World Wide Web Consortium*, Consorcio WWW). El lenguaje es un recurso de diseño universal para ayudar a los diseñadores web a crear aplicaciones de Internet ricas y accesibles que cumplan con los estándares más recientes. Para evaluar el lenguaje de los patrones de diseño, se realizaron tres pasos: (1) un análisis heurístico con un panel de interacción hombre-máquina, (2) un estudio con un grupo de diseñadores web y (3) una validación aplicada en casos reales.

En cuanto a los generadores de aplicaciones híbridas, se discuten las siguientes obras: Primero, Heitkötter et al. [27]; Heitkötter, Majchrzak y Kuchen [28] presentaron a MD², un marco de desarrollo de aplicaciones multiplataforma basado en técnicas dirigidas por modelos. MD² consiste en 1) un lenguaje específico para describir concisamente las aplicaciones de negocio y 2) tres generadores de código que crean automáticamente aplicaciones completas de iOS, Android™ y *Backend* a partir de esta especificación. Además, Colombo Mendoza et al. [29] propusieron una plataforma como un servicio *PaaS* (*Platform as a Service*, Plataforma como Servicio) llamado MobiCloUP! La plataforma implementó un modelo de programación de generación de código automático para aplicaciones RIA (*Rich Internet Applications*, aplicaciones ricas basadas en estándares Web), como HTML5, CSS3 y AJAX.

En el mismo entorno de generadores de aplicaciones híbridas, Vásquez Ramírez et al. [30] presentaron un sistema híbrido de gestión de contenidos educativos multidisciplinarios llamado Athena. Athena proporciona un conjunto de interfaces basadas en Web para generar, administrar y presentar aplicaciones de contenido educativo, al tiempo que reduce el tiempo y el esfuerzo invertido en el proceso de desarrollo de aplicaciones híbridas. Como características distintivas, Athena toma prestadas las principales características de CMS (*Content Management System*) y LMS (*Learning Management System*) para administrar y reutilizar contenido educativo. Bajo un enfoque similar, Vásquez Ramírez et al. [31] propusieron más adelante AthenaTV, un generador de aplicaciones educativas basadas en Android™ para la televisión digital que utiliza los patrones de diseño de interfaz de usuario. AthenaTV utiliza el esquema de 10 pies usado por

Capítulo 2. Estado de la práctica

Google® para desarrollar interfaces basadas en los patrones de diseño de interfaces de Google® TV.

Finalmente, dado que el diseño flexible de aplicaciones móviles demostró ser un desafío importante para los desarrolladores de software, Desruelle et al. [32] propusieron a Webinos, una plataforma de *middleware* de aplicaciones multi-dispositivos que toma como base la infraestructura del Internet Futuro. Su objetivo es diseñar y proporcionar una plataforma de código abierto que permita que las aplicaciones y los servicios Web se ejecuten en la aplicación web. La estructura arquitectónica de Webinos se aplicó al diseño de aplicaciones, que están disponibles para todo el mundo, en cualquier momento y se implementan en cualquier dispositivo.

2.1.2. Patrones de diseño de interfaz de usuario en el desarrollo de software

Tanto el desarrollo de aplicaciones móviles como el de aplicaciones Web utilizan un buen diseño para lograr la interacción adecuada del usuario. Hussein et al. [33] mencionaron que para evitar la generación de aplicaciones con interfaces gráficas difíciles de usar y entender, los patrones de diseño de interfaz de usuario surgieron como una solución eficaz que mejora la facilidad de uso de la interfaz de una aplicación. Además, Cremonesi et al. [34] afirmaron que los patrones de diseño son reconocidos como potentes herramientas conceptuales para mejorar la calidad del diseño y reducir el tiempo y el costo del proceso de desarrollo, ya que permiten a los desarrolladores reutilizar soluciones de diseño.

SkillVille es una aplicación de aprendizaje basada en juegos desarrollada con el SDK de Corona para la plataforma Android™. Balayan et al. [35] presentaron a SkillVille, una herramienta de aprendizaje para el lenguaje y la ortografía, búsqueda y clasificación, y habilidades de memoria. Los resultados mostraron que SkillVille tenía potencial como herramienta de aprendizaje para los niños. Desde una perspectiva similar, Freire y Painho [36] estudiaron m-SportGIS, una aplicación desarrollada para la plataforma Android™ 4.0 utilizando Apache R Cordova™ y Sencha Touch. M-SportGIS utiliza capacidades de dispositivo e integra un servicio de mapas web (WMS) y un servicio de mapas de mosaico (TMP). La aplicación es explotada actualmente por el gobierno de Mozambique para inventariar las instalaciones deportivas.

Capítulo 2. Estado de la práctica

Aunque el uso de libros interactivos para niños ha aumentado exponencialmente en los últimos años, todavía no está claro qué beneficios traen estos libros. Ruttkay et al. [37] realizaron un análisis de los problemas interactivos del diseño de libros infantiles y proporcionaron una evaluación exploratoria de la experiencia de los niños con *Little Rooster*, un libro interactivo diseñado por los autores para niños de siete a ocho años. Los resultados del experimento mostraron que los niños simpatizaban más con los personajes del libro interactivo y encontraban su lectura más atractiva.

Quizás la cuestión principal de los masivos cursos en línea abiertos (MOOC) es su incapacidad para proteger la individualidad del usuario durante el proceso de aprendizaje. En su investigación, Ovesleová [38] analizó los factores que influyen en los usuarios de MOOC a lo largo del proceso de aprendizaje para definir criterios de evaluación para la formulación de interfaces adaptables. Además, motivado por el impacto de las tecnologías móviles en el dominio del aprendizaje de idiomas, Milutinović et al. [39] desarrollaron un modelo de objetos de aprendizaje ligero para aprender japonés. El modelo funciona bajo un sistema para almacenar y entregar estos objetos a las aplicaciones del cliente, y se usó en el sistema J-GO. J-GO utiliza un conjunto de UIDP para adaptarse a diferentes resoluciones de pantalla.

Finalmente, las capacidades de los patrones de diseño en el dominio médico se utilizan con fines de prevención y control de salud. En este sentido, Zargaran et al. [40] presentaron el *eTHR* (*Electronic Health Trauma Registry*, Registro Electrónico de Trauma de Salud), una aplicación basada en tabletas que se utilizará como un registro médico electrónico, que proporciona datos de vigilancia de lesiones sin obstaculizar el flujo de trabajo hospitalario. La interfaz de usuario de *eTHR* fue construida utilizando el marco de trabajo jQuery Mobile 1.0 y fue diseñada para guardar y actualizar datos en una base de datos MySQL a través de solicitudes JavaScript asincrónicas.

2.2. Análisis comparativo

Para proporcionar un informe más claro, la Tabla 2.1 presenta un análisis comparativo de los diferentes enfoques para el desarrollo de aplicaciones móviles mencionados en la sección 2.1.1. Como se logra observar, el uso de patrones de diseño de interfaz de usuario en generadores de

Capítulo 2. Estado de la práctica

aplicación de multi-plataforma/multi-dispositivo analizados se reporta una vez para aplicaciones nativas, una vez para aplicaciones web y una vez para desarrollo de aplicaciones híbridas.

Por otro lado, la Tabla 2.2 presenta el análisis comparativo de las diferentes aplicaciones que utilizan los patrones de diseño de interfaz de usuario. Como se observa, los patrones de diseño facilitan el proceso de desarrollo y la eficiencia de una aplicación, independientemente del dispositivo, la plataforma o incluso el dominio.

Como conclusión, en los trabajos se menciona que los patrones de diseño se utilizan para facilitar el trabajo en equipo, porque permiten a los desarrolladores enfocarse en una meta en común. Pero lamentablemente no son muy utilizados, porque los desarrolladores tienen problemas al utilizarlos dada su complejidad y la falta de herramientas de apoyo. Además, con el avance tecnológico y la existencia de nuevos dispositivos, el diseñar una Interfaz de Usuario se ha vuelto más complicado, porque cada dispositivo tiene diferentes capacidades y limitaciones. Lo que genera que una aplicación tenga patrones diferentes para cada dispositivo o plataforma.

También cabe resaltar que la televisión se volvió más interactiva, porque hoy en día se utiliza para diversos usos, incluidos el entretenimiento y el aprendizaje. Aunque todavía no tiene las mismas capacidades que un teléfono inteligente, tableta o *PC* se espera que en un futuro próximo tenga la misma experiencia de usuario, usabilidad y utilidad.

Capítulo 2. Estado de la práctica

Tabla 2.1 Comparación de enfoques para el desarrollo de aplicaciones móviles.

Autores	Dominio	Técnicas de desarrollo de software	Aplicación desarrollada	Multi-plataforma / Multi-dispositivo	Utiliza patrones de diseño	Genera código fuente
Pokress y Veiga (2013)	Indefinido	Programación visual	Aplicaciones Nativas	No / Sí	No	No
Grønli et al. (2014)	Indefinido	Arquitectura de software	Aplicaciones Nativas	No / No	No	Sí
Marin et al. (2015)	Indefinido	Dirigida por modelos	Aplicaciones Nativas	Sí / Sí	Sí	Sí
Vaupel et al. (2016)	Multi-dominio	Dirigida por modelos	Aplicaciones Nativas	Sí / No	Sí	Sí
Núñez Valdez et al. (2016)	Entretenimiento	Ingeniería orientada a modelos:	Aplicaciones Nativas	Sí / No	No	Sí
Fogli et al. (2014)	Indefinido	Lenguaje de patrones de diseño	Aplicaciones Web	Sí / Sí	Sí	Sí

Capítulo 2. Estado de la práctica

Autores	Dominio	Técnicas de desarrollo de software	Aplicación desarrollada	Multi-plataforma / Multi-dispositivo	Utiliza patrones de diseño	Genera código fuente
Heitkötter et al. (2013)	Negocios	Dirigida por modelos	Aplicaciones híbridas	Sí / No	No	Sí
Colombo Mendoza et al. (2014)	Indefinido	Herramienta de configuración	Aplicaciones híbridas	Sí / Sí	No	Sí
Vásquez Ramírez et al. (2014)	Educación	CMS y LMS	Aplicaciones híbridas	Sí / Sí	No	Sí
Ramirez Vasquez et al. (2014)	Educación	Patrones de diseño de interfaz de usuario	Aplicaciones híbridas	No / No	Sí	No
Desruell et al. (2016)	Indefinido	Framework	Aplicaciones híbridas	Sí / Sí	Sí	Sí

Capítulo 2. Estado de la práctica

Tabla 2.2 Análisis comparativo de las aplicaciones que utilizan patrones de diseño de interfaz de usuario

Autor	Tipo de aplicación	Dispositivo	Plataforma	Resultado
Balayan et al. (2014)	Aprendizaje móvil	Móvil	Android™	SkillVille es un juego móvil destinado a promover el aprendizaje de tres habilidades básicas de percepción visual: identificación, organización e interpretación de los datos sensoriales recibidos por el individuo a través del ojo.
Freire y Painho (2014)	Deportes	Móvil	Android™	M-SportGIS es una aplicación desarrollada para la plataforma Android™ 4.0 que utiliza Apache® Cordova™ y Sencha Touch.
Ruttkay et al. (2014)	Libro interactivo	Móvil	Indefinido	Análisis profundo de problemas interactivos de diseño de libros infantiles y evaluación exploratoria de la experiencia con "Little Rooster", un libro interactivo diseñado e implementado en el laboratorio, principalmente para la investigación.

Capítulo 2. Estado de la práctica

Autor	Tipo de aplicación	Dispositivo	Plataforma	Resultado
Ovesleová (2015)	MOOC	Web	Web	Estudio de los factores que influyen en los usuarios de MOOC a lo largo del proceso de aprendizaje para definir criterios de evaluación para la formulación de interfaces adaptables.
Milutinović et al. (2015)	Aprendizaje móvil	Móvil	Android™	El sistema J-GO utiliza un modelo para almacenar y entregar objetos de aprendizaje ligeros destinados a aprender japonés.
Zargarán et al. (2014)	Medicina	Móvil	Web	La aplicación eTHR se encuentra basada en tabletas y se utiliza como un expediente médico electrónico. Proporciona datos de vigilancia de lesiones sin obstaculizar el flujo de trabajo del hospital.

2.3. Propuesta de solución

Este apartado presenta un análisis de las tecnologías de información utilizadas para seleccionar la solución al problema abordado en este tema de tesis y las razones que justifican la elección de la solución.

2.3.1. Análisis de las tecnologías de información (TI)

Existe una gran variedad de tecnologías de información que permiten el desarrollo del componente de software que se propone en esta tesis. Por tal razón, se realizó un análisis de los

Capítulo 2. Estado de la práctica

lenguajes de programación, los entornos de desarrollo y las bibliotecas para el procesamiento de imágenes por computadora.

A continuación, en la Tabla 2.3 se presenta un análisis acerca de los entornos de desarrollo para el procesamiento de imágenes, los lenguajes de programación que lo implementan, la licencia de software que utilizan, el formato de imágenes con las cuales es compatible y las características más relevantes de cada software.

Tabla 2.3 Tabla comparativa de bibliotecas para el procesamiento de imágenes

Procesamiento de imágenes				
Software	Lenguajes compatibles	Licencia	Formatos de imágenes	Características
Matlab	C, C++, C#, Fortran, Java, Visual Basic	Licencia comercial	bmp, gif, hdf, jpeg, jpg, jp2, jpf, jpx, j2c, j2k, pbm, pcx, pgm, png, pnm, ppm, ras, tiff, tif, xwd, cur, ico	<ul style="list-style-type: none"> • Facilita el desarrollo del software con el uso de bibliotecas. • Gestiona la memoria de la computadora.
OpenCV	Android™, C, C++, C#, Java, Python,	Licencia gratuita	bmp, dib, jpeg, jpg, jpe, jp2, png, pbm, pgm, ppm, sr, ras, tiff, tif	<ul style="list-style-type: none"> • Es rápido. • Dificulta el desarrollo de un software. • La gestión de memoria es realizada por el desarrollador de software.

Capítulo 2. Estado de la práctica

Procesamiento de imágenes				
Software	Lenguajes compatibles	Licencia	Formatos de imágenes	Características
SimpleCV	Python	Licencia gratuita	bmp, dib, jpeg, jpg, jpe, jp2, png, pbm, pgm, ppm, sr, ras, tiff, tif	<ul style="list-style-type: none">• Es un <i>framework</i> que ocupa las bibliotecas de OpenCV.• Utiliza Python para aplicaciones Web, Microcontroladores, escritorio, entre otras.

2.3.1.1. Justificación de la solución seleccionada

De acuerdo con el análisis de la sección de la Tabla 2.3 el uso de una aplicación Web facilitará al usuario el uso y aprendizaje de la aplicación, Matlab es un lenguaje que soporta un mayor número de algoritmos de procesamiento y gracias a la sencillez que se tiene para desarrollar un software permitió identificar adecuadamente los patrones de diseño en una imagen delimitada.

Existen dos trabajos relacionados con esta tesis. El primero fue desarrollado por la M.S.C. Laura Nely Sánchez Morales, quien desarrolló un componente de software que permite la generación de código fuente de aplicaciones multi-dispositivo, a través de técnicas de reconocimiento de imágenes y patrones de diseño de interfaces de usuario; el segundo proviene de la Dr. Raquel Vásquez Ramírez quien desarrolló una infraestructura tecnológica para la creación de aplicaciones multi-dispositivo y multi-plataforma sobre una arquitectura de cómputo en la nube para crear, editar y publicar contenidos en un software educativo que permita la exportación y visualización de estos contenidos en múltiples dispositivos y múltiples plataformas con el fin de apoyar al profesor en el proceso de enseñanza – aprendizaje.

Por tal razón, se escogió a PHP como lenguaje de programación para desarrollar una aplicación Web que permita consumir los componentes que se desarrollaron, para el procesamiento de imágenes, además se utilizó MATLAB porque proporciona una amplia documentación de libre acceso, además en lo relacionado al procesamiento de imágenes se tienen mayores

Capítulo 2. Estado de la práctica

implementaciones documentadas facilitando el aprendizaje y el desarrollo del componente a generar.

Capítulo 3. Aplicación de la metodología

Capítulo 3. Aplicación de la metodología

En este capítulo se presenta los patrones de diseño de interfaz de usuario utilizados en el módulo generador de código, su arquitectura basada en capas y el flujo de trabajo necesario para la generación del código de una aplicación educativa.

3.1. Introducción

En esta sección se presenta la metodología utilizada para el desarrollo de la tesis, la cual se basa en el método científico inductivo y se representa a través de cinco etapas. El seguimiento y cumplimiento de cada una de estas etapas permitió culminar en tiempo y forma la solución del problema propuesto. En la Figura 3.1 se presenta cada una de las etapas y las actividades que se realizaron en ellas, las cuales se describirán a continuación:

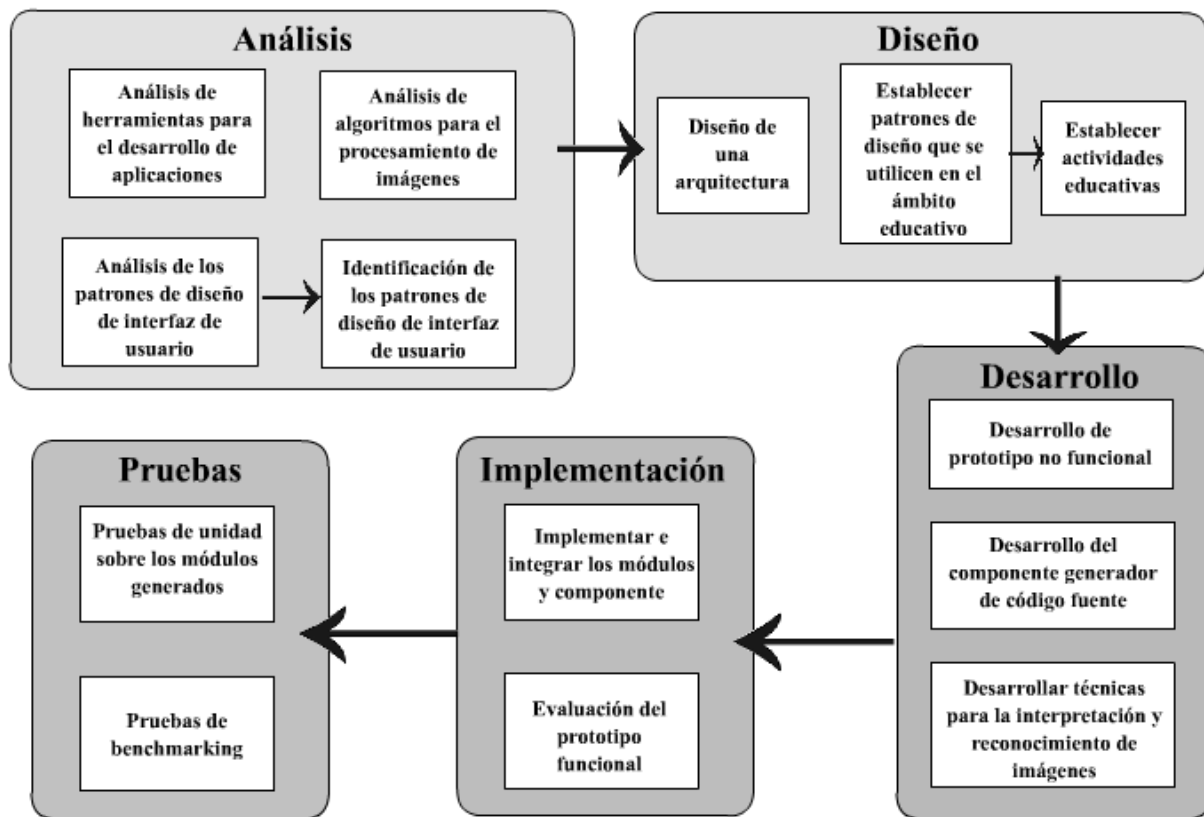


Figura 3.1 Metodología dividida en etapas y actividades

- **Análisis:** Durante esta etapa se realizó la revisión del estado del arte de las técnicas de análisis y procesamiento de imágenes digitales, así como la detección de patrones; así

Capítulo 3. Aplicación de la metodología

también se realizó la revisión de los patrones de diseño de interfaces de usuario aplicados a la generación de aplicaciones educativas. Resultado de dichas revisiones se generó un análisis comparativo tanto de las técnicas de análisis y procesamiento de imágenes digitales y los patrones de diseño de interfaces de usuario.

- **Diseño:** En esta etapa se diseñó una arquitectura que permita cumplir con el propósito de la tesis, así mismo se revisaron los patrones de diseño de interfaces de usuario identificados en la etapa anterior con el objetivo de determinar los aspectos importantes en el área del procesamiento de imágenes, así como los patrones de diseño de interfaces de usuario, y así ofrecer mejores soluciones en términos de eficiencia, consumo de recursos computacionales, eficiencia en la identificación de elementos en imágenes no ideales y otros factores decisivos al momento de hablar de generación automática de software.
- **Desarrollo:** Con la finalidad de automatizar parte del proceso de desarrollo de software, en esta se desarrolló un prototipo no funcional, se construyó el componente generador de código fuente y las técnicas para la interpretación y el reconocimiento de imágenes. Así como el desarrollo de los módulos que conforman la arquitectura diseñada en la etapa 3.
- **Implementación:** Durante esta etapa se implantó el software desarrollado en la etapa de desarrollo. Como prueba de concepto se evaluó el funcionamiento en conjuntos de los módulos y componentes.
- **Pruebas:** Durante esta etapa se realizaron diversas pruebas a los métodos computacionales propuestos utilizando diversos casos de estudio para validar los resultados obtenidos. Por tal razón, se realizaron pruebas de unidad sobre los módulos de generación automática de software y pruebas de benchmarking sobre los métodos computacionales propuestos.

3.2. Arquitectura y funcionalidad del componente generador de código

Después se realizó el diseño de la arquitectura del componente generador de código, donde se estableció el diseño orientado a una arquitectura en capas y se diseñaron los componentes que forman parte de la aplicación.

Capítulo 3. Aplicación de la metodología

3.2.1. Descripción de la arquitectura

En la arquitectura propuesta se utilizaron seis patrones de diseño de interfaz de usuario, estos patrones de diseño de interfaz de usuario fueron seleccionados a partir del análisis que se realizó en la sección anterior, la arquitectura se pensó con el fin de permitir el desarrollo de aplicaciones educativas para teléfonos inteligentes, tabletas, televisiones inteligentes y Web, además de ser compatible con diferentes plataformas.

El generador de aplicaciones educativas tiene una arquitectura de seis capas representada en la Figura 3.2. A través de una plataforma Web, EduGene recibe un archivo XML para obtener un paquete que contiene los proyectos de las plataformas seleccionadas en el proceso de configuración del despliegue. EduGene utiliza una arquitectura de múltiples capas para simplificar la comprensión de tareas y permitir cambios en la plataforma sin afectar a las otras capas. Las capas de EduGene se describen a continuación:

- **Capa de presentación:** Esta capa despliega contenido mediante el uso de HTML5, CSS3 y JavaScript. Además, a través de una interfaz gráfica de usuario permite la interacción con la capa de integración para facilitar el proceso de generación de aplicaciones.
- **Capa de integración:** Esta capa utiliza los patrones de diseño de interfaz de usuario para generar un archivo XML que permite la generación de una aplicación dependiendo del tipo de aplicación educativa y envía a la capa de presentación la ruta que dirige al archivo recién creado que contiene la navegabilidad y la estructura de la aplicación educativa solicitada. Sin embargo, si hay un problema con la generación de la aplicación, esta capa envía un mensaje de error.
- **Capa de servicios:** En esta capa se proporciona un servicio que recibe el archivo XML y genera la aplicación educativa basada en las especificaciones del archivo basado en XML.
- **Capa de negocio:** Esta capa gestiona las solicitudes recibidas por la capa de servicios, a continuación, a través del archivo basado en XML genera la aplicación educativa utilizando especificaciones de archivo basadas en XML.
- **Capa de datos:** Esta capa contiene la estructura básica de cada plataforma y dispositivo, además de un conjunto de patrones de diseño que permiten la generación de aplicaciones.

Capítulo 3. Aplicación de la metodología

- **Capa administradora de datos:** Esta capa contiene la estructura básica de los objetos codificados a través del archivo XML, estos objetos contienen la estructura de cada plataforma y dispositivo, su navegación y los módulos que contienen un conjunto de patrones de diseño que permiten la generación de aplicaciones.

EduGene genera proyectos para cada dispositivo basado en la plataforma seleccionada. Por ejemplo, cuando necesite crear una aplicación educativa para Firefox[®] OS y Android[™], y también se requiere especificar que la aplicación se ejecute en un teléfono inteligente y una televisión inteligente. El generador de aplicaciones genera proyectos dependiendo de los dispositivos disponibles para cada plataforma. En el caso de la plataforma Firefox, el generador de aplicaciones genera un proyecto para el teléfono inteligente. Sin embargo, construye dos proyectos para la plataforma Android[™], uno para el teléfono inteligente y otro para la televisión inteligente.

Considerando la arquitectura de EduGene representada en la Figura 3.2, el proceso de generación de aplicaciones educativas se presenta a continuación:

1. Inicia sesión para acceder a la plataforma de EduGene.
2. Asistente para introducir las características de la aplicación.
 - a) Ingresa una imagen con los patrones de diseño de IU que son identificados a través de un módulo que se encuentra en la capa de datos, posteriormente determina qué aplicación educativa permite el uso de esos patrones de diseño.
 - b) Selecciona los patrones de diseño de IU de acuerdo a la aplicación educativa seleccionada.
3. El generador crea un archivo XML que posteriormente se envía a la capa de integración.
4. La capa de integración recibe el archivo basado en XML, después prepara la estructura y envía el archivo XML a la capa de servicios.
5. El componente interfaces de servicios recibe el archivo XML y lo envía al componente de fachada de la aplicación que se encuentra en la capa de negocio.
6. El componente de fachada de la aplicación determina los pasos necesarios para cumplir con la solicitud.

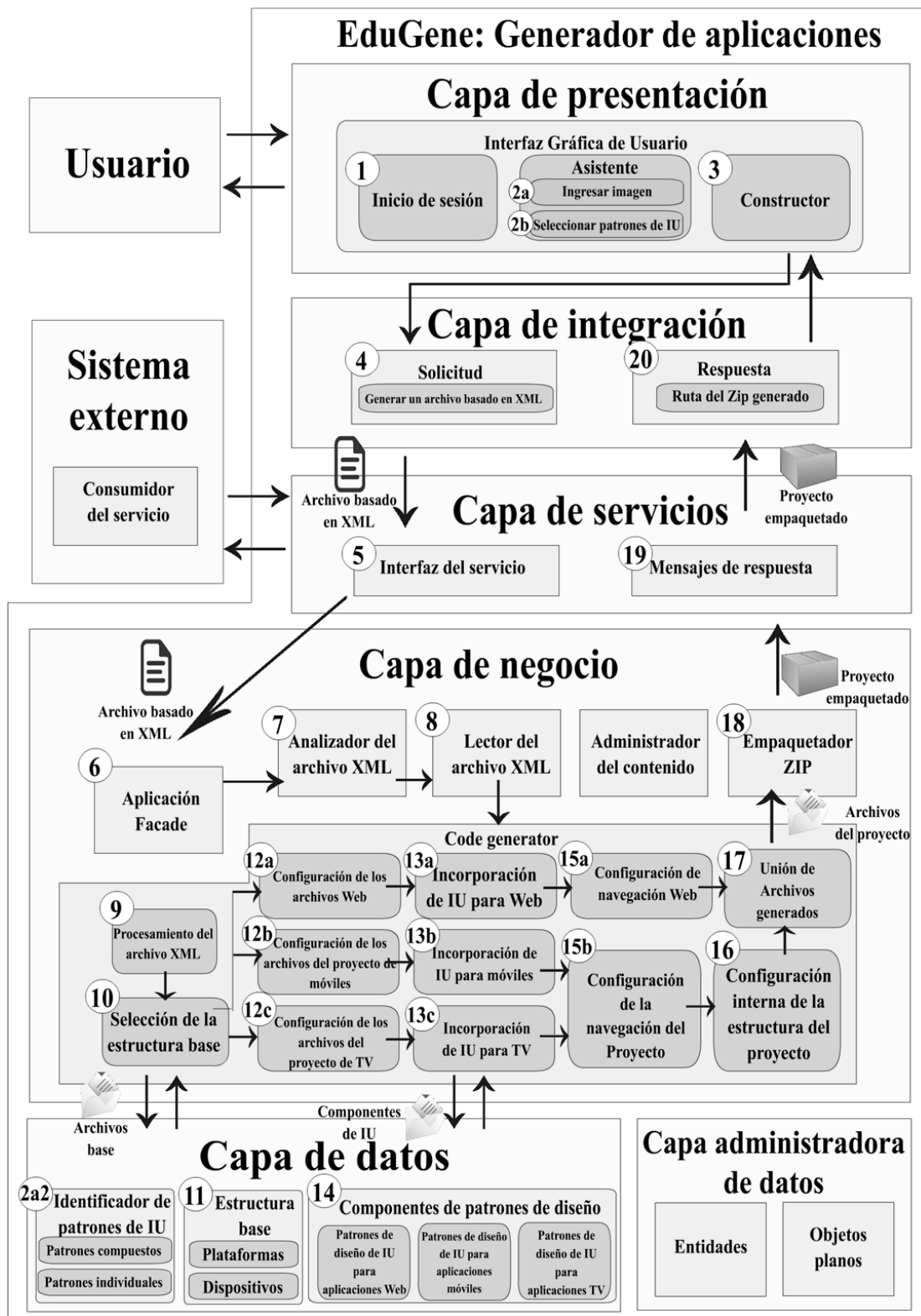


Figura 3.2 Arquitectura en capas de EduGene

Capítulo 3. Aplicación de la metodología

7. El archivo XML se analiza para verificar su estructura correcta.
8. Se procesa la información contenida en el archivo XML, esta información es utilizada por el generador de código.
9. Se reciben los objetos generados para comenzar a generar la aplicación.
10. Se seleccionan las estructuras básicas de plataformas y dispositivos. Para conseguirlo, la selección de la estructura base del módulo envía una petición a la capa de datos para obtener los archivos necesarios.
11. El componente llamado estructuras básicas recibe la solicitud y envía los proyectos y archivos necesarios para cada plataforma y dispositivo.
12. En esta parte, cada proyecto se configura utilizando los archivos base obtenidos de la capa de datos. Dependiendo del proyecto, se implementan las siguientes configuraciones:
 - a) Configura lo necesario para su correcto funcionamiento en un proyecto Web.
 - b) Configura lo necesario para su correcto funcionamiento en un dispositivo móvil.
 - c) Configura y habilita los permisos que el proyecto necesita para ejecutarse correctamente en un televisor inteligente.
13. A continuación, se agrega un conjunto específico de patrones de diseño, dependiendo del proyecto que se va a generar:
 - a) Realiza la solicitud a la capa de datos para obtener los patrones de diseño necesarios para trabajar en la Web.
 - b) Realiza la solicitud a la capa de datos para obtener los patrones de diseño necesarios para trabajar en dispositivos móviles.
 - c) Realiza la solicitud a la capa de datos para obtener los patrones de diseño necesarios para trabajar en el televisor inteligente.
14. La capa de datos devuelve los patrones de diseño apropiados, dependiendo del proyecto que se va a generar.
15. A continuación, se agregan patrones de diseño dependiendo del proyecto que se generará:
 - a) Comprueba y corrige los errores de navegación del proyecto Web generado.
 - b) Comprueba y corrige los errores de navegación generados para dispositivos móviles y proyectos de TV inteligente.

Capítulo 3. Aplicación de la metodología

16. Realiza las últimas configuraciones en el proyecto para que funcione correctamente en cada entorno de desarrollo.
17. Recibe todos los proyectos y procede a organizarlos en un conjunto predeterminado de carpetas.
18. Empaqueta en un archivo ZIP al conjunto de proyectos desarrollados por el generador de aplicaciones.
19. Utiliza la ruta del archivo ZIP y devuelve una dirección URL que permite descargar el archivo ZIP.
20. Devuelve la URL a la GUI para generar un botón para descargar el archivo

3.2.2. Componentes de software utilizados en EduGene

Cada capa de la arquitectura contiene un conjunto de componentes, cuyas funciones se describen a continuación:

- **Interfaz gráfica de usuario:** Este componente utiliza un inicio de sesión para administrar los usuarios y un asistente para seleccionar los patrones de diseño de interfaz de usuario que usará la aplicación educativa, finalmente envía los patrones de diseño a la capa de integración para generar el archivo XML. Además, muestra el hipervínculo para descargar la aplicación educativa o muestra un mensaje de error cuando suceda un problema al generar la aplicación.
- **Solicitud:** Este componente recibe los patrones de diseño basados en el componente de interfaz gráfica de usuario, posteriormente desarrolla una estructura para el archivo XML y finalmente crea el archivo XML y lo envía al componente generador de código.
- **Respuesta:** Este componente recibe la ruta de los archivos generados por el componente generador de código. Además, envía a la interfaz gráfica de usuario el hipervínculo para descargar la aplicación educativa.
- **Interfaz de servicios:** El siguiente componente es la piedra angular de la capa de servicio, ya que recibe llamadas de la capa de integración o incluso de un consumidor de servicios. Luego se encarga de validar la información y enviar los datos a la capa de negocio.
- **Mensaje de respuesta:** Este componente proporciona una respuesta al momento de construir la aplicación educativa.

Capítulo 3. Aplicación de la metodología

- **Aplicación Fachada:** Este componente hace uso del patrón de diseño *Facade* y utilizada sus ventajas para ofrecer una interfaz más simple para la comunicación entre los demás componentes de la capa de negocio.
- **Analizador del archivo XML:** Dentro de la capa de negocios se encuentra este componente, que se encarga de analizar que el archivo XML se encuentre correctamente estructurado.
- **Lector de archivo XML:** La tarea principal de este componente es tomar la información contenida dentro del archivo XML y construir los objetos necesarios para facilitar la lectura del archivo, cabe mencionar que hace uso de los componentes que se encuentran dentro de la capa administradora de datos.
- **Administrador del contenido:** Este componente administra los patrones de diseño de interfaz de usuario, a las estructuras de las plataformas y los dispositivos.
- **Generador de código:** Este es un componente muy importante porque procesa el archivo XML y genera la aplicación educativa. Para ello, utiliza los siguientes módulos:
 - Procesamiento del archivo XML: Este módulo recibe los objetos que representan al archivo XML para la gestión de contenidos.
 - Selección de la estructura base: Este módulo busca en la información obtenida por el archivo XML las plataformas y dispositivos seleccionados para generar la aplicación.
 - Configuración de archivos de proyecto: Este módulo representa tres formas de configuración, tales configuraciones dependen del dispositivo. Por tal razón, el componente generador de código administra su estructura y facilita la agregación de los patrones de diseño de interfaz de usuario de la aplicación educativa.
 - Incorporación de los patrones de diseño de interfaz de usuario: Debido a que la estructura base de la aplicación se basa en los patrones de diseño, este módulo se encarga de incorporar a cada proyecto dependiendo del dispositivo y la plataforma desarrollada.
 - Configuración de la navegación: Debido a que cada dispositivo utiliza un entorno de desarrollo diferente, es necesario utilizar una configuración diferente para cada dispositivo.
 - Configuración interna de la estructura del proyecto: Este módulo modifica internamente el proyecto haciendo principal referencia al nombre del proyecto, nombre de la

Capítulo 3. Aplicación de la metodología

aplicación, autores, licencia y cualquier modificación necesaria para el correcto funcionamiento del proyecto.

- Unión de archivos generados: Este módulo reúne todos los proyectos y archivos que se generaron en el componente generador y facilita el proceso de empaquetado.
- **Identificador de patrones de IU:** Este componente utiliza un conjunto de bosquejos de patrones de diseño de interfaz de usuario, una red neuronal y un procesamiento de imágenes en Matlab para reconocer las características de las imágenes e identificar los patrones de diseño de IU que representa la imagen enviada a EduGene.
- **Estructura base:** Este componente contiene las estructuras base de los proyectos, organizándolos por plataforma y dispositivo, de manera que se genera un proyecto independiente para cada plataforma y dispositivo.
- **Componentes de patrones de diseño de interfaz de usuario:** Este componente contiene los patrones de diseño de interfaz de usuario, separados por dispositivo para facilitar su incorporación a los proyectos generados. A continuación, se presentan los nueve patrones de diseño que contiene el componente:
 - 1) Carrusel: Se utiliza para presentar información mediante el uso de imágenes.
 - 2) *Dashboard*: Permite una mayor navegación en la aplicación mediante un conjunto de operaciones.
 - 3) Diálogo: Interrumpe el flujo de la aplicación para recibir una respuesta del usuario.
 - 4) Listas: Muestra información de forma ordenada y en forma de una lista.
 - 5) *Login*: Proporciona seguridad a través de un nombre de usuario y una contraseña registrados.
 - 6) Barra de progreso: Indica el progreso de una operación ejecutada por la aplicación.
 - 7) Tabla: Muestra información compleja utilizando filas y columnas.
 - 8) Pantalla: Imagen o texto que aparece antes de iniciar la aplicación.
 - 9) Video: Reproduce un videoclip.
- **Entidades:** En este componente se hace referencia a las entidades que se utilizan en el proceso de lectura del archivo XML, para generar objetos que faciliten la información del archivo XML.

Capítulo 3. Aplicación de la metodología

- **Objetos planos:** En este componente se hace referencia a las clases que representan a la información del archivo XML.
- **Empaquetador ZIP:** Para finalizar la construcción del o los proyectos generados por el componente generador de código, este último componente se encarga de generar un archivo empaquetado ZIP utilizando los archivos generados.

3.2.3. Descripción del flujo de trabajo usado en la arquitectura

En la Figura 3.3 se presenta el flujo de trabajo de la arquitectura. La arquitectura permite la generación automatizada de aplicaciones educativas y realiza el siguiente flujo de trabajo: 1) Se inicia sesión para administrar a los usuarios que utilicen la aplicación; 2) Selecciona los patrones de diseño de interfaz de usuario que se planean utilizar en la aplicación educativa, estos patrones de diseño dependerán del tipo de aplicación educativa que se va a generar; 3) Se selecciona la plataforma en la cual se construye la aplicación educativa; 4) Posteriormente se genera un archivo XML con la información necesaria para generar la aplicación educativa; 5) Se envía la petición al servicio Web que proporciona el componente generador de código, el cual se encarga de validar, leer y procesar la información del archivo XML; 6) Después con la información del archivo XML se selecciona la estructura base del proyecto de la aplicación educativa; 7) Inmediatamente se procede a configurar los archivos y/o el proyecto para que a partir de ahí se incorporen los patrones de diseño necesarios para cada aplicación dependiendo del dispositivo; 8) Entonces se comienza a configurar la navegación de los archivos y/o proyecto generado, y posteriormente dependiendo si es una aplicación móvil o de televisión se procede a configurar la estructura interna del proyecto; 9) Después se procede a unir los proyectos que se generaron en caso de ser más de dos y se comienza a empaquetar en un archivo ZIP; y 10) finalmente el servicio regresa como respuesta la ruta del archivo ZIP para facilitar su descarga.

Capítulo 3. Aplicación de la metodología

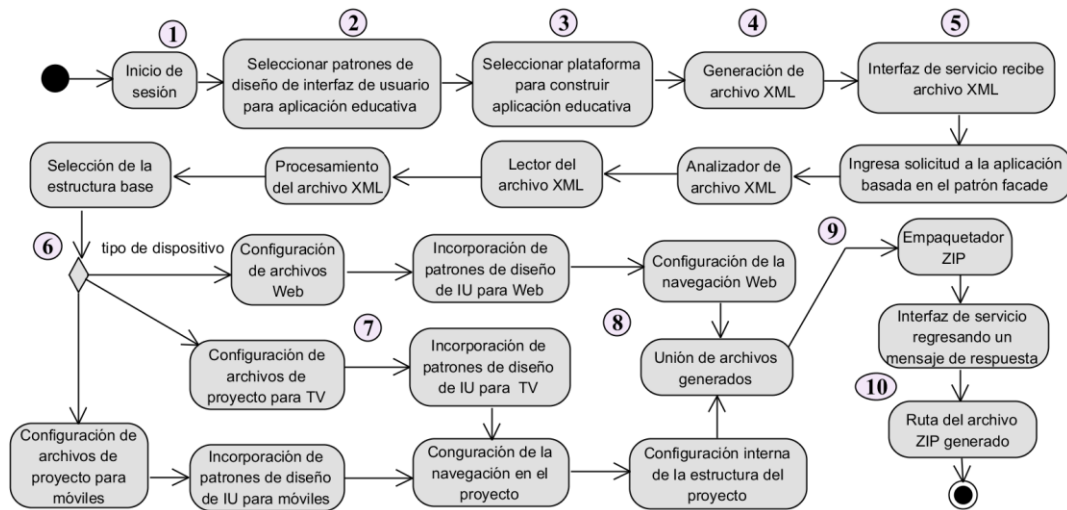


Figura 3.3 Flujo de trabajo de la arquitectura

3.3. Identificación de actividades de aprendizaje en interfaces gráficas de usuario

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 [41].

Este trabajo de tesis se centró exclusivamente en el desarrollo de blogs, MOOCs y *wikis*, ya que son las aplicaciones más utilizadas en los sistemas de *e-learning*. Por otra parte, el tipo de actividades educativas que promueven se representan fácilmente mediante la utilización de los patrones de diseño de interfaz de usuario [42], [43]. En la Tabla 3.1 se describen cada una de estas tres aplicaciones educativas, y por consiguiente se destacan sus fortalezas y las teorías de aprendizaje que soportan estas aplicaciones.

Capítulo 3. Aplicación de la metodología

Tabla 3.1 Descripción y beneficios de las aplicaciones educativas

Aplicaciones educativas	Descripción y beneficios	Teorías de aprendizaje
Blog	Utilizado para construir conocimiento y motivar la resolución de problemas, la reflexión y la comunicación de las emociones [44]. Los blogs permiten a los usuarios publicar contenido, comentar y hacer observaciones sobre los mensajes de otros usuarios [45].	Colaborativa [44]
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 [46].	Distributiva [47]
Wiki	Las <i>wikis</i> son una herramienta didáctica en línea que permite desarrollar procesos de gestión del conocimiento en la educación [48]. Una de las características principales de las <i>wikis</i> es que son beneficiosas en el desarrollo profesional [49].	Constructivista social [48] y comunitaria [49]

Cabe mencionar que los patrones de diseño satisfacen al menos una de las teorías de aprendizaje promovidas por los tres tipos de aplicaciones educativas a desarrollar. Considerando que el propósito del constructivismo colaborativo, distributivo, social y comunitario es lograr la interacción entre el usuario y el contenido educativo. Por ejemplo, las teorías antes mencionadas

Capítulo 3. Aplicación de la metodología

tienen en común el patrón de diseño *Login*. Sin embargo, ciertos patrones de diseño, como el Diálogo funcionan mejor para generar un MOOC, en lugar de un blog o un *wiki*.

3.4. Patrones de interfaz de usuario soportados

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 Theresa Neil [15], Jennifer Tidwell [14], Mari Sheibley [50], Anders Toxboe [51] y UNITiD [52], en el caso de las televisiones inteligentes se encuentra LG Developer [53], Android TV [54] y Apple TV [55]. En la Tabla 3.2 se presentan las categorías que cada autor asigna a sus patrones de diseño de interfaz de usuario.

Tabla 3.2 Categorías de patrones de diseño de interfaz de usuario presentado por autor

Autor	Plataforma	Categorías	Patrones identificados
Theresa Neil	Móvil	Navegación (10); Formularios (7); Tablas y listas (8); Búsqueda, ordenamiento y filtración (14); Herramientas (7); Gráficos (8); Llamadas (8); Retroalimentación y ofrecimiento (5); Ayuda (3)	70
Jennifer Tidwell	Móvil y Web	Acciones del usuario (14); Organización del contenido (10); Navegación, indicadores y señalización (13); Organización de la página (13); Listas (12); Acciones y comandos (11); Árboles y gráficas (11); Formularios y controles (11); Medios sociales (12); Diseño móvil (11); Diseño visual (7)	125
Mari Sheibley	Móvil	Patrones de interfaz de usuario (22)	22

Capítulo 3. Aplicación de la metodología

Autor	Plataforma	Categorías	Patrones identificados
Anders Toxboe	Móvil y Web	Obtención de entrada (28); Navegación (25); Incorporación (9); Manejo de datos (11); Social (11); Diverso (5)	89
UNITiD	Móvil	Manejo de datos (16); Obtener entradas (8); Navegación (30); Notificaciones (7); Personalizar (6); Interacciones de pantalla (6); Social (4)	77
LG Developer	TV	Controles de interfaz de usuario (25)	25
Android TV	TV	Diálogo (1); Asistente (1); Configuraciones (1); Notificaciones (1)	4
Apple TV	TV	Barra de etiquetas (1); Tablas y Colecciones (3); Texto y búsqueda (3); Botones (2); Barra de navegación (1); Página de control (1); Indicadores de progreso (2); Segmentos de control (2); Alertas (2)	17

3.4.1. Clasificación de patrones de diseño de interfaz de usuario

A continuación, en la Tabla 3.3 se presenta el análisis realizado para relacionar a las plataformas con los patrones de diseño de interfaz de usuario que tienen en común y a los autores que proponen los patrones de diseño.

Tabla 3.3 Relación entre patrones de diseño de interfaz de usuario

Plataforma	Patrones de diseño de interfaz de usuario	Autores
Móvil	Acordeón, Asistente, Barra de etiquetas, Barra de herramientas, Barra de progreso, Búsqueda, Carrusel, Dashboard, Diálogos, Formularios, Galería, Indicadores	Jennifer Tidwell, Mari Sheibley,

Capítulo 3. Aplicación de la metodología

Plataforma	Patrones de diseño de interfaz de usuario	Autores
	de carga, Listas, <i>Login</i> , Navegación por etiquetas, Notificaciones, Menú contextual, Segmentos de control, Tablas	Anders Toxboe y UNITiD
TV	Acordeón, Asistente, Barra de etiquetas, Barra de navegación, Barra de progreso, Botones, Búsqueda, Colecciones, Divisor, Formulario, Indicadores de carga, Listas, Notificación, Página de control, Selector expandible, Tablas, Vista detalle, Vista dividida	LG Developer, Android TV y Apple TV
Web	Acordeón, Asistente, Barra de progreso, Búsqueda, Carrusel, Dashboard, Diálogos, Formularios, Galería, Indicadores de carga, Listas, <i>Login</i> , Navegación por etiquetas, Notificaciones, Menú contextual, Segmentos de control, Tablas	Jennifer Tidwell y Anders Toxboe

Para desarrollar una aplicación se tiene que determinar el dispositivo y el tipo de aplicación que se desea desarrollar con el propósito de facilitar al usuario la interacción con la aplicación. J. Engel et al. [56] y K. Klemisch et al. [57] mencionaron en sus trabajos a los contextos de uso, donde un contexto de uso se relaciona con ciertos patrones de diseño de acuerdo a su funcionalidad. Por esta razón, se realizó un análisis para reconocer patrones de diseño de interfaz de usuario en aplicaciones educativas para los teléfonos inteligentes, tabletas y televisiones inteligentes, y de esta forma establecer contextos de uso orientados al ámbito educativo:

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.

Capítulo 3. Aplicación de la metodología

- 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.

En la Tabla 3.4 se muestra la relación entre los contextos de uso, las plataformas y los patrones de interfaz de usuario. Teniendo como resultado que los contextos de uso facilitan la selección de patrones de diseño de interfaz de usuario, permitiendo un uso adecuado y teniendo la confianza que diferentes aplicaciones educativas los utilizan en su interfaz gráfica.

Tabla 3.4 Contextos de usos detectados al utilizar los patrones

Contexto de uso	Plataformas	Patrones de diseño de interfaz de usuario relacionados
Asistente	Móvil, TV y Web	Asistente
Cuestionario	Móvil y Web	Formulario
	TV	Asistente, Barra de progreso
Curso	Móvil	Acordeón
	TV	Vista detalle
	Web	Acordeón
Lista de contenido	Móvil y Web	Lista, Acordeón
	TV	Listas, Colecciones
Login	Móvil, TV y Web	Iniciar sesión
Menú	Móvil	Menú contextual, menú emergente
	TV	Selector expandible, Barra de etiquetas
	Web	Menú contextual
Multimedia	Móvil	Carrusel, Galería, Rejilla de imágenes
	TV	Galería, Rejilla de imágenes
	Web	Carrusel
Notificación	Móvil y Web	Diálogo
	TV	Notificación

Capítulo 3. Aplicación de la metodología

De acuerdo a los tipos de aplicaciones educativas seleccionados y definidos en la sección anterior se prosiguió a elegir un conjunto de patrones de diseño de interfaz de usuario que se apliquen al desarrollo de aplicaciones educativas y cumplan con los objetivos de una teoría del aprendizaje. La Tabla 3.5 describe catorce patrones de diseño comúnmente empleados para desarrollar los tres tipos de aplicaciones educativas mencionadas anteriormente.

Tabla 3.5 Relación entre los patrones de diseño de IU y las aplicaciones educativas

Patrones de diseño de IU	Descripción	Blog	MOOC	Wiki
Acordeón	Expande y contrae información vertical u horizontalmente al hacer clic, mostrando las secciones más importantes y visualizando su contenido.	Sí	Sí	Sí
Asistente de configuración	Permite al usuario alcanzar una meta a través de una serie de pasos. El Asistente de configuración tiene como objetivo facilitar y racionalizar un proceso.	No	Sí	No
Barra de progreso	Permite al usuario ver el estado de una acción o una petición que realizará el sistema.	Sí	Sí	No
Carrusel	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

Capítulo 3. Aplicación de la metodología

Patrones de diseño de IU	Descripción	Blog	MOOC	Wiki
Diálogo	Recibe una respuesta explícita del usuario. Aunque un Diálogo no ocupa toda la pantalla, requiere la respuesta del usuario para continuar.	Sí	Sí	No
Formulario	Recibe información del usuario, la cual es interpretada por la aplicación para realizar alguna tarea específica.	Sí	Sí	No
Lista	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
Menú Contextual	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
Notificación	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
Tabla	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

Capítulo 3. Aplicación de la metodología

3.4.2. Patrones de diseño individuales de interfaz de usuario

En esta sección se menciona un conjunto de patrones de diseño de interfaz de usuario que permiten el desarrollo de las aplicaciones educativas y además se establece una posición que es relevante para que cumplan adecuadamente con sus propósitos. En este proyecto se tienen contempladas cinco posiciones (arriba, abajo, izquierda, derecha, centro), donde estas posiciones sirven para incorporar los patrones de diseño de IU, así como se muestra en la Figura 3.4.

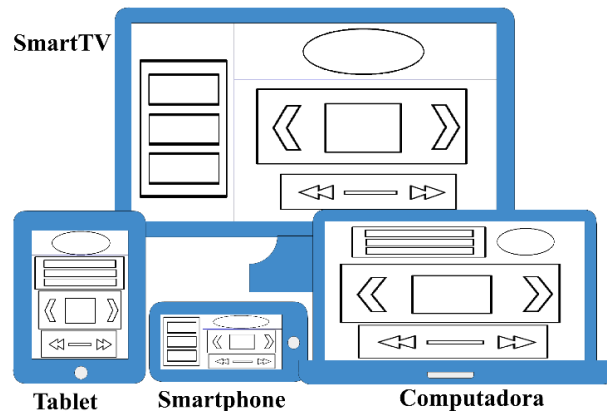


Figura 3.4 Vista de elementos multi-dispositivo

Estas cinco posiciones se visualizan individualmente o simultáneamente dependiendo de la necesidad de las aplicaciones generadas. Esta variedad de combinaciones genera un conjunto de plantillas, las cuales se presentan en la Figura 3.5, gracias a estas combinaciones se aumenta la variedad de diseño entre las aplicaciones educativas y se asigna a cada patrón de diseño una posición que utiliza para justificar su uso. En total se realizaron 16 diferentes plantillas que surgieron a partir de la combinación de las cinco posiciones que se mencionaron anteriormente.

Capítulo 3. Aplicación de la metodología

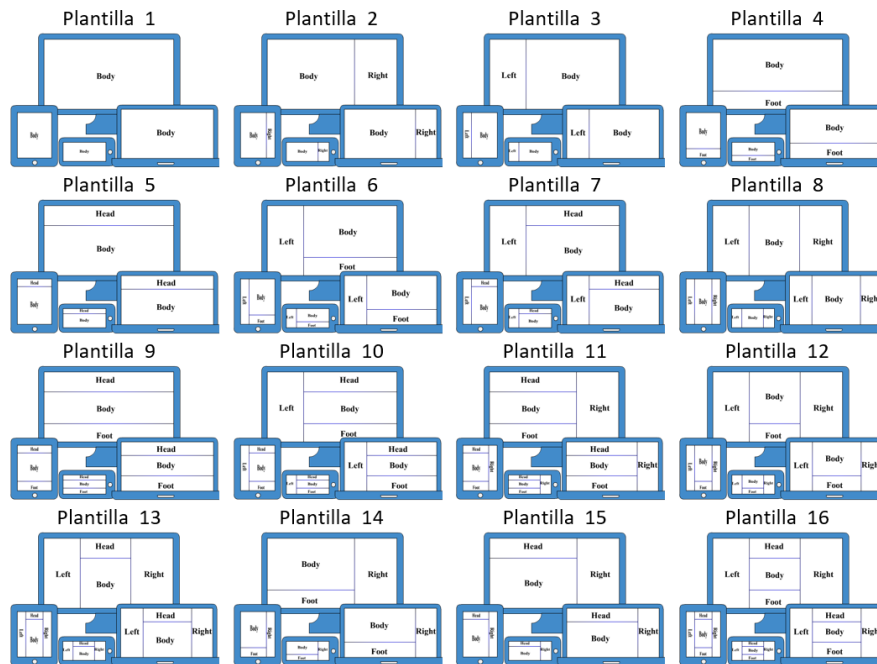


Figura 3.5 Plantillas generadas a partir de las cinco posiciones

En la Tabla 3.6 se muestra la relación entre los patrones de diseño de interfaz de usuario y las posiciones en donde es posible localizarlos, de igual forma se mencionan los patrones de diseño que suelen ocuparse al seleccionar el patrón de diseño.

Tabla 3.6 Relación de patrones de diseño de interfaz de usuario con las posiciones

Patrones de diseño de IU	Arriba	Izquierda	Centro	Derecha	Abajo	Patrones de diseño IU relacionados
Acordeón	✗	✗	✓	✗	✗	Tarjetas, video, lista
Asistente de configuración	✗	✗	✓	✗	✗	Asistente de configuración, <i>Dashboard</i> , formulario, lista, tabla, tarjetas
Barra de etiquetas	✓	✗	✓	✗	✓	Acordeón, video, tarjetas, lista

Capítulo 3. Aplicación de la metodología

Patrones de diseño de IU	Arriba	Izquierda	Centro	Derecha	Abajo	Patrones de diseño IU relacionados
Barra de progreso	x	x	✓	x	x	Asistente de configuración
Buscador	✓	✓	x	x	x	<i>Login</i>
Carrusel	x	x	✓	x	x	Video
<i>Dashboard</i>	x	x	✓	x	x	Tarjetas
Diálogo	x	x	✓	x	x	Formulario, menú contextual
Formulario	x	x	✓	x	x	Diálogo, lista, notificación
Lista	x	✓	✓	✓	x	Video, paginación
<i>Login</i>	✓	x	✓	x	x	
Maestro detalle	✓	✓	x	✓	✓	Lista, acordeón, barra de etiquetas, paginación, <i>login</i> , menú contextual
Menú contextual	x	x	✓	x	x	Formulario, menú contextual
Notificación	x	x	✓	x	x	Formulario, menú contextual
Paginación	x	x	✓	x	✓	Lista, tabla, buscador
<i>Splashscreen</i>	x	x	✓	x	x	

Capítulo 3. Aplicación de la metodología

Patrones de diseño de IU	Arriba	Izquierda	Centro	Derecha	Abajo	Patrones de diseño IU relacionados
Tabla	x	x	✓	x	x	buscador
Tarjetas	x	✓	✓	✓	x	Paginación
Video	x	x	✓	x	x	Menú contextual

3.4.3. Patrones de diseño compuestos de interfaz de usuario

Como es posible notar en la Tabla 3.7, los patrones de diseño de interfaz de usuario permiten la composición de patrones, esto quiere decir que un patrón de diseño tiene la capacidad de combinarse o en su defecto venir acompañado de uno o más patrones de diseño.

Tabla 3.7 Patrones de diseño compuestos de interfaces de usuario

Patrón de diseño de IU compuestos	Patrones de diseño de IU que lo conforman	Tipo de patrón compuesto	Descripción
Carrusel de videos	Carrusel, Video	Combinado	Organiza un conjunto de elementos en una línea horizontal, proporciona controles de navegación y permite a los usuarios seleccionar una opción a través de un video.
Navegación básica	<i>Login,</i> <i>Splashscreen,</i> <i>Dashboard,</i> Lista, Video	Navegatorio	Organiza la navegación necesaria que se necesita al incluir a los patrones de diseño individuales que contiene, facilitando la interacción entre ellos en las aplicaciones educativas.

Capítulo 3. Aplicación de la metodología

Patrón de diseño de IU compuestos	Patrones de diseño de IU que lo conforman	Tipo de patrón compuesto	Descripción
<i>Dashboard con lista</i>	<i>Dashboard, Lista</i>	Ordenamiento	Permite establecer la ubicación adecuada de un <i>dashboard</i> ante el uso de una lista en una aplicación educativa, facilitando la usabilidad de los patrones sin importar el dispositivo que lo utiliza.
<i>Dashboard con video</i>	<i>Dashboard, Video</i>	Ordenamiento	Permite establecer la ubicación adecuada de un <i>dashboard</i> ante el uso del patrón de videos en una aplicación educativa.
Lista de videos	Lista, Video	Combinado	Guía al usuario a través de un conjunto de información que sea relevante utilizando un conjunto de videos.

Los patrones de diseño compuestos se diferencian de los individuales, porque los patrones compuestos tienen la capacidad de incluir las ventajas que ofrece uno o más patrones individuales, permitiendo la creación o en su defecto la organización adecuada para generar un nuevo patrón de diseño de interfaz de usuario.

3.5. Proceso de identificación de patrones de diseño de interfaz de usuario

En esta sección se describe el proceso de identificación utilizado para el procesamiento de imágenes. Para lograr identificar los patrones de diseño de interfaz de usuario se diseñaron un conjunto de iconos que representan a los patrones compuestos y los patrones individuales.

Capítulo 3. Aplicación de la metodología

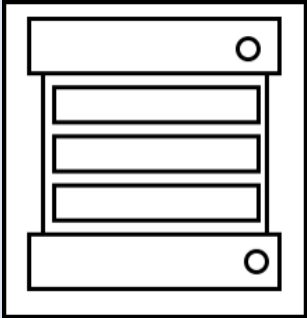
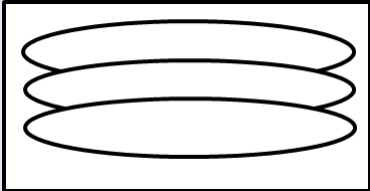




De igual forma se determinaron dos métodos para el proceso de identificación: El primero es utilizar un conjunto de reglas representadas a través del lenguaje RuleML y la segunda es a partir de un conjunto de características que se ingresan a una red neuronal.

3.5.1. Iconografía de los patrones de diseño de interfaz de usuario

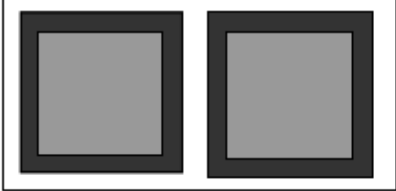
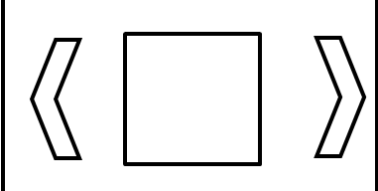
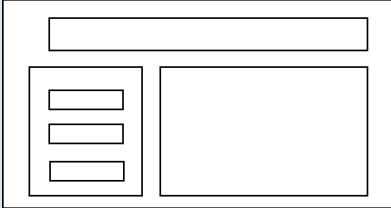
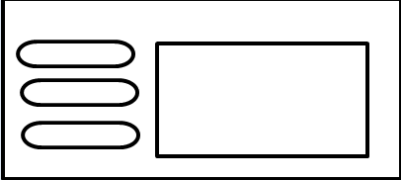
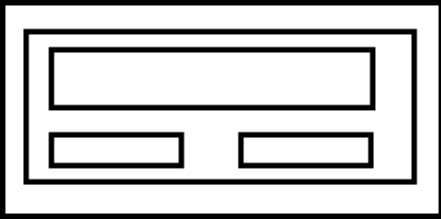
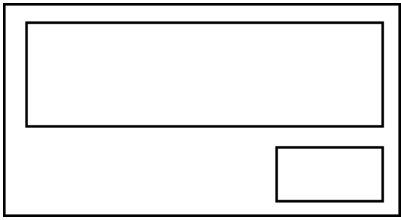
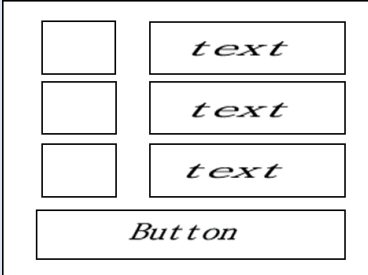
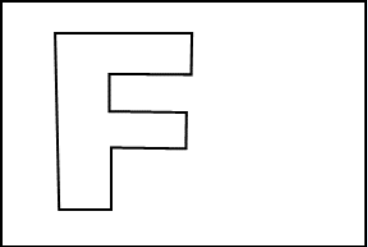
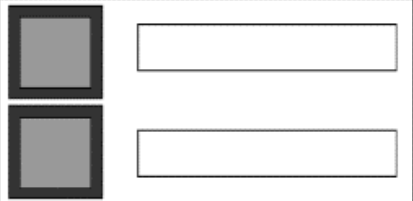
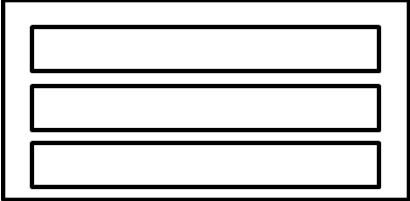
Para el proceso de identificación se utilizó un conjunto de iconos de los patrones de diseño, estos iconos representan a los patrones individuales y compuestos que se encuentran registrados en EduGene.

Los patrones de diseño individuales utilizan dos representaciones, el primer icono se utiliza para identificar los patrones individuales y el segundo icono se ocupa para la relación con los patrones compuestos. En la Tabla 3.8 se muestran los patrones de diseño individuales con su iconografía correspondiente.

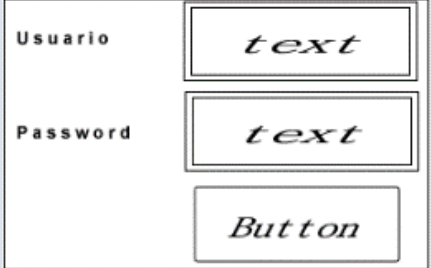
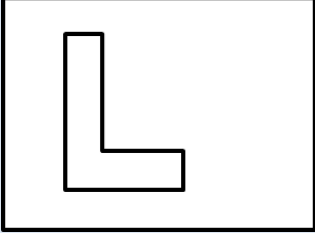
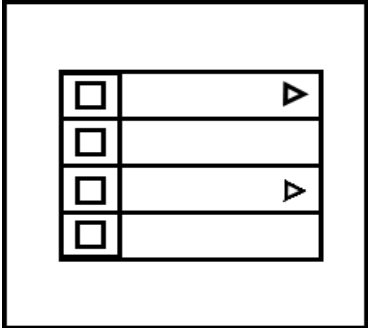

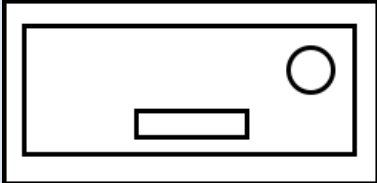

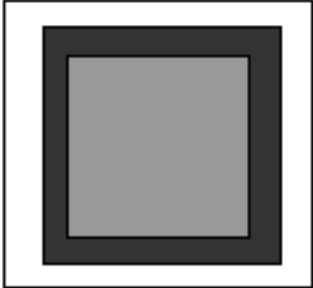
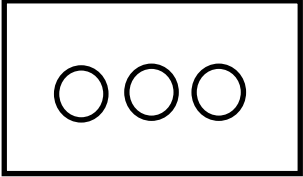
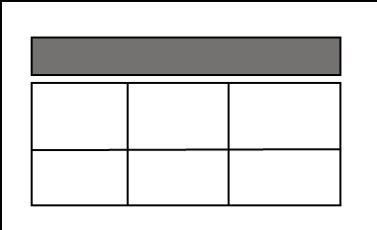
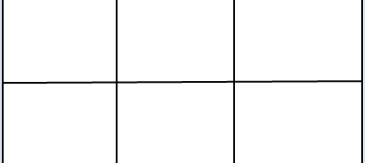
Tabla 3.8 Tabla de la iconografía de los patrones individuales de diseño de interfaz de usuario

Patrones individuales de diseño de IU	Icono para patrones individuales	Icono para patrones compuestos
Acordeón		
Asistente de configuración		
Barra de progreso		

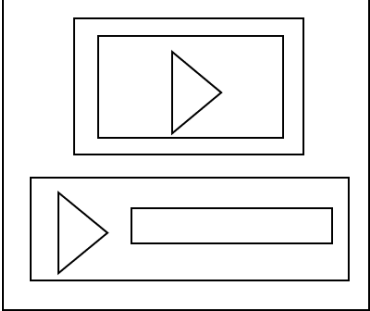
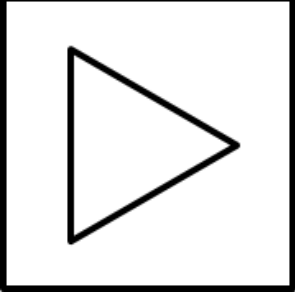
Capítulo 3. Aplicación de la metodología

Patrones individuales de diseño de IU	Icono para patrones individuales	Icono para patrones compuestos
Carrusel		
<i>Dashboard</i>		
Diálogo		
Formulario		
Lista		

Capítulo 3. Aplicación de la metodología

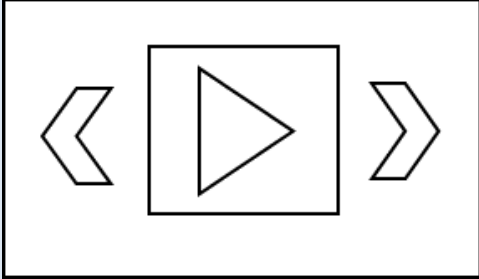
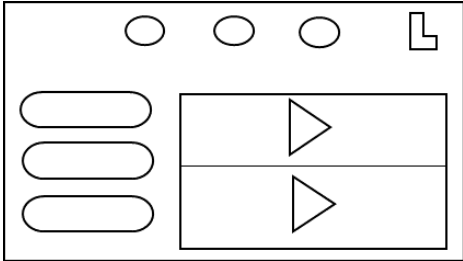
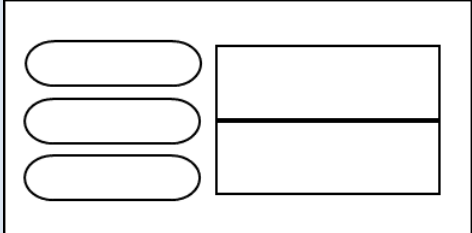
Patrones individuales de diseño de IU	Icono para patrones individuales	Icono para patrones compuestos
<i>Login</i>		
Menú contextual		
Notificación		
<i>Splashscreen</i>		
Tabla		

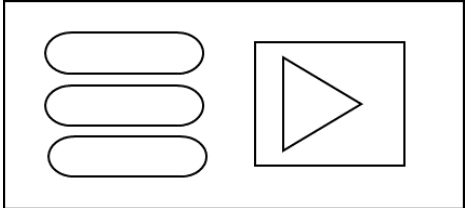
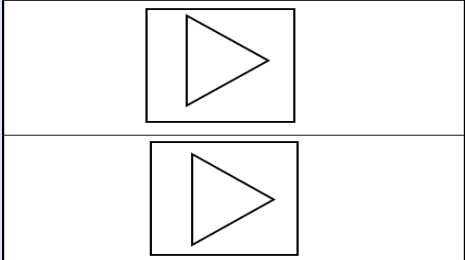
Capítulo 3. Aplicación de la metodología

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

Por otro lado, se tiene la iconografía de los patrones compuestos de diseño de interfaz de usuario predefinidos en EduGene que se presentan en la Tabla 3.9.

Tabla 3.9 Tabla de la iconografía de los patrones compuestos de diseño de interfaz de usuario

Patrones compuestos de diseño de IU	Icono
Carrusel de videos	
Navegación básica	
<i>Dashboard con lista</i>	

Patrones compuestos de diseño de IU	Icono
<i>Dashboard con video</i>	
Lista de videos	

3.5.2. Reconocimiento de patrones de diseño en imágenes usando un proceso ideal

Al hablar de proceso de ideal se hace referencia a las imágenes que utilizaron un software que permita dibujar las figuras de los iconos. Es decir, que la imagen que se ingresa a EduGene tiene sus figuras correctamente dibujadas sin unir líneas y sin líneas incompletas.

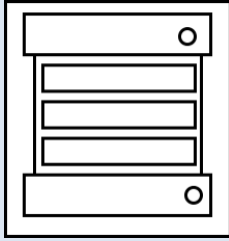

Por tal razón, para el proceso de identificación de patrones de diseño en una imagen utilizando un proceso ideal se utilizó el lenguaje RuleML para establecer las reglas necesarias para reconocer los patrones de diseño de interfaz de usuario.

3.5.2.1. Lenguaje RuleML


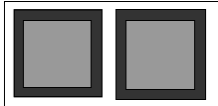
La finalidad de utilizar el lenguaje RuleML es con la finalidad de identificar un conjunto de reglas que por medio de un grupo de especificaciones permita establecer un identificador para cada imagen que se ingrese a la plataforma. En total se realizó un conjunto de 28 reglas para identificar a los 14 patrones individuales de diseño de interfaz de usuario que se presentan en la Tabla 3.10 y los 14 patrones individuales para patrones compuestos de diseño de interfaz de usuario que se presentan en la Tabla 3.11.

Capítulo 3. Aplicación de la metodología

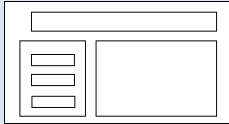
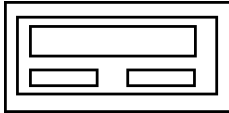
Tabla 3.10 Conjunto de reglas en RuleML para los patrones individuales de diseño de interfaz de usuario

Patrones individuales de diseño de IU	Icono	RuleML
<p>Acordeón</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>Asistente de configuración</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> </Implies> </pre>

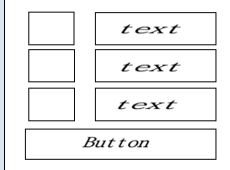
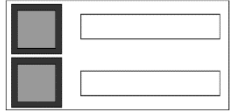
Capítulo 3. Aplicación de la metodología

Patrones individuales de diseño de IU	Icono	RuleML
<p>Barra de progreso</p>		<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>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>
<p>Carrusel</p>		<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>

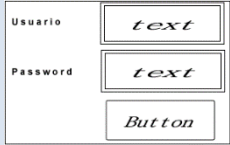
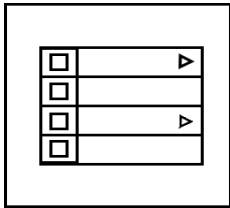
Capítulo 3. Aplicación de la metodología

Patrones individuales de diseño de IU	Icono	RuleML
<p><i>Dashboard</i></p>		<pre> <Implies> <If> <And> <Atom> <Var>Forma externa</var> <Rel>Rectángulo</Rel> </Atom> <Atom> <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>
<p>Diálogo</p>		<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>

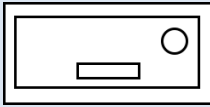
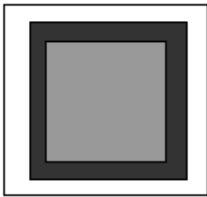
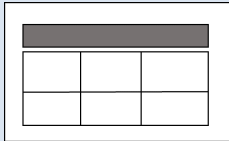
Capítulo 3. Aplicación de la metodología

Patrones individuales de diseño de IU	Icono	RuleML
Formulario		<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> <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>
Lista		<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>

Capítulo 3. Aplicación de la metodología

Patrones individuales de diseño de IU	Icono	RuleML
<p>Login</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> <Rel>Textos</Rel> <Ind>Cinco</Ind> </Atom> </And> </If> <Then> <Atom> <Var>Forma externa</Var> <Var>Figuras internas</Var> <Rel>Login</Rel> </Atom> </Then> </Implies> </pre>
<p>Menú contextual</p>		<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>

Capítulo 3. Aplicación de la metodología

Patrones individuales de diseño de IU	Icono	RuleML
<p>Notificación</p>		<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>
<p>Splashscreen</p>		<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>
<p>Tabla</p>		<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>

Capítulo 3. Aplicación de la metodología

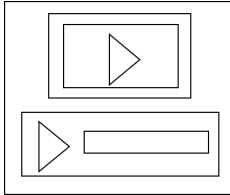
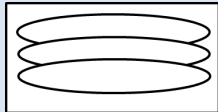


Patrones individuales de diseño de IU	Icono	RuleML
		<pre> </Atom> </And> </If> <Then> <Atom> <Var>Forma externa</Var> <Var>Figuras internas</var> <Rel>Tabla</Rel> </Atom> </Then> </Implies> </pre>
<p>Video</p>		<pre> <Implies> <If> <And> <Atom> <Var>Forma externa</var> <Rel>Cuadrado</Rel> </Atom> <Atom> <Var>Figuras internas</var> <Rel>Triángulo</Rel> <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>

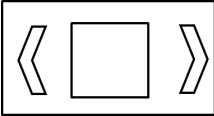
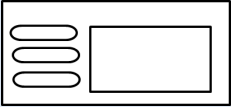
Tabla 3.11 Conjunto de reglas en RuleML para los patrones individuales para patrones compuestos de diseño de interfaz de usuario

Patrones individuales para patrones compuestos de diseño de IU	Icono	RuleML
<p>Acordeón</p>		<pre> <Implies> <If> <And> <Atom> <Var>Forma externa</var> <Rel>Rectángulo</Rel> </Atom> </And> </If> </pre>

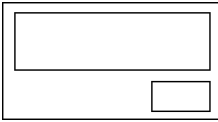
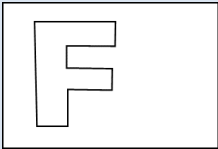
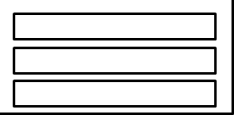
Capítulo 3. Aplicación de la metodología

Patrones individuales para patrones compuestos de diseño de IU	Icono	RuleML
		<pre> </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>
<p>Asistente de configuración</p>		<pre> <Implies> <If> <And> <Atom> <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>
<p>Barra de progreso</p>		<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>

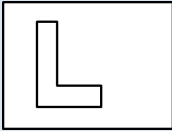

Capítulo 3. Aplicación de la metodología

Patrones individuales para patrones compuestos de diseño de IU	Icono	RuleML
Carrusel		<pre> </Then> </Implies> <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> </Then> </Implies> </pre>
<i>Dashboard</i>		<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>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>



Capítulo 3. Aplicación de la metodología

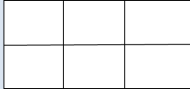
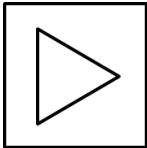
Patrones individuales para patrones compuestos de diseño de IU	Icono	RuleML
Diálogo		<pre> <Implies> <If> <And> <Atom> <Var>Forma externa</var> <Rel>Rectángulo</Rel> </Atom> <Atom> <Var>Figuras internas</var> <Rel>Rectángulos</Rel> <Ind>Dos</Ind> <Ind>Uno grande</Ind> <Ind>Uno chico</Ind> </Atom> </And> </If> <Then> <Atom> <Var>Forma externa</Var> <Var>Figuras internas</Var> <Rel>Diálogo</Rel> </Atom> </Then> </Implies> </pre>
Formulario		<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>
Lista		<pre> <Implies> <If> <And> <Atom> <Var>Forma externa</var> </pre>

Capítulo 3. Aplicación de la metodología

Patrones individuales para patrones compuestos de diseño de IU	Icono	RuleML
		<pre> <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>
<i>Login</i>		<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>Login</Rel> </Atom> </Then> </Implies> </pre>
Menú contextual		<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>

Capítulo 3. Aplicación de la metodología

Patrones individuales para patrones compuestos de diseño de IU	Icono	RuleML
		<pre> </And> </If> <Then> <Atom> <Var>Forma externa</Var> <Var>Figuras internas</Var> <Rel>Menú contextual</Rel> </Atom> </Then> </Implies> </pre>
Notificación		<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>
<i>Splashscreen</i>		<pre> <Implies> <If> <And> <Atom> <Var>Forma externa</var> <Rel>Rectángulo</Rel> </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>

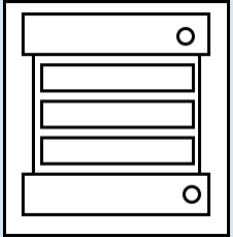
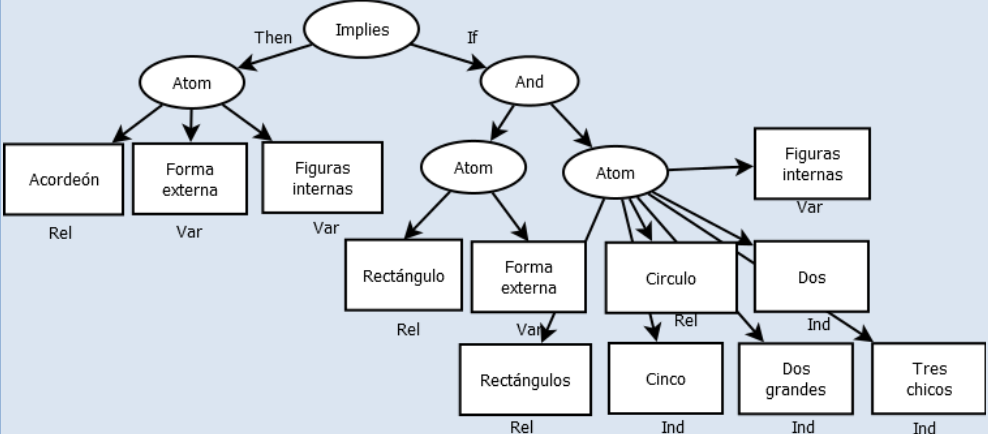

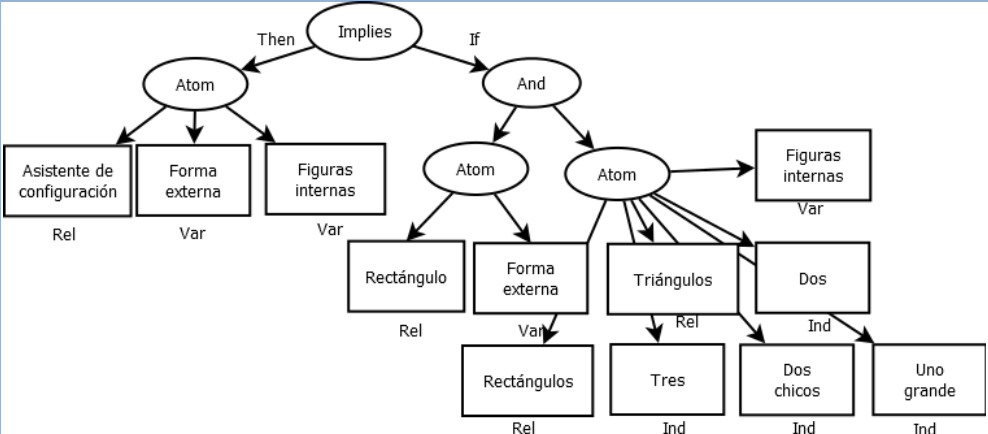
Patrones individuales para patrones compuestos de diseño de IU	Icono	RuleML
Tabla		<pre> <Implies> <If> <And> <Atom> <Var>Forma externa</var> <Rel>Rectángulos</Rel> <Ind>Seis juntos</Ind> </Atom> </And> </If> <Then> <Atom> <Var>Forma externa</Var> <Rel>Tabla</Rel> </Atom> </Then> </Implies> </pre>
Video		<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>


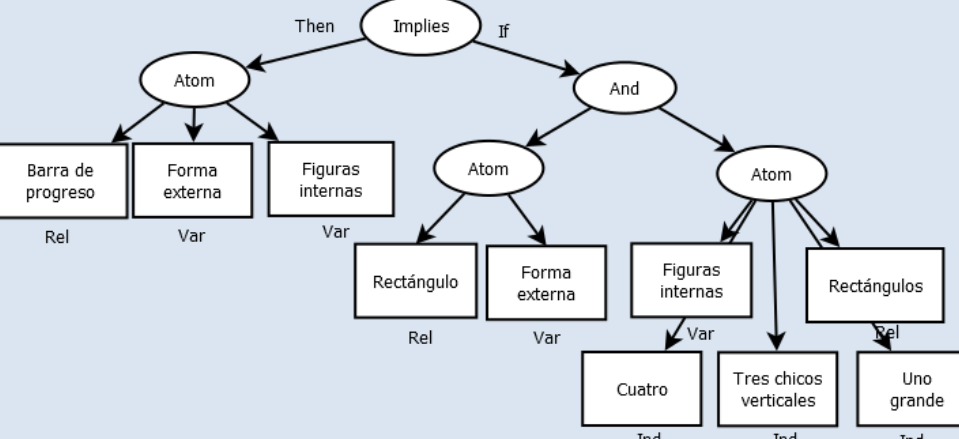
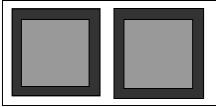
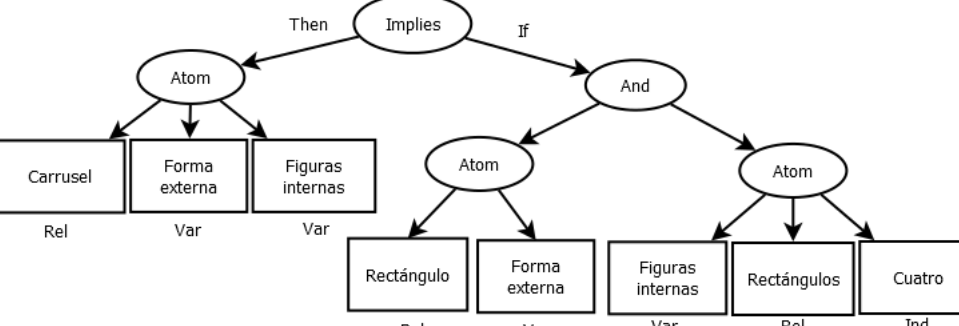
3.5.2.2. Árbol parcial de reglas

El conjunto de reglas creadas en el lenguaje RuleML se representan en un árbol desde su versión 1.0 [58]. Con base en las reglas descritas anteriormente en RuleML, en la Tabla 3.12 se presentan los diagramas de árbol para los patrones individuales y en la Tabla 3.13 se encuentran los diagramas para los patrones individuales de los patrones compuestos.

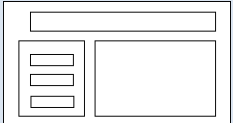
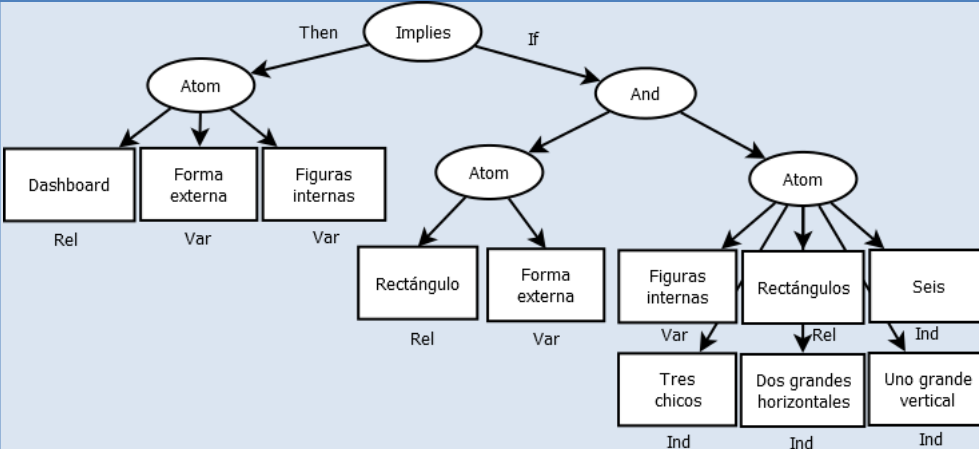
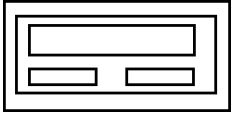
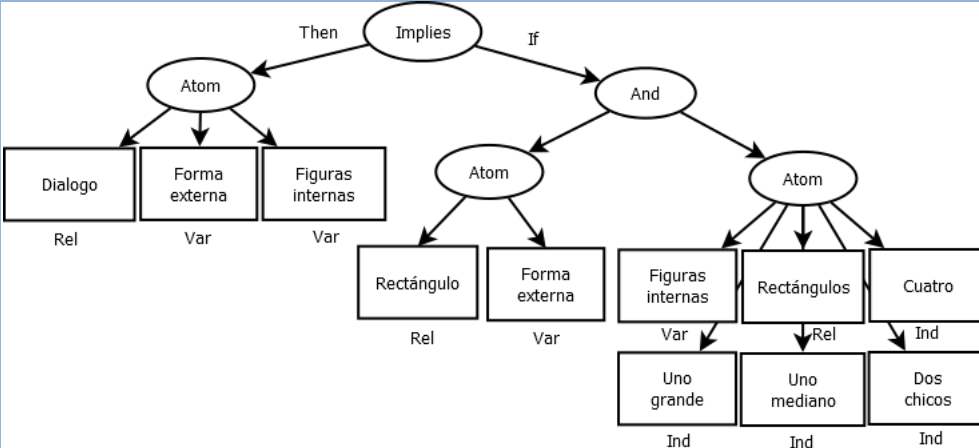
Capítulo 3. Aplicación de la metodología

Tabla 3.12 Conjunto de diagramas de árbol para los patrones individuales de diseño de interfaz de usuario

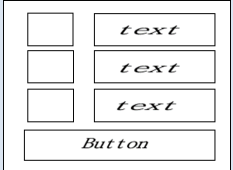
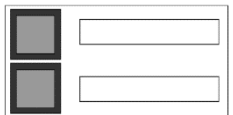
Patrones individuales de diseño de IU	Icono	Árbol parcial de reglas
<p>Acordeón</p>		
<p>Asistente de configuración</p>		

Patrones individuales de diseño de IU	Icono	Árbol parcial de reglas
<p>Barra de progreso</p>		
<p>Carrusel</p>		

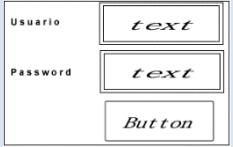
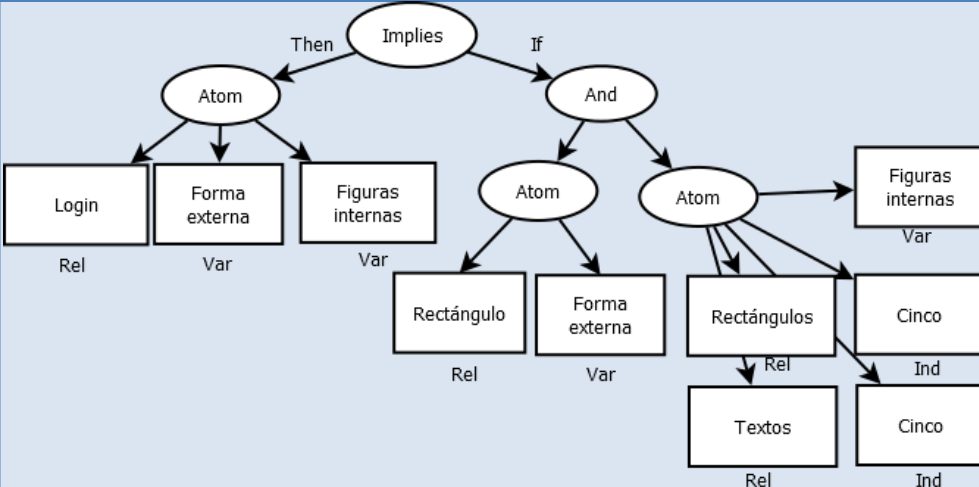
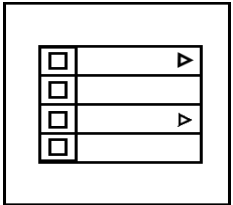
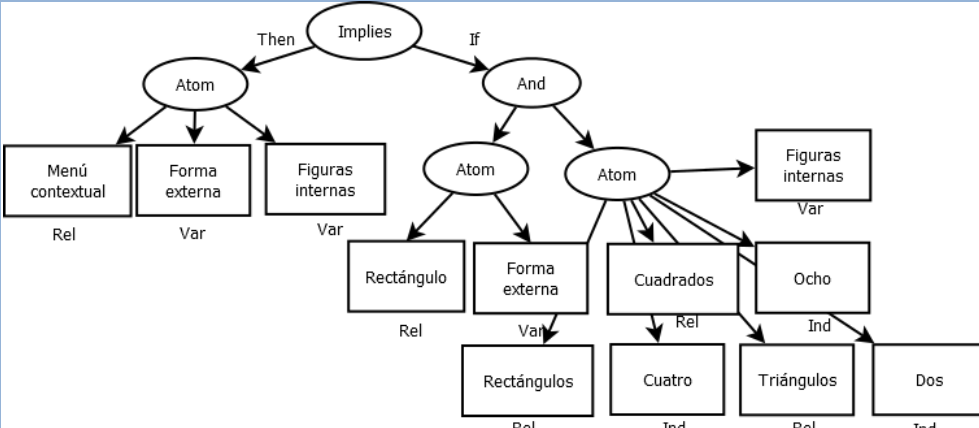
Capítulo 3. Aplicación de la metodología

Patrones individuales de diseño de IU	Icono	Árbol parcial de reglas
<p><i>Dashboard</i></p>		
<p>Diálogo</p>		

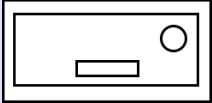
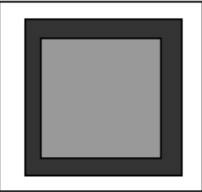
Capítulo 3. Aplicación de la metodología

Patrones individuales de diseño de IU	Icono	Árbol parcial de reglas
<p>Formulario</p>		
<p>Lista</p>		

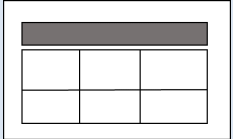
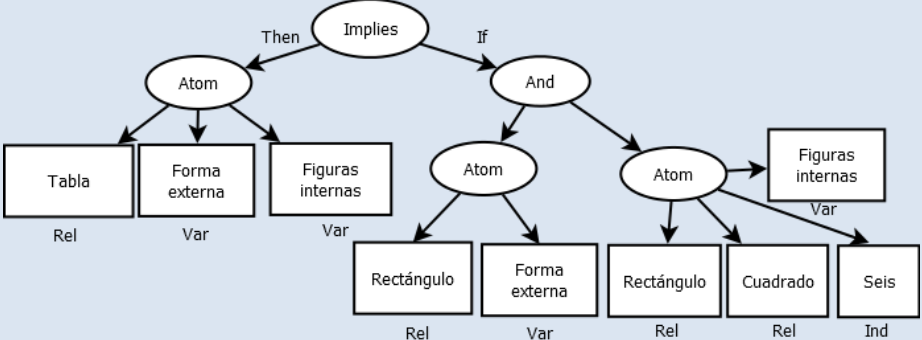
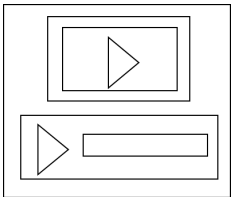
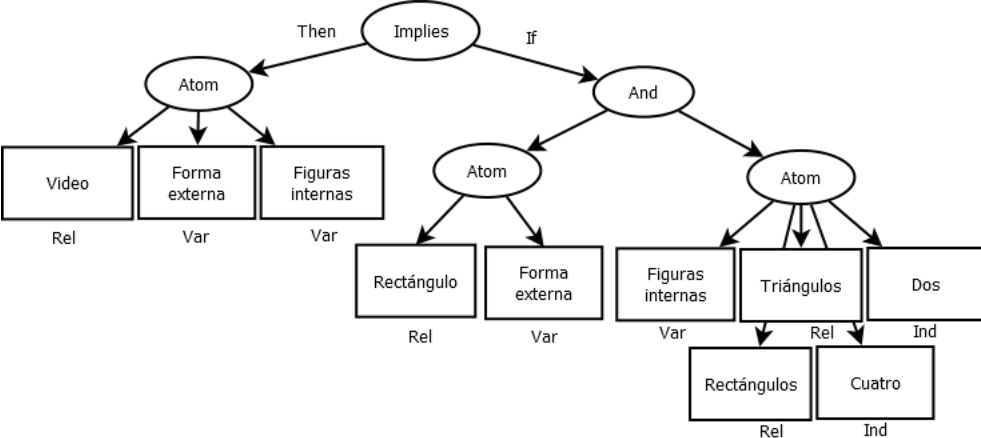
Capítulo 3. Aplicación de la metodología

Patrones individuales de diseño de IU	Icono	Árbol parcial de reglas
<p>Login</p>		
<p>Menú contextual</p>		

Capítulo 3. Aplicación de la metodología

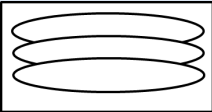
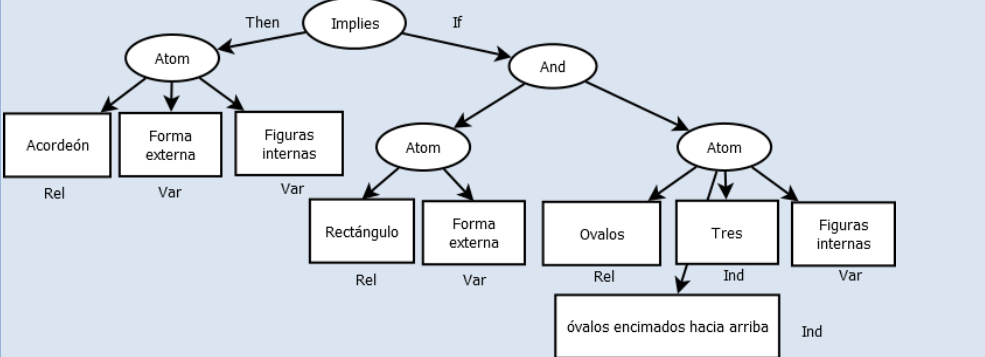

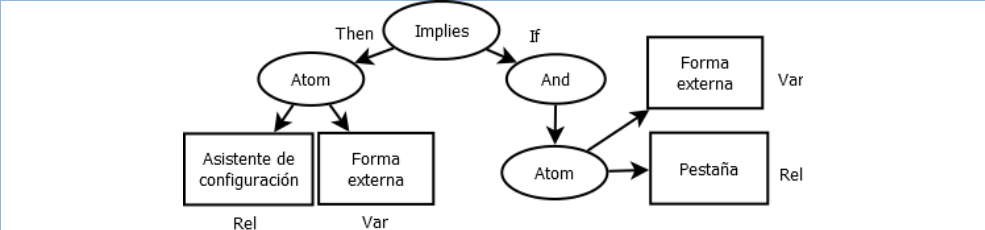
Patrones individuales de diseño de IU	Icono	Árbol parcial de reglas
<p>Notificación</p>		
<p><i>Splashscreen</i></p>		

Capítulo 3. Aplicación de la metodología


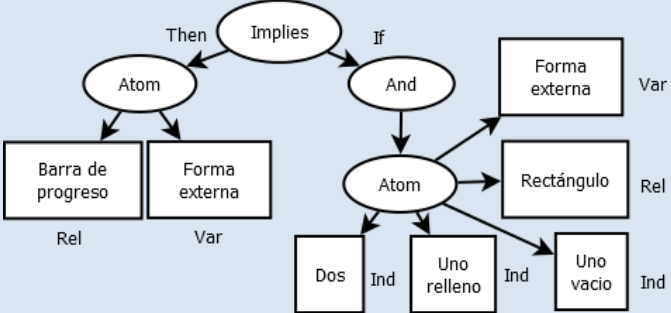
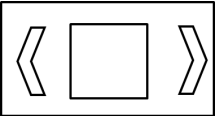
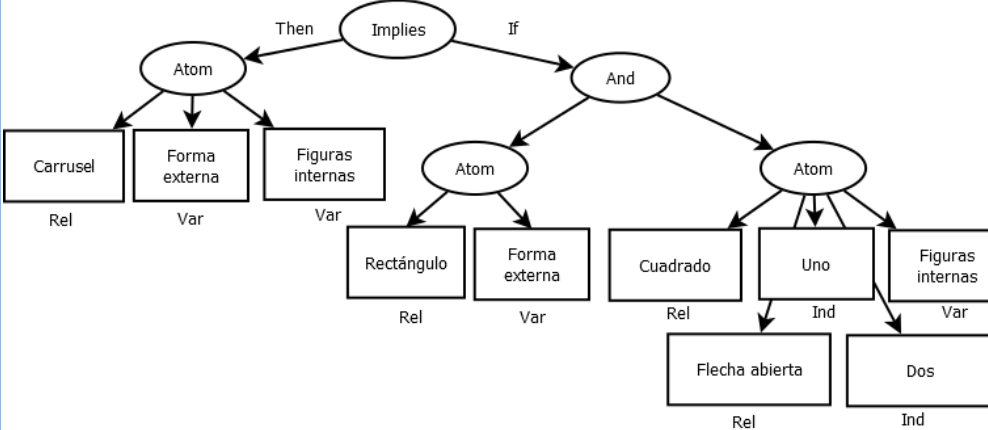
Patrones individuales de diseño de IU	Icono	Árbol parcial de reglas
<p>Tabla</p>		
<p>Video</p>		

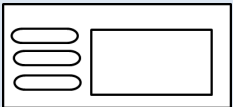
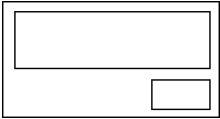
Capítulo 3. Aplicación de la metodología

Tabla 3.13 Conjunto de diagramas de árbol para los patrones individuales para patrones compuestos de diseño de interfaz de usuario

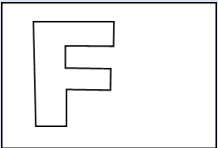
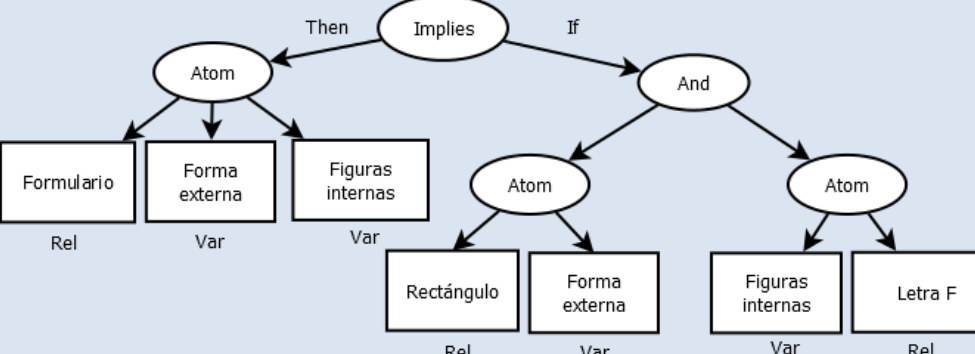
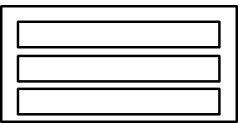
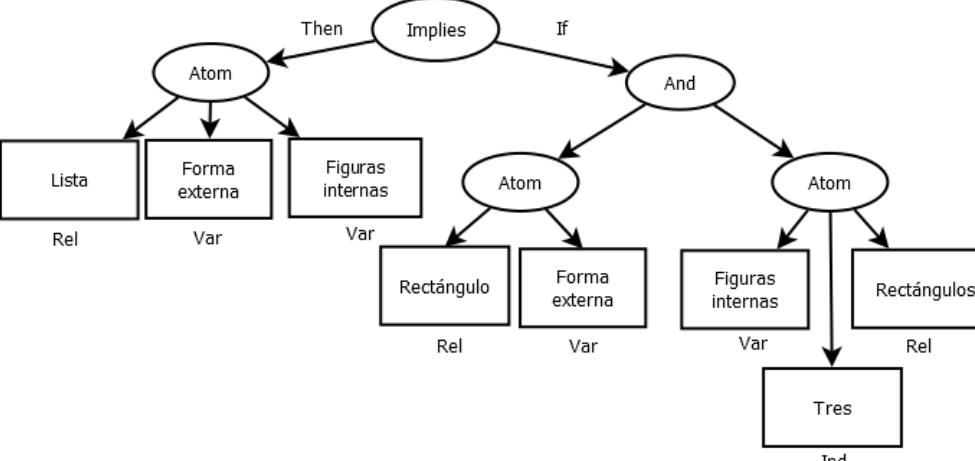
Patrones individuales para patrones compuestos de diseño de IU	Icono	Árbol parcial de reglas
<p>Acordeón</p>		
<p>Asistente de configuración</p>		

Capítulo 3. Aplicación de la metodología

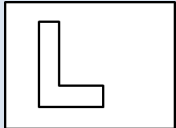
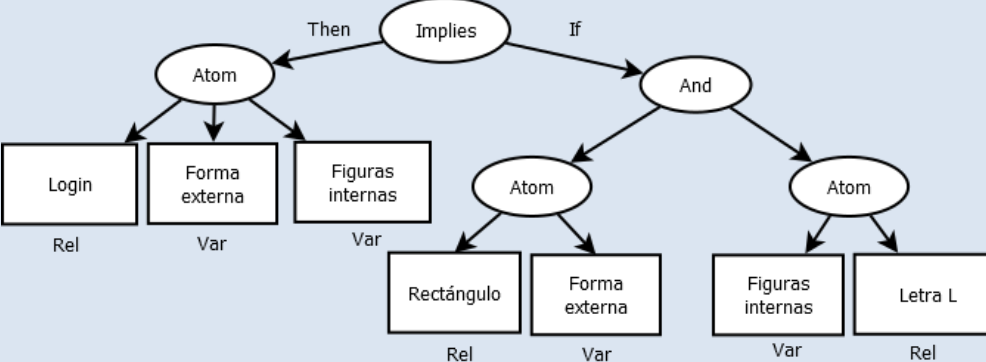

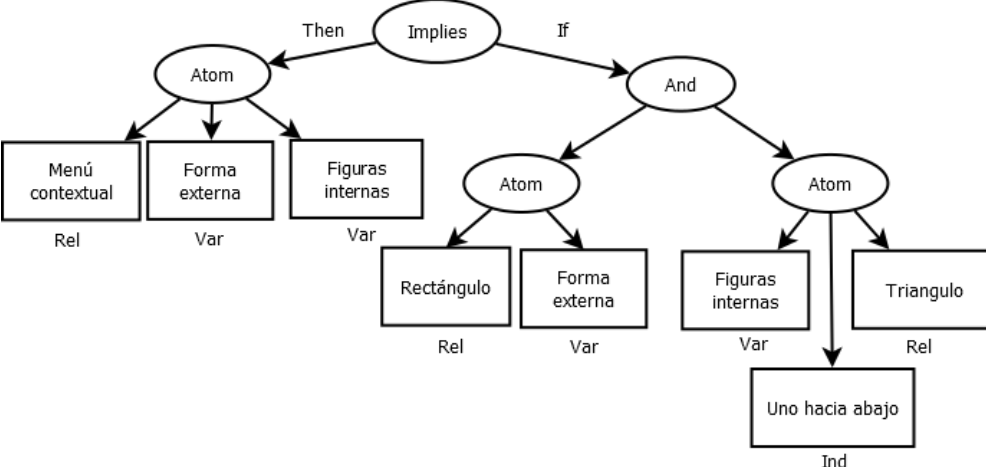
Patrones individuales para patrones compuestos de diseño de IU	Icono	Árbol parcial de reglas
<p>Barra de progreso</p>		
<p>Carrusel</p>		

Patrones individuales para patrones compuestos de diseño de IU	Icono	Árbol parcial de reglas
<p>Dashboard</p>		
<p>Diálogo</p>		


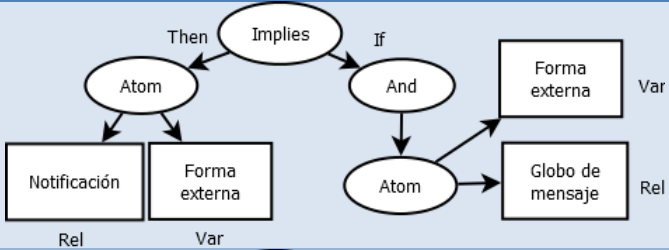
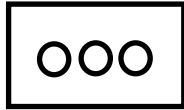
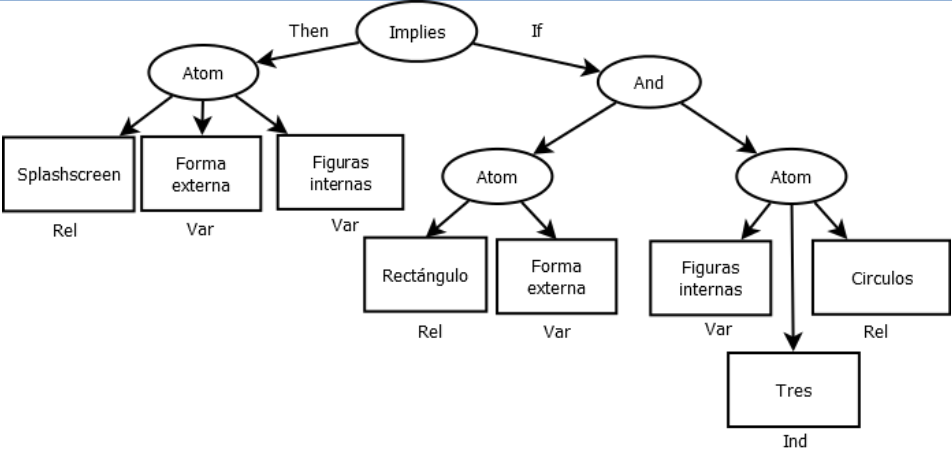
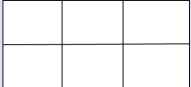
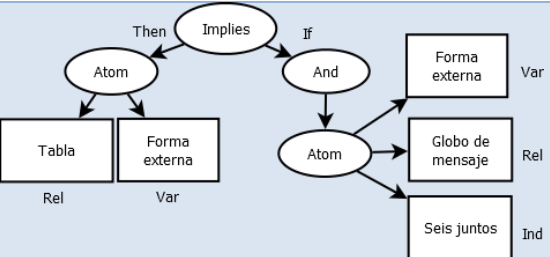
Capítulo 3. Aplicación de la metodología

Patrones individuales para patrones compuestos de diseño de IU	Icono	Árbol parcial de reglas
<p>Formulario</p>		
<p>Lista</p>		

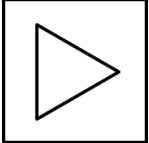
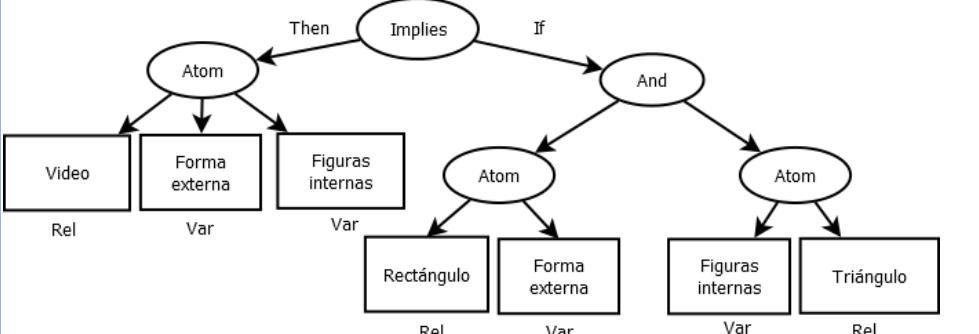
Capítulo 3. Aplicación de la metodología

Patrones individuales para patrones compuestos de diseño de IU	Icono	Árbol parcial de reglas
<p><i>Login</i></p>		
<p>Menú contextual</p>		

Capítulo 3. Aplicación de la metodología

Patrones individuales para patrones compuestos de diseño de IU	Icono	Árbol parcial de reglas
Notificación		
Splashscreen		
Tabla		

Capítulo 3. Aplicación de la metodología

Patrones individuales para patrones compuestos de diseño de IU	Icono	Árbol parcial de reglas
Video		 <p>The diagram is a partial rule tree for the Video pattern. At the top is an oval node labeled 'Implies'. Two arrows originate from 'Implies': one labeled 'Then' pointing to an oval node 'Atom', and one labeled 'If' pointing to an oval node 'And'. The 'Atom' node under 'Then' has three arrows pointing to rectangular nodes: 'Video' (labeled 'Rel' below), 'Forma externa' (labeled 'Var' below), and 'Figuras internas' (labeled 'Var' below). The 'And' node has two arrows pointing to oval nodes 'Atom'. The left 'Atom' node has two arrows pointing to rectangular nodes: 'Rectángulo' (labeled 'Rel' below) and 'Forma externa' (labeled 'Var' below). The right 'Atom' node has two arrows pointing to rectangular nodes: 'Figuras internas' (labeled 'Var' below) and 'Triángulo' (labeled 'Rel' below).</p>

Capítulo 3. Aplicación de la metodología

3.5.3. Reconocimiento de patrones de diseño en imágenes usando un proceso no ideal

A diferencia de un proceso ideal, en el proceso no ideal se tienen tres inconvenientes al realizar la identificación de los patrones de diseño en las imágenes. El primer inconveniente proviene al realizar el dibujo, porque cuando se utiliza una tableta digitalizadora las figuras que se crean son confusas, lo que provoca que sea más difícil de identificar y provoque un resultado erróneo. El segundo inconveniente se encuentra cuando algunas líneas de las figuras se unen, esto ocasiona que sea una nueva figura y en lugar de determinar dos figuras solo identifica una. Por último, el tercer inconveniente sucede al realizar el dibujo y las líneas se encuentran incompletas o mal delineadas generando que la figura no se identifique correctamente.

Ante tal problema, se procedió a establecer un conjunto de 18 características que permiten identificar con mayor precisión los patrones de diseño. Por tal razón, se diseñó una red neuronal en Matlab para ingresar 18 características de un conjunto de 50 bosquejos de los iconos de los patrones de diseño para entrenar la red neuronal. La red neuronal cuenta con 6 neuronas y realiza una propagación hacia delante, entre las 18 características se encuentran: los momentos de Hu, el total de objetos encontrados y el número específico de figuras (triángulo, rectángulo, cuadrado, entre otros). Las características se normalizaron a excepción de los momentos de Hu para obtener una mayor precisión.

3.6. Generador de aplicaciones

El siguiente apartado hace referencias a las tecnologías utilizadas en el componente generador de código, así como los componentes que lo conforman y la descripción de su funcionamiento.

3.6.1. Descripción del generador de aplicaciones

El generador de aplicaciones es un componente que utiliza el lenguaje de programación PHP que permite ejecutarse en un entorno Web y facilita la tarea de utilizar el componente como un servicio *RESTFul*.

3.6.2. Archivo basado en XML

En esta sección se describe la estructura del archivo XML que se utiliza para construir las aplicaciones educativas. En la Figura 3.6 se presenta cómo se compone el archivo XML, a continuación se describen las partes que dividen al archivo:

Capítulo 3. Aplicación de la metodología

- **Configuración del proyecto:** En esta sección del archivo XML se presentan 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 ingresan los datos referentes a los alcances de la aplicación, en otras palabras, se refiere a los dispositivos, las plataformas, la orientación y las plantillas que se utilizaron en el proyecto.
- **Navegación:** En esta sección se ingresan todas las capas del proyecto, en donde cada capa hace referencia a un *layout* en el caso de Android™, una página en el caso de Web, entre otras. Asimismo, se incluye la navegación que se utilizó en la aplicación lo cual permitió una apropiada comunicación entre capas.

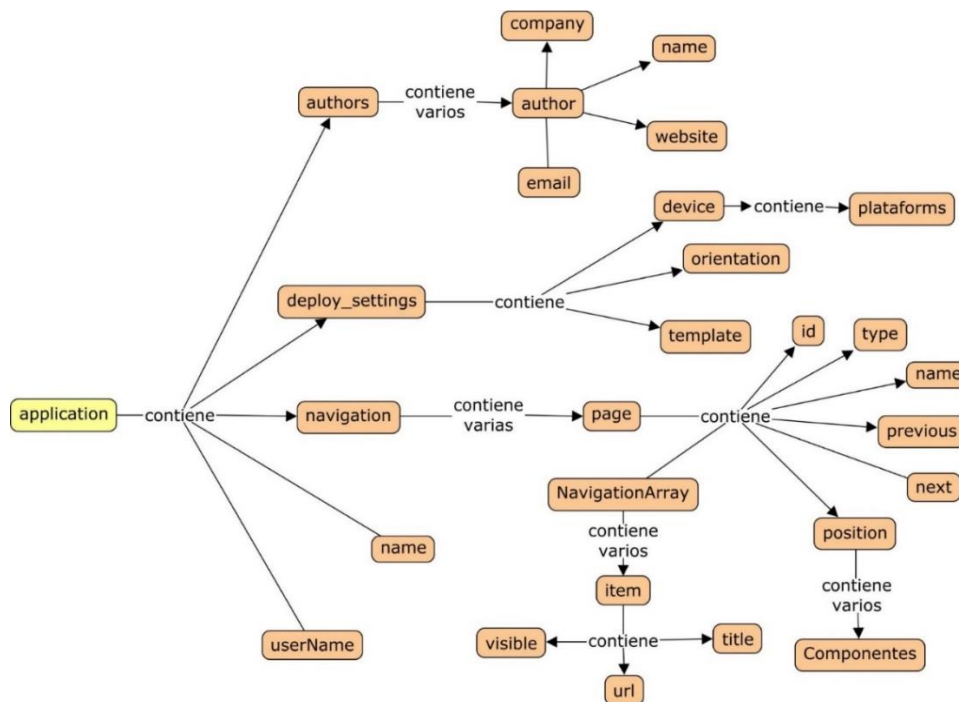


Figura 3.6 Estructura del archivo XML

3.6.3. Estructura de carpetas de los proyectos generados

Prosiguiendo con la información del generador de aplicaciones, en la Figura 3.7 se presenta la estructura de las carpetas que produce el generador de aplicaciones. Principalmente hay una carpeta principal, la cual tiene el nombre del proyecto, en esta carpeta se encuentran los proyectos divididos por el dispositivo que utiliza, es decir *Desktop*, *Mobile* y *TV*. Posteriormente

Capítulo 3. Aplicación de la metodología

cada carpeta de dispositivo contiene las plataformas que soporta: *Desktop* solamente tiene disponible Web; *Mobile* soporta los proyectos de Android™, Firefox® OS, MacOS®, Web y Windows Phone®; *TV* tiene soporte para los proyectos de Android™, Firefox® OS, Web y WebOS.

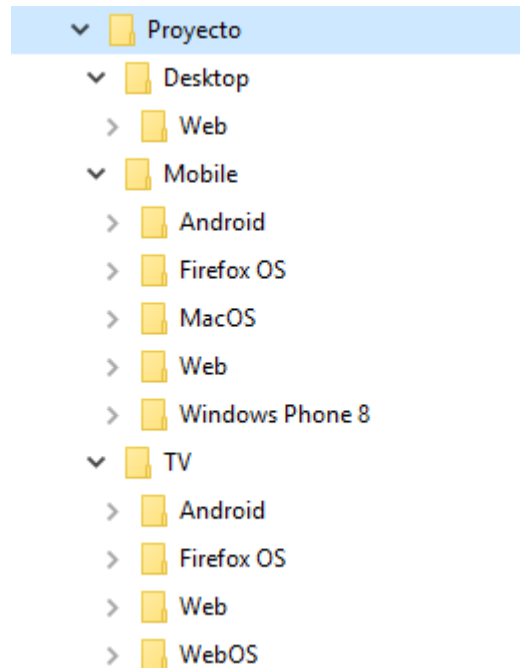


Figura 3.7 Estructura de las carpetas

Independiente de los dispositivos 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 se tienen contemplados son 3: 1) para las aplicaciones de Android™ se utilizó Android Studio; 2) para las aplicaciones en Windows Phone® se ocupó Visual Studio y; 3) para las aplicaciones en MacOS® se utilizó 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.

Las aplicaciones desarrolladas en Android™ se encuentran en las carpetas de *Mobile* y *TV*, porque EduGene permite crear aplicaciones Android™ para dispositivos móviles (celulares inteligentes y tabletas PC) y para la televisión inteligente. En la Figura 3.8 se presenta la estructura de carpetas que utilizan los proyectos en Android™ para Android™ Studio 2.3.

Capítulo 3. Aplicación de la metodología

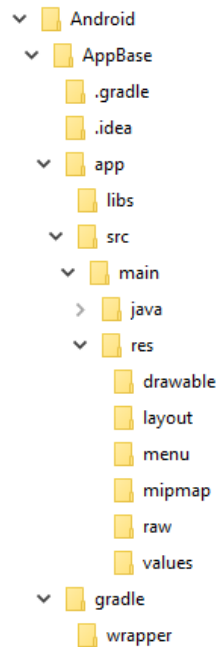


Figura 3.8 Estructura de carpetas para los proyectos de las aplicaciones de Android™

Las aplicaciones desarrolladas en MacOS® se encuentra en la carpeta de *Mobile*, en la Figura 3.9 se presenta la estructura de carpetas que utilizan los proyectos en MacOS® para Xcode 9.

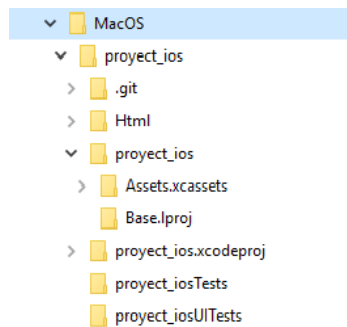


Figura 3.9 Estructura de carpetas para los proyectos de las aplicaciones de macOS®

Las aplicaciones desarrolladas en Windows Phone® se encuentra en la carpeta de *Mobile*, en la Figura 3.10 se presenta la estructura de carpetas que utilizan los proyectos en Windows Phone® para ejecutarse en Visual Studio 2016.

Capítulo 3. Aplicación de la metodología

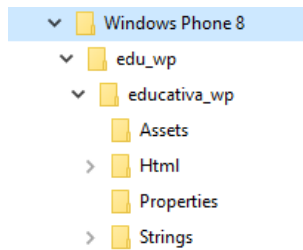


Figura 3.10 Estructura de carpetas para los proyectos de las aplicaciones de Windows Phone®

3.6.4. Componentes

El generador de aplicaciones se compone principalmente de cuatro funcionalidades, las cuales ofrecen la interacción con el servicio Web a través de un archivo XML y utilizando una caja de texto para ingresar el código XML, además se añaden nuevos componentes a través de un archivo ZIP y finalmente tiene la opción de administrar los componentes con ayuda del uso de una interfaz gráfica de usuario. A continuación, se describe cada una de las funcionalidades para aclarar mejor su funcionamiento.

En la Figura 3.11 se presenta la primera funcionalidad del componente desarrollado, que por medio de un archivo basado en XML describe las características de la aplicación, las plataformas y dispositivos en los que construir la aplicación y los patrones de diseño que se implantarán en la aplicación educativa, posteriormente el archivo se procesa para validar su estructura y al final generar un enlace de descarga que contiene el o los proyectos generados.

Una interfaz de usuario web para la creación de proyectos. En la parte superior hay una barra de navegación con los botones 'Read file', 'Read XML text', 'Add component' y 'Administrator'. El título principal es 'Select your XML file'. Debajo del título hay una etiqueta que indica 'Lowercase letters and no spaces'. Hay un campo de entrada de texto etiquetado como 'Project's name'. Debajo del campo de texto hay un botón 'Seleccionar archivo' y el texto 'No se eligió archivo'. En la parte inferior hay un botón 'Create project'.

Figura 3.11 Página de inicio de las funcionalidades

Capítulo 3. Aplicación de la metodología

En la Figura 3.12 se observa la segunda funcionalidad del generador de aplicaciones, a diferencia de la primera funcionalidad se utiliza una caja de texto para escribir el código correspondiente a una estructura XML para la generación de las aplicaciones educativas.

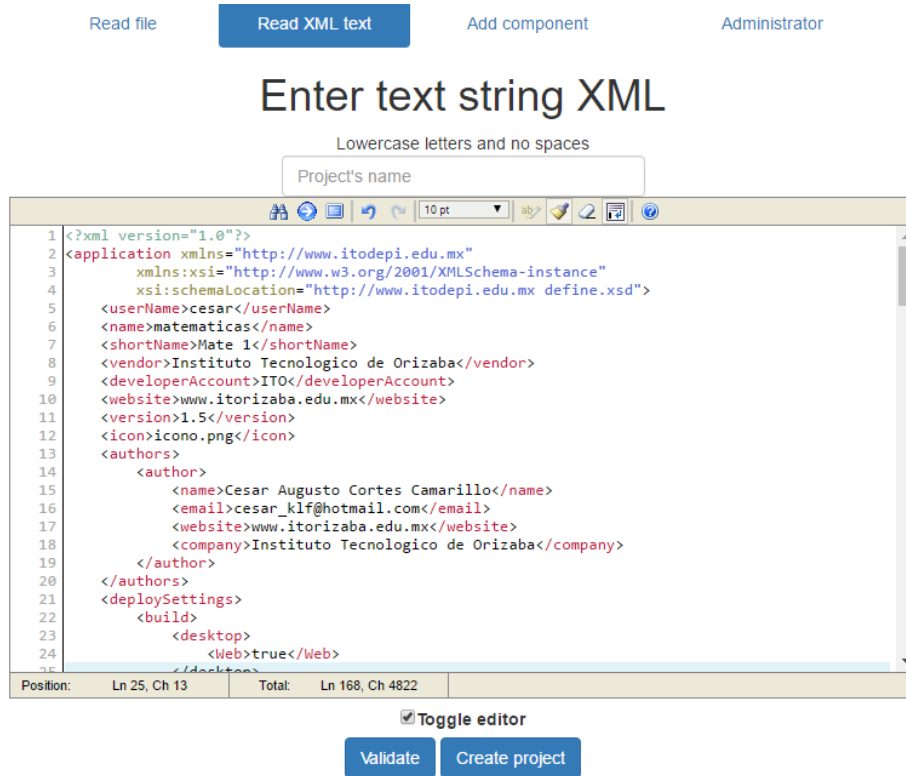


Figura 3.12 Leer texto basado en XML

Posteriormente en la Figura 3.13 se encuentra la tercera funcionalidad en la cual se ingresa un archivo ZIP que contiene un conjunto de carpetas debidamente ordenadas con la estructura de un nuevo componente y de esta forma agregarlo a la plataforma.

Capítulo 3. Aplicación de la metodología

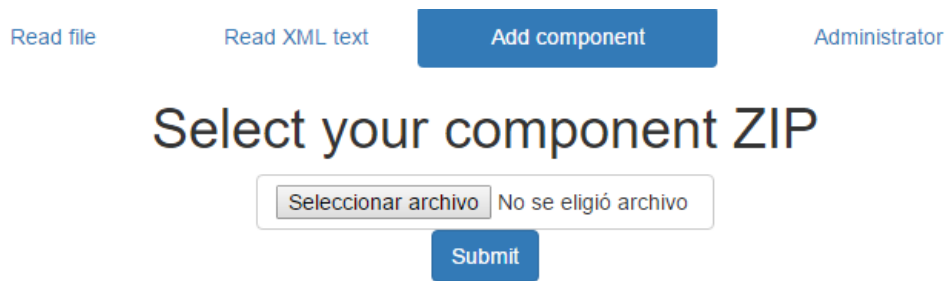


Figura 3.13 Añadir componente de un archivo ZIP

Finalmente en la Figura 3.14 se presenta la última funcionalidad donde se permite administrar los componentes que representan a un patrón de diseño de interfaz de usuario. Cada componente se integra de dos secciones: la primera es la que permite modificar sus configuraciones y la segunda sus propiedades.

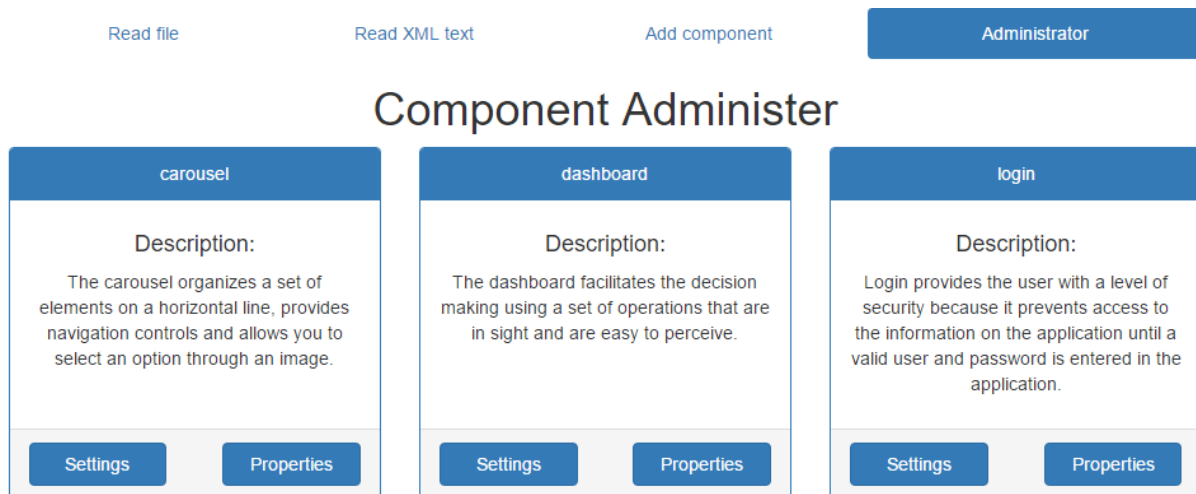
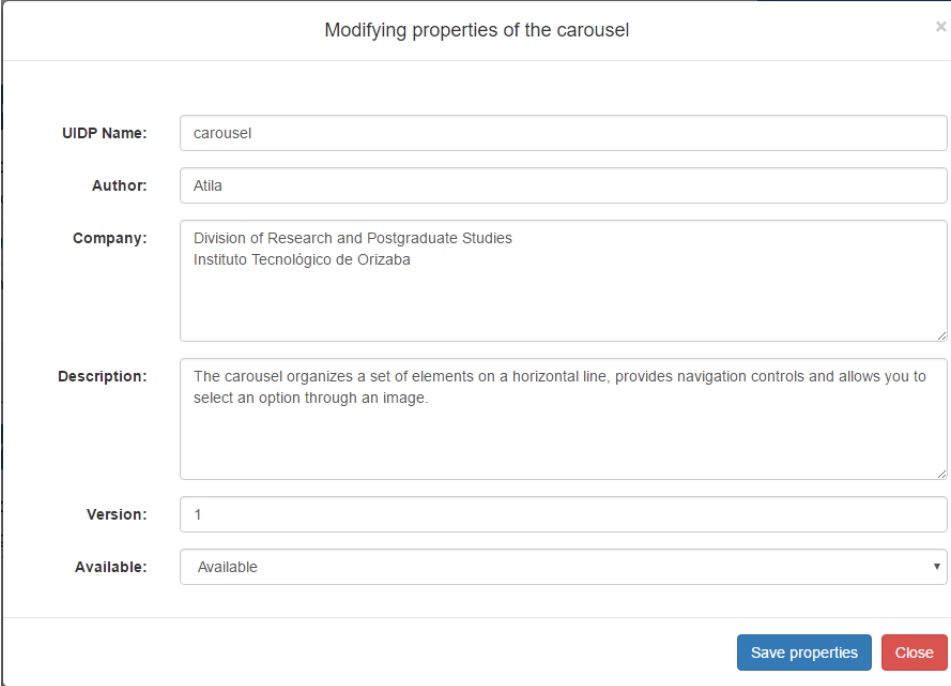


Figura 3.14 Administrar componentes

En la sección de propiedades, en donde se ingresan las características de cada componente, entre las cuales destacan el nombre, el autor, la compañía, una descripción, la versión y el estado de disponibilidad del patrón de diseño de interfaz de usuario, así como se presenta en la Figura 3.15.

Capítulo 3. Aplicación de la metodología



Modifying properties of the carousel

UIDP Name: carousel

Author: Atila

Company: Division of Research and Postgraduate Studies
Instituto Tecnológico de Orizaba

Description: The carousel organizes a set of elements on a horizontal line, provides navigation controls and allows you to select an option through an image.

Version: 1

Available: Available

Save properties Close

Figura 3.15 Propiedades de un patrón de diseño de IU

La sección de configuraciones se refiere a los dispositivos y las plataformas que cuenta cada patrón de diseño, esto es con la finalidad de administrar el código para cada componente gráfico. En la Figura 3.16 se observan las plataformas del patrón de diseño *carousel* para un dispositivo móvil y una PC.

Al seleccionar una plataforma se abre una nueva ventana en donde se observan cuadros de texto que permiten modificar el código, el lenguaje de marcado, las importaciones y los archivos multimedia necesarios para el correcto funcionamiento de la aplicación. En la Figura 3.17 se presenta la opción de *code*, en donde es posible ingresar el código fuente, los métodos, los campos, entre otros.

Capítulo 3. Aplicación de la metodología

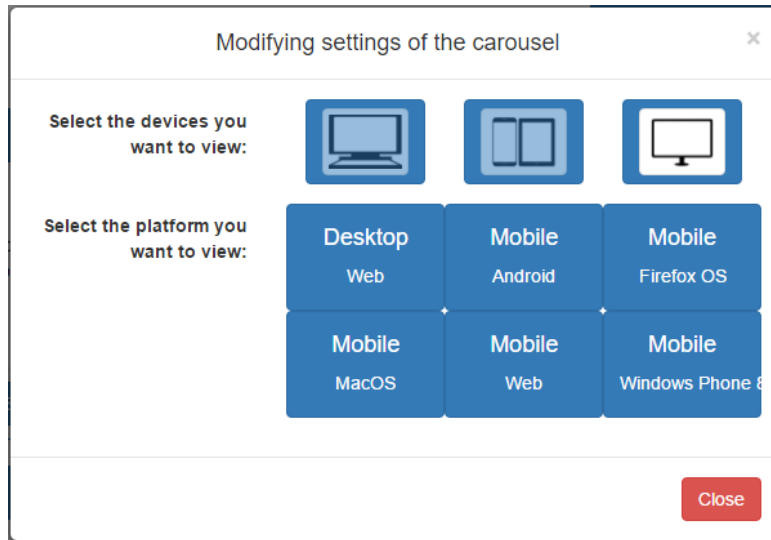


Figura 3.16 Configuraciones de un patrón de diseño de IU

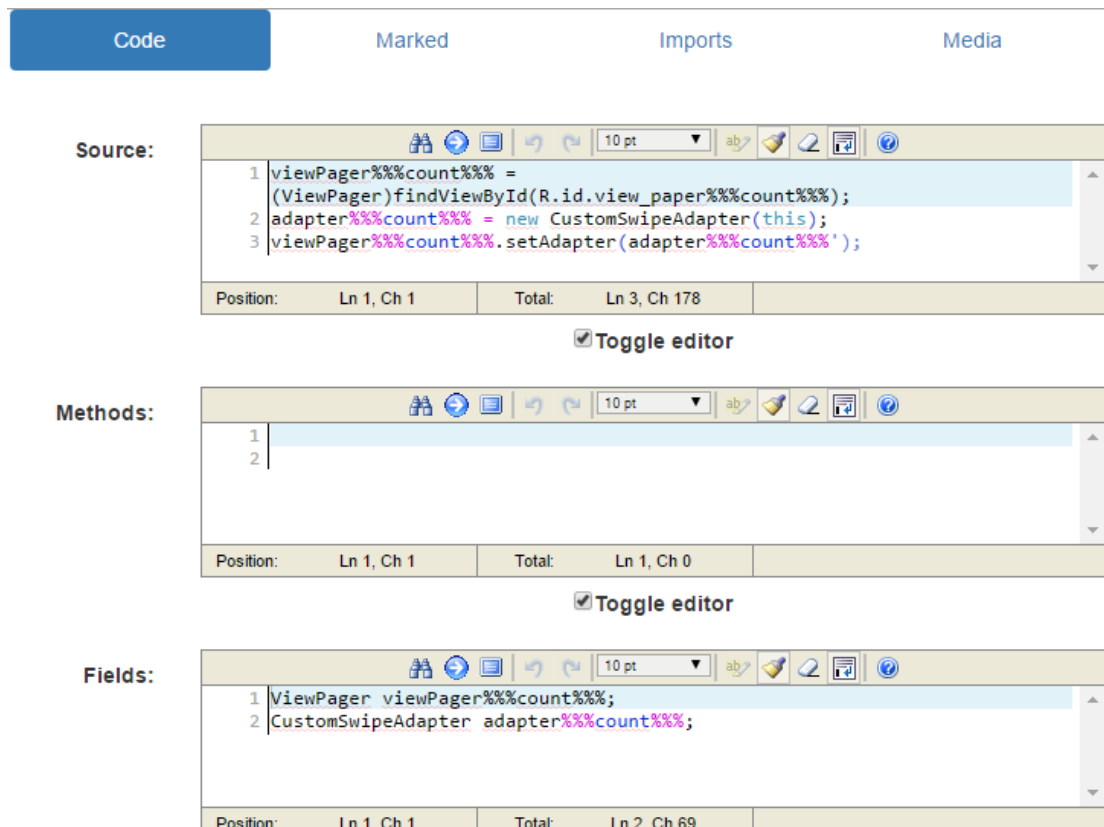


Figura 3.17 Configuraciones para cada patrón de diseño

Capítulo 4. Resultados

Capítulo 4. Resultados

En el siguiente capítulo se presentan tres casos de estudio, en donde se observa el funcionamiento del componente generador de aplicaciones educativas a través de diferentes métodos. En el primer caso de estudio se presenta la construcción de una aplicación educativa a través de un archivo XML, un proceso similar es el realizado en el segundo caso de estudio donde se presenta la construcción mediante el uso de una interfaz gráfica, finalmente el tercer caso de estudio utiliza una imagen que contiene los patrones de diseño de interfaz de usuario necesarios para construir la aplicación educativa.

4.1. Caso de estudio 1: Generación de MOOC utilizando un archivo XML

Como primer caso de uso se tiene el desarrollo de una aplicación educativa a partir de un archivo XML. Donde primero se presenta el diseño de la aplicación, el proceso para la construcción de la aplicación y por último el funcionamiento de la aplicación móvil desarrollada.

4.1.1. Diseño de la aplicación

En este caso de estudio, se propone la generación de una aplicación educativa para dispositivos móviles. Por tal razón, para este caso de estudio se seleccionaron dos plataformas que EduGene da soporte para los dispositivos móviles (MacOS® y Windows Phone®). La aplicación desarrollada sirve para enseñar español a través de un curso. EduGene requiere los siguientes patrones de diseño:

- *SplashScreen*: Muestra una imagen en el centro de la pantalla mientras la aplicación está cargando.
- *Dashboard*: Genera un conjunto de elementos para acceder a diferentes contenidos de aplicación.
- Inicio de sesión: Impide el acceso hasta que se introduzca el nombre de usuario y la contraseña correspondientes.
- Vídeo: Muestra un vídeo en la aplicación.
- Listas: Muestra una serie de imágenes en la aplicación.

Capítulo 4. Resultados

4.1.2. Construyendo la aplicación

Para iniciar el proceso de construcción de la aplicación, fue necesario activar y desactivar las características requeridas para que la aplicación sea desarrollada. El generador de aplicaciones utiliza una etiqueta *deploySettings* para determinar las siguientes características: dispositivos, orientación, plataformas y estructura de la aplicación educativa. Para este ejemplo, es necesario activar las plataformas macOS® y Windows Phone®. A continuación, las orientaciones de *portrait* y *landscape*, y finalmente *smartphones* y tabletas se activan. En el Listado 4.1, se presenta un fragmento del archivo basado en XML que describe la etiqueta *deploySettings*.

Listado 4.1 Caso de estudio 1: Elementos principales de la etiqueta *deploySettings* en un archivo XML

```
1 <deploySettings>
2     <device>
3         <desktop>false</desktop>
4         <tablet>true</tablet>
5         <smartphone>true</smartphone>
6         <smarttv>false</smarttv>
7     </device>
8     <orientation>
9         <portrait>true</portrait>
10        <landscape>true</landscape>
11    </orientation>
12    <platforms>
13        <android> false </android>
14        <macOS>true</macOS>
15        <firefox>false</firefox>
16        <windowsPhone> true </windowsPhone>
17        <web>false</web>
18    </platforms>
19    <template>
20        <head>true</head>
21        <body>true</body>
22        <foot>true</foot>
23        <left>true</left>
24        <right>true</right>
25    </template>
26 </deploySettings>
```

Capítulo 4. Resultados

Posteriormente se ingresó el código para generar la aplicación. En esta sección es donde cada capa de página es registrada, incluyendo los componentes involucrados y el tipo de navegación, entre otros. Para lograr esto, una estructura de cinco posiciones (*head*, *body*, *foot*, *left* y *right*) localiza los componentes y la etiqueta *NavigationArray* indica el modo de navegación en la aplicación a través de cada elemento, como se muestra en el Listado 4.2. Para esta aplicación, se utilizaron cinco patrones de diseño de IU divididos en cinco capas, para aprovechar los beneficios del uso de patrones de diseño en una aplicación educativa. Además, de obtener una aplicación que funcione correctamente en plataformas macOS® y Windows Phone®.

Listado 4.2 Caso de estudio 1: Ejemplo de una capa

```
1 <page>
2   <id>1</id>
3   <name>Splashscreen</name>
4   <title>Prueba</title>
5   <type>splashscreen</type>
6   <modal>true</modal>
7   <next>2</next>
8   <previous>1</previous>
9   <position>
10      <head></head>
11      <body> @acsplashscreenac@ </body>
12      <left></left>
13      <right></right>
14      <foot></foot>
15   </position>
16   <navigationArray>
17     <item>
18       <id>1</id>
19       <title></title>
20       <url>1.html</url>
21       <type>blank</type>
22       <target></target>
23       <img></img>
24       <backgroundcolor> red </backgroundcolor>
25       <foreground></foreground>
26       <visible>true</visible>
27       <disabled>>false</disabled>
28       <alt></alt>
```


Capítulo 4. Resultados

```
29 | <value></value>
30 | </item>
31 | </navigationArray>
32 | </page>
```

A continuación, se proporciona el archivo basado en XML dentro de la aplicación para generar los proyectos de las aplicaciones seleccionadas. Este proceso se representa en la Figura 4.1, donde el usuario selecciona el archivo basado en XML y obtiene un hipervínculo para descargar los proyectos empaquetados.

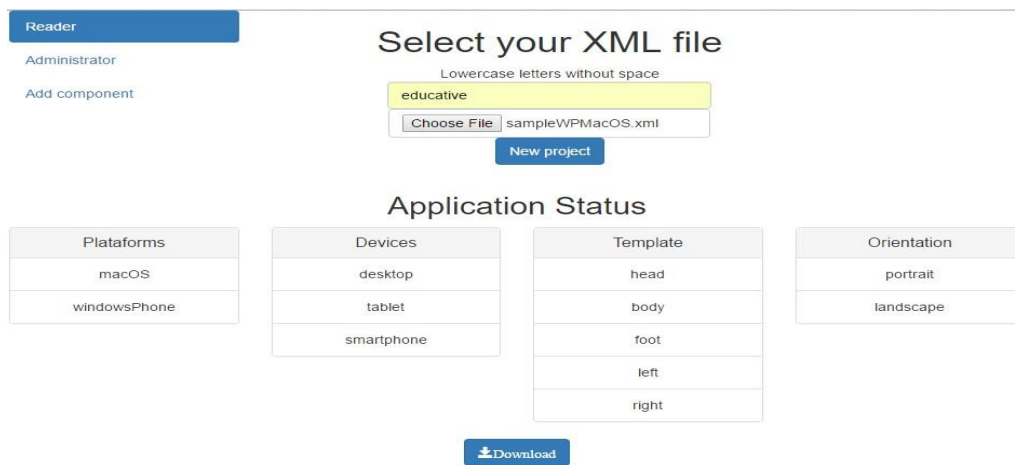


Figura 4.1 La interfaz gráfica del generador de aplicaciones para una aplicación en macOS® y Windows Phone®

4.1.3. Probando la aplicación

Continuando con el caso de estudio, en la Figura 4.2 se muestra la aplicación educativa que se ejecuta bajo un simulador de Windows Phone® 8.1 utilizando Visual Studio. Por otro lado, la Figura 4.3 muestra la misma aplicación que se ejecuta en la plataforma MacOS® en un simulador de iPhone 6 usando Xcode. Como se observa la aplicación tiene cinco capas descritas a continuación:

- *SplashScreen*: Esta es la primera capa y contiene el patrón de diseño SplashScreen.
- Inicio de sesión: Esta segunda capa utiliza el patrón de diseño de inicio de sesión.
- *Dashboard*: Esta capa utiliza dos patrones de diseño de IU: el patrón de diseño *Dashboard* en la posición *body* y un patrón de diseño *Menú* en la posición *head*.

Capítulo 4. Resultados

- Contenido: Esta capa utiliza un patrón de diseño de lista para mostrar un conjunto de elementos (lista) que dirigen a elementos multimedia.
- Vídeo: Esta capa utiliza el patrón de diseño de vídeo.

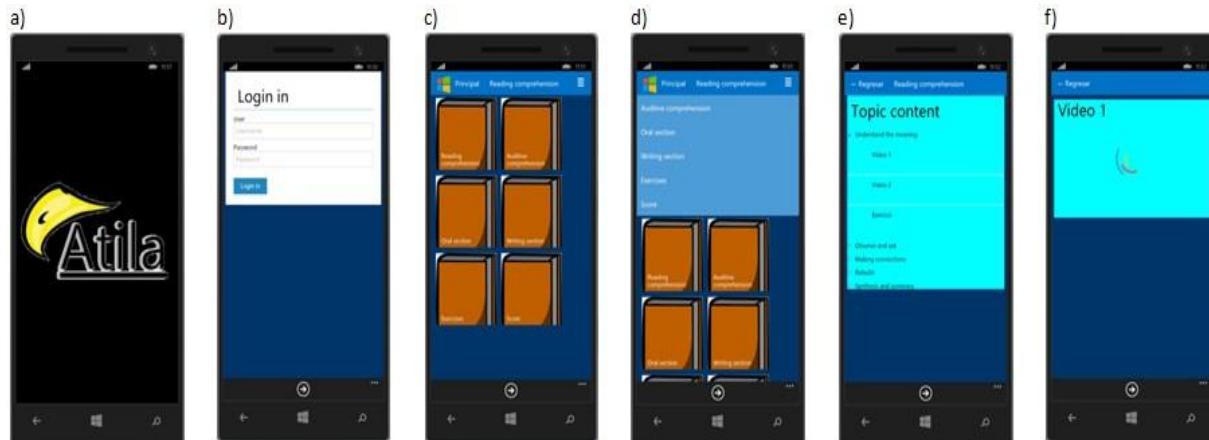


Figura 4.2 Caso de estudio 1: Ejecución de una aplicación educativa con Visual Studio: a) SplashScreen; b) Inicio de sesión; c) Dashboard; d) Menú; e) Lista; f) Video

Para generar la aplicación, solo es necesario indicar el nombre del patrón de diseño y su posición en la aplicación. EduGene lee el archivo XML y genera un proyecto compatible con la plataforma a utilizar. Además, EduGene permite al usuario modificar libremente la interfaz de usuario de la aplicación.

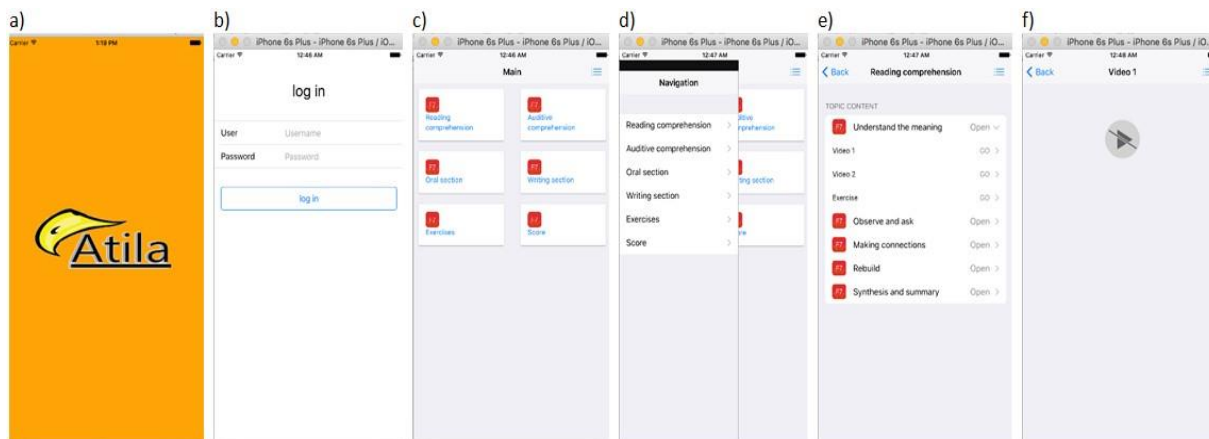


Figura 4.3 Caso de estudio 1: Ejecución de la aplicación educativa mediante Xcode: a) SplashScreen; b) Inicio de sesión; c) Dashboard; d) Menú; e) Lista; f) Video

Capítulo 4. Resultados

Entre los beneficios que ofrece EduGene está la facilidad para desarrollar aplicaciones educativas para dispositivos móviles a través del uso de patrones de diseño de interfaz de usuario, así también, la generación de código fuente de los componentes de un archivo basado en XML.

4.2. Caso de estudio 2: Generación de un MOOC a través de una interfaz gráfica de usuario

A continuación, se presenta el segundo caso de estudio en el cual se realizó una aplicación educativa utilizando una interfaz gráfica de usuario, esto permitió demostrar que el archivo XML se genera exitosamente a través de un intermediario, además de que utiliza el servicio REST que se diseñó para utilizarse por EduGene.

4.2.1. Diseño de la aplicación

Para ejemplificar las capacidades de EduGene como un generador de aplicaciones educativas, se determinó la generación de un MOOC para MacOS® y Windows Phone® para impartir un curso de Física. La aplicación educativa a desarrollar con EduGene cumple con las siguientes características:

- Visualizar una imagen que muestre el logotipo de la aplicación antes de ejecutar la aplicación.
- Proporcionar un sistema de autenticación que otorgue acceso sólo a los estudiantes inscritos.
- Mostrar cursos relacionados con la Física.
- Muestre de manera ordenada cada programa del curso.
- Avisar al usuario en caso de problemas en la aplicación.
- Reproducir videoclips.

Para seleccionar el tipo de aplicación a desarrollar (por ejemplo, blog, *wiki* o MOCC), se consideraron las características de la aplicación de Física a desarrollar y el conjunto de patrones de diseño de interfaz de usuario que ofrece EduGene. Con base en el análisis realizado en la Tabla 3.1 se llegó a la conclusión de que MOOC es la aplicación educativa que se necesita y

Capítulo 4. Resultados

además los patrones de diseño necesarios para cumplir con las características de la aplicación de física, son los que se mencionan a continuación:

- *Splashscreen*: Permite al usuario presentar el logotipo de la aplicación.
- *Login*: Proporciona seguridad de inicio de sesión al permitir el acceso a la aplicación sólo a través de un nombre de usuario registrado y una contraseña.
- *Dashboard*: Muestra los cursos disponibles.
- Lista: Muestra el programa de cada curso en forma de una lista.
- Diálogo: envía mensajes de advertencia al usuario.
- Video: Reproduce videoclips.

4.2.2. Construyendo la aplicación

El primer paso para la construcción del MOOC es ingresar a la plataforma, la cual se muestra en la Figura 4.4. La Sección A muestra un acordeón que despliega las aplicaciones educativas registradas, la opción de crear un proyecto de aplicación educativa y la opción de construir una aplicación educativa según las plataformas disponibles en EduGene. La sección B ilustra el panel de control de EduGene que permite a los usuarios visualizar proyectos de aplicaciones ya creadas.

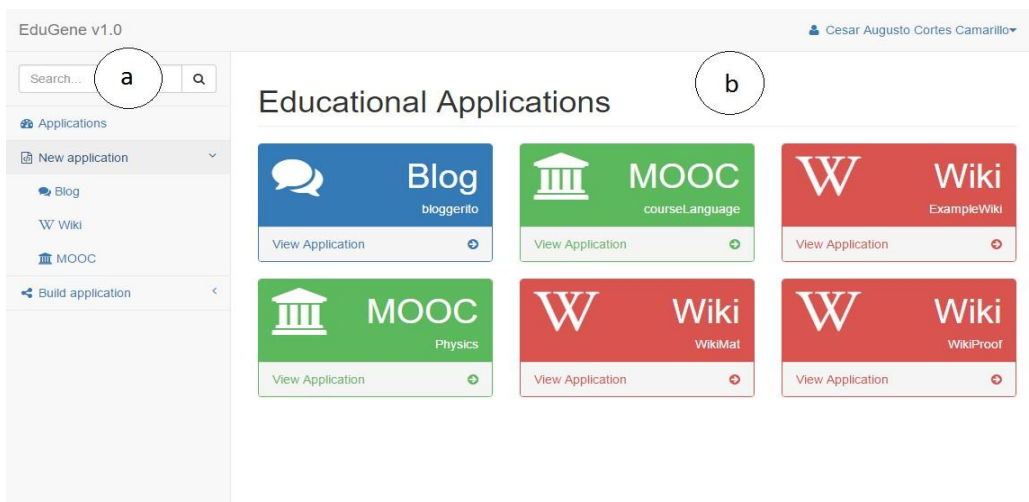


Figura 4.4 Plataforma EduGene

Capítulo 4. Resultados

En el segundo paso para construir la aplicación es necesario seleccionar que se va a construir un MOOC. En esta etapa, es necesario proporcionar un nombre a la aplicación y seleccionar al menos tres UIDP para integrarlos. Este procedimiento se ilustra en la Figura 4.5.

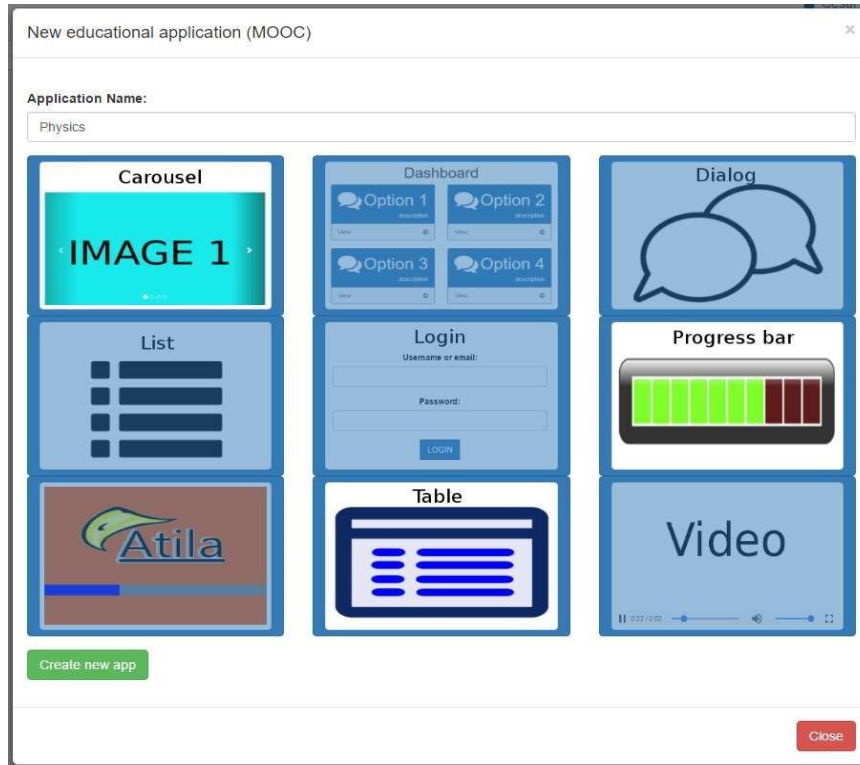
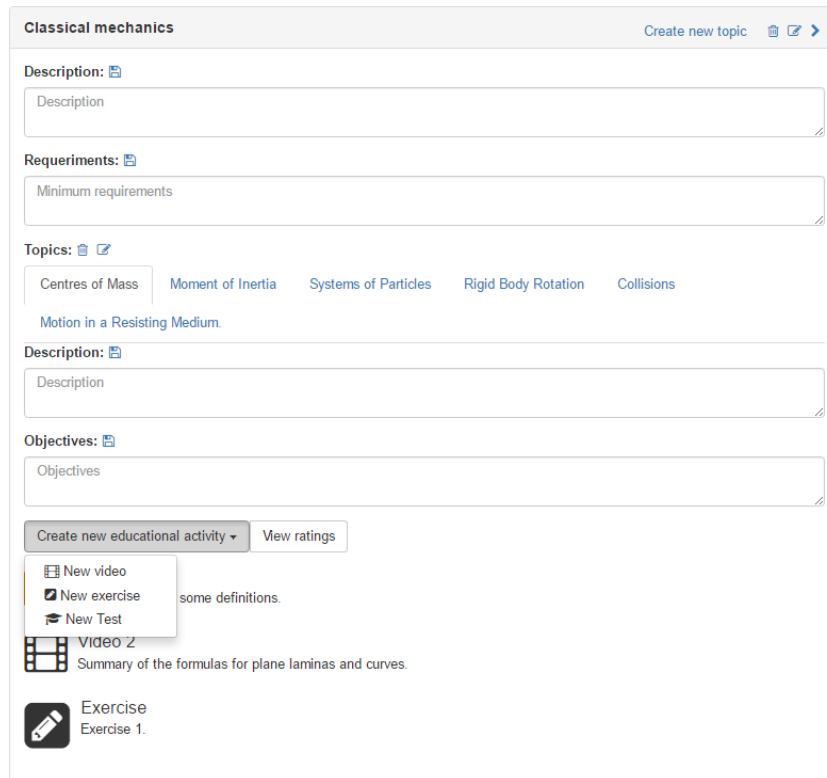


Figura 4.5 Desarrollo de aplicaciones educativas con EduGene

El tercer paso consiste en seleccionar la plataforma o plataformas en las que se ejecutará la aplicación educativa. Actualmente, las plataformas disponibles en EduGene incluyen: Android™, Firefox® OS, MacOS®, Windows Phone® y la Web. Al seleccionar la opción para crear la aplicación, EduGene muestra un menú que enumera los diferentes dispositivos y plataformas en los que se desarrolla la aplicación educativa.

El cuarto paso es agregar contenido a la aplicación educativa. En este caso, dependiendo de la aplicación educativa seleccionada, EduGene despliega diferentes opciones para incorporar contenido. La Figura 4.6 muestra cómo crear o eliminar listas de temas, cursos y temas. Entonces, EduGene construye una aplicación educativa basada en este contenido.

Capítulo 4. Resultados



Classical mechanics Create new topic

Description:

Requeriments:

Topics:

[Motion in a Resisting Medium.](#)

Description:

Objectives:

Create new educational activity View ratings

- New video
- New exercise some definitions.
- New Test

VIDEO 2
Summary of the formulas for plane laminas and curves.

Exercise
Exercise 1.

Figura 4.6 Edición de contenido de aplicaciones educativas

El quinto y último paso para construir la aplicación es esperar hasta que se complete la generación del código fuente para cada plataforma.

Una vez completada la generación del código fuente, EduGene muestra un hipervínculo para descargar los proyectos generados. En este caso, EduGene genera un hipervínculo para que el usuario descargue las aplicaciones en iOS y Windows Phone®.

Listado 4.3 Caso de estudio 2: Información relevante de la aplicación en el archivo XML

```
1 <userName>cesar</userName>
2 <name>Mathematics</name>
3 <shortName>Maths</shortName>
4 <vendor>Instituto Tecnologico de Orizaba</vendor>
5 <developerAccount>ITO</developerAccount>
6 <website>www.itorizaba.edu.mx</website>
7 <version>1.5</version>
8 <icon>icon.png</icon>
9 <authors>
```

Capítulo 4. Resultados

```
10      <author>
11          <name>Cesar Augusto Cortes Camarillo</name>
12          <email>cesar_klf@hotmail.com</email>
13          <website>www.itorizaba.edu.mx</website>
14          <company>Instituto Tecnologico de Orizaba</company>
15      </author>
16 </authors>
```

Cada código fuente que se genera por EduGene se origina a partir de un archivo basado en XML, el generador de aplicaciones tiene dos funcionalidades importantes: la primera es generar el archivo basado en XML necesario para construir la aplicación y utilizar ese archivo para construir la aplicación educativa.

Listado 4.4 Caso de estudio 2: Elementos principales de la etiqueta deploySettings en un archivo XML

```
1 <deploySettings>
2     <build>
3         <desktop>
4             <Web>false</Web>
5         </desktop>
6         <mobile>
7             <Android>false</Android>
8             <Firefox_OS>false</Firefox_OS>
9             <MacOS>true</MacOS>
10            <Web>false</Web>
11            <Windows_Phone_8>true</Windows_Phone_8>
12        </mobile>
13        <tv>
14            <Android>false</Android>
15            <Firefox_OS>false</Firefox_OS>
16            <Web>false</Web>
17            <WebOS>false</WebOS>
18        </tv>
19    </build>
20    <orientation>
21        <portrait>true</portrait>
22        <landscape>true</landscape>
23    </orientation>
24    <template>
```

Capítulo 4. Resultados

```
25 <head>true</head>
26 <body>true</body>
27 <foot>false</foot>
28 <left>true</left>
29 <right>false</right>
30 </template>
31 </deploySettings>
```

El archivo basado en XML consta de tres partes principales. La primera parte se observa en el Listado 4.3 y contiene información relevante sobre la aplicación, como nombres de los autores, nombre de la aplicación y logotipo, entre otros. La segunda parte del archivo basado en XML, que se muestra en el Listado 4.4, se refiere a la configuración del proyecto, es decir, orientaciones, dispositivos seleccionados, plataformas seleccionadas y plantilla utilizada. Finalmente, la tercera parte se ilustra en el Listado 4.5. Describe el contenido de la aplicación, es decir, las páginas o diseños y la navegación realizada.

Listado 4.5 Caso de estudio 2: Contenido y navegación de la aplicación en el archivo XML

```
1 <navigation>
2   <page>
3     <id>1</id>
4     <name>Splashscreen</name>
5     <title>Proof</title>
6     <type>splashscreen</type>
7     <next>2</next>
8     <position>
9       <head></head>
10      <body>
11        <uidp type="single" name="splashscreen"></uidp>
12      </body>
13      <left></left>
14      <right></right>
15      <foot></foot>
16    </position>
17  </page>
18 </navigation>
```

El archivo XML se genera dependiendo del tipo de aplicación educativa y de los patrones de diseño necesarios. En otras palabras, EduGene establece el orden y la relevancia en la cual los

Capítulo 4. Resultados

patrones de diseño aparecerán en la aplicación educativa; se incorpora el nombre de la aplicación, las características y la navegación que necesita el archivo XML para construir una aplicación educativa.

4.2.3. Probando la aplicación

El código fuente generado por EduGene es equivalente a una plantilla que se modifica para satisfacer los requisitos de la aplicación. Por ejemplo, EduGene desarrolla una plantilla que incluye el patrón *Login*, pero el desarrollador es responsable de conectar ese patrón a una base de datos o servicio Web. Gracias a EduGene, el desarrollador ahorra tiempo en la tarea de diseñar la interfaz gráfica de usuario y por lo tanto reduce el tiempo de desarrollo de un producto de software.



Figura 4.7 Caso de estudio 2: Ejecución de una aplicación educativa con Visual Studio: a) SplashScreen; B) Inicio de sesión; C) Dashboard; D) Menú; E) Lista; F) Video

Capítulo 4. Resultados

Para terminar esta sección, la Figura 4.7 y la Figura 4.8 muestran cómo EduGene ejecuta los códigos fuente desarrollados para plataformas MacOS® y Windows Phone®, respectivamente. Es posible observar que el código fuente generado es una plantilla que se ajusta a las necesidades de la aplicación que se pretende desarrollar al final.

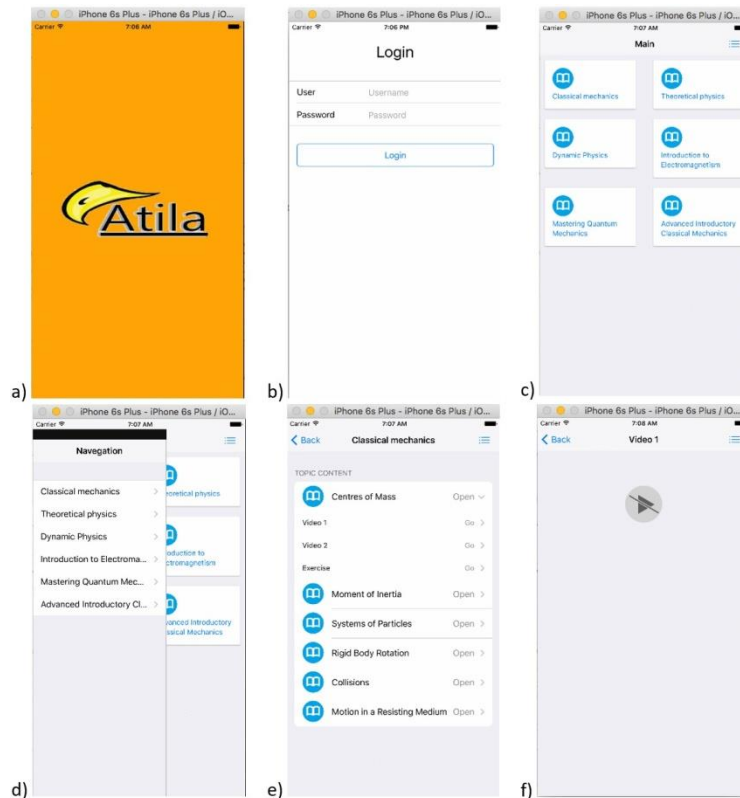


Figura 4.8 Caso de estudio 2: Ejecución de la aplicación educativa mediante Xcode: a) SplashScreen; B) Inicio de sesión; C) Dashboard; D) Menú; E) Lista; F) Video

4.3. Caso de estudio 3: Creación de un MOOC por medio de una imagen

A continuación, se presenta el tercer caso de estudio en el cual se realizó una aplicación educativa a partir de una imagen que representa a los patrones de diseño de interfaz de usuario que se utilizarán, esto permitió demostrar que no solamente a través de una interfaz gráfica se logra desarrollar una aplicación educativa como se visualizó en el segundo caso de estudio, sino que también a través de una imagen es posible generar exitosamente el archivo XML, además de seguir utilizando apropiadamente el servicio REST que se diseñó para utilizarse por EduGene.

Capítulo 4. Resultados

4.3.1. Diseño de la aplicación

Para ejemplificar las capacidades de EduGene como un generador de aplicaciones educativas, se describió la generación de un MOOC para Android™ para generar la estructura de un curso a partir de los patrones de diseño de interfaz de usuario que contiene una imagen delimitada. La aplicación educativa a desarrollar con EduGene cumple con las siguientes características:

- Visualizar una imagen que muestre el logotipo de la aplicación antes de ejecutar la aplicación.
- Proporcionar un sistema de autenticación que otorgue acceso sólo a los estudiantes inscritos.
- Mostrar los cursos que estarán disponibles.
- Visualizar los diferentes temas que contiene un curso.
- Mostrar una lista de los temas que impartirá cada curso.
- Reproducir videoclips.

De acuerdo a las características de impartir cursos se llegó a la conclusión de construir una aplicación educativa de tipo MOOC y con la finalidad de cumplir con las necesidades solicitadas, se utilizarán los siguientes patrones de diseño de interfaz de usuario:

- *Splashscreen*: Permite al usuario presentar el logotipo de la aplicación.
- *Login*: Proporciona seguridad de inicio de sesión al permitir el acceso a la aplicación sólo a través de un nombre de usuario registrado y una contraseña.
- *Dashboard*: Muestra los cursos disponibles.
- *Carrusel*: Muestra los temas más relevantes del curso seleccionado.
- *Lista*: Muestra el programa de cada curso en forma de una lista.
- *Video*: Reproduce videoclips.

4.3.2. Construyendo la aplicación

El primer paso para la construcción del MOOC es ingresar a la plataforma, la cual se presentó en la Figura 4.4. Posteriormente es necesario seleccionar los patrones de diseño que se utilizarán, estos patrones se encuentran representados en la Tabla 3.8 y en la Tabla 3.9 con bases en esto

Capítulo 4. Resultados

se dibujaron los patrones de diseño necesarios para realizar la aplicación educativa, así como se muestra en la Figura 4.9 donde se dibujaron los iconos del *splashscreen*, el *login*, el *dashboard*, el carrusel, la lista y el video.

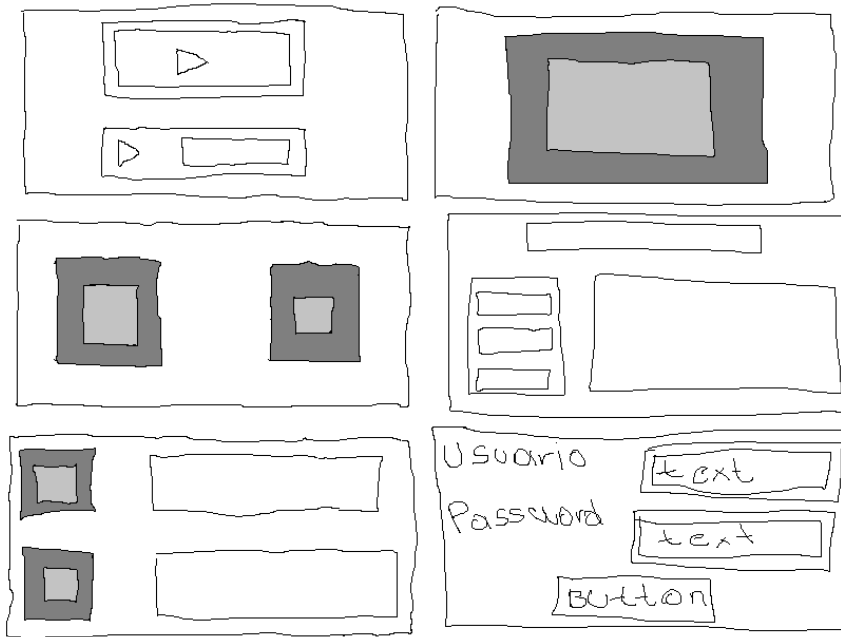


Figura 4.9 Caso de estudio 3: Bosquejos de los patrones de diseño de interfaz de usuario

Posteriormente se ingresó a la plataforma para ingresar la imagen como se muestra en la Figura 4.10 y enviarla al componente identificador de patrones, quien devuelve un JSON con los patrones que se reconocieron en la imagen.

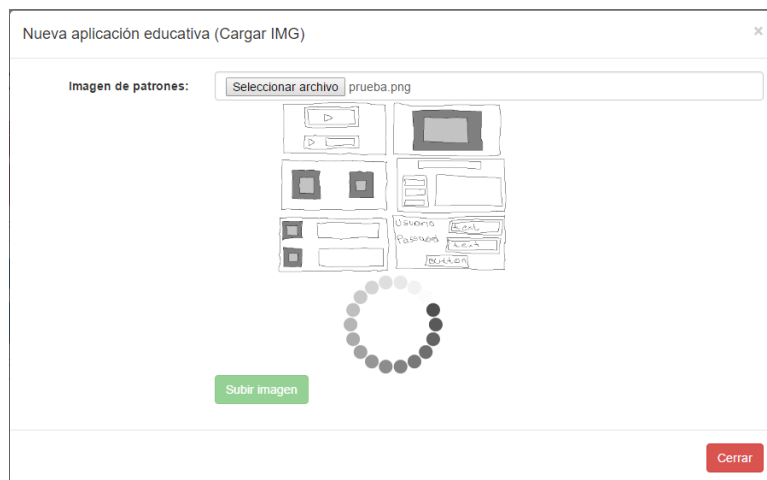


Figura 4.10 Caso de estudio 3: Carga de imagen de los patrones de diseño de interfaz de usuario

Capítulo 4. Resultados

Después de reconocer los patrones de diseño, EduGene muestra los patrones identificados, las aplicaciones educativas que permite desarrollar y se permite ingresar el nombre de la aplicación, tal como se muestra en la Figura 4.11.

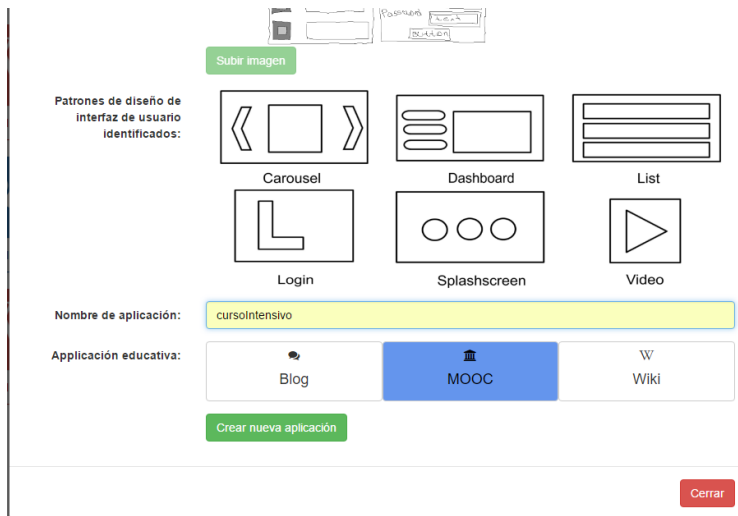


Figura 4.11 Caso de estudio 3: Patrones de diseño de interfaz de usuario identificados

Para finalizar la construcción de la aplicación, se selecciona construir proyecto como en el caso de estudio anterior.

4.3.3. Probando la aplicación

Para terminar esta sección, en la Figura 4.12 se muestra que el proyecto generado por EduGene se compiló y es ejecutado en un dispositivo Android™. Es posible observar que el código fuente generado es una plantilla que se ajusta a las necesidades de la aplicación que se pretende desarrollar al final.

Capítulo 4. Resultados

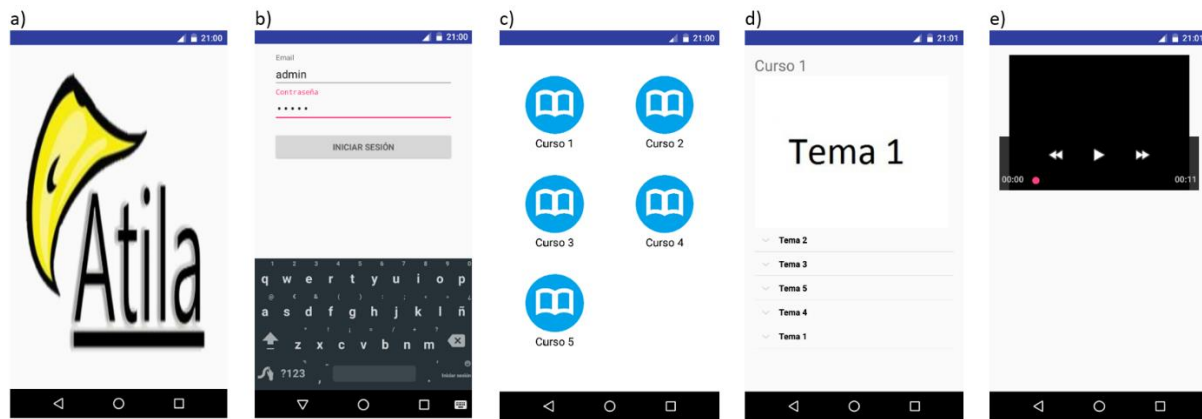


Figura 4.12 Caso de estudio 3: Ejecución de una aplicación educativa en un dispositivo Android™: a) SplashScreen; B) Inicio de sesión; C) Dashboard; D) Carrusel y Lista; E) Patrón de video

4.4. Evaluación

En todo proyecto de desarrollo de software es necesario evaluar el desempeño, la velocidad, la eficiencia y la calidad de producto de software que se ha creado. Por esta razón, este proyecto de tesis no es la excepción, de tal manera que se realizaron dos evaluaciones: una cualitativa para determinar los aspectos que no son posibles determinar con un número en concreto y una cuantitativa que a través del uso de la herramienta y/o componente devuelve un valor numérico.

De acuerdo con Twining et al. [59], existe un desequilibrio entre los trabajos cuantitativos y cualitativos publicados en revistas de investigación altamente calificadas. Tal afirmación es apoyada por numerosos hallazgos, entre los cuales los propuestos por Chiu, Lin y Chou [60] revelaron que durante los últimos 31 años, el 37,6% de los estudios utilizan un enfoque cualitativo, mientras que el 17,2% utilizaron un enfoque cuantitativo. Además, el 19,8% de los estudios adoptaron un enfoque mixto (es decir, cualitativo y cuantitativo), otro 19,8% cuantifica datos cualitativos en revistas internacionales y el 17,5% de investigaciones realizaron estudios no empíricos para métodos de codificación. Asimismo, los autores encontraron una tasa creciente de publicaciones enfatizando el análisis de datos cualitativos, desde el 0,3% en el primer período (1982-1987) al 11,5% en el último período (2008-2012). Por esta razón, la evaluación de software relaciona apropiadamente cada característica de software con su método de evaluación correspondiente (es decir, cuantitativo o cualitativo).

Capítulo 4. Resultados

Diferentes estudios propusieron sus propias versiones de aquellas características que son esenciales para las aplicaciones de *e-learning*. Por ejemplo, Joo et al. [61] 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. De manera similar, Mohammadi [62] implementó un modelo integrado de sistemas de información y el Modelo de Aceptación de Tecnología (*TAM, Technology Acceptance Model*) 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*. Asimismo, McGill et al. [63] presentaron 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. [64] proporcionaron una comprensión más profunda del impacto de las características culturales de los estudiantes, para el individualismo/colectivismo, sobre los resultados percibidos del uso de sistemas de *e-learning*. El estudio propuso un modelo exitoso de sistemas de *e-learning* que incluía una construcción cultural, individualismo/colectivismo. Del mismo modo, Porter et al. [65] determinaron hasta qué punto las estrategias, la estructura y las medidas de apoyo de las instituciones facilitaron o impidieron la adopción de la metodología de aprendizaje combinado entre los profesores de educación superior. Por último, Abdullah y Ward [66] identificaron los factores externos más utilizados del *TAM* en el contexto de la adopción del *e-learning*. Los autores realizaron un meta análisis cuantitativo de 107 artículos en los últimos diez años.

Dado que EduGene se evalúa al considerar factores tanto cualitativos como cuantitativos, se realizó una evaluación mixta. Las siguientes secciones describen las etapas de los análisis cualitativos y cuantitativos.

4.4.1. Evaluación cualitativa

En la siguiente sección, se presenta el diseño de la evaluación cualitativa y los resultados obtenidos a partir de un conjunto de aspectos que permitieron determinar por medio de una

Capítulo 4. Resultados

escala de Likert un número en concreto para evaluar la herramienta generadora de código y las aplicaciones educativas que se desarrollaron.

4.4.1.1. Diseño de la evaluación

Para diseñar la evaluación, se consideraron cuatro aspectos clave de las aplicaciones educativas. Estos aspectos fueron propuestos por la literatura sobre sistemas de *e-learning* y aseguran un aprendizaje adecuado y la enseñanza a través de las aplicaciones educativas generadas.

Necesidad 1 (N1): Comentarios.

Harrati et al. [67] sostienen que el contenido de una aplicación educativa es capaz de transmitir correctamente el mensaje a los estudiantes. Por lo tanto, una aplicación educativa apropiada muestra su contenido en una forma bien estructurada y apropiadamente tiene que capturar los resultados recibidos por el sistema.

Característica 1 (C1): Permite capturar la respuesta del usuario.

Característica 2 (C2): Soportes visuales.

Necesidad 2 (N2): Claridad - Presentación.

Poulova y Simonova [68] afirman que una aplicación de *e-learning* necesita ser entendida por los usuarios. Esta característica se logra al combinar las interfaces gráficas de usuario con el contenido apropiado. Asimismo, García-Cabot et al. [69] sostienen que el contenido presentado por las aplicaciones de *e-learning*, necesitan mostrarse eficientemente.

Característica 1 (C1): Diseño ordenado.

Característica 2 (C2): Actividades que facilitan el aprendizaje.

Característica 3 (C3): Presentación eficiente.

Necesidad 3 (N3): Usabilidad de las aplicaciones educativas generadas.

De acuerdo con Alshammari et al. [70], para que una aplicación de *e-learning* sea utilizable, su interfaz gráfica necesita ser adaptable a diferentes tamaños de pantalla y tiene que ser fácil de

Capítulo 4. Resultados

aprender. Del mismo modo, Joo et al. [61] postularon que un diseño de aplicación eficiente es un sinónimo de una interfaz fácil de usar.

Característica 1 (C1): Diseño adaptativo.

Característica 2 (C2): Fácil de usar.

Característica 3 (C3): Fácil de aprender.

Necesidad 4 (N4): Estabilidad.

Con base en lo que mencionaron McGill et al. [63], una aplicación de *e-learning* estable utiliza tecnologías novedosas y no obsoletas.

Característica 1 (C1): Tecnologías innovadoras.

Característica 2 (C2): Funciona con un propósito bien definido.

Lubiano et al. [71] y Norman [72] sostienen que la escala de Likert permite efectivamente medir niveles para varios factores. De hecho, la escala de Likert se utiliza ampliamente en diferentes áreas, como la medicina [73], el cálculo [74] y la química [75]. En este estudio, se empleó una escala de cinco puntos Likert para medir los aspectos identificados de las aplicaciones educativas. El valor más bajo (1) representó la peor puntuación, y el valor más alto (5) representó la mejor puntuación en la siguiente escala:

- 5 puntos: Totalmente de acuerdo
- 4 puntos: De acuerdo
- 3 puntos: Neutro
- 2 puntos: En desacuerdo
- 1 punto: Totalmente en desacuerdo

La puntuación global de cada aplicación educativa fue la suma de las sub-puntuaciones de todos los aspectos antes mencionados.

Capítulo 4. Resultados

4.4.1.2. Resultados

Según lo propuesto por Malganova y Rahkimova [76], Queirós et al. [77], Abdullah y Ward [66], se consideraron las plataformas Moodle [78], Sakai [79] y Atutor [80] para realizar la evaluación comparativa de EduGene. Moodle es un sistema gratuito de gestión de aprendizaje en línea que permite a los educadores crear su propio sitio privado lleno de cursos dinámicos. Sakai es una plataforma abierta, flexible y rica en funciones para el aprendizaje, la enseñanza y la colaboración. Por último, ATutor es un *Open Source LMS* gratuito diseñado para desarrollar cursos en línea y crear contenido de *e-learning*.

Tabla 4.1 Resultados de la evaluación cualitativa

Aspecto	Moodle Puntuación	Atutor Puntuación	Sakai Puntuación	EduGene Puntuación
N1				
C1	5	5	4	2
C2	4	4	4	4
N2				
C1	5	4	4	5
C2	4	4	4	5
C3	4	3	3	5
N3				
C1	4	3	3	5
C2	4	3	3	5
C3	4	3	3	5
N4				
C1	4	3	3	5
C2	5	5	5	5
Total	43	37	36	46

La Tabla 4.1 y la Figura 4.13 muestran los resultados obtenidos después de evaluar los aspectos usando la escala de Likert mencionada anteriormente. En cuanto al aspecto N1-C1, se encontró que Moodle y ATutor capturan adecuadamente la respuesta del usuario. Sakai también logra

Capítulo 4. Resultados

capturar la respuesta del usuario, pero con menos capacidad, mientras que EduGene no tiene esta capacidad. Sin embargo, posee la interfaz gráfica que permitirá captar la respuesta del usuario en el futuro. En cuanto al aspecto N1-C2, se encontró que todas las aplicaciones tienen soporte visual, pero no son accesibles para todos los usuarios existentes. En lo que respecta a N2-C1, los resultados revelaron que tanto Moodle como EduGene tienen un diseño ordenado, pero ATutor y Sakai necesitan mejorarlo.

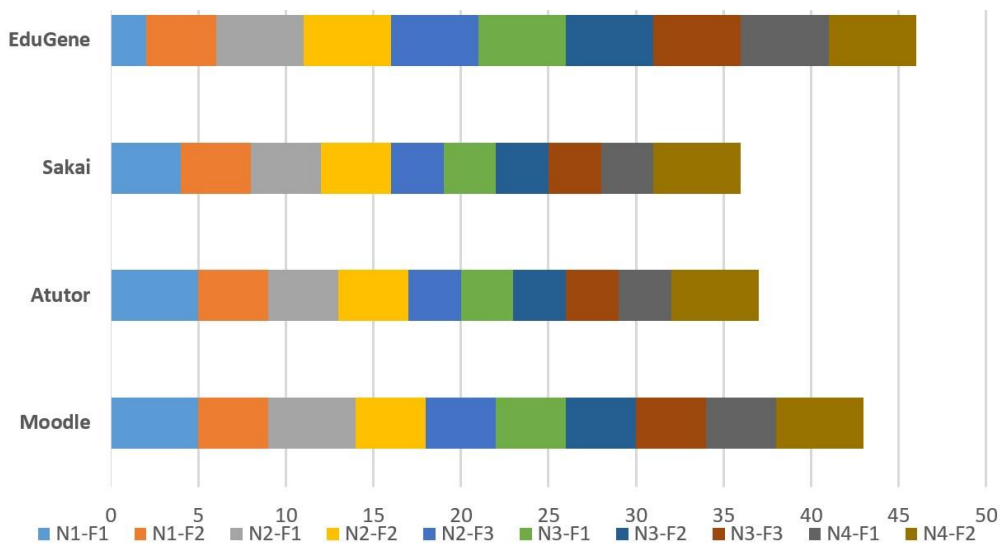


Figura 4.13 Resultados de la evaluación cualitativa

Los resultados de la evaluación del aspecto N2-C2 demostraron que el diseño basado en patrones de diseño que proporciona EduGene facilita el aprendizaje. ATutor y Sakai también desarrollan actividades efectivas, pero no dependen de los patrones de diseño; por lo tanto, no es posible asegurar que estas plataformas realmente faciliten el aprendizaje. Del mismo modo, para el aspecto N2-C3, se encuentra que gracias al uso de los patrones de diseño, el contenido en EduGene se presenta de manera eficiente. Sin embargo, el diseño de las aplicaciones generado por las otras plataformas no es muy claro y comprensible. En cuanto al aspecto N3-C1, el análisis reveló que EduGene ajusta su contenido a diferentes plataformas y dispositivos. Moodle también tiene esta capacidad, pero se encontraron defectos en su diseño sensible. En cuanto a ATutor y Sakai, estas plataformas no utilizan todos los diseños de respuesta. En otras palabras, el diseño de EduGene y Moodle facilitan el aprendizaje y la interacción con la aplicación, mientras que ATutor y Sakai no poseen interfaces amigables.

Capítulo 4. Resultados

Por último, en cuanto al aspecto N4-C1, los resultados demuestran que EduGene utiliza un amplio repertorio de plataformas y dispositivos, para lo cual se basa en tecnologías actuales. Por otro lado, las otras aplicaciones sólo se visualizan a través de un navegador web, aunque, Moodle utiliza plataformas Android™ y MacOS®. En cuanto al aspecto N4-C2, se encontró que todas las plataformas apuntan a facilitar el proceso de enseñanza-aprendizaje. En conclusión, la herramienta desarrollada con el componente generador de código sobresale entre otras aplicaciones educativas y es muy similar a Moodle; Sin embargo, Moodle carece de algunas de las características de EduGene. Además, parece que ATutor y Sakai necesitan mejorar su diseño y despliegue de contenido para llamar la atención de estudiantes y profesores.

4.4.2. Evaluación cuantitativa

En la siguiente sección, se presenta el diseño de la evaluación cuantitativa y los resultados obtenidos a través del uso de la herramienta y/o componentes.

4.4.2.1. Diseño de la evaluación

La *ISO (International Organization for Standardization, Organización Internacional de Normalización)* e *IEC (International Electrotechnical Commission, Comisión Electrotécnica Internacional)* propusieron la norma internacional *ISO/IEC 25000: SQUARE (Software Quality Requirements and Evaluation, Requisitos y evaluación de la calidad del software)*. Este modelo se basa en *ISO/IEC 9126* e *ISO/IEC 14598*, y se divide en cinco temas: 1) Gestión de calidad (2500n), 2) Modelo de calidad (2501n), 3) Medidas de calidad (2502n), 4) Calidad Requisitos (2503n) y 5-Evaluación de la calidad (2504n) [81]–[83]. Teniendo en cuenta este modelo de calidad, se signaron cuatro aspectos clave de las aplicaciones educativas eficaces a las sub características de ISO / IEC 2502n (Medidas de Calidad). Los factores de calidad evaluados y sus correspondientes métricas de software se describen a continuación.

Necesidad 1 (NI): Educación.

Fulantelli et al. [84] y Peña Ayala et al. [85] sostienen que una aplicación educativa exitosa contiene actividades educativas que facilitan el proceso de enseñanza-aprendizaje. Del mismo modo, Aparicio et al. [64] mencionan que una aplicación educativa eficaz basa su uso en modelos educativos o teorías de aprendizaje para justificar sus objetivos y alcanzarlos.

Capítulo 4. Resultados

Característica 1 (C1): Actividades educativas usadas.

Característica 2 (C2): Teorías del aprendizaje aplicadas.

Necesidad 2 (N2): Efectividad.

Xu et al. [86] afirman que para determinar la efectividad de una aplicación educativa generada, se necesitan considerar sus características en términos de tiempo de ejecución, objetivos alcanzados y número de pasos. El tiempo de ejecución se obtuvo a partir del promedio realizado a un conjunto de 15 ejecuciones elaboradas a cada herramienta educativa tomando como medida de tiempo los minutos. Además, para la evaluar los objetivos alcanzados se propuso un conjunto de 15 objetivos que favorecen la enseñanza-aprendizaje de los usuarios. Los objetivos planteados se presentan a continuación.

1. Aumentar el acceso a la información.
2. Aumentar la participación de alumnos y docentes.
3. Otorgar el acceso a la educación a personas jóvenes y adultas.
4. Garantizar el acceso y la permanencia de los miembros.
5. Proporcionar una educación bilingüe y multicultural.
6. Apoyar la inclusión de contenido educativo.
7. Transferir adecuadamente los conocimientos.
8. Mejorar el nivel de adquisición de los conocimientos.
9. Incluir un sistema de evaluación.
10. Incluir un sistema de retro-alimentación.
11. Fortalecer los valores para tener una sociedad justa.
12. Mejorar y adaptar el diseño del contenido educativo.
13. Permitir el mantenimiento continuo del contenido educativo.
14. Proporcionar una red que permita incrementar la colaboración entre los integrantes.
15. Reforzar la investigación científica y tecnológica.

Característica 1 (C1): Tiempo de ejecución.

Característica 2 (C2): Objetivos alcanzados.

Capítulo 4. Resultados

Característica 3 (C3): Número de pasos para realizar una actividad educativa.

Necesidad 3 (N3): Apoyo.

Porter et al. [87] y, Poulova y Simonova [68] afirman que una aplicación educativa eficaz ofrece apoyo a los usuarios en caso de que quieran hacer modificaciones o utilizar una herramienta en particular.

Característica 1 (C1): Soporte para el uso de la herramienta.

Característica 2 (C2): Soporte para modificar la herramienta.

Característica 3 (C3): Plataformas compatibles.

Característica 4 (C4): Dispositivos compatibles.

4.4.2.2. Resultados

Para la evaluación cuantitativa, se analizó la generación de aplicaciones de Física Educativa, donde cada sistema fue utilizado para desarrollar aplicaciones educativas que facilitaran el proceso de enseñanza-aprendizaje de la Física.

Como se muestra en la Tabla 4.2, EduGene, Moodle y Sakai se utilizaron para crear tres aplicaciones educativas diferentes, mientras que ATutor sólo desarrolla un curso. Además, nuestros resultados revelaron que EduGene tiene alguna desventaja si se compara con Moodle y Sakai, ya que no permite a los usuarios almacenar información. Por otra parte, la gran ventaja de EduGene es el uso de interfaces gráficas de usuario, y los patrones de diseño apoyan el diseño gráfico.

Tabla 4.2 Medición del aspecto educativo

Aspecto	Moodle	Atutor	Sakai	EduGene
---------	--------	--------	-------	---------

Capítulo 4. Resultados

Aplicaciones educativas usadas	3	1	3	3
Teorías del aprendizaje aplicadas	4	1	4	4

En cuanto a la efectividad de las aplicaciones educativas, se analizó el tiempo de ejecución, los objetivos alcanzados y el número de pasos necesarios para completar una actividad educativa. En cuanto al primer aspecto, la Tabla 4.3 muestra que el tiempo necesario para realizar una actividad educativa en EduGene y Moodle es similar. Además, se encontró que tarda más tiempo en completar una actividad usando Sakai y ATutor. Además, a partir de un conjunto de 15 metas a lograr en una actividad educativa, todas las plataformas alcanzaron más de la mitad de ellas. Sin embargo, EduGene sobresalió con 13 metas alcanzadas. Finalmente, todos los sistemas requieren el mismo número de pasos para realizar una actividad educativa.

Tabla 4.3 Medición del aspecto de efectividad

Aspecto	Moodle	Atutor	Sakai	EduGene
Tiempo tomado	~5min	~7min	~6min	~5min
Objetivos alcanzados	12	8	10	13
Número de pasos	5	5	5	5

Como último paso en la evaluación cualitativa, se analizó el soporte técnico que ofrece cada plataforma, incluyendo el soporte para utilizar una herramienta y soporte para modificar una herramienta. Además, se evaluó el número de dispositivos y plataformas soportados por cada generador de aplicaciones. En este sentido, la Tabla 4.4 muestra que Moodle, ATutor y Sakai ofrecen mayor soporte técnico en comparación con EduGene. Asimismo, Moodle ofrece mayores posibilidades de configuración de plataformas que cualquier otro sistema bajo evaluación. En cuanto a plataformas y dispositivos soportados, se encontró que EduGene ofrece soporte para una gama más amplia, fomentando así la interacción de los estudiantes y profesores con las tecnologías de la información.

Tabla 4.4 Medición del aspecto del soporte.

Aspecto	Moodle	Atutor	Sakai	EduGene
---------	--------	--------	-------	---------

Capítulo 4. Resultados

Soporte para utilizar 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

En conclusión, la Figura 4.14 muestra que EduGene se destaca en varios puntos de métricas de calidad, pero carece de algunas otras características que los sistemas restantes en evaluación poseen, como el soporte técnico.

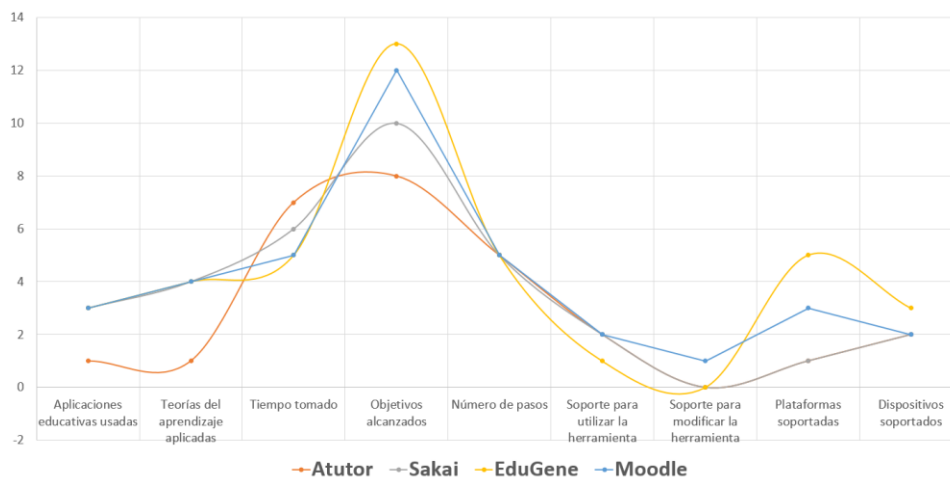


Figura 4.14 Resultados de la evaluación cuantitativa

Capítulo 5. Conclusiones y recomendaciones

Capítulo 5. Conclusiones y recomendaciones

Para finalizar la escritura de este trabajo de tesis se presentan a continuación las conclusiones obtenidas en la realización de este documento y las recomendaciones a futuro para mejorar y expandir este trabajo.

5.1. Conclusiones

En este trabajo de tesis se logró desarrollar un componente de software que cumple con los objetivos planteados al iniciar el proyecto, donde a partir del procesamiento de imágenes y el uso de patrones de diseño de interfaz de usuario fue posible generar aplicaciones multi-dispositivo y multi-plataforma en el ámbito educativo.

Durante la elaboración del trabajo presentado se aprendió que los patrones de diseño de interfaz de usuario facilitaron el desarrollo de las aplicaciones educativas sin afectar su calidad, gracias a que proveen la solución a diferentes problemas de diseño. Por otra parte, el proceso de identificación de patrones de diseño de interfaz de usuario a través de una imagen provee dos grandes beneficios, el primero es que permite visualizar el contenido de la aplicación a través de un icono y el segundo es cambiar el modo en cómo construir una aplicación para permitir a la computadora identificar lo que un desarrollador quiere construir. Además, se demostró que los patrones de diseño permiten construir nuevos patrones que combinan los beneficios de cada uno.

Este trabajo ofrece tres aportes en el área de desarrollo de software automático de aplicaciones: 1) Estableció un conjunto de patrones de diseño de interfaz de usuario para facilitar el desarrollo de aplicaciones educativas; 2) Generó patrones compuestos a partir de dos o más patrones de diseño de interfaz de usuario para comprobar la compatibilidad de los patrones de diseño y sus beneficios; e 3) Implementó técnicas de reconocimiento de patrones en una imagen delimitada para identificar un conjunto de patrones de diseño y así generar el código fuente de aplicaciones educativas.

5.2. Recomendaciones

Tomando en cuenta el componente de software desarrollado es necesario considerar las siguientes recomendaciones:

Capítulo 5. Conclusiones y recomendaciones

Con base en las aplicaciones educativas establecidas en esta tesis se recomienda agregar nuevas actividades educativas que permitan considerar las aplicaciones educativas que no fueron contempladas como los son *WebQuest*, tutoriales, entre otros. Así mismo agregar al componente desarrollado nuevos patrones de diseño para aumentar el repositorio y abarcar más aplicaciones educativas.

Otra recomendación es acerca de los patrones de diseño de interfaz de usuario debido a su uso en diferentes dominios como lo son: el de negocios, el entretenimiento, la medicina, entre otros. Por lo tanto, se requiere realizar un análisis de los patrones de diseño existentes y establecer una relación con los diferentes dominios que existen para incorporarlos al componente desarrollado en esta tesis.

Con respecto a los resultados que provee la red neuronal es necesario obtener un mínimo de 500 bosquejos para cada patrón individual o compuesto de diseño de interfaz de usuario, para aumentar la eficiencia al reconocer un patrón de diseño.

Además, la tecnología va aumentando constantemente lo que provoca que aparezcan nuevas plataformas en los diferentes dispositivos, o en caso contrario dejen de tener soporte para las ya existentes. Consecuentemente a esto es necesario realizar un nuevo análisis de las tecnologías vigentes para seguir actualizando los proyectos y evitar que el componente generador de código se vuelva obsoleto en poco tiempo.

Productos académicos

Productos académicos

A continuación, se presentan los trabajos derivados de este proyecto de investigación.

Artículos JCR



Cesar Augusto Cortes-Camarillo, Giner Alor-Hernández, Laura Nely Sánchez-Morales, Viviana Yarel Rosales-Morales, Lisbeth Rodríguez-Mazahua, José Luis Sánchez-Cervantes. **EduGene: A UIDPs-based educational application generator for multiple devices and platforms.** Behaviour & Information Technology. **Estatus:** Enviado

Artículos en otros índices

Advances in Software Engineering
and its Applications

Matías Abarado Mentado
Perfecto Malaquias Quintero Flores (eds.)

Cesar Augusto Cortes-Camarillo, Giner Alor-Hernández, Beatriz Alejandra Olivares-Zepahua, Lisbeth Rodríguez-Mazahua, Silvestre Gustavo Peláez-Camarena. **Análisis comparativo de patrones de diseño de interfaz de usuario para el desarrollo de aplicaciones educativas.** Research in Computing Science 126: 31-41. ISSN: 1870-4069. Indexing: LATINDEX, PERIODICA, DBLP.



Conferencias internacionales



Cesar Augusto Cortes Camarillo, Viviana Yarel Rosales Morales, Laura Nely Sanchez Morales, Giner Alor-Hernández, Lisbeth Rodríguez Mazahua. **Atila: A UIDPs-based educational application generator for mobile devices.** 27th International Conference on Electronics, Communications and Computers. IEEE Press. ISSN: 2474-9044. DOI: 10.1109/CONIELECOMP.2017.7891810

Productos académicos

Capítulos de libro



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.** New Perspectives on Applied Industrial Tools and Techniques (pp. 211-231). Springer Verlag. ISSN: 2365-0532. DOI: 10.1007/978-3-319-56871-3_11

Registros de derechos de autor



Alor-Hernández, G. & **Cortes-Camarillo, C. A.**, ATILA: Generador de aplicaciones educativas para dispositivos móviles basado en patrones de diseño de interfaz de usuario. **Estatus: Registrado.** INDAUTOR. Número de registro: **03-2016-092813015100-01**

Proyectos de Investigación



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. TecNM. **Estatus: Aprobado.**

Estancias Académicas



Instituto Tecnológico de Culiacán. Periodo: 8 de Mayo – 9 de Junio de 2017. Dr. Ramón Zatarain Cabada.

Terminología involucrada

Terminología involucrada

HTML. (*Hypertext Markup Language*, Lenguaje de Marcas de Hipertexto), es el lenguaje de marcado predominante para la construcción de páginas Web.

TI. (*Technology Information*, Tecnología de la información) se entiende como aquellas herramientas y métodos empleados para recabar, retener, manipular o distribuir información.

IU. Siglas de Interfaz de Usuario.

W3C: (*World Wide Web Consortium*), es un consorcio internacional donde las organizaciones, miembros y el público en general trabajan conjuntamente para desarrollar estándares Web.

Web. Sistema de documentos de hipertexto y/o hipermedios enlazados y accesibles a través de Internet.

XML: (*eXtensible Markup Language*, Lenguaje de Mercado Extensible), es un meta-lenguaje desarrollado por el *World Wide Web Consortium* (W3C).

Bibliografía

Bibliografía

- [1] P. Querejeta Simbeni, “Procesamiento digital de imágenes,” *Univ. Nac. del Sur*, pp. 1–5, 2015.
- [2] Ieee, “IEEE Standard Glossary of Software Engineering Terminology,” *Office*, vol. 121990, no. 1, p. 1, 1990.
- [3] I. The MathWorks, “Getting Started with MATLAB,” *MATLAB Lang. Tech. Comput.*, p. 136, 2001.
- [4] I. The MathWorks, “Image Processing Toolbox™ User’s Guide,” *MathWorks Lang. Tech. Comput.*, p. 766, 2015.
- [5] H. The MathWorks, Inc.; Hudson Beale, Mark; T. Hagan, Martin; B. Demuth, “Neural Network Toolbox™ Getting Started Guide,” *MathWorks Lang. Tech. Comput.*, p. 120, 2015.
- [6] K. Hoa Ly, *Use of a Smartphone Application in the Treatment of Depression: The New Wave of Digital Tools in Psychological Treatment*. Linköping University Electronic Press, 2015.
- [7] M. Duggan and A. Smith, “Cell Internet Use 2013,” *Pew Internet Washington, D.C.*, p. 15, 2013.
- [8] P. G. S. de Viguera, “Análisis de la calidad de aplicaciones educativas para dispositivos móviles,” *Trab. FIN GRADO*, 2014.
- [9] J. J. Nieto Ruiz, “Usabilidad en dispositivos Smart TV,” *Tesis Licenciatura. Univ. Oberta Catalunya*, p. 60, 2013.
- [10] Á. Santiago, Raúl; Trbaldo, Susana; Kamijo, Mercedes; Fernández, *Mobile Learning: Nuevas realidades en el aula*. Editorial Oceano, 2015.
- [11] Z. del P. Martínez Pereiro, Brais; Melián Herrera, “LeeMed, aplicación para dispositivos móviles en el ámbito de la discapacidad visual,” *Univ. las Palmas Gran Canar. - Trab.*

Bibliografía

fin grado, p. 80.

- [12] J. García Portero, Raúl; Ceballos Villach, “Introducción al Firefox OS,” *Univ. Oberta Catalunya - Trab. Final Carrera/Grado*, 2013.
- [13] C. Chase, “Using Design Patterns in User Interface Design,” *Univ. Cincinnati - Tesis grado*, p. 90, 2012.
- [14] J. Tidwell, *Designing interfaces: Patterns for effective interaction design*. “O’Reilly Media, Inc.,” 2010.
- [15] T. Neil, *Mobile design pattern gallery: UI patterns for smartphone apps*. “O’Reilly Media, Inc.,” 2014.
- [16] A. Ferrate *et al.*, *Building Web Apps for Google TV*. “O’Reilly Media, Inc.,” 2011.
- [17] Intel Corporation, “The Next Unit of Computing Drive your 10 foot experience,” 2013.
- [18] R. Vázquez-Ramírez, H. Marín-Vega, and G. Alor-Hernández, “Overview of UI Patterns on Mobile Platform for Educational Applications.,” *Res. Comput. Sci.*, vol. 65, pp. 95–102, 2013.
- [19] H. Heitkötter, S. Hanschke, and T. A. Majchrzak, “Evaluating cross-platform development approaches for mobile applications,” in *Lecture Notes in Business Information Processing*, 2013, vol. 140 LNBIP, pp. 120–138.
- [20] A. D. Brucker and M. Herzberg, “On the Static Analysis of Hybrid Mobile Apps,” in *Engineering Secure Software and Systems: 8th International Symposium, ESSoS 2016, London, UK, April 6--8, 2016. Proceedings*, J. Caballero, E. Bodden, and E. Athanasopoulos, Eds. Cham: Springer International Publishing, 2016, pp. 72–88.
- [21] S. C. Pokress and J. J. D. Veiga, “MIT App Inventor: Enabling Personal Mobile Computing,” *arXiv Prepr. arXiv1310.2830*, p. 3, 2013.
- [22] T.-M. Grønli, G. Ghinea, and M. Younas, “Context-aware and automatic configuration of mobile devices in cloud-enabled ubiquitous computing,” *Pers. Ubiquitous Comput.*,

Bibliografía

- vol. 18, no. 4, pp. 883–894, 2014.
- [23] I. . Marin, F. . Ortin, G. . Pedrosa, and J. . Rodriguez, “Generating native user interfaces for multiple devices by means of model transformation,” *Front. Inf. Technol. Electron. Eng.*, vol. 16, no. 12, pp. 995–1017, 2015.
- [24] S. Vaupel, G. Taentzer, R. Gerlach, and M. Guckert, “Model-driven development of mobile applications for Android and iOS supporting role-based app variability,” *Software and Systems Modeling*, pp. 1–29, 2016.
- [25] E. R. Núñez-Valdez, V. García-Díaz, J. M. C. Lovelle, Y. S. Achaerandio, and R. González-Crespo, “A model-driven approach to generate and deploy videogames on multiple platforms,” *J. Ambient Intell. Humaniz. Comput.*, pp. 1–13, 2016.
- [26] D. Fogli, L. Parasiliti Provenza, and C. Bernareggi, “A universal design resource for rich Internet applications based on design patterns,” *Univers. Access Inf. Soc.*, vol. 13, no. 2, pp. 205–226, 2014.
- [27] H. Heitkötter and T. A. Majchrzak, “Cross-platform development of business apps with MD2,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2013, vol. 7939 LNCS, pp. 405–412.
- [28] H. Heitkötter, T. A. Majchrzak, and H. Kuchen, “Cross-platform Model-driven Development of Mobile Applications with Md2,” in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, 2013, pp. 526–533.
- [29] L. O. Colombo-Mendoza, G. Alor-Hernández, A. Rodríguez-gonzález, and R. Valencia-garcía, “MobiCloUP!: a PaaS for cloud services-based mobile applications,” *Autom. Softw. Eng.*, vol. 21, no. 3, pp. 391–437, 2014.
- [30] R. Vásquez-Ramirez, G. Alor-Hernández, and A. Rodríguez-González, “Athena: A Hybrid Management System for Multi-device Educational Content,” *Comput. Appl. Eng. Educ.*, vol. 22, no. 4, pp. 750–763, 2014.

Bibliografía

- [31] R. Vásquez-Ramírez, G. Alor-Hernández, C. Sánchez-Ramírez, J. Guzmán-Luna, R. Zatarain-Cabada, and M.-L. Barrón-Estrada, “AthenaTV: an authoring tool of educational applications for TV using android-based interface design patterns,” *New Rev. Hypermedia Multimed.*, vol. 20, no. 3, pp. 251–280, 2014.
- [32] H. Desruelle, S. Isenberg, A. Botsikas, P. Vergori, and F. Gielen, “Accessible user interface support for multi-device ubiquitous applications: architectural modifiability considerations,” *Univers. Access Inf. Soc.*, vol. 15, no. 1, pp. 5–19, 2016.
- [33] I. Hussein, M. Mahmud, and A. Osman Md Tap, “HCI knowledge for UX practices in the web development process,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014, vol. 8518 LNCS, no. PART 2, pp. 116–126.
- [34] P. Cremonesi, M. Elahi, and F. Garzotto, “User interface patterns in recommendation-empowered content intensive multimedia applications,” *Multimed. Tools Appl.*, pp. 1–35, 2016.
- [35] M. P. A. Balayan, V. V. B. Conoza, J. M. M. Tolentino, R. C. Solamo, and R. P. Feria, “On evaluating skillville: An educational mobile game on visual perception skills,” in *IISA 2014 - 5th International Conference on Information, Intelligence, Systems and Applications*, 2014, pp. 69–74.
- [36] C. E. de A. Freire and M. Painho, “Development of a Mobile Mapping Solution for Spatial Data Collection Using Open-Source Technologies,” *Procedia Technol.*, vol. 16, pp. 481–490, 2014.
- [37] Z. Ruttkay, J. Bényei, and Z. Sárközi, “Evaluation of interactive children book design: The case study of ‘Little Rooster,’” *Adv. Intell. Syst. Comput.*, vol. 292, pp. 109–117, 2014.
- [38] H. Ovesleová, “E-learning platforms and lacking motivation in students: Concept of adaptable UI for online courses,” in *Lecture Notes in Computer Science (including*

Bibliografía

- subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics*), 2015, vol. 9188, pp. 218–227.
- [39] M. Milutinović, A. Labus, V. Stojiljković, Z. Bogdanović, and M. Despotović-Zrakić, “Designing a mobile language learning system based on lightweight learning objects,” *Multimed. Tools Appl.*, vol. 74, no. 3, pp. 903–935, 2013.
- [40] E. Zargaran *et al.*, “The electronic trauma health record: Design and usability of a novel tablet-based tool for trauma care and injury surveillance in low resource settings,” *J. Am. Coll. Surg.*, vol. 218, no. 1, pp. 41–50, 2014.
- [41] Z. Zhan and H. Mei, “Academic self-concept and social presence in face-to-face and online learning: Perceptions and effects on students’ learning achievement and satisfaction across environments,” *Comput. Educ.*, vol. 69, pp. 131–138, 2013.
- [42] M. Aydemir, E. E. Özkeskin, and A. A. Akkurt, “A Theoretical Framework on Open and Distance Learning,” *Procedia - Soc. Behav. Sci.*, vol. 174, no. 2003, pp. 1750–1757, 2015.
- [43] Z. Zhan, P. S. W. Fong, H. Mei, and T. Liang, “Effects of gender grouping on students’ group performance, individual achievements and attitudes in computer-supported collaborative learning,” *Comput. Human Behav.*, vol. 48, pp. 587–596, 2015.
- [44] S. K. W. Chu, C. K. K. Chan, and A. F. Y. Tiwari, “Using blogs to support learning during internship,” *Comput. Educ.*, vol. 58, no. 3, pp. 989–1000, 2012.
- [45] P. A. Tess, “The role of social media in higher education classes (real and virtual)-A literature review,” *Computers in Human Behavior*, vol. 29, no. 5, 2013.
- [46] S. L. Watson, J. Loizzo, W. R. Watson, C. Mueller, J. Lim, and P. A. Ertmer, “Instructional design, facilitation, and perceived learning outcomes: an exploratory case study of a human trafficking MOOC for attitudinal change,” *Educ. Technol. Res. Dev.*, vol. 64, no. 6, pp. 1273–1300, 2016.
- [47] A. Margaryan, M. Bianco, and A. Littlejohn, “Instructional quality of Massive Open

Bibliografía

- Online Courses (MOOCs),” *Comput. Educ.*, vol. 80, pp. 77–83, 2015.
- [48] M. Biasutti and H. El-Deghaidy, “Using Wiki in teacher education: Impact on knowledge management processes and student satisfaction,” *Comput. Educ.*, vol. 59, no. 3, pp. 861–872, 2012.
- [49] A. Hutchison and J. Colwell, “Using a wiki to facilitate an online professional learning community for induction and mentoring teachers,” *Educ. Inf. Technol.*, vol. 17, no. 3, pp. 273–289, 2012.
- [50] M. Sheibley, “Mobile Patterns.” 2013.
- [51] A. Toxboe, “UI Patterns User Interface Design pattern Library.” 2016.
- [52] UNITiD, “Android Patterns.” 2016.
- [53] L. G. Developer, “LG Developer.” 2013.
- [54] A. TV, “Android TV Patterns.” 2015.
- [55] A. TV, “Human Interface Guidelines.” 2016.
- [56] J. Engel, C. Märtin, and P. Forbrig, “HCI patterns as a means to transform interactive user interfaces to diverse contexts of use,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2011, vol. 6761 LNCS, no. PART 1, pp. 204–213.
- [57] K. Klemisch, I. Weber, and B. Benatallah, “Context-Aware UI Component Reuse,” in *Advanced Information Systems Engineering: 25th International Conference, CAiSE 2013, Valencia, Spain, June 17-21, 2013. Proceedings*, C. Salinesi, M. C. Norrie, and Ó. Pastor, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 68–83.
- [58] H. Boley, A. Paschke, and O. Shafiq, “RuleML 1.0: The overarching specification of web rules,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2010, vol. 6403 LNCS, pp. 162–178.

Bibliografía

- [59] P. Twining, R. S. Heller, M. Nussbaum, and C.-C. Tsai, “Some guidance on conducting and reporting qualitative studies,” *Comput. Educ.*, vol. 106, pp. A1–A9, 2017.
- [60] M.-H. Chiu, J.-W. Lin, and C.-C. Chou, “Content Analysis of Conceptual Change Research and Practice in Science Education: From Localization to Globalization,” in *Science Education Research and Practices in Taiwan: Challenges and Opportunities*, M.-H. Chiu, Ed. Singapore: Springer Singapore, 2016, pp. 89–131.
- [61] Y. J. Joo, H. W. Lee, and Y. Ham, “Integrating user interface and personal innovativeness into the TAM for mobile learning in Cyber University,” *J. Comput. High. Educ.*, vol. 26, no. 2, pp. 143–158, 2014.
- [62] H. Mohammadi, “Investigating users’ perspectives on e-learning: An integration of TAM and IS success model,” *Comput. Human Behav.*, vol. 45, pp. 359–374, 2015.
- [63] T. J. McGill, J. E. Klobas, and S. Renzi, “Critical success factors for the continuation of e-learning initiatives,” *Internet High. Educ.*, vol. 22, pp. 24–36, 2014.
- [64] M. Aparicio, F. Bacao, and T. Oliveira, “Cultural impacts on e-learning systems’ success,” *Internet High. Educ.*, vol. 31, pp. 58–70, 2016.
- [65] W. W. Porter, C. R. Graham, R. G. Bodily, and D. S. Sandberg, *A qualitative analysis of institutional drivers and barriers to blended learning adoption in higher education*, vol. 28. 2016, pp. 17–27.
- [66] F. Abdullah and R. Ward, “Developing a General Extended Technology Acceptance Model for E-Learning (GETAMEL) by analysing commonly used external factors,” *Comput. Human Behav.*, vol. 56, pp. 238–256, 2016.
- [67] N. Harrati, I. Bouchrika, A. Tari, and A. Ladjailia, “Exploring user satisfaction for e-learning systems via usage-based metrics and system usability scale analysis,” *Comput. Human Behav.*, vol. 61, pp. 463–471, 2016.
- [68] P. Poulouva and I. Simonova, “E-learning Reflected in Research Studies in Czech Republic: Comparative Analyses,” *Procedia - Soc. Behav. Sci.*, vol. 116, pp. 1298–1304,

Bibliografía

- 2014.
- [69] A. Garcia-Cabot, L. de-Marcos, and E. Garcia-Lopez, “An empirical study on m-learning adaptation: Learning performance and learning contexts,” *Comput. Educ.*, vol. 82, pp. 450–459, 2015.
- [70] M. Alshammari, R. Anane, and R. J. Hendley, “Design and Usability Evaluation of Adaptive e-learning Systems Based on Learner Knowledge and Learning Style,” in *Human-Computer Interaction -- INTERACT 2015: 15th IFIP TC 13 International Conference, Bamberg, Germany, September 14-18, 2015, Proceedings, Part II*, J. Abascal, S. Barbosa, M. Fetter, T. Gross, P. Palanque, and M. Winckler, Eds. Cham: Springer International Publishing, 2015, pp. 584–591.
- [71] M. A. Lubiano, A. Salas, S. de da R. De Sáa, M. Montenegro, and M. Á. Gil, “An empirical analysis of the coherence between fuzzy rating scale-and likert scale-based responses to questionnaires,” in *Advances in Intelligent Systems and Computing*, 2017, vol. 456, pp. 329–337.
- [72] G. Norman, “Likert scales, levels of measurement and the ‘laws’ of statistics,” *Adv. Heal. Sci. Educ.*, vol. 15, no. 5, pp. 625–632, 2010.
- [73] J. Sjövall, U. Bitzén, E. Kjellén, P. Nilsson, P. Wahlberg, and E. Brun, “Qualitative interpretation of PET scans using a Likert scale to assess neck node response to radiotherapy in head and neck cancer,” *Eur. J. Nucl. Med. Mol. Imaging*, vol. 43, no. 4, pp. 609–616, 2016.
- [74] M. Dantlgraber, E. Wetzal, P. Schützenberger, S. Stieger, and U.-D. Reips, “Simple construct evaluation with latent class analysis: An investigation of Facebook addiction and the development of a short form of the Facebook Addiction Test (F-AT).,” *Behav. Res. Methods*, vol. 48, no. 3, pp. 869–79, 2016.
- [75] R. Matilainen, A. Koliseva, P. Valto, and J. Väliisaari, “Reconstruction of undergraduate analytical chemistry laboratory course,” *Anal. Bioanal. Chem.*, vol. 409, no. 1, pp. 3–10,

Bibliografía

- 2017.
- [76] I. Malganova and A. Rahkimova, “The Usage of Distance Education Practice of the World Leading Universities in Russian Multiethnic Region,” *Procedia - Soc. Behav. Sci.*, vol. 191, pp. 2622–2625, 2015.
- [77] R. Queirós, J. P. Leal, and J. C. Paiva, “Integrating Rich Learning Applications in LMS,” in *State-of-the-Art and Future Directions of Smart Learning*, Y. Li, M. Chang, M. Kravcik, E. Popescu, R. Huang, Kinshuk, and N.-S. Chen, Eds. Singapore: Springer Singapore, 2016, pp. 381–386.
- [78] L. M. S. Moodle, “Moodle - Open-source learning platform | Moodle.org.” 2017.
- [79] L. M. S. Sakai, “Introducing Sakai 11 | Sakai.” 2014.
- [80] L. M. S. Atutor, “ATutor Learning Management System: Information:” 2015.
- [81] I. 25000, “The ISO/IEC 25000 series of standards.” 2015.
- [82] N. Bevan, J. Carter, J. Earthy, T. Geis, and S. Harker, “New ISO Standards for Usability, Usability Reports and Usability Measures,” in *Human-Computer Interaction. Theory, Design, Development and Practice : 18th International Conference, HCI International 2016, Toronto, ON, Canada, July 17-22, 2016. Proceedings, Part I*, M. Kurosu, Ed. Cham: Springer International Publishing, 2016, pp. 268–278.
- [83] L. Garcés, A. Ampatzoglou, P. Avgeriou, and E. Y. Nakagawa, “Quality attributes and quality models for ambient assisted living software systems: A systematic mapping,” *Inf. Softw. Technol.*, vol. 82, pp. 121–138, 2017.
- [84] G. Fulantelli, D. Taibi, and M. Arrigo, “A framework to support educational decision making in mobile learning,” *Comput. Human Behav.*, vol. 47, pp. 50–59, 2015.
- [85] A. Peña-Ayala, H. Sossa, and I. Méndez, “Activity theory as a framework for building adaptive e-learning systems: A case to provide empirical evidence,” *Comput. Human Behav.*, vol. 30, pp. 131–145, 2014.

Bibliografía

- [86] D. Xu, W. W. Huang, H. Wang, and J. Heales, “Enhancing e-learning effectiveness using an intelligent agent-supported personalized virtual learning environment: An empirical investigation,” *Inf. Manag.*, vol. 51, no. 4, pp. 430–440, 2014.
- [87] W. W. Porter, C. R. Graham, K. A. Spring, and K. R. Welch, “Blended learning in higher education: Institutional adoption and implementation,” *Comput. Educ.*, vol. 75, pp. 185–195, 2014.