Instituto Tecnológico de Orizaba

División de Estudios de Posgrado e Investigación
Maestría en Sistemas Computacionales

PROYECTO DE TESIS

"Sistema Generador de aplicaciones de NFC para Android**."**

PRESENTADO POR:

ISC JOSÉ MARÍA MARTÍNEZ MERLO M09011289

PARA OBTENER EL GRADO DE:

MAESTRO EN SISTEMAS COMPUTACIONALES

DIRECTOR DEL ANTEPROYECTO DE TESIS:

M.R.T. IGNACIO LÓPEZ MARTÍNEZ

Índice

Contenido

Índice de figuras

No se encuentran elementos de tabla de ilustraciones.

Índice de tablas

Resumen

La problemática con las tecnologías emergentes, en especial NFC, es que caen en desuso, son sustituidas por otras tecnologías o simplemente desaparecen porque no son utilizadas cómo prometían, la función principal de NFC es que los usuarios puedan realizar sus configurar sencillamente teniendo la menor interacción en esta. La eliminación de la figura del programador o bien la posibilidad de que los propios usuarios diseñen y construyan sus aplicaciones es una tendencia que cada vez toma más auge; un sistema generador de aplicaciones de NFC permitirá a los usuarios recurrir a la tecnología con las particularidades que cada consumidor necesite.

El generador de aplicaciones de NFC establecerá una gama de usos generales y definirá una serie de acciones específicas que el usuario quiera personalizar permitiendo entonces la construcción de aplicaciones NFC específicas para usos particulares, las cuales se modifican y actualizan por los propios usuarios sin la necesidad de la intervención de un programador.

Introducción

Al ser lanzadas al mercado, las tecnologías emergentes prometen diversas acciones, alcances o estrategias; sin embargo, no siempre lo alcanzan o bien en el camino son orientadas a un área para el cual no fueron pensadas, tal es el caso de *Google Glass* [1] que al presentarse para el uso del consumidor fue poco recibido y posteriormente dicha tecnología fue retirada del mercado a pesar de haber tenido gran aceptación en el ámbito profesional, por su parte la tecnología *NFC* se utiliza principalmente para *contactless payment* (pagos sin contacto), con lo que no se toman en cuenta las ventajas que ofrece en diferentes tareas, adicionalmente con el poco uso de la tecnología puede provocar que sea reemplazada porque no alcanzó la aceptación adecuada.

NFC es una tecnología que utiliza frecuencias de radio para la transferencia de datos entre un emisor y un receptor, los emisores pueden estar en una gran variedad de dispositivos entre los que se encuentran teléfonos inteligentes, computadoras y tabletas. Los receptores pueden encontrarse entre los dispositivos antes mencionados o también pueden ser dispositivos que no cuentan con su propia fuente de energía denominados *NFC* pasivos, de estos, los principales son las etiquetas *NFC*, pequeños dispositivos que pueden encontrarse dentro de etiquetas, llaveros, tarjetas, entre otros.

NFC provee tres modos de operación: el primer modo es de punto a punto, donde dos dispositivos activos intercambian información entre ellos; el segundo modo de lectura/escritura, donde un dispositivo activo se comunica con un dispositivo pasivo, y por último, el modo de emulación de tarjeta, donde un dispositivo activo se utiliza como tarjeta inteligente.

La problemática que estudia esta investigación es que el desarrollo de aplicaciones que hacen uso de la tecnología *NFC* requieren conocimientos específicos, por lo que se plantea el uso de un sistema generador en donde los usuarios puedan generar aplicaciones limitadas y a medida que hagan uso de esta tecnología se incrementará su aceptación, por lo que se estudiara la manera en que se desarrollará dicho sistema.

El presente trabajo está constituido de tres capítulos: en el capítulo 1 antecedentes donde: se detalla el marco teórico, planteamiento del problema, objetivos generales y específicos, y justificación; en el capítulo 2 estado de la práctica, se detallan los trabajos relacionados y se realiza un análisis comparativo de estos; en el capítulo 3 propuesta de solución, se describe la solución, análisis de las tecnologías de información y solución propuesta. Finalmente se presentan las conclusiones.

Capítulo 1 antecedentes

1.1 Marco teórico

A continuación, se presentan los conceptos más relevantes del proyecto de tesis para una mejor comprensión de este.

1.1.1 *Near Field Communication* (NFC)

NFC es una tecnología inalámbrica que proporciona comunicación entre dos dispositivos móviles que cuenten con etiquetas *NFC* con ondas de radio de corto alcance. Utiliza la inducción de campo magnético para este fin [2].

NFC permite la comunicación entre dispositivos próximos, con un bajo nivel de poder de operación, está basado en la tecnología de *RFID (Radio Frequency Identification,* identificación por radiofrecuencias*)* y provee comunicación sin contacto entre dispositivos en un rango no mayor a 4 cm, opera en una frecuencia de 13.56 MHz y tiene una taza de transferencia de hasta 424kb/s, el rango corto elimina cualquier interferencia, por lo que es una tecnología confiable [3].

La popularidad de *NFC* se incrementó en los últimos años, gracias a esto la industria de teléfonos inteligentes produce más dispositivos con estas funcionalidades y se integran sensores *NFC* en cada vez más dispositivos [4].

NFC trabaja en modos activo y pasivo; en modo activo, los dos dispositivos equipados con NFC producen su propio campo electromagnético alternadamente para el intercambio de información. Ambos dispositivos están activos en este modo, por lo que uno desactiva su campo electromagnético durante la transferencia de datos. En modo pasivo, uno de los dispositivos actúa como un transpondedor y utiliza el campo electromagnético del otro dispositivo como su propio poder operativo [2].

Un dispositivo *NFC* puede comunicarse también con *NFC tags* (Etiquetas *NFC*), las *NFC tags* contienen una pequeña cantidad de memoria y una antena, se configuran como sea requerido, éstas son dispositivos *NFC* pasivos, son generalmente de un tamaño pequeño y pueden integrarse en objetos cotidianos [5].

1.1.1.1 Ventajas de la tecnología

Las ventajas que provee la tecnología *NFC* son las siguientes:

- Tiene gran seguridad al transmitir en rangos no mayores a 4cm.
- La comunicación entre dispositivos se realiza de forma rápida sin necesidad de una conexión alámbrica entre ellos.
- Debido a sus modos de operación se utiliza en una amplia gama de aplicaciones.
- Actualmente se encuentran en el mercado una gran cantidad de teléfonos celulares que cuentan con dicha tecnología.
- Los dispositivos *NFC* pasivos al no requerir energía propia para funcionar, pueden estar incluidos en etiquetas, llaveros, estampas, entre otros.
- El usuario final no requiere conocimientos técnicos para realizar la comunicación entre dispositivos.

1.1.1.2 Aplicaciones de *NFC*

En este anteproyecto, a las aplicaciones que hagan uso de la tecnología *NFC* se les llama *aplicaciones NFC.*

Entre las distintas aplicaciones *NFC* se encuentran [6]:

- **Autentificación:** se pueden autentificar personas, ya sea para reemplazar llaves, tarjetas o contraseñas, o productos con *NFC tags*.
- **Interfaces con el ambiente:** es posible obtener información de lugares tales como museos, exposiciones, sitios turísticos, anuncios, entre otros, que puedan encontrarse en el medio.
- **Almacenamiento de *Tokens*:** un *Token* (ficha) representa un boleto virtual y este se puede volver a utilizar con un boleto diferente.
- **Servicio de pagos:** una de las funcionalidades más utilizada al tener la seguridad que brinda esta tecnología es que se pueden realizar pagos o transacciones monetarias en cajeros automáticos o estaciones de pago.

1.1.1.3 Estándares y protocolos de NFC

NFC es una tecnología de plataforma abierta desarrollada por *Philips* y *Sony* descrita *como NFCIP-1 (Near Field Communication Interface and Protocol 1*, protocolo e interfaz de comunicación de campo cercana 1), estandarizado después bajo la normativa ISO (*International Organization for Standardization,* Organización internacional para estandarización) 18092, ECMA (*European Computer Manufacturers Association*, Asociación Europea de manufactura de computadoras)

340 como también en ETSI TS (*European Telecommunications Standards Institute Technical Specifications*, Especificación Técnica del Instituto Europeo de Normas de Telecomunicaciones) 102 190, estos estándares especifican las capacidades básicas como son; velocidad de transferencia, esquemas de codificación de bits, modulación, arquitectura, protocolo de transporte, entre otras [7].

Entre los protocolos actuales de *NFC* se encuentran los siguientes [8]:

Las normativas ISO/IEC (*International Electrotechnical Commission*, Comisión Electrotécnica Internacional) 18092 and ISO/IEC 14443 MAC (*Message Authentication Codes*, Códigos de autentificación de mensajes) son servicios de conexión que ofrecen una configuración mínima y sin garantías de control de flujo.

El LLCP (*Logical Link Control Protocol*, Protocolo de Control de Enlace Lógico) basado en el estándar IEEE (*Institute of Electrical and Electronics Engineers*, Instituto de Ingeniería Eléctrica y Electrónica) 802.2 diseñado para soportar aplicaciones pequeñas con transportes de datos limitados o protocolos de red como son OBEX (*OBject Exchange*, intercambio de objetos) y TCP (*Transmission Control Protocol*, Protocolo de control de transmisión) / IP (Internet Protocol, Protocolo de internet), los cuales proveen servicios de entorno robustos para aplicaciones.

El NFC LLCP provee una sólida función para aplicaciones *peer-to-peer* (punto a punto) enriqueciendo la funcionalidad básica ofrecida por la normativa ISO/IEC 18092.

1.1.1.4 Modos de operación de la tecnología

NFC Fórum [9], organización a favor del uso y desarrollo de NFC describe los modos de operación en la Figura 1.1

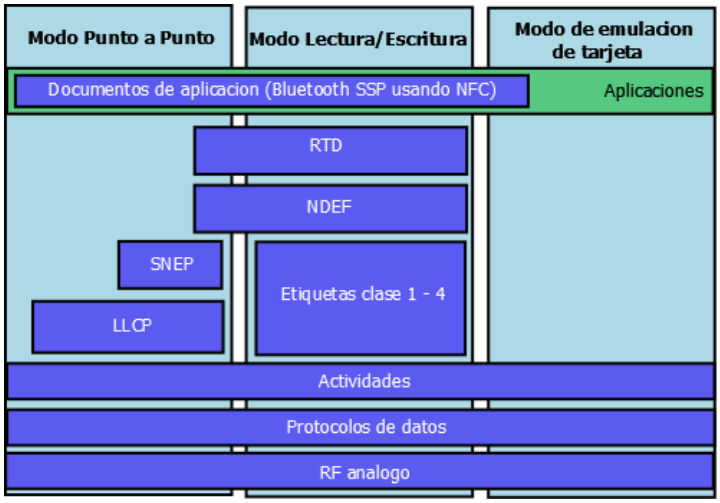

Figura 1.1 Modos de operación NFC Traducido de [9]

- **Modo punto a punto:** permite a dos dispositivos NFC crear una conexión de dispositivo a dispositivo para intercambiar cualquier tipo de datos.
- **Modo Lectura/Escritura:** permite a los dispositivos leer y escribir datos en *NFC tags* pasivas.
- **Modo emulación de tarjeta:** permite a los dispositivos NFC actuar como una tarjeta inteligente.

Dichos modos son una característica esencial al estudiar las aplicaciones *NFC*, estas proveen diferentes beneficios a usuarios y sugieren diferentes formas de uso [10].

1.1.1.5 Dispositivos

Entre los principales dispositivos que ocupan la tecnología NFC se encuentran:

- Teléfonos móviles
- Lectores NFC
- *Tags* NFC

Además de los ya antes listados se encuentran cajeros automáticos, computadoras, televisores, audífonos, entre otros.

1.1.2 Android

Android provee un *Framework* (marco de trabajo) enriquecido y adaptable que permite construir aplicaciones innovadoras y juegos para dispositivos móviles en un ambiente Java, permite también la adaptabilidad de los recursos a diferentes configuraciones de dispositivo [11].

Las aplicaciones Android se escriben los lenguajes Kotlin, Java o C++. La herramienta Android SDK (*Software Developer Kit*, Kit de desarrollo de software) compila el código en conjunto con los recursos en un archivo APK (*Android Application Package*, Aplicación empaquetada de Android), un paquete Android, que es un archivo con un sufijo. apk, un archivo APK contiene todo lo necesario para ejecutarse en dispositivos que ocupen el sistema operativo Android.

- Cada aplicación Android se ejecuta en su propia caja de arena, protegida por las funciones de seguridad Android.
- El sistema operativo Android es un sistema Linux multiusuario donde cada aplicación es un usuario diferente.

- Por defecto el sistema asigna a cada aplicación un identificador Linux, el sistema otorga permisos por todos los archivos en la aplicación y solo el usuario con identificador asignado tiene acceso a ellos.
- Cada proceso es su propia máquina virtual, de esa manera cada aplicación se ejecuta de forma independiente de las demás.
- Por defecto cada aplicación tiene su propio proceso Linux, el sistema inicia el proceso cuando cualquiera de los componentes de la aplicación necesite ejecutarse.

1.1.3 Generador de aplicaciones

Un generador de aplicaciones Android es una herramienta que permite facilitar el proceso de creación de aplicaciones. Todas las aplicaciones generadas por dichas herramientas siguen una misma estructura, por lo que se cuenta con código escrito en lenguaje Java que contiene la estructura base de las aplicaciones y mecanismos necesarios para la creación de estas, creando un archivo *XML (Extensible Markup Language,* lenguaje de marcado extensible*)* como recurso para proveer la función esperada por el usuario [12].

Cabe mencionar que no se encontraron reportes de generadores de aplicaciones Android que utilicen la tecnología *NFC* en la literatura revisada.

1.4 Lenguajes para desarrollo web

Los lenguajes para desarrollo web pueden ser interpretados o ejecutados mandando peticiones de un cliente a un servidor alojado en la web

1.4.1 JSF (JavaServer Faces)

JavaServer Faces es un *Framework* que establece el estándar para construir interfaces de usuario del lado del servidor. Con las contribuciones del grupo de expertos, las API JavaServer Faces son diseñadas para que puedan se aprovechen por herramientas que harán que el desarrollo de aplicaciones web sea aún más fácil. Proporciona un motor de renderizado personalizado y una biblioteca de etiquetas personalizadas JSP para renderizar en un cliente HTML, permitiendo a los desarrolladores de aplicaciones Java Platform, Enterprise Edition (Java EE), utilizar la tecnología JavaServer Faces en sus aplicaciones [13].

1.5 Metodología de desarrollo web

Una metodología se encarga de proveer estrategias, tareas y artefactos de desarrollo de software necesarias para la resolución de un problema [14].

Existen diversos tipos de metodologías entre las cuales se encuentran las metodologías web, a continuación, se menciona una de las principales metodologías diseñadas para el desarrollo de aplicaciones web.

1.5.1 Unified Modeling Language-Based Web Engineering

UWE (Unified Modeling Language-Based Web Engineering, ingeniería basada en el modelo de lenguaje unificado) es similar a otros métodos de ingeniería web, el proceso es separado del modelo que describe un sistema web, los modelos son construidos con diferentes fases de requerimientos, y la implementación del proceso se usa para la representación. UWE se enfoca en una estricta separación de las fases de desarrollo y la implementación del modelo [15].

UWE es un proceso iterativo e incremental que incluye flujos de trabajo, las fases de desarrollo coinciden con las propuestas por el Proceso Unificado, plantea técnicas apropiadas para la captura de los requisitos de un sistema Web. Entre los modelos que utiliza se encuentran: modelo de requerimientos, modelo de contenido, modelo de navegación y modelo de procesos.

1.6 Planteamiento del problema

La problemática con las tecnologías emergentes, en especial NFC, es que caen en desuso, son sustituidas por otras tecnologías o simplemente desaparecen porque nunca llegaron a ser utilizadas cómo prometían, puesto que para usarlas se necesita forzosamente de la interacción de programadores que plasmen las ideas de los usuarios en aplicaciones funcionales.

La eliminación de la figura del programador o bien la posibilidad de que los propios usuarios diseñen y construyan sus aplicaciones es una tendencia que cada vez toma más auge; un sistema generador de aplicaciones de *NFC* permitirá a los usuarios a través de una aplicación web recurrir a la tecnología con las particularidades que cada consumidor necesite construyendo aplicaciones Android limitadas pero funcionales que hagan uso de la tecnología *NFC*.

1.7 Objetivos generales y específicos

En esta sección se enumeran los objetivos generales y específicos manejados en el anteproyecto.

1.8.1 Objetivo general

Desarrollar un sistema generador de aplicaciones móviles con identificación por radio frecuencia NFC en Android.

1.8.2 Objetivos específicos

- Analizar el estado del arte de los sistemas que permiten la construcción de Aplicaciones de identificación por NFC.
- Analizar las características de NFC para identificar los elementos a implementar en el generador.
- Describir las condiciones que debe tener un Generador de Aplicaciones de NFC que facilite el desarrollo de aplicaciones de identificación por radio frecuencia.
- Desarrollar el sistema propuesto para la solución.
- Realizar los casos de estudio (inventarios, lista de cotejo, identificación de objetos).

1.9 Justificación

NFC es una tecnología la cual puede tener una amplia aplicación en áreas que no se ha ocupado antes, a su vez la implementación de dicha tecnología puede tener beneficios en dichas áreas, como ya se había mencionado anteriormente dado que no se encontró en la literatura revisada que en el mercado exista un generador de aplicaciones *NFC,* se propone que las tecnologías referentes a la generación de aplicaciones y las aplicaciones *NFC* serán analizadas para la creación de dicho producto.

El generador de aplicaciones de *NFC* establecerá una gama de usos generales y definirá una serie de acciones específicas que el usuario quiera personalizar permitiendo entonces la construcción de aplicaciones *NFC* específicas para usos particulares, las cuales pueden ser modificadas y actualizadas por los propios usuarios sin la necesidad de la intervención de un programador, las necesidades que se satisficieran serán para aquellos usuarios interesados en el desarrollo de aplicación con tecnología *NFC* que carezcan de conocimientos de programación.

El generador tendrá como beneficio la adopción de la tecnología *NFC* y que dicha tecnología no caiga en desuso. El Instituto Tecnológico de Orizaba será el beneficiario de la creación del generador, así como de su uso y distribución.

Capítulo 2. Estado de la práctica

En este capítulo se presentan algunos trabajos relacionados directa o indirectamente con el proyecto de tesis, con el objetivo de recopilar información relevante para el desarrollo de este.

2.1. Trabajos relacionados

En [12] se estableció que los sistemas de tránsito masivo han existido por más de un siglo. En la estructura de pago tradicional, los boletos, las tarjetas y los *tokens* se utilizan para el transporte público. Los descuentos para el tipo específico de pasajeros y los beneficios que vienen con la tarifa de transferencia solo están disponibles con tarjetas diseñadas específicamente para cada ciudad.

Alan y Birant [16] propusieron un sistema de cálculo de tarifas basadas en un servidor sin tener que obtener una tarjeta por cada ciudad, la creación del sistema propuesto se aplicó en cinco ciudades de Turquía con el estándar *NFC* (*Near Field Communication*, Comunicación de campo cercano) con el que recopilaron los datos para crear las reglas de asociación para el sistema propuesto utilizando el algoritmo *FP-Growth* como una técnica de minería de datos.

Con los juegos que ahora integran objetos físicos a través de *NFC*, han surgido nuevas técnicas de trampas, que incluyen el aumento característico de los objetos, la duplicación de objetos y la introducción de nuevos objetos no autorizados. En [17] abordó este problema para juegos basados en objetos *NFC* activos.

Fortat et al. [17] realizaron un análisis original presentando un modelo con objetos *NFC* activos que se compone de un servidor, un dispositivo (computadora, consola o teléfono inteligente) equipado con un lector *NFC* y objetos *NFC*. También diseñaron cuatro requisitos de seguridad rentables a abordar utilizando ataques realistas que explotan las vulnerabilidades del sistema.

La tecnología NFC se ha probado con dispositivos médicos *implantables* tales como marcapasos, Implantes cocleares e implantes subcutáneos entre otros, que cobraron importancia combatiendo desórdenes médicos, suministrando químicos, o asistiendo a tratamientos; En [18], se enfatiza que la transferencia eficiente de

energía inalámbrica a través del tejido permite eliminar los componentes de energía voluminosos, la mayoría de estos están basados en bobinas unidas por campos magnéticos que varían lentamente, de los cuales algunos están expuestos o fuera del cuerpo human

Ho et al. [18] realizaron un análisis de la potencia encontrada en los campos de transferencia cercanos y los modelos analíticos que mostraron niveles de eficiencia significativos en el centro del campo electromagnético.

Los resultados obtenidos sugieren que la eficiencia dependió de que la fuente y el receptor estuvieran débilmente acoplados. Para alcanzar las eficiencias obtenidas en el análisis, la distribución de corriente de fuente óptima que es sintetizada por una antena física superando con creces las eficiencias obtenidas por diseños convencionales.

En [19] se estableció que los sistemas *AFC* (*Automated Fare Collection*, recolección de tarifas automáticas) se implementaron a nivel global durante décadas, especialmente en el transporte público, la transferencia de los mensajes de transacción de estos son principalmente en texto sin formato, lo cual es obviamente inseguro, en los últimos años con la creciente producción de teléfonos inteligentes equipados con *NFC* los sistemas *AFC* se encuentran en una situación peligrosa.

NFC es un componente típico de la mayoría de los teléfonos inteligentes e implementa el mismo estándar de comunicación que los sistemas *AFC.* Además, puede funcionar en un modo especial de *HCE* (*Host-based Card Emulation*, Emulación de tarjeta basada en host) que permite a cualquier aplicación de Android emular una tarjeta *AFC* y comunicarse directamente con una terminal *AFC.* Motivados por eso, Dang et al. [19] diseñaron un nuevo paradigma de ataque en los sistemas de fijación de precios *AFC* basados en la distancia, lo que permite a los usuarios pagar mucho menos de lo que realmente se requiere.

El ataque construido tiene dos propiedades importantes:

1) Es invisible para los operadores del sistema *AFC* porque el ataque nunca causa ninguna inconsistencia en la base de datos de los operadores.

2) Es escalable para un gran número de usuarios manteniendo un grupo de tarjetas de *AFC* de tamaño moderado.

Con base en este ataque construido, se desarrolló una aplicación *HCE*, llamada *LessPay.*

Los avances en la tecnología han dado paso a la digitalización de actividades cotidianas. En la actualidad muchos individuos tienen un dispositivo móvil

inteligente, es parte de la tecnología ubicua, el desarrollo de estas tecnologías desencadenó la disminución en el uso de tarjetas de presentación tradicionales, siendo estas unas de las últimas herramientas analógicas para compartir información formalmente; como una alternativa a estas, se guardan los contactos de manera digital en los teléfonos, careciendo de la capacidad de almacenar datos complejos y siendo vulnerables a la pérdida de información.

Tatiraju et al. [3] presentaron una alternativa de solución utilizando tarjetas de presentación compatibles con NFC donde los contactos comerciales son trasferidos al teléfono inteligente del usuario, además de una aplicación de Android que permite al usuario navegar por todos los contactos y almacenar o recuperar estos de la nube. El modelo propuesto es una mejora sobre el existente sistema de almacenamiento de tarjetas de presentación usando NFC, ya que implementa estrategias de seguridad al utilizar cifrado, por lo que se evita el espionaje por parte de terceros.

La popularidad de las *Digital wallets* (carteras digitales) y *contactless payment* (pagos sin contacto) se incrementó en los últimos años, de todas las opciones que hay disponibles para realizar las funciones con estas tecnologías los *Mobile payment* (pagos con dispositivos móviles) utilizando *NFC* son los que tienen mayor auge en la industria, gracias a esto la industria de teléfonos inteligentes produce más dispositivos con estas funcionalidades y se integran sensores *NFC* en cada vez más dispositivos *ATM* (*Asynchronous Transfer Mode*, modo de transferencia asíncrona).

Meschtscherjakov et al. [4] llevaron a cabo una investigación en cooperación con SBS Software para obtener conocimientos sobre *NFC* y encontrar posibles casos de uso para *NFC* en *ATM* (*Asynchronous Transfer Mode*, Modo de transferencia asíncrona), utilizando el protocolo *XFS* (*extensions for financial services*, extensión para servicios financieros) y el *Inspirational Bits method* (método inspirador de bits), describiendo un conjunto de pequeñas aplicaciones que exponen diferentes propiedades de *NFC (* denominadas *NFC-Bits*).

Los *NFC-Bits* revelan una amplia gama de características *NFC* que fueron aplicadas para los *Mobile payment*, su uso en *ATM* y *MyPocketATM*, una aplicación derivada de esta.

En los últimos años el progreso de los dispositivos biomédicos *implantables* ha ido en aumento, sin embargo, estos dispositivos son únicamente utilizados en

investigaciones debido en muchos casos a las limitaciones de tamaño, de fuentes de alimentación y falta de interfaces entre estos y dispositivos que puedan monitorearlos.

La alimentación y comunicación inalámbrica son claves en la miniaturización de los componentes, así como en la reducción de la complejidad y el consumo energético, pero sin sacrificar la solidez y las funcionalidades. A pesar de que estos no son conceptos nuevos, los mecanismos existentes no conducen a sistemas miniaturizados.

En [20] se presentó un análisis y técnicas para la trasferencia inalámbrica de energía y transferencia eficiente de datos teórica y prácticamente. Utilizando una arquitectura de dispositivo *implantable* activa se logra la optimización de trasferencia de datos y energía.

Entre las razones principales para desinstalar una aplicación se considera el tiempo de respuesta lenta en los dispositivos, por lo tanto, los desarrolladores deben tener cuidado de no incluir defectos en el código que causan problemas de respuesta. La respuesta deficiente es un defecto importante que afecta la percepción del usuario.

Ongkosit y Takada [21] propusieron una herramienta de análisis de capacidad de respuesta para aplicaciones Android que ayudó a los desarrolladores a descubrir posibles problemas de respuesta en el código fuente. Un experimento preliminar mostró que dicha herramienta descubre eficazmente los posibles defectos de respuesta.

La precisión del análisis puede mejorarse con la información del *perfilaje* (*profiling*) La herramienta proporciona un análisis más profundo, especialmente para la operación de bases de datos.

NFC proporciona la forma más rápida de comunicar dos dispositivos entre sí, al igual que tecnologías como Bluetooth, funciona solo en corto plazo y la trasferencia de datos es a baja velocidad. *NFC* tiene problemas de seguridad como lo son: espionaje, modificación y robo de datos.

Sharma et al. [2] definieron características de *NFC* tales como: tiene dos modos de uso activo y pasivo en ambos, cada etiqueta produce su propio campo electromagnético alternadamente para realizar el cambio de información, es compatible con *RFID (Radio Frequency Identification*, identificador de frecuencia de

radio), entre otras. Además, compararon las ventajas y desventajas de *NFC* con tecnologías similares.

Los usos potenciales de *NFC* deben ser apoyados por bancos, comerciantes y compañías tecnológicas para proveer una plataforma segura a los usuarios de esta tecnología.

Las compras hasta hace poco se relacionaban con dinero en efectivo o tarjetas de crédito, con al auge de los teléfonos móviles nuevas formas de pago han surgido, por ejemplo, los *Mobile payment*, pagos por bienes, servicios y facturas con un dispositivo móvil como teléfono móvil, teléfono inteligente o asistente digital personal, aprovechando las tecnologías inalámbricas y otras tecnologías de comunicación.

Los *Mobile payment* han crecido con la introducción de varios métodos de pago tales como el Protocolo de Aplicación Inalámbrica, Datos de Servicio Suplementario No Estructurado, servicios de mensajes cortos, entre otros. Si bien cada método proporciona ventajas, estos no son ideales al compararse con el método tradicional de pago.

En [22] suministraron implicaciones teóricas útiles para la industria de desarrollo, mercados y manufactura, así como la aceptación de la *NFC MCC (NFC Mobile credit card*, Tarjeta de crédito móvil *NFC*) en el área de los *ATM* y mercados emergentes.

Un tema difícil de abordar para los análisis estáticos, tipo de análisis de software que se realiza sin ejecutar el programa, es la reflexión puesto que estos no la manejan adecuadamente, los marcos de trabajo de Android que hacen uso de ellos tienen gran cantidad de problemas debido a que generalmente eligen crear modelos en lugar de analizar el código dentro el marco de trabajo. Debido a que los modelos cuidadosamente diseñados son importantes se desea que sean reutilizables.

Blackshear et al. [23] presentaron una herramienta para el modelado de aplicaciones Android de propósito general para los análisis estáticos llamada DROIDEL, la cual genera automáticamente apéndices específicos de cada aplicación que obtienen el comportamiento reflexivo de una aplicación en particular.

La sustitución de llamadas reflexivas y la generación de código para simular el comportamiento se realiza a nivel de código de Java, la salida de DROIDEL es una aplicación Java con un solo punto de entrada que puede procesarse por cualquier plataforma de análisis existente.

La experiencia del visitante es la parte más importante de una visita al museo, esta comienza antes que llegar y continúa mucho tiempo después de la visita. Para mantener a los visitantes entretenidos, se experimentan con nuevos canales para lograr mayor participación de los visitantes, la intención principal es que la experiencia física de visitar un museo o un sitio histórico esté relacionada con los medios digitales.

En [24], se presentó el diseño, implementación, uso y evaluación de un *tangible data souvenir* (recuerdo de datos tangible) para la interacción con exposiciones de un museo. Un *souvenir* (recuerdo) se definió como la materialización de cómo se experimenta una visita por cada persona, y se crea dinámicamente con una base de datos grabada durante la visita.

Los *tangible data souvenir* se crean con información de las exposiciones en forma de cartas además de contar con réplicas de piezas con sensores *NFC* que reproducen audios y videos dentro de lectores disponibles en los recorridos. Los *tangible data souvenir* actúan además como la puerta de acceso a más contenido en línea.

La confianza entre los usuarios y los proveedores de servicios en línea depende de una sólida autentificación mutua, dicha autentificación depende de dos componentes en cada ubicación, el verificador y el objetivo a autentificar. Además, también debe tenerse en cuenta que hay personas, organizaciones y sistemas con identidades que deben ser administradas, tanto en el lado del cliente como del servidor.

Debido a que el origen de los datos no puede derivarse únicamente de la autenticación de la entidad, es necesario considerar la autenticación de datos como un servicio de seguridad separado en el contexto de la gestión de identidad. Con estas observaciones se identifica que el problema de la gestión de identidades es bastante complejo.

Jøsang et al. [25] describieron la gestión local de identidad centrada en el usuario, este enfoque fortalece la autentificación, mejora la usabilidad, minimiza los requisitos de confianza y tiene la ventaja de que la interacción en línea confiable puede mantenerse incluso en presencia de infección de *malware* en las plataformas de los clientes utilizando *OffPAD* (*Autentication Personal Offline Device*, dispositivo de autentificación personal fuera de línea).

OffPAD, como un dispositivo confiable, tiene conectividad limitada con las plataformas de los clientes a través de *NFC o USB* (*Universal Serial Bus*, puerto de conexión universal) controlados cuidadosamente, además de estar protegido contra ataques físicos con una contraseña numérica y datos biométricos.

NFC permite una comunicación bidireccional de baja velocidad de datos entre dispositivos en un rango cercano, generalmente unos centímetros, la principal ventaja de NFC es eliminar la necesidad de realizar ajustes de configuración de red al utilizar alternativas como *Bluetooth* o *WiFi.* Esto se debe a su propiedad inherente de asociación por proximidad física; si dos dispositivos se pueden comunicar usando NFC, entonces implica que deben estar ubicados conjuntamente.

En muchos casos, *NFC* se usa para iniciar y configurar automáticamente un canal de comunicación de alta velocidad de datos, como *WiFi* o *Bluetooth.* Nandakumar et al. [26] describieron que la adopción de *NFC* se ve obstaculizada por los bajos niveles de penetración del *hardware NFC* debido a que la mayoría de los teléfonos no están habilitados para *NFC* por lo que las aplicaciones que usan esta tecnología seguirán estando muy limitadas.

Con base en esta problemática se desarrolló Dhwani, un sistema *NFC* basado en la acústica que usa el micrófono y los altavoces en los teléfonos móviles, eliminando así la necesidad de cualquier *hardware NFC* especializado.

Con el crecimiento de los servicios web personalizados los usuarios ya no están obligados a utilizar una estación de trabajo para sus actividades en línea. Es necesaria una solución de credenciales portátil y amigable con la privacidad. Existen dos tipos de credenciales de uso frecuente: contraseñas y soluciones *PKI* (*Public Key Infrastructure*, infraestructura de llave pública) integradas en tarjetas inteligentes, ambas cuentan con desventajas, las contraseñas son débiles y no permiten que el proveedor del servicio obtenga información confiable sobre el usuario; y las tarjetas *PKI* requieren la presencia de hardware adicional, además la instalación del *middleware* (lógica de intercambio de información) puede suponer una carga administrativa que la impide o simplemente no es posible debido a permisos insuficientes en la estación de trabajo. El *middleware* también debe ser generalmente confiable, algo que los usuarios no siempre pueden o no están dispuestos a hacer.

En [27], propusieron una estrategia para la autentificación desde un dispositivo móvil de forma remota a un servidor web para obtener acceso a un recurso en ese mismo servidor utilizando las capacidades de comunicación como *Bluetooth*, *NFC*, entre

otras, este método usó credenciales almacenadas de forma segura en dispositivos móviles, además de ser lo suficientemente flexible para integrar otras tecnologías de credenciales.

2.2. Análisis comparativo

Se realizó un análisis comparativo de los artículos presentados en la sección 2.1, los más relacionados se documentan en la Tabla 2.1

Tabla 2.1 Trabajos relacionados

Articulo	Objetivo	Problema	Tecnologías	Resultados	Estado
[16]	Creación de un sistema de cálculo de tarifas basadas en un servidor utilizando la tecnología NFC.	Los descuentos para el tipo específico de pasajeros y los beneficios que vienen con la tarifa de transferencia solo están disponibles con tarjetas diseñadas específicamente para cada ciudad.	NFC Algoritmo FP-Growth.	Creación de sistema inteligente de transportación pública con sistemas NFC.	Concluido.
[17]	Diseñar requisitos de seguridad rentables para juegos basados en NFC activo.	Con los juegos que ahora integran objetos físicos a través de NFC, surgieron nuevas técnicas de trampas, que incluyen el aumento característico de los objetos, la duplicación de objetos y la introducción de nuevos objetos no autorizados.	NFC.	Cuatro requisitos de seguridad rentables.	Concluido.
[18]	La transferencia eficiente de	La transferencia eficiente de energía inalámbrica a	No se mencionan.	La eficiencia depende de que la	Concluido.

Articulo	Objetivo	Problema	Tecnologías	Resultados	Estado
	energía inalámbrica a través del tejido.	través del tejido es altamente deseable para eliminar los componentes de energía voluminosos.		distribución de la corriente de la fuente óptima sea sintetizada por una antena física.	
[19]	Ataque a sistemas AFC usando dispositivos con NFC y HCE.	NFC funciona en un modo especial de HCE que permite a cualquier aplicación de Android emular una tarjeta AFC y comunicarse directamente con una terminal *AFC.*	NFC, AFC, HCE	Creación de un paradigma de ataque en los sistemas de fijación de precios *AFC* basados en la distancia, aplicación *HCE: LessPay.*	En proceso
[3]	Creación de una alternativa a las tarjetas de presentación tradicionales.	Las tarjetas de presentación tradicionales carecen de la capacidad de almacenar datos complejos y son vulnerables a la pérdida de información.	NFC, Android.	Creación de un sistema de gestión de contactos de negocio NFiD.	Concluido.
[4]	Obtener conocimientos sobre *NFC* y encontrar posibles casos de uso para *NFC* en *ATM.*	Conocimientos insuficientes del mercado de dispositivos con *Mobile payment* y sensores *NFC* en dispositivos *ATM.*	Protocolo *XFS, Inspirational Bits method, ATM, NFC.*	*NFC-Bits a*mplia el conocimiento de características *NFC* que fueron aplicadas para los *Mobile payment*, su uso en *ATM* y *MyPocketATM*.	Concluido

Articulo	Objetivo	Problema	Tecnologías	Resultados	Estado
[20]	Obtener técnicas de transferencia de datos eficientes.	Dispositivos biomédicos *implantables* únicamente utilizados en investigaciones.	*NFC*	Arquitectura de dispositivo *implantable* activa	Concluido.
[21]	Diseñar una herramienta para encontrar errores en código Android.	La capacidad de respuesta es un tipo importante de factor de calidad en la aplicación de Android, ya que afecta directamente a la experiencia del usuario.	Android	Herramienta de análisis de la capacidad del tiempo de respuesta para aplicaciones Android.	En proceso.
[2]	Análisis de tecnología *NFC.*	*NFC* tiene problemas de seguridad como lo son: espionaje, modificación y robo de datos.	*NFC, Bluetooth, RFID.*	*Aplicaciones NFC* deben ser apoyadas por bancos, comerciantes y compañías tecnológicas para proveer una plataforma segura a los usuarios.	En proceso.
[22]	Análisis de *Mobile Credit Card NFC.*	*Mobile payment* tiene deficiencias al compararse con el método tradicional de pago.	*NFC, MCC NFC, ATM.*	Implicaciones teóricas útiles para la industria de desarrollo, mercados y manufactura.	En proceso.
[23]	Modelado de aplicaciones Android de	Los marcos de trabajo de Android que hacen uso de análisis estático tienen	Android	DROIDEL, herramienta que genera	En proceso

Articulo	Objetivo	Problema	Tecnologías	Resultados	Estado
	propósito general para los análisis estáticos.	gran cantidad de problemas.		automáticamente apéndices específicos de cada aplicación que obtienen el comportamiento reflexivo de una aplicación en particular.	
[24]	Crear etiquetas para la interacción con medios digitales.	La experiencia física de visitar un museo o un sitio histórico no está relacionada con los medios digitales.	*NFC.*	Diseño, implementación, uso y evaluación de un *tangible data souvenir* para la interacción con exposiciones de un museo.	En proceso.
[25]	Autentificación local basada en los usuarios.	Autenticación de datos como un servicio de seguridad separado en el contexto de la gestión de identidad.	*OffPAD, NFC, Bluetooh.*	Gestión central de identidad centrada en el usuario, *OffPAD.*	Concluido.
[26]	Crear una alternativa para los dispositivos sin hardware *NFC.*	Los bajos niveles de penetración del *hardware NFC.*	*NFC.*	Dhwani, sistema *NFC* que elimina la necesidad de *hardware NFC* especializado.	Concluido.
[27]	Acceso personalizado a	Los usuarios ya no están obligados a utilizar una	*NFC, PKI.*	Estrategia para la autentificación	Concluido.

Articulo	Objetivo	Problema	Tecnologías	Resultados	Estado
	servicios web desde un dispositivo móvil.	estación de trabajo para sus actividades en línea.		desde un dispositivo móvil de forma remota a un servidor web para obtener acceso a un recurso web	

Después de realizar el análisis de los artículos citados se observa que hay una gran variedad de uso para los dispositivos *NFC,* así mismo se determina que son necesarios amplios conocimientos para el desarrollo de dichas aplicaciones, por lo que un generador de aplicaciones NFC ayudará a que los usuarios construyan aplicaciones estándar con una estructura base, teniendo de esta manera aplicaciones básicas pero funcionales al alcance de la mano.

2.3 solución propuesta

La solución propuesta plantea el uso de la metodología UWE para el desarrollo web, el SGBD MySQL de código abierto y licencia libre en conjunto con el IDE NetBeans y JSF para la parte de desarrollo web y Android Studio para el desarrollo de la aplicación base Android.

2.3.1 Justificación de la solución propuesta

La solución propuesta para el desarrollo de esta tesis está basada en que la metodología UWE es la que mejor se adapta a las necesidades de un sistema basado en web, en la parte del SGBD se opta por MySQL al ser un gestor de código abierto y licencia libre además que este es popular para desarrollo de aplicaciones web , el IDE que se utilizará es NetBeans gracias a la compatibilidad integrada con el desarrollo web utilizando JSF; debido a que el soporte para aplicaciones Android es superior en Android Studio, ya que este es el IDE recomendado en el mismo SDK de Android, se opta por este para realizar el desarrollo de la aplicación base que después se ocupará por el sistema para hacer aplicaciones personalizadas.

Para esto se propone un sistema basado en la web con el cual se podrán construir aplicaciones que hagan uso de la tecnología *NFC* en Android, donde el usuario personalizará la aplicación partiendo de una estructura base, para después incluir las personalizaciones utilizando lenguaje *XML* y por último generar un archivo *APK* que será entregado al usuario.

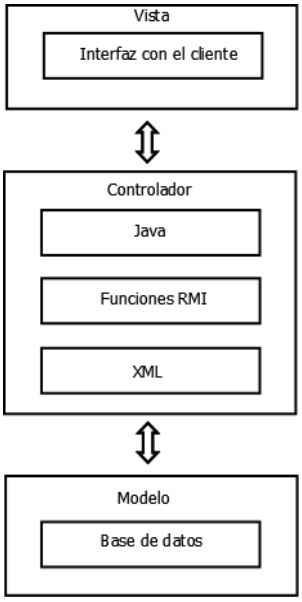

Figura 2.1 Diagrama MVC

El generador de aplicaciones *NFC* seguirá el patrón MVC mostrado en la Figura 2.1, donde las funciones de interacción con el usuario estarán en la vista, el controlador basado en Java tendrá las funciones de *RMI* (Remote Method Invocation, Método de invocación remoto), las cuales traducirán los elementos con los que interactúa la capa de vista traduciéndose a XML para después interpretarse a código java dentro de la consola, además de contar con el acceso a la base de datos para la autentificación del usuario y despliegue de cambios realizados por este anteriormente.

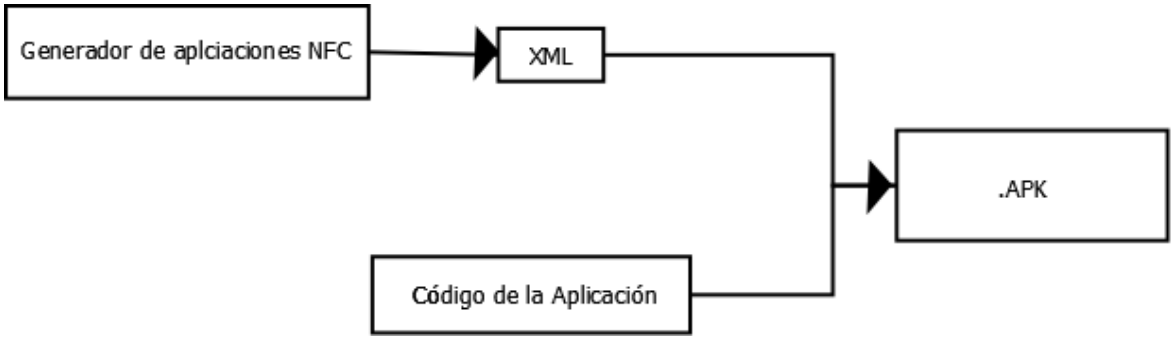

Figura 2.2 Proceso de construcción de la aplicación android

El proceso de construcción de las aplicaciones que seguirá el generador de la misma manera en la Figura 2.2.

Capítulo 3. Aplicación de la metodología

Este capítulo tiene como objetivo presentar el seguimiento de la metodología UWE para resolver el problema ya mencionado en el capítulo 1, dicha metodología propone 5 modelos que serán abordados en esta sección.

3.1 Modelo de requisitos

El objetivo del modelo de requisitos es encontrar los requisitos funcionales de una aplicación, identificación de actores y los procesos en los que se involucren para ser representados en un diagrama de casos de uso.

Los actores que se vean involucrados se dan a conocer en la tabla 3.1

Tabla 3.1 Actores del sistema

Actor	Descripción
Administrador	Representa al usuario que se encarga de gestionar los usuarios.
Usuario	Representa al usuario que utilizara el sistema para generar su aplicación
Solicitante	Representa al usuario que aun no tiene registro dentro de la aplicación

Los actores deberán estar registrados en el sistema para hacer uso de este, de otro modo deberán de solicitar un registro en el sistema.

Los procesos que el sistema realizara son los siguientes;

- Registrar usuario: crea un registro de usuario nuevo.
- Crear aplicación: crea una aplicación nueva utilizando los datos generales descritos en el sistema.
- Autentificación: permite el acceso a la aplicación por medio de un usuario y contraseña.
- Editar aplicación: permite la modificación de los valores de la aplicación.
- Elimina el registro de la aplicación creada por el usuario.
- Generar aplicación: hace uso de la herramienta de generación para la creación de la aplicación .APK

3.1.1 Diagrama de casos de uso

Los actores interactúan con el sistema de la manera mostrada en la figura 3.1

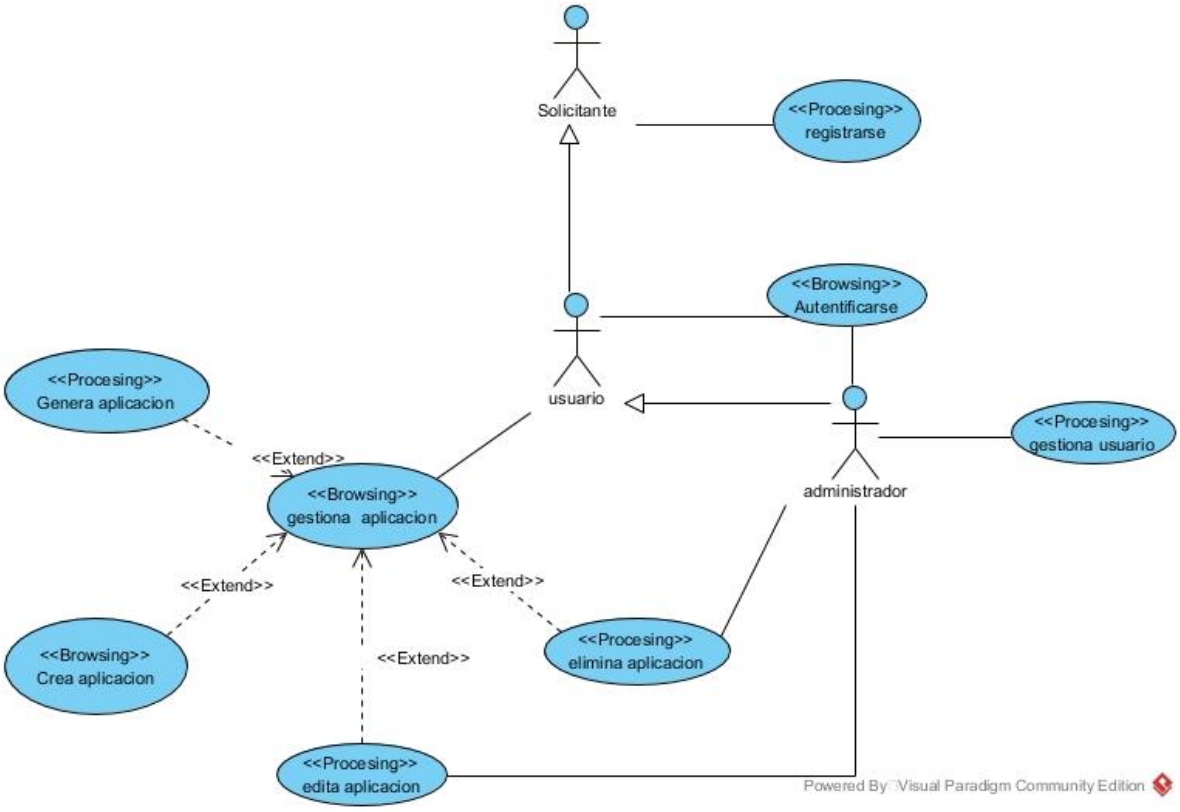

Figura 3.1 Diagrama de casos de uso

Para este caso, se tienen usuario, administrador y solicitante, un usuario puede gestionar aplicaciones, un usuario solicitante puede registrarse, por su parte del administrador puede editar y eliminar aplicaciones y usuarios.

3.1.2 Diagramas de actividades de casos de uso (workflow)

Las actividades realizadas por cada actor se describen en los diagramas de actividades de caso de uso.

El primer diagrama mostrado en la figura 3.2 corresponde a las actividades que se pueden realizar al registrarse en el sistema.

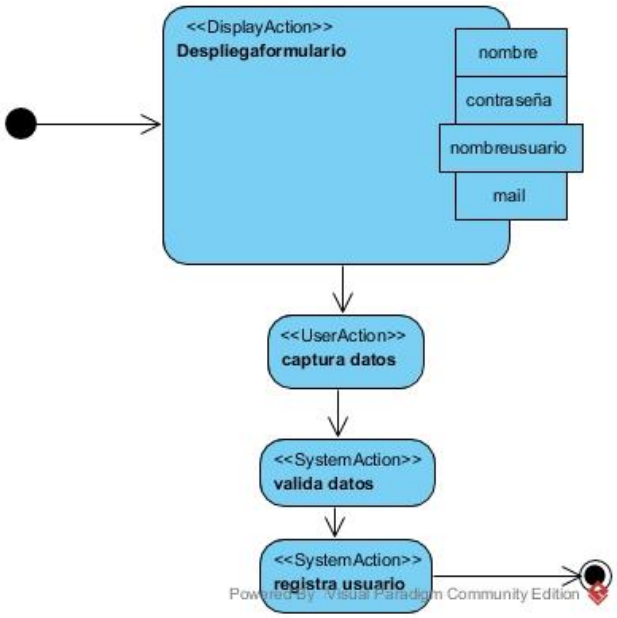

Figura 3.2 Workflow registro de usuario

Para acceder al sistema ya sea como usuario o administrador se debe de seguir el siguiente workflow mostrado en la figura 3.3

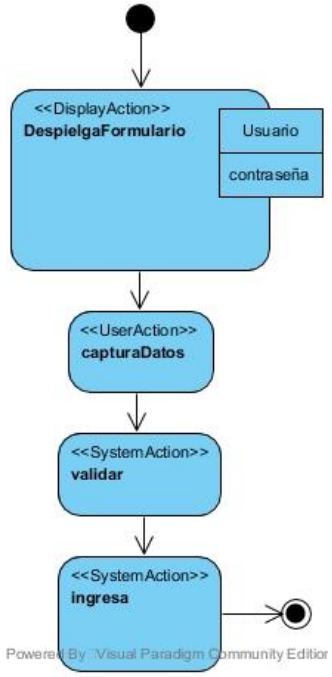

Figura 3.3 workflow acceder al sistema

Las funciones que realice el usuario dentro del sistema se muestran en la figura 3.4

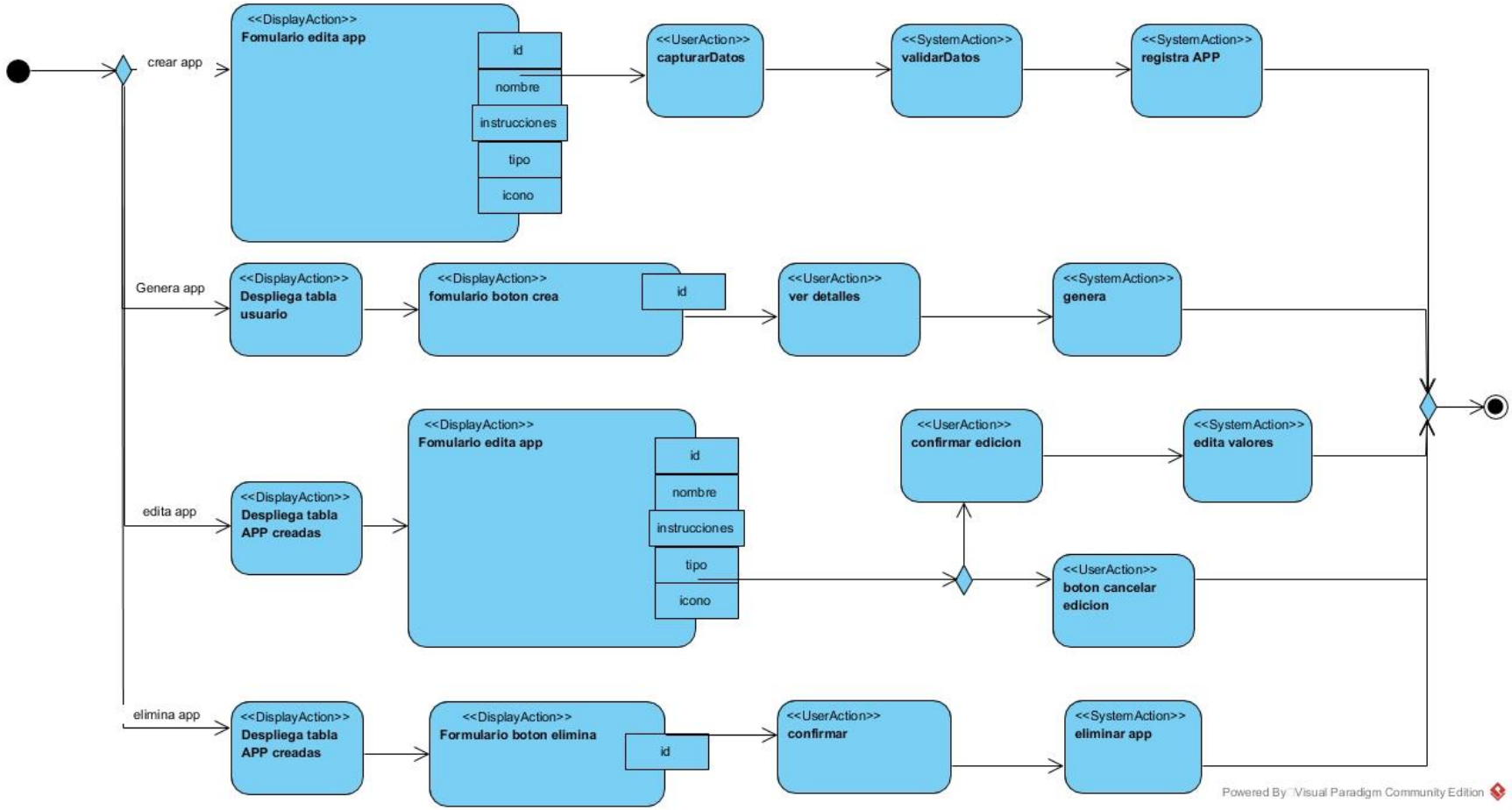

Figura 3.4 workflow acciones usuario

Por ultimo las funciones realizadas por el administrador se muestran en la figura 3.5

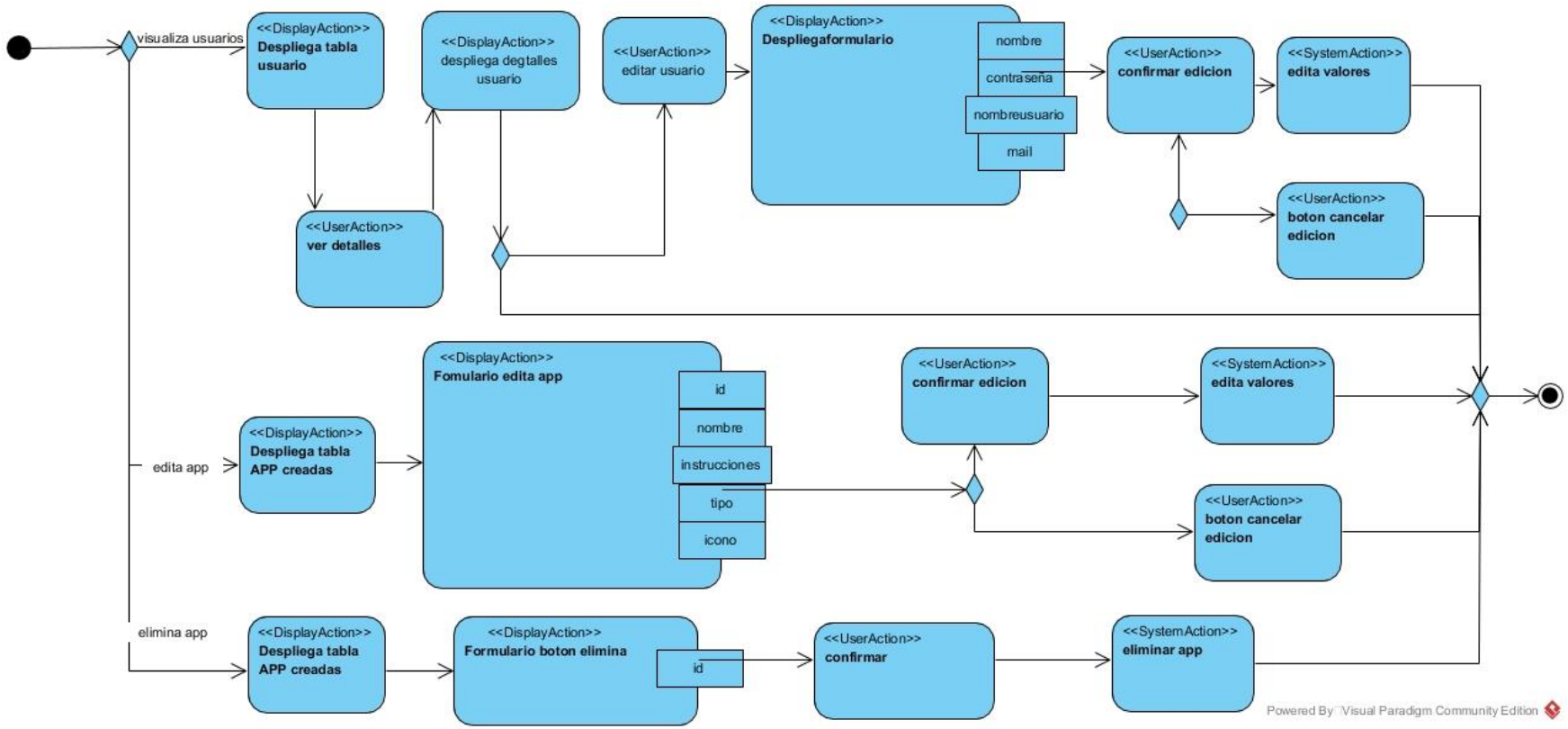

Figura 3.5 workflow acciones de administrador

3.2 Modelo de dominio

El modelo de dominio contiene los diagramas de clase

3.2.1 Diagramas de clase

Un diagrama de clase describe la estructura de un sistema mostrando las **clases** del sistema, sus atributos, métodos, y las relaciones entre los objetos.

A continuación, se muestran las clases principales; generador, usuarios y aplicaciones en la figura 3.5

Figura 3.6 Diagrama de clases

3.3 Modelo de navegación

Los modelos de navegación serán los encargados de ejemplificar la distribución por donde el usuario navegara entre los menús y opciones que tiene el sistema.

3.3.1 Diagramas de clases de navegación

El diagrama correspondiente a la navegación del solicitante se muestra en la figura 3.7, donde la debido a que el sistema necesitara autentificarse para funcionar solo se permite la navegación dentro de la opción de registro.

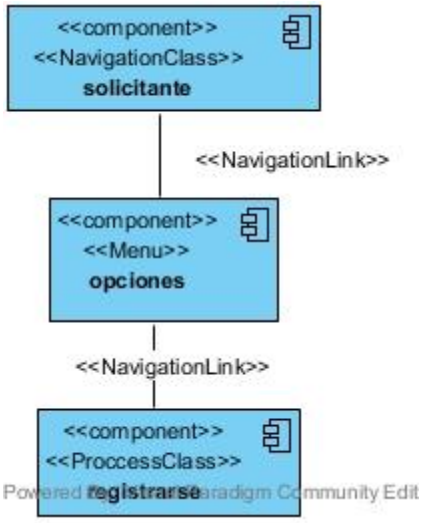

Figura 3.7 Diagrama navegacional de solicitante

La navegación de usuarios es mostrada en la figura 3.8, en esta se muestran las opciones en las que puede navegar el usuario.

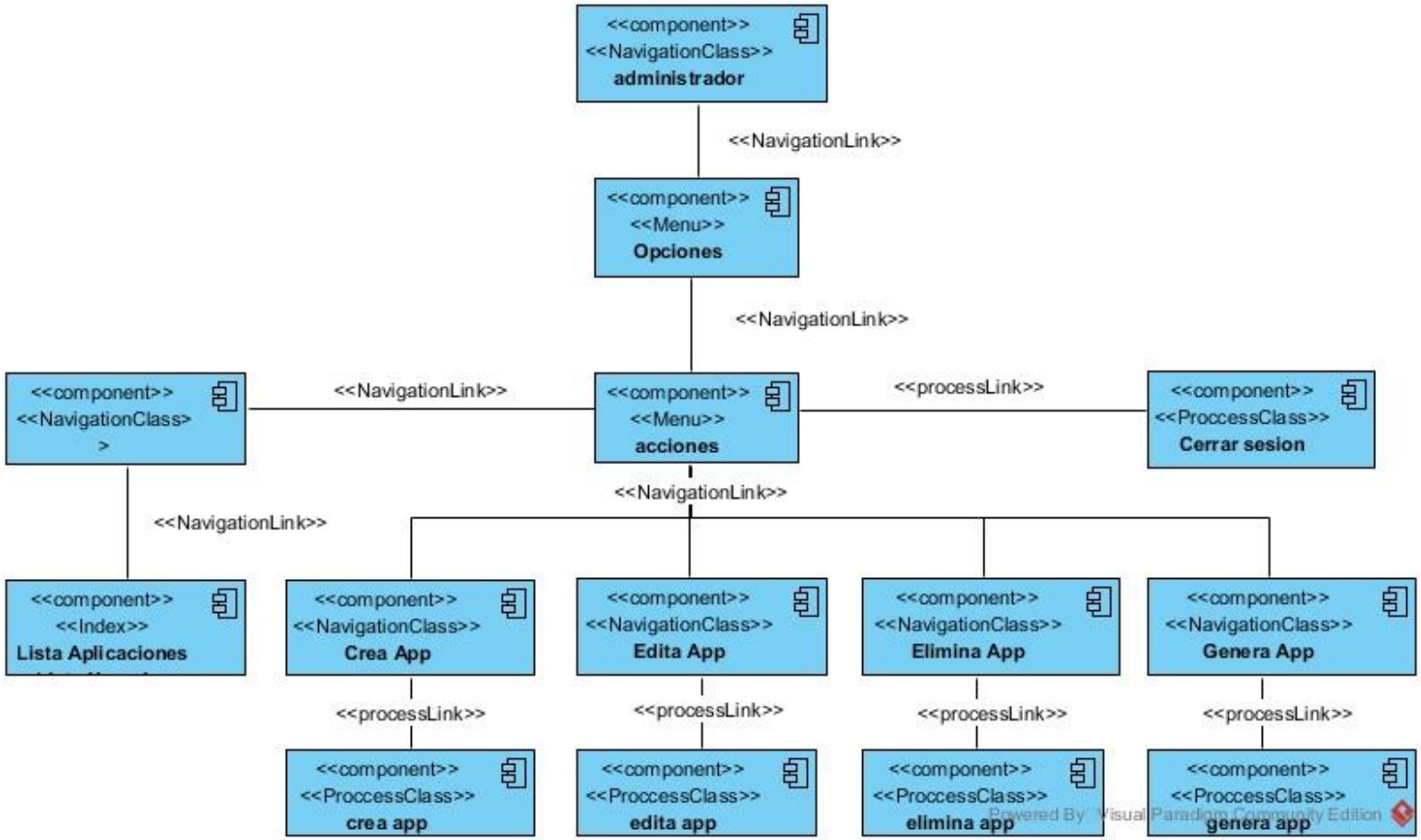

Figura 3.8 Diagrama navegacional de usuario

La navegación para los administradores es mostrada en la figura 3.8

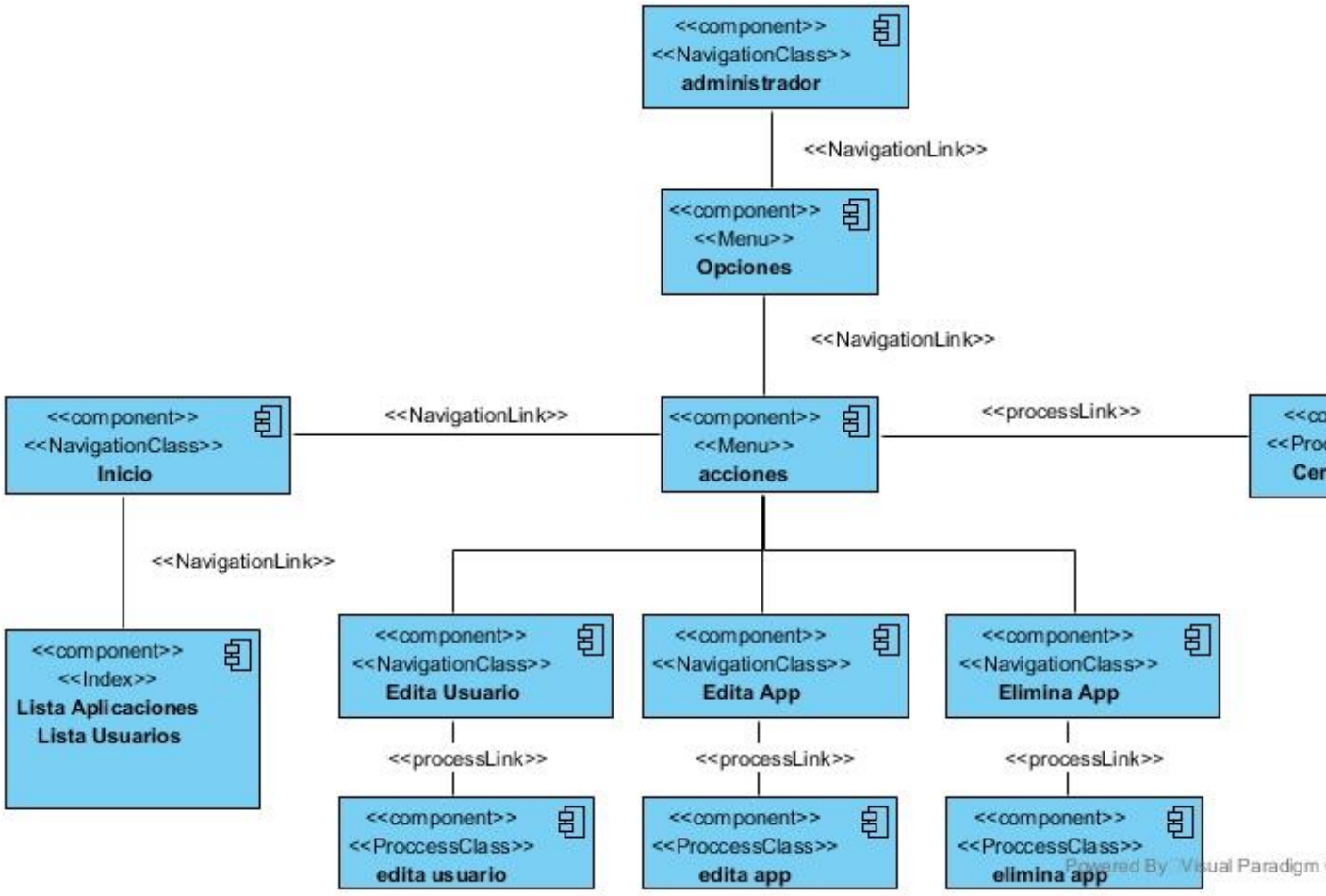

Figura 3.9 Diagrama navegacional de administrado

3.4 Modelo de presentación

En esta sección se muestran los diagramas de clases de presentación los cuales dan una visión más extensa de la distribución final de cada página.

3.4.1 Diagramas de clases de presentación

Cuando se accede al sistema la primera pantalla es el inicio mostrado en la figura 3.10.

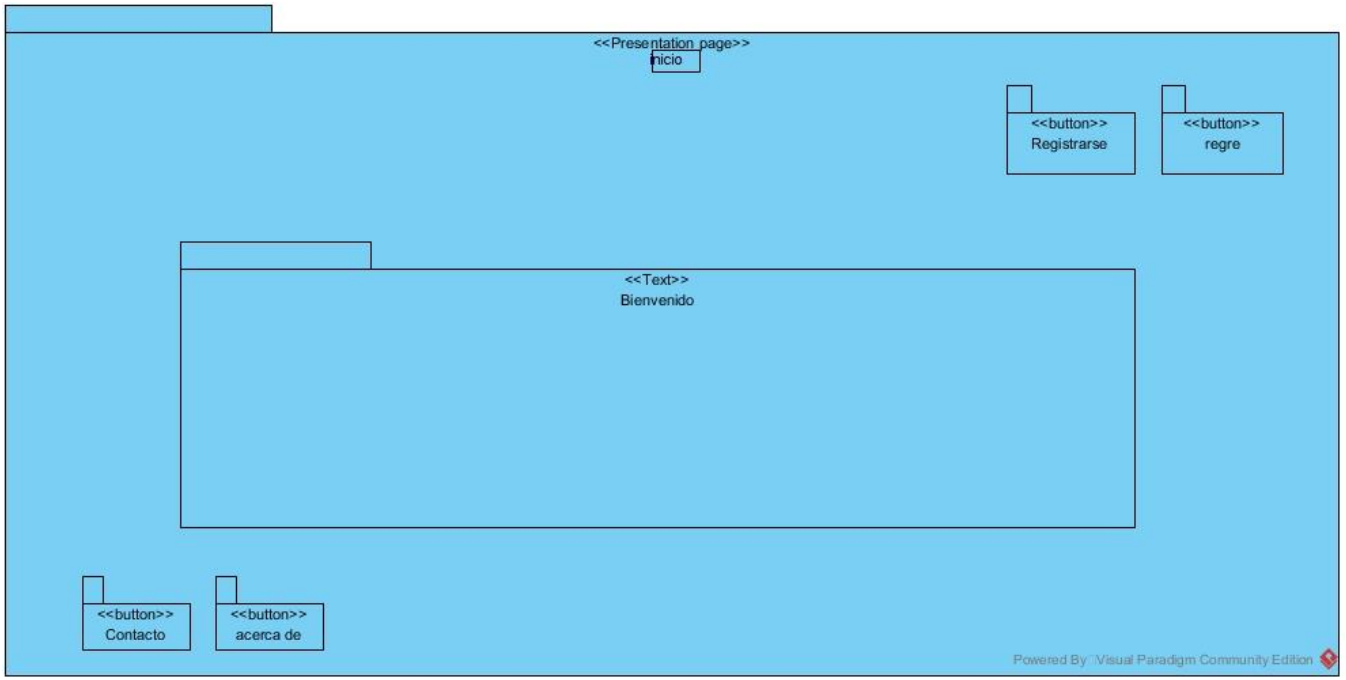

Figura 3.10 Diagrama de presentación inicio

En caso de no estar registrado se accede a la pantalla de registro mostrada en la Figura 3.11, mostrando el formulario de registro correspondiente para esta acción.

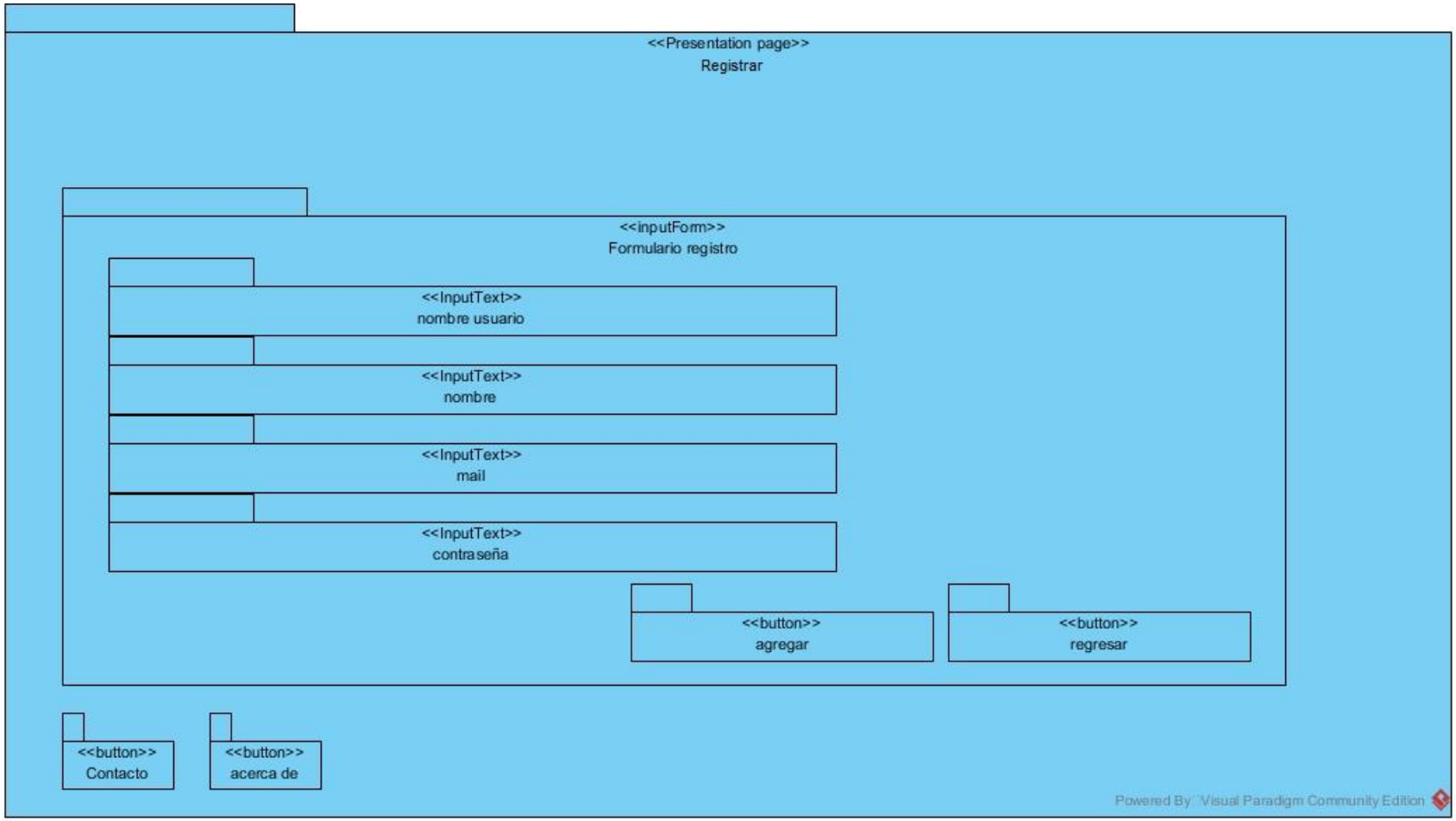

Figura 3.11 Diagrama de presentación registro de usuario

Al ya estar registrado se debe de acceder al sistema por medio de la pantalla mostrada en la figura 3.12

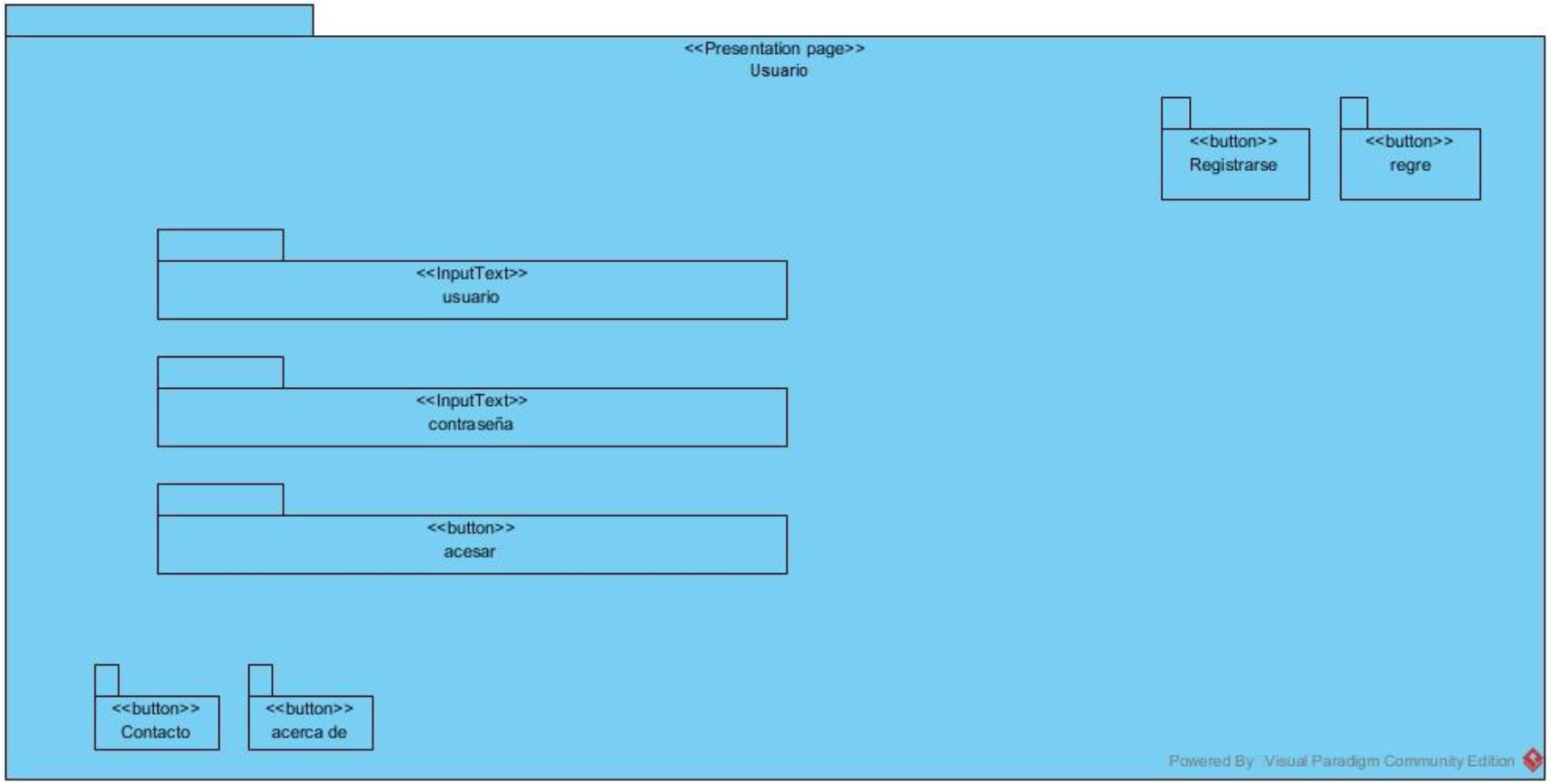

Figura 3.12 Diseño de presentación acceso al sistema

Al entrar en el sistema como usuario se observa la pantalla de la figura 3.13

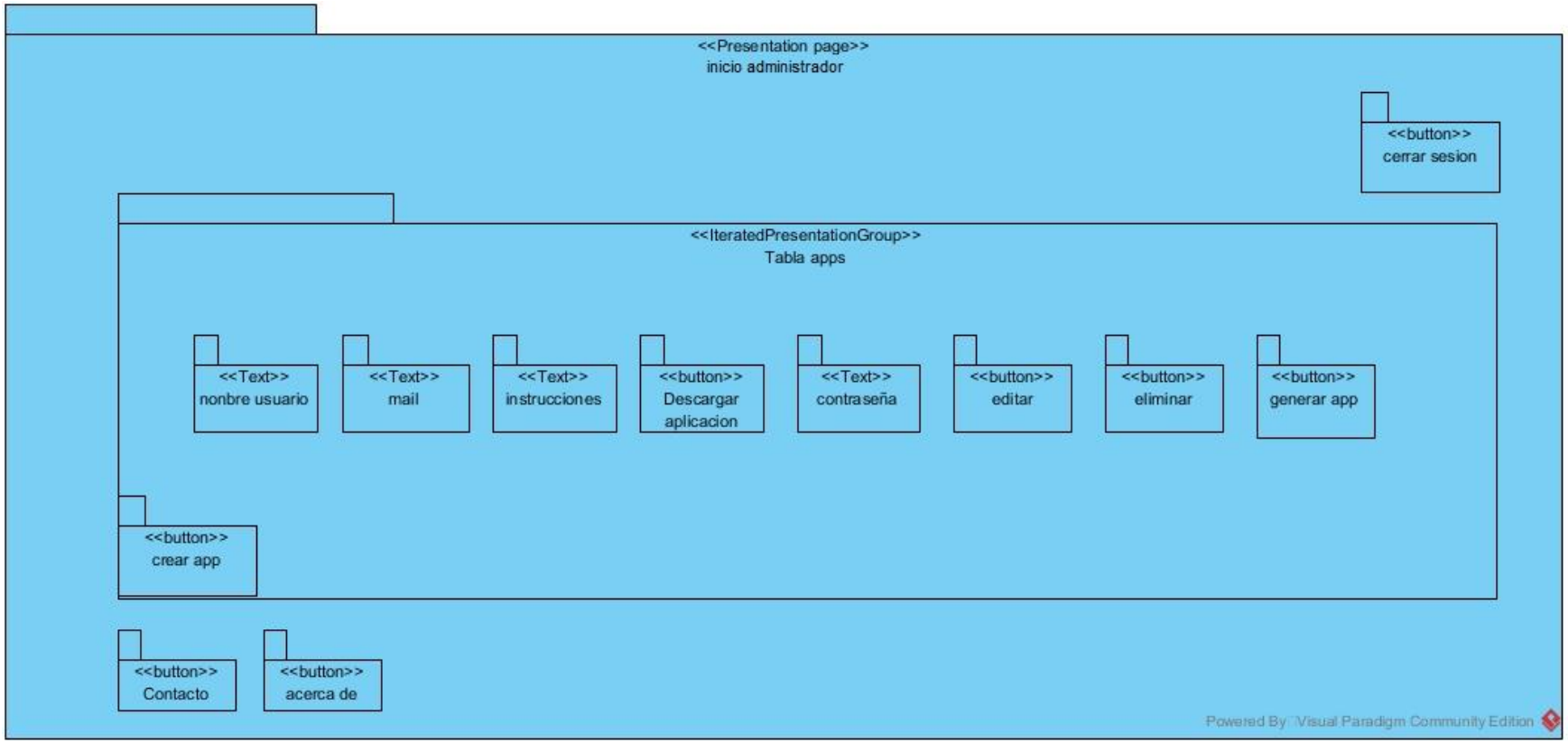

Figura 3.13 Diseño de presentación index usuario

Seleccionando la opción crear app así como editar se despliega el formulario de creación de aplicaciones mostrado en la figura 3.14

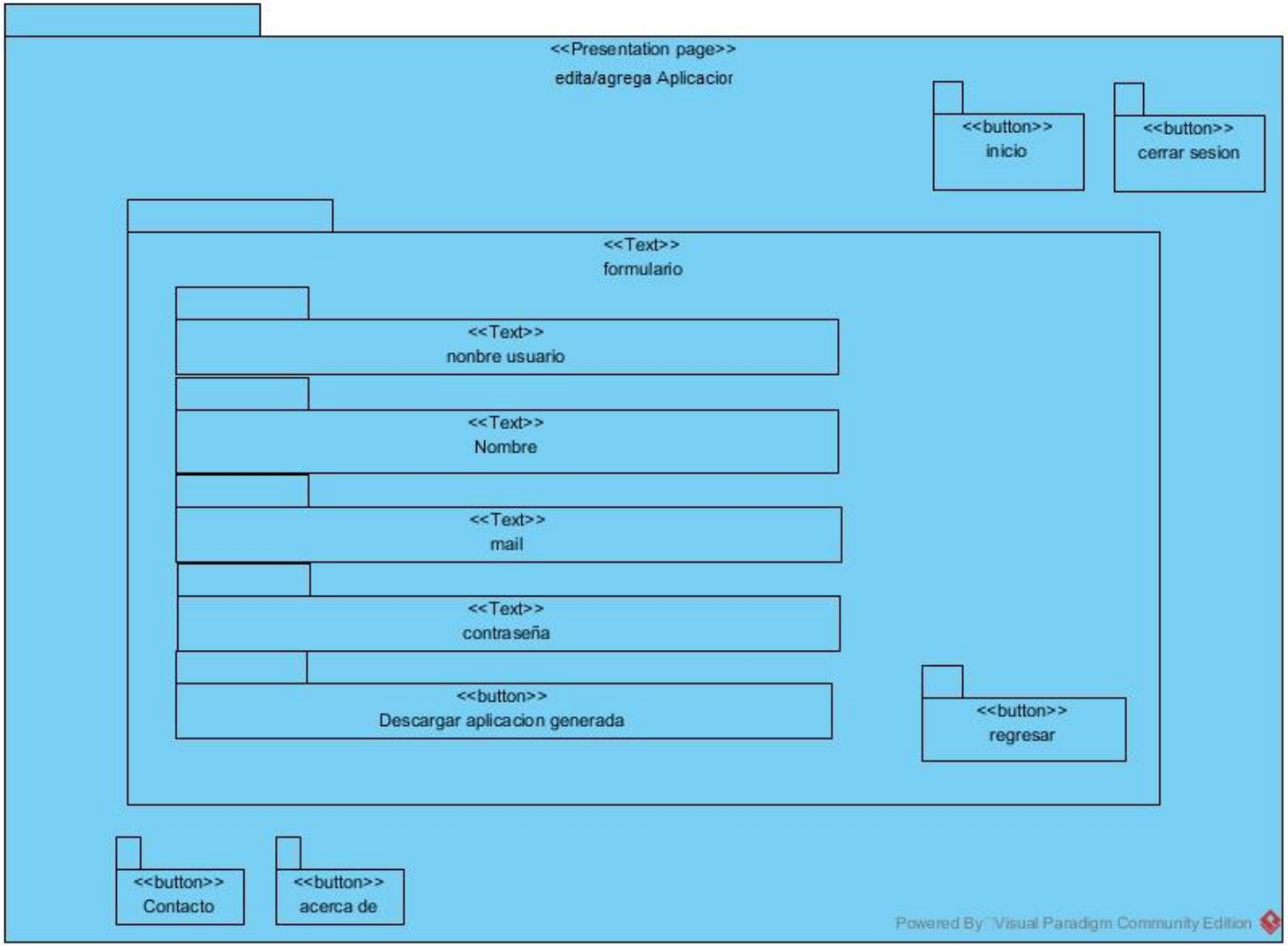

Figura 3.14 Diseño de presentación agregar/editar aplicación

Al seleccionar generar app se despliega botón para la descarga en la pantalla mostrada en la figura 3.15

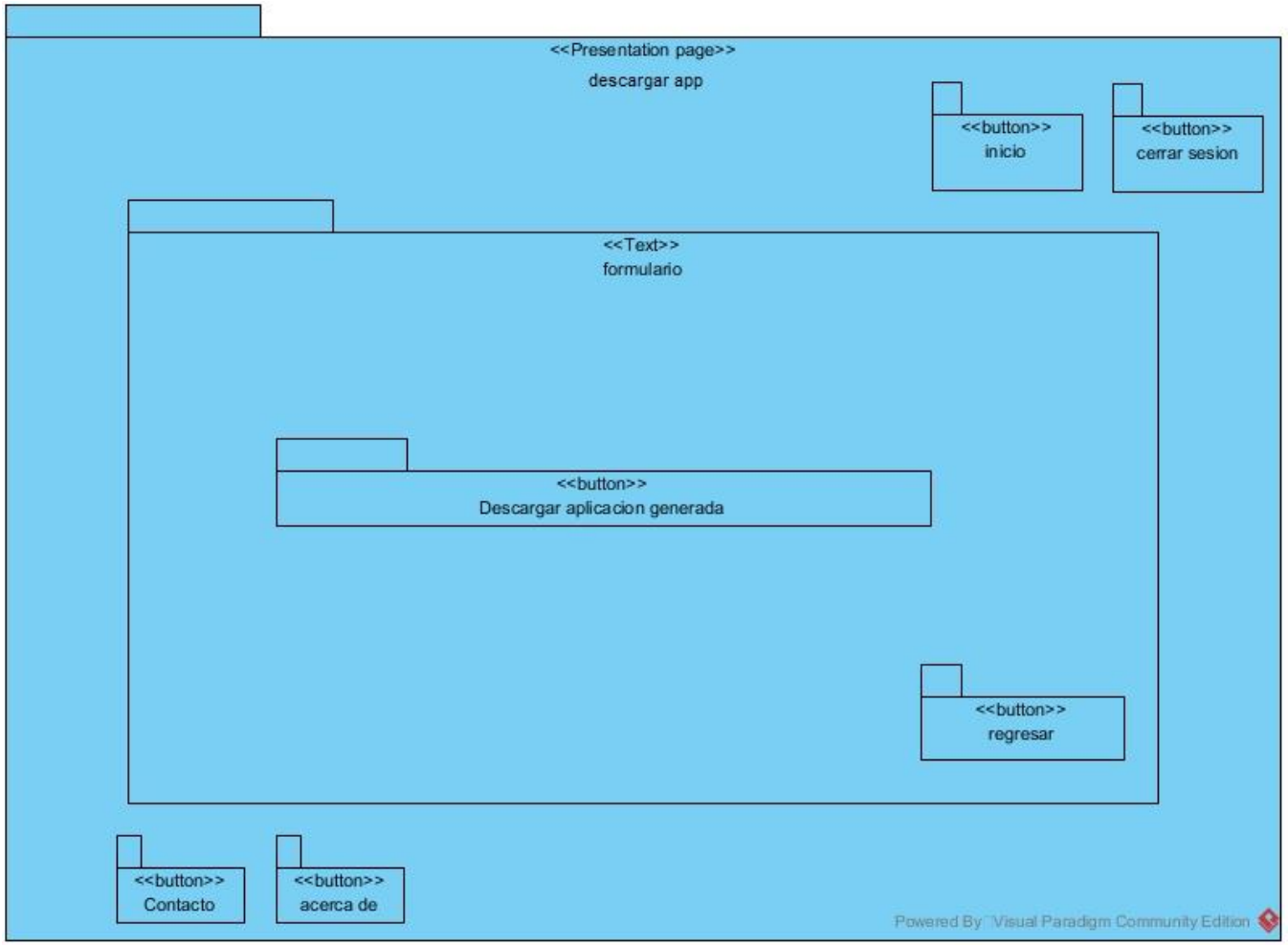

Figura 3.15 Diseño de presentación descargar app

Al ingresar al sistema como administrador se despliega la pantalla mostrada en la figura 3.16

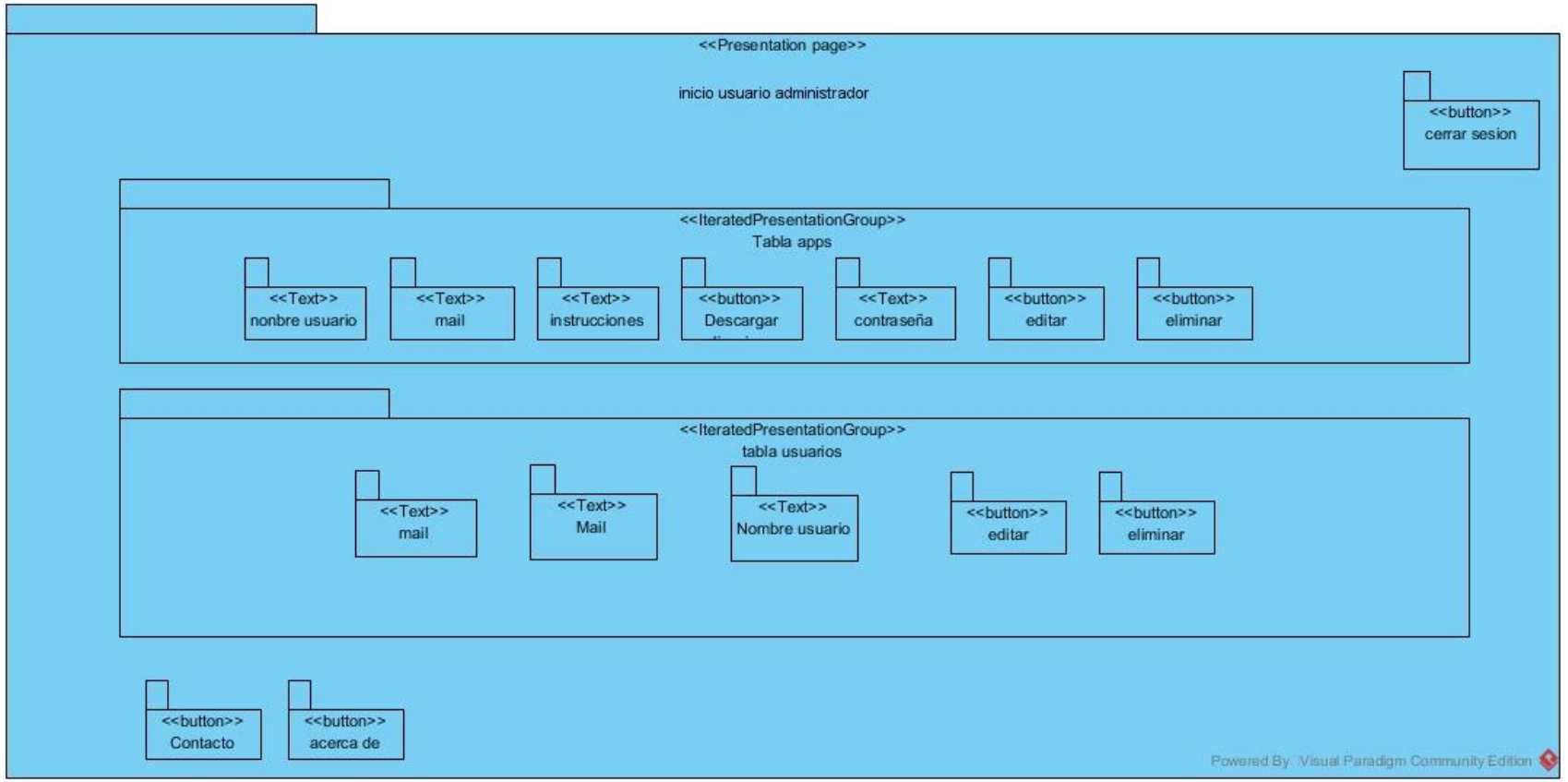

Figura 3.16 Diagrama de presentación inicio administrador

De la misma forma se muestra la pantalla de editar usuarios en la figura 3.17

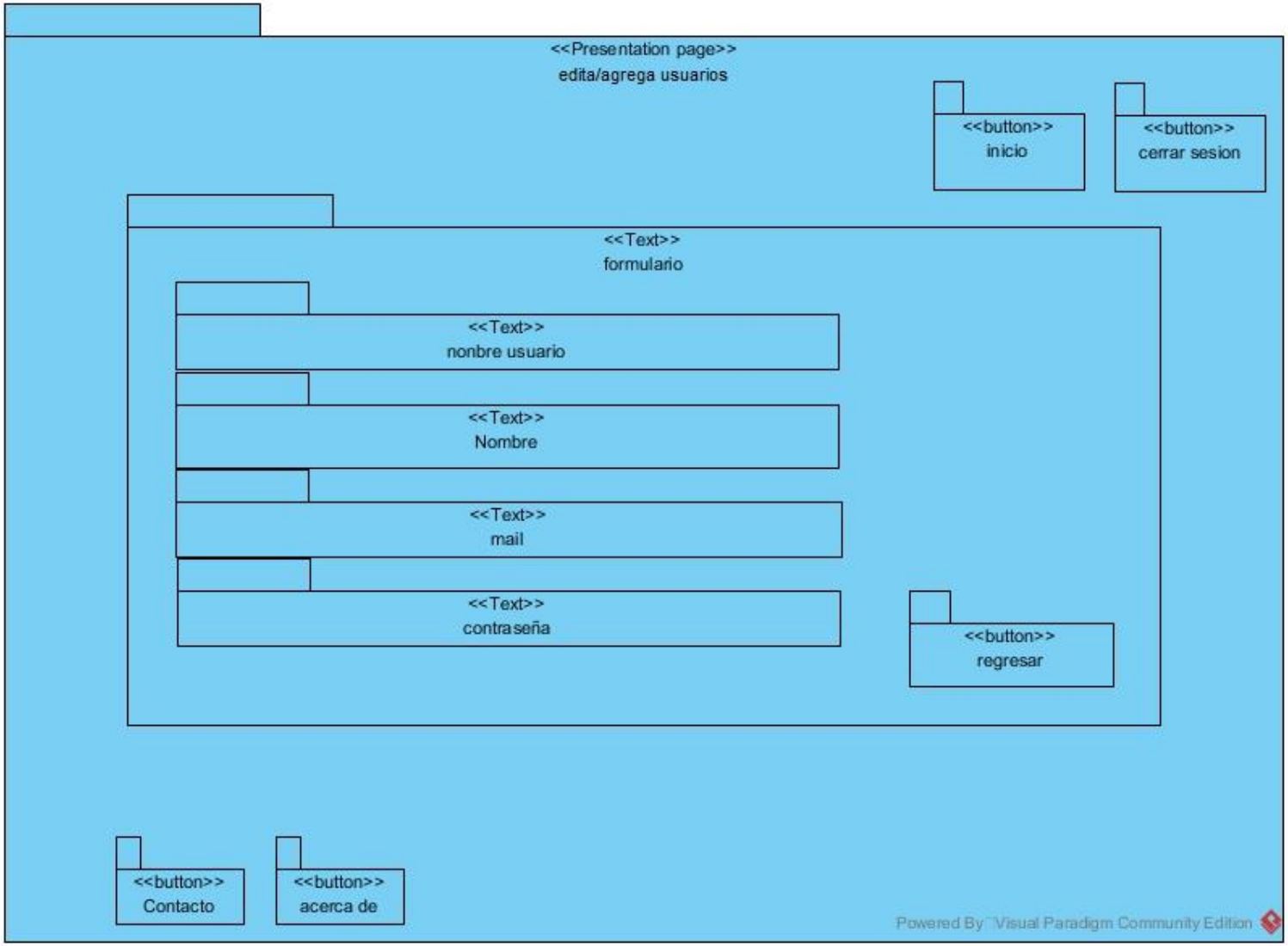

Figura 3.18 Diagrama de presentación edición de usuarios

3.5 Modelo de proceso

Esta es la sección final de los diagramas que maneja la metodología, en esta se muestran los diagramas de procesos basados directamente en los diagramas de workflow.

3.5.1 Diagramas de actividades de proceso

Los diagramas de procesos basados en directamente de los workflows se muestran en las figuras 3.19, 3.20, 3.21 y 3.22

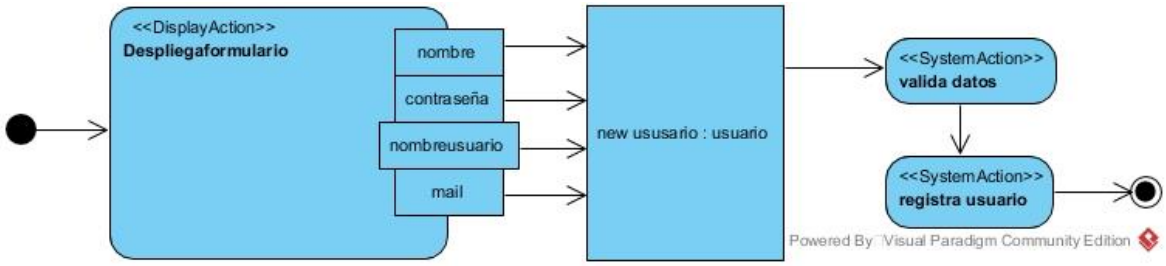

Figura 3.19 Diagrama de proceso registrar usuario

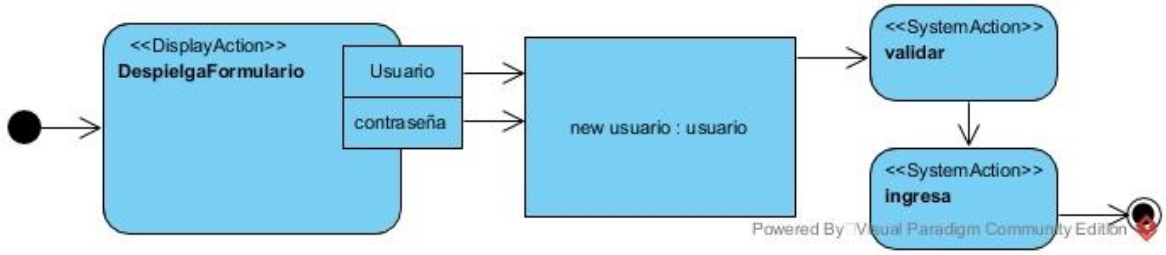

Figura 3.20 Diagrama de proceso acceso al sistema

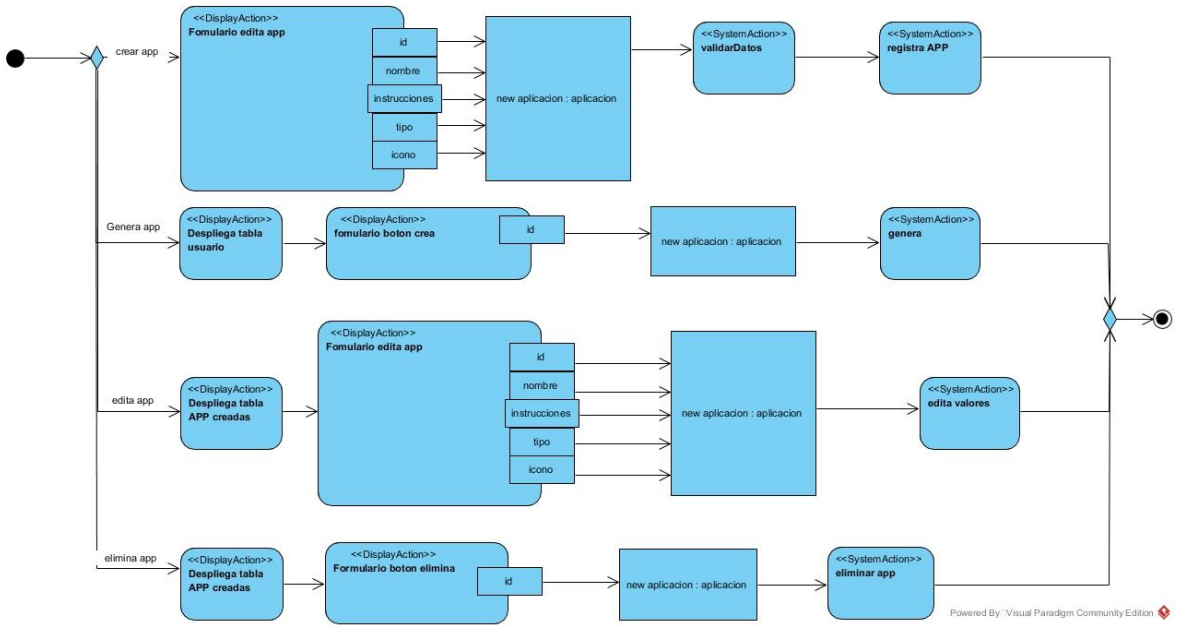

Figura 3.21 Diagrama de proceso funciones de usuario

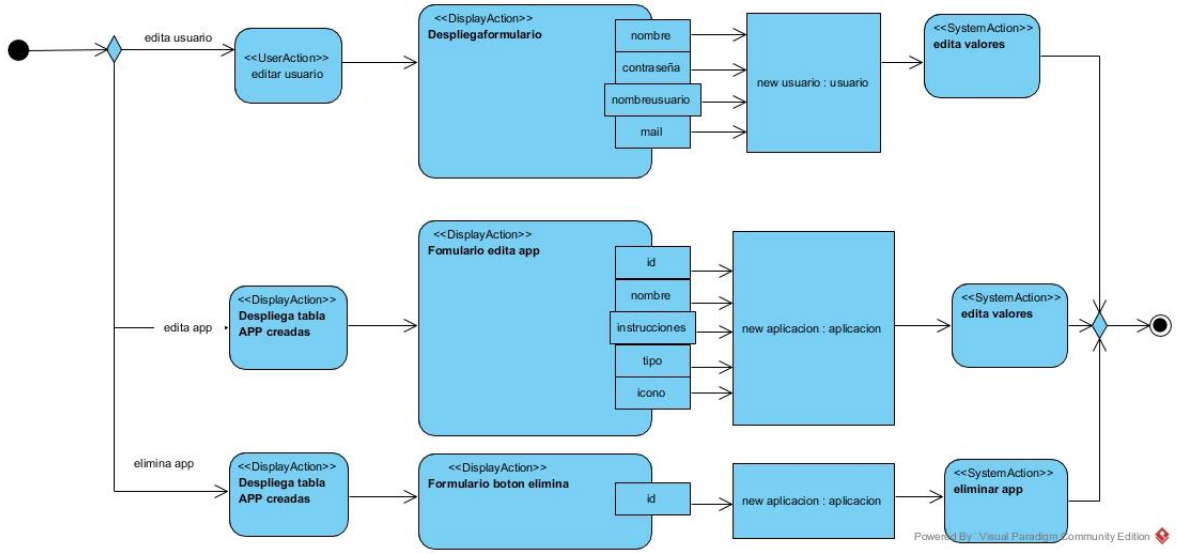

Figura 3.22 Diagrama de proceso funciones administrador

3.6 Actividades complementarias

El desarrollo de aplicaciones *NFC* para *Android* se lleva acabo con el *Android framework API* el cual soporta estas funcionalidades utilizando el formato de mensajes NDEF

3.6.1 NDEF

Es implementado con las clases *NdefMessage* y *NdefRecord* utilizando Java, provee los métodos para obtener y modificar los datos dentro de un Tag NFC que puede ser del tipo 1 al tipo 4, para el generador se utilizan etiquetas correspondientes al tipo 4 utilizando los estándares:

Tabla 3.2 estándares de propiedades NFC

Clase	Características	Especificaciones
MifareClassic (MIFARE Standard)	Provee el acceso a propiedades MIFARE Classic así como de entrada / salida, si el dispositivo soporta MIFARE	Dividido en dos sectores, y cada uno dividido en bloques, cada bloque es de 16 bytes.
MifareUltralight	Prove el acceso a propiedades MIFARE Ultralight aso como de entrada / salida, si el dispositivo soporta MIFARE	Consiste en una EEPROM Electrically Erasable Programmable Read-Only Memory (ROM programable y borrable eléctricamente) de 64bytes dividida en área de OTP *One True Pairing* (contraseña de un solo uso) y área de lectura y escritura.

La lectura del formato NDEF se maneja con un sistema de despachamiento de tags que analiza el tag descubierto, clasifica los datos e inicia una aplicación que reciba estos datos categorizados, al tener una aplicación que maneje tags NFC se declara un filtro de intención y se solicita el manejo de datos

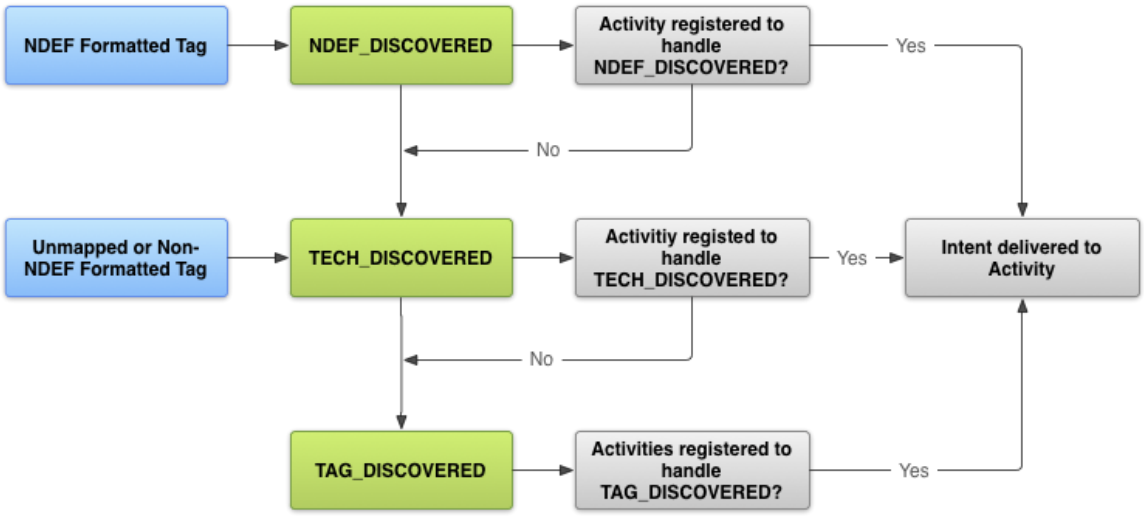

Figura 3.23 lectura de tag formateados NDEF Obtenido de Android Developers

3.6.2 Desarrollo de aplicación base

El desarrollo de la aplicación base se realiza con el IDE Android Studio, dicho IDE nos permite crear fácilmente dicha aplicación,

Para hacer uso de NFC dentro de una aplicación debe modificarse el manifest utilizando < uses-permission android:name="android.permission.NFC" />, la API nivel 9 es la mínima versión soportada.

```
public static void setupForegroundDispatch(final Activity activity, NfcAdapter adapter) throws
RuntimeException {
    final Intent intent = new Intent(activity.getApplicationContext(), activity.getClass());
    intent.setFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP);
    final PendingIntent pendingIntent = PendingIntent.getActivity(activity.getApplicationContext(),
    0, intent, 0);
    IntentFilter[] filters = new IntentFilter[1];
    String[][] techList = new String[][]{};
    filters[0] = new IntentFilter();
    filters[0].addAction(NfcAdapter.ACTION_NDEF_DISCOVERED);
    filters[0].addCategory(Intent.CATEGORY_DEFAULT);
    try {
        filters[0].addDataType(MIME_TEXT_PLAIN);
    } catch (IntentFilter.MalformedMimeTypeException e) {
        throw new RuntimeException("Check your mime type.");
    }
    adapter.enableForegroundDispatch(activity, pendingIntent, filters, techList);
}
public static void stopForegroundDispatch(final Activity activity, NfcAdapter adapter) {
    adapter.disableForegroundDispatch(activity);
}
```

Figura 3.24 lectura de tag formateados NDEF

Como se muestra en la *Figura 3.24* se le indica al despachador que la aplicación será la que resuelva los eventos de NFC.

Se crean los manejadores de eventos al encontrar un tag

```
private void handleIntent(Intent intent) {
    String action = intent.getAction();
    if (NfcAdapter.ACTION_NDEF_DISCOVERED.equals(action)) {
        String type = intent.getType();
        if (MIME_TEXT_PLAIN.equals(type)) {
            Tag tag = intent.getParcelableExtra(NfcAdapter.EXTRA_TAG);
            new NdefReaderTask().execute(tag);
        } else Log.d(TAG, "Wrong mime type: " + type);
    }
    else if (NfcAdapter.ACTION_TECH_DISCOVERED.equals(action)) {
        Tag tag = intent.getParcelableExtra(NfcAdapter.EXTRA_TAG);
        String[] techList = tag.getTechList();
        String searchedTech = Ndef.class.getName();
        for (String tech : techList) {
            if (searchedTech.equals(tech)) {
                new NdefReaderTask().execute(tag);
                break;
            }
        }
    }
}
```

Figura 3.25 Creador de manejador de eventos

Se realiza la lectura de los tags y se muestra en una etiqueta como se muestra en la figura 3.26.

```
private String readText(NdefRecord record) throws UnsupportedEncodingException {
      byte[] payload = record.getPayload();
      String textEncoding = ((payload[0] & 128) == 0) ? "UTF-8" : "UTF-16";
      int languageCodeLength = payload[0] & 0063;
      return new String(payload, languageCodeLength + 1, payload.length - languageCodeLength - 1,
      textEncoding);
  }

  @Override
  protected void onPostExecute(String result) {
      if (result != null) {
          mTextView.setText(result);
      }
  }
```

Figura 3.26 Lectura de etiquetas

La lectura de tags NFC puede llevarse a cabo de la forma mostrada en la figura 3.27

```
Button btnWrite = (Button)findViewById(R.id.button);
final TextView message = (TextView)findViewById(R.id.edit_message);
btnWrite.setOnClickListener(new OnClickListener() {

@Override
public void onClick(View v) {
    try{
        if(myTag == null){
        Toast.makeText(context, context.getString(R.string.error_notag), Toast.LENGTH_LONG).show();
        }else{
            write(message.getText().toString(),myTag);
            Toast.makeText(context, context.getString(R.string.ok_write), Toast.LENGTH_LONG).show();
        }
    }catch(IOException e){
       Toast.makeText(context, context.getString(R.string.error_write),Toast.LENGTH_LONG).show();
       e.printStackTrace();
    }catch(FormatException e){
        Toast.makeText(context, context.getString(R.string.error_write), Toast.LENGTH_LONG).show();
        e.printStackTrace();
    }
 }
});
```

Figura 3.27 Lectura de tags

De la misma manera para escribir un mensaje se lleva a cabo de la forma mostrada en la figura 3.28

```
private void write(String text, Tag tag) throws IOException, FormatException{
    NdefRecord[] records = {createRecord(text)};
    NdefMessage message = new NdefMessage(records);
    Ndef ndef = Ndef.get(tag);
    ndef.connect();
    ndef.writeNdefMessage(message);
    ndef.close();
}
@SuppressLint("NewApi") private NdefRecord createRecord(String text) throws
UnsupportedEncodingException{
    String lang = "us";
    byte[] textBytes = text.getBytes();
    byte[] langBytes = lang.getBytes("US-ASCII");
    int langLength = langBytes.length;
    int textLength = textBytes.length;
    byte[] arrMensaje = new byte[1 + langLength + textLength];
    arrMensaje[0] = (byte) langLength;
    System.arraycopy(langBytes, 0, arrMensaje, 1, langLength);
    System.arraycopy(textBytes, 0, arrMensaje, 1+langLength, textLength);
    NdefRecord recordNFC = new NdefRecord(NdefRecord.TNF_WELL_KNOWN,
                                NdefRecord.RTD_TEXT, new byte[0], arrMensaje);
    return recordNFC;
}
```

Figura 3.28 Escritura tags NFC

Tomando como base los códigos anteriores se crean 3 aplicaciones Android con funciones diferentes que servirán de base para el generador de aplicaciones. Las aplicaciones se enlistan a continuación:

- Aplicación de escritura: aplicación capaz de escribir en etiquetas NFC mensajes cortos utilizando un pequeño formulario.

- Aplicación de lectura: aplicación encargada de la lectura de las etiquetas NFC mostrando su contenido en pantalla.
- Aplicación de lista: aplicación capaz de leer etiquetas NFC para posteriormente ordenar las lecturas en una lista.

Las tres aplicaciones tienen por defecto un nombre, características e icono, estos estarán almacenados en una dirección dentro del disco duro, preferiblemente una ubicación dentro de el disco local D, teniendo una distribución como la mostrada en la Figura 3.29 la cual corresponde a la aplicación de escritura.

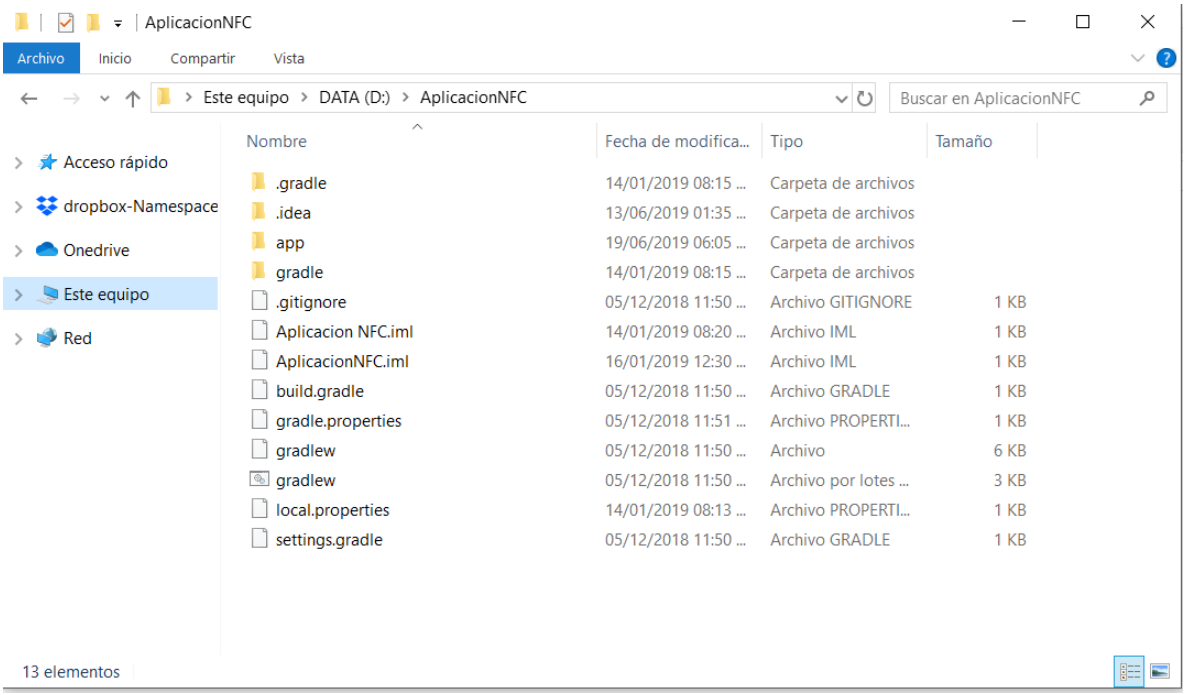

Figura 3.29 Estructura de la aplicación base de escritura.

3.7 Generación de aplicaciones

Teniendo una aplicación base que haga uso de la tecnología NFC en Android, el siguiente paso es la replicación de está utilizando las herramientas que brinda el SDK de Android Studio para la compilación desde línea de código.

3.7.1 Gradle

Gladle [28] es un avanzado kit de herramientas de construcción, para automatizar y gestionar el proceso de construcción. Cada configuración de compilación puede definir su propio conjunto de código y recursos, a la vez que reutiliza las partes comunes a todas las versiones de su aplicación. El plugin Android que hace uso del Gradle trabaja con el kit de herramientas de construcción para proporcionar procesos y configuraciones que son específicas para la construcción y prueba de aplicaciones en entornos Android.

Gradle y el plugin de Android funcionan independientemente de Android Studio. Por lo que se pueden crear aplicaciones Android desde la línea de comandos. La salida de la compilación es la misma tanto si está creando un proyecto desde la línea de comandos, en un equipo remoto o utilizando Android Studio.

El proceso de construcción de una aplicación Android, como se muestra en la figura 3.29, sigue estos pasos generales:

- El compilador convierte el código fuente en archivos DEX (Dalvik Executable, ejecutables Dalvik), que incluyen el código de bytes que se ejecuta en los dispositivos Android.
- El APK Packager (Empaquetador de APK) combina los archivos DEX y los recursos compilados en un solo APK. Sin embargo, antes de que su aplicación pueda instalarse y desplegarse en un dispositivo Android, debe firmar el APK.
- El APK Packager firma el APK utilizando el almacén de claves de depuración o de liberación:
  Si está creando una versión de depuración de su aplicación, es decir, una aplicación que sólo pretende probar y perfilar, el Packager firma la aplicación con el almacén de claves de depuración. Android Studio configura automáticamente los nuevos proyectos con un almacén de claves de depuración.
  Si está creando una versión de lanzamiento de la aplicación que desea publicar externamente, el empaquetador firma la aplicación con el almacén de claves de lanzamiento.
- Antes de generar el APK final, el Packager utiliza la herramienta zipalign[29] para optimizar su aplicación y utilizar menos memoria cuando se ejecuta en un dispositivo.
- Al final del proceso de compilación, tiene un APK de depuración o un APK de liberación de su aplicación que puede utilizar para desplegar, probar o liberar a usuarios externos.

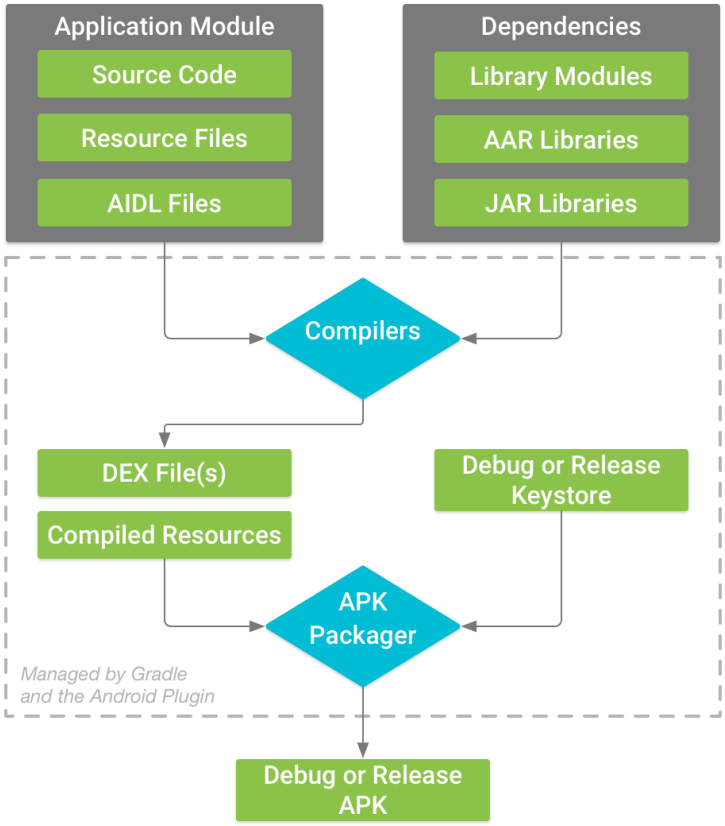

Figura 3.29 proceso de construcción de un típico modulo Android Obtenido de Android Developers

Para la compilación desde línea de comandos de una aplicación Android [30] se hace uso de los comandos del contenedor del gradlew disponible como archivo por lotes para Windows (gradlew.bat) en este caso se hace uso del comando "gradlew assembleDebug" mostrada en la Figura 3.30.

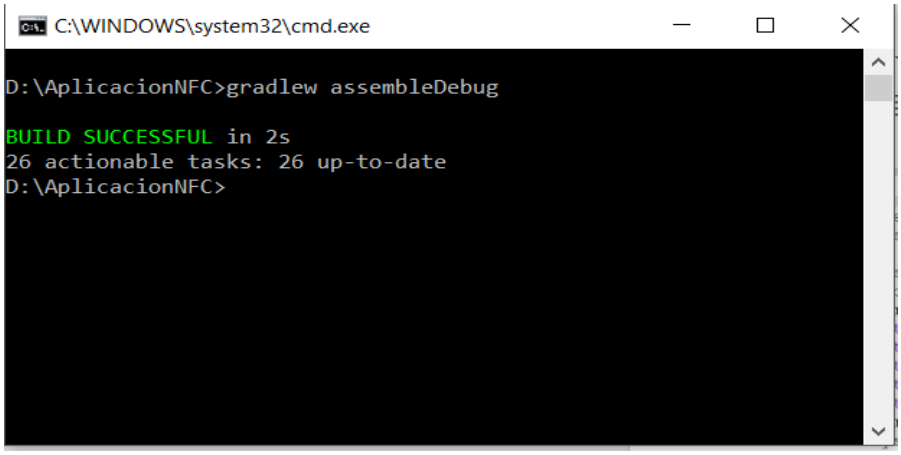

Figura 3.30 Compilación desde línea de comandos

El resultado es una aplicación .APK que puede ser instalada en dispositivos Android compatibles con NFC y que cuenten con la versión 5.0 en adelante, que es mostrada en la figura 3.3.

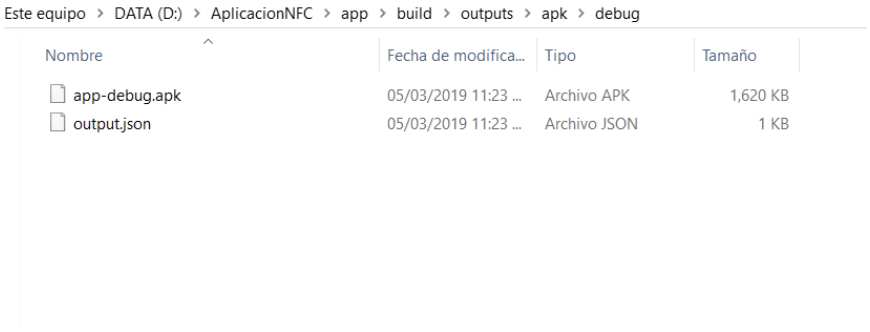

Figura 3.31 Archivo APK generado desde línea de comandos

3.7.2 Ejecución de instrucciones de línea de comando desde java

El sistema generador al ser construido en java debe de utilizar funciones de RMI[31] (Remote Method Invocation, Método de invocación remoto), las cuales ejecutan los comandos que puede realizar el gradlew desde una clase java, en la Figura 3.32 se observa el método para generar una copia de la aplicación base sin modificaciones, esta esta alojada en una dirección de el disco local D, de la misma forma se tiene la invocación al método *comandoPorEjecutar* el cual recibe la ubicación y el comando a ejecutar dentro de una clase java.

```
String ubicacionAppBase="D:\\\\AplicacionNFC";

System.out.println("inicia construccion");
comandoPorEjecutar("cmd /c cd "+ubicacionAppBase+ " & "
                   + "gradlew assembleDebug");

public static void comandoPorEjecutar(String ComandoPE){
    try{
        System.out.println("ejecutando: "+ComandoPE);
        Process proceso = Runtime.getRuntime().exec(ComandoPE);

        InputStreamReader entrada = new InputStreamReader(proceso.getInputStream());
        BufferedReader bufferLeer = new BufferedReader(entrada);
        String salida = null;

        if((salida=bufferLeer.readLine()) != null){
            System.out.println("se realizo la ejecusion");
            while ((salida=bufferLeer.readLine()) != null){
              System.out.println(salida);
            }
        }else{
            System.out.println("no existe una salida");
        }
    }catch (IOException e) {
            System.out.println("Excepción: fallo");
            e.printStackTrace();
    }
}
```

Figura 3.32 Generación desde una clase java

Con la ejecución del código mostrado en la anterior, se tiene la salida mostrada en la Figura 3.33 donde se muestra el proceso y correcta ejecución de las sentencias para la compilación del APK

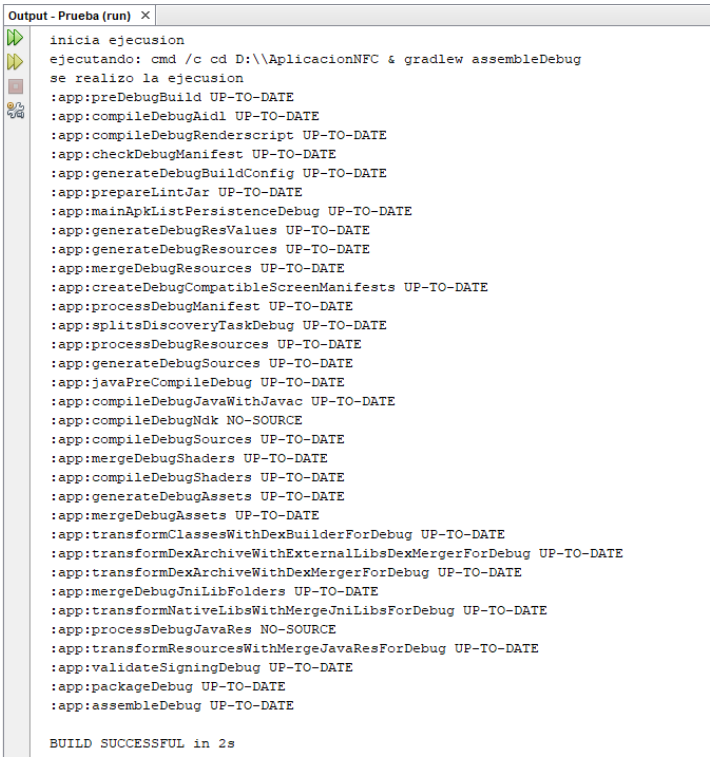

Figura 3.32 Output de la generación desde una clase Java

3.7.3 Modificación de archivos XML

Las aplicaciones Android declaran elementos de la interfaz en XML[32], proporcionando un vocabulario XML simple que puede ser manipulado fácilmente desde una clase java utilizando los métodos de BufferReader dentro de las clases java. Para la aplicación base se definen los elementos a contener dentro de la interfaz en XML, tomando como ejemplo el archivo Strings.xml de la aplicación de escritura mostrado en la figura 3.33, dicho archivo es el encargado de contener el texto a mostrar dentro de las etiquetas de la aplicación.

```
<resources>
    <string name="app_name">Nombre</string>
    <string name="error_write">error de escritura</string>
    <string name="ok_write">escritura correcta</string>
    <string name="error_notag">error no hay tag</string>
    <string name="ok_detected">ok detectado</string>
    <string name="escibe_tu_mensaje">escribe el mensaje</string>
    <string name="mensaje">Mensaje</string>
    <string name="escribir">Escribir</string>
</resources>
```

Figura 3.33 archivo strings.xml

Los cambios que se realizan en el archivo se muestran en la figura 3.34 donde se llama a la clase EscribirFichero para realizar las modificaciones al archivo.

```
File Ffichero=new File("D:\\AplicacionNFC\\app\\src\\main\\res\\values\\strings.xml");

 EcribirFichero(Ffichero, "<resources>\n" +
"    <string name=\"app_name\">"+nombre+"</string>\n" +
"    <string name=\"error_write\">error de escritura</string>\n" +
"    <string name=\"ok_write\">escritura correcta</string>\n" +
"    <string name=\"error_notag\">"+error+"</string>\n" +
"    <string name=\"ok_detected\">ok detectado</string>\n" +
"    <string name=\"escibe_tu_mensaje\">"+mensaje+"</string>\n" +
"    <string name=\"mensaje\">"+msj+"</string>\n" +
"    <string name=\"escribir\">"+escribir+"</string>\n" +
"</resources>");


public static void EcribirFichero(File Ffichero,String SCadena){
    try {
        if(!Ffichero.exists()){
            Ffichero.createNewFile();
        }
        BufferedWriter Fescribe=new BufferedWriter(new OutputStreamWriter(
            new FileOutputStream(Ffichero,true), "utf-8"));
        Fescribe.write(SCadena + "\r\n");
        Fescribe.close();
   } catch (Exception ex) {
       System.out.println(ex.getMessage());
   }
}
```

Figura 3.34 modificación de archivos xml desde java

Al modificar estas etiquetas el contenido en la pantalla de la aplicación cambia dependiendo de los datos proporcionados por el usuario al momento de generar la aplicación los cambios se muestran en la Figura 3.35 donde se observa la aplicación ya compilada con los datos proporcionados por el usuario y la Figura 3.36 donde se muestra el cambio dentro del archivo XML.

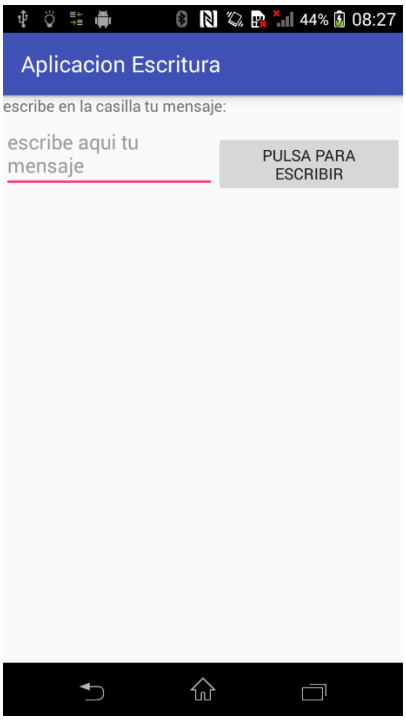

Figura 3.35 captura de pantalla de la aplicación con cambios desde el XML

```
<resources>
    <string name="app_name">Aplicacion Escritura</string>
    <string name="error_write">error de escritura</string>
    <string name="ok_write">escritura correcta</string>
    <string name="error_notag">hubo un error de lectura</string>
    <string name="ok_detected">ok detectado</string>
    <string name="escibe_tu_mensaje">escribe en la casilla tu mensaje:</string>
    <string name="mensaje">escribe aqui tu mensaje</string>
    <string name="escribir">pulsa para escribir</string>
</resources>
```

Figura 3.36 Archivo XML resultante.

De la misma forma se modifica el archivo biuld.gradle de cada aplicación para asignar una versión de la aplicación dependiendo del número de compilación realizado por el usuario de la aplicación.

Capítulo 4. Resultados

El sistema propuesto en el presente documento esta diseñado para poner a disposición del publico en general la tecnología NFC, de esta manera permitiendo generar aplicaciones con fines específicos sin necesidad de escribir líneas de código.

4.1 Casos de estudio

Para los casos de estudio se tomaron en cuenta dos problemáticas por un lado el control de préstamo de libros a estudiantes que portaran una identificación, y el control de alumnos dentro de un aula.

4.1.1 Biblioteca

Para el primer caso de estudio se toman en cuenta las funcionalidades de inventarios, e identificación de objetos, por lo que se hacen uno de dos aplicaciones que pueden generarse con el sistema, la aplicación de escritura para llevar el control de las etiquetas de la cual solo hará uso el encargado al realizar el préstamo y actualización y la aplicación de lectura de la cual se podrá hacer uso los estudiantes para identificar la fecha de vencimiento de su préstamo. De tal manera que al hacer uso del generador se crean dos aplicaciones que en conjunto podrán realizar las siguientes acciones:

- El encargado de la biblioteca registre, modifique y elimine datos de libros en la etiqueta adherida al libro.
- El alumno se identifique fácilmente y registre el préstamo de un libro su identificación.
- El alumno podrá ver la vigencia de su préstamo directamente al leer la etiqueta dentro del libro.
- El encargado de la biblioteca verifique que el alumno no tenga adeudos de libros para poder realizar un nuevo préstamo.

El usuario ya registrado en el sistema genera primero un proyecto, en este elige el tipo de aplicación por crear, mostrado en la figura 4.1

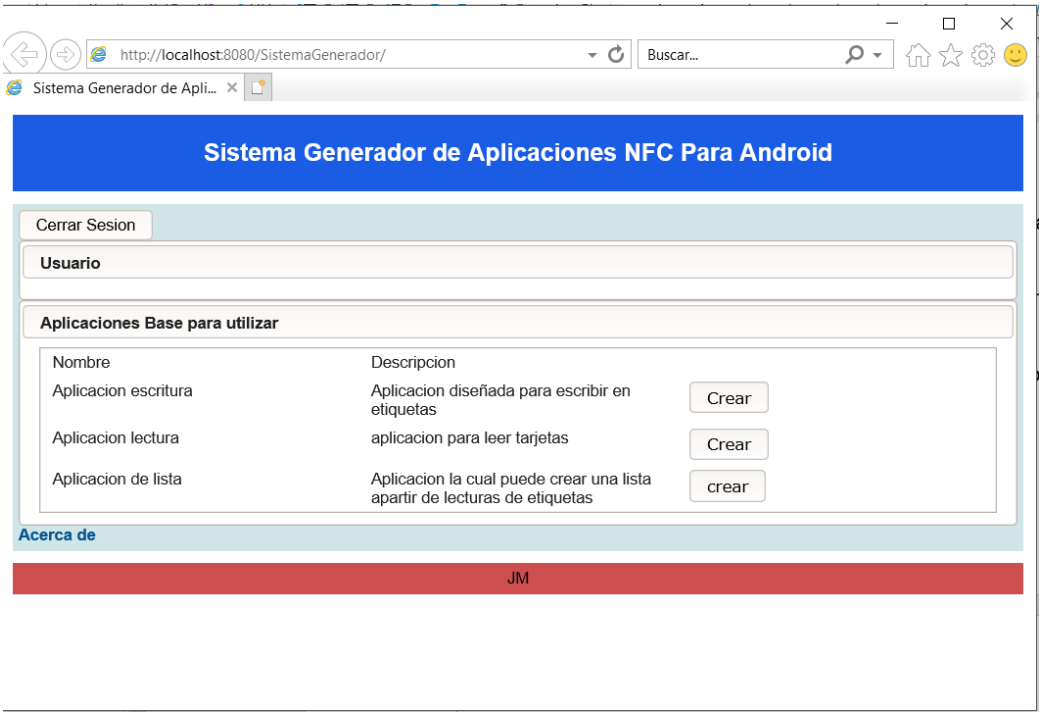

Figura 4.1 ventana de elección de aplicación por crear.

Después de hacer la elección de la aplicación por construir, se eligen los valores que tendrá la aplicación, entre ellos nombre, icono e instrucción a escribir, mostrado en la Figura 4.2.

Figura 4.2 ventana de elección de opciones.

Al seleccionar generar aplicación permite tener un enlace del cual se descarga la aplicación como un APK, mostrado en la Figura 4.3, para su instalación en cualquier dispositivo Android compatible con la tecnología NFC, mostrado en la figura 4.4 .

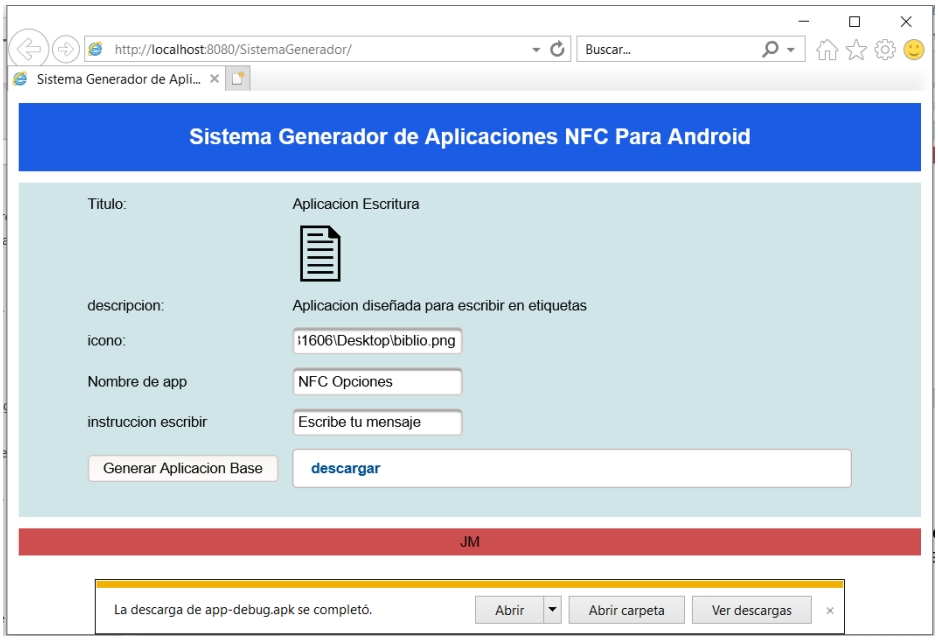

Figura 4.3 descarga de la aplicación.

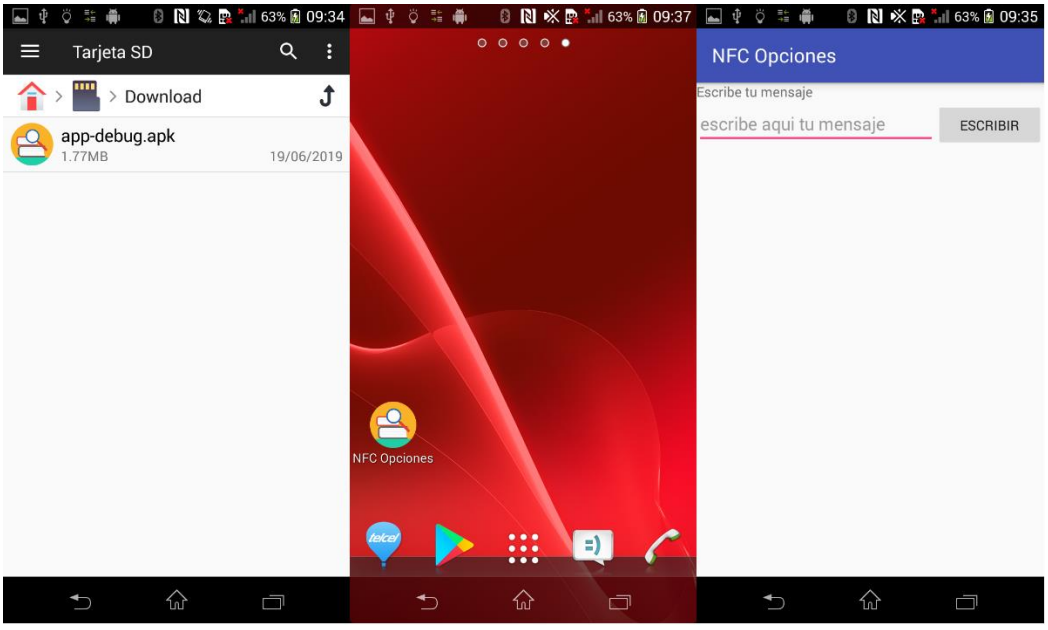

Figura 4.4 Instalación y ejecución de la aplicación generada.

Por la parte del uso de esta aplicación los libros contarán con una etiqueta NFC adherible mostrada en la Figura 4.5, que puede colocarse sin problemas en la tapa de estos, y para los estudiantes una tarjeta plástica para su identificación, mostrada

en la figura 4.6, la cual puede ser impresa en su exterior para cualquier otro uso de identificación con métodos de baja tecnología.

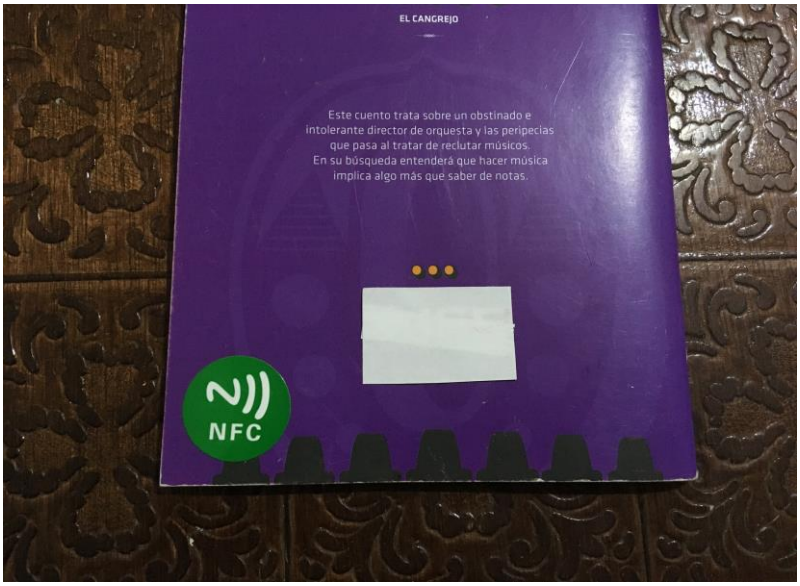

Figura 4.5. tag NFC en forma de etiqueta adherible.

Figura 4.6. tag NFC en formato tarjeta plástica.

La aplicación generada es capaz de escribir y modificar los datos en los libros y las credenciales plásticas, mostrado en la Figura 4.7, la aplicación complementaria para los estudiantes es capaz de leer datos en los tags NFC, mostrado en la Figura 4.8, llevando así un control de las fechas de devolución y los estudiantes que solicitan préstamos de libros.

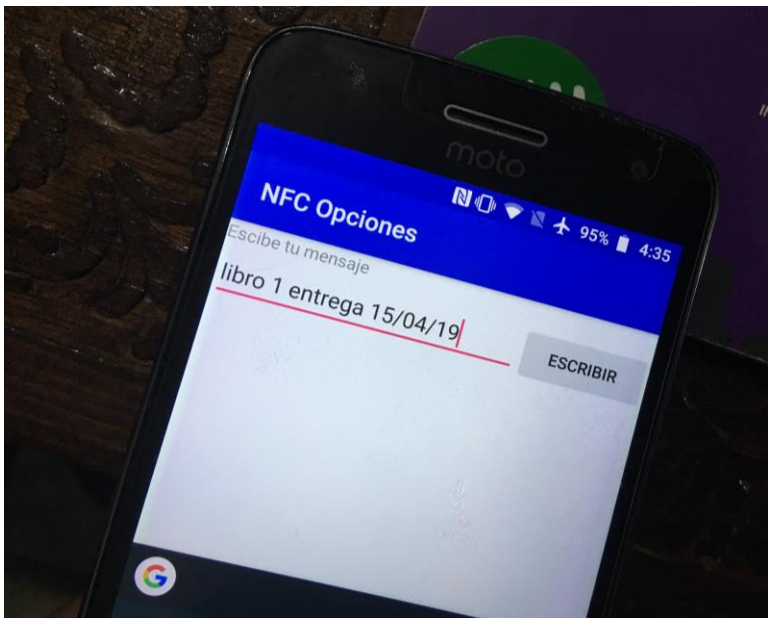

Figura 4.7. escritura en un tag NFC

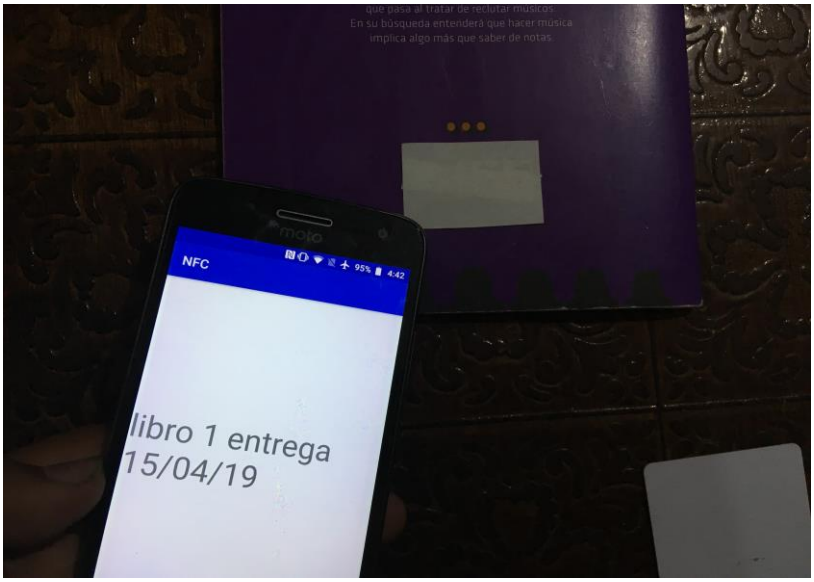

Figura 4.8. lectura de la información del tag NFC de un libro.

4.1.2 Pase de lista

Para el segundo caso de estudio se toma en cuenta la funcionalidad de lista de cotejo por lo que se vuelve a hacer uso de dos aplicaciones, la aplicación de escritura y la aplicación de lista, para en primer lugar registrar a los alumnos con su propia tarjeta NFC plástica, seguido de poder tomar lista de asistencia al pasar la tarjeta sobre el dispositivo que haga uso de NFC, en conjunto podrán realizar las siguientes acciones:

- El encargado de la clase registre cada uno de los alumnos en su respectiva tarjeta NFC.
- El alumno se identifique fácilmente al ingresar a la clase.

El usuario ya registrado en el sistema genera las aplicaciones a usar como en el caso de estudio anterior.

Las aplicaciones ya generadas permiten llevar el registro y el control de los alumnos dentro de la clase de manera sencilla.

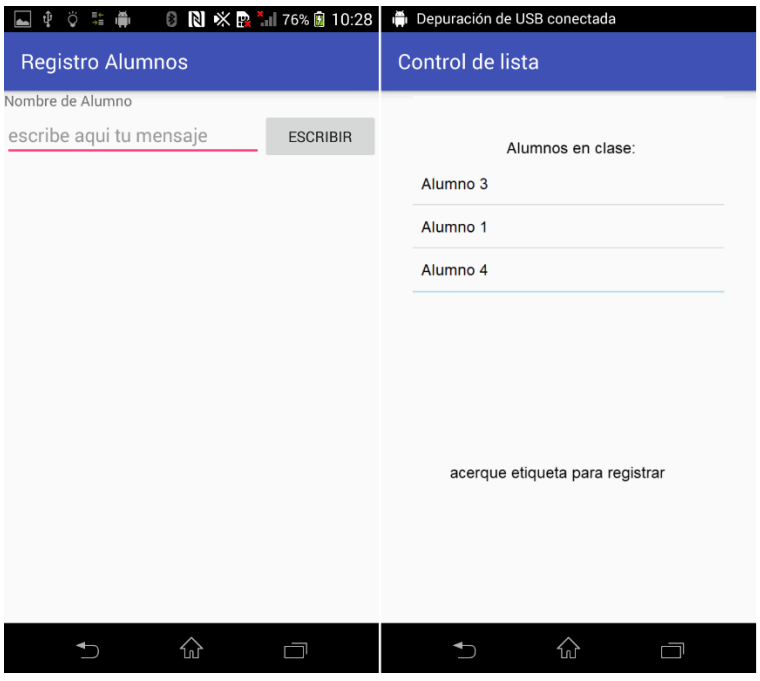

Figura 4.8. aplicación de registro y control de lista.

Para la parte de los usuarios estos tendrán una tarjeta NFC plástica donde estarán registrados sus datos como se muestra en la figura 4.9

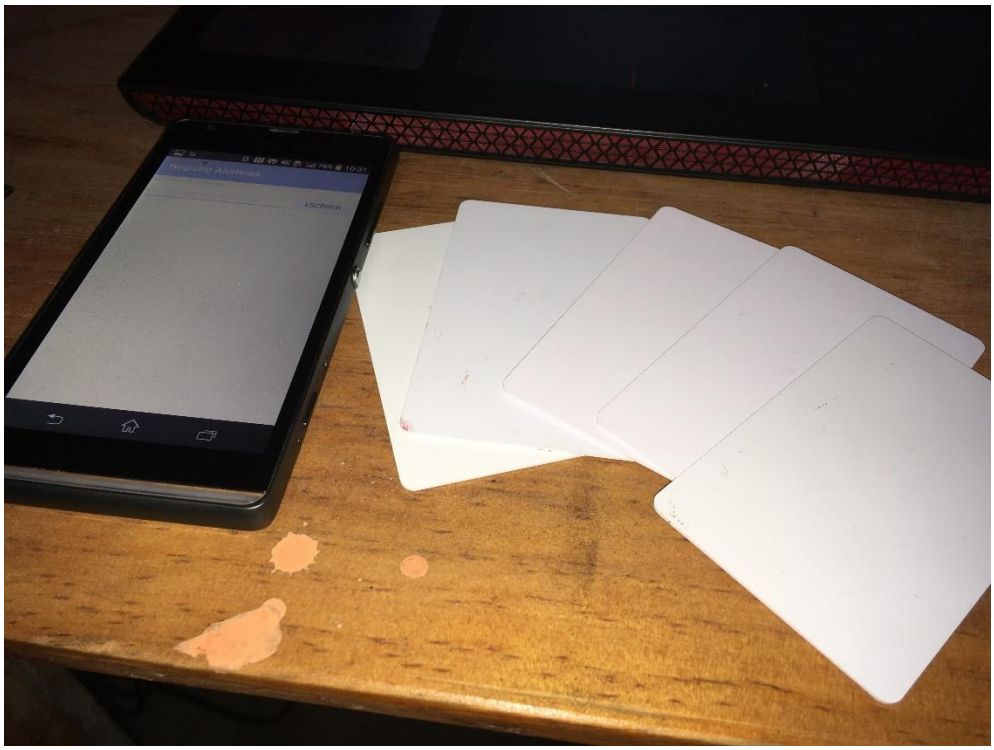

Figura 4.9. tarjetas NFC para el registro de usuarios.

Conclusiones

Como se menciona en el presente documento, *NFC* tiene diversas aplicaciones que no se han explotado totalmente debido a ciertos conocimientos que son requeridos para su desarrollo e implementación.

El desarrollo de aplicaciones móviles es se ha convertido en parte importante de la computación, ya que la mayoría de los dispositivos móviles se ocupan día con día para distintas actividades en las que podría integrarse tecnología NFC.

El desarrollo de aplicaciones móviles es fundamental, ya que la mayoría de los dispositivos móviles se ocupan día con día para distintas actividades en las que podría integrarse tecnología NFC.

El desarrollo del sistema generador de aplicaciones Android NFC, realizado en un entorno web, permitirá que el uso de las aplicaciones NFC sea más frecuente y se consolide como una tecnología de uso diario.

Los usuarios que utilicen el generador tendrán solucionada una necesidad fácilmente con el uso de esta herramienta.

Recomendaciones

La forma en que las aplicaciones pueden asistir al usuario es variada y su efectividad es conforme a como se esté utilizando.

El uso que se le den a las aplicaciones creadas con esta herramienta es responsabilidad del usuario, ya que no se cuentan con revisiones sistemáticas de cada aplicación al momento de ser construidas, aunque, en un futuro se podría implementar un módulo que filtre y no permita el uso de lenguaje indebido o sensible.

Las etiquetas NFC son fáciles de conseguir y tienen distintas presentaciones por lo que se tienen un sinfín de usos con ellas que no se encuentran totalmente analizadas en la realización de este sistema.

De la misma forma, la parte del hardware para hacer uso de este sistema es variado y podría presentar fallas por lo que se recomiendan etiquetas NFC tipo 4 y dispositivos Android que cuenten con la versión 5.0 en adelante.

Bibliografía

[1] Google, “Glass,” 2018. [Online]. Available: https://www.x.company/glass/.

[2] S. VIBHOR, P. Gusain, and K. PRASHANT, “Near Field Communication,” *Adv. Intell. Syst. Res.*, vol. 2013, no. Cac2s, pp. 342–345, 2013.

[3] R. Tatiraju, R. Gupta, S. Salunke, and R. Yeolekar, “NFiD : An NFC based system for Digital Business Cards,” pp. 1510–1512, 2017.

[4] A. Meschtscherjakov, C. Gschwendtner, M. Tscheligi, and P. Sundström, “Co-designing for NFC and ATMs: An Inspirational Bits Approach,” in *Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services*, 2013, pp. 422–427.

[5] U. Jambusaria, N. Katwala, and D. Mistry, “Secure smartphone unlocking using NFC,” *Procedia Comput. Sci.*, vol. 45, no. C, pp. 465–469, 2015.

[6] D. Remedios, L. Sousa, M. Barata, and L. Osorio, “Nfc Technologies in Mobile Phones and Emerging Applications,” pp. 425–434, 2006.

[7] A. Paus, “Near Field Communication in Cell Phones,” *White Pap. Ruhr-Universitat Bochum*, 2007.

[8] T. N. F. C. (NFC) Forum, “Protocol Technical Specifications,” *NFC Forum Technical Specifications*, 2017. [Online]. Available: https://nfc-forum.org/our-work/specifications-and-application-documents/specifications/nfc-forum-technical-specifications/#protocol.

[9] The Near Field Communication (NFC) Forum, “NFC Forum Specification Architecture,” 2017. [Online]. Available: https://nfc-forum.org/our-work/specifications-and-application-documents/specifications/.

[10] A. P. Volpentesta, N. Frega, and G. Filice, “Interactions patterns in NFC interfaces for applications and services,” *IFIP Adv. Inf. Commun. Technol.*, vol. 408, pp. 324–334, 2013.

[11] “Application Fundamentals,” 2018. [Online]. Available: https://developer.android.com/guide/components/fundamentals.html.

[12] L. Banda and I. López Martínez, “Proceso para la Generación Automática de Aplicaciones de Realidad Aumentada para Dispositivos Móviles,” in *Congreso Internacional de Investigación e Innovación en Ingeniería de Software, CONISOFT’13*, 2013.

[13] Oracle Corporation, “JavaServer Faces Technology Overview,” 2018. [Online]. Available: http://www.oracle.com/technetwork/java/javaee/overview-140548.html.

[14] J. C. Preciado, M. Linaje, F. Sanchez, and S. Comai, “Necessity of Methodologies to Model Rich Internet Applications,” in *Proceedings of the Seventh IEEE International Symposium on Web Site Evolution*, 2005, pp. 7–13.

[15] N. Koch, A. Knapp, G. Zhang, and H. Baumeister, “Uml-Based Web Engineering,” in *Web Engineering: Modelling and Implementing Web Applications*, G. Rossi, O. Pastor, D. Schwabe, and L. Olsina, Eds. London: Springer London, 2008, pp. 157–

191.

[16] U. D. Alan and D. Birant, “Server-Based Intelligent Public Transportation System with NFC,” *IEEE Intell. Transp. Syst. Mag.*, vol. 10, no. 1, pp. 30–46, 2018.

[17] F. Fortat, M. Laurent, and M. Simatic, “Games Based on Active NFC Objects: Model and Security Requirements,” in *Proceedings of the 2015 International Workshop on Network and Systems Support for Games*, 2015, p. 12:1--12:3.

[18] J. S. Ho, A. J. Yeh, S. Kim, and A. S. Y. Poon, “Wireless Powering for Miniature Implantable Systems,” in *Neural Computation, Neural Devices, and Neural Prosthesis*, Z. Yang, Ed. New York, NY: Springer New York, 2014, pp. 313–333.

[19] F. Dang *et al.*, “Large-scale invisible attack on AFC systems with NFC-equipped smartphones,” *Proc. - IEEE INFOCOM*, pp. 1–9, 2017.

[20] A. Yakovlev, S. Kim, and A. Poon, “Implantable biomedical devices: Wireless powering and communication,” *IEEE Commun. Mag.*, vol. 50, no. 4, pp. 152–159, 2012.

[21] T. Ongkosit and S. Takada, “Responsiveness Analysis Tool for Android Application,” *Proc. 2Nd Int. Work. Softw. Dev. Lifecycle Mob.*, pp. 1–4, 2014.

[22] G. W. H. Tan, K. B. Ooi, S. C. Chong, and T. S. Hew, “NFC mobile credit card: The next frontier of mobile payment?,” *Telemat. Informatics*, vol. 31, no. 2, pp. 292–307, 2014.

[23] S. Blackshear, A. Gendreau, and B.-Y. E. Chang, “Droidel: A General Approach to Android Framework Modeling,” in *Proceedings of the 4th ACM SIGPLAN International Workshop on State Of the Art in Program Analysis*, 2015, pp. 19–25.

[24] D. Petrelli, M. T. Marshall, S. O’Brien, P. McEntaggart, and I. Gwilt, “Tangible data souvenirs as a bridge between a physical museum visit and online digital experience,” *Pers. Ubiquitous Comput.*, vol. 21, no. 2, pp. 281–295, 2017.

[25] A. Jøsang *et al.*, “Local user-centric identity management,” *J. Trust Manag.*, vol. 2, no. 1, p. 1, 2015.

[26] R. Nandakumar, K. K. Chintalapudi, V. Padmanabhan, and R. Venkatesan, “Dhwani: Secure peer-to-peer acoustic NFC,” *Annu. Conf. ACM Spec. Interes. Gr. Data Commun. Appl. Technol. Archit. Protoc. Comput. Commun. ACM SIGCOMM 2013*, vol. 43, no. 4, pp. 63–74, 2013.

[27] F. Boukayoua, J. Vossaert, B. De Decker, and V. Naessens, “Using a smartphone to access personalized web services on a workstation,” *IFIP Adv. Inf. Commun. Technol.*, vol. 375 AICT, pp. 144–156, 2012.

[28] Android Developers, “Configure your build,” 2018. [Online]. Available: https://developer.android.com/studio/build/index.html.

[29] Android Developers, “zipalign,” 2018. [Online]. Available: https://developer.android.com/studio/command-line/zipalign.html.

[30] Android Developers, “Compilar Aplicacion desde linea de commando,” 2018. [Online]. Available: https://developer.android.com/studio/build/building-cmdline?hl=es-419.

[31] Oracle Corporation, “Getting Started Using Java™ RMI,” 2018. [Online]. Available: https://docs.oracle.com/javase/7/docs/technotes/guides/rmi/hello/hello-world.html#start%0A%0A.

[32] Android Developers, “Diseños,” 2018. [Online]. Available: https://developer.android.com/guide/topics/ui/declaring-layout?hl=es-419.