



Universidad de las Ciencias Informáticas

Facultad 4

*“Sistema para la gestión de actualizaciones de videojuegos
Android mediante la red”*

*Trabajo de diploma para optar por el título de Ingeniero en Ciencias
Informáticas*

Autor:

René Orta Martínez

Tutores:

MSc. Andy Hernández Paez

Ing. Fredy Almenares Fleitas

Ing. Reinaldo García Maturell

La Habana, 2018



“Revolución y Educación son una sola cosa”

-Fidel Castro

Declaración de autoría

Declaramos ser autores del trabajo de diploma: “Sistema para la gestión de actualizaciones de videojuegos Android mediante la red” y autorizo al Centro de Entornos Interactivos 3D, Vertex de la Universidad de las Ciencias Informáticas; así como a dicha Universidad para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

René Orta Martínez

Firma del Autor

Ing. Fredy Almenares Fleitas

Firma del Tutor

Ing. Reinaldo Garcia Maturell

Firma del Tutor

MSc. Andy Hernández Paez

Firma del Tutor

Agradecimientos

A mis tutores y todas aquellas personas que contribuyeron de manera positiva en esta etapa de mi vida.



Dedicatoria

Dedicada a mi Mamá

Resumen

En la actualidad los grandes avances alcanzados a nivel mundial en las telecomunicaciones, la telefonía móvil y el sistema operativo Android, así como en las redes, han propiciado el surgimiento de un nuevo estilo de vida social en el cual los usuarios interactúan con el mundo que les rodea a través de diversas aplicaciones móviles. Estos tipos de aplicaciones deben mantenerse actualizadas, para proporcionarle al individuo las últimas mejoras, noticias e informaciones. En el Centro de Entornos Interactivos 3D, Vertex, de la Universidad de las Ciencias Informáticas, se desarrollan videojuegos para la plataforma Android sin contar con una herramienta o sistema para su actualización. Teniendo en cuenta lo anterior, el presente trabajo tiene como objetivo desarrollar un sistema para la gestión de las actualizaciones de los videojuegos que se implementen en el centro Vertex. El sistema desarrollado cuenta con una serie de interfaces para gestionar actualizaciones y una Interfaz de Programación de Aplicaciones para el envío de las mismas en Notación de Objetos JavaScript hacia los videojuegos Android mediante un servicio web REST. El proceso de desarrollo de la propuesta de solución se encaminó utilizando las fases de Planificación, Diseño, Implementación y Pruebas, propuestas por la metodología Programación Extrema. Los procesos de verificación y validación de la solución desarrollada se aplicaron mediante las pruebas de unidad y aceptación, diseñando casos de pruebas consistentes para comprobar el funcionamiento adecuado de las operaciones implementadas.

Palabras clave: actualización, servicio web, sistema de gestión de actualizaciones, Vertex

Índice de contenidos

Resumen	V
Introducción	1
Capítulo I: Fundamentación teórica	5
1.1 Introducción.....	5
1.2 Conceptos fundamentales asociados a la investigación	5
1.2.1 Servicio web.....	5
1.3 Estudios de los sistemas homólogos	6
1.3.1 Estudios de los sistemas homólogos en el ámbito internacional	6
1.3.2 Estudios de los sistemas homólogos en el ámbito nacional.....	7
1.4 Herramientas, tecnologías y lenguajes	9
1.4.1 Lenguajes de Programación	9
1.4.2 Entornos de desarrollo integrado	10
1.4.3 Herramienta de modelado.....	12
1.4.4 Lenguaje de modelado.....	12
1.4.5 Sistemas Gestores de Base de Datos.....	12
1.4.6 Servidor Web	13
1.4.7 Metodología de desarrollo de software.....	14
1.5 Conclusiones parciales	17
Capítulo II: Características y diseño.....	18
2.1 Introducción.....	18
2.2 Propuesta de Solución	18
2.3 Levantamiento de Requisitos Funcionales y Requisitos no Funcionales	20
2.4 Planificación	22
2.4.1 Historias de Usuario.....	22
2.4.2 Estimación de esfuerzo por historia de usuario	29
2.4.3 Plan de Duración de Iteraciones	30
2.4.4 Plan de Entrega	31
2.5 Diseño	32
2.5.1 Patrón Arquitectónico.....	32

2.5.2 Tarjetas CRC	32
2.5.3 Patrones de Diseño.....	37
2.6 Conclusiones parciales.....	39
Capítulo III: Implementación y pruebas.....	40
3.1 Introducción.....	40
3.2 Implementación	40
3.2.1 Tareas de Ingeniería	40
3.3 Pruebas.....	42
3.3.1 Pruebas Unitarias y de Aceptación	42
3.4 Aplicación Android para pruebas	47
3.5 Conclusiones parciales.....	50
Conclusiones generales.....	51
Recomendaciones	52
Referencias Bibliográficas	53
Anexos.....	57
Anexo 1	57
Anexo 2	¡Error! Marcador no definido.
Anexo 3	¡Error! Marcador no definido.

Índice de Tablas

Tabla 1. REST VS SOAP (Fuente: Elaboración propia)	6
Tabla 2. Sistemas Homólogos (Fuente: Elaboración propia).....	8
Tabla 3. Metodologías (Fuente: Elaboración propia).....	15
Tabla 4. Metodología XP (Fuente: Elaboración propia).....	16
Tabla 5. Requisitos Funcionales (Fuente: Elaboración propia).....	20
Tabla 6. Requisitos No Funcionales (Fuente: Elaboración propia)	21
Tabla 7. HU_1 (Fuente: Elaboración propia)	22
Tabla 8. HU_2 (Fuente: Elaboración propia).....	23
Tabla 9. HU_4 (Fuente: Elaboración propia).....	24
Tabla 10. HU_6 (Fuente: Elaboración propia)	25
Tabla 11. HU_7 (Fuente: Elaboración propia).....	26
Tabla 12. HU_9 (Fuente: Elaboración propia).....	27
Tabla 13. HU_14 (Fuente: Elaboración propia)	28
Tabla 14. Estimación de Esfuerzo por Historia de Usuario (Fuente: Elaboración propia)	29
Tabla 15. Plan de Duración de Iteraciones (Fuente: Elaboración propia).....	30
Tabla 16. Plan de Entrega (Fuente: Elaboración propia).....	31
Tabla 17. TarjCRC-Actualización (Fuente: Elaboración propia)	33
Tabla 18. TarjCRC-ActualizaciónPlugin (Fuente: Elaboración propia).....	33
Tabla 19. TarjCRC-ActualizaciónGeneradas (Fuente: Elaboración propia).....	33
Tabla 20. TarjCRC-Question (Fuente: Elaboración propia)	34
Tabla 21. TarjCRC-Usuario (Fuente: Elaboración propia)	34
Tabla 22. TarjCRC-ClaseAbstracta (Fuente: Elaboración propia)	34
Tabla 23. TarjCRC-UsuarioDB (Fuente: Elaboración propia)	35
Tabla 24. TarjCRC-QuestionDB (Fuente: Elaboración propia)	35
Tabla 25. TarjCRC-ActualizaciónDB (Fuente: Elaboración propia)	35
Tabla 26. TarjCRC-ActualizaciónPluginDB (Fuente: Elaboración propia).....	36
Tabla 27. TarjCRC-ActualizaciónGeneradasDB (Fuente: Elaboración propia)	36
Tabla 28. TarjCRC-Servidor (Fuente: Elaboración propia).....	36
Tabla 29. TI_1 (Fuente: Elaboración propia).....	40
Tabla 30. TI_2 (Fuente: Elaboración propia).....	41
Tabla 31. TI_3 (Fuente: Elaboración propia).....	41
Tabla 32. PA1_HU1 (Fuente: Elaboración propia)	43
Tabla 33. PA1_HU21 (Fuente: Elaboración propia)	43
Tabla 34. PA1_HU22 (Fuente: Elaboración propia)	44
Tabla 35. PA1_HU3 (Fuente: Elaboración propia)	44
Tabla 36. HU_3 (Fuente: Elaboración propia)	57
Tabla 37. HU_5 (Fuente: Elaboración propia).....	59
Tabla 38. HU_8 (Fuente: Elaboración propia).....	60
Tabla 39. HU_10 (Fuente: Elaboración propia)	61
Tabla 40. HU_11 (Fuente: Elaboración propia).....	62
Tabla 41. HU_12 (Fuente: Elaboración propia).....	64
Tabla 42. HU_13 (Fuente: Elaboración propia).....	66
Tabla 43. HU_15 (Fuente: Elaboración propia).....	67

*Sistema para la gestión de actualizaciones de videojuegos Android
mediante la red*

Tabla 44. HU_16 (Fuente: Elaboración propia)	68
Tabla 45. HU_17 (Fuente: Elaboración propia)	69

Índice de Ilustraciones

Ilustración 1. Prototipo Autenticación Interfaz (Fuente: Elaboración propia).....	18
Ilustración 2. Prototipo Principal Interfaz (Fuente: Elaboración propia)	19
Ilustración 3. Prototipo Actualizaciones Interfaz (Fuente: Elaboración propia)	19
Ilustración 4. Observador (Fuente: Elaboración propia)	38
Ilustración 5. Instancia única (Fuente: Elaboración propia)	39
Ilustración 6. Fachada (Fuente: Elaboración propia)	39
Ilustración 7. V.0.1 (Fuente: Elaboración propia).....	46
Ilustración 8. V.0.7 (Fuente: Elaboración propia).....	46
Ilustración 9. V.1.0 (Fuente: Elaboración propia).....	47
Ilustración 10. Interfaces APK Android (Fuente: Elaboración propia)	48
Ilustración 11. Función Android HttpPostPeticionAsync (Fuente: Elaboración propia)	48
Ilustración 12. Respuesta de actualización de videojuego a partir de texto (Fuente: Elaboración propia) ..	49
Ilustración 13. Respuesta de actualización de videojuego a partir de plugin (Fuente: Elaboración propia)	49
Ilustración 14. Matriz Urgencia-Importancia (Fuente: Elaboración propia)	57

Introducción

En las últimas décadas el gran desarrollo alcanzado en las telecomunicaciones, las redes inalámbricas, la telefonía móvil y los *Smartphone*, junto con el sistema operativo Android, han dado paso a nuevos y revolucionarios comportamientos sociales, basados en las facilidades que traen consigo las innovadoras aplicaciones móviles existentes en el mercado internacional. Las aplicaciones móviles brindan a los usuarios una nueva forma de interactuar con el mundo que les rodea a través de los servicios de mensajería, almacenamiento, información, comunicación, entretenimiento, descarga de datos, entre otros, que implementan estas herramientas portátiles. Las mismas, son capaces de interrelacionarse con el entorno teleinformático existente a través de diversos mecanismos, herramientas o sistemas, logrando así su constante actualización.

Los sistemas dedicados a la actualización de aplicaciones móviles juegan un papel fundamental para brindar soporte a las mismas, a través de las Interfaces de Programación de Aplicaciones (API) que suelen tener asociadas. Las APIs son vitales en los procesos de transferencia de datos en Internet, mediante los servicios web profesionales que son implementados a partir de esta alternativa. Utilizando el protocolo de transferencia de hipertextos (HTTP), las APIs permiten obtener y generar datos en formatos como el Lenguaje de Marcas Extensible (XML) o la Notación de Objetos JavaScript (JSON). Esto permite el flujo de información y la actualización de los sistemas informáticos capaces de generar peticiones sobre las especificaciones HTTP.

Un proceso de actualización permite contar con los datos más actuales o recientes. Es posible debido a las Tecnologías de la Información y las Comunicaciones (TIC) e Internet, las cuales brindan al hombre las vías necesarias para acceder a diversas fuentes de conocimientos y adquirir aquellos datos, archivos y herramientas que necesiten, independientemente de su localización geográfica o lenguaje.

En Cuba, a pesar de existir varias adversidades a la hora de acceder a tecnologías de punta para el desarrollo de sus telecomunicaciones e infraestructura, se han realizado diversos procesos, investigaciones e inversiones encaminados a lograr paulatinamente la informatización de sectores y áreas fundamentales del país mediante las tendencias informáticas más utilizadas en la actualidad como es, el intercambio de datos entre diversos sistemas. En estos procesos, la Empresa de Telecomunicaciones de Cuba S.A (ETECSA), principal responsable de la distribución, montaje y reparación de las redes e infraestructuras existentes en la nación y la Universidad de las Ciencias Informáticas (UCI) juegan un papel fundamental.

La UCI tiene como objetivo desde su surgimiento en el año 2002, introducir a Cuba en la industria del *software*, brindar asesoramiento y desarrollar soluciones informáticas para las diversas empresas nacionales e internacionales que así lo requieran. En la actualidad las áreas de la Universidad que contribuyen a los objetivos antes mencionados son los centros productivos. Uno de ellos es el Centro de Entornos Interactivos 3D, Vertex de la Facultad 4.

Dentro de las líneas de producción del centro Vertex se encuentra el diseño y desarrollo de videojuegos sobre entornos virtuales para diversas plataformas. Las recientes aplicaciones de entretenimiento lanzadas para los dispositivos Android, han contado con una amplia aceptación por parte del público cubano, a pesar de no poseer una vía de comunicación capaz de vincularlo a algún sistema o herramienta base para la actualización de sus contenidos. Tal situación:

- Imposibilita la actualización de los videojuegos Android desarrollados en el centro Vertex después de sus lanzamientos.
- Exige que las aplicaciones contengan toda la información del videojuego, para alargar su tiempo de vida útil en el mercado, sobresaturando la memoria del dispositivo móvil.
- Obliga a los especialistas del centro a relanzar continuamente nuevas versiones del sistema para mantenerlo actualizado, ocasionando un gasto de tiempo y recursos.

Teniendo en cuenta la situación problemática anteriormente expuesta, se plantea como **problema a resolver** ¿Cómo contribuir a la gestión de actualizaciones de los videojuegos Android mediante la red, desarrollados en el centro Vertex?

Para la realización de la investigación se define como **objeto de estudio** el proceso de transferencia de datos para la gestión de actualizaciones mediante la red.

Para dar solución al problema planteado, se define como **objetivo** desarrollar un sistema para la gestión de actualizaciones de videojuegos Android, implementados en el centro Vertex.

El **campo de acción** se centra en la gestión de actualizaciones mediante la transferencia de datos a través de la red en videojuegos Android.

Para dar cumplimiento al objetivo, se definen las siguientes **tareas de la investigación**:

- Elaboración del marco teórico de la investigación a través del estudio del estado del arte sobre los sistemas para la gestión de las actualizaciones de aplicaciones Android.

- Caracterización de los principales elementos que componen a los sistemas para la gestión de las actualizaciones de aplicaciones Android.
- Realización del levantamiento de requisitos funcionales y no funcionales del sistema para la gestión de las actualizaciones de aplicaciones Android.
- Selección de la metodología, herramientas y tecnología para el desarrollo del sistema para la gestión de las actualizaciones de aplicaciones Android.
- Generación de artefactos ingenieriles acordes a la metodología seleccionada para guiar el desarrollo del sistema para la gestión de las actualizaciones de aplicaciones Android.
- Implementación de un sistema para la gestión de las actualizaciones de videojuegos Android para dar solución al problema planteado.
- Validación de la propuesta de solución mediante las pruebas definidas en la investigación.

Métodos de investigación científica

Entre los **métodos teóricos** que se emplearán para la investigación se encuentran:

El método **Analítico-Sintético** para la búsqueda, análisis y síntesis de los elementos fundamentales de los sistemas para la gestión de las actualizaciones de aplicaciones Android, así como el de las estructuras pertinentes para el envío de datos a partir de ellas.

El método **Histórico-Lógico** para el estudio y análisis de la evolución de los sistemas para la gestión de las actualizaciones de aplicaciones Android. Permite determinar las principales herramientas y tecnologías a utilizar para dar solución al problema, así como las principales características con las que debe contar el mismo.

La **Modelación** para la búsqueda de las relaciones existentes entre cada uno de los componentes de los sistemas para la gestión de las actualizaciones de aplicaciones Android, apoyándose fundamentalmente en el uso de diagramas.

El método **Inductivo-Deductivo** para determinar una serie de elementos relacionados con los sistemas para la gestión de las actualizaciones de aplicaciones Android a desarrollar, para posteriormente arribar a razonamientos aplicables a la solución del problema.

Los **métodos empíricos** representan un nivel de la investigación cuyo contenido procede de la experiencia y es sometido a determinada elaboración racional; de ellos se emplearon los siguientes:

Observación: Se emplea durante la investigación para observar y estudiar el funcionamiento de los sistemas para la gestión de actualizaciones de videojuegos Android, así como las principales deficiencias y ventajas de su empleo. Permite alcanzar una mayor comprensión del problema, especifica el objeto de estudio y el campo de acción proporcionando un enfoque a la investigación hacia una posible solución.

Entrevista: Se emplea a través de conversaciones con especialistas y trabajadores del centro Vertex sobre diversos temas relacionados con la investigación. Permite obtener ideas, referencias e información útil para elaborar el levantamiento de requisitos del sistema.

El presente documento consta de tres capítulos, desarrollados a partir del estudio realizado. La descripción de los mismos se presenta a continuación:

Capítulo I. Fundamentación teórica: En este capítulo se estudia y analiza el estado del arte para definir las bases que dieron solución al problema existente. Se definen los conceptos fundamentales, se analizan las ventajas y desventajas de cada una de las herramientas, lenguajes, metodologías de desarrollo, técnicas e innovaciones que se utilizarán posteriormente en la solución informática propuesta.

Capítulo II. Características y diseño: En este capítulo se describen las características del sistema. Se define una propuesta de solución, se identifican los requerimientos y restricciones del mismo, se realizan los procesos de planificación y diseño siguiendo las pautas de la metodología Programación Extrema (XP). Finalmente se generan los artefactos necesarios para comenzar la fase de implementación.

Capítulo III. Implementación y pruebas: En este capítulo como parte de los procesos de implementación y pruebas propuestos por la metodología XP, se describen las tareas de programación y se realizan pruebas unitarias y de aceptación.

Capítulo I: Fundamentación teórica

1.1 Introducción

En este capítulo se realiza el estudio de los sistemas homólogos en el escenario nacional e internacional. Se analizan las ventajas y desventajas de las herramientas, tecnologías, lenguajes y metodologías a emplear durante el desarrollo del sistema. Se abordan los aspectos teóricos y conceptos fundamentales necesarios para dar solución al problema propuesto.

1.2 Conceptos fundamentales asociados a la investigación

En esta sección se explican algunos aspectos fundamentales para la comprensión del contexto de la investigación.

1.2.1 Servicio web

Un servicio web es una colección de procedimientos o métodos llamados desde cualquier lugar de Internet o la intranet. Es un mecanismo de invocación independiente de la plataforma utilizada y del lenguaje de programación en el que se haya implementado internamente el servicio (Dept. Ciencia de la Computación e IA, 2013).

- **Simple Object Access Protocol (SOAP):** es un protocolo basado en XML para el intercambio de información con reglas de codificación para instancias de tipos de datos; diseñado para procesamiento entre sistemas ligeramente acoplados sin manejo de referencias remotas (Sosa, 2015).

Características:

- ✓ Las respuestas son difíciles de interpretar.
- ✓ Una vez desarrollados los cambios en el servidor impactan de forma negativa en los clientes.
- ✓ Requiere de librerías para su interpretación de parte del cliente y de parte del servidor.

- **Transferencia de Estado Representacional (REST):** es una arquitectura que brinda un tipo de servicio que permite manejar los servicios web con métodos definidos, manteniendo la simpleza del protocolo como XML, cada servicio es identificado únicamente con un solo Identificador de Recursos Uniforme (URI por sus siglas en inglés).

Características:

- ✓ Es sencillo su desarrollo e interpretación.

- ✓ No se necesita mucho código extra.
- ✓ Tiene una arquitectura cliente-servidor.
- ✓ Es un sistema de capas e interfaz uniforme.

REST VS SOAP

Tabla 1. REST VS SOAP (Fuente: Elaboración propia)

REST VS SOAP	
Ventajas REST	Ventajas SOAP
Pocas operaciones con muchos recursos	Muchas operaciones con pocos recursos
Se centra en la escalabilidad y rendimiento a gran escala para sistemas distribuidos hipermedia	Se centra en el diseño de aplicaciones distribuidas
HTTP GET, HTTP POST, HTTP PUT, HTTP DEL	SMTP, HTTP POST, MQ
XML auto descriptivo	Tipado fuerte, XML <i>Schema</i>
Síncrono	Síncrono y Asíncrono
HTTPS	<i>WS SECURITY</i>
Comunicación punto a punto y segura	Comunicación origen a destino seguro

Selección de servicio web

Se decidió desarrollar un servicio web de Transferencia de Estado Representacional asociado al sistema que dará solución al problema informático presente ya que es una arquitectura simple y flexible. Además, es un servicio universal que permite acceder a los recursos de forma sencilla a través de URLs. No necesita configuraciones específicas ni la introducción de mucho código adicional. Su eficiente uso del ancho de banda y sus mejoras de rendimiento y escalabilidad permitirán al sistema atender eficientemente las peticiones de los usuarios que soliciten el servicio.

1.3 Estudios de los sistemas homólogos

A partir de la necesidad existente de desarrollar un sistema para la gestión de las actualizaciones de los videojuegos Android del centro Vertex, se realiza un estudio y análisis de las herramientas relacionadas con la presente investigación.

1.3.1 Estudios de los sistemas homólogos en el ámbito internacional

- **Google Play Store**

Es la tienda de aplicaciones creada por *Google*, donde el usuario puede centralizar su música, juegos, libros aplicaciones y contenidos multimedia favoritos y luego acceder a ellos desde otros dispositivos. Está basada en la tecnología “nube” permitiendo al usuario tener siempre disponible sus archivos. *Play Store* no almacena los archivos físicamente en el dispositivo sino en servidores manteniéndolos disponibles para el usuario en todo momento. *Google Play* es también un *Marketplace* donde los desarrolladores de aplicaciones para Android pueden ofrecer sus aplicaciones a cientos de millones de usuarios y potenciales clientes del planeta (Qode, 2013).

Play Store mantiene los destinos web y móvil sincronizados permitiendo que el contenido multimedia adquirido se encuentre inmediatamente disponible en el teléfono (o tableta) Android, sin consumir espacio de almacenamiento (Profis, 2012).

- ***F-Droid***

Es un repositorio alternativo a *Google Play* para aplicaciones Android así como un *kit* de tienda de aplicaciones completo, que proporciona todas las herramientas necesarias para configurar y ejecutar una tienda de aplicaciones. Sus funciones básicas permiten buscar aplicaciones, descargarlas, actualizarlas e informar de los cambios cuando hay una nueva versión. Todo el contenido que publica se basa en una licencia de código libre, que permite su consulta y modificación sin ningún tipo de restricción (González, 2017).

F-Droid no depende de una cuenta de *Google* para ser utilizado. Todo su contenido es gratuito y además, incluye algunas funciones muy interesantes y únicas, como por ejemplo poder instalar aplicaciones sin necesidad de tener conexión a Internet. También incluye herramientas completas de creación y lanzamiento para administrar el proceso de convertir el código fuente de la aplicación en compilaciones publicadas (*F-Droid*, 2018).

1.3.2 Estudios de los sistemas homólogos en el ámbito nacional

- **Centro de Aplicaciones Android**

El principal objetivo del Centro de Aplicaciones Android o *Apklis* es proveer un servicio oficial y confiable para la distribución de aplicaciones cubanas y foráneas para Android, potenciando así al desarrollo autóctono de las aplicaciones y proporcionando una vía para publicitar los mencionados productos. Con la distinción de proveer en mecanismo de actualización de las aplicaciones, permite la interacción de usuarios

y desarrolladores. Genera modelos de negocios favorables a todos los involucrados, utiliza medios de pago electrónico y permite modelar aplicaciones de acuerdo a los intereses y valores que inculca la Revolución Cubana. Apklis está enfocada a dos tipos de usuarios: desarrolladores que generan contenido y consumidores que son los usuarios consumidores del servicio (Apklis, 2018).

Apklis permite:

- ✓ Gestionar usuario
- ✓ Crear aplicaciones
- ✓ Descargar/Actualizar aplicaciones
- ✓ Filtrar aplicaciones por diversos criterios (búsqueda, autor, categoría)
- ✓ Gestionar Opiniones y evaluaciones

- **Android UCLV**

UCLV es un repositorio para aplicaciones Android desarrollado en la Universidad Central de las Villas. Tiene como objetivo poner al alcance de todos los usuarios aplicaciones útiles, novedosas y actualizadas; así como tonos, fondos y herramientas para móviles. El sistema creado es una aplicación con un ambiente de fácil uso que sigue los principales estilos de diseño especificados en las aplicaciones Android. Android UCLV se basa en el proyecto libre y de código abierto *F-Droid*. Sus principales funciones permiten buscar, descargar y actualizar aplicaciones (UCLV, 2018).

Resultados del estudio de los sistemas homólogos.

Tabla 2. Sistemas Homólogos (Fuente: Elaboración propia)

	<i>Google Play Store</i>	<i>F-Droid</i>	<i>Apklis</i>	<i>Android UCLV</i>
Open Source (Libre)	Si	Si	Si	Si
Subir aplicaciones (Orientado a cliente final)	Si	Si	No	No
Permite desarrollar y gestionar actualizaciones Android (Orientado a cliente final)	No	No	No	No

Luego del análisis y caracterización de los sistemas homólogos seleccionados, fue posible llegar a la conclusión que ninguno de ellos brinda una solución a la situación problemática. Estos sistemas a pesar

de contar con una política *Open Source* y permitir en algunos casos que los desarrolladores o clientes finales suban sus aplicaciones, no proporcionan un mecanismo mediante el cual los usuarios desarrollen y gestionen sus propias actualizaciones de videojuegos Android. Por todo lo antes planteado, se evidencia la necesidad de implementar una nueva solución para la gestión de las actualizaciones de los videojuegos Android del centro Vertex.

1.4 Herramientas, tecnologías y lenguajes

En este epígrafe se estudiarán y caracterizarán las herramientas, tecnologías y lenguajes a utilizar en el desarrollo de la propuesta de solución.

1.4.1 Lenguajes de Programación

- **Java**

Java es un lenguaje concurrente, basado en clases, y orientado a objetos, fue diseñado específicamente para tener pocas dependencias de implementación. Una de las características más importantes es que los programas ejecutables, creados por el compilador de Java, son independientes de la arquitectura. Es multiplataforma, un *software* de distribución libre, es completo y poderoso. Posee librerías y utilidades muy completas que facilitan la programación de aplicaciones (Gironés, 2013).

- **Procesador de Hipertexto**

El Procesador de Hipertexto (PHP por sus siglas en inglés) es un lenguaje interpretado de propósito general ampliamente usado. Está diseñado especialmente para el desarrollo web y puede ser embebido dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores y en todos los sistemas operativos y plataformas sin costo alguno (Bravo, 2009).

- **Lenguaje de Marcas de Hipertexto**

El Lenguaje de Marcas de Hipertexto (HTML por sus siglas en inglés) es un lenguaje de marcado o de marcas. Es una forma de codificar un documento que, junto con el texto, incorpora etiquetas o marcas que contienen información adicional acerca de la estructura del texto o su presentación (Flores, 2015). Es el lenguaje utilizado para la creación de páginas web, a pesar de no ser considerado un lenguaje de programación debido a la falta de variables o estructuras en él. Es la forma recomendada a la hora de definir la estructura y parte de la apariencia de las páginas web y su contenido mediante la incorporación de texto, imágenes y etiquetas.

- **Hojas de Estilo en Cascada**

Las Hojas de Estilo en Cascada 3 (CSS3 por sus siglas en inglés) es la nueva versión de las Hojas de Estilo en Cascada (CSS por sus siglas en inglés), viene con múltiples mejoras y nuevas características. A diferencia de HTML5 y CSS no elimina ni modifica la especificación de las antiguas versiones, de modo que todo lo ya existente sigue siendo perfectamente válido en esta nueva versión. CSS3 trae mejoras en aquellos aspectos que CSS2 no cubría o que le faltaba utilidad, proporcionando así un mayor control al desarrollador sobre la presentación. De este modo, características que antes no se podían implementar directamente, ya que se debía acudir a lenguajes externos como *JavaScript*, ahora definiendo una única característica, se pueden obtener los mismos resultados (Huguet, 2012).

- ***JavaScript***

JavaScript es un lenguaje de *scripts* desarrollado por *Netscape* para incrementar las funcionalidades del lenguaje HTML. Dentro de sus características más importantes se pueden encontrar que es un lenguaje interpretado, es decir, no requiere compilación pues el navegador se encarga de interpretar las sentencias *JavaScript* contenidas en una página HTML y ejecutarlas adecuadamente. Es un lenguaje orientado a eventos a través de *scripts* que se ejecutan como respuesta de determinados eventos. Su modelo de objetos está reducido y simplificado, pero incluye los elementos necesarios para que los *scripts* puedan acceder a la información de una página y puedan actuar sobre la interfaz del navegador (Alvarez, 2009).

Selección de los lenguajes de programación

Fueron seleccionados los lenguajes HTML5, CSS3 y *JavaScript* pues poseen importantes mejoras para el desarrollo de interfaces web. Además, el empleo de CSS y JavaScript son fundamentales para de incorporar a la parte visual del sistema los complementos necesarios para lograr el desarrollo de un entorno atractivo para el usuario. También fue seleccionado el lenguaje Java 8.0 para el desarrollo de las funcionalidades del sistema debido a que es de código abierto y permite el desarrollo de aplicaciones potentes, seguras y estables.

1.4.2 Entornos de desarrollo integrado

Un Entorno de Desarrollo Integrado (IDE, por sus siglas en inglés) es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDE proveen un marco de trabajo para la mayoría de los lenguajes de programación tales como C++, PHP, Python, Java, C#, Delphi, Visual

Basic, y otros. En algunos lenguajes, un IDE puede funcionar como un sistema en tiempo de ejecución, en donde se permite utilizar el lenguaje de programación en forma interactiva, sin necesidad de trabajo orientado a archivos de texto (Fergarcia, 2013).

- **Android Studio**

Android Studio es un entorno de desarrollo basado en *IntelliJ IDEA*. Posee distintos componentes que ayudan en la tarea de la construcción de aplicaciones; sistema de construcción basado en *Gradle*, la construcción de variantes y múltiples archivos de Aplicación Empaquetada de Android (APK), como también plantillas de código. Contiene una interfaz de usuario que es construida o diseñada previamente, con variados modelos de pantalla, donde los elementos existentes pueden ser desplazados. Adicionalmente se abarcan depuradores para emuladores y la posibilidad de trabajo con *Logcat*. Herramienta que mejora el rendimiento, facilita el uso y no producen problemas de compatibilidad con las diferentes versiones. Facilita la reutilización de código, de recursos, posibilitando además configurar, extender y personalizar el proceso (Ibarreche, 2010).

- **Eclipse**

Eclipse es una plataforma de desarrollo de código abierto, diseñado para ser extendido a través de *plugins*. Es un entorno de desarrollo multiplataforma para crear, integrar e implementar herramientas de desarrollo de aplicaciones. No tiene un lenguaje específico, sino que es un IDE genérico. Goza de una amplia popularidad entre la comunidad de desarrolladores del lenguaje Java. Es un entorno de desarrollo en el que se localizan todas las herramientas y funciones necesarias para el trabajo, recogidas además en una atractiva interfaz que lo hace fácil y agradable de usar (The Eclipse Foundation, 2016).

- **NetBeans**

NetBeans es una herramienta utilizada por programadores, pensada para escribir, compilar, depurar y ejecutar programas. Esta herramienta está codificada en el lenguaje de programación Java y permite desarrollar aplicaciones web, de escritorio y móviles usando las plataformas de Java. Es libre y gratuito tanto para uso comercial como no comercial, sin restricciones para su uso. Está soportado por una activa comunidad de desarrolladores que ofrece una amplia documentación y recursos de entrenamiento. Es multiplataforma, disponible para diversos sistemas operativos como *Windows*, *MacOS* y *GNU Linux*. Su instalación y actualización es muy simple y una vez instalado se le pueden adicionar módulos que permiten extender sus funcionalidades (Netbeans, 2017).

Resultado del estudio de los entornos de desarrollo integrados

Se seleccionó *NetBeans* como entorno de desarrollo, ya que además de poseer las propiedades presentadas anteriormente, brinda soporte para la creación de servicios web REST y cuenta con una interfaz para comprobar el correcto funcionamiento del mismo. Por otra parte, presenta: un buen editor de texto, asistentes de ayuda y un depurador de código, los mismos facilitan los procesos de desarrollo de aplicaciones. Además, *Netbeans* presenta condiciones favorables para el desarrollo de aplicaciones con el lenguaje de programación Java.

1.4.3 Herramienta de modelado

- ***Visual Paradigm***

Visual Paradigm es una herramienta CASE multiplataforma, que soporta el ciclo completo de desarrollo de *software*: análisis, diseño, implementación y pruebas. Facilita la construcción de aplicaciones informáticas con un menor coste que destacan por su alta calidad y contribuye a mejorar la experiencia del usuario mediante el diseño de un gran número de artefactos de ingeniería de *software*. Permite la generación de bases de datos, conversión de diagramas entidad-relación a tablas de base de datos, mapeos de objetos y relaciones, ingeniería directa e inversa, la gestión de requisitos de *software* y la modelación de procesos del negocio (*Visual Paradigm*, 2014). *Visual Paradigm* permite la generación de código, documentación y la integración con el entorno de desarrollo *NetBeans*. Además, brinda un conjunto de ayudas para los procesos de análisis, diseño y desarrollo de aplicaciones informáticas.

1.4.4 Lenguaje de modelado

- **Lenguaje de Modelado Unificado**

El Lenguaje Unificado de Modelado (UML, por sus siglas en inglés) es el lenguaje gráfico que es utilizado para facilitar el desarrollo de *software*, proporciona un vocabulario, reglas y un lenguaje de modelado, todo para lograr una representación conceptual y física del sistema. Este lenguaje ayuda a interpretar grandes sistemas utilizando gráficos o textos obtenidos de modelos explícitos, los cuales ayudan a la comunicación durante el desarrollo de herramientas informáticas (UNID, 2015). Este lenguaje permite establecer los requerimientos y estructuras necesarias para plasmar el sistema de *software*.

1.4.5 Sistemas Gestores de Base de Datos

- ***PostgreSQL***

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es un sistema de gestión de bases de datos de código abierto potente. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multi-hilos para garantizar la estabilidad del sistema. Es compatible con el almacenamiento de objetos binarios de gran tamaño, como imágenes, sonidos o video. Cuenta con interfaces de programación nativas para C/ C++, Java, .NET, Python, Ruby, Tcl, Perl, ODBC, entre otros. Cuenta con características sofisticadas como la replicación asincrónica y sincrónica, transacciones anidadas, *tablespaces*, copias de seguridad en caliente, múltiples métodos de autenticación, MVCC (Control de Concurrencia Multi-Versión) y un sofisticado planificador de consultas optimizado (PostgreSQL, 2016).

- **MySQL**

MySQL es una base de datos robusta que puede ser comparada con una base de datos comercial. Compite con sistemas RDBMS propietarios como *Oracle*, *SQL Server* y *DB2*, disponiendo de procesamiento de transacciones a través del motor de almacenamiento *InnoDB*, dispone de procedimientos almacenados, *triggers*, y vistas. Es capaz de desarrollar sus propias aplicaciones de bases de datos en la mayor parte de lenguajes de programación utilizados en la actualidad y ejecutarlos en casi todos los sistemas operativos. *MySQL* es lo suficientemente flexible para trabajar en entornos con gran demanda, tales como aplicaciones web. Puede impulsar aplicaciones empotradas, almacenes de datos, indexación de contenidos, sistemas de mensajería, sistemas redundantes de alta disponibilidad y procesamiento de transacciones en línea (Cevallos, 2014).

Sistema gestor de base de datos seleccionado

Se seleccionó *PostgreSQL* ya que es ideal para el desarrollo de proyectos extensos que continuamente realizan peticiones a la base de datos, permite realizar consultas complejas e implementa funciones de escritura y lectura en paralelo. Además, actualmente la Universidad lleva a cabo en sus proyectos productivos un proceso de migración hacia *PostgreSQL*.

1.4.6 Servidor Web

- **Apache**

Apache es el servidor web por excelencia, su configurabilidad, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa. La licencia *Apache* es una descendiente de la licencia BSD. El Servidor *Apache* HTTP es un servidor web de tecnología *Open Source* sólido. (Ciberaula, 2010).

A continuación se muestran algunas características de *Apache*:

- ✓ Corre en una multitud de sistemas operativos, lo que lo hace prácticamente universal.
- ✓ Trabaja con varios lenguajes de *script* tales como *Perl*, *PHP* y otros, además funciona con *Java* y páginas *Java Server Pages*.
- ✓ Permite personalizar las respuestas ante los posibles errores que puedan ocurrir en el servidor.
- ✓ *Apache* es un servidor altamente configurable de diseño modular.

- ***Glassfish***

Glassfish es un servidor de aplicaciones de *software* libre, que implementa las tecnologías definidas en la plataforma *Java EE* y permite ejecutar aplicaciones que siguen esta especificación. Es gratuito, de código libre y se distribuye bajo la licencia dual a través de la *CDDL* y *GNU GPL*. Permite subdividir el servidor de aplicaciones en dominios que sirven para agrupar aplicaciones que pueden compartirse. Además de ser un servidor de aplicaciones, es una comunidad de usuarios, que lo descargan y utilizan libremente (Vergara, 2016).

Servidor web seleccionado

Se seleccionó *Glassfish* ya que es un servidor de aplicaciones *Java* que admite lenguajes *script* como: *PHP*, *Ruby* y *JavaScript*. Además, soporta e implementa especificaciones como *Java Server Faces*, *Java Server Pages* y *Servlets* los cuales son necesarios para el desarrollo de aplicaciones web.

1.4.7 Metodología de desarrollo de *software*

La metodología para el desarrollo de *software* es un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito. Una metodología para el desarrollo de *software* comprende los procesos a seguir sistemáticamente para idear, implementar y mantener un producto de *software* desde que surge la necesidad del mismo hasta que se cumple el objetivo por el cual fue creado (Maida, y otros, 2015).

Las metodologías tradicionales imponen una disciplina de trabajo sobre el proceso de desarrollo del *software*, con el fin de conseguir un *software* más eficiente. Para ello, se hace énfasis en la planificación total de todo el trabajo a realizar y una vez que está todo detallado, comienza el ciclo de desarrollo del producto. Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas, notaciones y documentación detallada. Además, las metodologías

tradicionales no se adaptan adecuadamente a los cambios, por lo que no son métodos adecuados cuando se trabaja en un entorno, donde los requisitos no pueden predecirse o bien pueden variar (Condori, 2010).

Las metodologías ágiles son sistemas de gestión de proyectos que ayudan a usar el tiempo de manera efectiva y creativa. Visualiza de manera clara las tareas a realizar. Gracias a la organización, incrementan la calidad del trabajo y disminuyen de manera considerable el tiempo invertido en cada tarea. Mejoran la comunicación interna del equipo. Generan espacios propicios para el trabajo y la creatividad grupal. Son muy útiles para visualizar, organizar las tareas a realizar y para mejorar el rendimiento y el trabajo en equipo. Permiten tener un seguimiento detallado de cada etapa de un proyecto, tanto a nivel personal como grupal (Wingu, 2016).

Las diferencias entre las Metodologías Ágiles y las Metodologías Tradicionales se muestran en la siguiente tabla.

Tabla 3. Metodologías (Fuente: Elaboración propia)

Metodologías Ágiles	Metodologías Tradicionales
Especialmente preparadas para cambios durante el proyecto	Cierta resistencia a los cambios
Proceso menos controlado, con pocos principios	Proceso controlado, con numerosas políticas y normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
Pocos roles	Muchos roles
Pocos artefactos	Muchos artefactos
Grupos pequeños	Grupos grandes y distribuidos
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones

Luego del estudio realizado se decidió emplear una metodología ágil para el desarrollo de la herramienta. Entre las más conocidas se encuentran Programación Extrema (XP por sus siglas en inglés), SCRUM, AUP entre otras.

- **SCRUM**

La metodología SCRUM define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. El

desarrollo de *software* se realiza mediante iteraciones, denominadas *Sprint*, con una duración de 30 días. El resultado de cada *Sprint* es un incremento ejecutable que se muestra al cliente. El equipo de desarrollo realiza reuniones diarias de 15 minutos para la coordinación e integración (Castellanos, 2015).

- **AUP**

AUP es una metodología que tiene la adopción de muchas de las técnicas ágiles de la metodología XP y de las formalidades de RUP. Tiene como filosofía adaptarse a las necesidades del proyecto y no al contrario como lo planteado en las metodologías tradicionales. Esta metodología, plantea un ciclo de vida iterativo, que se basa en la ampliación y refinamiento sucesivo del sistema, mediante múltiples iteraciones con retroalimentación cíclica (Mori, 2010).

- **XP**

XP es una de las llamadas metodologías ágiles de desarrollo de *software* más exitosas. La metodología propuesta en XP está diseñada para entregar el *software* que los clientes necesitan en el momento en que lo necesitan. XP alienta a los desarrolladores a responder a los requerimientos cambiantes de los clientes, aún en fases tardías del ciclo de vida del desarrollo. Enfatiza el trabajo en equipo, tanto gerentes como clientes y desarrolladores son partes del mismo equipo dedicado a entregar un *software* de calidad (Joskowicz, 2008).

Valores de la metodología XP (Rodríguez, 2014):

- ✓ Simplicidad
- ✓ Comunicación
- ✓ Retroalimentación
- ✓ Coraje o Valentía

Las fases de la metodología XP se muestran en la siguiente tabla.

Tabla 4. Metodología XP (Fuente: Elaboración propia)

Fases Metodología XP	
Fase	Tarea
Planificación del proyecto	Historias de usuario
Diseño	Tarjetas (clase-responsabilidad-colaborador)
Codificación	Tareas de programación
Pruebas	Pruebas Unitarias y de Aceptación

Metodología de desarrollo de software seleccionada

Como metodología de desarrollo de *software* se selecciona XP. Esta permite desarrollar aplicaciones propensas a sufrir continuos cambios en su estructura. Es ideal su uso por su facilidad y simpleza a la hora del desarrollo de *software* y por el ambiente de comunicación que genera entre el cliente y los desarrolladores. Además, es la que mejor se adapta a las condiciones de trabajo y al tiempo de desarrollo disponible para implementar la solución propuesta.

1.5 Conclusiones parciales

- El estudio de los sistemas homólogos nacionales e internacionales existentes relacionados con la presente investigación evidenció la necesidad de desarrollar una herramienta para la gestión de las actualizaciones de los videojuegos Android del centro Vertex.
- El análisis y caracterización de las herramientas y tecnologías informáticas existentes permitió definir la base tecnológica en la que estará sustentada la investigación. Se seleccionó XP como metodología de desarrollo de *software* y como lenguajes de programación Java 8.0, HTML5 y CSS3. Se escogió como gestor de base de datos *PostgreSQL* 9.4, como servidor web *Glassfish*. Se definió UML como lenguaje de representación visual y *Visual Paradigm* 8.0 como herramienta *CASE* para el modelado del sistema.

Capítulo II: Características y diseño

2.1 Introducción

En este capítulo se presenta la propuesta de solución del sistema a desarrollar, siguiendo una serie de pasos y procesos con el fin de lograr cumplir con las expectativas del cliente. Se realiza el levantamiento de los requisitos funcionales y no funcionales del sistema. Se definen los patrones de diseño y arquitectónicos de la herramienta. Como propuesta de la metodología XP, se realizan las historias de usuario para la descripción de los requerimientos del sistema y las tarjetas CRC así como, los planes de entrega y de duración de las iteraciones.

2.2 Propuesta de Solución

Se desarrollará un sistema para la gestión de actualizaciones de videojuegos Android. La misma contará con diversas interfaces para: la validación de credenciales y gestión de actualizaciones formuladas a partir de modelos de datos o *plugins* predefinidos en un ambiente externo. El envío de las nuevas versiones hacia los videojuegos Android se realizará a través de una API REST asociada a la herramienta, en formato JSON mediante un servicio REST. Los datos generados podrán ser almacenados previamente en la base de datos del sistema e interpretados posteriormente por las aplicaciones portátiles que los hayan recibido.

Interfaces del sistema

El sistema contará con una serie de interfaces, con las que el usuario podrá interactuar y llevar a cabo las acciones para las que fue diseñada la herramienta.

- *Index*

La página *Index* permitirá al usuario ingresar sus credenciales en los campos usuario y contraseña para su posterior autenticación, garantizando que solo el administrador pueda acceder al sistema. En esta sesión también se muestran enlaces para descargar aplicaciones Android desarrolladas en el centro Vertex.



Ilustración 1. Prototipo Autenticación Interfaz (Fuente: Elaboración propia)

- Principal

La herramienta contará con una interfaz principal para la visualización de las actualizaciones generadas hasta el momento en el sistema. Cada versión se mostrará en forma de imagen con un nombre en la parte posterior de la misma.

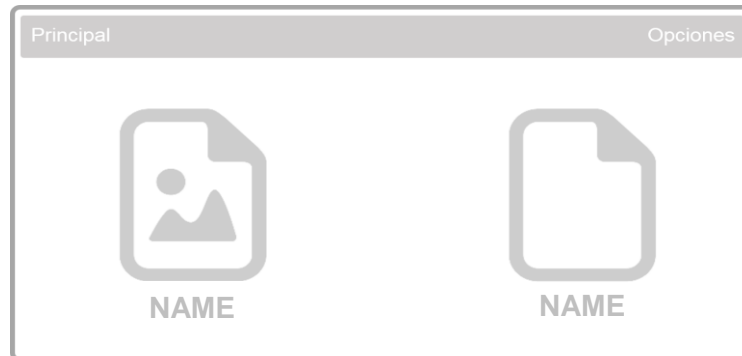


Ilustración 2. Prototipo Principal Interfaz (Fuente: Elaboración propia)

- **Actualizaciones**

El sistema tendrá dos sesiones para la gestión de las actualizaciones de videojuegos. En ambas el usuario interactuará elaborando un modelo de datos compuesto por textos y direcciones URL vinculadas a imágenes o archivos según el tipo de actualización que se desee generar.

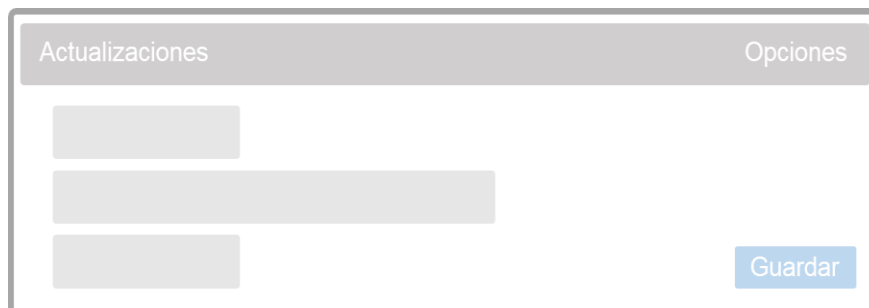


Ilustración 3. Prototipo Actualizaciones Interfaz (Fuente: Elaboración propia)

Funcionalidades del sistema

Las principales funcionalidades de la herramienta serán:

- Gestionar dos tipos de modelos de datos compuestos por:
 1. El nombre del paquete, su versión, la fecha en que fue generado y el vínculo de un archivo comprimido en formato ZIP o RAR previamente elaborado en un ambiente externo, para la generación de actualizaciones a partir de un *plugin*.
 2. Un nombre de actualización, una descripción y el vínculo de una imagen, estas dos últimas serán opcionales. Cada actualización tendrá un grupo de preguntas de las que se conoce la pregunta, su dificultad y cuatro posibles respuestas entre las que estará la correcta.

- Referenciar, acceder y almacenar las direcciones de las imágenes y archivos en formato ZIP o RAR utilizadas en los proceso de gestión de las actualizaciones.

Servicio

La herramienta tendrá vinculada una API REST para la creación de servicios web para la obtención de los datos de las actualizaciones generadas. Las nuevas versiones podrán ser descargadas desde los dispositivos móviles a través de la red, serán enviadas en formato JSON.

2.3 Levantamiento de Requisitos Funcionales y Requisitos no Funcionales

El levantamiento de requisitos o requerimientos es la captura de las características que determinará lo que hará el sistema, definiendo las restricciones de sus operaciones e implementación. Este proceso permite establecer una relación e identificación de las necesidades del usuario, además de describir las condiciones necesarias para que los requisitos puedan cumplirse.

Los requerimientos funcionales son declaraciones de los servicios que el sistema debe brindar.

Tabla 5. Requisitos Funcionales (Fuente: Elaboración propia)

Requisitos Funcionales		
Código	Nombre	Prioridad
RF.1	Autenticar usuario	Alta
RF.2	Modificar credenciales	Alta
RF.3	Crear actualización de videojuego Android a partir de texto	Alta
RF.4	Mostrar listado de actualizaciones de videojuegos Android	Media
RF.5	Modificar actualización de videojuego Android a partir de texto	Alta
RF.6	Seleccionar actualización de videojuego Android	Media
RF.7	Eliminar actualización de videojuego Android a partir de texto	Alta
RF.8	Generar actualización de videojuego Android a partir de texto	Alta
RF.9	Cargar imagen principal de actualización de videojuego Android a partir de texto	Media
RF.10	Mostrar listados de preguntas de actualización de videojuego Android a partir de texto	Media
RF.11	Crear pregunta de actualización de videojuego Android a partir de texto	Alta
RF.12	Modificar pregunta de actualización de videojuego Android a partir de texto	Alta

RF.13	Eliminar pregunta de actualización de videojuego Android a partir de texto	Alta
RF.14	Crear actualización de videojuego Android a partir de <i>plugin</i>	Alta
RF.15	Modificar actualización de videojuego Android a partir de <i>plugin</i>	Alta
RF.16	Eliminar actualización de videojuego Android a partir de <i>plugin</i>	Alta
RF.17	Cargar archivo de actualización de videojuego Android a partir de <i>plugin</i>	Media

Durante el proceso de definición de los requisitos funcionales se estableció su prioridad para el sistema mediante el uso de una matriz de Importancia-Urgencia ([Ver Anexo 1](#)).

Los requerimientos no funcionales se refieren directamente a las funciones o propiedades emergentes como la fiabilidad, tiempo de respuesta o la capacidad de almacenamiento que debe poseer el sistema.

Tabla 6. Requisitos No Funcionales (Fuente: Elaboración propia)

Requisitos No Funcionales	
	Usabilidad
RNF.1	El sistema debe poseer interfaces amigables, con botones que tengan nombres sugerentes para que usuarios inexpertos puedan interactuar fácilmente con el <i>software</i>
RNF.2	El usuario final deberá conocer y saber elaborar el modelo de datos de las actualizaciones
	Eficiencia
RNF.3	El sistema deberá responder a las peticiones del usuario en un tiempo menor a 1000 mili-segundos
RNF.4	El sistema debe generar las actualizaciones en un tiempo menor o igual a un 10000 mili-segundos
	Hardware
RNF.5	<ul style="list-style-type: none"> • 1GB de RAM o superior • La PC debe contar con un CPU de 256 MHZ o superior
	Software
RNF.6	La PC donde se desea trabajar con el herramienta debe contar con el Sistema Operativo (SO) Windows 7 o superior
	Diseño e Implementación
RNF.7	Como lenguaje de programación para el desarrollo del sistema se utilizará Java 8.0
RNF.8	Como gestor de base de datos se utilizará <i>PostgreSQL</i>

RNF.9	Como herramienta para el modelado y diseño del sistema se utilizará <i>Visual Paradigm</i>
RNF.10	Como IDE para el desarrollo del sistema de utilizará <i>NetBeans</i>
	Apariencia o Interfaz Interna
RNF.11	Las interfaces del sistema contarán con un predominio de los colores azul, gris, blanco y negro. Tendrá un diseño sencillo
RNF.12	Las interfaces del sistema contarán con los componentes visuales necesarios para realizar las funciones definidas por el cliente
	Seguridad
RNF.13	La aplicación contará con un sistema de autenticación con el objetivo de garantizar que solamente el administrador tenga acceso a la herramienta
RNF.14	La información solo podrá ser creada o modificada por el administrador del sistema

2.4 Planificación

En la etapa de planificación el cliente especifica sus necesidades mediante la redacción de sencillas historias de usuario para la estimación del tiempo de desarrollo. En esta, las estimaciones realizadas son primarias, ya que están basadas en datos de muy alto nivel que podrían variar al ser analizadas con más detalle en otras iteraciones. Esta fase dura dos semanas y el resultado es una visión general del sistema y un plazo total estimado de su desarrollo (Garcia, y otros, 2013).

2.4.1 Historias de Usuario

Las Historias de Usuario (HU) representan una breve descripción del comportamiento del sistema, se realizan por cada característica principal del sistema y son utilizadas para cumplir estimaciones de tiempo y el plan de lanzamientos, así mismo reemplazan un gran documento de requisitos y presiden la creación de las pruebas de aceptación. Cada HU debe ser lo suficientemente comprensible y delimitada para que los programadores puedan implementarlas en unas semanas (Meléndez, 2018).

Tabla 7. HU_1 (Fuente: Elaboración propia)

Historia de Usuario	
Número: HU_1	Nombre: Autenticar usuario
Usuario: Administrador	

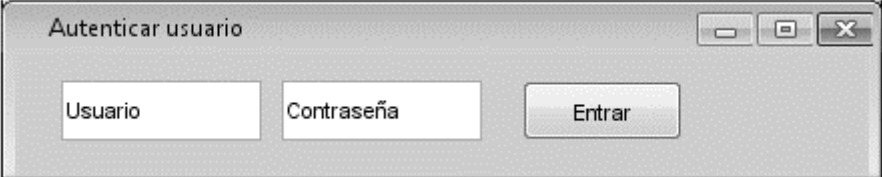
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1.1	Iteración Asignada: 1
Programador responsable: René Orta Martínez	
<p>Descripción: Inicia cuando el usuario decide acceder al sistema. Se muestra una interfaz con los campos: Usuario y Contraseña y el botón Entrar.</p> <p>Al introducir los datos y presionar el botón, se verifica la autenticidad de los mismos y el usuario accede al sistema</p>	
<p>Observaciones:</p> <ul style="list-style-type: none"> Los campos Usuario y Contraseña no pueden ser vacíos 	
<p>Prototipo de interfaz:</p> 	

Tabla 8. HU_2 (Fuente: Elaboración propia)

Historia de Usuario	
Número: HU_2	Nombre: Modificar credenciales
Usuario: Administrador	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1.3	Iteración Asignada: 1
Programador responsable: René Orta Martínez	
<p>Descripción: Inicia cuando el usuario decide modificar sus credenciales y presiona el ícono con la imagen de opciones en la parte superior derecha de la pantalla. Se muestra una interfaz con las opciones Cambiar Contraseña y Cambiar Usuario; en la primera opción se muestran los campos: Contraseña, Nueva Contraseña y Repita Nueva Contraseña y en la segunda: Contraseña y Nuevo Usuario. Ambas opciones cuentan con los botones</p>	

Cancelar para abandonar la sesión y Guardar Cambios para salvar las modificaciones realizadas a las credenciales siempre y cuando los datos ingresados sean correctos y hayan sido autenticados

Observaciones:

- Los campos no pueden ser vacíos y la nueva contraseña deberá contar con 8 caracteres y algún signo o número

Prototipo de interfaz:

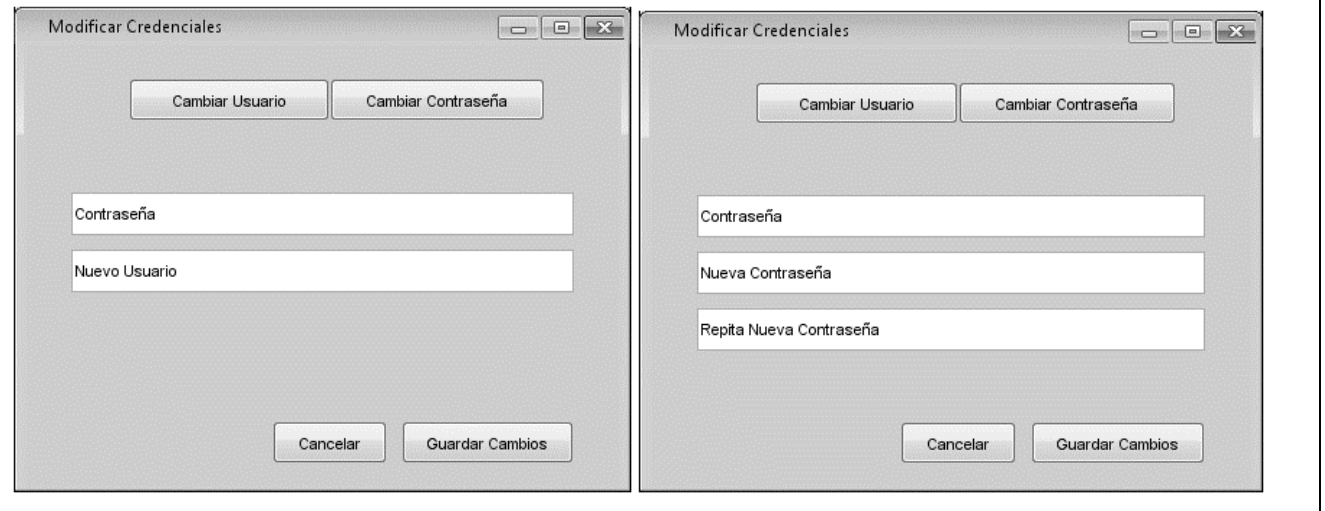


Tabla 9. HU_4 (Fuente: Elaboración propia)

Historia de Usuario	
Número: HU_4	Nombre: Mostrar listado de actualizaciones de videojuegos Android
Usuario: Administrador	
Prioridad en Negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración Asignada: 1
Programador responsable: René Orta Martínez	
Descripción: Inicia cuando el usuario se encuentra en la página Actualizaciones o Principal del sistema. Se muestran las actualizaciones Android existentes	
Observaciones:	
Prototipo de interfaz:	

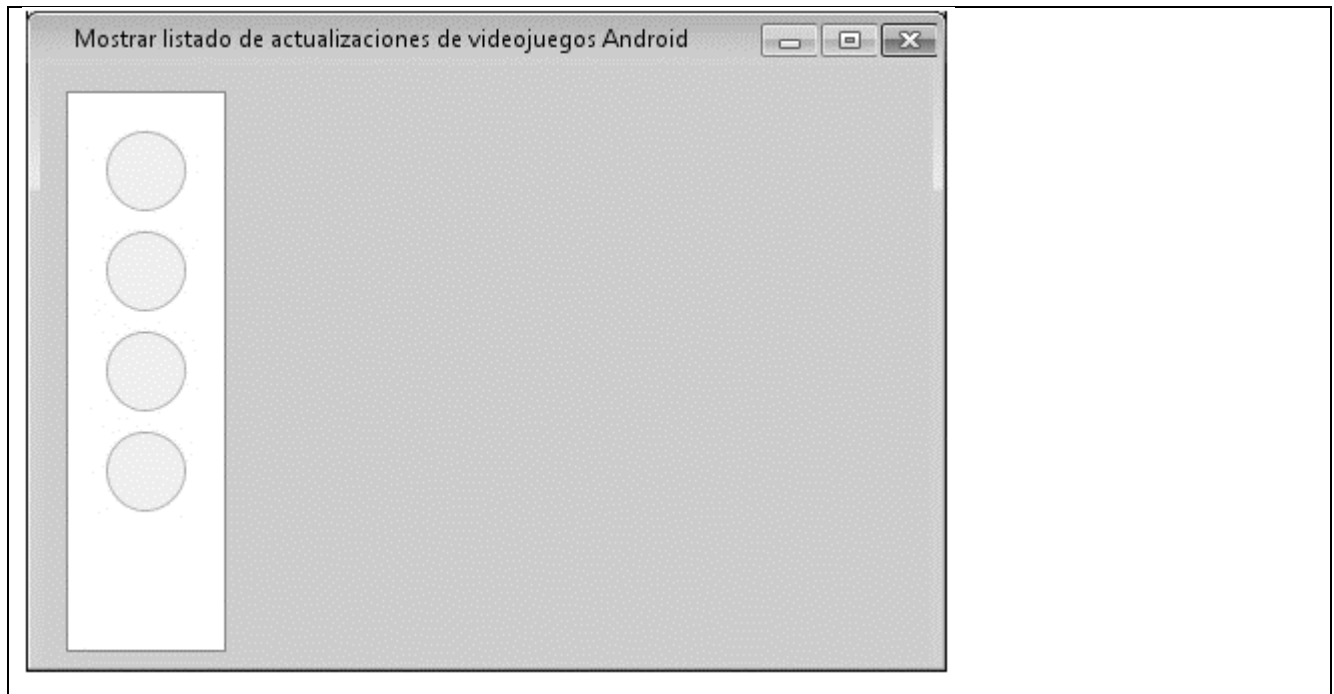


Tabla 10. HU_6 (Fuente: Elaboración propia)

Historia de Usuario	
Número: HU_6	Nombre: Seleccionar actualización de videojuego Android
Usuario: Administrador	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración Asignada: 2
Programador responsable: René Orta Martínez	
Descripción: Inicia cuando el usuario presiona el botón con el ícono de Eliminar modificando parcialmente la interfaz. Se muestra junto a cada actualización de videojuego Android: un <i>checkbox</i> y los botones: Seleccionar Todo para chequear todos los <i>checkbox</i> y Eliminar para remover las actualizaciones de los videojuegos Android del sistema	
Observaciones:	
Prototipo de interfaz:	

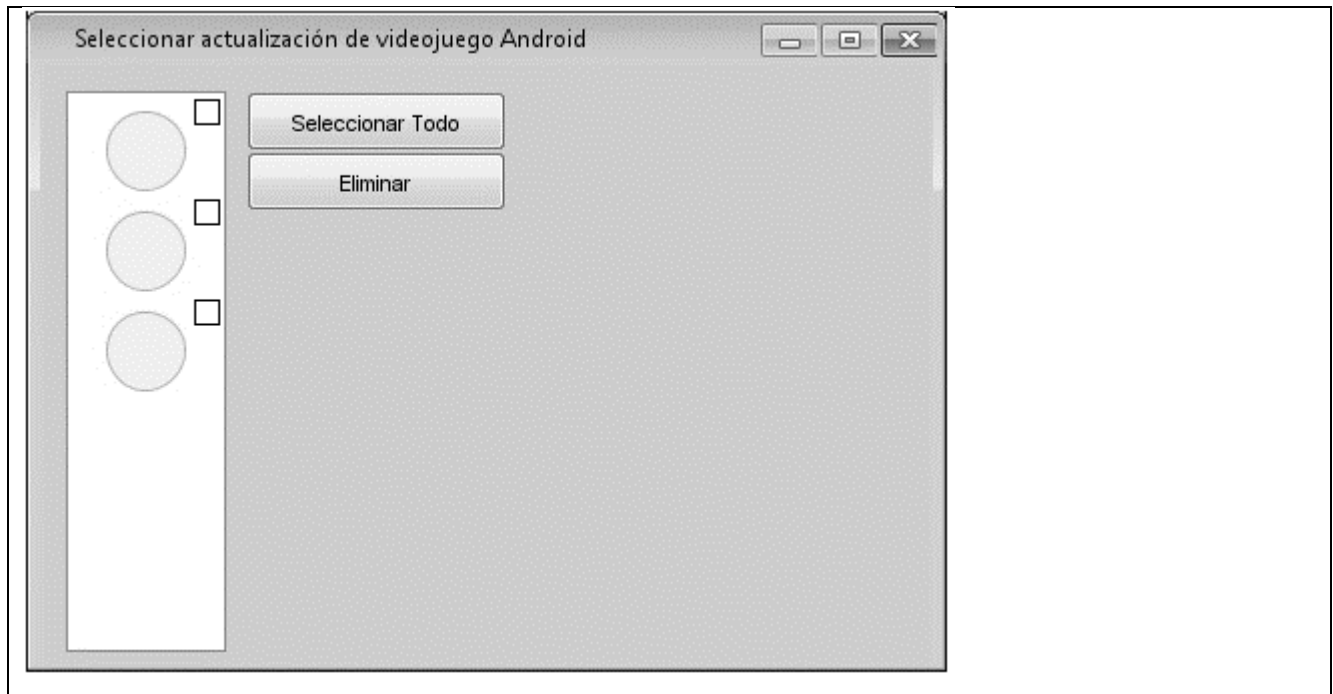


Tabla 11. HU_7 (Fuente: Elaboración propia)

Historia de Usuario	
Número: HU_7	Nombre: Eliminar actualización de videojuego Android a partir de texto
Usuario: Administrador	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1.1	Iteración Asignada: 2
Programador responsable: René Orta Martínez	
Descripción: Inicia al presionar el botón Eliminar, a continuación surge una nueva interfaz con un mensaje y los botones Cancelar para suspender el proceso de eliminación de la actualización del videojuego Android a partir de texto y Aceptar para aprobarlo	
Observaciones:	
<ul style="list-style-type: none"> • Previamente se debe seleccionar alguna actualización de videojuego Android a partir de texto 	

Prototipo de interfaz:



Tabla 12. HU_9 (Fuente: Elaboración propia)

Historia de Usuario	
Número: HU_9	Nombre: Cargar imagen principal de actualización de videojuego Android a partir de texto
Usuario: Administrador	
Prioridad en Negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración Asignada: 2
Programador responsable: René Orta Martínez	
Descripción: Inicia cuando el usuario presiona el botón Buscar durante el proceso de creación o modificación de las actualizaciones de videojuegos Android a partir de texto. Se muestra un explorador donde el usuario selecciona la imagen que desea adjuntar a la actualización del videojuego	
Observaciones:	
Prototipo de interfaz:	



Tabla 13. HU_14 (Fuente: Elaboración propia)

Historia de Usuario	
Número: HU_14	Nombre: Crear actualización de videojuego Android a partir de <i>plugin</i>
Usuario: Administrador	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1.3	Iteración Asignada: 3
Programador responsable: René Orta Martínez	
<p>Descripción: Inicia cuando el usuario decide crear una actualización de videojuego Android a partir de <i>plugin</i>. Se muestra una interfaz con los campos: Nombre Paquete, Versión y Fecha y los botones: Cancelar para abandonar la sección, Buscar para cargar la dirección del paquete de la actualización del videojuego y Guardar para crear la actualización del videojuego siempre y cuando los datos ingresados sean correctos</p>	
<p>Observaciones:</p> <ul style="list-style-type: none"> • Los campos no pueden ser vacíos ni contener caracteres reservados del lenguaje de programación Java 	

- Al presionarse el botón Guardar surge una nueva interfaz con un mensaje y los botones: Cancelar para suspender el proceso de creación de la actualización y Aceptar para aprobarlo

Prototipo de interfaz:

Crear actualización de videojuego Android a partir de plugin

Nombre Paquete	Fecha	Versión
Nombre Paquete	Fecha	Versión

Buscar Guardar Cancelar

Crear actualización de videojuego Android a partir de plugin

¿ Está seguro que desea crear la actualización del videojuego Android a partir de plugin ?

Aceptar Cancelar

El resto de historias de usuario se encuentran en los Anexos ([Ver Anexo 1](#)).

2.4.2 Estimación de esfuerzo por historia de usuario

La estimación de esfuerzo por historias de usuario la realiza el equipo de desarrollo. Se establecen por cada HU una serie de puntos que equivalen a la cantidad de semanas de programación necesarias para lograr la implementación de la funcionalidad.

Tabla 14. Estimación de Esfuerzo por Historia de Usuario (Fuente: Elaboración propia)

No.	Historia de usuario	Puntos Estimados (Semanas)
1	Autenticar usuario	1.1
2	Modificar credenciales	1.3
3	Crear actualización de videojuego Android a partir de texto	1.2

4	Mostrar listado de actualizaciones de videojuegos Android	1
5	Modificar actualización de videojuego Android a partir de texto	1.2
6	Seleccionar actualización de videojuego Android	1
7	Eliminar actualización de videojuego Android a partir de texto	1.1
8	Generar actualización de videojuego Android a partir de texto	1.5
9	Cargar imagen principal de actualización de videojuego Android a partir de texto	1
10	Mostrar listados de preguntas de actualización de videojuego Android a partir de texto	1
11	Crear pregunta de actualización de videojuego Android a partir de texto	1.1
12	Modificar pregunta de actualización de videojuego Android a partir de texto	1.1
13	Eliminar pregunta de actualización de videojuego Android a partir de texto	1.2
14	Crear actualización de videojuego Android a partir de <i>plugin</i>	1.3
15	Modificar actualización de videojuego Android a partir de <i>plugin</i>	1.2
16	Eliminar actualización de videojuego Android a partir de <i>plugin</i>	1.2
17	Cargar archivo de actualización de videojuego Android a partir de <i>plugin</i>	1

2.4.3 Plan de Duración de Iteraciones

Como práctica de la metodología XP se propone el uso de un plan de duración de iteraciones el cual permitirá el establecimiento del tiempo de demora de desarrollo de cada una de las versiones del sistema. Además, otorgará una mayor organización a la hora de establecer el orden en que serán implementadas las HU.

Tabla 15. Plan de Duración de Iteraciones (Fuente: Elaboración propia)

Iteraciones	No.	Historia de usuario	Puntos Estimados (Semanas)
1	1	Autenticar usuario	1.1
	2	Modificar credenciales	1.3
2	3	Crear actualización de videojuego Android a partir de texto	1.2
	4	Mostrar listado de actualizaciones de videojuegos Android	1
	5	Modificar actualización de videojuego Android a partir de texto	1.2
	6	Seleccionar actualización de videojuego Android	1

	7	Eliminar actualización de videojuego Android a partir de texto	1.1
	8	Generar actualización de videojuego Android a partir de texto	1.5
	9	Cargar imagen principal de actualización de videojuego Android a partir de texto	1
	10	Mostrar listados de preguntas de actualización de videojuego Android a partir de texto	1
	11	Crear pregunta de actualización de videojuego Android a partir de texto	1.1
	12	Modificar pregunta de actualización de videojuego Android a partir de texto	1.1
	13	Eliminar pregunta de actualización de videojuego Android a partir de texto	1.2
3	14	Crear actualización de videojuego Android a partir de <i>plugin</i>	1.3
	15	Modificar actualización de videojuego Android a partir de <i>plugin</i>	1.2
	16	Eliminar actualización de videojuego Android a partir de <i>plugin</i>	1.2
	17	Cargar archivo de actualización de videojuego Android a partir de <i>plugin</i>	1
Duración Total (Semanas).			19.5
Duración Total (Meses).			4.875

2.4.4 Plan de Entrega

El plan de entrega pronostica la duración de las iteraciones que generarán las versiones del sistema.

Tabla 16. Plan de Entrega (Fuente: Elaboración propia)

Iteración	Fecha de inicio	Fecha de finalización	Entregable
1	01/02/2018	18/02/2018	Versión 0.1
2	19/02/2018	17/05/2018	Versión 0.7
3	18/05/2018	21/06/2018	Versión 1.0

2.5 Diseño

XP establece prácticas sobre el diseño para lograr un sistema robusto y reutilizable manteniendo su simplicidad. Logrando que cambios o modificaciones futuras puedan realizarse de manera más sencilla. XP no define una técnica específica de modelado, pueden utilizarse indistintamente sencillos esquemas, tarjetas CRC o diagramas de clase utilizando UML, siempre que sean útiles y no requieran mucho tiempo en su creación (Beck, 2012).

La metodología XP, por su parte, propone un diseño sencillo, simple y robusto como parte del proceso de desarrollo de *software*, capaz de generar un sistema escalable y de fácil implementación.

2.5.1 Patrón Arquitectónico

Expresa una organización estructural parcial de un sistema de *software*, define un conjunto de elementos así como cada una de sus responsabilidades y restricciones; permitiendo la comprensión de las propiedades generales del sistema.

Un correcto uso de los patrones arquitectónicos garantizará una mayor seguridad, usabilidad, confiabilidad, rendimiento y reusabilidad del sistema.

El estilo arquitectónico seleccionado fue Modelo – Vista – Controlador al estar los datos, la lógica de negocios y las interfaces de usuario del sistema separadas.

Modelo: Las clases *Usuario.java*, *ActualizacionPlugin.java*, *ActualizacionGeneradas.java*, entre otras, serán las encargadas de acceder a la información e interactuar y actualizar los datos del sistema.

Vista: Las páginas JSP e *Index.html* serán las responsables de presentar los datos que el modelo les proporcione. Contendrán código HTML junto con CSS y algunas funciones *JavaScript*.

Controlador: La clase *Servidor.java*, servirá de enlace entre la vista y el modelo, a través del envío de comandos entre ambos permitiendo su actualización y modificando la vista según las peticiones realizadas por el usuario.

2.5.2 Tarjetas CRC

Las tarjetas CRC (clase-responsabilidad-colaborador) son fichas, una por cada clase, en las que se describen brevemente las responsabilidades de la clase, y una lista de los objetos con los que colabora para llevar a cabo sus responsabilidades (Larman, 2003).

La metodología XP propone el uso de tarjetas CRC como forma de modelación con el fin de ayudar a los desarrolladores a crear y entender los diseños del *software*. Permiten inventariar cada una de las clases utilizadas en el sistema, sus responsabilidades e interacciones, logrando definir las especificaciones de la herramienta.

Tabla 17. TarjCRC-Actualización (Fuente: Elaboración propia)

Tarjeta CRC	
Clase: <i>Actualizacion.java</i>	
Responsabilidad:	Colaboradores
<ul style="list-style-type: none"> • <i>Actualizacion()</i>: Constructor de la clase <i>Actualizacion.java</i> • <i>Getters()</i>: Devuelve los atributos de la clase <i>Actualizacion.java</i> • <i>Setters()</i>: Modifica los atributos de la clase <i>Actualizacion.java</i> 	<ul style="list-style-type: none"> • <i>Servidor.java</i> • <i>ActualizacionDB.java</i>

Tabla 18. TarjCRC-ActualizaciónPlugin (Fuente: Elaboración propia)

Tarjeta CRC	
Clase: <i>ActualizacionPlugin.java</i>	
Responsabilidad:	Colaboradores
<ul style="list-style-type: none"> • <i>ActualizacionPlugin()</i>: Constructor de la clase <i>ActualizacionPlugin.java</i> • <i>Getters()</i>: Devuelve los atributos de la clase <i>ActualizacionPlugin.java</i> • <i>Setters()</i>: Modifica los atributos de la clase <i>ActualizacionPlugin.java</i> 	<ul style="list-style-type: none"> • <i>Servidor.java</i> • <i>ActualizacionPluginDB.java</i>

Tabla 19. TarjCRC-ActualizaciónGeneradas (Fuente: Elaboración propia)

Tarjeta CRC	
Clase: <i>ActualizacionGeneradas.java</i>	

Responsabilidad:	Colaboradores
<ul style="list-style-type: none"> • <i>ActualizacionGeneradas()</i>: Constructor de la clase <i>ActualizacionGeneradas.java</i> • <i>Getters()</i>: Devuelve los atributos de la clase <i>ActualizacionGeneradas.java</i> • <i>Setters()</i>: Modifica los atributos de la clase <i>ActualizacionGeneradas.java</i> 	<ul style="list-style-type: none"> • <i>Servidor.java</i> • <i>ActualizacionGeneradasDB.java</i>

Tabla 20. TarjCRC-Question (Fuente: Elaboración propia)

Tarjeta CRC	
Clase: <i>Question.java</i>	
Responsabilidad:	Colaboradores
<ul style="list-style-type: none"> • <i>Question ()</i>: Constructor de la clase <i>Question.java</i> • <i>Getters()</i>: Devuelve los atributos de la clase <i>Question.java</i> • <i>Setters()</i>: Modifica los atributos de la clase <i>Question.java</i> 	<ul style="list-style-type: none"> • <i>Servidor.java</i> • <i>QuestionDB.java</i>

Tabla 21. TarjCRC-Usuario (Fuente: Elaboración propia)

Tarjeta CRC	
Clase: <i>Usuario.java</i>	
Responsabilidad:	Colaboradores
<ul style="list-style-type: none"> • <i>Usuario()</i>: Constructor de la clase <i>Usuario.java</i> • <i>Getters()</i>: Devuelve los atributos de la clase <i>Usuario.java</i> • <i>Setters()</i>: Modifica los atributos de la clase <i>Usuario.java</i> 	<ul style="list-style-type: none"> • <i>Servidor.java</i> • <i>UsuarioDB.java</i>

Tabla 22. TarjCRC-ClaseAbstracta (Fuente: Elaboración propia)

Tarjeta CRC
Clase: <i>ClaseAbstracta.java</i>

Responsabilidad:	Colaboradores
<ul style="list-style-type: none"> • crear(): Crear un objeto de una determinada clase • editar(): Edita un objeto seleccionado de una determinada clase • eliminar(): Elimina un objeto seleccionado de una determinada clase • obtener(): Obtiene un determinado objeto • obtenerTodos(): Obtiene todos los objetos de una determinada clase 	<ul style="list-style-type: none"> • <i>UsuarioDB .java</i> • <i>QuestionDB .java</i> • <i>ActualizacionDB.java</i> • <i>ActualizacionPluginDB .java</i> • <i>ActualizacionGeneradasDB.java</i>

Tabla 23. TarjCRC-UsuarioDB (Fuente: Elaboración propia)

Tarjeta CRC	
Clase: <i>UsuarioDB.java</i>	
Responsabilidad:	Colaboradores
<ul style="list-style-type: none"> • <i>ListUsuario()</i>: Devuelve todos los objetos de la clase <i>Usuario.java</i> presentes en la base de datos 	<ul style="list-style-type: none"> • <i>Usuario.java</i> • <i>ClaseAbstracta.java</i>

Tabla 24. TarjCRC-QuestionDB (Fuente: Elaboración propia)

Tarjeta CRC	
Clase: <i>QuestionDB.java</i>	
Responsabilidad:	Colaboradores
<ul style="list-style-type: none"> • <i>ListQuestion()</i>: Devuelve todos los objetos de la clase <i>Question.java</i> presentes en la base de datos 	<ul style="list-style-type: none"> • <i>Question.java</i> • <i>ClaseAbstracta.java</i>

Tabla 25. TarjCRC-ActualizaciónDB (Fuente: Elaboración propia)

Tarjeta CRC	
Clase: <i>ActualizacionDB.java</i>	

Responsabilidad:	Colaboradores
<ul style="list-style-type: none"> • <i>ListActualizacion()</i>: Devuelve todos los objetos de la clase <i>Actualizacion.java</i> presentes en la base de datos 	<ul style="list-style-type: none"> • <i>Actualizacion.java</i> • <i>ClaseAbstracta.java</i>

Tabla 26. TarjCRC-ActualizaciónPluginDB (Fuente: Elaboración propia)

Tarjeta CRC	
Clase: <i>ActualizacionPluginDB.java</i>	
Responsabilidad:	Colaboradores
<ul style="list-style-type: none"> • <i>ListActualizacionPlugin()</i>: Devuelve todos los objetos de la clase <i>ActualizacionPlugin.java</i> presentes en la base de datos 	<ul style="list-style-type: none"> • <i>ActualizacionPlugin.java</i> • <i>ClaseAbstracta.java</i>

Tabla 27. TarjCRC-ActualizaciónGeneradasDB (Fuente: Elaboración propia)

Tarjeta CRC	
Clase: <i>ActualizacionGeneradasDB.java</i>	
Responsabilidad:	Colaboradores
<ul style="list-style-type: none"> • <i>ListActualizacionGeneradas()</i>: Devuelve todos los objetos de la clase <i>ActualizacionGeneradas.java</i> presentes en la base de datos 	<ul style="list-style-type: none"> • <i>ActualizacionGeneradas.java</i> • <i>ClaseAbstracta.java</i>

Tabla 28. TarjCRC-Servidor (Fuente: Elaboración propia)

Tarjeta CRC	
Clase: <i>Servidor.java</i>	
Responsabilidad:	Colaboradores
<ul style="list-style-type: none"> • <i>processRequest()</i>: Encargado de recibir y dar respuesta a las peticiones del cliente 	<ul style="list-style-type: none"> • <i>HttpServlet</i> • <i>Usuario.java</i>

	<ul style="list-style-type: none">• <i>UsuarioDB.java</i>• <i>Question.java</i>• <i>QuestionDB.java</i>• <i>Actualizacion.java</i> <i>ActualizacionPlugin.java</i>• <i>ActualizacionPluginDB.java</i>• <i>ActualizacionDB.java</i>• <i>ActualizacionGeneradas.java</i>• <i>ActualizacionGeneradas.java DB</i>
--	--

2.5.3 Patrones de Diseño

Un patrón de diseño es una solución a un problema en un contexto. Es una descripción de clases y objetos que se comunican entre sí, adaptada para resolver un problema general de diseño en un contexto particular (Prieto, 2009).

Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de *software*.

Patrones GRASP

Los patrones GRASP son una forma de ayuda a la hora de entender el diseño de objetos, el comportamiento y asignación de responsabilidades en el sistema.

- **Patrón Creador:** Guía la asignación de responsabilidades relacionadas con la creación de objetos. Las nuevas instancias solo pueden ser creadas por clases que contienen los constructores y datos necesarios.
Por ejemplo, la clase *Servidor.java* será la responsable de crear instancias de la clase *actualizacion.java*, *question.java* y *usuario.java*.
- **Patrón Controlador:** Es un evento generado por actores externos, se pone en marcha cuando el controlador recibe una petición del usuario, posteriormente este procesa la petición y envía los datos a las clases pertinentes dando respuesta a la solicitud recibida.

Por ejemplo, la clase *Servidor.java* será la responsable de atender todos los eventos del sistema.

- **Patrón Alta Cohesión:** Este patrón garantiza que cada elemento del diseño realice una única labor dentro del sistema, no desempeñada por ninguno de los demás elementos.

Por *Servidor.java* ejemplo, las clases del sistema tendrán una baja complejidad y sus responsabilidades estarán bien definidas respondiendo a funcionalidades correspondientes a los datos que manejen las clases en cuestión.

- **Patrón Bajo Acoplamiento:** Este patrón garantiza que cada elemento del diseño contenga la menor cantidad de dependencias con las demás clases del negocio y contradicciones al realizar algún cambio en el sistema.

Por ejemplo, a la clase se le asignarán responsabilidades garantizando que solamente se comunique con *Updates.jsp*, *Home.jsp* e *Index.html*.

- **Patrón Experto:** Este patrón define que la responsabilidad de ciertas labores solo deben ser realizadas por las clases que contengan la información necesaria.

Por ejemplo, la clase *Servidor.java* del sistema contará con la información necesaria para cumplir con sus responsabilidades.

Patrones GOF

- **Patrón Observador:** Este patrón define que un cambio sobre alguno de los datos del modelo o propiedades del sistema tiene que ser notificado a las demás dependencias del sistema.

Por ejemplo, los cambios realizados en las actualizaciones existentes en la base de datos de la aplicación generaran modificaciones en las otras dependencias del sistema.

```
<script>
var names = [];
var generator = [];
var ids = [];
var listas = [];
var descripciones = [];
</script>

<%List<actualizacion> listado = ((List<actualizacion>) request.getAttribute("listado"));
List<Question> Lista = new ArrayList<Question>();
List<Question> ListQuestion = new ArrayList<Question>();
Question pregunta = new Question();
actualizacion objeto = (actualizacion) request.getAttribute("objeto");
usuario control = (usuario) request.getAttribute("controlador");
String valor = control.getUsuario();

if(valor == "0")
{
String Nivel = control.getPassword();
if(Nivel.equals("m"))
{
ListQuestion = objeto.getListQuestion();
for(int i = 0;i<ListQuestion.size();i++)
{
if(ListQuestion.get(i).getDificulty().contains("2"))
(Lista.add(ListQuestion.get(i)));
}
ListQuestion = Lista;
}
}
```

Ilustración 4. Observador (Fuente: Elaboración propia)

- **Patrón Instancia única:** Este patrón garantiza que exista solo una instancia de una determinada clase y que esta sea accesible.

Por ejemplo, en el sistema a desarrollar, existe una sola instancia de la clase controladora llamada *Servidor.java*, será la encargada de responder a las peticiones del usuario.

```
@WebServlet(name = "Servidor", urlPatterns = {"/Servidor"})
public class Servidor extends HttpServlet {

    @EJB
    usuarioDB usuarioDB;
    @EJB
    questionDB questionDB;
    @EJB
    actualizacionDB actualizacionDB;
    @EJB
    actualizacionGeneradasDB actualizacionGeneradasDB;
    @EJB
    actualizacionPluginDB actualizacionPluginDB;
```

Ilustración 5. Instancia única (Fuente: Elaboración propia)

Patrón Fachada: Este patrón se aplica para la reutilización de una clase o interfaz en otras interfaces o clases, permitiendo simplificar el código y disminuir la complejidad del mismo.

Por ejemplo, las clases *ActualizaciónDB.java*, *UsuarioDB.java* y *QuestionDB.java* reutilizarán el código de la clase *ClaseAbstracta.java*.

```
@Stateless
public class actualizacionDB extends ClaseAbstracta<actualizacion>
```

Ilustración 6. Fachada (Fuente: Elaboración propia)

2.6 Conclusiones parciales

- La caracterización de la propuesta de solución permitió alcanzar un mayor conocimiento del funcionamiento y aspectos claves de la herramienta a desarrollar.
- La definición de los requisitos funcionales, no funcionales y la declaración de las historias de usuarios contribuyeron a la identificación de las necesidades del cliente, dando lugar a un total de 17 historias de usuario.
- La realización de los planes de iteración y entrega permitieron planificar y organizar los procesos a ejecutarse en un plazo de 4.9 meses.
- La definición de los patrones de diseño, el patrón arquitectónico y las tarjetas CRC sentaron las bases para el desarrollo de un sistema sencillo y de fácil implementación.

Capítulo III: Implementación y pruebas

3.1 Introducción

El presente capítulo estará enmarcado en las etapas de implementación y pruebas propuestas por la metodología XP. Se generan las tareas de ingeniería a partir de las historias de usuario definidas anteriormente, simplificando la fase de desarrollo. Se aplicarán pruebas unitarias y de aceptación al sistema validando los requerimientos establecidos por el cliente. Por último, se desarrollará una aplicación Android para validar el correcto envío de los datos desde el sistema.

3.2 Implementación

La etapa de implementación del desarrollo de *software* es el proceso de convertir una especificación del sistema en un sistema ejecutable (Sommerville, 2006).

3.2.1 Tareas de Ingeniería

Una HU se descompone en varias tareas de ingeniería, las cuales describen las actividades que se realizarán en cada HU, así mismo las tareas de ingeniería se vinculan más al desarrollador, ya que permite tener un acercamiento con el código (Escutia, 2009).

Primera Iteración

En la primera iteración fueron implementadas las tareas de ingeniería relacionadas con las credenciales del usuario del sistema.

Tabla 29. TI_1 (Fuente: Elaboración propia)

Tarea de Ingeniería	
Número de Tarea: 1	Número de Historia de Usuario: 1
Nombre de la Tarea: Autenticar usuario	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1.1
Fecha Inicio: 01/02/2018	Fecha Fin: 09/02/2018
Programador Responsable: René Orta Martínez	

Descripción: El sistema muestra los campos Usuario y Contraseña con el fin de llevar a cabo la autenticación

Tabla 30. TI_2 (Fuente: Elaboración propia)

Tarea de Ingeniería	
Número de Tarea: 2	Número de Historia de Usuario: 2
Nombre de la Tarea: Modificar la credencial usuario	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.6
Fecha Inicio: 09/02/2018	Fecha Fin: 13/02/2018
Programador Responsable: René Orta Martínez	
Descripción: El sistema superpone una nueva interfaz en la cual se muestran los campos Contraseña y Nuevo Usuario, los cuales al ser rellenados de manera correcta por el usuario permiten al mismo modificar su usuario.	

Tabla 31. TI_3 (Fuente: Elaboración propia)

Tarea de Ingeniería	
Número de Tarea: 3	Número de Historia de Usuario: 2
Nombre de la Tarea: Modificar la credencial contraseña	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.7
Fecha Inicio: 13/02/2018	Fecha Fin: 18/02/2018
Programador Responsable: René Orta Martínez	
Descripción: El sistema superpone una nueva interfaz en la cual se muestran los campos Contraseña, Nueva Contraseña y Repita Nueva Contraseña los cuales al ser rellenados de manera correcta por el usuario permiten al mismo modificar su contraseña.	

Segunda Iteración

En la segunda iteración fueron implementadas las tareas de ingeniería relacionadas con las actualizaciones de videojuegos Android generadas a partir de textos y sus respectivas preguntas ([Ver Anexo 2](#)).

Tercera Iteración

En la tercera iteración fueron implementadas las tareas de ingeniería relacionadas con las actualizaciones de videojuegos Android generadas a partir de *plugins* ([Ver Anexo 2](#)).

3.3 Pruebas

Las pruebas son un elemento crítico para la calidad del *software*. La importancia de los costos asociados a los errores, promueve la definición y aplicación de un proceso de pruebas minuciosas y bien planificadas. Las pruebas permiten validar y verificar el *software*, entendiendo como validación del *software* el proceso, externo al equipo de desarrollo, que determina si el *software* satisface los requisitos. Determina si los productos de una fase satisfacen las condiciones de dicha fase (Pressman, 2010).

3.3.1 Pruebas Unitarias y de Aceptación

- **Pruebas Unitarias**

Procedimiento utilizado para validar que un módulo de un objeto fuente funciona apropiadamente y de forma independiente. A través de ellos se verifica que cierto método se ejecuta dentro de los parámetros y especificaciones concretadas en documentos, tales como los casos de uso y el diseño detallado; que permiten la detección de defectos durante las fases sucesivas de desarrollo o mantenimiento (Herrmann, 2011).

Las pruebas unitarias permiten comprobar que el código no contiene errores, fallos y que su puesta en marcha devuelve los valores esperados, su diseño es responsabilidad de los programadores y son probadas durante la etapa de implementación. Para la validación se diseñaron casos de pruebas a partir de las unidades mínimas del sistema, obtenidas mediante el desglose de las historias de usuario en tareas de programación. Las pruebas unitarias fueron realizadas teniendo en cuenta las pruebas de caja blanca a través de la técnica del camino básico ([Ver Anexo 3](#)). Los casos de pruebas diseñados a partir de esta técnica se realizaron según se muestra para cada tarea de programación, los cuales cubren todos los caminos lineales independientes obtenidos.

- **Pruebas de Aceptación**

Son pruebas diseñadas por el propio equipo de desarrollo en base a los requisitos funcionales especificados en la fase de análisis, y ejecutadas por el propio usuario final dándole validez y conformidad al producto a entregar según lo acordado inicialmente (Sanz, 2010).

Se realizaron un total de 26 pruebas de aceptación, organizadas en 3 iteraciones. A continuación se muestran las principales.

Tabla 32. PA1_HU1 (Fuente: Elaboración propia)

Caso de Prueba de Aceptación	
Código: PA1_HU1	Número de Historia de Usuario: 1
Nombre: Autenticar usuario	
Descripción: Lograr autenticar usuario	
Condiciones de Ejecución: Debe estar iniciado el sistema	
Entradas/Pasos de Ejecución:	
<ol style="list-style-type: none"> 1. El usuario rellena los campos Usuario y Contraseña 2. El usuario presione el botón Entrar 	
Resultado Esperado: Si el usuario esta autenticado accede a la página principal del sistema, en caso contrario los campos vuelven a su estado inicial y se muestra un mensaje indicando "Credenciales Incorrectas"	
Evaluación de la Prueba: Satisfactoria	

Tabla 33. PA1_HU21 (Fuente: Elaboración propia)

Caso de Prueba de Aceptación	
Código: PA1_HU21	Número de Historia de Usuario: 2
Nombre: Modificar credenciales (Usuario)	
Descripción: Lograr modificar el usuario del administrador	
Condiciones de Ejecución: Se debe haber presionado el ícono de configuración, presente en la zona superior derecha de la pantalla principal del sistema	
Entradas/Pasos de Ejecución:	
<ol style="list-style-type: none"> 1. El usuario rellena los campos Contraseña y Nuevo Usuario 	

2. El usuario presiona el botón Guardar Cambios
Resultado Esperado: Si la contraseña es correcta, el usuario es modificado, en caso contrario muestra un mensaje indicando “Contraseña incorrecta” Si existe algún campo vacío se mostrará un mensaje indicando “Campo vacío”
Evaluación de la Prueba: Satisfactoria

Tabla 34. PA1_HU22 (Fuente: Elaboración propia)

Caso de Prueba de Aceptación	
Código: PA1_HU22	Número de Historia de Usuario: 2
Nombre: Modificar credenciales (Contraseña)	
Descripción: Lograr modificar la contraseña del administrador	
Condiciones de Ejecución: Se debe haber presionado el ícono de configuración, presente en la zona superior derecha de la pantalla principal del sistema y posteriormente seleccionada la opción Cambiar Contraseña	
Entradas/Pasos de Ejecución:	
<ol style="list-style-type: none"> 1. El usuario rellena los campos Contraseña, Nueva Contraseña y Repita Nueva Contraseña 2. El usuario presiona el botón Guardar Cambios 	
Resultado Esperado: Si la contraseña es correcta y la Nueva Contraseña contiene hasta 8 caracteres con al menos uno especial y esta coincide con Repita Nueva Contraseña; la contraseña es modificada, en caso contrario muestra un mensaje indicando “Contraseña incorrecta”, “Su nueva contraseña debe contar con más de 7 caracteres” o “Repita nuevamente su contraseña”. Si existe algún campo vacío se mostrará un mensaje indicando “Campo vacío”	
Evaluación de la Prueba: Satisfactoria	

Tabla 35. PA1_HU3 (Fuente: Elaboración propia)

Caso de Prueba de Aceptación	
Código: PA1_HU3	Número de Historia de Usuario: 3
Nombre: Crear actualización de videojuego a partir de texto	
Descripción: Lograr crear una nueva actualización de videojuego a partir de texto	
Condiciones de Ejecución: El usuario debe haber accedido a la sesión de Actualizaciones y	

presionado el ícono crear
Entradas/Pasos de Ejecución: <ol style="list-style-type: none">1. El usuario rellena el campo Nombre de Actualización y campo opcional Descripción2. El usuario presiona el botón Guardar3. El usuario presiona el botón Aceptar para aprobar el proceso
Resultado Esperado: Si el nombre de la actualización del videojuego a partir de texto y la descripción no contienen caracteres reservados del lenguaje de programación Java, se inicia el proceso para la creación de una nueva actualización. Surge una nueva interfaz con el botón aceptar para aprobar el procedimiento que se lleva a cabo, el cual al ser presionado genera una nueva actualización de videojuego a partir de texto en el sistema, en caso contrario, se muestra un mensaje de “Existen caracteres reservados del lenguaje” Si el campo Nombre de Actualización quedo vacío se muestra el mensaje “Debe introducir el nombre de la actualización”
Evaluación de la Prueba: Satisfactoria

El resto de los casos de pruebas de aceptación se encuentran en los Anexos ([Ver Anexo 3](#)).

Se realizaron tres iteraciones de pruebas de aceptación a la Versión 0.1, generando los siguientes resultados:

1. En la primera iteración se detectaron un total de 5 no conformidades.
 - 3 errores de validación.
 - 2 faltas ortográficas.
2. En la segunda iteración se detectaron un total de 2 no conformidades.
 - 1 error de validación.
 - 1 falta ortográfica.
3. En la tercera iteración se detectaron un total de 0 no conformidades.

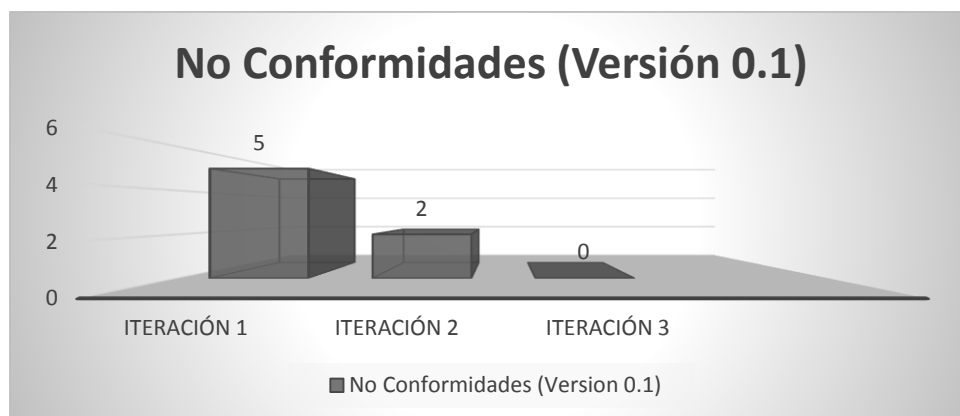


Ilustración 7. V.0.1 (Fuente: Elaboración propia)

Se realizaron tres iteraciones de pruebas de aceptación a la Versión 0.7, generando los siguientes resultados:

1. En la primera iteración se detectaron un total de 11 no conformidades.
 - 8 errores de validación.
 - 3 faltas ortográficas.
2. En la segunda iteración se detectaron un total de 6 no conformidades.
 - 5 errores de validación.
 - 1 falta ortográfica.
3. En la tercera iteración se detectaron un total de 0 no conformidades.

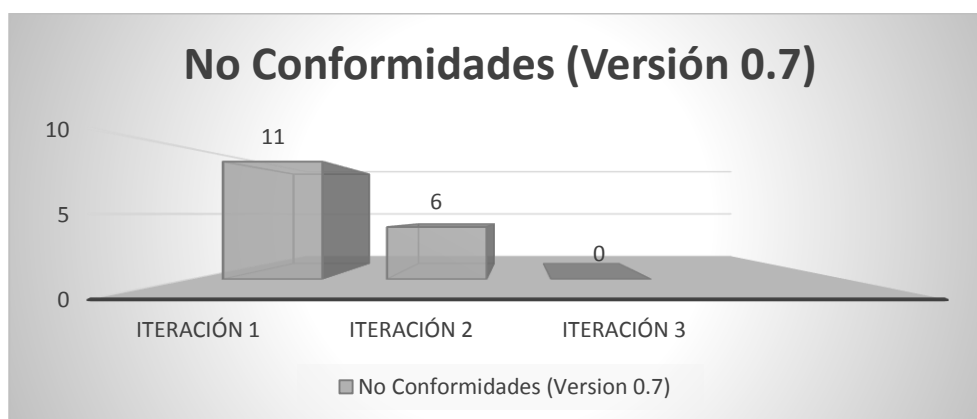


Ilustración 8. V.0.7 (Fuente: Elaboración propia)

Se realizaron tres iteraciones de pruebas de aceptación a la Versión 1.0, generando los siguientes resultados:

1. En la primera iteración se detectaron un total de 9 no conformidades.

- 7 errores de validación.
 - 2 faltas ortográficas.
2. En la segunda iteración se detectaron un total de 4 no conformidades.
 - 4 errores de validación.
 3. En la tercera iteración se detectaron un total de 0 no conformidades.

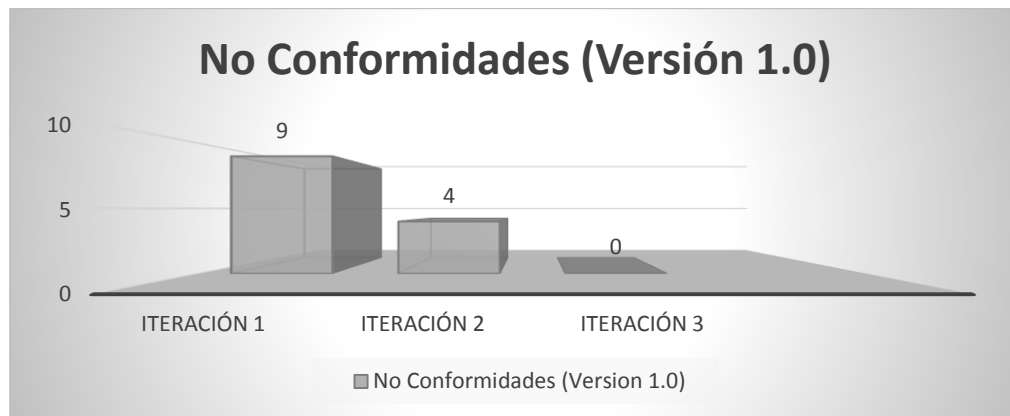


Ilustración 9. V.1.0 (Fuente: Elaboración propia)

3.4 Aplicación Android para pruebas

Se desarrolló una aplicación Android para validar el correcto funcionamiento del sistema. La misma, recibirá los datos gestionados en el sistema y los visualizará en un campo de texto a través del consumo de un servicio GET generado por la API REST asociada a la herramienta.

Interfaces de la aplicación Android:

La primera interfaz cuenta con:

- ✓ Dos botones para hacer llamadas a los servicios REST implementados en la herramienta para la gestión de actualizaciones.
- ✓ Un *TextView* para visualizar los datos que se reciben a través del servicio implementado en la herramienta.
- ✓ Un botón para configurar la dirección IP a la que se conectará la APK.

La segunda interfaz muestra el logo de la aplicación, desde esta se accede mediante un *Thread* a la primera interfaz.

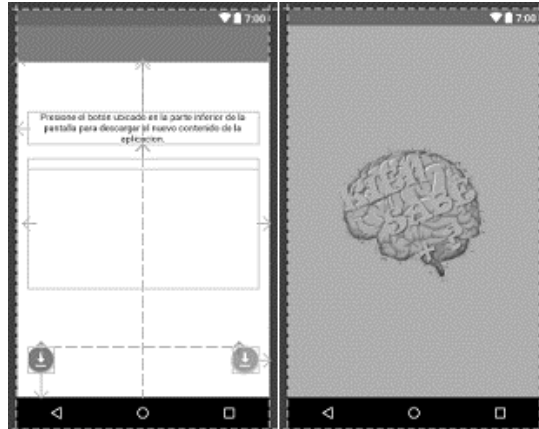


Ilustración 10. Interfaces APK Android (Fuente: Elaboración propia)

Funcionalidades:

- ✓ `HttpPostPeticiónAsync ()`: Será el encargado de hacer la petición al servicio web implementado en la herramienta, recibir y visualizar en pantalla los datos que se reciben de la misma. Es un método asíncrono, es decir, se ejecuta en un hilo secundario para garantizar que la vista de la aplicación no se congele durante el proceso de descarga de los datos.

```
private class HttpPostPeticiónAsync extends AsyncTask<String, Integer, String> {
    @Override
    protected void onPreExecute() { super.onPreExecute(); }
    @Override
    protected String doInBackground(String... params) {
        StrictMode.ThreadPolicy policy=new StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);
        URL url;
        HttpURLConnection conn;
        String json="";
        try {
            url=new URL(params[0]);
            conn=(HttpURLConnection) url.openConnection();
            conn.connect();
            BufferedReader in=new BufferedReader(new InputStreamReader(conn.getInputStream()));
            String inputLine;
            StringBuffer response=new StringBuffer();
            while ((inputLine=in.readLine())!=null){
                response.append(inputLine);
            }
            json=response.toString();
        } catch (Exception e) {
            e.printStackTrace();
            Log.e("error", "error");
        }
        return json;
    }
    @Override
    protected void onPostExecute(String result) {
        super.onPostExecute(result);
        tv.setText(tv.getText().toString()+"\n\n"+result);
    }
}
```

Ilustración 11. Función Android `HttpPostPeticiónAsync` (Fuente: Elaboración propia)

Resultados obtenidos de la aplicación Android

Sistema para la gestión de actualizaciones de videojuegos Android mediante la red

Al consumir los servicios implementados en el sistema desde la aplicación Android se recibieron los datos requeridos según el tipo de petición realizada.

Los datos de las actualizaciones Android generadas a partir de textos se recibieron en formato JSON de la siguiente manera:

```
[{"id":3,"image":"r00ABXNyABNqYXZLnV0aWwuQXJyYXlMaXN0eIHSZnHYZ0DAAFJAARzaXpleHAAAAABdwQAAAABdAAQVzBKQU5qTTRZVGt3TIRRPXg=","listquestion":"Preguntas","name":"Polo"}]
```

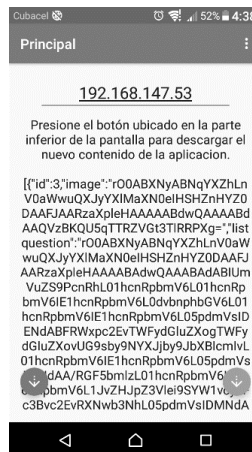


Ilustración 12. Respuesta de actualización de videojuego a partir de texto (Fuente: Elaboración propia)

Los datos de las actualizaciones Android generadas a partir de *plugins* se recibieron en formato JSON de la siguiente manera:

```
[{"dir":"10.55.18.240:8080\\AplicacionTesis\\FILES\\FILES\\annotations.zip","fecha":"01/01/2200","id":52,"name":"Tribunal Corte","version":33}]
```

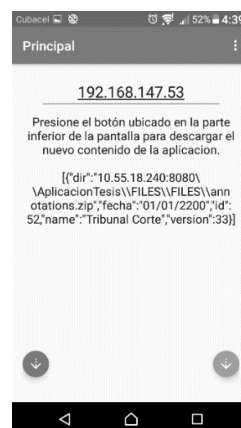


Ilustración 13. Respuesta de actualización de videojuego a partir de plugin (Fuente: Elaboración propia)

3.5 Conclusiones parciales

- La definición de las tareas de ingeniería generadas, a partir de las historias de usuario especificadas por el cliente, permitió contar con un mayor nivel de conocimiento, organización y rapidez para la implementación de las funcionalidades del sistema.
- La implementación de la propuesta de solución permitió al centro Vertex contar con un sistema para la actualización de sus videojuegos en la plataforma Android.
- El desarrollo de una APK Android de prueba permitió verificar el correcto envío de contenidos entre el sistema desarrollado y aplicaciones Android.
- La validación del sistema a través de las pruebas unitarias y de aceptación permitió detectar un total de 25 no conformidades, que fueron solucionadas garantizando el correcto funcionamiento del sistema.

Conclusiones generales

Con el desarrollo de la presente investigación se obtuvo un sistema para la gestión de las actualizaciones de los videojuegos Android mediante la red, para lo cual:

- El estudio de los sistemas para la gestión de actualizaciones Android y sus características permitió sentar las bases para el desarrollo del sistema.
- La definición de los requisitos funcionales, historias de usuario, planes y arquitectura a utilizar, permitieron precisar las características, restricciones y diseño del sistema a desarrollar.
- La metodología XP permitió documentar el ciclo de vida del desarrollo de la solución informática.
- La selección de las herramientas: *NetBeans*, *Visual Paradigm*, *PostgreSQL* y las tecnologías: *Java*, *JavaScript*, *CSS3* y *HTML5* garantizó el desarrollo de un sistema con las características especificadas por el cliente.
- El desarrollo de un sistema para la gestión de las actualizaciones, permitió solucionar el problema informático propuesto, proporcionándole al centro *Vertex* una herramienta generadora de actualizaciones para sus videojuegos Android.
- El desarrollo de una *APK* Android de prueba permitió verificar el correcto envío de contenidos entre el sistema desarrollado y aplicaciones Android.
- La validación del sistema a partir de las pruebas unitarias y de aceptación propuesta por la metodología *XP*, permitió comprobar el correcto funcionamiento de la herramienta.

Recomendaciones

A partir de los resultados obtenidos se recomienda:

- Elaborar un modelo de datos más complejo en aras de lograr actualizaciones más potentes en los videojuegos nativos Android del centro Vertex.

Referencias Bibliográficas

Alvarez, Miguel Angel. 2009. Introducción a JavaScript. [En línea] 2009. [Citado el: 1 de Marzo de 2018.] <http://www.desarrolloweb.com/articulos/490.php>.

Apklis. 2018. Apklist. [En línea] 2018. [Citado el: 21 de Junio de 2018.] <https://www.apklis.cu/es/information/about>.

Beck, Kent. 2012. Extreme Programming. Explained Second Edition. [En línea] Enero de 2012. [Citado el: 21 de Febrero de 2018.] <http://ptgmedia.pearsoncmg.com/images/9780321278654/samplepages/9780321278654.pdf>.

Bravo, José Raúl Salao. 2009. Estudio de las Técnicas de Inteligencia Artificial mediante el apoyo de un Software Educativo. [En línea] 2009. [Citado el: 27 de Febrero de 2017.] <http://dspace.esoch.edu.ec/bitstream/123456789/90/1/18T00366.pdf>.

Castellanos, Diana. 2015. Gestión de Tecnología . [En línea] 2015. [Citado el: 26 Gestión de Tecnología de Abril de 2018.] <http://dcastellanosg.blogspot.com/2015/05/metodologias-agiles-la-cabeza-del-exito.html>.

Cevallos, Ingrid Evelyn Cevallos. 2014. Análisis Comparativo de Respaldo y Recuperación de Base de Datos Licenciada (Oracle Utilizando RMAN) VS Open Source (MYSQL Utilizando MYSQL Administrator). [En línea] 21 de Marzo de 2014. <http://repositorio.ug.edu.ec/handle/redug/6540>.

Ciberaula. 2010. Una introducción a Apache. [En línea] 2010. [Citado el: 12 de Diciembre de 2017.] http://linux.ciberaula.com/articulo/linux_apache_intro/.

Condori, Shirley Fabiola Ticona. 2010. Comparación de Metodologías. [En línea] 2010. [Citado el: 30 de Diciembre de 2017.]

Dept. Ciencia de la Computación e IA. 2013. Servicios Web y Soap. [En línea] Dept. Ciencia de la Computación e IA, 2013. [Citado el: 2017 de Octubre de 15.] <http://www.jtech.ua.es/j2ee/publico/servc-web-2012-13/wholesite.pdf>.

Escutia, Rogelio Ferreira. 2009. XP Xtreme Programming. [En línea] 2009. [Citado el: 30 de Septiembre de 2017.] <http://slideplayer.es/slide/84721/>.

F-Droid. 2018. F-Droid. [En línea] 2018. [Citado el: 12 de Junio de 2018.] <https://f-droid.org/en/docs/>.

Fergarciac. 2013. Entorno de Desarrollo Integrado (IDE). [En línea] 2013. [Citado el: 10 de Noviembre de 2017.] <https://fergarcia.wordpress.com/2013/01/25/entorno-de-desarrollo-integrado-ide/>.

Flores, Frederick Valentín Álvarez. 2015. PLATAFORMA INTERACTIVA MULTI-USUARIO DE GESTION Y ANALISIS DEL PROYECTO CIEMPIESS-UNAM. [En línea] 2015. [Citado el: 20 de Abril de 2018.] <http://www.ptolomeo.unam.mx:8080/xmlui/handle/132.248.52.100/7454>.

García, Antonio Collazo y Labrador, Marycarmen Díaz. 2013. La Programación Extrema. [En línea] 27 de Noviembre de 2013. [Citado el: 21 de Febrero de 2018.] https://www.researchgate.net/publication/318211906_La_programacion_extrema#pf6.

Gironés, Jesús Tomás. 2013. El gran libro de Android Avanzado. [En línea] 2013. [Citado el: 25 de Noviembre de 2017.] <https://www.scribd.com/doc/231841722/El-gran-libro-Android-Avanzado-pdf>.

González, Juan Carlos. 2017. F-Droid: el repositorio alternativo a Google Play que Android necesita. [En línea] 4 de Diciembre de 2017. [Citado el: 12 de Junio de 2018.] <https://www.xatakandroid.com/aplicaciones-android/f-droid-el-repositorio-alternativo-a-google-play-que-android-necesita>.

Herrmann, Elisa. 2011. Pruebas de Software: Herramientas: Pruebas Unitarias. [En línea] 14 de Noviembre de 2011. [Citado el: 10 de Marzo de 2018.] http://www.kybele.etsii.urjc.es/docencia/IS_LADE/2011-2012/Material/Pruebas%20de%20SoftwareHerramientas.pdf.

Huguet, Josep Mir. 2012. Estudio de los futuros estándares HTML5 y CSS3. [En línea] Septiembre de 2012. [Citado el: 5 de Noviembre de 2017.] <https://repositori.udl.cat/bitstream/handle/10459.1/45984/jmirh.pdf>.

Ibarreche, José Miguel Reyes. 2010. Creación de una plataforma de desarrollo de aplicaciones para Android. Escuela Técnica Superior de Ingeniería de Madrid. España. Tesis. 2010. . [En línea] 2010. [Citado el: 20 de Noviembre de 2017.] <https://updoc.tips/download/free-pdf-ebook-creacion-de-una-plataforma-de-desarrollo-de-aplicaciones-para-android#>.

Joskowicz, Jose. 2008. Reglas y Prácticas en Xtreme Programming. [En línea] 10 de febrero de 2008. [Citado el: 21 de febrero de 2018.] <https://ie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>.

Larman, Craig. 2003. UML y Patrones. [En línea] 2003. [Citado el: 24 de Enero de 2018.] <http://www.fmonje.com/UTN/ADES%20-%202008/UML%20y%20Patrones%20%202da%20Edicion.pdf>.

Maida, Esteban Gabriel y Pacienza, Julián. 2015. Metodologías de desarrollo de software. [En línea] 2015. [Citado el: 8 de Noviembre de 2017.] <http://bibliotecadigital.uca.edu.ar/repositorio/tesis/metodologias-desarrollo-software.pdf>.

Meléndez, Gaitan, Pérez. 2018. METODOLOGIA ÁGIL. [En línea] 8 de Junio de 2018. [Citado el: 28 de Enero de 2016.] <http://repositorio.unan.edu.ni/1365/1/62161.pdf>.

Mori, Jose Germán Núñez. 2010. Usabilidad en Metodologías Ágiles. [En línea] Noviembre de 2010. [Citado el: 4 de Marzo de 2018.] https://www.fi.upm.es/catedra-ibmrational/sites/www.fi.upm.es/catedra-ibmrational/files/Tesis_MarcoAgilTrabajo.pdf.

Netbeans. 2017. NetBeans. [En línea] 2017. [Citado el: 21 de Noviembre de 2017.] <https://netbeans.org/community/releases/80/>.

PostgreSQL. 2016. [En línea] 2016. [Citado el: 30 de Enero de 2018.] <http://www.postgresql.org/about/>.

Pressman, Roger S. 2010. Ingeniería del Software. Un Enfoque Práctico. [En línea] 2010. [Citado el: 28 de Octubre de 2017.] <http://cotana.informatica.edu.bo/downloads/Id-Ingenieria.de.software.enfoque.practico.7ed.Pressman.PDF#page=8&zoom=auto,-72,710>.

Prieto, Félix. 2009. Patrones de diseño. [En línea] 2009. [Citado el: 10 de Diciembre de 2017.] https://www.infor.uva.es/~felix/datos/priiii/tr_patrones-2x4.pdf.

Profis, Sharon. 2012. Five things you may not know about Google Play. [En línea] 6 de Marzo de 2012. [Citado el: 12 de Junio de 2018.] <https://www.cnet.com/how-to/five-things-you-may-not-know-about-google-play/>.

Qode. 2013. ¿ Qué es Google Play ? [En línea] 2 de Octubre de 2013. [Citado el: 10 de Junio de 2018.] <http://qode.pro/blog/que-es-google-play/>.

Rodríguez, Dayana Bustamante y Jean C. 2014. Metodología Actual Metodología XP. [En línea] Marzo de 2014. [Citado el: 4 de Octubre de 2017.] <http://blogs.unellez.edu.ve/dsilva/files/2014/07/Metodologia-XP.pdf>.

Sanz, Luís Fernández. 2010. Revista Española de Innovación, Calidad e Ingeniería de Software. [En línea] 2010. [Citado el: 1 de Diciembre de 2017.] https://idus.us.es/xmlui/bitstream/handle/11441/23094/file_1.pdf;sequence=1.

- Sommerville, Ian. 2006.** Ingeniería de Software. Séptima Edición. [En línea] 2006. [Citado el: 23 de Junio de 2018.] <https://books.google.com.cu/books?id=gQWd49zSut4C&printsec=frontcover&dq=inauthor:%22Ian+Sommerville%22&hl=es-419&sa=X&ved=0ahUKEwi806bLx-XbAhVh7YMKHYsZDhkQ6AEILTAA#v=onepage&q&f=false>.
- Sosa, Dr. Victor J. Sosa. 2015.** Servicios Web: SOAP y REST. [En línea] 10 de marzo de 2015. [Citado el: 22 de Octubre de 2017.] http://www.tamps.cinvestav.mx/~vjsosa/clases/sd/ServiciosWeb_Axis_REST.pdf.
- The Eclipse Foundation. 2016.** THE ECLIPSE FOUNDATION. Eclipse. About the Eclipse Foundation. [En línea] 2016. [Citado el: 25 de Octubre de 2017.] [https://eclipse.org/org/.](https://eclipse.org/org/)
- UCLV, Android. 2018.** Android UCLV. [En línea] 2018. [Citado el: 13 de Junio de 2018.] [http://android.uclv.edu.cu/.](http://android.uclv.edu.cu/)
- UNID. 2015.** Ingeniería del Software. [En línea] 2015. [Citado el: 21 de febrero de 2018.] http://moodle2.unid.edu.mx/dts_cursos_md/pos/TI/IS/S07/IS07_Lectura.pdf.
- Vergara, Alida. 2016.** Tomcat vs Glassfish. [En línea] 2016. [Citado el: 4 de Junio de 2018.] [https://www.facilcloud.com/noticias/tomcat-vs-glassfish/.](https://www.facilcloud.com/noticias/tomcat-vs-glassfish/)
- Visual Paradigm. 2014.** Visual Paradigm for UML-Software design tools for agile software development. [En línea] 2014. [Citado el: 10 de Noviembre de 2017.] [http://www.visual-paradigm.com/product/vpuml/.](http://www.visual-paradigm.com/product/vpuml/)
- Wingu. 2016.** Manual de Metodologías Ágiles. [En línea] 1 de Agosto de 2016. [Citado el: 2 de Abril de 2018.] https://www.winguweb.org/system/files/biblioteca/manual_de_metologias_agiles_final.pdf.

Anexos

Anexo 1

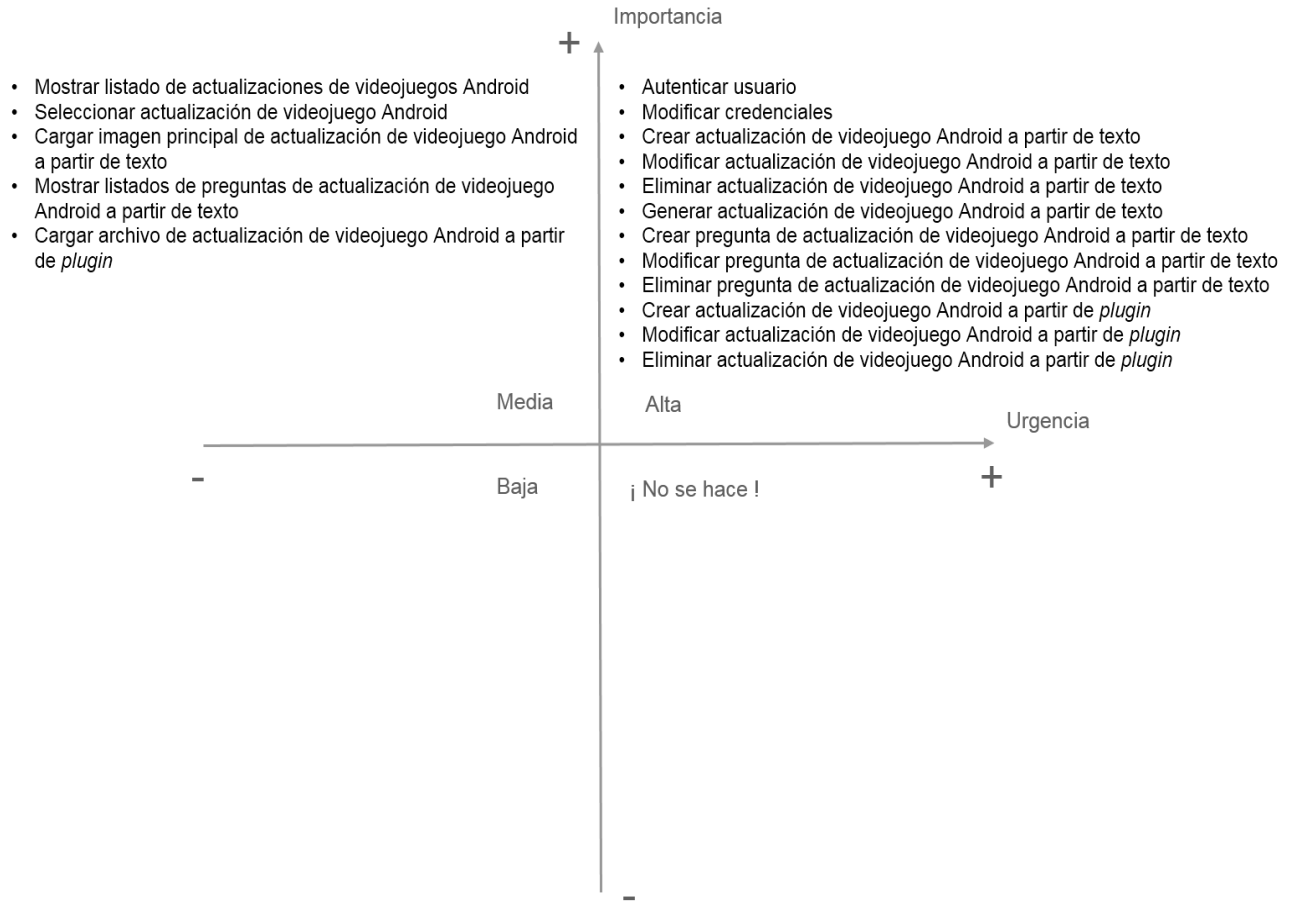


Ilustración 14. Matriz Urgencia-Importancia (Fuente: Elaboración propia)

Tabla 36. HU_3 (Fuente: Elaboración propia)

Historia de Usuario	
Número: HU_3	Nombre: Crear actualización de videojuego Android a partir de texto
Usuario: Historia Administrador	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alta

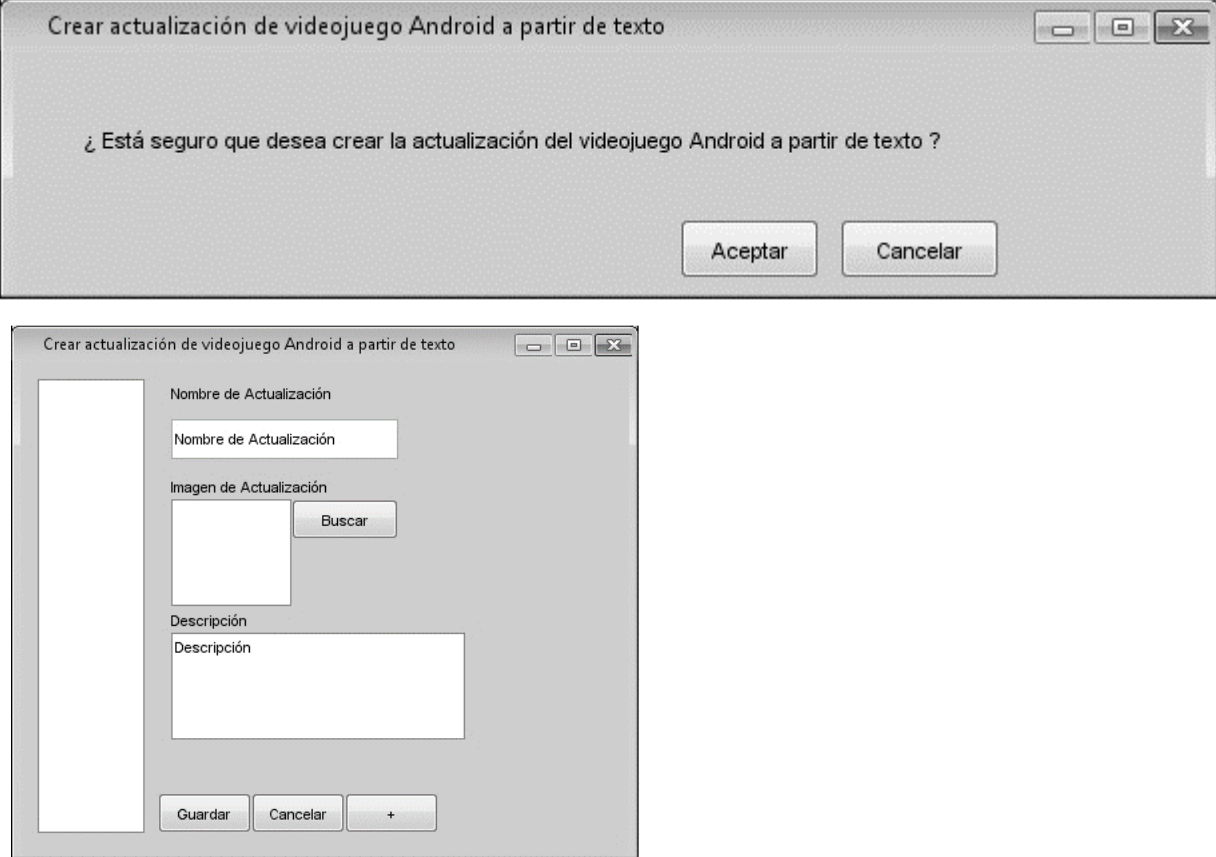
Puntos estimados: 1.2	Iteración Asignada: 2
Programador responsable: René Orta Martínez	
Descripción: Inicia cuando el usuario presiona el botón con el ícono de crear. Se muestra una interfaz con los campos: Nombre de Actualización y Descripción; la sección Imagen de Actualización, los botones: Guardar para acceder a guardar la actualización del videojuego, Buscar para cargar la imagen de la actualización del videojuego y Cancelar para reiniciar la interfaz sin guardar los datos y el botón con el ícono de agregar pregunta para incluir una serie de preguntas en la misma	
Observaciones: <ul style="list-style-type: none">• El campo Nombre de Actualización no puede ser vacío• Al presionar el botón Guardar surge una nueva interfaz con un mensaje y los botones: Cancelar para suspender el proceso de crear y guardar la nueva actualización del videojuego y Aceptar para aprobarlo	
Prototipo de interfaz:  <p>El prototipo de interfaz muestra dos ventanas de usuario. La primera es una ventana de confirmación con el título "Crear actualización de videojuego Android a partir de texto" y el mensaje "¿ Está seguro que desea crear la actualización del videojuego Android a partir de texto ?". Tiene botones "Aceptar" y "Cancelar". La segunda es una ventana de formulario con el mismo título, que contiene un campo de texto para "Nombre de Actualización", un campo de imagen para "Imagen de Actualización" con un botón "Buscar", un campo de texto para "Descripción", y botones "Guardar", "Cancelar" y "+".</p>	

Tabla 37. HU_5 (Fuente: Elaboración propia)

Historia de Usuario	
Número: HU_5	Nombre: Modificar actualización de videojuego Android a partir de texto
Usuario: Administrador	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1.2	Iteración Asignada: 1
Programador responsable: René Orta Martínez	
<p>Descripción: Inicia cuando el usuario presione alguna de las actualizaciones. Se muestra una interfaz con los campos: Nombre de Actualización y Descripción; la sección Imagen de Actualización, los botones: Guardar para acceder a salvar las modificaciones realizadas en la actualización del videojuego, Buscar para cargar una nueva imagen de la actualización del videojuego, Cancelar para reiniciar la interfaz sin guardar las modificaciones en los datos y Generar Actualización para generar la actualización del videojuego y los botones con íconos: Agregar Pregunta para insertar una nueva pregunta en la actualización del videojuego y los tres botones con íconos de dificultad para acceder al listado de preguntas de la actualización del videojuego según el nivel seleccionado</p>	
<p>Observaciones:</p> <ul style="list-style-type: none"> Al presionar el botón Guardar surge una nueva interfaz con un mensaje y los botones Cancelar para suspender el proceso de modificación de los datos de actualización del videojuego y Aceptar para aprobarlo 	
<p>Prototipo de interfaz:</p> <div style="border: 1px solid gray; padding: 10px; width: fit-content; margin: 10px auto;"> <div style="border: 1px solid gray; padding: 5px; width: 100%; background-color: #f0f0f0;"> <div style="display: flex; justify-content: space-between; align-items: center;"> Modificar actualización de videojuego Android a partir de texto - □ × </div> <div style="text-align: center; padding: 20px 0;"> <p>¿ Está seguro que desea modificar la actualización del videojuego Android a partir de texto ?</p> <div style="display: flex; justify-content: center; gap: 20px; margin-top: 10px;"> <input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/> </div> </div> </div> </div>	

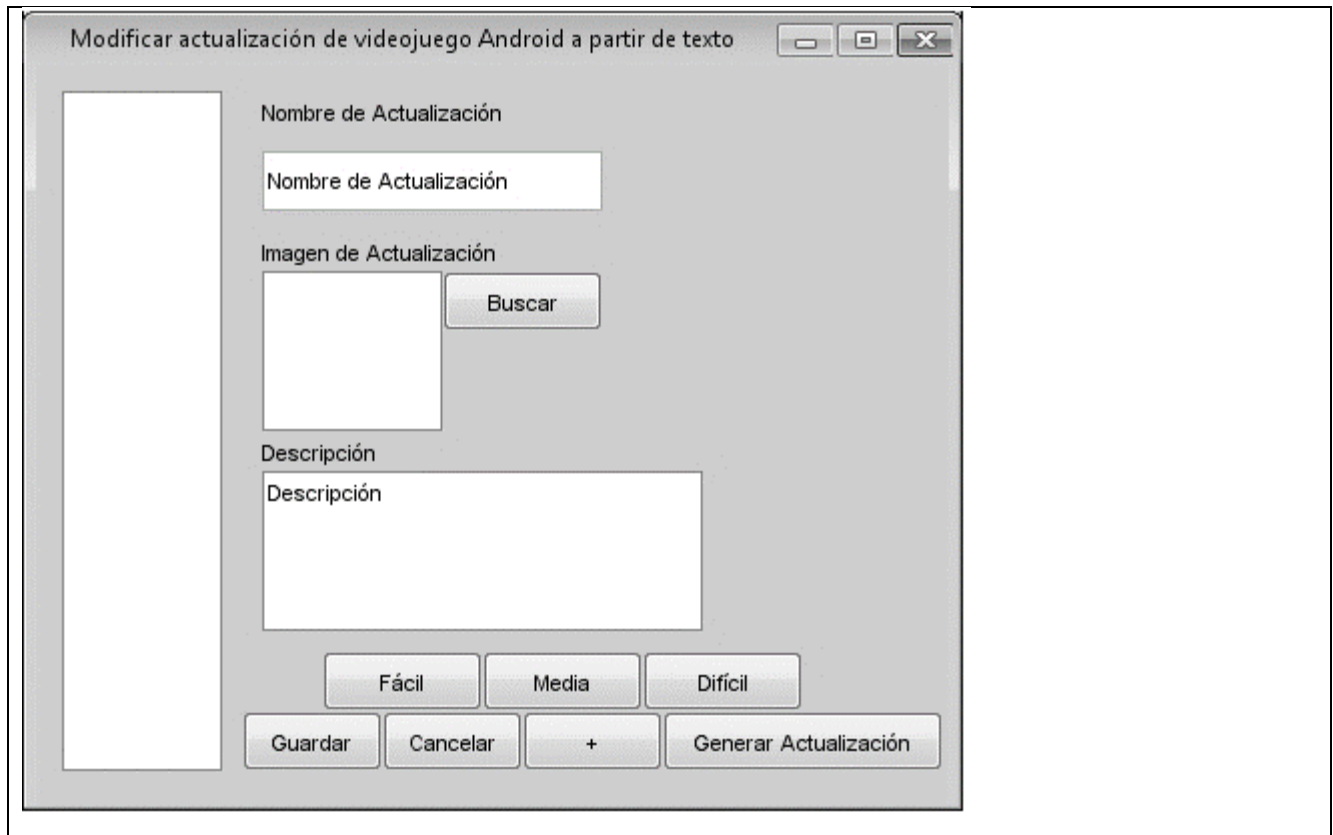


Tabla 38. HU_8 (Fuente: Elaboración propia)

Historia de Usuario	
Número: HU_8	Nombre: Generar actualización de videojuego Android a partir de texto
Usuario: Administrador	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1.5	Iteración Asignada: 2
Programador responsable: René Orta Martínez	
Descripción: Inicia cuando el usuario desea consultar o modificar datos de alguna actualización de videojuego Android a partir de texto y presione el botón Generar Actualización surgiendo una nueva interfaz con un mensaje y los botones: Cancelar para suspender el proceso de generación de la actualización del videojuego y Aceptar para aprobarlo	

Observaciones:

- Esta opción solo estará disponible posteriormente a que la actualización del videojuego seleccionado presente los parámetros establecidos

Prototipo de interfaz:

Modificar actualización de videojuego Android a partir de texto

Nombre de Actualización

Imagen de Actualización

Descripción

Fácil Media Difícil

Generar actualización de videojuego Android a partir de texto

¿ Está seguro que desea generar la actualización del videojuego Android a partir de texto ?

Tabla 39. HU_10 (Fuente: Elaboración propia)

Historia de Usuario	
Número: HU_10	Nombre: Mostrar listados de preguntas de actualización de videojuego Android a partir de texto

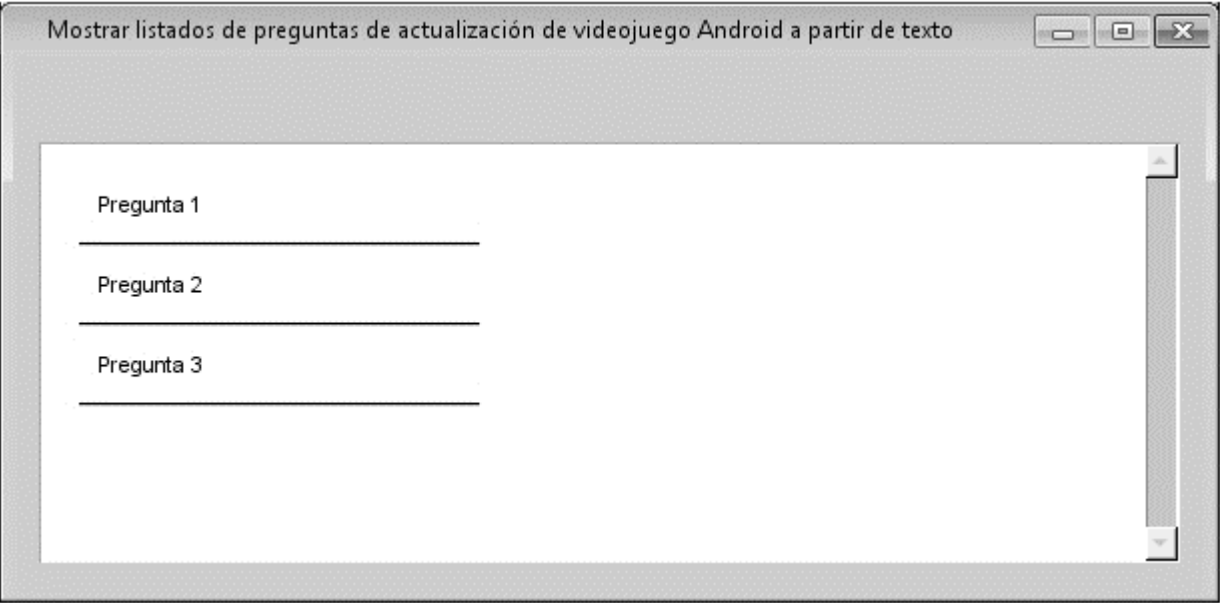
Usuario: Administrador	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración Asignada: 2
Programador responsable: René Orta Martínez	
Descripción: Inicia cuando el usuario accede a alguna de las actualizaciones de videojuegos Android a partir de texto y presiona algún ícono de control de niveles. Se muestra una nueva interfaz con el listado de las preguntas existentes en dicho nivel	
Observaciones:	
Prototipo de interfaz:	
	

Tabla 40. HU_11 (Fuente: Elaboración propia)

Historia de Usuario	
Número: HU_11	Nombre: Crear pregunta de actualización de videojuego Android a partir de texto
Usuario: Administrador	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alta


Puntos estimados: 1.1	Iteración Asignada: 2
Programador responsable: René Orta Martínez	
Descripción: Inicia cuando el usuario presiona el botón con el ícono de Agregar Pregunta. Se muestran los campos: Nueva Pregunta y Preguntas Pre-Insertadas; los <i>RadioButtons</i> con los campos: Primera Opción, Segunda Opción, Tercera Opción y Cuarta Opción relacionados, el <i>Select</i> Dificultad y los botones: Pre-Insertar para pre-agregar la pregunta a la actualización del videojuego, Cancelar para reiniciar la interfaz, Guardar para salvar las preguntas pre- agregadas y Eliminar para eliminar las preguntas pre- agregadas	
Observaciones: <ul style="list-style-type: none">• El <i>Select</i> debe tomar un valor de dificultad, el campo Nueva Pregunta no puede ser vacío y no puede contener caracteres reservados del lenguaje de programación Java• Debe haber marcado algún <i>RadioButton</i>.• Los campos destinados a las respuestas no pueden ser vacíos ni contener caracteres reservados del lenguaje de programación Java• Para que las preguntas sean guardadas se debe de presionar el botón con el ícono Insertar Pregunta y posteriormente el de Guardar, al presionar éste, surge una nueva interfaz con un mensaje y los botones: Cancelar para suspender el proceso de creación de la nueva pregunta y Aceptar para aprobarlo	
Prototipo de interfaz: 	

Tabla 41. HU_12 (Fuente: Elaboración propia)

Historia de Usuario	
Número: HU_12	Nombre: Modificar pregunta de actualización de videojuego Android a partir de texto
Usuario: Administrador	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1.1	Iteración Asignada: 2
Programador responsable: René Orta Martínez	
<p>Descripción: Inicia cuando se selecciona alguna de las preguntas de la lista de control de niveles de las actualizaciones de videojuegos Android a partir de texto. Se modifica parcialmente la interfaz y surge el campo Pregunta, los <i>RadioButtons</i> con los campos: Primera Opción, Segunda Opción, Tercera Opción y Cuarta Opción relacionados, el <i>Select</i> Dificultad y los botones: Guardar Cambios para modificar la pregunta y Cancelar para reiniciar la interfaz y desechar la opción de salvar los</p>	

cambios

Observaciones:

- El *Select* debe tomar un valor de dificultad
- El campo *Pregunta* no puede ser vacío y no puede contener caracteres reservados del lenguaje de programación Java
- Debe haber marcado algún *RadioButton*
- Los campos destinados a las respuestas no pueden ser vacíos ni contener caracteres reservados del lenguaje de programación Java
- Al presionar el botón *Guardar Cambios* surge una nueva interfaz con un mensaje y los botones: *Cancelar* para suspender el proceso de modificación de la pregunta y *Aceptar* para aprobarlo

Prototipo de interfaz:

Modificar pregunta de actualización de videojuego Android a partir ...

Pregunta

Primera Opción Segunda Opción

Tercera Opción Cuarta Opción

Dificultad

Dificultad

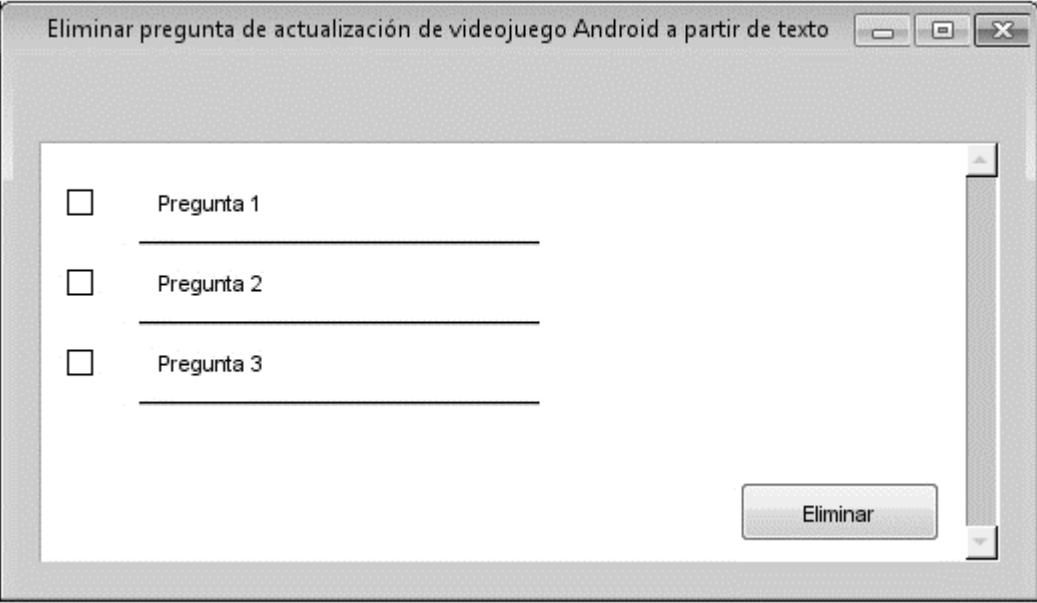
Guardar Cambios Cancelar

Modificar pregunta de actualización de videojuego Android a partir de texto

¿ Está seguro que desea modificar la pregunta de la actualización del videojuego Android a partir de texto ?

Aceptar Cancelar

Tabla 42. HU_13 (Fuente: Elaboración propia)

Historia de Usuario	
Número: HU_13	Nombre: Eliminar pregunta de actualización de videojuego Android a partir de texto
Usuario: Administrador	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1.2	Iteración Asignada: 2
Programador responsable: René Orta Martínez	
Descripción: Inicia cuando el usuario presiona el botón eliminar dentro de la lista de control de niveles de una de las actualizaciones de videojuego Android a partir de texto. Surge una nueva interfaz con un mensaje y los botones Cancelar para suspender el proceso de eliminación de la pregunta y Aceptar para aprobarlo	
Observaciones: <ul style="list-style-type: none">• Previamente se debe haber seleccionado alguna pregunta	
Prototipo de interfaz:  <p>Eliminar pregunta de actualización de videojuego Android a partir de texto</p> <p><input type="checkbox"/> Pregunta 1</p> <p><input type="checkbox"/> Pregunta 2</p> <p><input type="checkbox"/> Pregunta 3</p> <p>Eliminar</p>	

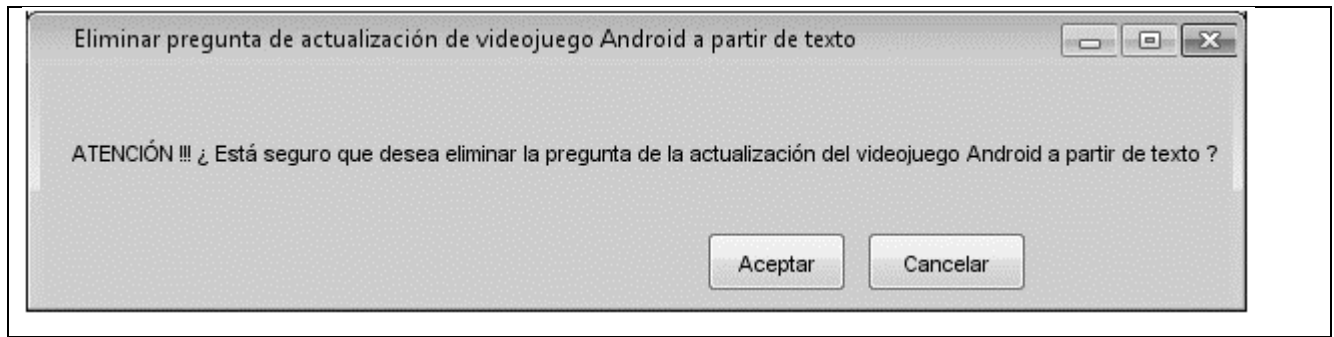


Tabla 43. HU_15 (Fuente: Elaboración propia)

Historia de Usuario	
Número: HU_15	Nombre: Modificar actualización de videojuego Android a partir de <i>plugin</i>
Usuario: Administrador	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1.2	Iteración Asignada: 3
Programador responsable: René Orta Martínez	
Descripción: Inicia cuando el usuario presiona alguna actualización de videojuego Android a partir de <i>plugin</i> . Se muestra una interfaz con los campos: Nombre Paquete, Versión y Fecha y los botones: Cancelar para abandonar la sección y Guardar para crear la actualización del videojuego siempre y cuando los datos ingresados sean correctos	
Observaciones: <ul style="list-style-type: none"> • Los campos no pueden ser vacíos ni contener caracteres reservados del lenguaje de programación Java • Al presionarse el botón Guardar surge una nueva interfaz con un mensaje y los botones: Cancelar para suspender el proceso de modificación de la actualización y Aceptar para aprobarlo 	
Prototipo de interfaz:	

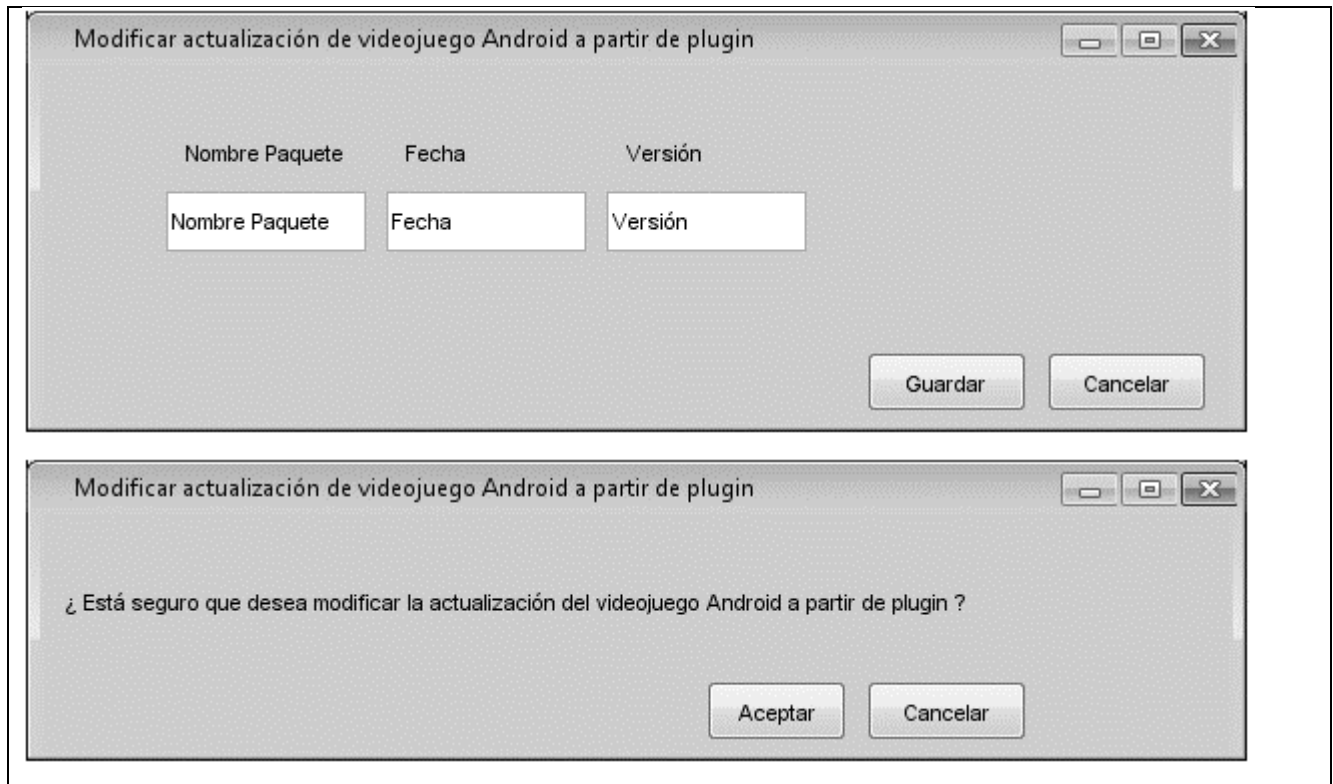


Tabla 44. HU_16 (Fuente: Elaboración propia)

Historia de Usuario	
Número: HU_16	Nombre: Eliminar actualización de videojuego Android a partir de <i>plugin</i>
Usuario: Administrador	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1.2	Iteración Asignada: 3
Programador responsable: René Orta Martínez	
Descripción: Inicia al presionar el botón Eliminar, a continuación surge una nueva interfaz con un mensaje y los botones Cancelar para suspender el proceso de eliminación de la actualización del videojuego Android a partir de plugin y Aceptar para aprobarlo	
Observaciones:	
<ul style="list-style-type: none"> • Previamente se debe seleccionar alguna actualización de videojuego Android a partir de 	



Tabla 45. HU_17 (Fuente: Elaboración propia)

Historia de Usuario	
Número: HU_17	Nombre: Cargar archivo de actualización de videojuego Android a partir de <i>plugin</i>
Usuario: Administrador	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración Asignada: 3
Programador responsable: René Orta Martínez	
Descripción: Inicia cuando el usuario presiona el botón <i>Buscar</i> durante el proceso de creación de las actualizaciones de videojuegos Android a partir de <i>plugin</i> . Se muestra un explorador donde el usuario selecciona el archivo que desea adjuntar a la actualización del videojuego	
Observaciones:	
Prototipo de interfaz:	

