

Universidad de las Ciencias Informáticas
Facultad 4



Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Título: Aplicación Xedro-GESPRO para dispositivos móviles
versión 2.0

Autor: Josué Pérez Núñez

Tutor: Ing. Julio Cesar Espronceda Pérez

La Habana, 2017

Declaración de Autoría

Declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas hacer uso del mismo en su beneficio.

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

Josué Pérez Núñez

Autor

Ing. Julio Cesar Espronceda Pérez

Tutor

**Hasta que el pensamiento no este
acompañado de un propósito no habrá
logro inteligente alguno.**

James Allen



Agradecimientos

En especial agradezco a mis padres por todo el esfuerzo que han realizado para poderme ayudar a lograr esta meta alcanzada, gracias por la confianza depositada en mí y por demostrarme que todo en la vida se puede lograr.

A mi hermano José Manuel gracias por formar parte de mi vida a pesar de que no nos une lazos de sangre quiero que sepas que serás una persona especial en mi vida.

A mi tutor gracias por todo el apoyo que me ha brindado que sin usted no hubiera sido posible este triunfo, gracias por demostrarme que un tropiezo no significa nada porque la vida está hecha de errores y a pesar de todas las dificultades que se presentaron para el desarrollo de la tesis usted me enseñó a no rendirse nunca.

A mis amigos por formar parte de estos hermosos 5 años en la universidad junto a mí donde compartimos momentos malos y buenos que nunca olvidaré porque son momentos que no se repiten ni, aunque pasen los siglos.

Resumen

El mundo actual es un mundo en constante evolución y desarrollo en el campo de la programación de dispositivos móviles su incremento ha propiciado que sean objetos del diario vivir. En la actualidad su importancia se basa en llevar a donde queramos nuestros programas favoritos, una función importante que tienen estos dispositivos es que nos facilitan nuestro trabajo ya sea con un editor de texto que nos permita realizar una programación de nuestras tareas, como una aplicación que nos permita introducir las tareas que debemos realizar, modificarlas o eliminarlas, así como poder visualizar su estado. Actualmente en la Universidad de las Ciencias Informáticas (UCI) existe una aplicación para la gestión de proyectos denominada Xedro-GESPRO v1.0 en su versión móvil que permite al usuario poder visualizar sus proyectos como las peticiones asignadas al mismo además de incorporar un cronograma de las tareas, sin embargo esta aplicación no dispone de los restantes indicadores asociados a los proyectos asignados de acuerdo al rol del usuario ni permite una adaptabilidad a versiones inferiores de la v4.4. para sistemas operativos android. Por ello el departamento de GESPRO identificó como necesidad seguir desarrollando la aplicación sobre dispositivos móviles basándose en los indicadores restantes para permitir su visualización independientemente de su estado de conectividad y garantizar su portabilidad hacia versiones inferiores.

Palabras Claves:

Dispositivos Móviles, Gestión de Proyectos, Xedro-GESPRO, Indicadores

Índice

Introducción.....	- 1 -
Capítulo 1: Fundamento teórico	- 7 -
1.1 Gestión de Proyectos	- 7 -
1.1.1 Áreas de conocimiento de la Gestión de Proyectos	- 7 -
1.2 Tipos de aplicaciones móviles	- 8 -
1.2.1 Aplicación nativa	- 8 -
1.2.2 Aplicación web o <i>webapp</i>	- 8 -
1.2.3 Aplicación híbrida	- 9 -
1.3 Tecnologías utilizadas en el desarrollo de la aplicación.	- 11 -
1.3.1 Metodología de desarrollo.....	- 11 -
1.3.2 Herramientas utilizadas.....	- 14 -
1.3.3 Lenguajes para el modelado y desarrollo.....	- 15 -
1.3.4 Frameworks	- 17 -
Conclusiones Parciales.....	- 18 -
Capítulo 2: Propuesta de Solución	- 19 -
2.1 Descripción de la propuesta de solución	- 19 -
2.2 Requisitos.....	- 19 -
2.2.1 Técnicas de Obtención de Requisitos (Ver anexo 1)	- 19 -
2.2.2 Requisitos Funcionales	- 20 -
2.2.3 Requisitos No Funcionales.....	- 21 -
2.3 Historia de Usuarios.....	- 22 -
2.4 Diseño.....	- 23 -
2.4.1 Arquitectura de la aplicación	- 23 -

2.5 Tarjetas CRC	- 25 -
2.6 Limitaciones.....	- 29 -
Conclusiones Parciales.....	- 29 -
3.1 Diagrama de componentes.....	- 30 -
3.2 Pruebas de Validación	- 32 -
3.3 Comparación entre la aplicación vigente y la propuesta de solución sobre entorno móvil.	- 34 -
3.4 Pruebas unitarias.....	- 36 -
3.5 Encuesta a especialistas	- 37 -
Conclusiones Parciales.....	- 38 -
Conclusiones Generales	- 39 -
Recomendaciones.....	- 40 -
Referencias	- 41 -
Anexos	- 44 -
Anexo 1. Encuesta de satisfacción	- 44 -
Anexo 2. Manual de instalación y configuración de las herramientas.....	- 44 -
utilizadas.....	- 44 -
Anexo 3. Manual de instalación y configuración de SQLite.	- 47 -
Anexo 4. Pasos para la sincronización de la base de datos SQLite con la base de datos de GESPRO versión web.....	- 49 -
Anexo 5. Manual de usuario.	- 50 -
Anexo 6. Historias de Usuarios.....	- 57 -
Anexo 7.Casos de Pruebas	- 67 -

Índice de Tablas

Tabla 1. Comparación de los tipos de aplicaciones móviles	- 10 -
Tabla 2 HU_AutenticarUsuario.....	- 22 -
Tabla 3 Tarjeta CRC "Login"	- 26 -
Tabla 4 Tarjeta CRC "Proyecto"	- 26 -
Tabla 5 Tarjeta CRC "Petición".....	- 26 -
Tabla 6 Tarjeta CRC "Estado General "	- 27 -
Tabla 7 Tarjeta CRC "Costo "	- 27 -
Tabla 8 Tarjeta CRC "Tiempo "	- 27 -
Tabla 9 Tarjeta CRC "Recursos Humanos "	- 28 -
Tabla 10 Tarjeta CRC "Riesgo "	- 28 -
Tabla 11 Tarjeta CRC "AuthProyectos "	- 29 -
Tabla 12 Tarjeta CRC "DB "	- 29 -
Tabla 13 Caso de Prueba Autenticar Usuario	- 32 -
Tabla 14: Comparación entre versiones para móviles de Xedro-Gespro	- 35 -
Tabla 15 Descripción de las tablas.....	- 49 -
Tabla 16 HU_VisualizarProyectos.....	- 57 -
Tabla 17 HU_Visualizar_ Peticiones	- 58 -
Tabla 18 HU_Estado General	- 59 -
Tabla 19 HU_Costo.....	- 61 -
Tabla 20 HU_Recursos Humanos.....	- 62 -
Tabla 21 HU_Riesgo	- 63 -
Tabla 22 HU_Tiempo	- 64 -
Tabla 23 HU_Calidad	- 66 -

Índice de Figuras

Figura 1 Diagrama de componentes	- 31 -
Figura 2 Autenticar Usuario Web	- 33 -
Figura 3 Autenticar Usuario Móvil	- 33 -
Figura 4 Autenticar Usuario Datos Incorrectos Web	- 33 -
Figura 5 Autenticar Usuario Datos Incorrectos Móvil	- 34 -
Figura 6 Errores en la 1ra iteración	- 36 -
Figura 7. Errores en la 2ra iteración	- 36 -
Figura 8 Errores en la 3ra iteración	- 37 -
Figura 9 Nivel de aceptación de la aplicación	- 37 -
Figura 10 Modelo de Datos	- 47 -
Figura 11 Página de Autenticación.....	- 51 -
Figura 12 Acción de Autenticar	- 51 -
Figura 13 Proyectos Asociados.....	- 52 -
Figura 14 Peticiones Asignadas	- 52 -
Figura 15 Estado del Proyecto	- 53 -
Figura 16 Menú desplegable	- 53 -
Figura 17 Indicadores en el menú	- 54 -
Figura 18 Indicador de Tiempo.....	- 54 -
Figura 19 Indicador de Costo	- 55 -
Figura 20 Indicador de Recursos Humanos	- 55 -
Figura 21 Indicador de Calidad	- 56 -
Figura 22 Indicador de Riesgo	- 56 -
Figura 23 Visualizar Proyectos Versión Web	- 68 -
Figura 24 Visualizar Proyectos Versión Móvil	- 68 -
Figura 25 Visualizar Estado General Versión Web	- 69 -
Figura 26 Visualizar Estado General Versión Móvil	- 69 -
Figura 27 Visualizar Costo Versión Web.....	- 70 -
Figura 28 Visualizar Costo Versión Móvil.....	- 71 -
Figura 29 Visualizar Tiempo Versión Web	- 72 -
Figura 30 Visualizar Tiempo Versión Móvil	- 72 -

Figura 31 Visualizar Recursos Humanos Versión Web.....	- 73 -
Figura 32 Visualizar Recursos Humanos Versión Móvil.....	- 74 -
Figura 33 Visualizar Indicadores de Calidad Versión Web.....	- 75 -
Figura 34 Visualizar Indicadores de Calidad Versión Móvil.....	- 75 -
Figura 35 Visualizar Indicadores de Riesgo Versión Web.....	- 76 -
Figura 36 Visualizar Indicadores de Riesgo Versión Móvil.....	- 77 -
Figura 37 Peticiones Versión Web	- 78 -
Figura 38 Peticiones Versión Móvil	- 78 -

Introducción

Actualmente la sociedad en la que vivimos se encuentra altamente ligada al uso de las tecnologías de la información y comunicación (TIC), así como en todos los ámbitos. Estamos hablando de la cultura informática que vive la sociedad de hoy día, a la cual todos los usuarios prestan atención para ver las tendencias que nos prepara la sociedad sobre las tecnologías en beneficios del hombre. Se trata de una sociedad en constante cambio, una sociedad que se mueve a gran velocidad, y que exige a los individuos un proceso de aprendizaje continuo no solo para su desempeño profesional sino para el pleno desarrollo de su vida cotidiana. Los individuos se ven obligados a adaptarse a situaciones cambiantes en todos los ámbitos de actuación humana y a adoptar nuevos conocimientos y competencias para hacer frente a dichos cambios. En definitiva, se trata de una sociedad del conocimiento que exige a los individuos gran capacidad de aprendizaje, adaptabilidad y flexibilidad. Los avances tecnológicos dan respuesta a las necesidades que plantea esta sociedad. Así, en una sociedad en movimiento surgen las tecnologías móviles para dar respuesta a las necesidades constantes de acceso a la información y de comunicación (Portal Educativo de las Américas, 2013).

En 1973, Martin Cooper inventó el primer celular. Como al principio eran tan grandes y caros, solo eran utilizados por los militares y algunas empresas. Para el 1983 eran más pequeños y económicos y podían ser utilizados por el público en general. El teléfono móvil o celular, fue inventado en 1947 por la empresa norteamericana AT&T, pero no se hizo portátil de manera práctica hasta 1983 cuando Motorola culmina el proyecto DynaTAC 8000X, el que es presentado oficialmente en 1984, a partir de ahí la telefonía móvil ha seguido revolucionando en la historia (Celular, 2014).

La tecnología móvil ha pasado a ser uno de los elementos imprescindibles para la vida social de cualquier persona, y eso es precisamente porque estos fabulosos dispositivos acercan a los usuarios una gran variedad de prestaciones, las cuales se incrementan día a día en los nuevos modelos de terminales, en las que se

incluyen funciones adicionales para ofrecer el servicio más completo al consumidor. Hoy no sólo podemos comunicarnos con alguien de forma convencional, como solíamos hacerlo con el tradicional teléfono fijo, no sólo entablamos una comunicación telefónica, sino que además podemos tomar fotografías, capturar vídeo, navegar por Internet, jugar, escuchar música y un sinnúmero de tareas que en la actualidad podemos desarrollar con la terminal. Todo ello producto a los dispositivos nominados móviles (Informaticahoy,2016).

Los dispositivos móviles no son sólo teléfonos portátiles con más funciones. Su funcionamiento y prestaciones se acercan más a “pequeños ordenadores”, donde realizar y recibir llamadas de teléfono es sólo una aplicación más entre muchas otras. Aplicaciones comunes en los teléfonos inteligentes son: llamar y recibir llamadas telefónicas, navegadores de Internet, accesos directos a Internet libros de referencia, traductores, reproductores de audio y vídeo, gestores de redes sociales, aplicaciones para acceder al correo electrónico, agendas, calendarios, juegos, además nos posibilita planificar nuestro trabajo ya sea desde un editor de texto, donde podamos introducir las tareas a realizar, como por una aplicación donde podamos revisar el estado de nuestras tareas (Navarra, 2015).

El empleo de dispositivos móviles es tan beneficioso como la presencia de computadoras en las empresas, es por ello que la Universidad de las Ciencias Informática (UCI) necesita expandir más su desarrollo de aplicaciones móviles. La UCI cuenta con diversos centros productivos que participan en el desarrollo de aplicaciones móviles entre los que se encuentran el Centro de Consultoría y Desarrollo de Arquitecturas Empresariales por sus siglas CDAE. Como parte de su estructura se encuentra el departamento de Gestión de Proyectos (GESPRO) donde fue desarrollada la aplicación denominada XEDRO-GESPRO en su versión móvil (Padron, 2017). Tiene como función principal visualizar los proyectos asignados de acuerdo a su rol, así como visualizar las peticiones y el cronograma de tareas asociados al proyecto.

XEDRO-GESPRO versión móvil es una aplicación totalmente desarrollada para entorno de telefonía móvil. El cronograma de tareas es el módulo más usado y con una simple consulta se puede acceder a las tareas planificadas en un proyecto para una persona determinada. En este sentido, entre las principales dificultades detectadas se encuentran:

- Los indicadores por las distintas áreas de recursos humanos, tiempo de ejecución, riesgo, costo, estado general, calidad no se evidencian en la propuesta de solución lo que imposibilita que no tengan visión sobre lo mismo por lo que no cuenta con las expectativas que se esperaban.
- Bajo rendimiento en los tiempos de respuesta de la aplicación propuesta por el número de recursos que se consumen.
- No es adaptable a versiones de sistema operativo androide inferiores a la versión 4.4 lo que implica que aquellos usuarios con versiones inferiores no puedan acceder a la aplicación ya que limita la necesidad a la fuente de información y obliga a las personas a una actualización que no siempre está acorde con su valor adquisitivo.

Debido a la situación descrita anteriormente, sale a relucir el siguiente problema a resolver: ¿Cómo contribuir a la solución de las principales dificultades de Xedro-Gespro en su versión móvil?

Este problema se enmarca en el **objeto de estudio**: Aplicación de gestión para dispositivos móviles, dentro del que se ha identificado como **campo de acción**: Aplicación de XEDRO-GESPRO para dispositivos móviles.

Se define como **objetivo** de este trabajo: Desarrollar una versión superior de Xedro-Gespro versión móvil que dé respuesta a las principales dificultades detectadas en la versión anterior. Dicho esto, tenemos como **posibles resultados**:

- 1- Debe quedar elaborado un informe detallado con toda la base teórica-práctica sobre la cual se sustenta la solución propuesta.
- 2- Quedar implementada una nueva versión de Xedro-Gespro para dispositivos móviles que tenga en cuenta los indicadores de Estado

General, Calidad, Recursos Humanos, Costo, Tiempo, Riesgo, así como garantizar mejores tiempos de respuesta y que se encuentre disponible la aplicación para versiones inferiores a la v.4.4. para dispositivos móviles.

Idea a defender:

El desarrollo de una nueva versión de Xedro-Gespro para dispositivos móviles debe resolver las deficiencias detectadas en la aplicación desarrollada en años anteriores, en cuanto a rendimiento, áreas del conocimiento y adaptabilidad.

Para dar cumplimiento al objetivo, se ha organizado el trabajo a partir de las siguientes **tareas de investigación**:

1. Elaboración de un marco conceptual para precisar los principales conceptos que se emplean en la investigación.
2. Estudio del estado del arte de las herramientas existentes para desarrollo e implementación de una aplicación de tecnología móvil que le permita al usuario el fácil acceso a la información disponible sobre la gestión de proyectos de su entidad.
3. Selección de la metodología para definir los métodos y técnicas necesarias que guiarán el desarrollo.
4. Descripción de las herramientas, tecnologías y lenguajes a utilizar para definir el ambiente de desarrollo.
5. Definición de los requisitos funcionales y no funcionales para identificar las capacidades que tienen que ser alcanzadas por el sistema para cumplir los objetivos trazados.
6. Descripción de la arquitectura que soporta la implementación de las funcionalidades.
7. Realización del análisis y diseño del mecanismo para describir los artefactos.
8. Implementación de las funcionalidades que dan cumplimiento a los requisitos identificados.

9. Realización de pruebas a la propuesta de solución en un entorno controlado para valorar la calidad del producto implementado.
10. Valoración de los resultados obtenidos.

Para alcanzar el objetivo propuesto se utilizan los siguientes métodos científicos:

Métodos teóricos:

Análisis histórico-lógico: Este método se empleó para la fundamentación de los aspectos teóricos contemplados en el desarrollo de la investigación acerca de la Gestión de Proyecto y desarrollo de aplicaciones para dispositivos móviles.

Modelación: Método teórico que ofrece parte de la información necesaria acerca del objeto que se estudia, se trata de explicar la realidad con la creación de diagramas; los cuales pueden ser presentados en sustitución de la realidad. Mediante su utilización se elaborarán diferentes tipos de diagramas que brindarán información clara sobre el tema de estudio mediante el descubrimiento de nuevas relaciones y cualidades del objeto de estudio.

Métodos empíricos:

Encuesta: Aplicada para obtener información de los especialistas y usuarios de la aplicación, para realizar análisis del grado de aceptación, aplicabilidad y utilidad de la propuesta.

La Observación Científica: Permite valorar los avances realizados en el estudio de las diferentes herramientas y tecnologías a usar.

El método de Consulta de la Información: Utilizado para elaborar el marco teórico o el estado del arte de la investigación, permitiendo el conocimiento y el acceso a las referencias bibliográficas de los múltiples criterios que han sido citados para comprender mejor el problema de investigación planteado.

Estructura de la tesis:

El trabajo está estructurado en 3 capítulos, tal y como se describe a continuación:

Capítulo 1: Fundamento teórico: En este capítulo se presenta la definición del marco teórico de la investigación, se fundamenta el uso de los distintos temas referentes a la metodología, herramientas y tecnologías a utilizar para el desarrollo del sistema XEDRO-GESPRO para dispositivos móviles.

Capítulo 2: Propuesta de solución: En este capítulo se presenta los requisitos funcionales y no funcionales, así como también se muestran las historias de usuarios, tarjetas CRC y el modelo de datos los cuales serán de ayuda para la implementación de la aplicación, finalmente se concluye con una sección de bondades y limitaciones.

Capítulo 3: Validación y pruebas: En este capítulo se presenta los componentes que tiene la aplicación mediante un diagrama, además se exponen los resultados de las pruebas realizadas a la solución.

Capítulo 1: Fundamento teórico

En este capítulo se abordan las tecnologías existentes en la actualidad para el desarrollo de aplicaciones móviles vinculadas al campo de acción, semejanzas entre los diferentes tipos de aplicaciones y sus rasgos fundamentales, para de esta forma entender la necesidad de una nueva solución acorde a los requerimientos que en la actualidad presenta la institución.

1.1 Gestión de Proyectos

La gestión de proyectos es la aplicación de conocimientos, habilidades, herramientas y técnicas para proyectar actividades que cumplan requerimientos del proyecto. La gestión del proyecto se lleva a cabo mediante la aplicación adecuada y la integración de los procesos de gestión de proyectos identificados para el proyecto (PMBOOK, 2017).

1.1.1 Áreas de conocimiento de la Gestión de Proyectos

Un área de conocimiento es, según PMBOK (*Project Management Body of Knowledge*), “un área identificada de la dirección de proyectos definida por sus requisitos de conocimientos y que se describe en términos de sus procesos, prácticas, datos iniciales, resultados, herramientas y técnicas que los componen.”

Según el PMBOK_6ta Edición la gestión de proyectos tiene 10 áreas las cuales son:

1. Administración de la integración
2. Administración del alcance
3. Administración del cronograma
4. Administración de los costos
5. Administración de la calidad
6. Administración de los recursos
7. Administración de las comunicaciones
8. Administración de los riesgos
9. Administración de las adquisiciones

10. Administración de los interesados (PMBOOK, 2017).

De las áreas del conocimiento expuestas en la tesis de años anteriores solo se centra en el área del conocimiento de la gestión del alcance por lo que serán incluidas las áreas de administración de los costos, administración de calidad, administración de los recursos, administración de los riesgos formando parte de los indicadores a visualizar en la propuesta de solución.

1.2 Tipos de aplicaciones móviles

Existen 3 tipos de aplicaciones móviles que serán abordadas en los epígrafes siguientes.

1.2.1 Aplicación nativa

Una aplicación nativa es la que se desarrolla de forma específica para un determinado sistema operativo, llamado *Software Development Kit* o *SDK*. Cada una de las plataformas, *Android*, *iOS* o *Windows Phone*, tienen un sistema diferente, por lo que si quieres que tu aplicación esté disponible en todas las plataformas se deberán de crear varias aplicaciones con el lenguaje del sistema operativo seleccionado (LanceTalent, 2015).

Cuando hablamos de desarrollo móvil casi siempre nos estamos refiriendo a aplicaciones nativas. La principal ventaja con respecto a los otros dos tipos, es la posibilidad de acceder a todas las características del hardware del móvil: cámara, GPS, agenda, dispositivos de almacenamiento y otras muchas. Esto hace que la experiencia del usuario sea mucho más positiva que con otro tipo de aplicaciones. Además las aplicaciones nativas no necesitan conexión a internet para que funcionen (LanceTalent, 2015).

1.2.2 Aplicación web o *webapp*

Una aplicación web o *webapp* es la desarrollada con lenguajes muy conocidos por los programadores, como es el HTML, *Javascript* y CSS. La principal ventaja con respecto a la nativa es la posibilidad de programar independiente del sistema

operativo en el que se usará la aplicación. De esta forma se pueden ejecutar en diferentes dispositivos sin tener que crear varias aplicaciones (LanceTalent, 2015).

Las aplicaciones web se ejecutan dentro del propio navegador web del dispositivo a través de una URL. En realidad, la gran diferencia con una aplicación nativa es que no necesita instalación por lo que no pueden estar visibles en tienda de aplicaciones y la promoción y comercialización debe realizarse de forma independiente. De todas formas, se puede crear un acceso directo que sería como “instalar” la aplicación en el dispositivo. Las aplicaciones web móviles son siempre una buena opción si nuestro objetivo es adaptar la web a formato móvil (LanceTalent, 2015).

1.2.3 Aplicación híbrida

Una aplicación híbrida es una combinación de las dos anteriores, se podría decir que recoge lo mejor de cada una de ellas. Las aplicaciones híbridas se desarrollan con lenguajes propios de las *webabpp*, es decir, HTML, *Javascript* y CSS por lo que permite su uso en diferentes plataformas, pero también dan la posibilidad de acceder a gran parte de las características del hardware del dispositivo. La principal ventaja es que a pesar de estar desarrollada con HTML, Java o CSS, es posible agrupar los códigos y distribuirla en tienda de aplicaciones (LanceTalent, 2015).

PhoneGap es uno de los *frameworks* más utilizados por los programadores para el desarrollo multiplataforma de aplicaciones híbridas. Un ejemplo claro para desarrollar aplicaciones híbridas es el empleo de la herramienta Cordova (LanceTalent, 2015).

A continuación, en la **Tabla 1** se muestra una comparación entre los diferentes tipos de aplicaciones móviles teniendo en cuenta cuatro aspectos fundamentales que permiten identificar cual es la más adecuada para desarrollar la solución propuesta.

Tabla 1. Comparación de los tipos de aplicaciones móviles

Tipo de aplicación	Reutilización de código	Publicación de servicios en la web	Multiplataforma	Portabilidad
Nativa	El código no es reutilizable para cada una de las plataformas.	No se puede publicar servicios en la web ya que la aplicación no se ejecuta a través de un navegador	Solo se puede ejecutar en la plataforma para la cual fue programada.	La aplicación nativa es portable.
Webapp	El código es reutilizable para cada una de las plataformas.	Se pueden publicar servicios en la web ya que la aplicación se ejecuta sobre el navegador.	Es multiplataforma, ya que no importa la plataforma donde se está ejecutando la aplicación ya que se ejecuta a través del navegador.	La aplicación Webapp no es portable.
Híbrida	El código es reutilizable para cada una de las plataformas.	Si se pueden publicar servicios en la web porque este tipo de aplicación se instala como una aplicación nativa pero se programa con lenguajes propios de las aplicaciones web, además de que pueden ser ejecutadas en la web.	Es multiplataforma porque con un único código se puede compilar la aplicación para todas las plataformas.	La aplicación híbrida es portable.

Luego de haber analizado la **Tabla 1** se ha decidido que la aplicación a desarrollar sea híbrida porque brinda la posibilidad de que sea funcional en todos los sistemas operativos; se debe destacar que la Universidad de las Ciencias Informáticas no cuenta con las licencias existentes para los demás sistemas operativos por lo que la aplicación propuesta se desarrollará para el sistema operativo Android, quedando

disponible el código para cuando la universidad adquiriera las licencias se pueda generar para los demás sistemas operativos.

1.3 Tecnologías utilizadas en el desarrollo de la aplicación.

En este epígrafe se expone la metodología que guiará el proceso de desarrollo del *software* y los conceptos relacionados a las herramientas y tecnologías empleadas en la aplicación Xedro-Gespro versión móvil. Debido a que estas herramientas y tecnologías favorecen el incremento de la productividad, disminuyendo así el tiempo de desarrollo.

1.3.1 Metodología de desarrollo

Una Metodología de desarrollo de software, consiste principalmente en hacer uso de diversas herramientas, técnicas, métodos y modelos para el desarrollo. Regularmente este tipo de metodología tienen la necesidad de venir documentadas para que los programadores que estarán dentro de la planeación del proyecto, comprendan perfectamente la metodología y en algunos casos el ciclo de vida del *software* que se pretende seguir (Software, 2015) . Existen dos tipos de metodologías, las Ágiles y las Pesadas o Tradicionales; de las anteriormente mencionadas nos centraremos en las ágiles ya que para el proceso de desarrollo se utilizará una metodología Ágil.

La metodología ágil o *agile* en inglés es una metodología de gestión de proyectos que utiliza ciclos de desarrollo cortos llamados *sprints* para centrarse en la mejora continua del desarrollo de un producto o servicio, más que centrarse en la gestión del propio proyecto (recursosenprojectmanagement, 2018).

Entre las principales ventajas de la metodología ágil podemos destacar:

1. Permite obtener un producto funcional, aunque a veces con limitaciones, de forma más rápida. Esto se traduce en un retorno de la inversión mayor, y en la posibilidad de probar el producto con clientes reales antes de completar el proyecto.

2. Optimización de los recursos al focalizarse en tareas y partes pequeñas del proyecto, más simples de cuantificar y gestionar.
3. Mayor flexibilidad y capacidad para adaptarse a los cambios.
4. Mayor posibilidad de éxito del proyecto debido a la mayor capacidad y velocidad de adaptación, y su mayor foco en el cliente y el negocio.
5. Es una metodología menos formal y menos basada en procesos, por lo que es difícil de aplicar en organizaciones grandes y tradicionales.
6. Su foco en objetivos pequeños y concretos, base de su flexibilidad, hace que se pierda la visión global del proyecto. Esto puede ser un problema en proyectos con objetivos muy claros que requieran decidir sobre grandes inversiones desde su inicio (recursosenprojectmanagement, 2018).

Como seguimiento al trabajo de diploma realizado en años anteriores se propone utilizar como metodología de desarrollo AUP-UCI, la cual es una variante de la metodología Proceso Unificado Ágil (AUP por sus siglas en inglés), de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI. La misma está compuesta por tres fases: Inicio, Ejecución y Cierre que en su conjunto ayudan a mejorar el proceso de desarrollo de *software* y dan cumplimiento además a las buenas prácticas que define CMMI-DEV v1.3(*Capability Maturity Model Integration for Development*) (Silva, 2015). La UCI decide para el ciclo de vida de los proyectos mantener la fase de Inicio, en la cual se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto. Se unifican las restantes tres fases de AUP en una sola, a la que se denomina Ejecución que sería aquella fase en la que se ejecutan las actividades requeridas para desarrollar el *software*, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.

AUP propone siete disciplinas (Modelo, Implementación, Prueba, Despliegue, Gestión de configuración, Gestión de proyecto y Entorno), se decide para el ciclo de

vida de los proyectos de la UCI tener ocho disciplinas, pero a un nivel más atómico que el definido en AUP. Los flujos de trabajos: Modelado de negocio, Requisitos y Análisis y diseño en AUP están unidos en la disciplina Modelo, en la variación para la UCI se consideran a cada uno de ellos disciplinas. Se mantiene la disciplina Implementación, en el caso de Prueba se desagrega en tres disciplinas: Pruebas Internas, de Liberación y Aceptación y la disciplina Despliegue se considera opcional. Las restantes tres disciplinas de AUP asociadas a la parte de gestión, para la variación UCI, se cubren con las áreas de procesos que define CMMI-DEV (*Capability Maturity Model Integration for Development*) v1.3 para el nivel dos, serían CM (Gestión de la configuración), PP (Planeación de proyecto) y PMC (Monitoreo y control de proyecto) (Silva, 2015). En cada disciplina la metodología plantea las diferentes actividades y artefactos a producir, lo cual no implica que se realicen o se produzcan todos los planteados sino más bien lo que se necesita en el proyecto.

Ventajas de la Metodología AUP (Sinapps, 2016):

- Simplicidad: todo se describe concisamente utilizando un puñado de páginas, no miles de ellos.
- Permite hacer uso de un amplio conjunto de herramientas. Lo aconsejable es utilizar las que son las más adecuadas para el trabajo, que a menudo son las más simples o incluso herramientas de código abierto.
- Incluye explícitamente actividades y artefactos a los que la mayoría de desarrolladores ya están, de alguna manera, acostumbrados.
- Se preocupa especialmente de la gestión de riesgos. Propone que aquellos elementos con alto riesgo obtengan prioridad en el proceso de desarrollo y sean abordados en etapas tempranas del mismo.

Desventajas de la Metodología AUP (Sinapps, 2016):

- Como es un proceso simplificado, muchos desarrolladores eligen trabajar con RUP, por tener a disposición más detalles en el proceso.

Por los recursos, la documentación necesaria, el escaso tiempo para la realización del proyecto, y personal, se decide escoger esta metodología en su escenario 4,

para guiar el proceso de desarrollo de la solución propuesta, adaptada al ciclo de vida definido para la actividad productiva de la UCI.

1.3.2 Herramientas utilizadas

Para la realización de la aplicación se hizo uso de las siguientes herramientas propuestas (Ver [Anexo 2](#)):

Visual Paradigm for UML 8.0

Este es un *software* de modelado del lenguaje Unificado (UML) que nos permite analizar, diseñar, codificar, probar y desplegar. Dibuja todo tipo de diagramas UML, genera código fuente a partir de dichos diagramas (LanceTalent, 2015).

Visual Paradigm para UML *Enterprise Edition* soporta: UML, SysML, ERD, BPMN, DFD, ArchiMate, diagramas, entre otros. El programa cuenta con innumerables ventajas; una de las más importantes es la aptitud para representar todas las funciones posibles (CCM, 2015).

Para la realización del sistema propuesto es aconsejable que se trabaje con esta herramienta, pues permite diseñar y modelar un conjunto de diagramas que servirán como puntos de partida para el desarrollo de las tareas de implementación propuestas.

Visual Studio Code

Visual Studio Code es un nuevo tipo de herramienta que combina la simplicidad de un editor de código con lo que los desarrolladores necesitan para su ciclo central de edición-compilación-depuración. *Code* proporciona soporte completo de edición y depuración, un modelo de extensibilidad e integración liviana con herramientas existentes (Github, 2018).

SQLite (Ver [Anexo 3](#) y [Anexo 4](#))

SQLite es sin duda uno de los motores de base de datos más interesante, las bases de datos generadas por SQLite son como un archivo más en tu dispositivo de almacenamiento, y que dicho archivo contiene tanto los datos almacenados como la estructura de base de datos relacionados definida en su creación. SQLite ofrece

un rendimiento más que notable, permitiendo realizar operaciones tanto en disco como en memoria para proporcionar respuestas a consultas complejas en cuestión de pocos milisegundos (PhoneGapSpain, 2016).

Jasmine

Es un marco de desarrollo utilizado para probar código *JavaScript* no requiere ninguna dependencia de ningún otro *framework* de *JavaScript*. Por lo que su sintaxis clara permite realizar las pruebas con una gran simplicidad.

Karma

Es una herramienta que genera un servidor web que permite emular el código fuente de la aplicación contra el código de prueba para los navegadores instalados en el ordenador. Cada resultado se examina y se muestra a través del navegador para permitir al usuario visualizar que pruebas y navegadores fallaron en las iteraciones realizadas.

1.3.3 Lenguajes para el modelado y desarrollo

UML

El Lenguaje Unificado de Modelado fue creado para forjar un lenguaje de modelado visual común, semántica y sintácticamente rico para la arquitectura, el diseño y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento. UML tiene aplicaciones más allá del desarrollo de software, en el flujo de procesos en la fabricación (Lucidchart, 2017).

Es comparable a los planos usados en otros campos y consiste en diferentes tipos de diagramas. En general, los diagramas UML describen los límites, la estructura y el comportamiento del sistema y los objetos que contiene (Lucidchart, 2017).

UML no es un lenguaje de programación, pero existen herramientas que se pueden usar para generar código en diversos lenguajes usando los diagramas UML que a

su vez guarda una relación directa con el análisis y el diseño orientados a objetos (Lucidchart, 2017).

HTML5

HTML5 es un lenguaje margen (de hecho, las siglas de HTML significan *Hyper Text Markup Language*) usado para estructurar y presentar el contenido para la web. Es uno de los aspectos fundamentales para el funcionamiento de los sitios, pero no es el primero (Hipertextual, 2013).

Se trata de un sistema para formatear el diseño de nuestras páginas, así como hacer algunos ajustes a su aspecto. Con HTML5, los navegadores como Firefox, Chrome, Explorer, Safari y más pueden saber cómo mostrar una determinada página web, saber dónde están los elementos, dónde poner las imágenes, dónde ubicar el texto. En este sentido, el HTML5 no se diferencia demasiado de su predecesor. La diferencia principal, sin embargo, es el nivel de sofisticación del código que podremos construir usando HTML5 (hipertextual, 2013).

TypeScript

TypeScript es un lenguaje de programación de alto nivel que implementa muchos de los mecanismos más habituales de la programación orientada a objetos, pudiendo extraer grandes beneficios que serán especialmente deseables en aplicaciones grandes, capaces de escalar correctamente durante todo su tiempo de mantenimiento. La característica fundamental de *TypeScript* es que compila en *Javascript* nativo, por lo que se puede usar en todo proyecto donde se esté usando *Javascript* (Guerra, 2016).

1.3.4 Frameworks

Ionic

Ionic es una herramienta, gratuita y de código abierto, para el desarrollo de aplicaciones híbridas basadas en HTML5, CSS y JS. Está construido con Sass y optimizado con AngularJS (PhoneGapSpain, 2016).

Principales características

1.- Alto rendimiento Ionic está construido para ser rápido gracias a la mínima manipulación del DOM, con cero jQuery y con aceleraciones de transiciones por hardware (PhoneGapSpain, 2016).

2.- Ionic utiliza AngularJS con el fin de crear un marco más adecuado para desarrollar aplicaciones ricas y robustas. Ionic no sólo se ve bien, sino que su arquitectura central es robusta y seria para el desarrollo de aplicaciones. Trabaja perfectamente con AngularJS (PhoneGapSpain, 2016).

3.- Ionic se inspira en las SDK de desarrollo móviles nativos más populares, por lo que es fácil de entender para cualquier persona que ha construido una aplicación nativa para *iOS* o *Android* (PhoneGapSpain, 2016).

4.- Limpio, sencillo y funcional. *Ionic* ha sido diseñado para poder trabajar con todos los dispositivos móviles actuales. Con muchos componentes usados en móviles, tipografía, elementos interactivos (PhoneGapSpain, 2016).

Apache Cordova

Apache Cordova es un marco de desarrollo móvil de código abierto. Permite utilizar las tecnologías estándar web como HTML5, CSS3 y *JavaScript* para desarrollo multiplataforma, evitando el lenguaje de desarrollo nativo de cada plataforma móvil. Las aplicaciones ejecutan dentro de envolturas para cada plataforma y dependen de enlaces estándares API para acceder a de cada dispositivo sensores, datos y estado de la red (Cordova, 2015).

La aplicación propuesta será desarrollada sobre la plataforma *Android* solamente, al no existir las prestaciones necesarias para las plataformas restantes.

Conclusiones Parciales

El análisis de Xedro-Gespro v1.0 para dispositivos móviles permitió identificar:

- Carencia y limitaciones entre dispositivos móviles con respecto a la información referente a la Gestión de Proyectos.
- Para la presente investigación se tomarán las áreas del conocimiento de la administración del alcance, administración de los recursos, administración de la calidad, administración de los costos, administración de los riesgos y se tomara como referencia en su versión web.
- AUP-UCI como metodología adaptada al proceso de desarrollo de software de la UCI.
- Visual Paradigm como herramienta de diseño de aplicaciones mediante diagramas de casos de usos, prototipo de interfaz, entre otros.
- Apache Cordova para la creación de aplicaciones sobre distintas plataformas móviles utilizando como lenguajes de programación HTML, CSS y JS.

Capítulo 2: Propuesta de Solución

En el presente capítulo se dará a conocer la propuesta del sistema y los diagramas para apoyar la comprensión del funcionamiento del mismo, tomándose como base la metodología AUP para guiar el proceso de desarrollo. Esta metodología propone diferentes artefactos, la presente investigación utiliza los mencionados para la implementación del *software* y se evidencia la arquitectura a utilizar.

2.1 Descripción de la propuesta de solución

La aplicación de Xedro-Gespro para dispositivos móviles en su versión 2.0 se desarrolla mediante el *framework Ionic* 3.0 bajo un entorno de conectividad con dominio uci.cu. Donde se realizan peticiones mediante HTTP hacia los servicios desplegados con estructura JSON, el usuario debe completar el formulario de autenticación y se realiza la petición al servidor para verificar si las credenciales son correctas y obtener la llave correspondiente, una vez autenticado se despliega la interfaz asociada a los proyectos vinculados con la llave obtenida, donde se muestran los restantes indicadores de acuerdo al rol del usuario autenticado en el sistema. La información mostrada se almacena en la base de datos creada mediante SQLite, para que pueda ser visualizada independientemente del estado de conectividad en que se encuentre el dispositivo móvil. Para la descripción de la propuesta de solución se puede hacer uso del manual de usuario(Ver [Anexo 5](#))

2.2 Requisitos

2.2.1 Técnicas de Obtención de Requisitos (Ver anexo 1)

Un área de conocimiento de gran importancia en el desarrollo de software, es la ingeniería de requerimientos. Esta comprende las actividades de obtención (captura, descubrimiento y adquisición), análisis (y negociación), especificación, y validación de requisitos. Además, establece una actividad de gestión de requerimientos para manejar los cambios, mantenimiento y rastreabilidad de los requerimientos.

Existe un gran número de técnicas para obtener requerimientos. A continuación se describe las utilizadas para realizar la propuesta de solución:

La entrevista: De gran utilidad para obtener información cualitativa como opiniones, o descripciones subjetivas de actividades.

Tormenta de ideas: Esta técnica se puede utilizar para identificar un primer conjunto de requisitos en aquellos casos donde no están muy claras las necesidades que hay que cubrir, o cuando se está creando un sistema que habilitará un servicio nuevo para la organización (Guerra, 2016).

2.2.2 Requisitos Funcionales

Un requisito funcional define una función del sistema de software o sus componentes. Para ello se definen los siguientes requisitos funcionales:

RF1. Autenticar Usuario: Esta funcionalidad les permite a los usuarios una vez instalada la aplicación en su dispositivo móvil poder autenticarse para guardar sus credenciales brindando además la posibilidad de al conectarse en una red WI-FI poder visualizar sus datos.

RF2. Visualizar Proyectos: Esta funcionalidad les permite a los usuarios según los permisos que tenga, ver una lista de todos los proyectos que tiene asignado.

RF3. Visualizar Peticiones: Esta funcionalidad les permite a los usuarios ver las peticiones luego de seleccionar un proyecto.

RF4. Visualizar Estado General: Esta funcionalidad permite a los usuarios autenticados poder visualizar el estado en que se encuentra los proyectos que se le fueron asignados.

RF5. Visualizar Tiempo de Ejecución: Muestra el tiempo de ejecución en que se desarrollara el proyecto.

RF6. Visualizar Costo de Proyecto: Muestra información pertinente a los costos asociados al proyecto.

RF7. Visualizar Indicador de Riesgo: Muestra la *incertidumbre de forma sistemática para ver los resultados negativos que afronta el proyecto.*

RF8. Visualizar Indicador de Recursos Humanos: Muestra la información que describe el trabajo que aporta el conjunto de los empleados o colaboradores asignados al proyecto.

RF9. Visualizar Indicador de Calidad: Se visualiza las no conformidades asociadas a un proyecto específico asignado a un usuario.

2.2.3 Requisitos No Funcionales

Los requisitos no funcionales son propiedades y características que hacen la aplicación más atractiva, usable y confiable. Se refieren a todos los requisitos que ni describen información a guardar, ni funciones a realizar. Para la solución propuesta se definen los siguientes requisitos no funcionales:

RnF1. Requisito de Usabilidad

RnF1.1. El nombre de los botones debe tener relación con las funciones que se debe realizar

RnF1.2. La aplicación debe estar disponible todo el tiempo.

RnF1.3. La herramienta tiene que ser portable.

RnF1.4. La herramienta debe ser accesible desde el menú de aplicaciones del dispositivo móvil.

RnF2. Requisito de Interfaz y Apariencia externa.

RnF2.1. Debe de ser semejante al sitio de Xedro-GESPRO versión web y poseer su logo.

RnF3. Requisito de Seguridad

RnF3.1. El sistema debe permitir el acceso a la información según el nivel de privilegios de cada rol.

RnF4. Requisito de software

Requisitos mínimos para el dispositivo móvil:

RnF4.1. Soporte para conexiones WIFI.

RnF4.2. Sistema operativo Android v4.0 ó superior.

RnF5. Restricción del diseño y la implementación

RnF5.1. Visual Paradigm como herramienta para el modelado.

RnF5.2 Ionic 3 como *framework* para la implementación de la propuesta de solución.

RnF5.3 Metodología AUP v. UCI para guiar el proceso de desarrollo.

2.3 Historia de Usuarios

Las Historias de Usuario son peticiones muy sencillas y claras que definen cuales van a ser los requisitos del proyecto además describen la funcionalidad de algo que es valioso para un usuario de un sistema o software (LLopiz, 2016). A continuación se evidencia la historia de usuario sobre el requisito funcional Autenticar Usuario las restantes historias de usuario podrán ser consultada en los anexos (Ver [Anexo 6](#))

Tabla 2 HU_AutenticarUsuario

Historia de Usuario	
Número: 1	Nombre del requisito: AutenticarUsuario
Programador: Josué Pérez Núñez	Iteración Asignada: 1era
Prioridad: Alta	Tiempo Estimado: 20 días
Riesgo en Desarrollo: N/A	Tiempo Real: 18 días
Descripción:	
1- Objetivo:	
<i>Permitir al usuario autenticarse en la aplicación.</i>	
2- Acciones para lograr el objetivo (precondiciones y datos):	
Para autenticarse:	
<i>- El usuario debe de estar registrado en el sistema.</i>	
3- Flujo de la acción a realizar:	
<i>Cuando el usuario se autentica podrá ver la lista de los proyectos a los cuales está vinculado.</i>	
Observaciones: N/A	

Prototipo de interfaz:

El prototipo de interfaz de usuario muestra una ventana de login con el título "Xedro-Gespro". Dentro de la ventana, hay un formulario con los siguientes elementos:

- Un campo de texto etiquetado "Usuario".
- Un campo de texto etiquetado "Password".
- Un campo de texto etiquetado "URL".
- Un botón etiquetado "Acceder".

La ventana también tiene un botón "Login" en la esquina superior izquierda y botones de control de ventana (minimizar, maximizar, cerrar) en la esquina superior derecha.

2.4 Diseño

El diseño de *software* agrupa el conjunto de principios conceptos y prácticas que llevan al desarrollo de un sistema o producto de alta calidad, el objeto del diseño es producir un modelo representación que tenga resistencia funcionalidad y belleza el modelo del diseño proporcionando detalles sobre la arquitectura del software, estructuras de datos, interfaces y componentes que se necesitan para implementar el sistema. (Ing.Marco, 2014)

2.4.1 Arquitectura de la aplicación

MVVM significa Modelo, Vista, Vista Modelo, en este patrón de diseño se separan los datos de la aplicación, la interfaz de usuario, pero en vez de controlar manualmente los cambios en la vista o en los datos, estos se actualizan directamente cuando sucede un cambio en ellos. (Matias, 2017)

La finalidad principal del patrón MVVM (Modelo Vista Vista-Modelo) es tratar de desacoplar lo más que se pueda la interfaz de usuario de la lógica de la aplicación. (Matias, 2017)

La vista (View):

La misión de la vista es representar la información a través de los elementos visuales que la componen. Las vistas en MVVM son activas, contienen comportamientos, eventos y enlaces a datos que, en cierta manera, necesitan tener conocimiento del modelo subyacente. (Matias, 2017)

Modelo:

Se encarga de contener la información de los servicios a consumir pero nunca representa las acciones que lo manipulan.

Modelo de vista (ViewModel):

El *ViewModel* (modelo de vista) es un actor intermediario entre el modelo y la vista, contiene toda la lógica de presentación y se comporta como una abstracción de la interfaz. La comunicación entre la vista y el modelo de vista se realiza por medio de los enlaces de datos (*binders*). (Matias, 2017)

Este Patrón nos da ciertas ventajas:

1. Separar el desarrollo de la interfaz de usuario del resto del código. Es decir, podemos tener un equipo de diseño trabajando en la interfaz de usuario y a los programadores haciendo el resto de la aplicación.
2. La lógica de presentación puede ser probada con pruebas unitarias (unit testing) al estar desacoplada de nuestras vistas.
3. Muy útil cuando estamos desarrollando aplicaciones multiplataforma, ya que tanto la lógica de negocio, como la lógica de presentación es común a todas las plataformas (Matias, 2017).

Un patrón es un par problema/solución con nombre que se puede aplicar en nuevos contextos, con consejos acerca de cómo aplicarlo en nuevas situaciones y discusiones sobre sus compromisos (Larman, 2014).

Los patrones GRASP (Patrones Generales de Software para Asignar Responsabilidades) describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades, expresados como patrones.

Los patrones GRASP definidos son:

Bajo acoplamiento: Posibilita que una clase no dependa mucho de otras clases, este patrón se empleará en clases como “*proyectos*”, “*peticiones*”, pues al crear la clase “*DB*” se disminuye la dependencia al no tener que conocer ni recurrir demasiado a la clase que angular propone. (Larman, 2014)

Alta Cohesión: Este patrón caracteriza a las clases que posean responsabilidades estrechamente relacionadas, es decir, que no realicen un trabajo enorme. Con el objetivo de que las clases “*proyectos*”, “*peticiones*”, no realizaran un trabajo enorme y poder reutilizar código se creará la clase “*DB*”, encargada de las funcionalidades de conexión a la base de datos (Larman, 2014).

Experto: Mediante su uso, se asignan responsabilidades a la clase que cuenta con la información necesaria. Se evidenciará en la clase “*DB*” pues esta poseerá un objeto de la clase “*SQLite*” para poder acceder a la información y funcionalidades de la base de datos.

Lo patrones GOF (Pandilla de cuatro) definidos son:

Singleton (Instancia única): El patrón Singleton se encarga de controlar que únicamente se crea una instancia de una clase en toda la aplicación mediante el uso de un único punto de acceso (Antonio, 2015). Este patrón se evidencia en la clase “*DB*”.

2.5 Tarjetas CRC

Aunque en general el diseño es realizado por los propios desarrolladores en ocasiones se reúnen aquellos con más experiencia o incluso se involucra al cliente para diseñar las partes más complejas. En estas reuniones se emplean un tipo de

tarjetas denominadas CRC (*Class, Responsibilities and Collaborator-Class*, Responsabilidad y Colaborador) cuyo objetivo es facilitar la comunicación y documentar los resultados. Para cada clase identificada se rellenará una tarjeta de este tipo, se especificará su finalidad y las clases con las que interactúa.

Tabla 3 Tarjeta CRC "Login"

Tarjeta CRC
Clase: Login
Responsabilidades: Es la clase controladora que se encarga de manipular la información obtenida del servicio para realizar la autenticación del usuario.
Colaboraciones: -DBProvider -AuthProyectosProvider

Tabla 4 Tarjeta CRC "Proyecto"

Tarjeta CRC
Clase: Proyecto
Responsabilidades: Es la clase controladora que guarda la información obtenida de los proyectos asignados a un usuario.
Colaboraciones: -DBProvider -AuthProyectosProvider -Login

Tabla 5 Tarjeta CRC "Peticiones"

Tarjeta CRC
Clase: Peticiones

<p>Responsabilidades: Es la clase controladora que permite gestionar la información consultada a través de un servicio referente a un proyecto asignado.</p>
<p>Colaboraciones:</p> <ul style="list-style-type: none"> -DBProvider -AuthProyectosProvider -Proyectos -Login

Tabla 6 Tarjeta CRC "Estado General "

<p>Tarjeta CRC</p>
<p>Clase: Estado General</p>
<p>Responsabilidades: Es la clase que permite manipular la información sobre el estado general de un proyecto asignado a un usuario.</p>
<p>Colaboraciones:</p> <ul style="list-style-type: none"> -DBProvider -AuthProyectosProvider -Proyectos

Tabla 7 Tarjeta CRC "Costo "

<p>Tarjeta CRC</p>
<p>Clase: Costo</p>
<p>Responsabilidades: Es la clase que permite obtener la información a partir de un servicio a consumir generando los datos sobre el indicador de costo de un proyecto seleccionado.</p>
<p>Colaboraciones:</p> <ul style="list-style-type: none"> -DBProvider -AuthProyectosProvider -Proyectos -Login

Tabla 8 Tarjeta CRC "Tiempo "

Tarjeta CRC
Clase: Tiempo
Responsabilidades: Es la clase que permite generar la información obtenida del servicio pertinente a partir de un proyecto asignado.
Colaboraciones: -DBProvider -AuthProyectosProvider -Proyectos -Login -Estado

Tabla 9 Tarjeta CRC "Recursos Humanos "

Tarjeta CRC
Clase: Recursos Humanos
Responsabilidades: Posibilita la obtención de los indicadores de recursos humanos a partir del servicio consumido a un proyecto asignado.
Colaboraciones: -DBProvider -AuthProyectosProvider -Proyectos -Login -Estado

Tabla 10 Tarjeta CRC "Riesgo "

Tarjeta CRC
Clase: Recursos Humanos
Responsabilidades: Clase encargada de la manipulación de los datos obtenidos una vez realizada la petición al servicio.
Colaboraciones: -DBProvider -AuthProyectosProvider

-Proyectos
-Login
-Estado

Tabla 11 Tarjeta CRC "AuthProyectos "

Tarjeta CRC
Clase: AuthProyectos
Responsabilidades: Es la clase donde se encuentran los servicios a consumir por el cliente una vez realizada las peticiones.

Tabla 12 Tarjeta CRC "DB "

Tarjeta CRC
Clase: DB
Responsabilidades: Es la clase donde se encuentran los métodos para manipular la información almacenada en la Base de datos

2.6 Limitaciones

La propuesta de solución no se puede verificar en su versión Android debido a que el centro donde ésta se implementa refuta las direcciones ip a las cuales se debe acceder para lograr el despliegue a su plataforma correspondiente.

Conclusiones Parciales

- Fueron definidos nueve requisitos funcionales de los cuales uno presenta prioridad alta, dos prioridades bajas, seis prioridades medias.
- Se utiliza como arquitectura de la solución Modelo, Vista, Vista Modelo (MVVM).
- Se identificó nueve tarjetas CRC en las cuales se describieron las clases que intervienen en el proceso y las colaboraciones entre ellas.
- Se utilizó como patrones GRASP: Bajo Acoplamiento, Alta Cohesión, Experto y como patrón GOF el Singleton.

Capítulo 3: Validación y Pruebas

En este capítulo se exponen las pruebas de validación realizadas mediante los diseños de caso de pruebas, las pruebas unitarias, así como los resultados obtenidos en la encuesta de satisfacción realizada por las distintas áreas.

3.1 Diagrama de componentes

Un componente es un módulo de software que puede ser código fuente, código binario, un ejecutable, o una biblioteca con una interfaz definida. Una interfaz establece las operaciones externas de un componente, las cuales determinan una parte del comportamiento del mismo. Además, se representan las dependencias entre componentes o entre un componente y la interfaz de otro, es decir uno de ellos usa los servicios o facilidades del otro.

Estos diagramas pueden incluir paquetes que permiten organizar la construcción del sistema de información en subsistemas y que recogen aspectos prácticos relacionados con la secuencia de compilación entre componentes, la agrupación de elementos en bibliotecas entre otro (James Rumbaugh, 2000).

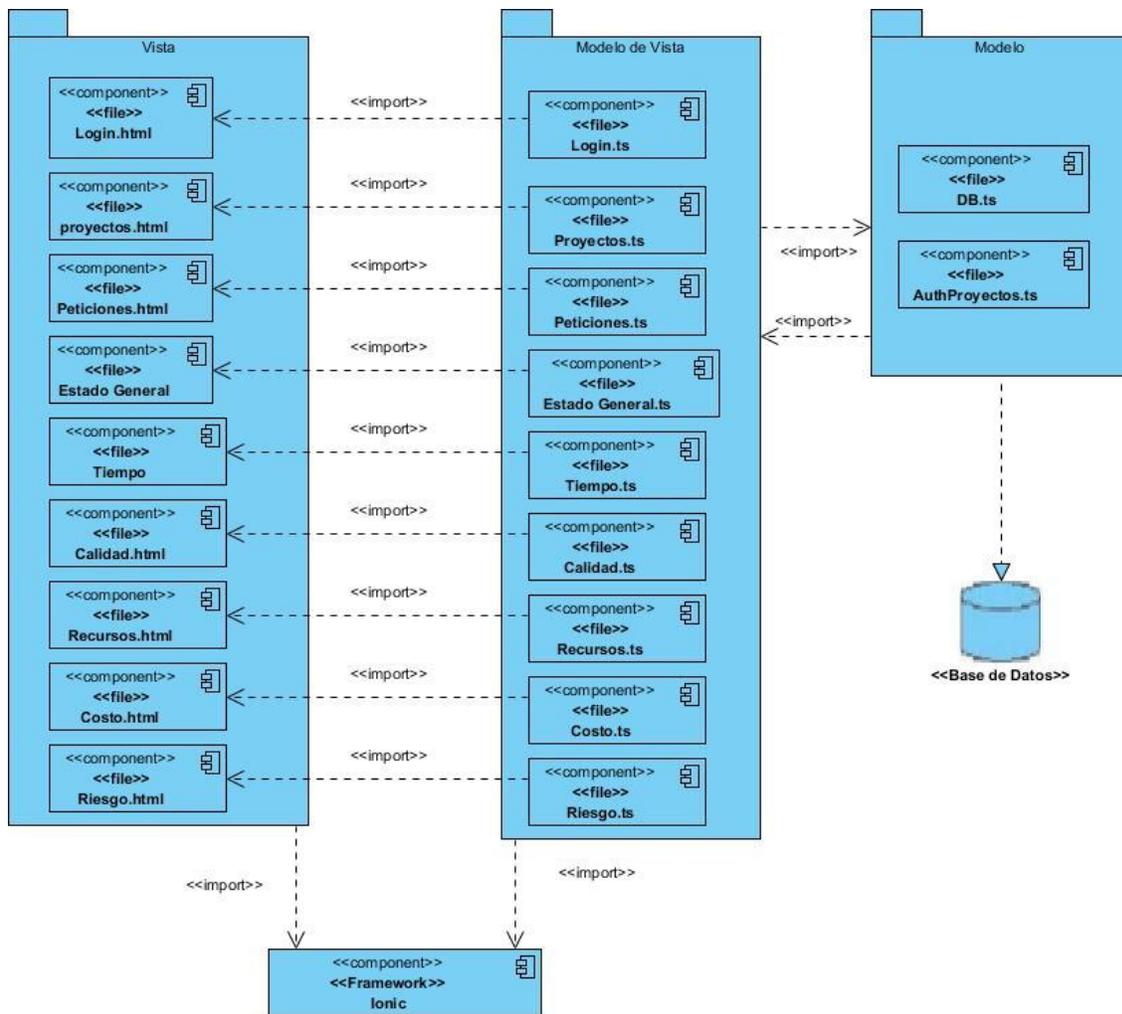


Figura 1 Diagrama de componentes

En la fig.4 se muestra el diagrama de componentes donde existe un paquete el cual conforma todas las vistas de la aplicación ,otro paquete donde se centra el modelo de vista el cual aborda toda la lógica del negocio y otro paquete que es el modelo donde se encuentran los servicios a consumir. Además existe el componente Ionic que es el *framework* por el cual se realiza la propuesta de solución así como un último componente que es database que es donde se guardara toda la información de la aplicación propuesta. Los paquetes de la vista y del modelo de vista usan de Ionic, el paquete de los modelos de vista es el que tiene acceso al modelo y el modelo es el que trabaja directamente con la base de datos SQLite; además cada componente del paquete de los modelos de vista que depende de cada componente de la vista.

En la tabla 19 se muestra la validación correspondiente a la funcionalidad de Autenticar Usuario.

Tabla 13 Caso de Prueba Autenticar Usuario

Id.del escenario	Escenario	Variable 1(Usuario)	Variable 2(Contraseña)	Respuesta del sistema
EC 1	Autenticar Usuario	V	I	Usuario o contraseña incorrectas
		I	V	Usuario o contraseña incorrectas
		I	I	Usuario o contraseña incorrectas
		V	V	Usuario o contraseña correctas

Al introducir los datos de usuario y contraseña estos pueden tomar valores válidos (V) o inválidos (I) y dependiendo de estos valores es la respuesta que tiene el sistema.

3.2 Pruebas de Validación

3.2.1 Caso de Prueba (Ver [Anexo 7](#))

Descripción General: La historia de usuario se inicia cuando el usuario introduce su usuario y contraseña y selecciona la opción de Acceder.

Condiciones de ejecución: El usuario debe llenar todos los campos del formulario.

Secciones a probar en la historia de usuario: Variaciones entre las distintas posibilidades que pueden tomar los valores de las variables entre verdaderos (V) e incorrectos (I) y dependiendo de ellos las distintas respuestas del sistema. Para la ejecución de los casos de prueba se puede hacer uso de un manual de usuario ([Anexo 5](#))

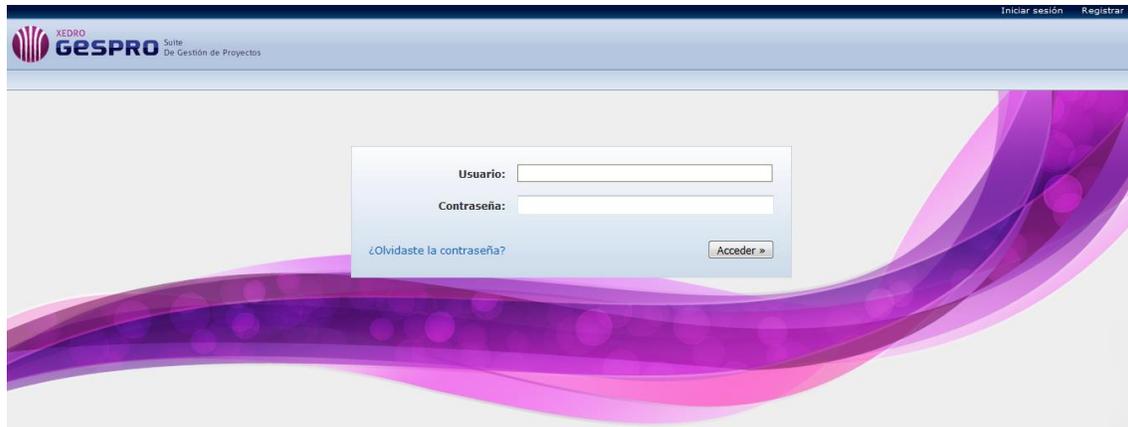


Figura 2 Autenticar Usuario Web

Xedro-Gespro

Usuario

Contraseña

URL

AUTENTICAR

Figura 3 Autenticar Usuario Móvil

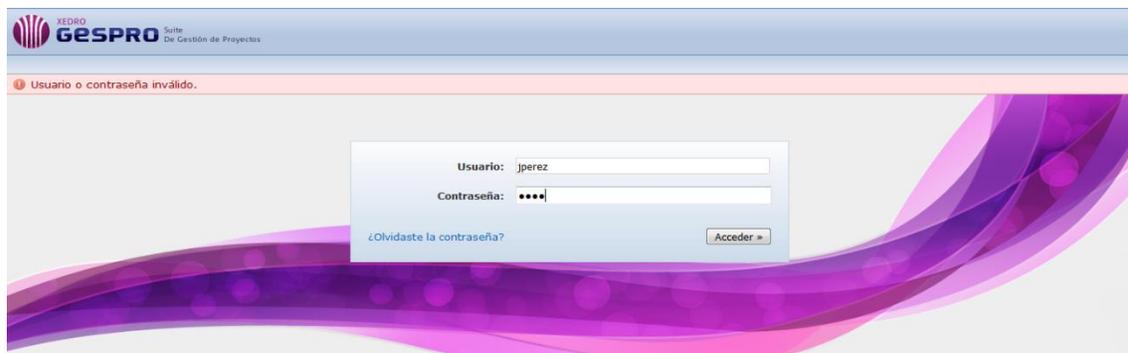


Figura 4 Autenticar Usuario Datos Incorrectos Web



Figura 5 Autenticar Usuario Datos Incorrectos Móvil

3.3 Comparación entre la aplicación vigente y la propuesta de solución sobre entorno móvil.

Para la comparación de las aplicaciones se tuvieron en cuenta los parámetros que para el equipo de desarrollo resultan relevantes y que son propuestos por el estándar internacional ISO 25010, eficiencia de desempeño o rendimiento y portabilidad; además se analiza la relación con las áreas de conocimiento propuestas por PMBOK en su versión 6 (ISO25010, 2018).

Áreas del conocimiento: Donde se determinan las áreas del conocimiento utilizadas durante el desarrollo de ambas propuestas de soluciones.

Adaptabilidad: Evidencia sobre el sistema operativo en cual se garantiza como mínimo el despliegue de la aplicación.

Rendimiento: Se evalúan los tiempos de respuesta que ofrece ambas propuestas de soluciones ejecutándose sobre un ambiente de pruebas para dispositivos móviles con versión Android 4.4.

Tabla 14: Comparación entre versiones para móviles de Xedro-Gespro

Parámetros a evaluar	Xedro-Gespro v1.0	Xedro-Gespro v2.0
Áreas del conocimiento	Incluye solamente el área del conocimiento de la administración del alcance.	Se adicionan las áreas del conocimiento de la administración de los costos, administración de recursos, administración de calidad, administración de los riesgos además de la inclusión de la administración del alcance.
Adaptabilidad	Soporta los sistemas operativos de android a partir de la versión 4.4.	Soporta los sistemas operativos de android a partir de la versión 4.0.
Rendimiento	El tiempo de ejecución cargando la plataforma Android es de 13.74 segundos para cuatro indicadores mostrados.	El tiempo de ejecución cargando la plataforma Android es de 16.10 segundos para nueve indicadores mostrados.

Evaluados los parámetros sobre las dos versiones de Xedro-Gespro para dispositivos móviles se define que la aplicación en su versión 2.0 es superior a la versión anterior en cuanto a las áreas del conocimiento que plantea PMBOK en su versión 6, además se logra un mejoramiento en portabilidad permitiendo a los usuarios realizar su despliegue en dispositivos móviles con sistema Android superior a 4.0 y garantiza mayor eficiencia de desempeño, para la evaluación del indicador de rendimiento se establece una proporcionalidad en cuanto a los tiempos de respuestas entre ambas versiones para determinar el valor óptimo máximo que puede alcanzar la versión 2.0, obteniendo como respuesta 30,91 segundos,

establecida una comparación con el valor alcanzado de 16.10 segundos se afirma que la propuesta de solución es óptima.

3.4 Pruebas unitarias

Fueron realizadas 3 iteraciones sobre el constructor de la clase MyApp al método Login () de la clase Login donde fueron detectados 3 errores, para el despliegue de las pruebas unitarias fueron utilizados los frameworks de Jasmine y Karma.

- El servicio dentro de la clase proveedor no se inicializo correctamente.
- La URL del servidor no tenía valor por defecto.
- La petición al servidor estaba mal formada por lo que la respuesta era fallida.

En las siguientes figuras se evidencian los errores detectados durante las 3 iteraciones realizadas:



Figura 6 Errores en la 1ra iteración



Figura 7. Errores en la 2ra iteración



Figura 8 Errores en la 3ra iteración

3.5 Encuesta a especialistas

Una vez realizada la encuesta de satisfacción (Ver [Anexo 1](#)) a una muestra de 5 especialistas, sobre las aplicaciones de Xedro-Gespro en su versión móvil y teniendo en cuenta los parámetros de área del conocimiento, rendimiento y adaptabilidad, arrojó resultados superiores entre las dos versiones, como se muestra en la figura 9.

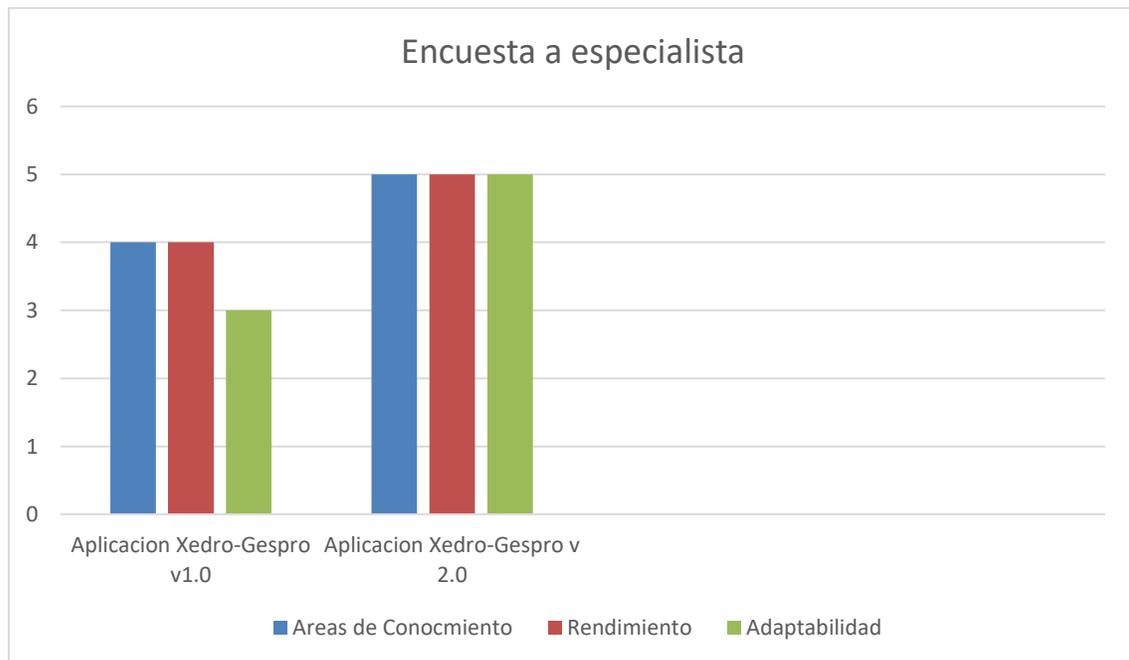


Figura 9 Nivel de aceptación de la aplicación

El resultado de la encuesta también arrojó sugerencias, las cuales son:

-Hacer una sincronización bidireccional entre servidor web y aplicación móvil.

-Lograr sincronizar la Base de datos Sqlite con la información guardada en la Base de datos del servidor Web.

-Que se integre con otras aplicaciones que podrían brindarse a los usuarios/clientes de Gespro.

-Hacer un mejoramiento de algunas particularidades en las interfaces de usuario.

-Reflejar el tiempo desde la última conexión activa cuando se está desconectado con el fin de que los usuarios puedan tener evidencia lo desactualizado que puedan estar los datos en pantalla.

Conclusiones Parciales

Durante el desarrollo de las validaciones y pruebas correspondientes a la propuesta de solución se demostró:

- Las interfaces diseñadas cumplen con los estándares de Gespro en su versión web.
- Fueron solucionadas todas las no conformidades encontradas durante las 3 iteraciones pertinentes a las pruebas unitarias.
- Quedo mejor detallada la propuesta de solución utilizando diagrama de componente porque permitió organizar la construcción del sistema de información en subsistemas.
- Se evidenciaron los mejoramientos entre los parámetros establecidos en las pruebas de aceptación.

Conclusiones Generales

Una vez concluido el trabajo se puede arribar a las siguientes conclusiones:

- El desarrollo de la versión 2.0 de Xedro-Gespro móvil permite a los usuarios acceder a los indicadores asignados a sus proyectos a través de sus dispositivos móviles.
- Las pruebas de validación realizadas permiten llevar al usuario a una confidencialidad y seguridad total, puesto que trabaja sobre los mismos datos que se utilizan para Xedro-Gespro en su versión web.
- Las encuestas de aceptación realizada a los especialistas demostraron superioridad de la V2.0 con respecto a la V1.0 teniendo en cuenta indicadores como rendimiento, adaptabilidad y trabajo con las áreas de conocimiento propuestas por PMBOK.

Recomendaciones

A partir de los resultados del trabajo de diploma realizado se proponen las siguientes recomendaciones:

- Realizar una sincronización de la información no solamente entre aplicación web y móvil sino que permita al usuario modificar sus tareas desde su dispositivo móvil y estas automáticamente se actualicen en la web.
- Evaluar, en la versión 2.0 de Xedro-Gespro para móviles, atributos de calidad como la usabilidad, seguridad y fiabilidad, según establece la norma ISO 25010.

Referencias

Angel, . 2016. desarrolloweb. *desarrolloweb*. [Online] 2016. [Cited: 13 febrero 2018.] <https://desarrolloweb.com/copyright/>.

Antonio. 2015. codecriticon. *codecriticon*. [Online] 8 septiembre 2015. [Cited: 15 febrero 2018.] <http://codecriticon.com/patron-singleton/>.

API_Market. 2016. bbvaopen4u. *bbvaopen4u*. [Online] BBVA, 22 enero 2016. [Cited: 24 enero 2018.] <https://bbvaopen4u.com/es/actualidad/javascript-rey-del-desarrollo-moderno-en-front-end-y-back-end>.

CCM. 2015. CCM. *CCM*. [Online] 2015. [Cited: 30 noviembre 2017.] <http://es.ccm.net/>.

Celular, historia y creacion. 2014. historia y creacion celular. *historia y creacion celular*. [Online] ATOM, 2014. [Cited: 24 enero 2018.] <http://histoeiaycreacioncelular.blogspot.com/p/creador.html>.

Cordova. 2015. Cordova. *Cordova*. [Online] 2015. [Cited: 30 noviembre 2017.] <https://cordova.apache.org/docs/es/latest/guide/overview/>.

FMU(Formulasproyectosurbanos). 2012. formulasproyectosurbanos. *formulasproyectosurbanos*. [Online] 8 mayo 2012. <https://formulaproyectosurbanospmipe.wordpress.com/2012/05/08/tema-n-5-gestion-de-alcance-del-proyecto-segun-la-guia-del-pmbok-30-04-2012-sesion-10/>.

Github. 2018. github. *github*. [Online] github, 2018. [Cited: 7 febrero 2018.] <https://github.com/Microsoft/vscode>.

Guerra, Enrique Fernadez. 2016. desarrollo web. *desarrollo web*. [Online] 2 junio 2016. [Cited: 30 noviembre 2017.] <https://desarrolloweb.com/articulos/introduccion-a-typescript.html>.

Hipertextual. 2013. hipertextual. *hipertextual*. [Online] 2013. [Cited: 30 noviembre 2017.] <https://hipertextual.com/archivo/2013/05/entendiendo-html5-guia-para-principiantes/>.

Informatica-hoy. 2016. informatica-hoy. *informatica-hoy*. [Online] 2016. [Cited: 24 enero 2018.] <https://www.informatica-hoy.com.ar/telefonos-celulares/La-historia-del-Telefono-Celular.php>.

Ing.Marco. 2014. superinformacionweb. *superinformacionweb*. [Online] 2014. [Cited: 13 febrero 2018.] <http://superinformacionweb.blogspot.com/2014/11/conceptos-de-diseno.html>.

LanceTalent. 2015. LanceTalent. *LanceTalent*. [Online] 2015. <https://www.lancetalent.com/blog/tipos-de-aplicaciones-moviles-ventajas-inconvenientes/>.

Larman. 2014. *UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado. 3da Edición*. Madrid : s.n., 2014.

LLopez, . 2016. Ingeniería Agil. *Ingeniería Agil*. [Online] 6 octubre 2016. [Cited: 12 febrero 2018.] <http://fernandollopis.dlsi.ua.es/?p=39>.

Lucidchart. 2017. lucidchart. *lucidchart*. [Online] 2017. [Cited: 30 noviembre 2017.] <https://www.lucidchart.com/pages/es/qu%C3%A9-es-el-lenguaje-unificado-de-modelado-uml>.

Matias, . 2017. xamarinlatino. *xamarinlatino*. [Online] 20 junio 2017. [Cited: 15 febrero 2018.] <https://xamarinlatino.com>.

Navarra. 2015. *Uso de dispositivos móviles*. España : s.n., 2015.

Nieves, . 2014. *Metodologías Ágiles. Proceso Unificado Ágil (AUP). Ingeniería del Software II – Análisis de Sistemas*. 2014. p.9.

Portal Educativo de las Américas. 2013. *Educoas*. s.l. : OEA, 2013. 147.

PhoneGapSpain. 2016. PhoneGapSpain. *PhoneGapSpain*. [Online] 2016. [Cited: 30 noviembre 2017.] <http://www.phonegapSpain.com/que-es-y-como-empezar-con-ionic-framework/>.

PMBOOK. 2017. *PMBOOK*. 2017. 6 edition.

recursosenprojectmanagement.2018.recursosenprojectmanagement.
recursosenprojectmanagement. [Online] 2018. [Cited: 16 enero 2018.]
www.recursosenprojectmanagement.com/metodologia-agil/.

Sánchez, . 2014. *PROGRAMA DE MEJORA Metodología de desarrollo para la Actividad productiva de la UCI.* 2014.

Silva, .2015. [Online] 2015. [Cited: 30 noviembre 2017.]
http://biblioteca.uniss.edu.cu/sites/default/files/CD/Yayabociencia%202015/documentos/11-Gest_Empresarial/1Carlos%20Silva%20Alvarez.pdf.

Sinapps. 2016. Todo sobre la Gestión de Proyectos. *Todo sobre la Gestión de Proyectos.* [Online] 2016. <https://www.sinnaps.com/blog-gestion-proyectos/gestion-de-proyectos>.

Software, Metodologia de desarrollo de. 2015. Metodologia de desarrollo de software. *Metodologia de desarrollo de software.* [Online] 2015. [Cited: 23 diciembre 2017.] <https://okhosting.com/blog/metodologias-del-desarrollo-de-software/>.

Synergix. 2014. synergix. *synergix.* [Online] 10 julio 2014. [Cited: 7 febrero 2018.] <https://synergix.wordpress.com/2008/07/10/modelo-de-dominio/>.

Turriatte, Wilson Flores. 2016. frontendlabs. *frontendlabs.* [Online] 2 febrero 2016. [Cited: 23 enero 2018.] <https://frontendlabs.io/3158--react-js-espanol-tutorial-basico-primeros-pasos-ejemplos>.

Padron, Lainet. 2017. *Aplicacion Xedro-Gespro para dispositivos moviles v1.0.* 2017.

James Rumbaugh, I. J. (2000). El Lenguaje Unificado de Modelado. Manual de referencia.Madrid

Guerra, Cesar Arturo. 2016. Software Guru. *Software Guru.* [En línea] SG, 2016. [Citado el: 12 de junio de 2018.] <https://sg.com.mx/revista/17/obtencion-requerimientos-tecnicas-y-estrategia>. 17.

25010, ISO. 2018. ISO 25000. *ISO 25000.* [En línea] 2018. [Citado el: 20 de Junio de 2018.] <http://iso25000.com/index.php/normas-iso-25000/iso-25010>.

Anexos

Anexo 1. Encuesta de satisfacción

1. ¿Cree usted que la aplicación resuelve el problema de adaptabilidad y rendimiento? Marque con una X según convenga:

Adaptabilidad Sí___ No___

Rendimiento Sí___ No___

2. ¿Existe variedad en las áreas de conocimiento que propone PMBOK y que se trabajan en la aplicación?

Sí___ No___

3. ¿Considera adecuado el diseño que posee la aplicación?

Sí___ No___

4. ¿Considera adecuados los indicadores que se muestran en la aplicación?

Sí___ No___

5. ¿Considera adecuados los tiempos de respuesta de la aplicación?

Sí___ No___

6. Sugerencias:
-

Anexo 2. Manual de instalación y configuración de las herramientas utilizadas.

Node JS

Node.js es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor.

- *sudo apt install nodejs*

NPM

Del término inglés: Node Package Manager, es un gestor de paquetes para el lenguaje de programación JavaScript. Es el gestor de paquetes predeterminado para el entorno de ejecución de JavaScript Node.js. Los pasos para su instalación son:

- *sudo apt install npm*

Para configurar NPM para el servidor de dependencia de la UCI, se debe sobre escribir en el fichero 'oculto' ~/.npmrc que se encuentra en el home del sistema, con la siguiente información:

- *registry = http://nexus.prod.uci.cu/repository/npm-all*
- *strict-ssl = false*

Ionic y Cordova

Ionic permite a los desarrolladores crear aplicaciones móviles con un diseño más semejante al de una aplicación nativa usando HTML5 y AngularJS y Cordova es un marco de desarrollo móvil de código abierto que permite utilizar las tecnologías estándar web como HTML5, CSS3 y JavaScript para desarrollo multiplataforma, evitando el lenguaje de desarrollo nativo de cada plataforma móvil. Los pasos para su instalación son:

- *\$ npm install -g ionic cordova*

-Crear nuevo proyecto

- *ionic start myApp blank -v3*

-Añadir plataforma

- *ionic platform add Android*

SDK Android

Del término inglés: Software Development Kit, es un kit de desarrollo de software el cual es el lenguaje de programación de Android.

- Descomprimir el SDK y abrir el archivo `~/.bashrc`
- `export ANDROID_HOME=<dirección donde se encuentra>`
- `PATH=${PATH}:${ANDROID_HOME}/tools:${ANDROID_HOME}/platform-tools`

-Para correr el proyecto en un móvil.

- `ionic cordova run android --device`

Jasmine y Karma para la realización de las pruebas unitarias

Instalar el paquete:

- `npm install -g karma-cli`

Instalar dependencias

- `npm install --save-dev @types/jasmine @types/node html-loader jasmine karma`

`karma-webpack ts-loader karma-sourcemap-loader karma-jasmine karma-jasmine-`

`html-reporter angular2-template-loader karma-chrome-launcher null-loader`

`typescript@latest`

57

Configurando el entorno de trabajo

- <https://nexus.prod.uci.cu/repository/github-proxy/roblouie/ionic-unit-test-config/archive/master.zip>

Extraer en la carpeta test-config

Añadir en la etiqueta "scripts" del archivo package.json lo siguiente:

- "test": "karma start ./test-config/karma.conf.js"

Añadir en el arreglo "exclude" del archivo tsconfig.json lo siguiente:

- "src/**/*spec.ts"
- Escribir las pruebas en archivos con extensión .spec.ts

Para correr las pruebas

- npm test

Anexo 3. Manual de instalación y configuración de SQLite.

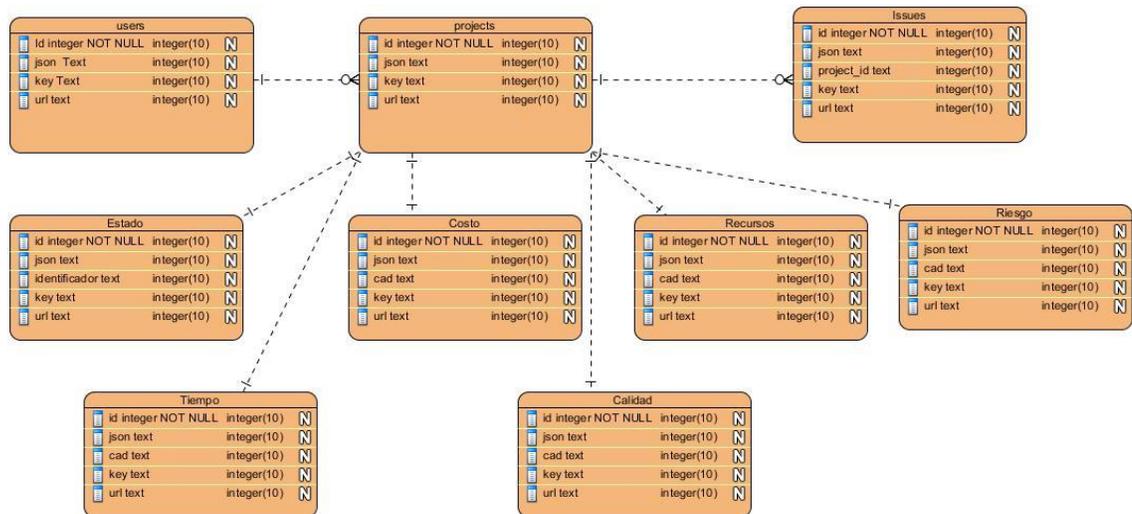


Figura 10 Modelo de Datos

A continuación, se muestra la descripción de las tablas del modelo de datos representado:

Nombre de la Tabla	Descripción
users	Es la tabla responsable de guardar la llave del usuario, la url de conexión así y también la notación de los objetos de JavaScript (JSON) que guarda toda la información que tiene el usuario en la aplicación.
projects	Es la encargada de guardar toda la información de los proyectos así como almacenar la llave del usuario y la url del servidor.
issues	Se encarga de almacenar los datos de las peticiones así como el identificador del proyecto, la llave del usuario y la url del servidor.
Estado	Almacena los datos del indicador de estado así como el identificador del proyecto, la llave del usuario y la url del servidor.
Costo	Encargada de almacenar los datos del indicador de costo así como el identificador del proyecto, la llave del usuario y la url del servidor.
Riesgo	Almacena los datos del indicador de riesgo así como el identificador del proyecto, la llave del usuario y la url del servidor.

Recursos	Guarda los datos del indicador de recursos humanos así como el identificador del proyecto, la llave del usuario y la url del servidor.
Tiempo	Permite almacenar los datos del indicador de tiempo así como el identificador del proyecto, la llave del usuario y la url del servidor.
Calidad	Almacena los datos del indicador de calidad así como el identificador del proyecto, la llave del usuario y la url del servidor.

Tabla 15 Descripción de las tablas

Anexo 4. Pasos para la sincronización de la base de datos SQLite con la base de datos de GESPRO versión web.

Para la explicación de este paso se tomó como ejemplo el método de autenticar usuario por ser el requisito funcional con prioridad alta.

Adquirir Información (Usuario) de Gespro en su versión web a través de la URL y guardas los datos del usuario.

- AuthUsuario: Instancia donde se almacena las direcciones URL donde están desplegados los servicios.
- Data: Objeto donde se guarda la información del usuario.
- Llave: Objeto donde se guarda la la llave del usuario.
- AddUsuario: Guarda los datos de Usuario en la base de datos.

```
login(usuario){
this.authusuario.getUsuario(this.usuario.user,this.usuario.password,this.usuario.url).subscribe(
  (datas: any) => {
    this.data=datas.user
```

```

        this.LLave=datas.user.api_key
        this.navCtrl.push(ProyectosPage,
        {LLave:this.LLave,url:this.usuario.url});
        //Declaracion de errores
        err => {
        if (err.status == 401) {
            console.log("asdasd")
            alert("Usuario o contraseña incorrecta");
        }
        else {
            console.log("asdasd")
            alert("No se pudo conectar con el servidor");
        }
        }
        //Añadiendo a la base de datos
        this.db.addUsuario(this.data,this.LLave,this.usuario.url);
    });
}

```

Anexo 5. Manual de usuario.

Requisitos:

- El usuario debe tener un dispositivo móvil con la versión de Android 4.0 o superior.
- El usuario debe tener copiado en su dispositivo la aplicación XedroGESPRO.apk.

Luego de cumplirse todos los requisitos el usuario debe de realizar los siguientes pasos:

1. Instalar la aplicación presionando en el botón de instalar.
2. Buscar en el menú la aplicación y seleccionarla para que se abra.

3. Luego de abierta la APK por primera vez saldrá la página de autenticación donde se muestran los campos a llenar para loguearse en la aplicación.



The screenshot shows a login form with a header 'Xedro-Gespro'. Below the header are three input fields labeled 'Usuario', 'Contraseña', and 'URL'. At the bottom of the form is a blue button labeled 'AUTENTICAR'.

Figura 11 Página de Autenticación

4. Luego el usuario introduce su usuario, contraseña y la dirección a donde quiere acceder y presiona el botón autenticar.



This screenshot is identical to the previous one, but the blue 'AUTENTICAR' button is highlighted with a red rectangular border, indicating the user's action.

Figura 12 Acción de Autenticar

5. Una vez autenticado en el sistema aparecerán los proyectos a los cuales está asignado dependiendo de su rol y dos botones uno para ver las Peticiones y otro para consultar el Estado General del proyecto.



Figura 13 Proyectos Asociados

- Si el usuario presiona el botón Peticiones se despliegan todas las peticiones del proyecto asignado.

#	Estados	Asunto	Asignado A	Fecha de Inicio	Fecha Fin
2069 4	Asignada	trabajo con robot de audio a seguir durante la creacion del video	Josue Perez	2016-06-06	2016-06-17
2069 3	Asignada	Redaccion de guión	Josue Perez	2016-06-13	2016-06-17
2069 2	Asignada	intalacion de herraminetas para los videos	Josue Perez	2016-06-06	2016-06-10
2069 1	Asignada	Creación de videos de auto ayuda en la herramineta Xedro Gespro	Josue Perez	2016-06-06	2016-06-17
2056 4	Nueva	T Descrpción la evaluación de competencias laborales en el XEDRO-GESPRO	Liosvel Medina	2016-06-13	2016-06-22
2056 3	Asignada	T Describir la relación usuario-rol-competencia-dimensiones	Liosvel Medina	2016-05-30	2016-06-14

Figura 14 Peticiones Asignadas

- Si el usuario presiona el botón Estado General se despliega una vista con el indicador de estado según el proyecto asignado.

← Estado

☰

indicadores	2018-02-10		2018-02-17	
Plan	90		78	%
RealIV	65		66	%
RealINV(Ejecucion Real)	93		82	%
IRE	1.05	B	1.03	B
IRP	1.02	B	1.03	B
IRC	0	M	0	M
IREF	0	M	0	M
IRHH	69	R	70.25	R
IRL	0	M	0	M
ICD	0.99	B	1	B

EVALUACION GENERAL INDICADORES DE RENDIMIENTO

Figura 15 Estado del Proyecto

8. Al presionar el botón en la esquina superior izquierda de la pantalla se despliega un menú deslizante en el cual se evidencia los demás indicadores.

← Estado

☰

indicadores	2018-02-10		2018-02-17	
Plan	90		78	%
RealIV	65		66	%
RealINV(Ejecucion Real)	93		82	%
IRE	1.05	B	1.03	B
IRP	1.02	B	1.03	B
IRC	0	M	0	M
IREF	0	M	0	M
IRHH	69	R	70.25	R
IRL	0	M	0	M
ICD	0.99	B	1	B

EVALUACION GENERAL INDICADORES DE RENDIMIENTO

Figura 16 Menú desplegable

9. Si el usuario presiona el botón del indicador de Costo se muestra la vista asociada al indicador pertinente según el proyecto asociado.

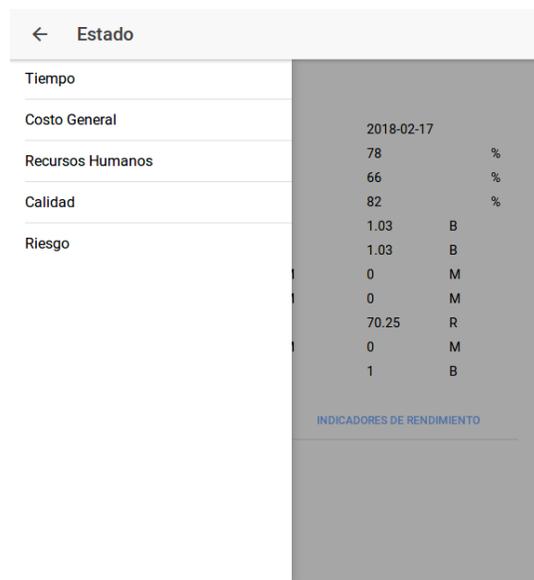


Figura 17 Indicadores en el menú

10. Si el usuario presiona el botón del indicador de Tiempo se muestra la vista asociada al indicador pertinente según el proyecto asociado.



Figura 18 Indicador de Tiempo

11. Si el usuario presiona el botón del indicador de Costo se muestra la vista asociada al indicador pertinente según el proyecto asociado.

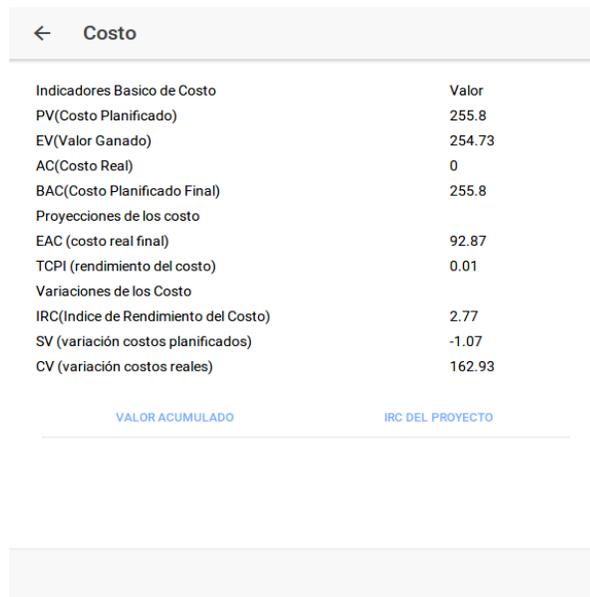


Figura 19 Indicador de Costo

12. Si el usuario presiona el botón del indicador de Recursos Humanos se muestra la vista asociada al indicador pertinente según el proyecto asociado.



Figura 20 Indicador de Recursos Humanos

13. Si el usuario presiona el botón del indicador de Calidad se muestra la vista asociada al indicador pertinente según el proyecto asociado.



Figura 21 Indicador de Calidad

14. Si el usuario presiona el botón del indicador de Riesgo se muestra la vista asociada al indicador pertinente según el proyecto asociado.

La tabla muestra el listado de riesgos de alta prioridad por su exposición. El encabezado es 'riesgo' y el subtítulo es 'Listado de riesgos de alta prioridad por su exposición.'.

	Total
Problemas de fallos eléctricos	Alto:0.33
Fallo en el agua	Alto:0.0
Corte Circuito	Muy alto:0.33

Figura 22 Indicador de Riesgo

Anexo 6. Historias de Usuarios

Tabla 16 HU_VisualizarProyectos

Número: 2		Nombre del requisito: VisualizarProyectos	
Programador: Josué Pérez Núñez		Iteración Asignada: 2da	
Prioridad: Baja		Tiempo Estimado: 12 días	
Riesgo en Desarrollo: N/A		Tiempo Real: 10 días	
Descripción: 1- Objetivo: <i>Permitir visualizar los proyectos a los cuales esté vinculado el usuario autenticado.</i> 2- Acciones para lograr el objetivo (precondiciones y datos): Para visualizar proyectos hay que: - <i>Ser miembro de algún proyecto.</i> - <i>Debe existir en el sistema al menos un proyecto.</i> 3- Flujo de la acción a realizar: <i>Cuando el usuario se autentica aparecen todos los proyectos a los cuales está vinculado.</i>			
Observaciones: Los proyectos contienen todas las peticiones asignadas a cada uno de sus miembros.			
Prototipo de interfaz:			

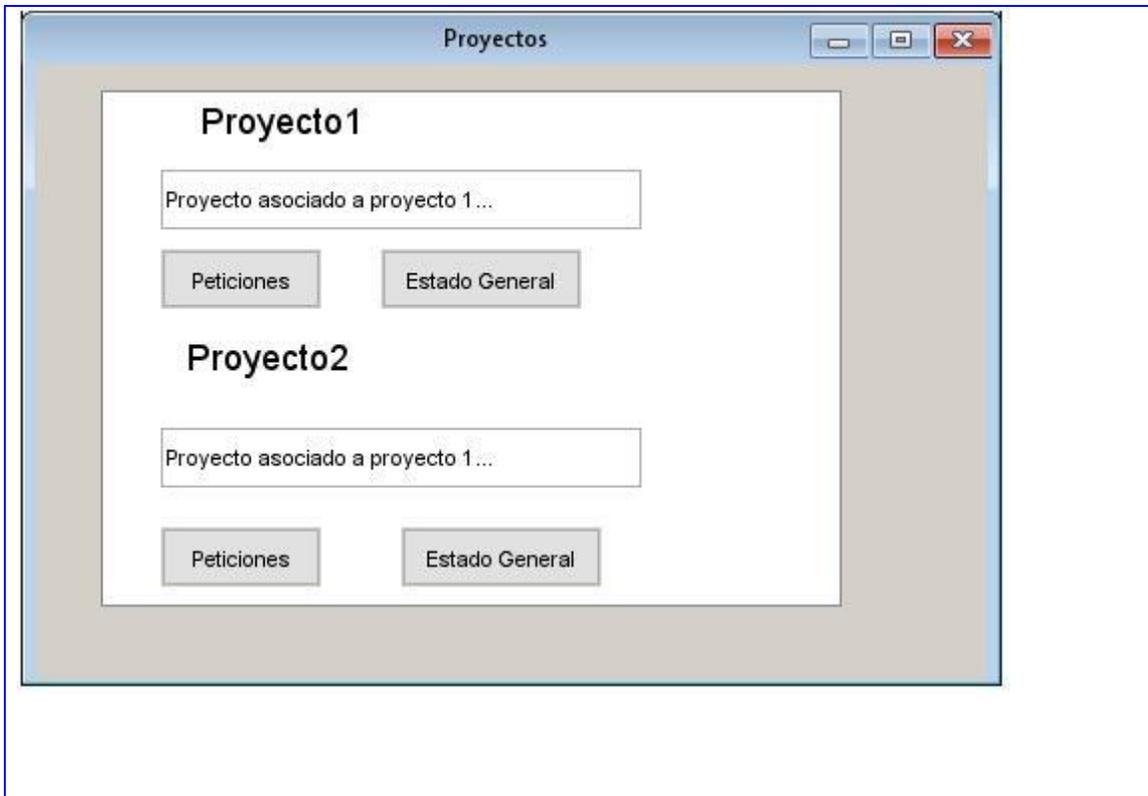


Tabla 17 HU_Visualizar_Peticiones

Número: 3	Nombre del requisito: Visualizar Peticiones
Programador: Josué Pérez Núñez	Iteración Asignada: 3era
Prioridad: Baja	Tiempo Estimado: 15 días
Riesgo en Desarrollo: N/A	Tiempo Real: 10 días
<p>Descripción:</p> <p>1- Objetivo:</p> <p><i>Permitir listar las peticiones de cada proyecto en la aplicación.</i></p> <p>2- Acciones para lograr el objetivo (precondiciones y datos):</p> <p>Para visualizar las peticiones hay que:</p> <ul style="list-style-type: none"> - <i>Estar autenticado en el sistema y ser miembro de al menos un proyecto.</i> 	

- Debe existir en el sistema al menos un proyecto.

- El proyecto debe contener al menos una petición.

3- Flujo de la acción a realizar:

Partiendo de la historia de usuario Visualizar Proyecto cuando el usuario selecciona la opción Peticiones, se mostrarán las peticiones que tenga el proyecto al cual está vinculado. Además, el usuario tiene la posibilidad de ver detalles de cada uno de ellas.

Observaciones: Las peticiones describen y controlan la ejecución de las actividades del proyecto.

Prototipo de interfaz:

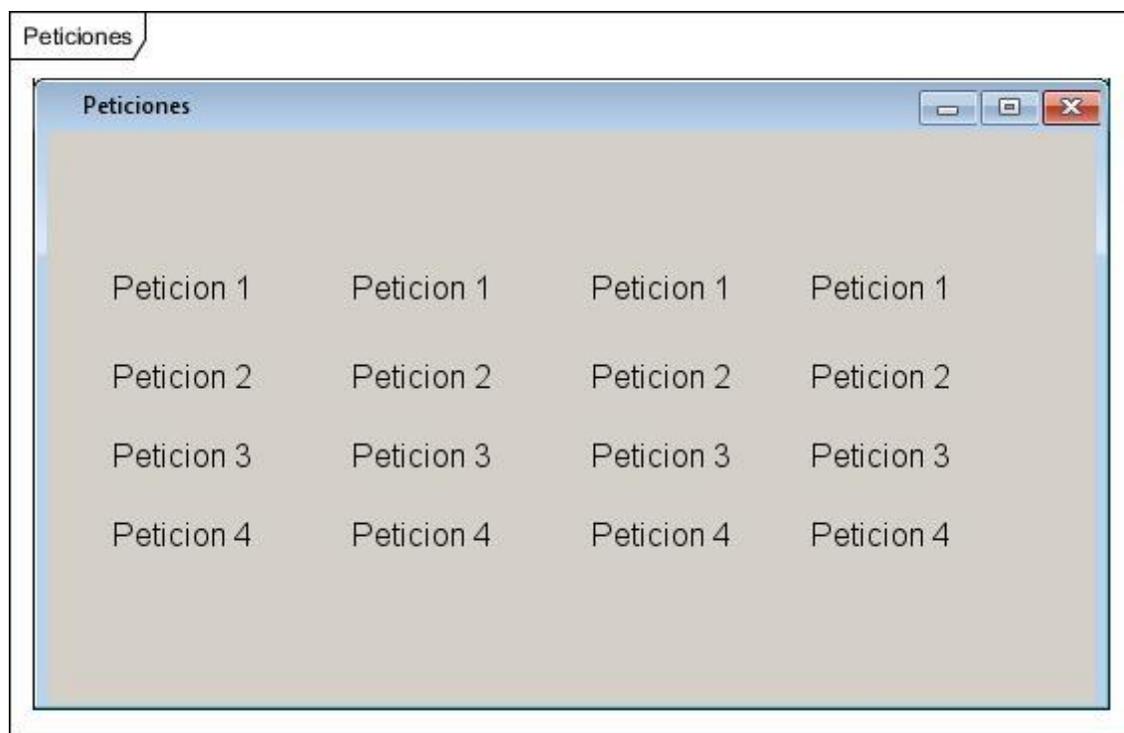


Tabla 18 HU_Estado General

Número: 5		Nombre del requisito: Visualizar Estado General	
Programador: Josué Pérez Núñez		Iteración Asignada: 5ta	
Prioridad: Media		Tiempo Estimado: 20 días	
Riesgo en Desarrollo: N/A		Tiempo Real: 12 días	

Descripción:

1- Objetivo:

Permitir visualizar el estado general en que se encuentra el proyecto

2- Acciones para lograr el objetivo (precondiciones y datos):

Para visualizar el estado general del proyecto hay que:

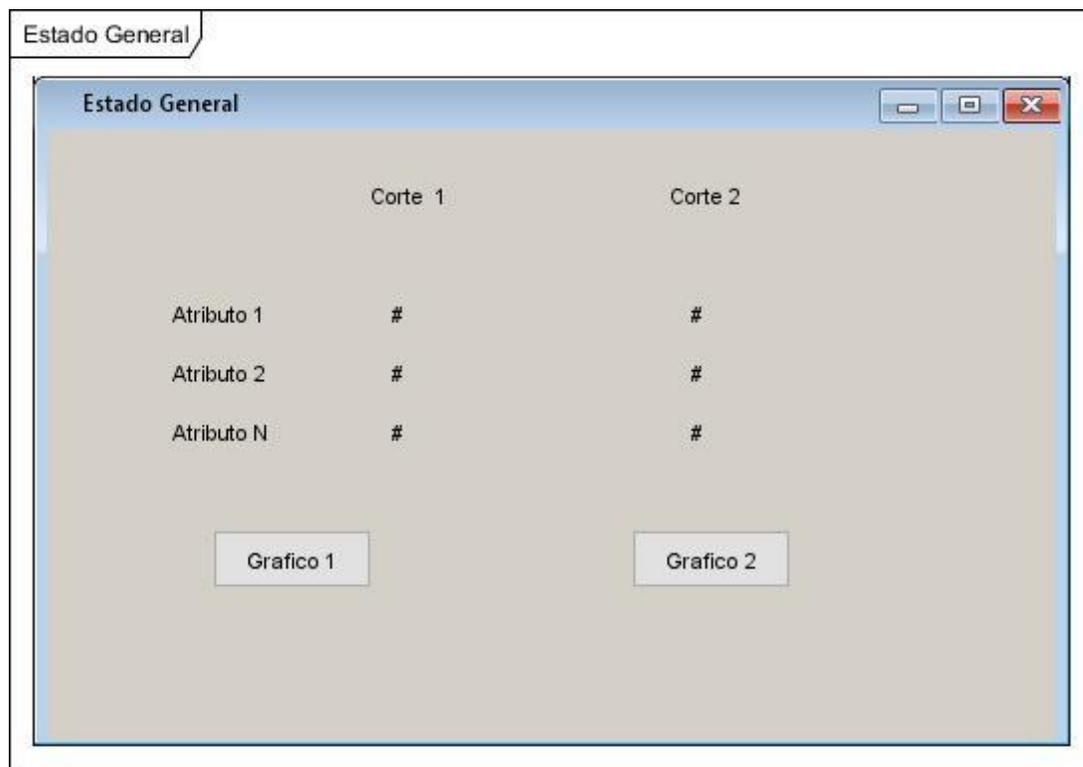
- *Estar autenticado en el sistema y ser miembro de al menos un proyecto.*
- *Debe existir en el sistema al menos un proyecto.*

3- Flujo de la acción a realizar:

Partiendo de Historia de Usuario Visualizar Proyecto el usuario selecciona la opción estado general y se visualiza el estado general en que se encuentra su proyecto.

Observaciones: El estado general describe como se encuentra el proyecto asignado.

Prototipo de interfaz:



--

Tabla 19 HU_Costo

Número: 6	Nombre del requisito: Visualizar Costo del Proyecto
Programador: Josué Pérez Núñez	Iteración Asignada: 6ta
Prioridad: Media	Tiempo Estimado: 20 días
Riesgo en Desarrollo: N/A	Tiempo Real: 12 días
Descripción:	
1- Objetivo:	
<i>Permitir visualizar el costo asignado al proyecto</i>	
2- Acciones para lograr el objetivo (precondiciones y datos):	
Para visualizar el costo del proyecto hay que:	
<ul style="list-style-type: none">- <i>Estar autenticado en el sistema y ser miembro de al menos un proyecto.</i>- <i>Debe existir en el sistema al menos un proyecto.</i>	
3- Flujo de la acción a realizar:	
<i>A partir de la historia de usuario Estado General el usuario despliega el menú y selecciona la opción costo se muestra el costo que se le fue asignado al proyecto.</i>	
Observaciones: El costo describe de forma general sus indicadores y un breve comportamiento de su flujo.	
Prototipo de interfaz:	

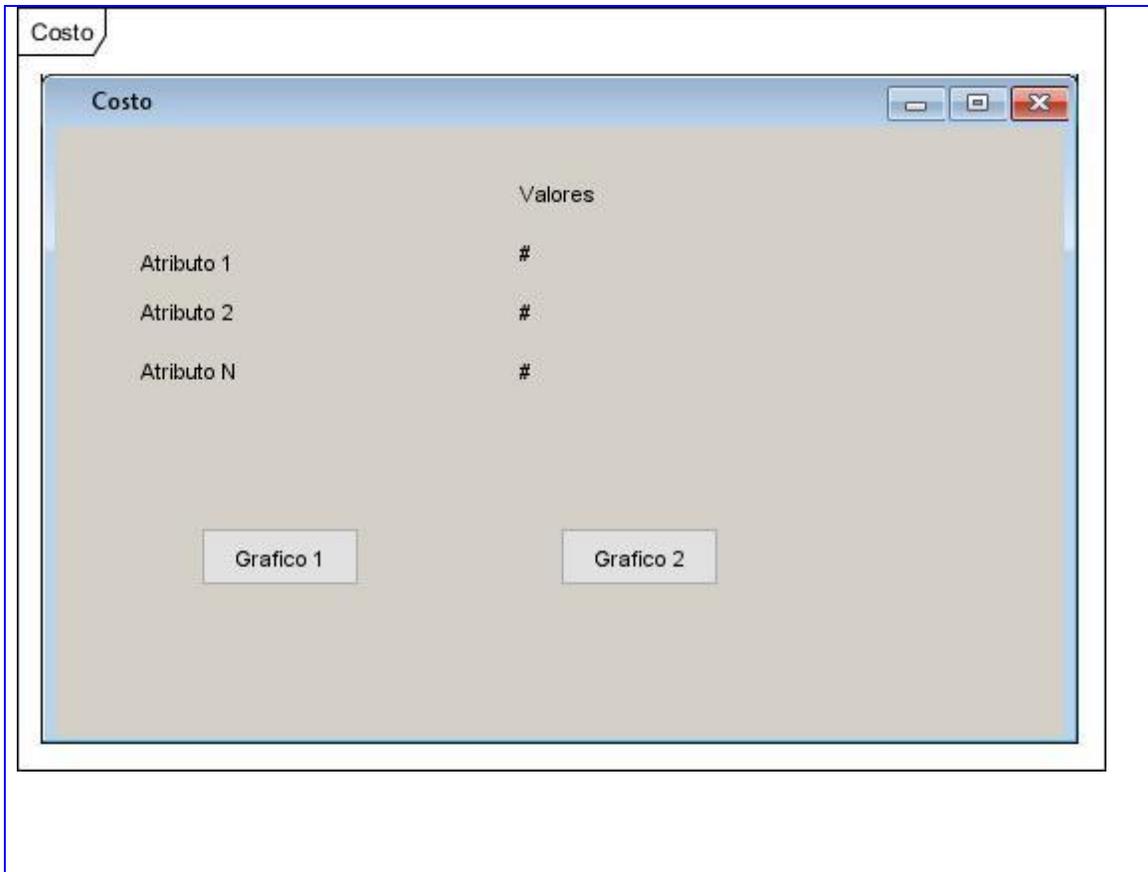


Tabla 20 HU_Recursos Humanos

Número: 7	Nombre del requisito: Visualizar Indicador Recursos Humanos
Programador: Josué Pérez Núñez	Iteración Asignada: 7ta
Prioridad: Baja	Tiempo Estimado: 10 días
Riesgo en Desarrollo: N/A	Tiempo Real: 8 días
Descripción:	
1- Objetivo:	
<i>Permitir visualizar los indicadores de recursos humanos asignados al proyecto.</i>	
2- Acciones para lograr el objetivo (precondiciones y datos):	
Para visualizar los indicadores de los recursos humanos hay que:	
- <i>Estar autenticado en el sistema y ser miembro de al menos un proyecto.</i>	

- Debe existir en el sistema al menos un proyecto.

3- Flujo de la acción a realizar:

A partir de la historia de usuario Estado General el usuario despliega el menú y selecciona la opción recursos humanos se visualiza los indicadores pertinentes al proyecto.

Observaciones: Los recursos humanos describen el trabajo que aporta el conjunto de los empleados o colaboradores asignados al proyecto.

Prototipo de interfaz:

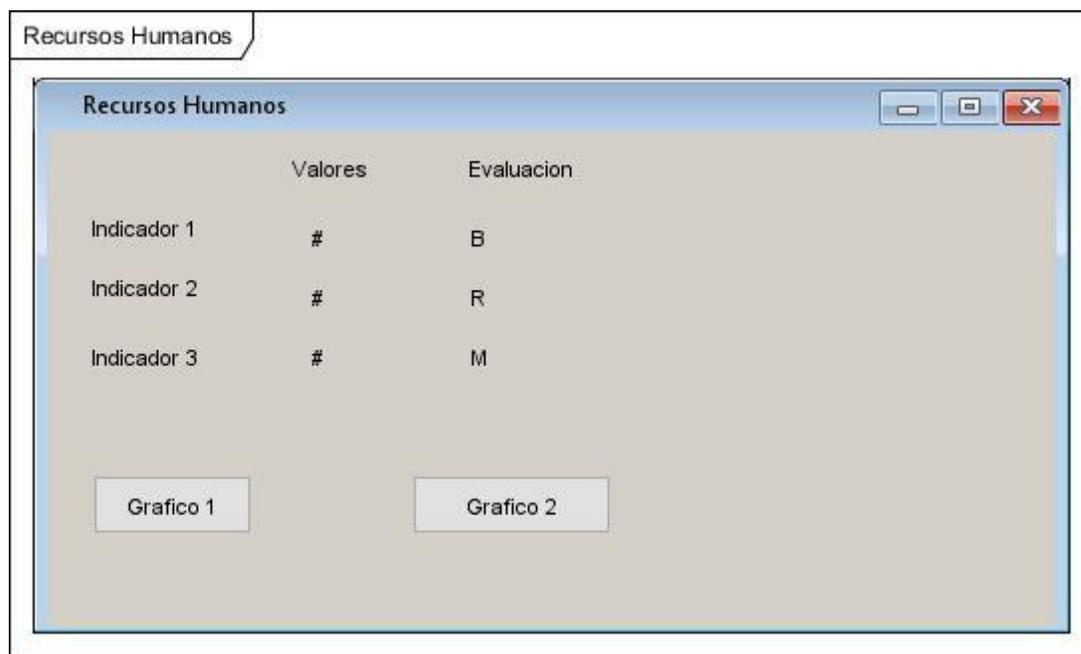


Tabla 21 HU_Riesgo

Número: 8	Nombre del requisito: Visualizar Indicador Riesgo
Programador: Josué Pérez Núñez	Iteración Asignada: 8ta
Prioridad: Baja	Tiempo Estimado: 15 días
Riesgo en Desarrollo: N/A	Tiempo Real: 10 días
Descripción:	

1- Objetivo:

Permitir visualizar los indicadores de riesgo asignados al proyecto.

2- Acciones para lograr el objetivo (precondiciones y datos):

Para visualizar los indicadores de riesgo hay que:

- *Estar autenticado en el sistema y ser miembro de al menos un proyecto.*
- *Debe existir en el sistema al menos un proyecto.*

3- Flujo de la acción a realizar:

A partir de la historia de usuario Estado General el usuario despliega el menú y selecciona la opción riesgo se visualiza los indicadores pertinentes al proyecto.

Observaciones: Los indicadores de riesgo manejan la *incertidumbre forma sistemática con la finalidad de disminuir la probabilidad de ocurrencia de resultados negativos.*

Prototipo de interfaz:

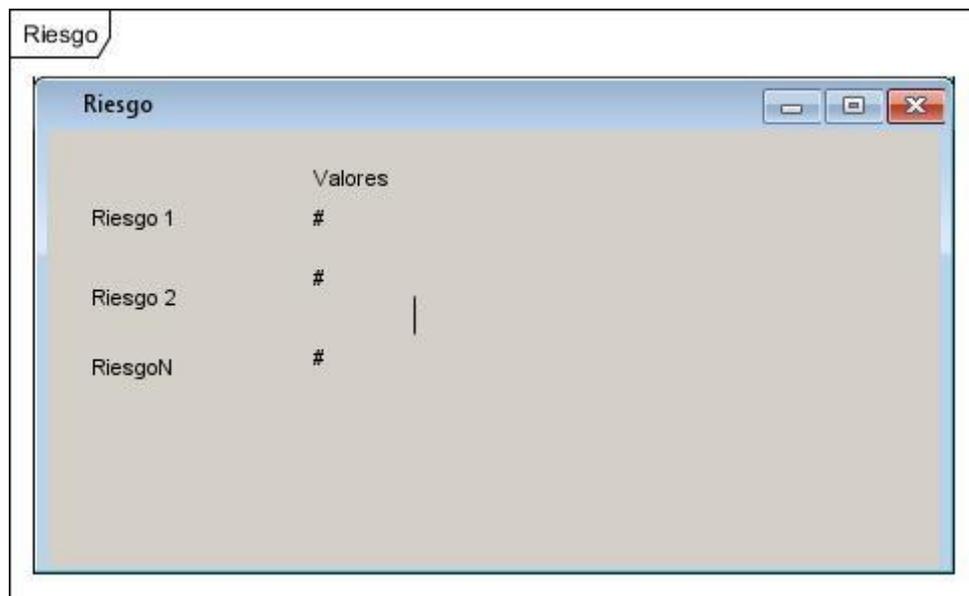


Tabla 22 HU_Tiempo

Número: 9	Nombre del requisito: Visualizar Tiempo de Ejecucion
Programador: Josué Pérez Núñez	Iteración Asignada: 9ta
Prioridad: Baja	Tiempo Estimado: 15 días
Riesgo en Desarrollo: N/A	Tiempo Real: 10 días
<p>Descripción:</p> <p>1- Objetivo:</p> <p><i>Permitir visualizar los indicadores de tiempo de ejecución asignados al proyecto.</i></p> <p>2- Acciones para lograr el objetivo (precondiciones y datos):</p> <p>Para visualizar los indicadores de tiempo de ejecución hay que:</p> <ul style="list-style-type: none"> - <i>Estar autenticado en el sistema y ser miembro de al menos un proyecto.</i> - <i>Debe existir en el sistema al menos un proyecto.</i> <p>3- Flujo de la acción a realizar:</p> <p><i>A partir de la historia de usuario Estado General el usuario despliega el menú y selecciona la opción de tiempo de ejecución por lo que se visualiza los indicadores de tiempo pertinentes al proyecto.</i></p>	
<p>Observaciones: Los indicadores de tiempo de ejecucion manejan el tiempo en que se estima el completamiento del proyecto.</p>	
<p>Prototipo de interfaz:</p>	

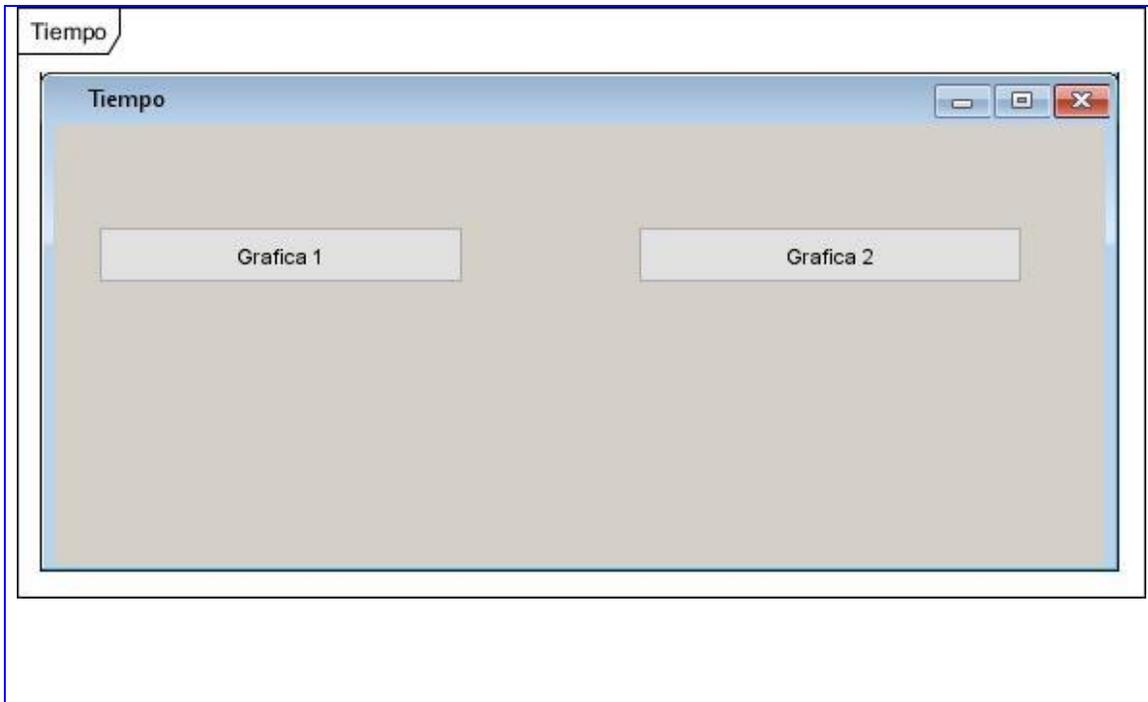


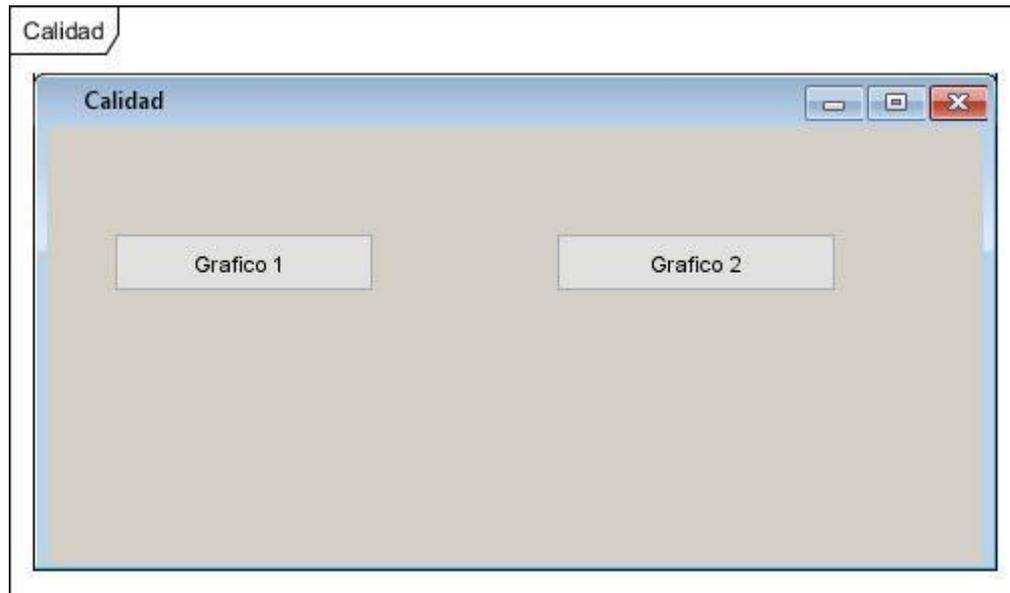
Tabla 23 HU_Calidad

Número: 9	Nombre del requisito: Visualizar Indicador de Calidad
Programador: Josué Pérez Núñez	Iteración Asignada: 10
Prioridad: Baja	Tiempo Estimado: 15 días
Riesgo en Desarrollo: N/A	Tiempo Real: 10 días
<p>Descripción:</p> <p>1- Objetivo:</p> <p><i>Permitir visualizar los indicadores de calidad asociado a un proyecto</i></p> <p>2- Acciones para lograr el objetivo (precondiciones y datos):</p> <p>Para visualizar los indicadores de calidad hay que:</p> <ul style="list-style-type: none"> - <i>Estar autenticado en el sistema y ser miembro de al menos un proyecto.</i> - <i>Debe existir en el sistema al menos un proyecto.</i> <p>3- Flujo de la acción a realizar:</p> <p><i>A partir de la historia de usuario Estado General el usuario despliega el menú y selecciona la</i></p>	

opción de tiempo calidad donde se muestran los indicadores de calidad asociado a un proyecto asignado

Observaciones: Los indicadores de calidad manejan las no conformidades pertenecientes a un proyecto vinculado.

Prototipo de interfaz:



Anexo 7.Casos de Pruebas

Descripción General: La historia de usuario se inicia luego de que el usuario se ha autenticado donde se muestran los proyectos a los cuales está vinculado.

Condiciones de Ejecución: El usuario debe de estar autenticado.

Secciones a probar en la historia de usuario.

A continuación, en las figuras de la 6 y 7 se muestran las similitudes que presenta la aplicación Xedro-Gespro en su versión web y en su versión móvil con los proyectos asociados a un usuario.



Figura 23 Visualizar Proyectos Versión Web

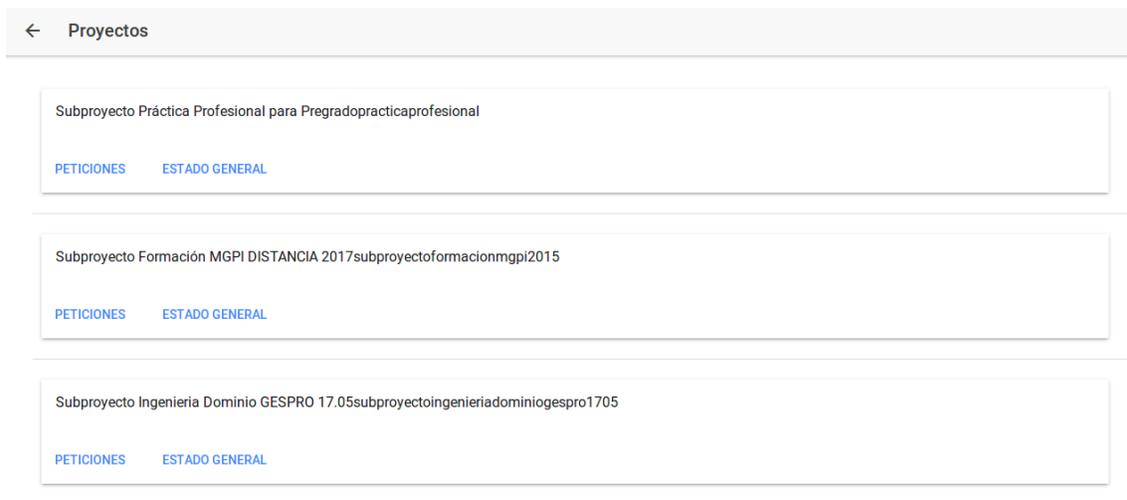


Figura 24 Visualizar Proyectos Versión Móvil

Descripción General: La historia de usuario se inicia luego de que el usuario visualiza sus proyectos donde puede seleccionar el indicador de estado general que tiene ese proyecto.

Condiciones de Ejecución: El usuario debe de estar autenticado y estar vinculado al menos a un proyecto.

Secciones a probar en la historia de usuario.

A continuación, en las figuras de la 8 y 9 se muestran las similitudes que presenta la aplicación Xedro-Gespro en su versión web y en su versión móvil para el indicador de estado general.



Figura 25 Visualizar Estado General Versión Web

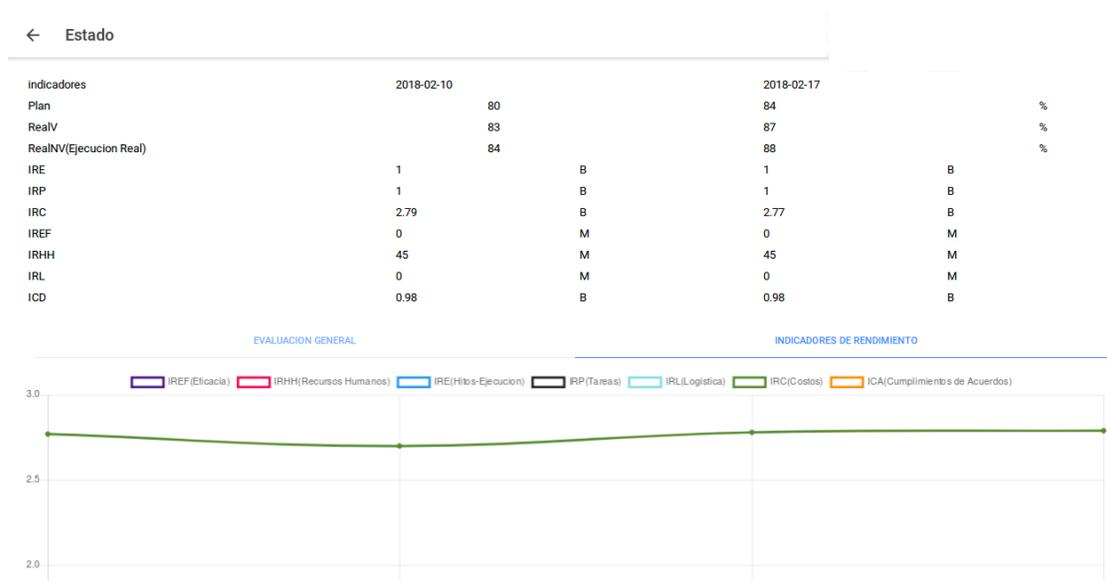


Figura 26 Visualizar Estado General Versión Móvil

Descripción General: La historia de usuario se inicia luego de que el usuario visualiza sus proyectos y accede al indicador de estado general y despliega el menú para acceder al indicador de costo.

Condiciones de Ejecución: El usuario debe de estar autenticado y estar vinculado al menos a un proyecto.

Secciones a probar en la historia de usuario.

A continuación, en las figuras de la 10 y 11 se muestran las similitudes que presenta la aplicación Xedro-Gespro en su versión web y en su versión móvil para el indicador de costo.

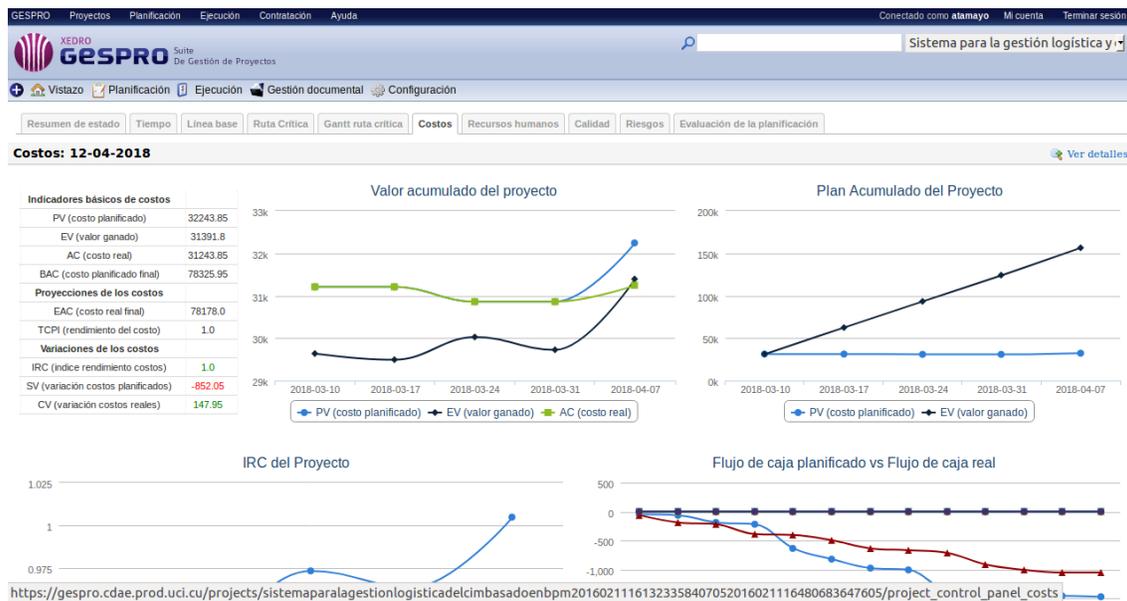


Figura 27 Visualizar Costo Versión Web



Figura 28 Visualizar Costo Versión Móvil

Descripción General: La historia de usuario se inicia luego de que el usuario visualiza sus proyectos y accede al indicador de estado general y despliega el menú para acceder al indicador de tiempo.

Condiciones de Ejecución: El usuario debe de estar autenticado y estar vinculado al menos a un proyecto.

Secciones a probar en la historia de usuario.

A continuación, en las figuras de la 12 y 13 se muestran las similitudes que presenta la aplicación Xedro-Gespro en su versión web y en su versión móvil para el indicador de tiempo.

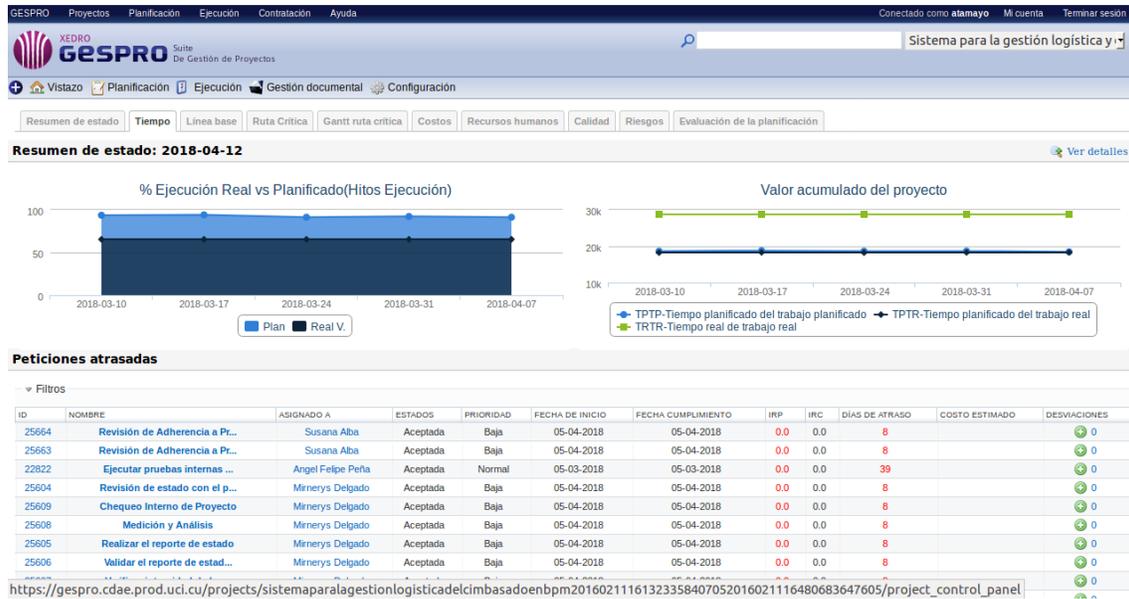


Figura 29 Visualizar Tiempo Versión Web

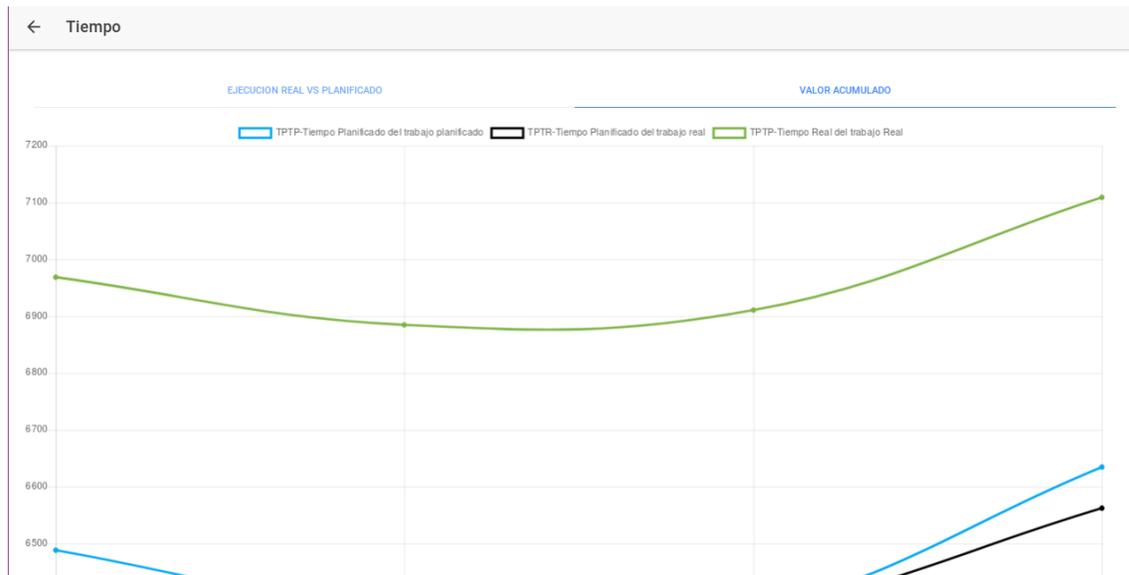


Figura 30 Visualizar Tiempo Versión Móvil

Descripción General: La historia de usuario se inicia luego de que el usuario visualiza sus proyectos y accede al indicador de estado general y despliega el menú para acceder al indicador de recursos humanos.

Condiciones de Ejecución: El usuario debe de estar autenticado y estar vinculado al menos a un proyecto.

Secciones a probar en la historia de usuario.

A continuación, en las figuras de la 14 y 15 se muestran las similitudes que presenta la aplicación Xedro-Gespro en su versión web y en su versión móvil para el indicador de recursos humanos.



Figura 31 Visualizar Recursos Humanos Versión Web

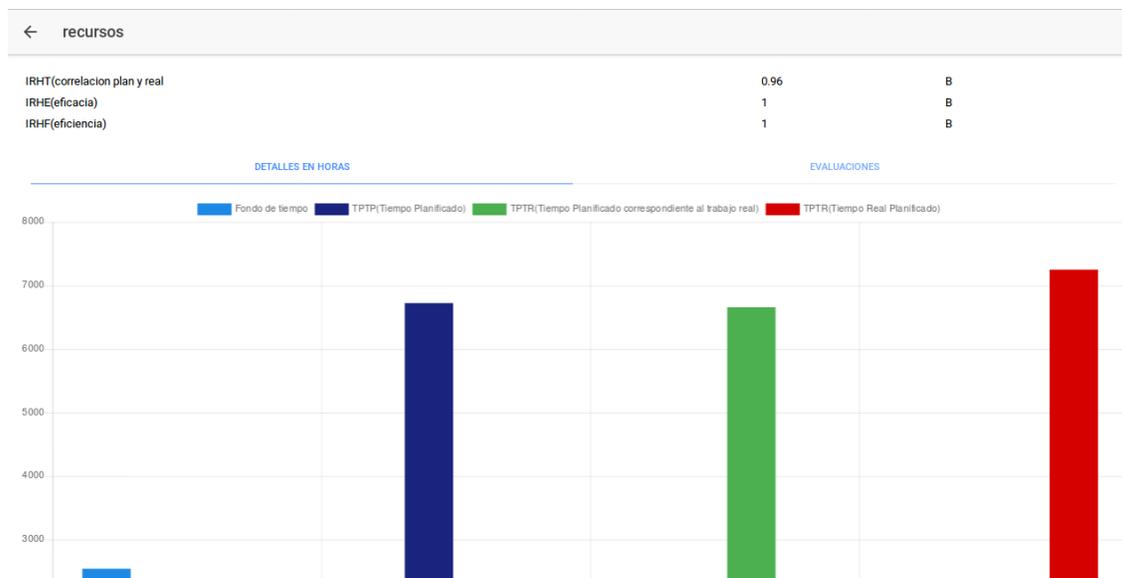


Figura 32 Visualizar Recursos Humanos Versión Móvil

Descripción General: La historia de usuario se inicia luego de que el usuario visualiza sus proyectos y accede al indicador de estado general y despliega el menú para acceder al indicador de calidad.

Condiciones de Ejecución: El usuario debe de estar autenticado y estar vinculado al menos a un proyecto.

Secciones a probar en la historia de usuario.

A continuación, en las figuras de la 16 y 17 se muestran las similitudes que presenta la aplicación Xedro-Gespro en su versión web y en su versión móvil para el indicador de calidad.



Figura 33 Visualizar Indicadores de Calidad Versión Web

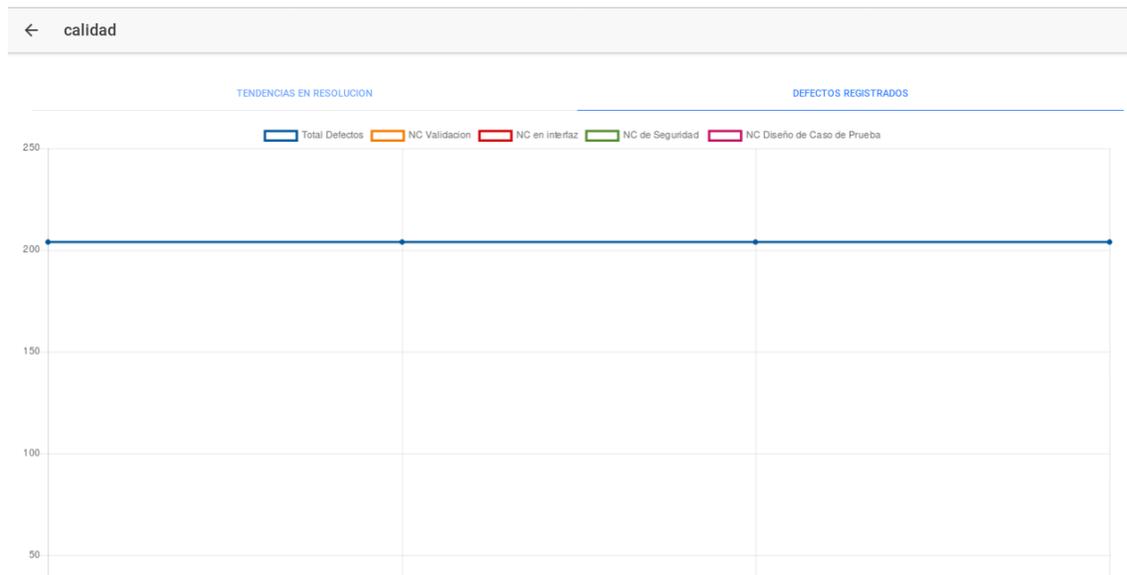


Figura 34 Visualizar Indicadores de Calidad Versión Móvil

Descripción General: La historia de usuario se inicia luego de que el usuario visualiza sus proyectos y accede al indicador de estado general y despliega el menú para acceder al indicador de riesgo.

Condiciones de Ejecución: El usuario debe de estar autenticado y estar vinculado al menos a un proyecto.

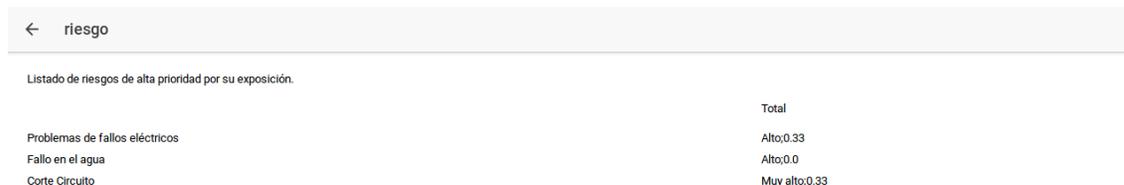
Secciones a probar en la historia de usuario.

A continuación, en las figuras de la 18 y 19 se muestran las similitudes que presenta la aplicación Xedro-Gespro en su versión web y en su versión móvil para el indicador de riesgo.

The screenshot displays the Xedro-Gespro web application interface. At the top, there is a navigation bar with tabs for 'Proyectos', 'Planificación', 'Ejecución', 'Contratación', and 'Ayuda'. The user is logged in as 'atanayo'. Below the navigation bar, there is a search bar and a dropdown menu for 'Sistema para la gestión logística y...'. The main content area is divided into several sections. On the left, there is a sidebar with tabs for 'Resumen de estado', 'Tiempo', 'Línea base', 'Ruta Crítica', 'Gantt ruta crítica', 'Costos', 'Recursos humanos', 'Calidad', 'Riesgos', and 'Evaluación de la planificación'. The 'Riesgos' tab is selected, showing a 'Listado de riesgos de alta prioridad por su exposición.' table. The table has three columns: 'Riesgo', 'Muy alto:0.33', and 'TOTAL'. The rows are: 'Dominio de la tecnología' (Alto:0.0), 'Problemas con el levantamiento de requisitos' (Muy alto:0.33), and 'Licencia del producto' (Muy alto:0.33). On the right side, there are two sections: 'Riesgos' and 'Desviaciones'. The 'Riesgos' section lists: 'Riesgos de mayor prioridad.', 'Exposición a los riesgos', 'Impacto de los riesgos', 'Probabilidad de ocurrencia', 'Estado de mitigación de los riesgos', and 'Categoría de riesgos'. The 'Desviaciones' section lists: 'Desviaciones relevantes', 'Tipos de desviaciones', and 'Impacto de las desviaciones'. At the bottom of the page, there is a footer with the text: 'Project Management Suite (GESPRO 18.05), Research Laboratory in Project Management, UCI. Powered by GESPRO, PostgreSQL, D3, Redmine © 2006-2017 maestriapp@ucl.ac.uk, gestion.proyectos@ucl.ac.uk, ppp@ucl.ac.uk'.

Riesgo	Muy alto:0.33	TOTAL
Dominio de la tecnología	Alto:0.0	
Problemas con el levantamiento de requisitos	Muy alto:0.33	
Licencia del producto	Muy alto:0.33	

Figura 35 Visualizar Indicadores de Riesgo Versión Web



riesgo	
Listado de riesgos de alta prioridad por su exposición.	
Problemas de fallos eléctricos	Total Alto,0.33
Fallo en el agua	Total Alto,0.0
Corte Circuito	Total Muy alto,0.33

Figura 36 Visualizar Indicadores de Riesgo Versión Móvil

Descripción General: La historia de usuario se inicia luego de que el usuario visualiza sus proyectos y accede a las peticiones asignadas al proyecto.

Condiciones de Ejecución: El usuario debe de estar autenticado y estar vinculado al menos a un proyecto.

Secciones a probar en la historia de usuario.

A continuación, en las figuras de la 20 y 21 se muestran las similitudes que presenta la aplicación Xedro-Gespro en su versión web y en su versión móvil para las peticiones.

✓ Aceptar Anular Guardar

<input type="checkbox"/>	PROYECTO	ESTADOS	ASUNTO	AUTOR	ASIGNADO A	FECHA DE INICIO	FECHA CUMPLIMIENTO	% REALIZ
<input type="checkbox"/>	25339 Subproyecto Formación MGPI DISTANCIA 2017	Asignada	Creación de OA+imágenes para lección 1 curso de gestión de la calidad	Laynier Piedra	Laynier Piedra	19/02/2018	26/02/2018	
<input type="checkbox"/>	25336 Subproyecto Formación MGPI DISTANCIA 2017	Resuelta	Organizar seminario de tesis para la maestría	Yairilee Cruz	Yairilee Cruz	15/02/2018	21/02/2018	
<input type="checkbox"/>	25335 Subproyecto Formación MGPI DISTANCIA 2017	Aceptada	Enviar convocatoria para el curso de liderazgo	Yairilee Cruz	Yairilee Cruz	19/02/2018	07/03/2018	
<input type="checkbox"/>	25334 Subproyecto Formación MGPI DISTANCIA 2017	Aceptada	Enviar convocatoria para el curso libre de gestión de riesgos	Yairilee Cruz	Yairilee Cruz	15/02/2018	07/03/2018	
<input type="checkbox"/>	25333 Subproyecto Formación MGPI DISTANCIA 2017	Aceptada	Crear grupos en el curso básico en la edición 1	Yairilee Cruz	Yairilee Cruz	15/02/2018	28/02/2018	
<input type="checkbox"/>	25332 Subproyecto Formación MGPI DISTANCIA 2017	Nueva	Actualizar preguntas autoevaluación y evaluaciones tema 7	Yairilee Cruz	Yairilee Cruz	22/02/2018	28/02/2018	
<input type="checkbox"/>	25331 Subproyecto Formación MGPI DISTANCIA 2017	Aceptada	Actualizar presentación y lección tema 7	Yairilee Cruz	Yairilee Cruz	12/02/2018	21/02/2018	
<input type="checkbox"/>	25330 Subproyecto Formación MGPI DISTANCIA 2017	Aceptada	CBGP Gestión de no conformidades	Yairilee Cruz	Yairilee Cruz	12/02/2018	28/02/2018	

Figura 37 Peticiones Versión Web

← peticiones

#	Estados	Asunto	Asignado A	Fecha de Inicio	Fecha Fin
2069 4	Asignada	trabajo con robot de audio a seguir durante la creacion del video	Josue Perez	2016-06-06	2016-06-17
2069 3	Asignada	Redaccion de guión	Josue Perez	2016-06-13	2016-06-17
2069 2	Asignada	intalacion de herraminetas para los videos	Josue Perez	2016-06-06	2016-06-10
2069 1	Asignada	Creación de videos de auto ayuda en la herramineta Xedro Gespro	Josue Perez	2016-06-06	2016-06-17
2056 4	Nueva	T Descrpción la evaluación de competencias laborales en el XEDRO-GESPRO	Liosvel Medina	2016-06-13	2016-06-22
2056 3	Asignada	T Describir la relación usuario-rol-competencia-dimensiones	Liosvel Medina	2016-05-30	2016-06-14

Figura 38 Peticiones Versión Móvil