



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE CIENCIAS DE LA
COMPUTACIÓN**

CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

**PROYECTO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO EN SISTEMAS E INFORMÁTICA**

**TEMA: DESARROLLO DE UNA APLICACIÓN MÓVIL
MULTIPLATAFORMA UTILIZANDO REALIDAD AUMENTADA
PARA LA ADMINISTRACIÓN DE LA CARGA HORARIA EN
LOS LABORATORIOS GENERALES DE COMPUTACIÓN DE
LA UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
MATRIZ.**

AUTOR: PAMELA ESTEFANÍA UTRERAS COLLAGUAZO

DIRECTOR: VILLACÍS SILVA, CÉSAR JA

SANGOLQUÍ, AGOSTO 2017



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

CERTIFICACIÓN

Certifico que el trabajo de titulación, **“DESARROLLO DE UNA APLICACIÓN MÓVIL MULTIPLATAFORMA UTILIZANDO REALIDAD AUMENTADA PARA LA ADMINISTRACIÓN DE LA CARGA HORARIA EN LOS LABORATORIOS GENERALES DE COMPUTACIÓN DE LA UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE MATRIZ”** realizado por la señora **PAMELA ESTEFANÍA UTRERAS COLLAGUAZO**, ha sido revisada en su totalidad y cumple con los requisitos teóricos, científicos, técnicos, metodologías y legales establecidos por la Universidad de Fuerzas Armadas ESPE, por lo tanto me permito acreditarlo y autorizar a la señora **PAMELA ESTEFANÍA UTRERAS COLLAGUAZO** para que lo sustente públicamente.

Sangolquí, 23 agosto de 2017

Ing. César Javier Villacís Silva

DIRECTOR





DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

AUTORÍA DE RESPONSABILIDAD

Yo, **PAMELA ESTEFANÍA UTRERAS COLLAGUAZO**, con cédula de identidad N° 1721333415, declaro que este trabajo de titulación **“DESARROLLO DE UNA APLICACIÓN MÓVIL MULTIPLATAFORMA UTILIZANDO REALIDAD AUMENTADA PARA LA ADMINISTRACIÓN DE LA CARGA HORARIA EN LOS LABORATORIOS GENERALES DE COMPUTACIÓN DE LA UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE MATRIZ”** ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente declaro que este trabajo es de mi autoría, en virtud de ello me declaro responsable del contenido, veracidad y alcance de la investigación mencionada.

Sangolquí, agosto de 2017

Pamela Estefanía Utreras Collaguazo

CC: 172133341-5



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

AUTORIZACIÓN

Yo, **PAMELA ESTEFANÍA UTRERAS COLLAGUAZO**, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar en la biblioteca Virtual de la institución el presente trabajo de titulación **“DESARROLLO DE UNA APLICACIÓN MÓVIL MULTIPLATAFORMA UTILIZANDO REALIDAD AUMENTADA PARA LA ADMINISTRACIÓN DE LA CARGA HORARIA EN LOS LABORATORIOS GENERALES DE COMPUTACIÓN DE LA UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE MATRIZ”** cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.

Sangolquí, agosto de 2017.

Pamela Estefanía Utreras Collaguazo

CC: 172133341-5

DEDICATORIA

El presente trabajo deseo dedicar con mucho cariño a todas las personas que han formado y seguirán formando parte de mi vida y mis sueños.

A Dios por demostrarme cada día que con mucha paciencia y sabiduría todo puede ser posible.

A mi familia y en especial a mis padres quienes han sido ejemplo de vida y que con su amor y apoyo incondicional estuvieron a lo largo de mi vida para poder cumplir cada una de mis metas.

A mis hermanas que han estado conmigo en los buenos y malos momentos y que con un consejo o su hermosa compañía lo han hecho todo más sencillo.

A mi esposo Luisito y a mi hijo Martincito que son el pilar y el soporte de mi vida y que gracias al amor que formamos cada día seguiremos luchando por alcanzar nuestras futuras metas.

Pamela Estefanía Utreras Collaguazo

AGRADECIMIENTO

Agradezco a Dios por permitirme recorrer un camino lleno de enseñanzas y alegría con las personas que me rodean y amo.

A mis padres y hermanas por enseñarme lo bella que es la vida compartiendo los mejores momentos en familia.

A mi esposo e hijito que con su infinito amor hicieron de este trabajo algo muy especial para ustedes.

A mis amigos que sin duda alguna hicieron de mi carrera universitaria algo muy divertido y siempre supimos apoyarnos en cualquier momento y circunstancia.

A mi Director de Trabajo de Titulación, Ing. César Villacís, por compartir conmigo su tiempo, dedicación y conocimientos, para hacer posible este trabajo.

Por último, agradezco a la Universidad de las Fuerzas Armadas – ESPE por acogerme en sus aulas y aportar con ese granito de arena para poder ser una excelente profesional.

Pamela Estefanía Utreras Collaguazo

ÍNDICE

CARÁTULA	
CERTIFICACIÓN	ii
AUTORIA DE RESPONSABILIDAD	iii
AUTORIZACION	iv
DEDICATORIA	v
AGRADECIMIENTO	vi
ÍNDICE	vii
ÍNDICE DE TABLAS	xi
ÍNDICE DE FIGURAS	xii
RESUMEN	xiv
ABSTRACT	xv
CAPÍTULO I	1
INTRODUCCIÓN	
1.1 Antecedentes.....	1
1.2 Problemática.....	2
1.3 Justificación.....	2
1.4 Objetivos.....	3
1.4.1 Objetivo General.....	3
1.4.2 Objetivos específicos.....	3
1.5 Alcance.....	4
CAPITULO II	6
MARCO TEÓRICO	
2.1 Metodología de Desarrollo de Software.....	6
2.1.1 El Manifiesto Ágil.....	6

2.1.2 SCRUM.....	10
2.1.3 Extreme Programming.....	15
2.2 SCRUM combinado con Extreme Programming	20
2.2.1 Actividades, artefactos y herramientas tomadas para SCRUM ...	20
2.2.2 Prácticas tomadas para Extreme Programming	21
2.3 Realidad Virtual	21
2.3.1 Las Tres i's de la Realidad Virtual	22
2.2.2 Características de las Realidad Virtual	23
2.4 Realidad Aumentada	23
2.4.1 Características de la Realidad Aumentada.....	24
2.4.2 Tipos de Realidad Aumentada.....	24
2.4.3 Elementos de la Realidad Aumentada.....	26
2.5 Framework de Desarrollo	26
2.5.1 Eclipse	27
2.5.2 Unity 3D.....	27
2.5.3 Entorno de Unity 3D.....	27
2.5.4 Visual Studio.....	28
2.6 Tecnologías de Desarrollo.....	29
2.6.1 Glassfish	29
2.6.2 MySql.....	29
2.6.3. Vuforia	30
2.6.4 Java 1.8	31
2.6.5 C#	33
CAPITULO III.....	34
ESPECIFICACIÓN DE REQUERIMIENTOS	
3.1 Introducción	34

3.2 Propósito	34
3.3 Alcance	34
3.4 Limitaciones del Proyecto	35
3.5 Personal Involucrado	36
3.6 Definiciones	37
3.7 Acrónimos	37
3.8 Referencias	37
3.9 Visión General del Documento	38
3.10 Descripción General	38
3.10.1 Perspectiva del Producto	38
3.10.2 Funciones del Producto	39
3.10.3 Características de los Usuarios	39
3.10.4 Restricciones	40
3.10.5 Suposiciones y Dependencias	41
3.11 Requisitos Específicos	41
CAPÍTULO IV	52
DESARROLLO DEL PROYECTO	
4.1 Diseño de la Base de Datos	52
4.2 Modelo de Datos	52
4.2.1 Modelo Conceptual	53
4.2.2 Modelo Físico	54
4.3 Diagramas de Casos de Uso	54
4.4 Diagrama de Secuencia	56
4.5 Diagrama de Clases	58
4.6 Diagrama de Arquitectura	59
CAPÍTULO V	61

PLANIFICACIÓN, DESARROLLO Y PRUEBAS

5.1 Planificación del proyecto utilizando la metodología SCRUM	61
5.2 Desarrollo y pruebas del Sprint 1.....	63
5.2.1 Sprint Backlog del Sprint 1.....	65
5.2.2 Revisión y seguimiento del Sprint 1	67
5.2.3 Pruebas del Sprint 1	70
5.3 Desarrollo y pruebas de Sprint 2	75
5.3.1 Sprint Backlog del Sprint 2.....	75
5.3.2 Revisión y seguimiento del Sprint 2.....	76
5.3.4 Pruebas del Sprint 2	79
5.4 Desarrollo y pruebas del Sprint 3.....	82
5.4.1 Sprint Backlog del Sprint 3.....	83
5.4.2 Revisión y seguimiento del Sprint 3.....	85
5.4.3 Pruebas del Sprint 3	88
CAPÍTULO VI	95
CONCLUSIONES Y RECOMENDACIONES	
6.1 Conclusiones	95
6.2 Recomendaciones.....	96
REFERENCIAS BIBLIOGRAFICAS	98
ANEXOS.....	101
ANEXO A: MANUAL DE USUARIO.....	101
ANEXO B: EJEMPLO CÓDIGO FUENTE	108

ÍNDICE DE TABLAS

Tabla 1 Resumen de los principios del Manifiesto	7
Tabla 2 Actividades de un Sprint	12
Tabla 3 Roles de SCRUM.....	13
Tabla 4 Artefactos de SCRUM.....	14
Tabla 5 Proceso y prácticas de Extreme Programming	16
Tabla 6 Roles de Extreme Programming	19
Tabla 7 Vistas de Unity 3D	28
Tabla 8 Arquitectura de Vuforia	30
Tabla 9 Contenedores servidor Java EE	32
Tabla 10 Personal involucrado.....	36
Tabla 11 Características de los usuarios	40
Tabla 12 Restricciones del desarrollador	40
Tabla 13 Requisitos específicos del Administrador Web	41
Tabla 14 Requisitos específicos de la Aplicación Móvil	42
Tabla 15 Product Backlog del Administrador Web.....	62
Tabla 16 Product Backlog de la Aplicación Móvil.....	63
Tabla 17 Historias de usuario para Sprint 1	65
Tabla 18 Sprint Backlog del Sprint 1	65
Tabla 19 Tareas completas del Sprint 1	67
Tabla 20 Historias de usuario para Sprint 2	75
Tabla 21 Sprint Backlog del Sprint 2.....	75
Tabla 22 Tareas completas del Sprint 2	77
Tabla 23 Historias de usuario para Sprint 3.....	83
Tabla 24 Sprint Backlog del Sprint 3.....	83
Tabla 25 Tareas completas del Sprint 3	85

ÍNDICE DE FIGURAS

Figura 1 Ciclo de metodología SCRUM	11
Figura 2 Diagrama Burndown Chart	15
Figura 3 El triángulo de la realidad virtual	23
Figura 4 Ejemplo de Marcador	25
Figura 5 Ejemplo de realidad aumentada en posición	26
Figura 6 Modelo Entidad – Relación	52
Figura 7 Modelo conceptual de la base de datos.....	53
Figura 8 Modelo físico de la base de datos	54
Figura 9 Caso de uso del Administrador Web “Smart Labs”	55
Figura 10 Caso de uso de la Aplicación Móvil “Smart Labs”	55
Figura 11 Escenario de módulo de Docentes de la aplicación móvil	56
Figura 12 Escenario de módulo de Asignaturas en la aplicación móvil	57
Figura 13 Escenario de módulo de Laboratorios en la aplicación móvil	57
Figura 14 Módulo de realidad aumentada en la aplicación móvil	58
Figura 15 Diagrama de clases “Smart Labs”	59
Figura 16 Diseño arquitectónico del sistema	60
Figura 17 Diseño de la arquitectura del sistema	60
Figura 18 Horas de trabajo pendiente para Sprint 1	69
Figura 19 Tareas pendientes para Sprint 1	69
Figura 20 Prueba servicio RESTFul para visualizar por docente.....	70
Figura 21 Prueba visualización por docente	71
Figura 22 Prueba servicio RESTful para visualizar asignatura	72
Figura 23 Prueba visualización asignaturas	73
Figura 24 Prueba servicio RESTFul de visualizar resumen general	74
Figura 25 Prueba resumen general	74
Figura 26 Horas de trabajo pendiente para el Sprint 2	78
Figura 27 Tareas pendientes para el Sprint 2	79
Figura 28 Prueba de servicio RESTFul para Información en tiempo real por piso de la aplicación móvil	80

Figura 29 Prueba de servicio RESTFul para Información en tiempo real por código de laboratorio en la aplicación móvil	80
Figura 30 Información en tiempo real por	81
Figura 31 Información en tiempo real	82
Figura 32 Horas de trabajo pendiente para el Sprint 3	87
Figura 33 Tareas pendientes para el Sprint 3	88
Figura 34 Prueba de caja blanca para Consulta de Docentes	89
Figura 35 Prueba de caja negra para visualización de Docentes	89
Figura 36 Prueba de caja blanca para Consulta de Asignaturas	90
Figura 37 Prueba de caja negra para Visualización de Asignaturas	90
Figura 38 Prueba de caja blanca para Consultar Laboratorios	91
Figura 39 Prueba de caja blanca para Actualización de Laboratorios	92
Figura 40 Prueba de caja negra para Asignación de carga horaria a Laboratorio	92
Figura 41 Prueba de caja negra para Actualización de Laboratorios	93
Figura 42 Prueba de caja blanca para eliminación de carga horaria en los laboratorios	93
Figura 43 Registro de datos eliminados en los laboratorios	94
Figura 44 Prueba de caja negra para eliminación de carga horaria en los laboratorios	94

RESUMEN

La Universidad de las Fuerzas Armadas – ESPE MATRIZ, es un centro de educación superior creada el 16 de junio de 1922, ubicada en Sangolquí, cuenta con Laboratorios de diferentes Facultades de excelente infraestructura, en las que se destacan los Laboratorios Generales de Computación. Sin embargo, actualmente en cada semestre tanto docentes como estudiantes se encuentran en la necesidad de obtener información en tiempo real sobre las materias que se están dictando en cada uno de los Laboratorios y de esta manera poder asistir a la correcta. La evolución de la tecnología en el país permite que se puedan dar soluciones a este tipo de inconvenientes es por ello que se hace hincapié en el desarrollo de una aplicación móvil que facilite el manejo de este tipo de información. Para la elaboración se utilizó las metodologías SCRUM y Extreme Programming (XP) mismas que permiten un manejo y organización adecuado para los requerimientos expuestos al equipo de desarrollo tanto para el Administrador Web que será manejado por los señores Laboristas para la administración de la carga horaria en los laboratorios y de la aplicación móvil que podrá ser utilizada por el público en general, estudiantes y docentes; gracias a las tecnologías de desarrollo proporcionadas por la Institución, las cuales servirán para la implementación una vez finalizado dicho sistema.

PALABRAS CLAVE:

- **SCRUM**
- **XP**
- **JAVA**
- **RESTFuI**
- **REALIDAD AUMENTADA**

ABSTRACT

The University of the Armed Forces – ESPE MATRIZ is a superior education center build at June 16th, 1922. It is located at Sangolquí and has Laboratories to different faculties, each one of excellent infrastructure in which the computer labs standout. However, every semester the teachers and students are in the need of get real time information about the subjects which are being taught in computer labs, so they can go to the correct laboratory to attend classes. The technology evolution in the country allows us to get quickly solutions to this kind of problems, this is why emphasis is placed on the development of a mobile application to ease the use this particularly information. For the elaboration of this project was used Agile Methodologies as SCRUM and Extreme Programming (XP), same that allows a suitable use and organization for the requirements exposed by the development team both WEB Administration which will be managed by laboratory workers to configure the workload in each laboratory, and the Mobile Application which can be used by anyone (teachers and students) with a smartphone. Thanks to development technologies provided by University of the Armed Forces – ESPE MATRIZ, the project can be deployed once the development is finished.

KEYWORDS:

- **SCRUM**
- **XP**
- **JAVA**
- **RESTFul**
- **AUGMENTED REALITY**

CAPÍTULO I

INTRODUCCIÓN

1.1 Antecedentes

La Universidad de las Fuerzas Armadas – ESPE MATRIZ, es un centro de educación superior creada el 16 de junio de 1922, ubicada en Sangolquí, e identificada como una Universidad de prestigio en el Ecuador, cuenta con Departamentos para la identificación de carreras. La Universidad cuenta con Laboratorios de diferentes Facultades de excelente infraestructura, en las que se destacan los Laboratorios Generales de Computación, que cuenta aproximadamente con 36 laboratorios especializados los cuales están a cargo de un Jefe y administrados por los señores Laboratoristas.

Cada semestre los Laboratorios están preparados para la nueva carga horaria, después de haber finalizado el proceso de matrículas el cual tiene una duración de aproximadamente dos semanas posteriormente la información de la carga horaria es entregada a cada Docente, y para los estudiantes dicha información es visualizada a través del Sistema Académico Banner.

Actualmente la Universidad de las Fuerzas Armadas – ESPE MATRIZ, no cuenta con una aplicación que permita solventar la necesidad de mostrar con facilidad y de manera rápida y sencilla la información de esta carga horaria en la que se pueda visualizar específicamente en que laboratorio de Computación se encuentra un Docente y que materia se encuentra dictando en ese momento, por lo que tienen que recurrir al ingreso del Sistema Académico y buscar en el horario de clases, la información necesitada.

La tendencia en el país en cuanto a tecnología incrementa conforme el tiempo avanza, por lo que los Smartphone llegan a ser una vía de comunicación e interacción entre las personas y el mundo. Gracias a los mismos, las aplicaciones también hacen su presencia para adaptarse a estos dispositivos los cuales brindaran soluciones a grandes problemáticas.

1.2 Problemática

Cada período académico, tanto Docentes como estudiantes de la Universidad de las Fuerzas Armadas – ESPE MATRIZ, se encuentran con un gran dilema, el cual es el de recordar con facilidad: el curso, la materia, el NRC, y otro tipo de información, el cual se les han sido asignados. La mayoría del personal de la Universidad optan por el proceso de ingresar al Sistema Académico Banner y revisar la información anteriormente detallada, generando en varias ocasiones que los estudiantes al no conocer estos parámetros se retiren de los laboratorios, acumulando faltas y perdiendo horas clases.

De igual manera, los Docentes disponen de un informe de la materia que se encuentran dictando en el presente período académico. Ellos acuden a este tipo de informe para conocer en qué aula y en qué horario deben impartir sus clases. Estos informes que están disponibles para los Docentes tienden a perderse, esto implica que se lo vuelva a imprimir y posteriormente se encuentre en la incertidumbre de tener la información disponible al momento.

Cuando los Docentes desean cambiar de aula, existe un gran inconveniente el cual es el de poder informarle a sus alumnos sobre este cambio, motivo por el cual los estudiantes al no conocer de esta decisión optan por sobre entender que no recibirán clases ese día, finalmente no asisten, o asisten de forma atrasada a las mismas.

1.3 Justificación

Debido a la necesidad de los Docentes y de los estudiantes por disponer con facilidad información con respecto a la carga horaria en el semestre que se encuentra cursando, existe la importancia de otorgar al personal relacionado a la Facultad una aplicación móvil que permita solucionar esta problemática.

En la Facultad de Ingeniería en Sistemas e Informática de la Universidad de las Fuerzas Armadas – ESPE MATRIZ, existen

aproximadamente 300 alumnos y 40 Docentes. Y se aproxima que el 90% de las personas indicadas anteriormente disponen de un Smartphone o dispositivo electrónico como lo puede ser una Tablet. Mediante el uso de tecnología innovadora como es el concepto de Realidad Aumentada y gracias a las Metodologías Ágiles de Desarrollo, los usuarios con esta aplicación podrán:

- Interactuar de forma dinámica con la información presentada a través de esta, con el simple hecho de capturar una imagen y desplegar la información necesaria para ellos.
- Brindar respuesta inmediata en las consultas de carga horaria de cada uno de los Docentes que dicten materias en los Laboratorios Generales de Computación de la Universidad de las Fuerzas Armadas – ESPE MATRIZ.
- Evitar inconvenientes en el transcurso del período académico que se encuentra cursando tanto por parte de los Docentes y de los alumnos que se encuentran matriculados en materias que hagan uso de los Laboratorios Generales de Computación.

1.4 Objetivos

1.4.1 Objetivo General

Desarrollar una aplicación móvil multiplataforma utilizando realidad aumentada para la administración de la carga horaria en los laboratorios de Computación de la Universidad de las Fuerzas Armadas – ESPE MATRIZ.

1.4.2 Objetivos Específicos

- Revisar el marco teórico referente a la metodología SCRUM y Extreme Programming para el desarrollo de sistemas de información.

- Realizar la especificación de requerimientos de la aplicación aplicando la norma IEEE830.
- Aplicar las metodologías SCRUM y Extreme Programming para el diseño y desarrollo del sistema.
- Realizar las pruebas de caja blanca y caja negra del sistema.

1.5 Alcance

Se desarrollará una aplicación móvil para los Laboratorios Generales de Computación de la Universidad de las Fuerzas Armadas – ESPE MATRIZ, la misma que al ser puesta en ejecución cumplirá las siguientes funcionalidades:

- La aplicación móvil debe cubrir las necesidades de: Obtener información sobre el Docente que se encuentra en cada uno de los Laboratorios en el que constará; Foto, Nombres y Apellidos del Docente, el nombre de la Materia que dicta, y el NRC.
- Se tomará en cuenta los laboratorios que se encuentren disponibles para el uso de cualquier Docente de la Universidad.

La aplicación comprenderá de los siguientes módulos:

- Módulo de Servicios

Este módulo constará de una capa de servicios REST expuestos en un servidor de Aplicaciones Glassfish, el cual permitirá interactuar con la aplicación móvil y enviar la información deseada por el usuario final.

- Módulo Consultas por Imagen (3D - R.A.)

Es el encargado de obtener información y mostrarla en forma de un mapa de ruta en la aplicación móvil; esto consiste en capturar la imagen y posteriormente enlazarla con la información respectiva.

- Módulo Consultas por Búsqueda

Es el encargado de obtener información y mostrarla en forma de una lista plana en la aplicación móvil; esto consiste en realizar una búsqueda

simple seleccionando los filtros deseados y posteriormente visualizar el resultado en modo de una lista de información.

- Administrador Web

Este módulo será manejado por una persona encargada Laboratorista y permitirá presentar una lista de docentes y laboratorios, los cuáles a través de este podrán ser asignados en el horario correspondiente, permitiendo la interoperabilidad entre la aplicación y el usuario.

CAPÍTULO II

MARCO TEÓRICO

Para el desarrollo de una aplicación móvil multiplataforma amigable, flexible y sostenible es necesario basarse en una metodología de desarrollo de software y herramientas, que permitan cumplir con los objetivos específicos planteados y finalizar la aplicación en el tiempo estimado.

La combinación de metodologías de desarrollo ya sea en sistemas o aplicativos móviles tiene gran auge por las ventajas que este implica, es decir, optimizan el tiempo de desarrollo de los mismos, existe alineamiento entre el cliente y el desarrollador y permiten elaborar y entregar un software de calidad, de esta manera se recalca lo siguiente: “La ingeniería de software ágil representa una alternativa razonable a la ingeniería de software convencional para ciertas clases de software y en algunos tipos de proyectos. Asimismo, se ha demostrado que concluye con rapidez sistemas exitosos”. (Pressman, 2010).

2.1 Metodología de Desarrollo de Software

Las metodologías escogidas para el desarrollo de la aplicación móvil son dos: SCRUM dirigida para la planificación y Extreme Programming para la programación del software.

2.1.1 El Manifiesto Ágil

El manifiesto ágil define las mejores prácticas para el desarrollo de software, sigue doce principios, los cuales diferencian un proceso ágil y surgen como alternativa de cualquier proceso tradicional. En la tabla 1 se resumen los principios del Manifiesto ágil.

Tabla 1

Resumen de los principios del Manifiesto

Principio	Descripción
I “Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.”	Se considera ágil, todo aquel proceso que a sus pocas semanas de haber empezado ya se poseen entregables. Pequeñas partes de software las cuales el cliente va a ser la persona que decida si poner en marcha la entrega, o revisar y emitir un informe con posibles cambios y mejoras que se puedan realizar.
II “Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.”	Siempre se visualiza a los cambios como algo malo, pero se debe ver que es totalmente lo opuesto. Los cambios tienen que verse de forma positiva. Se debe considerar siempre una arquitectura o estructura del software flexible, lo cual permita ejecutar los cambios de forma rápida y efectiva. Por ejemplo, el paradigma orientado a objetos permite conseguir esta flexibilidad.
III “Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.”	Al ejecutar estas entregas funcionales, permite al equipo evaluar los cambios, si existen, para que el producto final sea totalmente robusto y cumpla todas las necesidades del cliente. Mientras más entregas se realicen, será mejor debido a los cambios que surjan.

CONTINÚA ⇒

<p>IV “Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.”</p>	<p>El trabajo en equipo no solo de forma individual durante el transcurso del proyecto permite generar un producto muy estable con todas las funcionalidades necesarias para satisfacer las necesidades del cliente, o el usuario final del producto o sistema.</p>
<p>V “Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.”</p>	<p>El principal factor de éxito de un proyecto son las personas involucradas. Si un solo miembro tiene un efecto negativo, esto tiende a ser propagado hacia el resto de personas en el equipo. Por lo tanto, siempre deben ser motivados para ejecutar un buen desempeño con el resto del equipo.</p>
<p>VI “El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.”</p>	<p>Considerado como el principal modo de comunicación. Por muchos documentos o actas que se realicen, es preferible hacerlo “cara a cara”, de esta forma queda claro los requerimientos y que lo que se pide que se realice. Y en ese mismo momento intentar solventar toda duda que se tenga.</p>
<p>VII “El software funcionando es la medida principal de progreso.”</p>	<p>Esto nos indica que el software funcional que se entrega en pequeños intervalos de tiempo, o conocido como entregas parciales, permite al cliente o usuario final identificar el porcentaje de progreso acorde a los requerimientos propuestos.</p>

CONTINÚA ⇒

<p>VIII “Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.”</p>	<p>El concepto de los procesos ágiles no se trata de desarrollar lo más rápido posible. Esto puede conllevar a tener errores en código o en ejecución. Los procesos ágiles proponen mantener un mismo ritmo en la duración del proyecto y de esta manera asegurar la calidad del desarrollo.</p>
<p>IX “La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.”</p>	<p>Se puede considerar clave el realizar código claro y legible, no solo por el programador, sino por el resto del equipo, esto puede permitir avanzar de forma más rápida en el proyecto.</p>
<p>X “La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.”</p>	<p>Esto indica que no es necesario tener un código con alta complejidad, sino todo lo contrario. Todo código debe ser simple y sobre todo con alta calidad, lo cual va a permitir acoger cambios más rápidamente.</p>
<p>XI “Las mejores arquitecturas, requisitos y diseños emergen de equipos auto organizados.”</p>	<p>La mejor practica que se puede llevar en un equipo, es la comunicación. Cada uno de los miembros del equipo conoce las responsabilidades del resto, y así se pueden dividir las tareas a ejecutar.</p>

CONTINÚA ⇒

XII “A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para la continuación ajustar y perfeccionar su comportamiento consecuencia.” El entorno en el cual se desenvuelve el proyecto siempre se encuentra en un estado cambiante, esto nos indica que el o los equipos deben acoplarse a esto. Así asegurar el conocimiento en actual y mantener una mejora continua en el proyecto con las nuevas prácticas y tecnologías que salen en el mercado.

Fuente: Adaptado (Herrera Uribe & Valencia Ayala, 2007)

2.1.2 SCRUM

Es una metodología ágil de desarrollo basado en un marco de trabajo de procesos que involucra trabajar de forma dinámica, ordenada y colaborativa con las personas implicadas para un proyecto. La implementación de SCRUM en el desarrollo de un software hace que los requisitos puedan ser cambiantes, sin embargo, permite obtener de forma rápida resultados y ser presentados al cliente oportunamente.

SCRUM es una metodología de desarrollo muy simple, que requiere trabajo duro porque no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias de la evolución de un proyecto. Se orienta a las personas más que a los procesos y emplea la estructura de desarrollo ágil: incremental basada en iteraciones y revisiones. (Anónimo, s.f.)

Los procesos de SCRUM abordan las actividades a realizar y el flujo específico para el proyecto, y el cliente es quien queda satisfecho al comprometerse con el proyecto iteración tras iteración.

2.1.2.1 Proceso de SCRUM

El proceso SCRUM parte de la generación de una lista clara de objetivos, que serán separados en iteraciones y entregas, y desarrollados de manera incremental. El proceso define dos etapas principales que son la

etapa inicial o Sprint 0, e iteraciones secciones o Sprints, como se muestra en la figura1.

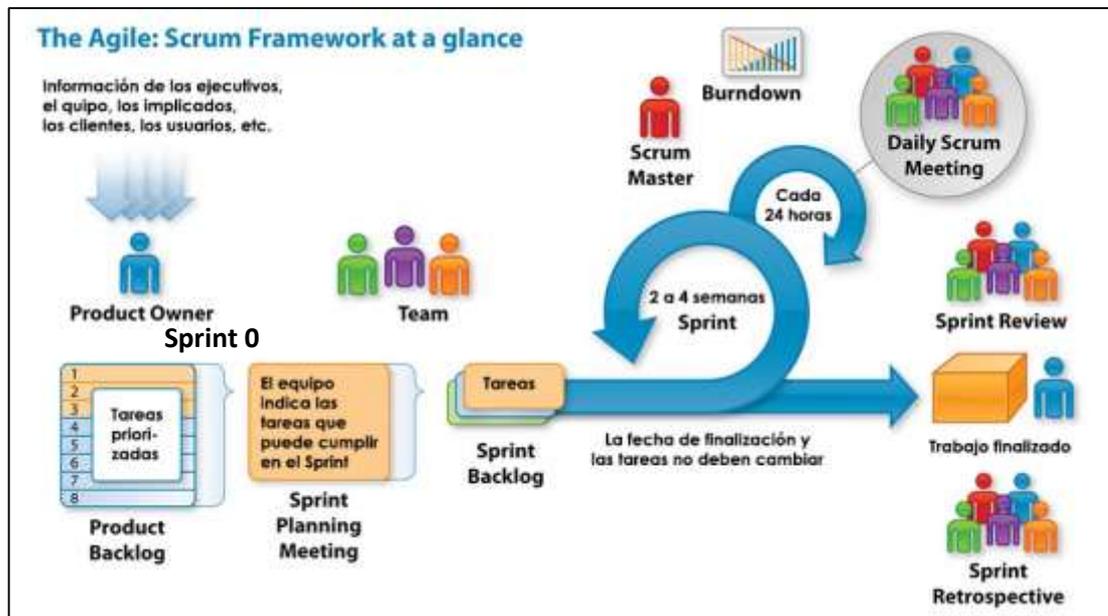


Figura 1 Ciclo de metodología SCRUM

Fuente: (Torras, 2015)

- Etapa inicial o Sprint 0, es de tiempo indefinido, determina las condiciones y viabilidad del proyecto, durante este período se realiza una lista de requerimientos del software y tareas prioritizadas llamándose de esta manera Product Backlog que posteriormente con las definiciones del usuario se irán especificando a través de las iteraciones siguientes.
- Iteraciones sucesivas o Sprints, esta etapa se divide en cuatro actividades las mismas que garantizan el cumplimiento de los objetivos del Sistema y conllevan a una secuencia de reuniones. En la Tabla 2 se especifica las actividades de los Sprints.

Tabla 2
Actividades de un Sprint

Actividad	Detalle
Planificación	En esta reunión se determina que acciones se llevarán a cabo para el desarrollo del Sprint, las mismas que son ordenadas, evaluadas y de acuerdo a este proceso son valoradas y definidas como el Product Backlog. Si el equipo se siente capaz de realizar más acciones en este Sprint se hará una consulta con el Product Owner para que estas sean añadidas caso contrario se decide qué acciones se eliminarán.
Standup Meeting	En esta reunión se permite compartir información con el equipo respecto al desarrollo del sistema, se determina los avances realizados, los planeados para el siguiente día, y si se detectaron inconvenientes para lograr el objetivo, con lo cual se llegan a ideas que permitan realizar acciones de manera productiva en el tiempo correspondiente, es así como se resuelven problemas a tiempo.
Reunión de Revisión	Se presenta el trabajo realizado y el que falta por terminar a los interesados, el equipo de desarrollo mostrará avances que puedan ser usados y probados para determinar si cumplen con los criterios de aceptación, los mismos que se encuentran definidos en el Sprint Planning.

CONTINÚA ⇒

Retrospectiva	Una vez finalizado el Sprint se lleva a cabo la Retrospectiva el cual permite que el equipo de desarrollo deje las observaciones del sprint recién terminado, con la finalidad de realizar una mejora continua en los siguientes.
----------------------	---

Fuente: Adaptado (Torras, Scrum.org, 2015)

2.1.2.2 Roles y Responsabilidades de SCRUM

El papel que desempeña cada persona durante el desarrollo del Sistema es fundamental, cada uno de ellos hace posible que el manejo de las actividades en esta metodología puede cumplirse y verse reflejado en la entrega del producto. En la tabla 3 se observa las responsabilidades que cumple este equipo.

Tabla 3

Roles de SCRUM

Rol	Descripción
SCRUM Master	Es la persona encargada en verificar que la metodología en el sistema se esté cumpliendo y sea entendida por todos los involucrados, trabaja con el Product Owner para la elaboración del Product Backlog. Interactúa directamente con el cliente, y durante el desarrollo soluciona y elimina los inconvenientes que detengan el mismo.
Product Owner	Es la persona que conoce el negocio del cliente y visualiza el producto requerido, se encarga de enlistar las ideas que tiene el cliente y las ordena de acuerdo a la prioridad para colocarlas en el Product Backlog, es por eso que mantiene una relación estrecha con el mismo y el equipo de desarrollo para verificar que se realice con las especificaciones correspondientes.

CONTINÚA ⇒

SCRUM Team	Este equipo de personas organiza y toman las decisiones para lograr los objetivos del Sprint y del producto. Su principal tarea es la de estimar esfuerzo posteriormente crean el Sprint Backlog mientras revisan la lista de tareas del Product Backlog para sugerir cambios o eliminar los posibles obstáculos que impidan cumplir con una de las mismas.
Customer	Es el cliente que está presente en las tareas que se definen en el Product Backlog.
Manager	Esta persona trabaja con el SCRUM Master y comprueba el progreso durante el desarrollo, finalmente toma decisiones y verifica el funcionamiento del sistema de acuerdo al Product Backlog.

2.1.2.3 Artefactos de SCRUM

Para un trabajo óptimo en esta metodología se trabaja con artefactos necesarios para el desarrollo del sistema, los cuales son utilizados por todas las personas involucradas en el mismo. A continuación, en la tabla 4, se detalla la lista de artefactos.

Tabla 4

Artefactos de SCRUM

Artefacto	Descripción
Product Backlog	Es la lista de requerimientos del cliente, la misma que será manejada y actualizada por el Product Owner se puede definir como un artefacto básico con el que se puede realizar un Sprint, y puede ser modificada a medida que se tiene una idea más clara del producto.
Sprint Backlog	Es la lista de tareas que el equipo de desarrollo realizará durante el tiempo que está definido un Sprint.
Product Increment	Constituye el incremento de cada objetivo en cada Sprint, y al finalizar cada uno de los mismos deben ser probados por el usuario y aceptados por el Product Owner.

2.1.2.4 Burndown Chart

Esta gráfica permite medir la velocidad y progreso del Sprint, de esta manera el equipo de trabajo puede aumentar o disminuir su ritmo de trabajo para poder cumplir con los objetivos en el tiempo estimado. En la figura 2 se determina lo siguiente: en el eje horizontal se encuentra la fecha inicial a la fecha final del Sprint y en el eje vertical se muestra la cantidad de requerimientos del mismo. Por lo tanto, la línea vertical indica el progreso del trabajo.

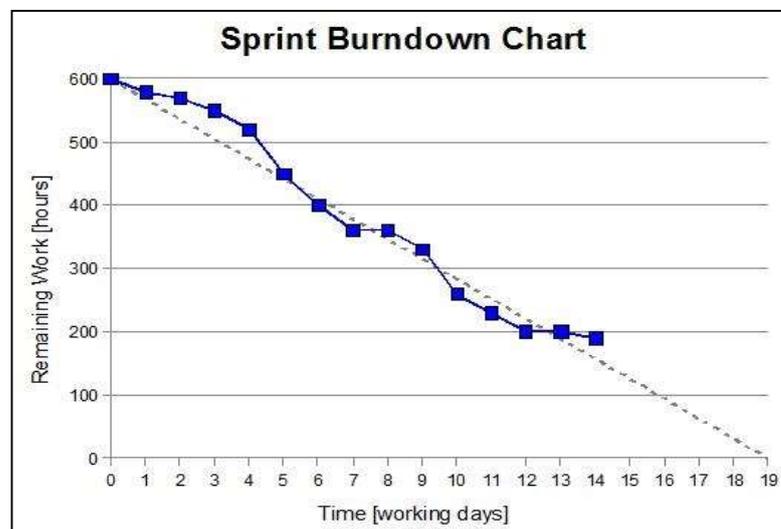


Figura 2 Diagrama Burndown Chart

Fuente: (Sprunck, s.f.)

2.1.3 Extreme Programming

La programación extrema o Extreme Programming (XP) es una metodología de desarrollo de la ingeniería de software formulada por Kent Beck, es el más destacado de los procesos ágiles de desarrollo de software. Al igual que éstos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad. (Bustamente, 2014)

Esta metodología es uno de muchos procesos ágiles que existen. Muchas compañías e industrias han tenido resultados positivos. El éxito de esta metodología ágil de desarrollo se basa en la satisfacción del cliente,

consiste en entregar el software que necesitas tal como lo necesitas y aumenta la productividad a la hora de desarrollar un sistema. Una de las principales características de esta metodología es que se permite realizar la corrección de los errores antes de realizar otra tarea.

2.1.3.1 Proceso y prácticas de XP

Este proceso consta de cuatro fases fundamentales que son: Planeación, Diseño, Desarrollo y Pruebas. Las prácticas determinan la manera en la que el equipo de desarrollo trabaja día a día. La adopción de una u otra práctica puede variar en función de contexto en el que se esté trabajando. (Álvarez García, de las Heras del Dedo, & Lasa Gómez, 2012). La tabla 5 especifica el desarrollo de estas actividades que guían el proceso de XP.

Tabla 5

Proceso y prácticas de Extreme Programming

Proceso	Descripción
Planeación	<p>Al ser una metodología de constante dialogo entre el equipo de desarrollo y los clientes, un proyecto XP empieza recopilando requerimientos, tomando en cuenta el grado de funcionalidad, para posteriormente asignarle un tiempo de desarrollo el mismo que será concordado con el cliente y los miembros del equipo XP. Durante esta actividad se toman en cuenta las siguientes prácticas:</p> <p>Historias de usuarios</p> <p>Estas historias de usuario sustituyen a varios documentos de especificaciones funcionales. Deben tener un detalle mínimo, el cuál permita a los programadores saber que se va a realizar o que se planea hacer, la dificultad de desarrollo de la misma y cuánto tiempo le tomará al programador ejecutarlo.</p>

CONTINÚA ⇒

Plan de iteraciones

Las historias de usuario tienen un gran aporte en la planificación de iteraciones, ya que debido a esto se procede a desarrollar el sistema, teniendo varios entregables funcionales. Al inicio de cada iteración, se efectúan las reuniones de planificación de iteración donde las historias de usuario se traducen a tareas de desarrollo.

Reuniones de Seguimiento

Habitualmente se ejecutan reuniones diarias de seguimiento, con el objetivo de mantener la comunicación entre el equipo de desarrollo y el cliente. En estas reuniones se comparten los problemas que se llegan a tener y las posibles soluciones a dichos inconvenientes.

Diseño

Durante el diseño se pretende realizarlo lo más sencillo posible, por lo general es una etapa de transición ya que puede modificarse antes o iniciada la codificación. En esta actividad se toma en cuenta lo siguiente:

Simplicidad

Todo diseño simple, siempre se implementa en un sistema de forma más rápida y sencilla que un diseño complejo. XP nos propone implementar el diseño más simple posible y que el mismo sea 100% operativo y funcional.

Recodificación

Esta práctica consiste en volver a escribir ciertas partes del código fuente en un programa. Todo esto sin cambiar la funcionalidad que tiene la misma, con el objetivo de convertirlo más simple y entendible (optimización del código).

CONTINÚA ⇒

Codificación Durante este proceso es necesario la presencia del usuario, esta debe ser programada bajo los estándares que ayuden a la comprensión del código, para ello es necesario que se realicen pruebas unitarias antes de iniciar con la codificación las cuales permitirán al desarrollador construir una solución más eficiente a esa tarea. Se recomienda tener en cuenta las siguientes prácticas:

Disponibilidad de cliente

Este es considerado uno de los requerimientos más importantes en XP; consiste en estar en constante contacto con el cliente. Así la mejora continua del software permitirá que englobe todas las necesidades del cliente, o en su mayoría. Se necesita del “cara a cara” con el cliente para poder definir completamente las historias de usuario.

Programación en pares

Otra de las propuestas de la metodología es la “programación en pares”, es decir, que los productos se desarrollen en pares. Esta práctica puede disminuir errores, y se pueden lograr mejores propuestas de diseño y sobre todo se compensa la inversión en horas para el sistema.

Integración continua

Una vez terminada una funcionalidad es importante probarlo y desplegarlo para continuar sin contratiempos.

CONTINÚA ⇒

Pruebas	<p>Las pruebas facilitan que el código sea revisado antes de ser implementado y alertar de forma oportuna los cambios que se deban realizar para el buen funcionamiento del sistema, se fundamentan en las historias de usuario especificadas en la fase de Planificación. Es recomendable seguir las siguientes prácticas:</p> <p>Detección y corrección de errores</p> <p>“Bug” conocidos a los errores de un sistema informático. Una vez encontrado estos “Bugs” deben ser corregidos inmediatamente teniendo las debidas precauciones de no dañar las funcionalidades que se encuentran ya operativas. Al finalizar las correcciones se vuelven a ejecutar las pruebas, avanzando en iteraciones de correcciones.</p> <p>Pruebas de aceptación</p> <p>Estas pruebas son creadas tomando en cuenta las historias de usuarios definidas por el cliente al inicio del proyecto. Conocidas como pruebas de “caja negra”. Y los responsables de ejecutar estas pruebas son los clientes.</p>
----------------	--

2.1.3.2 Roles y Responsabilidades

A continuación, en la tabla 6, se visualizan los roles XP aplicados en el sistema, según la propuesta original de Beck:

Tabla 6

Roles de Extreme Programming

Rol	Descripción
Cliente	El cliente es la persona más importante en el sistema. Él es quien define todas las especificaciones o requerimientos y realiza las pruebas funcionales que se deben efectuar en el sistema.

CONTINÚA ⇒

Programador	Es la persona encargada del desarrollo del sistema. Estos son responsables del código, diseño e integridad sobre el mismo. El programador es la pieza más importante del equipo.
Tester	El “tester” es quien prepara y realiza todo tipo de prueba del sistema, de igual manera da soporte y apoya al cliente verificando que el sistema esté acorde a las especificaciones dadas.
Consultor	Es una persona, ya sea interna o externa a la organización, que brinda un constante apoyo en el equipo XP con asuntos muy específicos. Estas personas pueden brindar su apoyo durante todo el proceso o por un tiempo definido.

2.2 SCRUM combinado con Extreme Programming

Tanto Scrum como XP son marcos de trabajo que se complementan muy bien en el desarrollo de un sistema, es así como Scrum organiza un equipo de personas dirigidas a cada una de sus tareas mientras que Extreme Programming construye un software de manera productiva y eficaz, utilizando menos recursos. (Álvarez García, de las Heras del Dedo, & Lasa Gómez, 2012). Gracias a estas metodologías ágiles podemos definir qué actividades, herramientas y artefactos se tomarán en cuenta para el desarrollo de este sistema.

2.2.1 Actividades, artefactos y herramientas tomadas para SCRUM

Las actividades a ser tomadas en cuenta para el desarrollo del sistema son las siguientes:

- Etapa inicial o Sprint 0: A través de esta etapa se elabora el Product Backlog con base a las ideas y requerimientos del cliente para posteriormente se determina la viabilidad del proyecto.

- Product Backlog: Lista de los requerimientos del cliente los cuales serán refinados por el Product Owner.
- Planificación del Sprint: Se define las tareas a realizar con el equipo de desarrollo para el Sprint.
- Sprint Backlog: Lista de requerimientos refinados que deberán cumplirse en el tiempo estimado del Sprint.
- Reunión de Scrum diario.
- Revisión del Sprint: El cliente evalúa el cumplimiento del trabajo de acuerdo a los requerimientos especificados.
- Burn down chart: Diagrama de seguimiento del Sprint.

2.2.2 Prácticas tomadas para Extreme Programming

Las prácticas a tomarse en cuenta para el desarrollo de este sistema son los siguientes:

- Pruebas antes del desarrollo: Se programa el sistema en base a las pruebas unitarias diseñadas para cada requerimiento.
- Reuniones de Seguimiento: Se realizan reuniones con el objetivo de mantener la comunicación entre el equipo de desarrollo y el cliente, y solucionar inconvenientes durante el desarrollo del sistema.
- Integración continua: Una vez terminada una funcionalidad es importante probarlo y desplegarlo.
- Pruebas de aceptación: Estas pruebas son creadas tomando en cuenta las historias de usuarios definidas por el cliente y los responsables de ejecutar las mismas.

2.3 Realidad Virtual

La realidad virtual comúnmente está generada por un conjunto de sistemas informáticos en la que tenemos la sensación de adentrarnos en un mundo que no son más que ilusiones. No obstante, también puede verse relacionado a este mundo virtual con la imaginación, los sueños, historias, libros, entre otros.

La simulación permite que se cree un mundo virtual a través de la inmersión sensorial creando lugares u objetos reales o no. El objetivo de esta tecnología es que una persona pueda sentirse protagonista de este ambiente y pueda interactuar de manera que hasta incluso se pueda obtener a través de sensores la voluntad implícita del usuario a través de sus movimientos.

2.3.1 Las Tres i's de la Realidad Virtual

2.3.1.1 Inmersión

El usuario se introduce en un ambiente en el cual pierde contacto con la realidad.

2.3.1.2 Interacción

El usuario interactúa directamente con dispositivos de entrada como un monitor, de manera que recibe respuestas a través de sus sentidos.

2.3.1.3 Imaginación

A través del mundo virtual se llegan a crear realidades que en la vida real son difíciles de construir.

En la figura 3, se muestra la relación de las tres i's con respecto a la realidad aumentada.

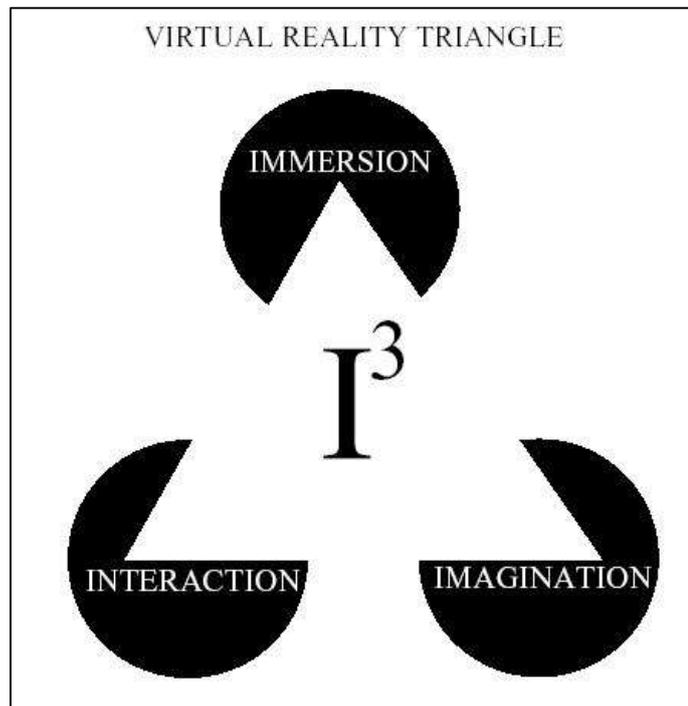


Figura 3 El triángulo de la realidad virtual

Fuente: (Realidad Virtual, s.f.)

2.2.2 Características de las Realidad Virtual

- Genera un entorno diferente al del mundo físico.
- La resolución de imágenes con las que se presenta la realidad virtual es de resolución alta.
- Permite que se interactúe de forma dinámica.
- Trabaja en tiempo real.
- Reacciona a estímulos de las personas, a través de sensores.
- Muestran amplios ángulos de visión tanto vertical como horizontal.
- Se muestra en gráficos tridimensionales.

2.4 Realidad Aumentada

La realidad aumentada es una combinación de la parte real con la realidad virtual, con esta última son totalmente diferentes ya que esta se centra únicamente en lo virtual y se aísla de lo real.

La idea de realidad aumentada, en concreto, se refiere a la inclusión, en tiempo real, de elementos virtuales dentro del universo físico. Utilizando unas gafas u otros dispositivos como cámaras. (Realidad Aumentada, s.f.)

Gracias a esta tecnología al usuario se le hace más fácil interactuar con elementos que quizá antes era imposible de manejar. La realidad aumentada actualmente está ganando espacio en el mercado por la gran cantidad de aplicaciones y usos que se le dan como es en el campo de la medicina, videojuegos, catálogos en 3D, etc.

2.4.1 Características de la Realidad Aumentada

La realidad aumentada cumple con tres características importantes:

- Combina el mundo real con el mundo virtual
- Interacción en tiempo real
- Visualizado en 3D

2.4.2 Tipos de Realidad Aumentada

Los tipos de Realidad Aumentada para integrar información virtual dentro de una escena real son esencialmente dos mencionadas a continuación:

2.4.2.1 Realidad Aumentada Basada en Marcadores e Imágenes

Este tipo de realidad aumentada maneja símbolos impresos en papel o imágenes en los que se superpone algún tipo de información, que por lo general serán imágenes en 3D, videos o texto.

Estos marcadores generalmente son de color negro con blanco conocidos como códigos QR o puede ser una simple imagen, y para acceder a la información virtual que este contiene no es necesario una conexión a Internet, sino únicamente el software apropiado que pueda hacer la lectura del mismo. En la figura 4 se indica un ejemplo de marcador utilizando código QR.

El software que captura dicho marcador será capaz de realizar un seguimiento del mismo de tal manera que si el usuario lo mueve, el objeto 3D superpuesto también sigue ese movimiento, si se gira el marcador se puede observar el objeto 3D desde diferentes ángulos y si se acerca o se aleja, el tamaño del objeto aumenta o se reduce respectivamente. (Reinoso, y otros, 2011)



Figura 4 Ejemplo de Marcador

Fuente: (Reinoso, y otros, 2011)

2.4.2.2 Realidad Aumentada Basada en Posición

Este tipo de realidad aumentada comienza a desarrollarse desde el año 2009, tomando el nombre de navegadores de realidad aumentada, se trabaja necesariamente con dispositivos que tengan GPS, brújula y acelerómetro permitiendo de esta manera que este por medio de la cámara pueda capturar la posición exacta de un lugar, tanto de altura y dirección en la que apunta.

Para extraer la información y mostrarla al dispositivo a través de la aplicación, es necesario que el dispositivo esté conectado a Internet. En la figura 5 se muestra un ejemplo sobre realidad aumentada basada en posición.



Figura 5 Ejemplo de realidad aumentada en posición

Fuente: (Reinoso, y otros, 2011)

2.4.3 Elementos de la Realidad Aumentada

Los elementos que conforman el sistema de realidad aumentada son los siguientes:

- **Cámara:** Elemento que captura las imágenes del mundo real, la misma que será transmitida al procesador de sistema de realidad aumentada.
- **Procesador:** Elemento que interpreta la información recibida por la cámara. Integra el mundo real y el virtual.
- **Marcador:** Son símbolos que un software debe identificar e interpretar y de acuerdo a este dar una respuesta a través de la pantalla ya sea generalmente una imagen u objeto 3D.
- **Elemento Activador:** Conocido también como el programa que hace uso de elementos como el GPS, acelerómetro y brújula para el funcionamiento del mismo y mostrar resultados a través del dispositivo móvil.

2.5 Framework de Desarrollo

Para el desarrollo de la aplicación se hace uso de una plataforma de desarrollo que cumpla con características que permitan un buen manejo del

mismo como tener un buen grado de madurez, pueda integrarse con otros sistemas y poseer escalabilidad.

2.5.1 Eclipse

Es una plataforma de software que posee herramientas de programación, desarrollado originalmente por IBM y ahora por la Fundación Eclipse, es multiplataforma y de código abierto basado en Java. Existe gran cantidad de complementos que lo hacen un marco de desarrollo unificado e integrado para las personas que hacen uso del mismo.

Entre las características de esta plataforma se destacan las siguientes: dispone de pruebas unitarias con JUnit, permite refactorización y maneja un control de versiones con Concurrent Versions System o CVS, el cual mantiene el registro de todo el trabajo y los cambios realizados en un proyecto.

2.5.2 Unity 3D

Unity es una de las herramientas más completas de desarrollo de Videojuegos que existe en el mercado, diseñado para diferentes plataformas entre las cuales los grupos principales están: móviles, consola, escritorio y navegadores.

El motor de Unity es adorado en la industria de los juegos por la profundidad y calidad de sus optimizaciones y la velocidad y eficacia de sus flujos de trabajo, lo cual permite a los usuarios de Unity producir contenido de alta gama en forma rápida. (Editor de Unity, 2016)

2.5.3 Entorno de Unity 3D

El editor de Unity 3D es uno de los más sencillos y potentes del mercado. Se divide en 5 vistas principales en el cual se detallan en la tabla 7.

Tabla 7

Vistas de Unity 3D

Vista	Descripción
Explorador	Lista todos los elementos (o activos) de un proyecto. Permite ordenar de forma sencilla tu aplicación. En esta vista se encuentran tus imágenes, escenas, scripts, audios, prefabs, texturas, atlas y todos los elementos que usarás o podrás usar en tu juego o aplicación.
Inspector	Muestra y define las propiedades de los elementos de un proyecto. Modifica valores de forma rápida, cambia texturas arrastrando ficheros desde el explorador.
Jerarquía	Lista jerárquica de los elementos de tu escena.
Escena	Diseño y maqueta de tu juego completo o una pantalla o sección de esta, cada escena representa un nivel o sección diferente del juego.
Juego	Visualiza tu juego a distintas resoluciones.

Fuente: Adaptado (Mocholi, 2014)

2.5.4 Visual Studio

Visual Studio es un entorno de desarrollo integrado (IDE) con todas las características para Android, iOS, Windows, la web y la nube. Visual permite escribir código de manera precisa y eficiente sin perder el contexto del archivo actual. Puede acercarse fácilmente a los detalles, como la estructura de llamadas, las funciones relacionadas, las inserciones en el repositorio y el estado de las pruebas. También permite refactorizar, identificar y corregir problemas de código. Las herramientas de pruebas de Visual Studio ayudan a proporcionar software de alta calidad. (Microsoft, 2017)

2.6 Tecnologías de Desarrollo

2.6.1 Glassfish

Es un servidor de aplicaciones multiplataforma adquirida por Oracle, implementa la tecnología Java Enterprise Edition, se lo puede utilizar libremente por lo cual siempre existen contribuciones al código, utiliza un componente adicional Grizzly para velocidad y escalabilidad. Gracias a las mejoras de la consola de administración permiten que su uso sea fácil de manejar, entre ellos está el soporte para REST. En una nueva versión de Glassfish permite dar a los usuarios soporte comercial, la cual está dirigida a pequeños negocios, universidades, empresas, entre otros, para interés personal o investigación.

2.6.2 MySql

Es un gestor de base de datos relacional, desarrollado por Oracle Corporation, posee altos niveles de seguridad, escalabilidad, fiabilidad, y herramientas de soporte técnico, lo cual hace que aumente la agilidad de la empresa o negocio mientras reducen recursos y gracias a esto es considerado como una base de datos potente en el mundo enfocada un poco más al desarrollo web, por la baja concurrencia en el manejo de los datos.

Gracias a la seguridad que el gestor posee permite que las contraseñas estén encriptadas y de esta manera asegurar la conexión con un servidor. La escalabilidad que MySql brinda es grande ya que existen bases de datos que contienen hasta 50 millones de registros, y al hablar de las funciones SQL aquí una de las principales características es que estas usan librerías para que sean lo más rápidas posibles.

2.6.3. Vuforia

Es un SDK creada por Qualcomm puesta a disposición para quienes desarrollan aplicaciones móviles con realidad aumentada, entre las características que lo hacen interesante es que permite el reconocimiento de texto, imágenes, detección rápida de targets, y la fijación de los mismos no se perderá aun cuando este se mueva.

Entre las plataformas con las que vuforia es compatible con Android, iOS y con el framework de desarrollo Unity, está disponible para los sistemas operativos más conocidos como Windows, Linux y Mac.

2.6.3.1 Arquitectura de Vuforia

La arquitectura de Vuforia está compuesta de los elementos que se detallan en la tabla 8.

Tabla 8

Arquitectura de Vuforia

Elemento	Descripción
Cámara	La cámara asegura que la imagen sea captada y procesada por el Tracker.
Base de datos	La base de datos del dispositivo es creada utilizando el Target Manage; ya sea la base de datos local o la base de datos en la nube, almacena una colección de Targets para ser reconocidos por el Tracker.
Target	<p>Son utilizadas por el rastreador (Tracker) para reconocer un objeto del mundo real; los Targets pueden ser de diferentes tipos:</p> <p>Image Targets: Imágenes; tales como: fotos, páginas de revistas, cubierta de libros, poster, etc.</p> <p>Word Targets: Elementos textuales que representen palabras simples o compuestas: Libros, revistas, etc. Sus modos de reconocimiento son: por palabra o caracteres.</p>

CONTINÚA ⇒

Tracker	Analiza la imagen de la cámara y detecta objetos del mundo real a través de los frame de la cámara con el fin de encontrar coincidencias en la base de datos.
----------------	---

Fuente: Adaptado (Cruz, 2014)

2.6.4 Java 1.8

Es un lenguaje de programación, el cual inicialmente tomó el nombre de Oak denominado así por un roble que se encontraba afuera de la oficina de las personas que lo crearon.

Actualmente el nombre de Java aboga varios significados ya sea por el acrónimo Just Another Vague Acronym o podría tratarse de las iniciales de sus diseñadores: *James Gosling, Arthur Van Hoff, y Andy Bechtolsheim*. La hipótesis que más fuerza tiene es la de que Java debe su nombre a un tipo de café disponible en la cafetería cercana, de ahí que el icono de java sea una taza de café caliente. (Wikipedia, 2017).

El lenguaje Java se creó con cinco objetivos principales:

- Debería usar el paradigma de la programación orientada a objetos.
- Debería permitir la ejecución de un mismo programa en múltiples sistemas operativos.
- Debería incluir por defecto soporte para trabajo en red.
- Debería diseñarse para ejecutar código en sistemas remotos de forma segura.
- Debería ser fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos, como C++. (Wikipedia, 2017)

2.6.4.1 Plataforma Java Enterprise Edition (JEE)

La plataforma Java Enterprise Edition o Java EE, por sus siglas en inglés, es un conjunto de componentes conocidas como API's, cuyo objetivo principal es simplificar el desarrollo de software empresarial y fue diseñada para soportar aplicaciones que implementan servicios empresariales para consumidores, empleados, proveedores y socios estratégicos de la empresa.

Este tipo de aplicaciones generalmente son complejas, acceden a varias fuentes de información y son consumidas por varios tipos clientes. (Oracle, 2014).

2.6.4.2 Servidor Java EE

Un servidor Java EE, comúnmente conocido como servidor de aplicaciones, implementa las API's que forman de la plataforma Java EE y brindan el acceso a la información almacenada. Estos servicios son expuestos en forma de contenedores, que son la interface entre los componentes de la aplicación y las funcionalidades de bajo nivel. La funcionalidad del contenedor está determinada por los diferentes tipos de componentes soportados entre los que se destacan los contenedores Web, EJB, RESTful Web Services (Oracle, 2014). En la tabla 9 se detallan los mismos.

Tabla 9

Contenedores servidor Java EE

Contenedores	Descripción
Contenedor Web	El contenedor Web es la interfaz entre los componentes Web y el servidor Web. Gestionan el ciclo de vida de las páginas, manejan los pedidos y son el nexo con el contexto de datos. Estos componentes pueden ser páginas JavaServer Faces (Oracle, 2014)
Contenedor EJB	Especifica un entorno de tiempo de ejecución de un Enterprise Bean que incluye servicios de seguridad, concurrencia, administración del ciclo de vida, transacciones, implementación, asignación de nombres y otros servicios. (Contenedor EJB, 2012)

CONTINÚA ⇒

Servicio RESTful	Web	Los Servicios Web de tipo REST o Transferencia de Estado Representacional, son un conjunto de principios arquitectónicos enfocados en los recursos que pueden ser abordados y transferidos sobre el protocolo de comunicaciones http (IBM, 2008)
-------------------------	------------	--

2.6.5 C#

Microsoft C# es un nuevo lenguaje de programación diseñado para crear un amplio número de aplicaciones empresariales que se ejecutan en .NET Framework. Supone una evolución de Microsoft C y Microsoft C++; es sencillo, moderno, proporciona seguridad de tipos y está orientado a objetos. Estos servicios incluyen interoperabilidad entre lenguajes, recolección de elementos no utilizados, mejora de la seguridad y mayor compatibilidad entre versiones. (Microsoft, 2017). Este lenguaje permite desarrollar diversas aplicaciones de manera práctica manteniendo la expresividad.

CAPÍTULO III

ESPECIFICACIÓN DE REQUERIMIENTOS

3.1 Introducción

La Ingeniería de Requerimientos es un amplio conjunto de tareas relacionadas que conforman las necesidades de un cliente, para posteriormente solucionarlos en un software ya sea uno existente o nuevo. El objetivo de este documento es que todos estos requisitos adquieran un estado óptimo hasta llegar a la fase de diseño del sistema.

A continuación, se elaborará una Especificación de Requerimientos de Software (ERS) para el sistema planteado el cual consta de una aplicación móvil multiplataforma utilizando realidad aumentada para la administración de la carga horaria en los Laboratorios Generales de Computación de la Universidad de las Fuerzas Armadas – ESPE MATRIZ.

3.2 Propósito

El propósito del documento es definir los requerimientos funcionales y no funcionales que debe tener el sistema, la misma que está dirigida a los usuarios o las personas que deseen conocer el funcionamiento del mismo. Esta especificación se basará en el estándar del Instituto de Ingenieros Eléctricos y Electrónicos (IEEE por sus siglas en Inglés), Practicas Recomendadas para los Requerimientos de Software IEEE/ANSI 830-1998 (IEEE, 2014).

3.3 Alcance

El presente Proyecto de Investigación denominado “Smart Labs”, está destinada a realizar lo siguiente:

- La aplicación móvil debe cubrir las necesidades de: Obtener información sobre el Docente que se encuentra en cada uno de los

Laboratorios en el que constará; Foto, Nombres y Apellidos del Docente, el nombre de la Materia que dicta, y el NRC.

- Se tomará en cuenta los laboratorios que se encuentren disponibles para el uso de cualquier Docente de la Universidad.

La aplicación comprenderá de los siguientes módulos:

- **Módulo de Servicios**

Este módulo constará de una capa de servicios REST expuestos en un servidor de Aplicaciones Glassfish, el cual permitirá interactuar con la aplicación móvil y enviar la información deseada por el usuario final.

- **Módulo Consultas por Imagen (3D - R.A.)**

Es el encargado de obtener información y mostrarla en forma de un mapa de ruta en la aplicación móvil; esto consiste en capturar la imagen y posteriormente enlazarla con la información respectiva.

- **Módulo Consultas por Búsqueda**

Es el encargado de obtener información y mostrarla en forma de una lista plana en la aplicación móvil; esto consiste en realizar una búsqueda simple seleccionando los filtros deseados y posteriormente visualizar el resultado en modo de una lista de información.

- **Administrador Web**

Este módulo será manejado por una persona encargada Laboratorista y permitirá presentar una lista de docentes y laboratorios, los cuáles a través de este podrán ser asignados en el horario correspondiente, permitiendo la interoperabilidad entre la aplicación y el usuario.

3.4 Limitaciones del Proyecto

El Proyecto de Investigación implementará las funciones principales de una aplicación completa de este tipo, por lo que es importante fijar limitaciones los cuales se describen a continuación:

- La aplicación web y móvil no indicará la cantidad de recursos que se encuentren en cada Laboratorio de Computación.

- La aplicación web y móvil no actualizará la información que se obtenga del Banner manejado por la Universidad de las Fuerzas Armadas – ESPE MATRIZ.
- La aplicación web y móvil no realizará notificaciones.
- La aplicación web y móvil no consultará información de terceros.
- El Administrador Web no generará reportes.

3.5 Personal Involucrado

El equipo de personas que se encuentran involucradas en este sistema está compuesto por profesionales que poseen conocimientos en el mismo campo, los cuales serán de ayuda para el desarrollo e implementación del mismo en la Universidad de las Fuerzas Armadas – ESPE MATRIZ. La tabla 10 describe los roles que cumplen cada uno de los miembros del equipo de trabajo.

Tabla 10

Personal involucrado

Nombre	Campo Profesional	Rol	Información de Contacto
César Villacís	Ingeniero en Sistemas	Director del proyecto	cvillacis@espe.edu.ec
Lorena Duque	Ingeniera en Sistemas	Analista de Base de Datos de las UTIC	lquique@espe.edu.ec
Linda Manosalvas	Ingeniera en Sistemas	Jefe de Laboratorios	lmanosalvas@espe.edu.ec
Pamela Utreras	Egresada de la Carrera de Ingeniería en Sistemas	Analista, diseñador y programador	peutreras@espe.edu.ec

3.6 Definiciones

- Aplicación móvil: Es una primera versión de la aplicación que será implementada en los Laboratorios de Computación y puede ser instalada en un dispositivo móvil.
- Administrador Web: Son sitios de Internet, que muestra información específica, y trabaja directamente con un diseñador o programador.
- Servicio Web: Es una tecnología que utiliza protocolos y permiten el intercambio de datos con otro servicio web.
- Base de Datos: Conjunto de datos que se encuentran relacionados entre sí.
- Sistema Gestor de Bases de Datos: Software que gestiona un conjunto de base de datos.

3.7 Acrónimos

- IR: Identificación de requerimiento.
- RE: Requerimiento específico.
- NR: Nombre del requerimiento.
- ERS: Especificación de Requerimientos de Software.
- JNDI: Interfaz de Nombrado y Directorio Java (Java Naming and Directory Interface) por sus siglas en inglés, permite a las aplicaciones guardar datos que se encuentran identificados por un nombre.
- RESTful: Representational State Transfer por sus siglas en inglés, (Transferencia de Estado Representacional) es un estilo arquitectónico que permite ser simples, rápidos y ligeros en el manejo de los datos.
- IEEE: Instituto de Ingenieros Eléctricos y Electrónicos.

3.8 Referencias

IEEE-STD-830. (1998). Especificación de Requerimientos de Software. (IEEE, 2014)

Pressman, R. (2010). Ingeniería del software: Un enfoque práctico. México: Interamericana Editores S.A. McGraw-Hill

3.9 Visión General del Documento

Esta Especificación de Requerimientos (ERS) está dividida en tres secciones. En la primera se describe la introducción a la Especificación de Requerimientos de Software y una visión del sistema a desarrollarse. En la segunda sección del ERS describe de manera general, las funciones de la aplicación móvil y el Administrador Web, las características de los distintos tipos de usuarios, restricciones, suposiciones y dependencias que afectan al desarrollo. Por último, se detalla los requerimientos funcionales y no funcionales del sistema.

3.10 Descripción General

En esta sección de la especificación de requerimientos, se hará una descripción general de los factores que afectan al producto y sus requisitos que pretenden ser más entendibles y brindar una solución a la problemática del presente proyecto.

3.10.1 Perspectiva del Producto

La aplicación móvil permitirá mostrar a los usuarios datos relevantes de la carga horaria en los Laboratorios de Computación, así como la foto, nombres, apellidos, email del Docente, asignatura y NRC de la misma, los cuales serán desplegados en tiempo real, ya sea en texto plano o en realidad aumentada. Adicionalmente habrá un Administrador Web el cual permite a los Laboratoritos asignar la materia y docente a un Laboratorio y servirá de apoyo para la carga de información en el aplicativo móvil.

3.10.2 Funciones del Producto

La aplicación móvil y el administrador Web serán capaces de realizar las siguientes funciones:

- Aplicación Móvil
 - Visualizar Docentes
 - Visualizar Asignaturas
 - Visualizar Laboratorios
 - Capturar imágenes en Realidad Aumentada
- Administrador Web
 - Iniciar sesión
 - Visualizar Docentes
 - Visualizar Asignaturas
 - Visualizar Laboratorios
 - Asociar Docente y Laboratorio
 - Finalizar Período Académico

3.10.3 Características de los Usuarios

La aplicación móvil es manejada por usuarios los mismos que pueden ser: público en general, estudiantes y docentes. Estos usuarios podrán hacer uso del mismo para poder visualizar las funciones especificadas como usuarios, como se especifica en la tabla 11. El administrador Web es manejado por los Señores Laboratoristas los mismos que podrán asignar los Docentes con los Laboratorios de acuerdo a la materia que cada uno de ellos dicta a sus estudiantes durante cada Período Académico.

Tabla 11

Características de los usuarios

Usuario	Descripción	Capacidad Técnica
Usuario	Podrá consultar la información que proporcione la aplicación móvil.	Debe estar familiarizado con el manejo básico de un Smartphone y aplicativos móviles.
Laboratoristas	Podrá registrarse con un usuario y contraseña al administrador web para asignar los diferentes Laboratorios de Computación a los Docentes que pertenecen a los Laboratorios del Departamento de Ciencias de la Computación.	Deberá tener formación especializada y estar familiarizado previamente con el administrador Web.

3.10.4 Restricciones

Las restricciones que presenta el desarrollador del sistema se especifican en la tabla 12.

Tabla 12

Restricciones del desarrollador

Restricciones	Descripción
Limitaciones del Hardware	El servidor es proporcionado por la Universidad de las Fuerzas Armadas – ESPE, por lo que no se tiene un control completo sobre la arquitectura en la se despliega la aplicación Smart Labs.
Lenguajes de programación	Se utiliza un Framework de desarrollo más actual “Spring Framework” el cual no tiene una buena compatibilidad

con el servidor de aplicaciones Glassfish proporcionado por la Institución.

CONTINÚA ⇒

Consideraciones acerca de la seguridad. Con respecto al administrador Web, a través de un inicio de sesión podrá gestionar la carga horaria que será visualizada en la aplicación móvil. El sistema se conecta a la base de datos por medio de JNDI, motivo por el cual el mismo no posee ningún tipo de información con respecto a la base.

La aplicación móvil no interactúa directamente con la base de datos, ya que obtiene información de servicios web RESTful.

3.10.5 Suposiciones y Dependencias

El administrador Web debe operarse sobre un sistema operativo CentOS, la base de datos se trabaja con MySQL Server Open Source Edition 4.1 y será orientado a la Web y servicios RESTful.

3.11 Requisitos Específicos

La tabla 13 recoge los requisitos específicos que permitirán planificar, diseñar, desarrollar y probar el Administrador Web, de igual manera en la tabla 14 para la aplicación móvil.

Tabla 13

Requisitos específicos del Administrador Web

RE	NR	Descripción
RE01	Iniciar sesión	El Laboratorista deberá identificarse para acceder al Administrador Web que gestiona la carga horaria de los Laboratorios.
RE02	Visualizar docentes	El Laboratorista puede visualizar la lista de los docentes que se encuentran activos para el

Período actual dictando asignaturas que pertenecen al Departamento de Ciencias de la Computación.

CONTINÚA ⇒

RE03	Visualizar Asignaturas	El Laboratorista puede visualizar la lista de las Asignaturas que dicta cada Docente del Departamento.
RE04	Asignar asignatura al Laboratorio	El Laboratorista se encarga de asignar un Laboratorio para cada asignatura verificando la disponibilidad del mismo.
RE05	Finalizar período académico	Permite limpiar la información creada para la carga horaria del Período finalizado

Tabla 14

Requisitos específicos de la Aplicación Móvil

RE	NR	Descripción
RE06	Visualizar Docente	El usuario podrá visualizar a través de la aplicación los detalles de la información de un Docente perteneciente al Departamento de Ciencias de la Computación.
RE07	Visualizar Asignatura	El usuario podrá visualizar a través de la aplicación los detalles de la información de la asignatura seleccionada de su interés.
RE08	Visualizar Laboratorio	El usuario podrá indagar a través de la aplicación la información requerida sobre el Laboratorio de su interés.
RE9	Capturar Imagen	El usuario capta la imagen del lugar que desea obtener información sobre los Laboratorios de

Computación y visualizarla a través de realidad aumentada.

3.11.1 Interfaces Externas

Se describen los requisitos que afectan a las interfaces de usuario, hardware, software y comunicaciones.

Interfaz con el usuario

La interfaz para el usuario en el Administrador Web será sencilla y construida de tal manera que sea intuitivo y fácil de manejarlo y navegar a través del mismo para obtener la información que el Laboratorista necesite o ya sea para asignar la carga horaria correspondiente a cada Laboratorio. En la aplicación móvil es predictiva y está diseñada para quienes tengan un conocimiento básico en manejo de un Smartphone. Presentará colores que definan al Departamento de Ciencias de la Computación.

Interfaz de Hardware

La aplicación Web se implementará en un servidor que cumpla con las características necesarias para la ejecución del servidor de aplicaciones y el sistema de base de datos. Para la ejecución de la aplicación móvil, el celular debe contar con permisos para acceder a la cámara y un sistema operativo actual

Interfaz de Software

En el presente proyecto utilizará CentOS como sistema operativo base para el despliegue de los servicios Web construidos y el acceso al administrador Web podrá ser posible mediante el uso de un navegador de Internet que preferencialmente puede ser Google Chrome o Mozilla Firefox.

Interfaz de Comunicación

La comunicación entre los usuarios y el administrador Web será implementada en base al protocolo de comunicación de datos TCP/IP.

3.11.2 Funciones

Requerimiento funcional del Administrador Web Inicio de sesión

Id. Requerimiento	RE01
Descripción	El Laboratorista deberá identificarse para acceder al Administrador Web que gestiona la carga horaria de los Laboratorios.
Parámetros de entrada	Usuario y contraseña
Resultado	Acceso a la interfaz del administrador Web
Proceso	<ol style="list-style-type: none"> 1. Ingresar a la URL: 2. Ingresar el usuario y contraseña correspondiente en cada campo. 3. Clic en el botón Ingresar. 4. Esperar el resultado.
Precondiciones	Acceder a la pantalla de inicio del administrador Web
Postcondiciones	Ninguna
Efectos secundarios	Ninguna
Prioridad	Alta
Persona encargada	Laboratorista

Requerimiento funcional del Administrador Web Visualizar docentes

Id. Requerimiento	RE02
Descripción	El Laboratorista puede visualizar la lista de los docentes que se encuentran activos para el Período actual dictando asignaturas que pertenecen al Departamento de Ciencias de la Computación.
Parámetros de entrada	Ninguna
Resultado	Lista de los Docentes activos cargados en Tablas

Proceso	Clic en la opción del Menú “Docentes”.
Precondiciones	Acceso al administrador Web
Postcondiciones	Ninguna
Efectos secundarios	Ninguna
Prioridad	Media
Persona encargada	Laboratorista

Requerimiento funcional del Administrador Web Visualizar Asignaturas

Id. Requerimiento	RE04
Descripción	El Laboratorista puede visualizar la lista de las Asignaturas que dicta cada Docente del Departamento.
Parámetros de entrada	Ninguna
Resultado	Lista de asignaturas dictas en el Departamento de Ciencias de la Computación.
Proceso	Clic en la opción del Menú “Asignaturas”.
Precondiciones	Acceso al Administrador Web
Postcondiciones	Ninguna
Efectos secundarios	Ninguna
Prioridad	Media
Persona encargada	Laboratorista

Requerimiento funcional del Administrador Web Asignar asignatura al Laboratorio

Id. Requerimiento	RE05
Descripción	El Laboratorista se encarga de asignar un Laboratorio para cada asignatura verificando la disponibilidad del mismo.
Parámetros de entrada	Datos del Laboratorio: bloque, piso, laboratorio, asignatura seleccionada.
Resultado	Mensaje de selección exitosa.

Proceso	<ol style="list-style-type: none"> 1. Ingresar a la pantalla correspondiente. 2. Escoger el bloque, piso, laboratorio. 3. Determinar la asignatura que ocupara dicho laboratorio en la hora correspondiente.
Precondiciones	Asignar a la pantalla del Laboratorio al que asignará la asignatura.
Postcondiciones	Se actualizará el horario.
Efectos secundarios	El usuario de la aplicación móvil no visualizará la carga horaria.
Prioridad	Alta
Persona encargada	Laboratorista

Requerimiento funcional del Administrador Web Finalizar Período Académico

Id. Requerimiento	RE06
Descripción	Permite limpiar la información creada para la carga horaria del período finalizado
Parámetros de entrada	Ninguno
Resultado	Mensaje de cierre exitoso del período finalizado.
Proceso	Clic en el botón Finalizar Período Académico
Precondiciones	Acceder a la pantalla de Asignación de Laboratorios.
Postcondiciones	Ninguna
Efectos secundarios	Se eliminan las asociaciones de las asignaturas con los Laboratorios.
Prioridad	Alta
Persona encargada	Laboratorista

Requerimiento funcional de la aplicación móvil Visualizar Docente

Id. Requerimiento	RE07
Descripción	El usuario podrá visualizar a través de la aplicación los detalles de la información de un Docente

	perteneciente al Departamento de Ciencias de la Computación.
Parámetros de entrada	Ninguna
Resultado	Lista de los Docentes activos cargados en Tablas
Proceso	Clic en la opción del Menú "Docentes".
Precondiciones	Acceso a la aplicación móvil Smart Labs
Postcondiciones	Ninguna
Efectos secundarios	Ninguna
Prioridad	Media
Persona encargada	Usuarios

Requerimiento funcional de la aplicación móvil Visualizar Asignatura

Id. Requerimiento	RE08
Descripción	El usuario podrá visualizar a través de la aplicación los detalles de la información de la asignatura seleccionada de su interés.
Parámetros de entrada	Ninguno
Resultado	Lista de asignaturas dictas en el Departamento de Ciencias de la Computación.
Proceso	Clic en la opción del Menú “Asignaturas”.
Precondiciones	Acceso a la aplicación móvil Smart Labs
Postcondiciones	Ninguna
Efectos secundarios	Ninguna
Prioridad	Media
Persona encargada	Usuarios

Requerimiento funcional de la aplicación móvil Visualizar Laboratorios

Id. Requerimiento	RE09
Descripción	El usuario podrá indagar a través de la aplicación la información requerida sobre el Laboratorio de su interés.
Parámetros de entrada	Ninguno
Resultado	Lista de los Laboratorios con la información de la carga horaria.
Proceso	Clic en la opción del Menú “Laboratorios”.
Precondiciones	Acceso a la aplicación móvil Smart Labs
Postcondiciones	Ninguna
Efectos secundarios	Ninguna
Prioridad	Media
Persona encargada	Usuarios

Requerimiento funcional de la aplicación móvil Capturar Imagen

Id. Requerimiento	RE10
Descripción	El usuario capta la imagen del lugar que desea obtener información sobre los Laboratorios de Computación y visualizarla a través de realidad aumentada.
Parámetros de entrada	Imagen capturada por la cámara
Resultado	Información de la carga horaria desplegada de acuerdo a la selección del Target.
Proceso	<ol style="list-style-type: none"> 1. Ingresar a la aplicación móvil Smart Labs 2. Clic sobre el botón de la cámara.
Precondiciones	Acceso a la aplicación móvil Smart Labs
Postcondiciones	Ninguna
Efectos secundarios	Ninguna
Prioridad	Media
Persona encargada	Usuarios

3.11.3 Requisitos No Funcionales

Requisitos de Rendimiento

Tanto el Administrador Web como la aplicación móvil garantizarán a los usuarios un manejo adecuado de cada uno de ellos, permitiendo dar solución a cada uno de los requisitos planteados inicialmente. Ambos trabajarán conjuntamente con la Base de datos mismo que trabajará de forma óptima para dar la información solicitada de forma rápida. Gracias a los recursos tecnológicos utilizados durante el sistema, la aplicación tendrá gran impacto y cumplirá con las expectativas previstas.

Seguridad

Únicamente los señores Laboratoristas podrán hacer uso del Administrador Web, ya que es su principal función una vez que ha iniciado un

Período Académico y este está orientado a la gestión de los Laboratorios, la interfaz es intuitiva y tiene un control de inicio de sesión para el manejo del mismo, de esta manera se garantiza la seguridad de la información manipulada de la carga horaria en cada uno de los Laboratorios de Computación.

La aplicación móvil será manejada por los usuarios (público en general, estudiantes, docentes) que dispongan de un Smartphone y requieran información de forma oportuna y en tiempo real sobre las materias o Docentes que se encuentran en los Laboratorios de Computación. La aplicación responderá a las peticiones del usuario y este dará respuestas a la misma de forma precisa evitando fugas de información que puedan perjudicar a terceros.

Fiabilidad

Todas las funciones que realiza tanto el Administrador Web como la aplicación móvil operarán de manera confiable, y se garantiza la integridad de los datos en el manejo de los mismos.

Disponibilidad

El administrador Web está diseñado y disponible únicamente para los señores Laboratoristas de los Laboratorios de Computación. La aplicación móvil está disponible para el público en general, estudiantes y docentes que dispongan de un Smartphone y requieran información sobre la carga horaria en los Laboratorios de Computación de la Universidad de las Fuerzas Armadas – ESPE MATRIZ.

CAPÍTULO IV

DESARROLLO DEL PROYECTO

4.1 Diseño de la Base de Datos

Una base de datos correctamente diseñada permite un acceso de información satisfactoria y precisa. Por ello es necesario invertir el tiempo suficiente para un buen diseño, de esta manera el resultado una base de datos que pueda aceptar cambios a futuro sin ningún inconveniente.

Es importante conocer la información que se necesitará, las tablas que formarán la base de datos y como están se relacionarán, para esto se utilizará el modelo entidad – relación misma que al ser diseñado correctamente proporcionará la información requerida para el Administrador Web y la aplicación móvil. La figura 6 muestra el modelo entidad – relación, que forman parte del diseño de la base de datos del sistema.

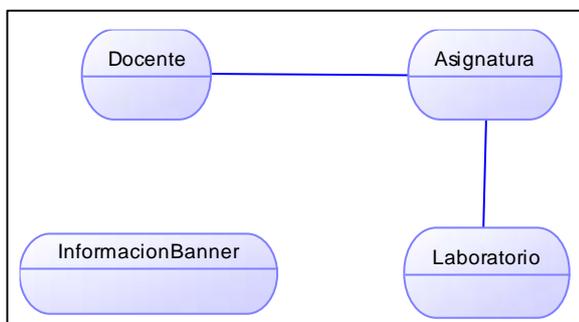


Figura 6 Modelo Entidad – Relación

4.2 Modelo de Datos

Cuando se requiere diseñar un sistema es importante poder manejar la información de una manera organizada y eficiente con el fin de que esta pueda ser manipulada de acuerdo a las necesidades de esa persona, esa parte de la realidad que deseamos la llamamos modelado de datos. Para poder crear una base de datos se emplean como herramientas los denominados modelos de datos. Se define un modelo de datos como un conjunto de símbolos,

conceptos y reglas que nos permiten representar los datos que se van a almacenar en una base de datos. El resultado de la aplicación de un modelo de datos, da lugar a lo que se denomina un esquema. Existen varios tipos de modelos de datos aplicables en distintos momentos a lo largo del proceso de creación de una base de datos. (Piñeiro Gómez, 2013).

4.2.1 Modelo Conceptual

El objetivo de este modelo es tener una base para las siguientes etapas del diseño de la base de datos, por ello es necesario identificar con los usuarios claramente las entidades, la relación entre las mismas y los atributos que lo asocian a cada uno. En la figura 7 se muestra el modelo conceptual de la Base de datos para el presente proyecto.

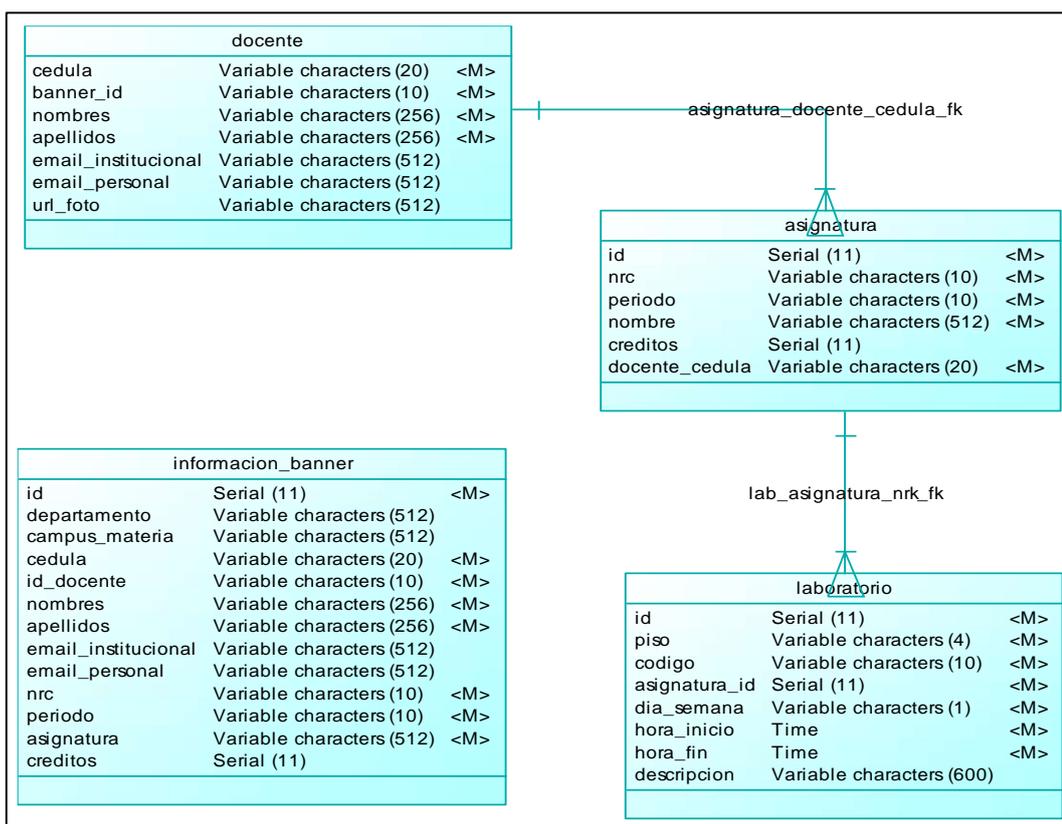


Figura 7 Modelo conceptual de la base de datos

4.2.2 Modelo Físico

El modelo físico es un modelo específico de bases de datos que representa objetos de datos relaciones (por ejemplo, tablas, columnas, claves principales y claves externas) y sus relaciones. (IBM) En la figura 8 se muestra el modelo físico correspondiente al sistema desarrollado.

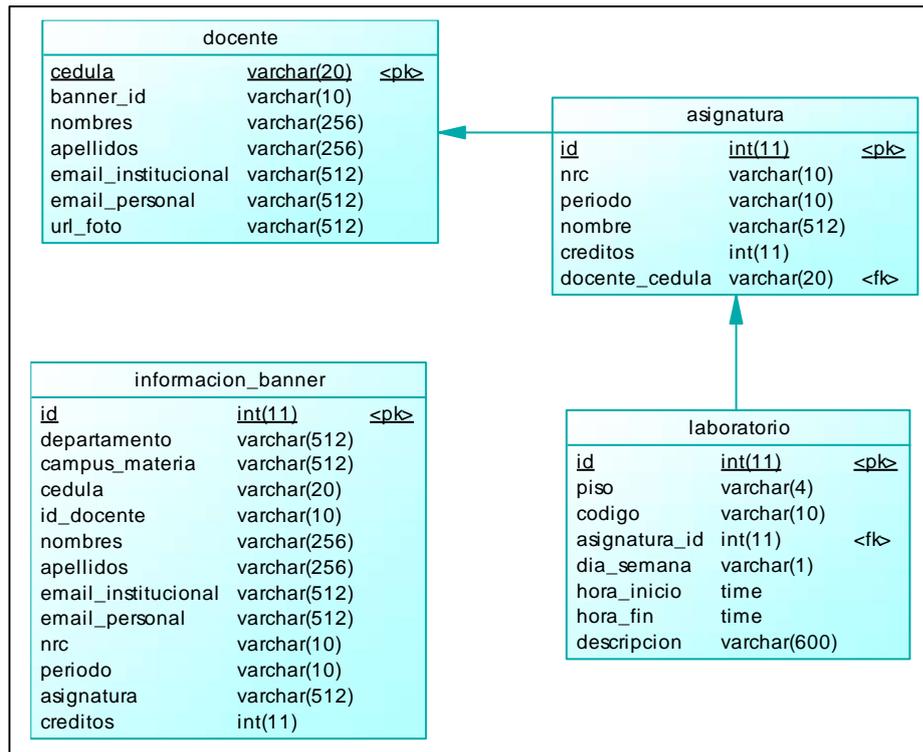


Figura 8 Modelo físico de la base de datos

4.3 Diagramas de Casos de Uso

Los diagramas de caso de uso son una forma de comportamiento UML (Lenguaje Modelado Unificado), representan como opera el sistema con las personas que usarán tanto el Administrador Web como la aplicación móvil, es decir la interacción típica entre los mismos. Por lo general encontramos se identifica los siguientes elementos: actor, casos de uso y las relaciones de uso. En la figura 9 se identifica los casos relacionados al Administrador Web, mientras que en la figura 10 se identifica los casos de uso relacionados a la aplicación móvil.

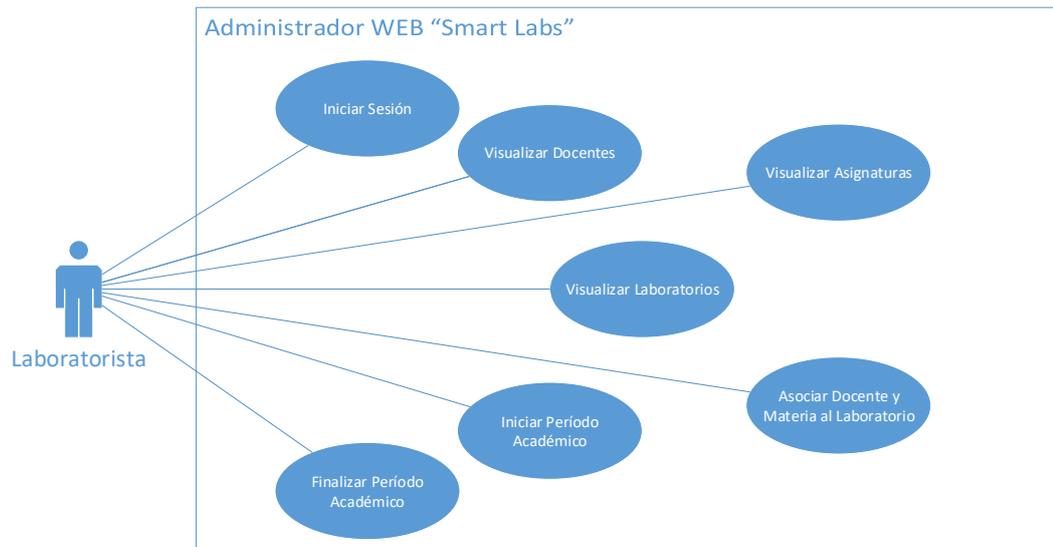


Figura 9 Caso de uso del Administrador Web "Smart Labs"

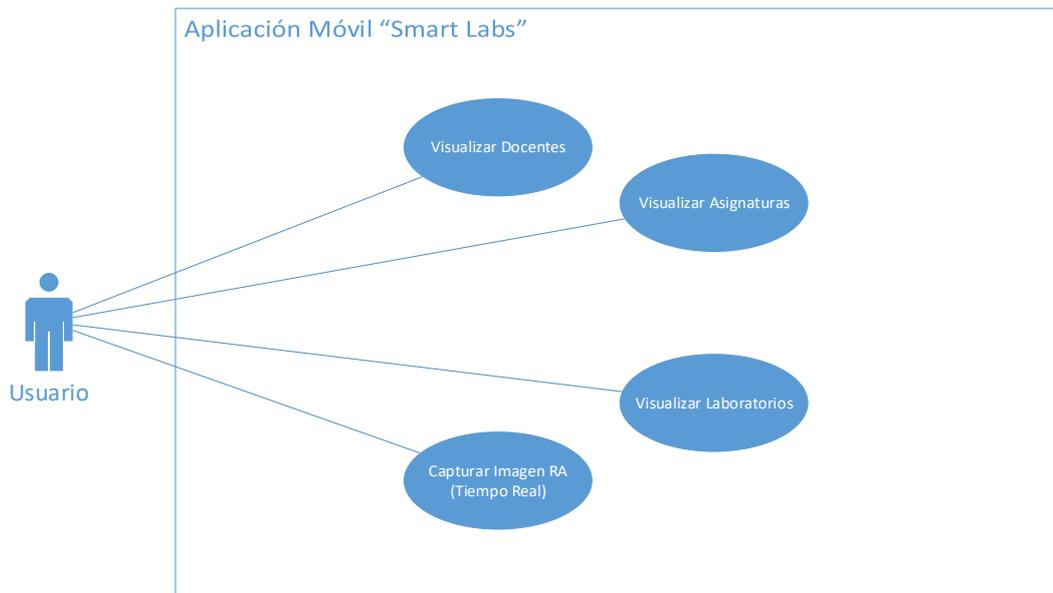


Figura 10 Caso de uso de la Aplicación Móvil "Smart Labs"

4.4 Diagrama de Secuencia

Este diagrama permite mostrar la interacción entre eventos de un sistema, con el fin de que sea más fácil de comprender. Uno de sus principales usos es para comprender la funcionalidad de un evento próximo. Los diagramas de secuencia se generan a partir del detalle de un diagrama de caso de uso. En la figura 11 se presenta el Diagrama de secuencia con relación al módulo de docentes, en la figura 12 se presenta el diagrama con respecto al módulo de asignaturas, en la figura 13 el diagrama del módulo de laboratorios y por último en la figura 14 se presenta el diagrama del módulo de realidad aumentada en la aplicación móvil.

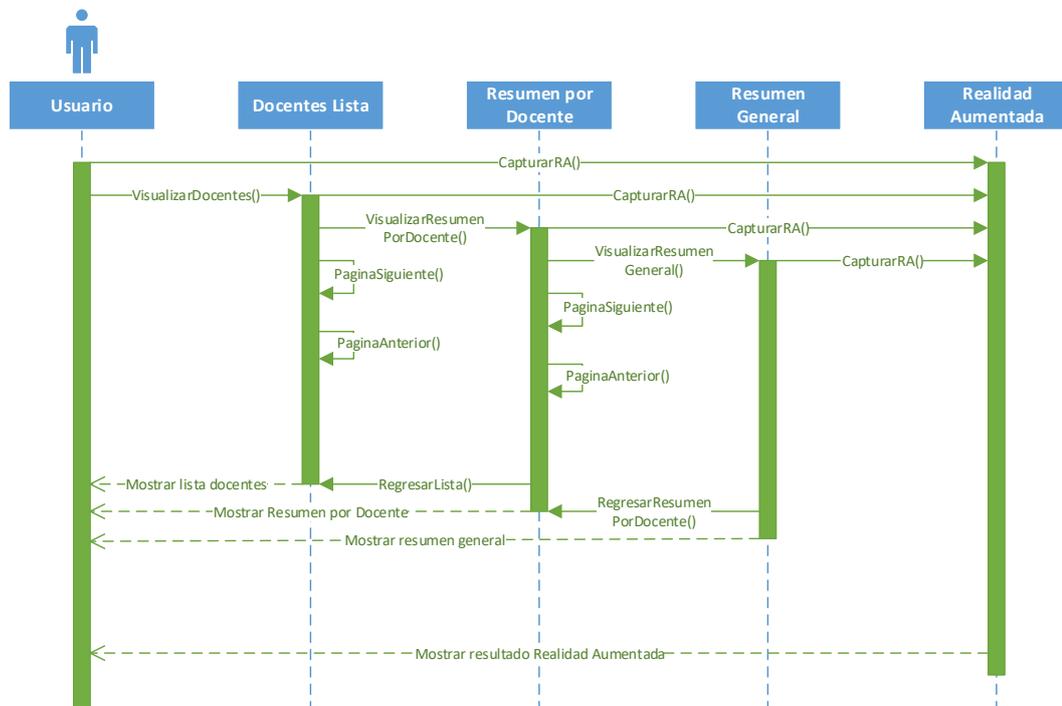


Figura 11 Escenario de módulo de Docentes de la aplicación móvil

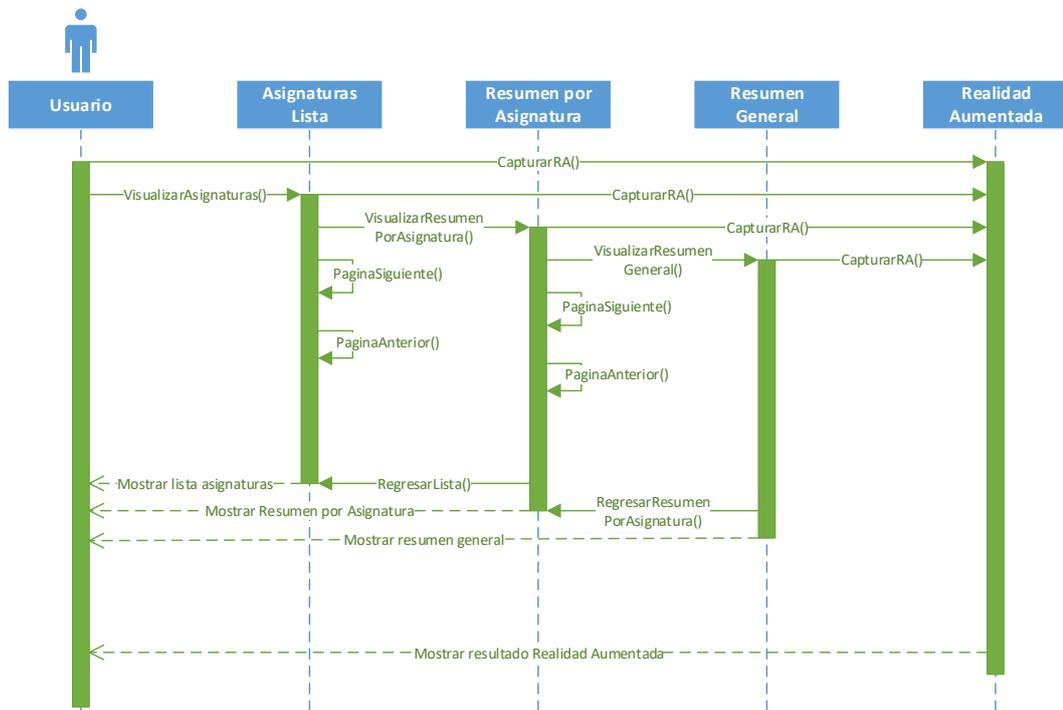


Figura 12 Escenario de módulo de Asignaturas en la aplicación móvil

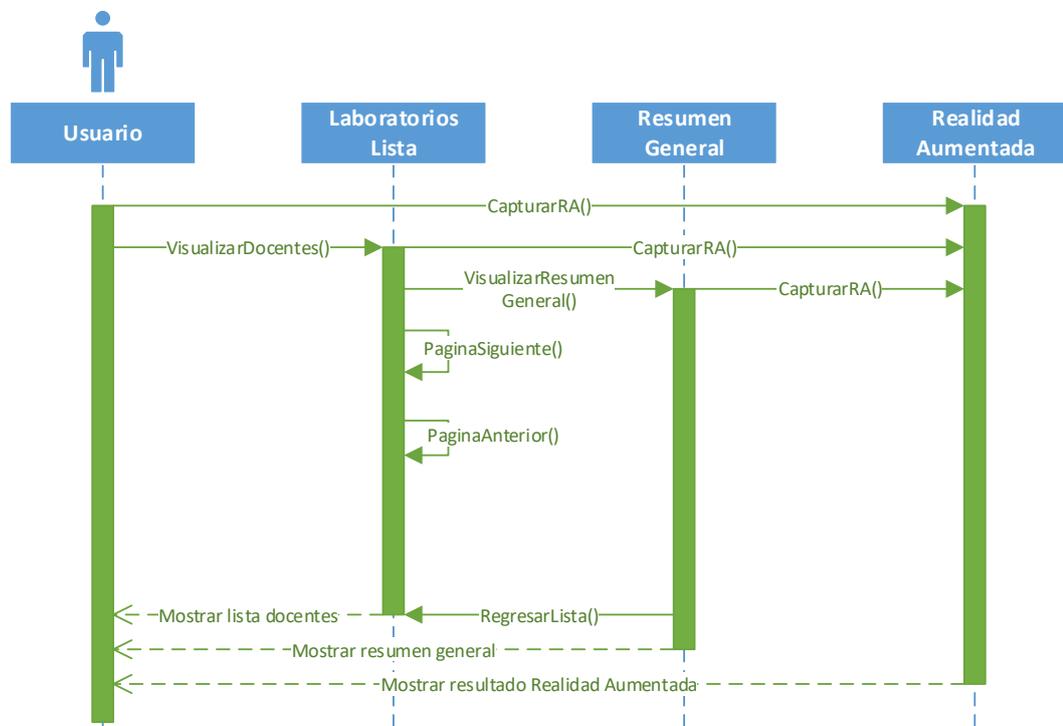


Figura 13 Escenario de módulo de Laboratorios en la aplicación móvil

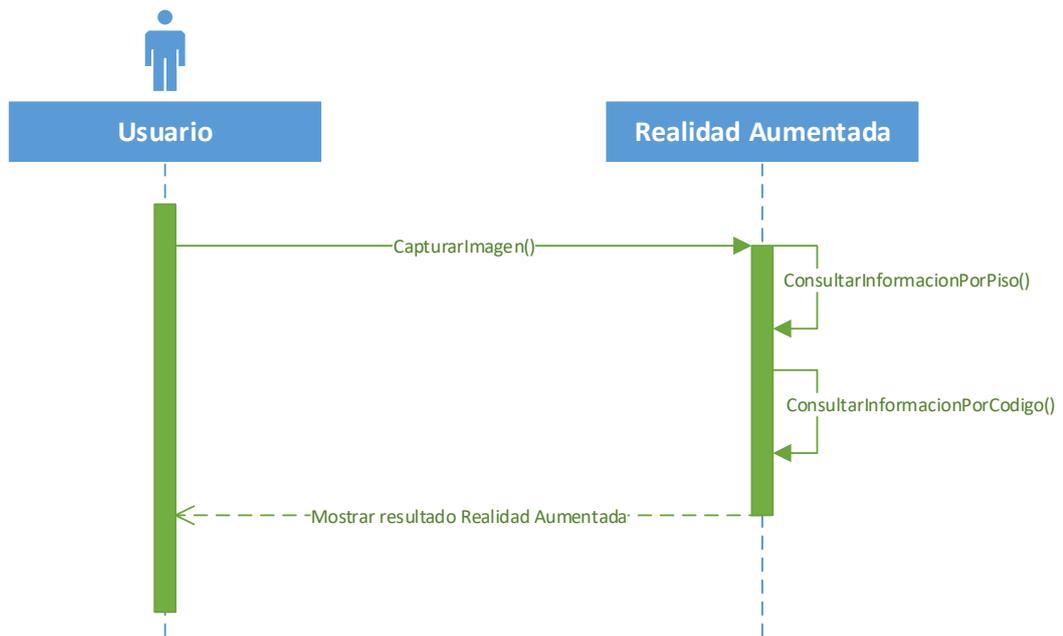


Figura 14 Módulo de realidad aumentada en la aplicación móvil

4.5 Diagrama de Clases

Es un diagrama que representa las clases del sistema mismas que están representadas por tres divisiones en donde encontramos: el nombre de la clase, los atributos, métodos y las relaciones que existen en los objetos tomando en cuenta que cada uno de estos pertenece a una de ellas. En la figura 15 se muestra el diagrama de clases perteneciente al sistema desarrollado.



Figura 15 Diagrama de clases “Smart Labs”

4.6 Diagrama de Arquitectura

En la figura 16 se muestra el diseño arquitectónico del sistema, que gestiona la carga horaria de los Laboratorios de Computación de la Universidad de las Fuerzas Armadas – ESPE MATRIZ, y en la figura 17 se muestra la arquitectura del sistema.

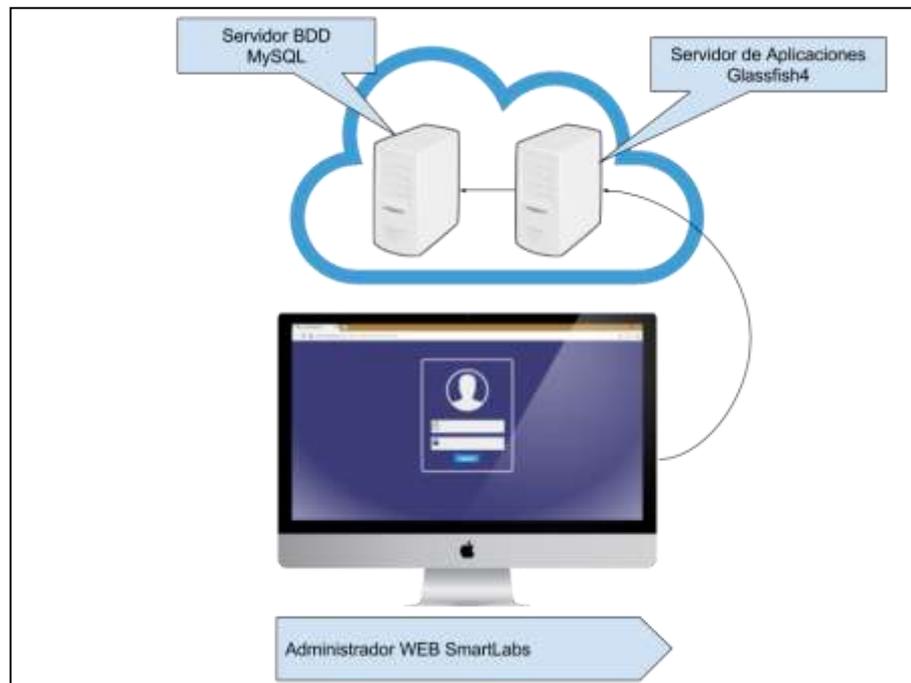


Figura 16 Diseño arquitectónico del sistema

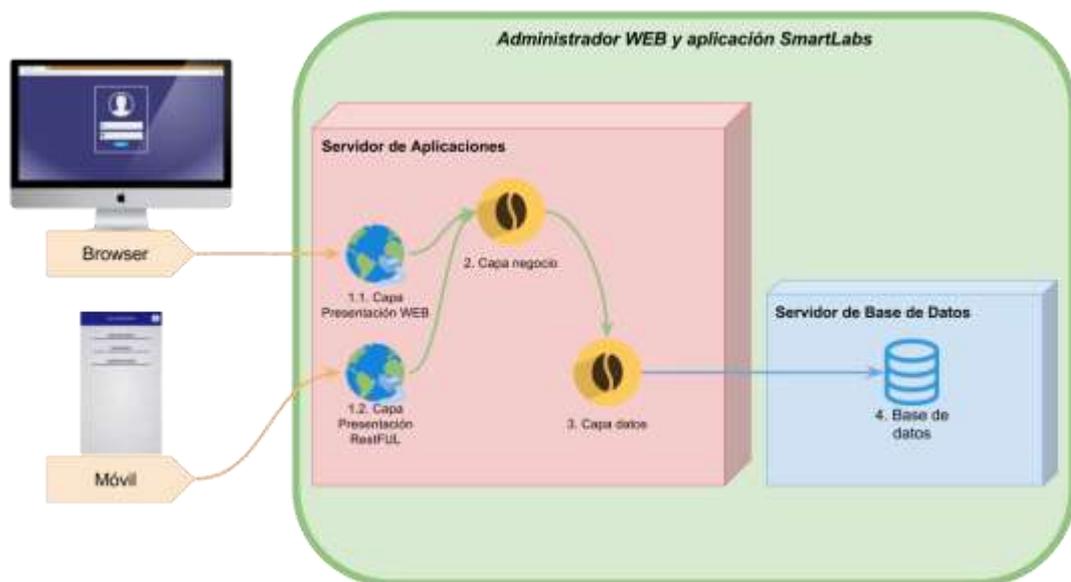


Figura 17 Diseño de la arquitectura del sistema

CAPÍTULO V

PLANIFICACIÓN, DESARROLLO Y PRUEBAS

5.1 Planificación del proyecto utilizando la metodología SCRUM

La principal herramienta en la metodología SCRUM es el Product Backlog o la pila de producto, y es de esta donde se parte para cumplir con los requerimientos que se especifican durante el capítulo 3 de este proyecto. Para ello es importante definir un formato de este documento en donde se detalle el proceso y construcción de dichos requerimientos como se muestra a continuación:

- ID: Identificador del elemento del listado
- Historia de usuario: Descripción del requerimiento del usuario
- Estimación: Tiempo requerido
- Prioridad: Grado de importancia con relación a otros elementos, el cual es definido por Product Owner.
- Criterio aceptación: criterio definido por el Product Manager o Product Owner, mismo que define si el producto cumple las expectativas del cliente.

La tabla 15 y tabla 16 detalla los nueve requerimientos identificados en el capítulo 3, las cuales llevan una serie de tareas que deben ser desarrolladas en cada iteración o Sprint y posteriormente se documentará en el Sprint Backlog o pila de tareas, mismas que deberán cumplir con el tiempo estimado. Para cumplir la directriz que conlleva el Product Backlog es necesario definir una escala de tiempo la misma que define la elaboración de una historia de usuario.

- Escala de tiempo estimado: 2 a 3 semanas.
- Prioridades: Alta, Media, Baja

Una vez estructurado y priorizado el Product Backlog, se da inicio a la etapa de Sprints o iteraciones, posteriormente el equipo de desarrollo toma

las historias de usuario y se ponen en ejecución en una sola iteración, a esto se le conoce como Sprint Backlog.

Tabla 15

Product Backlog del Administrador Web

ID	Historia de usuario	Sprint	Estimación (semanas)	Prioridad	Criterio de aceptación
1	Iniciar sesión	3	0.5	Alta	Ingreso al sistema, manteniendo un control de errores.
2	Visualizar docentes	3	0.5	Media	Listar los docentes incluyendo los filtros de la tabla.
3	Visualizar Asignaturas	3	0.5	Media	Listar las asignaturas incluyendo los filtros de la tabla.
4	Asignar asignatura al Laboratorio	3	1	Alta	Permitir asociar una materia con su respectivo docente a un laboratorio disponible.
5	Finalizar período académico	3	0.5	Media	Eliminar los datos del semestre actual, para poder cargar la información del nuevo semestre.

Tabla 16

Product Backlog de la Aplicación Móvil

ID	Historia de usuario	Sprint	Estimación (semanas)	Prioridad	Criterio de aceptación
6	Visualizar Docente	1	1	Media	Listar los docentes incluyendo los filtros de la tabla.
7	Visualizar Asignatura	1	1	Media	Listar las asignaturas incluyendo los filtros de la tabla.
8	Visualizar Laboratorio	1	1	Media	Listar los laboratorios.
9	Capturar Imagen	2	3	Alta	Enfocar la imagen y presentar la información en tiempo real con realidad aumentada.

- Pruebas del Software

Las pruebas en el desarrollo garantizan la calidad del software, siendo estas las más costosas, ya que a través de las mismas se verifica la integración y el cumplimiento de los requerimientos, permitiendo así corregir inconvenientes que se puedan presentar antes de ser entregado al cliente. Las pruebas que se conocen son las de caja blanca y negra.

- Prueba de Caja Blanca

Son pruebas estrictas y las más importantes también conocidas como estructurales, que se realizan en base al funcionamiento interno del software, es decir, verificando si las instrucciones del código muestran el comportamiento adecuado ante cualquier petición. Entre las técnicas que se usan para realizar estas pruebas se encuentran las siguientes: camino básico, prueba de bucles, camino de datos, flujo de control.

Para el sistema se hace uso de las pruebas de flujo de datos, las cuales permiten que se verifique correctamente el proceso que debe realizar el sistema de acuerdo a los datos que registra y muestra de la base de datos a la aplicación móvil y al Administrador Web.

- Prueba de Caja Negra

Son pruebas funcionales de un software realizadas desde la parte externa del mismo, sin tener en cuenta los procesos internos, por lo que es necesario que se encuentren bien definidas las interfaces, de manera que se pueda entender el funcionamiento y la interacción del software con el medio. Las técnicas que se usan para realizar estas pruebas son: análisis causa – efecto, pruebas de comparación, análisis de valores límite.

En el sistema se usan pruebas de análisis causa – efecto, por lo que se envían datos de entrada y se verifica la respuesta de dicha petición, de esta manera se comprueba que cumpla con cada uno de los requerimientos que puedan verse de forma externa en el sistema.

5.2 Desarrollo y pruebas del Sprint 1

Las historias de usuario que constan en el Sprint 1 han sido seleccionadas inicialmente para poder cumplir con el objetivo de la aplicación móvil, estas tienen una duración de 15 días laborables, es decir, 3 semanas de trabajo. En la tabla 17 se contempla las funcionalidades de visualizar docente, visualizar asignatura, y visualizar laboratorio.

Tabla 17

Historias de usuario para Sprint 1

ID	Historia de usuario	Prioridad	Criterio de aceptación
6	Visualizar Docente	Media	Listar los docentes en la interfaz de la aplicación móvil.
7	Visualizar Asignatura	Media	Listar las asignaturas en la interfaz de la aplicación móvil.
8	Visualizar Laboratorios	Media	Listar los laboratorios en la interfaz de la aplicación móvil.

5.2.1 Sprint Backlog del Sprint 1

Es importante que las historias de usuario mencionadas sean desarrolladas durante el primer Sprint ya que las mismas son parte de las principales funcionalidades de la aplicación móvil, ya que sin estas no es posible visualizar información detallada de la carga horaria en los Laboratorios. La tabla 18 describe las tareas, la persona responsable, el elemento del Product Backlog al que pertenece y la fecha de entrega para los requerimientos.

Tabla 18

Sprint Backlog del Sprint 1

Sprint 1		Inicio	Duración (semanas)	12 al 16 de junio	19 al 23 de junio	26 al 30 de junio
		12-Junio-2017	3			
ID	Tarea	Product Backlog				
1	Crear estructura de base de datos de la tabla docente.	6	X			
2	Crear capa de datos para la tabla docente.	6	X			

CONTINÚA ⇒

3	Crear servicio RESTFul para exponer la información del docente.	6	X		
4	Prueba del servicio RESTFul de docente (Prueba de caja blanca)	6	X		
5	Diseño de interfaz de usuario en aplicación móvil para docente.	6	X		
6	Prueba de la interfaz de usuario de docente (caja negra)	6	X		
7	Crear estructura de base de datos de la tabla asignatura.	7		X	
8	Crear capa de datos para la tabla asignatura.	7		X	
9	Crear servicio RESTFul para exponer la información de la asignatura.	7		X	
10	Prueba del servicio RESTFul de asignatura (Prueba de caja blanca).	7		X	
11	Diseño de interfaz de usuario en aplicación móvil para asignaturas.	7		X	
12	Prueba de la interfaz de usuario de asignaturas (caja negra).	7		X	
13	Crear estructura de base de datos de la tabla laboratorios	8			X
14	Crear capa de datos para la tabla laboratorios.	8			X
15	Crear servicio RESTFul para exponer la información de los laboratorios.	8			X
16	Prueba del servicio RESTFul de laboratorios (Prueba de caja blanca).	8			X
17	Diseño de interfaz de usuario en aplicación móvil para laboratorios.	8			X
18	Prueba de la interfaz de usuario de laboratorios (caja negra).	8			X

5.2.2 Revisión y seguimiento del Sprint 1

Se enlista las tareas completas y las pendientes en el equipo de desarrollo, con el objetivo de medir el avance de las mismas. En la tabla 19 se describe las tareas completas durante el primer Sprint. El gráfico Burn-Down Chart permite medir la velocidad y progreso del Sprint, de esta manera el equipo de trabajo puede aumentar o disminuir su ritmo de trabajo para poder cumplir con los objetivos en el tiempo estimado. La figura 18 muestra el Burn-Down-Chart del Sprint 1, con relación a las horas pendientes del trabajo, y en la figura 19 se muestra las tareas pendientes durante el Sprint.

Tabla 19

Tareas completas del Sprint 1

Tareas asignadas a: Pamela Utreras		Número de tareas: 18		
ID	Tarea	Estado	Fecha	Horas
1	Crear estructura de base de datos de la tabla docente.	Completa	12-junio-2017	8
2	Crear capa de datos para la tabla docente.	Completa	13-junio-2017	8
3	Crear servicio RESTFul para exponer la información del docente.	Completa	14-junio-2017	8
4	Prueba del servicio RESTFul de docente (Prueba de caja blanca)	Completa	15-junio-2017	4
5	Diseño de interfaz de usuario en aplicación móvil para docente.	Completa	15-16-junio-2017	8

CONTINÚA ⇒

6	Prueba de la interfaz de usuario de docente (caja negra)	Completa	16-junio-2017	4
7	Crear estructura de base de datos de la tabla asignatura.	Completa	19-junio-2017	8
8	Crear capa de datos para la tabla asignatura.	Completa	20-junio-2017	8
9	Crear servicio RESTFul para la información de la asignatura.	Completa	21-junio-2017	8
10	Prueba del servicio RESTFul de asignatura (Prueba de caja blanca).	Completa	22-junio-2017	4
11	Diseño de interfaz de usuario en aplicación móvil para asignaturas.	Completa	22-23-junio-2017	8
12	Prueba de la interfaz de usuario de laboratorios (caja negra).	Completa	23-junio-2017	4
13	Crear estructura de base de datos de la tabla laboratorios	Completa	26-junio-2017	8
14	Crear capa de datos para la tabla laboratorios.	Completa	27-junio-2017	8
15	Crear servicio RESTFul para la información de los laboratorios.	Completa	28-junio-2017	8
16	Prueba del servicio RESTFul de laboratorios (Prueba de caja blanca).	Completa	29-junio-2017	4

CONTINÚA ⇒

17	Diseño de interfaz de usuario en aplicación móvil para laboratorios.	Completa	29-30-junio-2017	8
18	Prueba de la interfaz de usuario de laboratorios (caja negra).	Completa	30-junio-2017	4

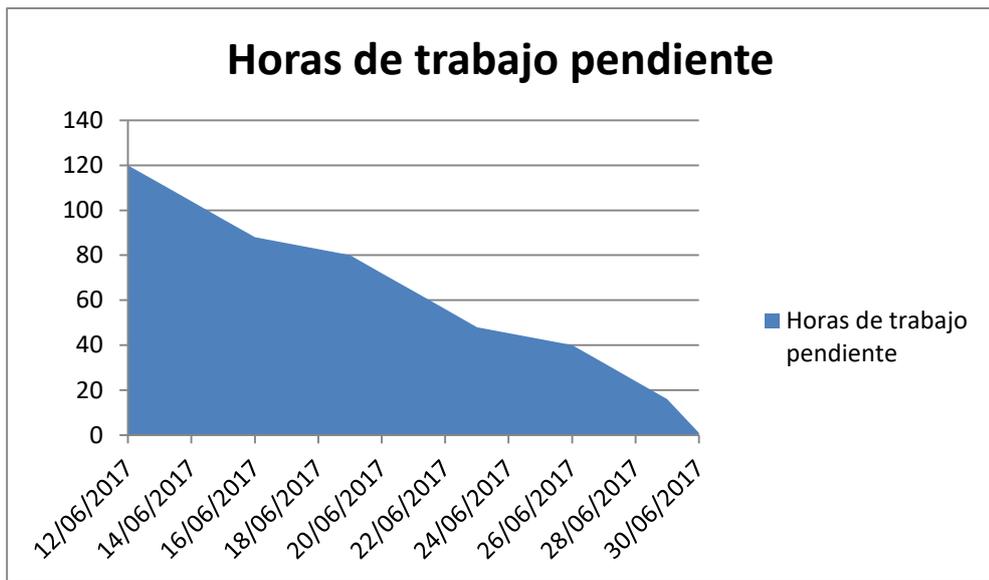


Figura 18 Horas de trabajo pendiente para Sprint 1

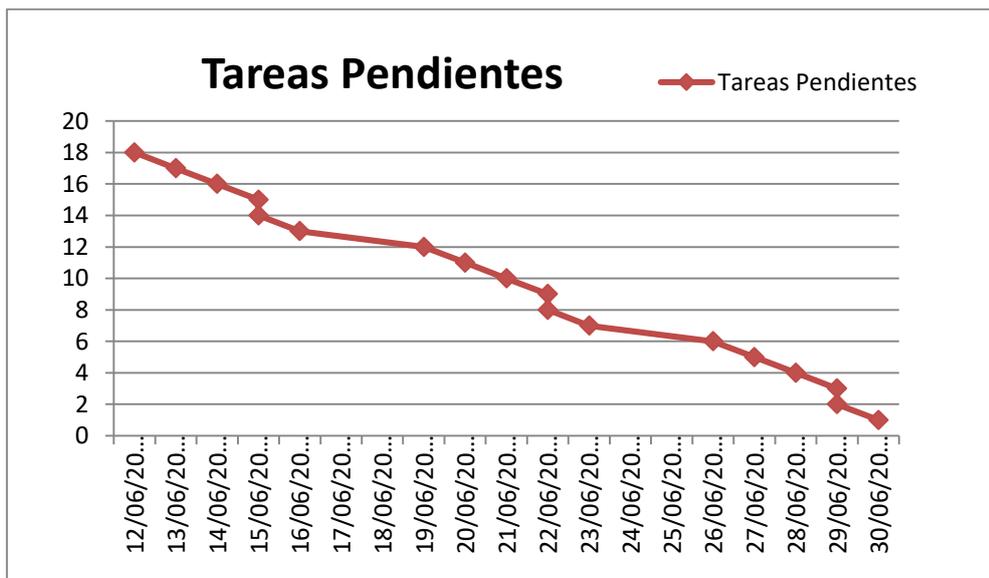


Figura 19 Tareas pendientes para Sprint 1

5.2.3 Pruebas del Sprint 1

La metodología Extreme Programming la cual es implementada en el desarrollo del sistema, señala la importancia de realizar pruebas constantemente en el desarrollo de la misma. En la figura 20 se muestra la prueba de caja blanca para la funcionalidad de visualizar docente y en la figura 21 se muestra la prueba de caja negra para esta funcionalidad durante el desarrollo del Sprint 1.

```
1 {
2   "content": [
3     {
4       "code": "G204",
5       "nrc": "2928",
6       "name": "PROGRAMACION I",
7       "document": "1704892726",
8       "names": "CESAR JAVIER",
9       "lastName": "VILLACIS SILVA",
10      "fullName": "CESAR JAVIER VILLACIS SILVA",
11      "floor": "G200"
12    },
13  {
14  },
15  "page": 1,
16  "elements": 2,
17  "pages": 1,
18  "totalElements": 2,
19  "isFirst": true,
20  "isLast": true,
21  "elementsByPage": 6
22 }
```

Figura 20 Prueba servicio RESTFul para visualizar por docente

CÓDIGO LAB.	NRC	ASIGNATURA	
C204	2928	PROGRAMACION I	👁
C205	4321	COMPUTACION GRAFICA	👁

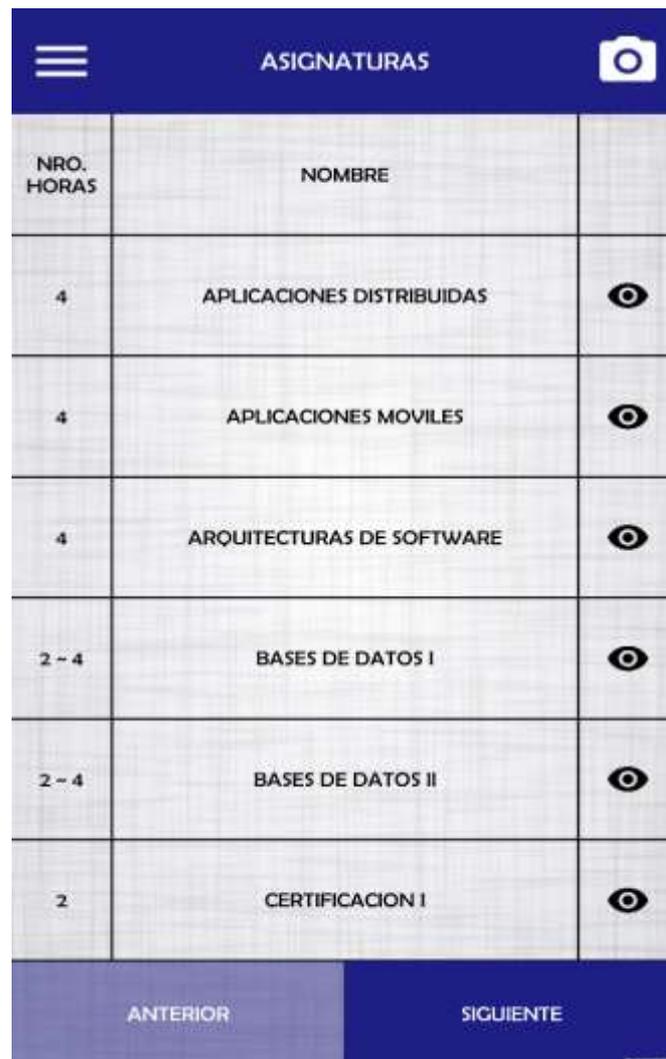
ANTERIOR SIGUIENTE

Figura 21 Prueba visualización por docente

En la figura 22 se presenta la prueba de caja blanca que cumple con la funcionalidad de visualizar asignatura, mientras que en la figura 23 se muestra la prueba de caja negra para la misma funcionalidad durante el desarrollo del Sprint 1.

```
1 {
2   "content": [
3     {
4       "name": "APLICACIONES DISTRIBUIDAS",
5       "weekHours": "4"
6     },
7     {
8       "name": "APLICACIONES MOVILES",
9       "weekHours": "4"
10    },
11   {
12     {
13       "name": "BASES DE DATOS I",
14       "weekHours": "2 ~ 4"
15     },
16     {
17       "name": "BASES DE DATOS II",
18       "weekHours": "2 ~ 4"
19     },
20     {
21       "name": "BASES DE DATOS III",
22       "weekHours": "2 ~ 4"
23     },
24     {
25       "name": "BASES DE DATOS IV",
26       "weekHours": "2 ~ 4"
27     },
28     "page": 1,
29     "elements": 6,
30     "pages": 13,
31     "totalElements": 77,
32     "isFirst": true,
33     "isLast": false,
34     "elementsByPage": 6
35   }
```

Figura 22 Prueba servicio RESTful para visualizar asignatura



NRO. HORAS	NOMBRE	
4	APLICACIONES DISTRIBUIDAS	👁️
4	APLICACIONES MOVILES	👁️
4	ARQUITECTURAS DE SOFTWARE	👁️
2 - 4	BASES DE DATOS I	👁️
2 - 4	BASES DE DATOS II	👁️
2	CERTIFICACION I	👁️

ANTERIOR SIGUIENTE

Figura 23 Prueba visualización asignaturas

En la figura 24 se presenta la prueba de caja blanca que cumple con la funcionalidad de visualizar laboratorios, mientras que en la figura 25 se muestra la prueba de caja negra para la misma funcionalidad durante el desarrollo del Sprint 1.

```

1 {
2   "floor": "G200",
3   "code": "G204",
4   "subject": {
5     "nrc": "2928",
6     "name": "PROGRAMACION I",
7     "weekHours": 4,
8     "teacherDocument": "1704892726",
9     "teacher": {
10      "document": "1704892726",
11      "names": "CESAR JAVIER",
12      "lastName": "VILLACIS SILVA",
13      "photoUrl": "http://decc-labs.von-development-studio.com:81/1704892726.jpg",
14      "fullName": "CESAR JAVIER VILLACIS SILVA"
15    }
16  },
17  "schedule": [
18    {
19      "day": "M",
20      "timeIn": "07:15:00",
21      "timeOut": "09:15:00"
22    }
23  ],
24  {
25  }
26 ]
27 }

```

Figura 24 Prueba servicio RESTful de visualizar resumen general



Figura 25 Prueba resumen general

5.3 Desarrollo y pruebas de Sprint 2

La historia de usuario que consta en el Sprint número dos cumplirá con la finalización de la aplicación móvil, estas tienen una duración de 15 días laborables, es decir, 3 semanas de trabajo. En la tabla 20 se contempla las funcionalidades de capturar imagen.

Tabla 20

Historias de usuario para Sprint 2

ID	Historia de usuario	Prioridad	Criterio de aceptación
9	Capturar Imagen	Alta	Enfocar la imagen y presentar la información en tiempo real con realidad aumentada.

5.3.1 Sprint Backlog del Sprint 2

Al igual que en el Sprint 1, es importante que las historias de usuario mencionadas sean desarrolladas en el tiempo estimado. La tabla 21 describe las tareas, la persona responsable, el elemento del Product Backlog al que pertenece y la fecha de entrega para los requerimientos para el segundo Sprint.

Tabla 21

Sprint Backlog del Sprint 2

Sprint	Inicio	Duración (semanas)			
2	3-Julio-2017	3			
ID	Tarea	Product Backlog	3 al 7 de julio	10 al 14 de julio	17 al 21 de julio
1	Crear cuenta administrativa en Furia y base de datos en la misma.	9	X		

CONTINÚA →

2	Tomar fotos de los Laboratorios de Ciencias de la Computación, los cuales son targets de realidad aumentada para Vuforia.	9	X		
3	Crear base de datos en Vuforia.	9	X		
4	Cargar fotos a la base de datos de Vuforia	9	X		
5	Cargar el paquete de base de datos de los targets al proyecto de la aplicación móvil en Unity.	9		X	
6	Enlazar cada target a un código de laboratorio.	9		X	
7	Crear servicio RESTFul para exponer la información en tiempo real de el/los laboratorios.	9			X
8	Prueba del servicio RESTFul (Prueba de caja blanca).	9			X
9	Diseño de interfaz de usuario en aplicación móvil para exposición de la información de el/los laboratorios.	9			X
10	Prueba de la interfaz de usuario (caja negra).	9			X

5.3.2 Revisión y seguimiento del Sprint 2

La tabla 22 describe las tareas completas durante el segundo Sprint. La figura 26 muestra el gráfico Burn-Down Chart que mide el progreso de este Sprint mediante las horas de trabajo pendiente, y la figura 27 presenta las tareas pendientes en relación al tiempo establecido para finalizarlas.

Tabla 22

Tareas completas del Sprint 2

Tareas asignadas a: Pamela Utreras			Número de tareas: 10	
ID	Tarea	Estado	Fecha	Horas
1	Crear cuenta administrativa en Furia y base de datos en la misma.	Completa	3-julio-2017	8
2	Tomar fotos de los Laboratorios de Ciencias de la Computación, los cuales son targets de realidad aumentada para Vuforia.	Completa	4-5julio-2017	16
3	Crear base de datos en Vuforia.	Completa	6-julio-2017	8
4	Cargar fotos a la base de datos de Vuforia	Completa	7-julio-2017	8
5	Cargar el paquete de base de datos de los targets al proyecto de la aplicación móvil en Unity.	Completa	10-11-julio-2017	16
6	Enlazar cada target a un código de laboratorio.	Completa	12-13-14-julio-2017	24
7	Crear servicio RESTFul para exponer la información en tiempo real de el/los laboratorios.	Completa	17-julio-2017	8

CONTINÚA ⇒

8	Prueba del servicio RESTFul (Prueba de caja blanca).	Completa	18-19-junio-2017	12
9	Diseño de interfaz de usuario en aplicación móvil para exposición de la información de el/los laboratorios.	Completa	19-20-junio-2017	12
10	Prueba de la interfaz de usuario (caja negra).	Completa	21-junio-2017	8

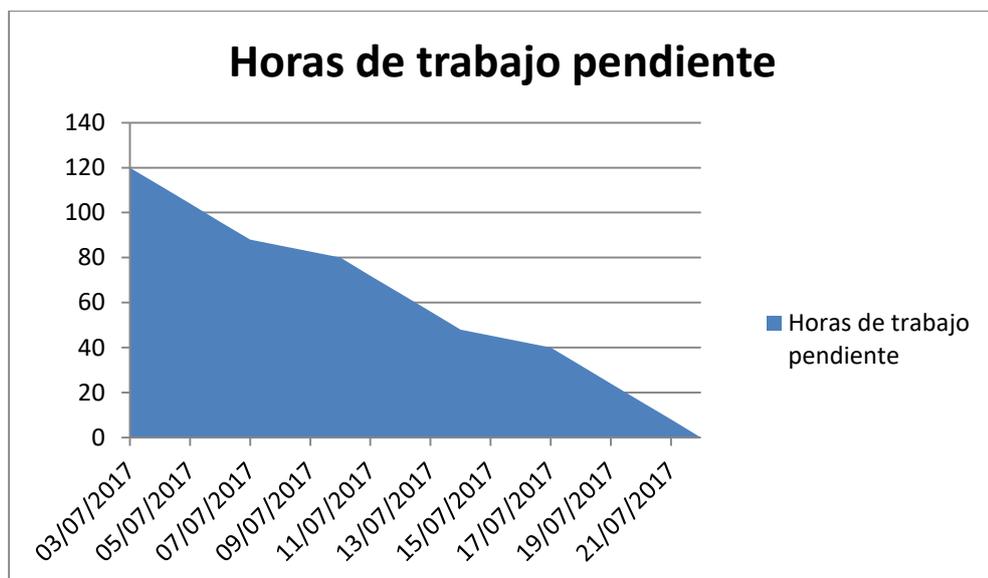


Figura 26 Horas de trabajo pendiente para el Sprint 2



Figura 27 Tareas pendientes para el Sprint 2

5.3.4 Pruebas del Sprint 2

La figura 28 y figura 29 muestran las pruebas de caja blanca con respecto a la prueba del servicio REST para exposición en tiempo real de la información de los Laboratorios en la aplicación móvil durante el desarrollo del Sprint 2, mientras que la figura 30 y figura 31 presenta las pruebas de caja negra de la funcionalidad nombrada, cumpliendo el objetivo para la historia de usuario en la aplicación.

```

1 {
2   "content": [
3     {
4       "code": "G202",
5       "nrc": "3958",
6       "name": "PROGRAMACION II",
7       "document": "1711464816",
8       "names": "JUAN FERNANDO",
9       "lastName": "GALARRAGA HURTADO",
10      "fullName": "JUAN FERNANDO GALARRAGA HURTADO",
11      "floor": "G200"
12    },
13    {
23  ],
24  "page": 1,
25  "elements": 2,
26  "pages": 1,
27  "totalElements": 2,
28  "isFirst": true,
29  "isLast": true,
30  "elementsByPage": 6
31 }

```

Figura 28 Prueba de servicio RESTful para Información en tiempo real por piso de la aplicación móvil

```

1 {
2   "floor": "G200",
3   "code": "G202",
4   "subject": {
5     "nrc": "3958",
6     "name": "PROGRAMACION II",
7     "weekHours": 6,
8     "teacherDocument": "1711464816",
9     "teacher": {
10      "document": "1711464816",
11      "names": "JUAN FERNANDO",
12      "lastName": "GALARRAGA HURTADO",
13      "photoUrl": "http://decc-labs.von-development-studio.com:81/1711464816.jpg",
14      "fullName": "JUAN FERNANDO GALARRAGA HURTADO"
15    }
16  },
17  "schedule": [
18    {
23    {
28    {
29      "day": "F",
30      "timeIn": "12:00:00",
31      "timeOut": "14:00:00"
32    }
33  ]
34 }

```

Figura 29 Prueba de servicio RESTful para Información en tiempo real por código de laboratorio en la aplicación móvil



Figura 30 Información en tiempo real por piso de laboratorios

DECC - LABORATORIOS

ASIGNATURA:
PROGRAMACION I

DOCENTE:
CESAR OSWALDO OSORIO
AGUALONGO

DÍA SEMANA	LUNES	MIÉRCOLES	VIERNES
HORA ENTRADA	09:30:00	09:30:00	09:30:00
HORA SALIDA	11:30:00	11:30:00	11:30:00

vuforia

Figura 31 Información en tiempo real por código de laboratorio

5.4 Desarrollo y pruebas del Sprint 3

Las historias de usuario que constan en el Sprint número tres cumplirán con el desarrollo del Administrador Web, estas tienen una duración de 15 días laborables, es decir, 3 semanas de trabajo. En la tabla 23 se contempla las funcionalidades de inicio de sesión, visualizar docente, visualizar asignatura, visualizar laboratorio, finalizar período académico.

Tabla 23

Historias de usuario para Sprint 3

ID	Historia de usuario	Prioridad	Criterio de aceptación
1	Iniciar sesión	Alta	Ingreso al sistema, manteniendo un control de errores.
2	Visualizar docente	Media	Listar los docentes incluyendo los filtros de la tabla.
3	Visualizar asignatura	Media	Listar las asignaturas incluyendo los filtros de la tabla.
4	Asignar asignatura al Laboratorio	Alta	Permitir asociar una materia con su respectivo docente a un laboratorio disponible.
5	Finalizar período académico	Media	Eliminar los datos del semestre actual, para poder cargar la información del nuevo semestre.

5.4.1 Sprint Backlog del Sprint 3

Como se ha mencionado en los Sprints anteriores es importante que las historias de usuario se cumplan en el tiempo planificado, de esta manera la tabla 24 describe las tareas que cumple el Sprint 3.

Tabla 24

Sprint Backlog del Sprint 3

Sprint	Inicio	Duración (semanas)	24 al 28 de julio	31 al 4 de agosto	7 al 11 de
3	24-Julio-2017	3			
ID	Tarea	Product Backlog			
1	Diseño de la interfaz del inicio de sesión.	1	X		

CONTINÚA ⇒

2	Control de pantallas por sesión activa.	1	X		
3	Inicio de sesión y salida del sistema	1	X		
4	Consulta a la base de datos para los docentes con filtros predefinidos.	2	X		
5	Prueba de caja blanca de la consulta de docentes	2	X		
6	Diseño de la interfaz de visualización de docentes.	2	X		
7	Prueba de caja negra en la interfaz de visualización de docentes	2	X		
8	Consulta a la base de datos para las asignaturas con filtros predefinidos.	3		X	
9	Prueba de caja blanca de la consulta para las asignaturas	3		X	
10	Diseño de la interfaz de visualización para las asignaturas	3		X	
11	Prueba de caja negra en la interfaz de visualización de las asignaturas	3		X	
12	Consulta a la base de datos para los laboratorios.	4		X	
13	Creación y actualización de registro de laboratorio en la base de datos.	4		X	
14	Prueba de caja blanca de la consulta, creación y actualización de laboratorios.	4		X	
15	Diseño de la interfaz de la asignación de carga horaria a los laboratorios.	4			X
16	Prueba de caja negra de la interfaz de la carga horaria.	4			X
17	Eliminación de registros de la carga horaria del semestre actual.	5			X

CONTINÚA ⇒

18	Pruebas de caja blanca de eliminación de registros de la carga horaria del semestre actual.	5			X
19	Prueba de caja negra de elección del botón para eliminación de registros del semestre actual.	5			X

5.4.2 Revisión y seguimiento del Sprint 3

La tabla 25 describe las tareas completas durante el tercer Sprint. La figura 32 muestra el gráfico Burn-Down Chart que mide el progreso de este Sprint mediante las horas de trabajo pendiente, y la figura 33 presenta las tareas pendientes en relación al tiempo establecido.

Tabla 25

Tareas completas del Sprint 3

Tareas asignadas a: Pamela Utreras		Número de tareas: 19		
ID	Tarea	Estado	Fecha	Horas
1	Diseño de la interfaz del inicio de sesión.	Completa	24-julio-17	8
2	Control de pantallas por sesión activa.	Completa	25-julio-17	8
3	Inicio de sesión y salida del sistema	Completa	26-julio17	8
4	Consulta a la base de datos para los docentes con filtros predefinidos.	Completa	27-julio-17	4
5	Prueba de caja blanca de la consulta de docentes	Completa	27-julio-17	4
6	Diseño de la interfaz de visualización de docentes.	Completa	28-julio-17	4

CONTINÚA ⇒

7	Prueba de caja negra en la interfaz de visualización de docentes	Completa	28-julio-17	4
8	Consulta a la base de datos para las asignaturas con filtros predefinidos.	Completa	31-julio-17	4
9	Prueba de caja blanca de la consulta para las asignaturas	Completa	31-julio-17	4
10	Diseño de la interfaz de visualización para las asignaturas	Completa	01-agosto-17	4
11	Prueba de caja negra en la interfaz de visualización de las asignaturas	Completa	01-agosto-17	4
12	Consulta a la base de datos para los laboratorios.	Completa	02-agosto-17	8
13	Creación y actualización de registro de laboratorio en la base de datos.	Completa	03-agosto-17	8
14	Prueba de caja blanca de la consulta, creación y actualización de laboratorios.	Completa	04-agosto-17	8
15	Diseño de la interfaz de la asignación de carga horaria a los laboratorios.	Completa	07-agosto-17	8
16	Prueba de caja negra de la interfaz de la carga horaria.	Completa	08-agosto-17	8

CONTINÚA ⇒

17	Eliminación de registros de la carga horaria del semestre actual.	Completa	09-agosto-17	8
18	Pruebas de caja blanca de eliminación de registros de la carga horaria del semestre actual.	Completa	10-agosto-17	8
19	Prueba de caja negra de elección del botón para eliminación de registros del semestre actual.	Completa	11-agosto-17	8

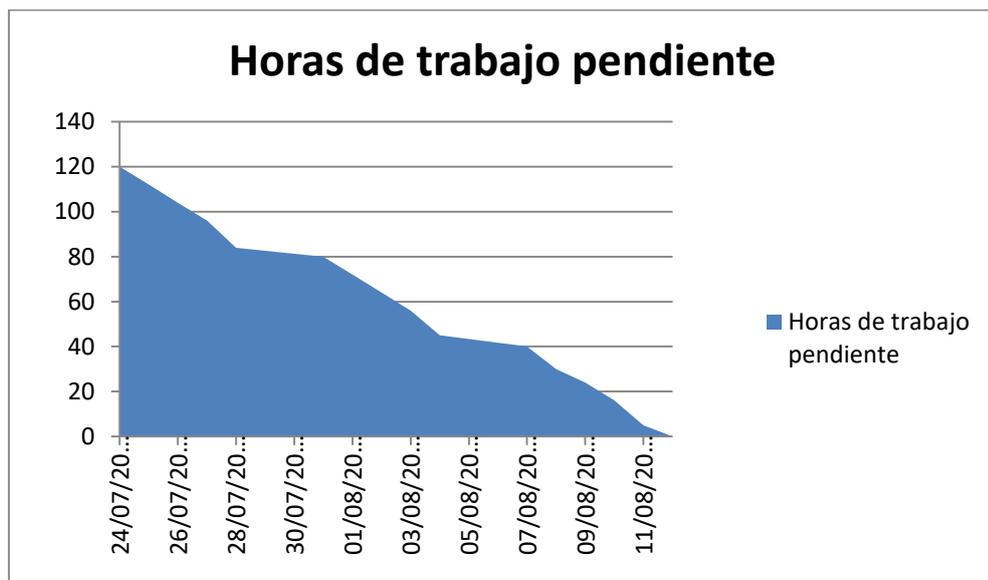


Figura 32 Horas de trabajo pendiente para el Sprint 3



Figura 33 Tareas pendientes para el Sprint 3

5.4.3 Pruebas del Sprint 3

Las pruebas del Sprint 3 serán detalladas de acuerdo al orden de las tareas relacionadas con el desarrollo del Administrador Web las cuales son:

- Consulta de docentes
- Consulta de asignaturas a la base de datos
- Consulta, creación y actualización de los Laboratorios
- Eliminación de carga horaria del semestre actual (Cierre Período Académico).

La figura 34 muestra la prueba de caja blanca que está relacionada a la consulta de Docentes, y la figura 35 presenta la caja negra que implica la misma funcionalidad.

The screenshot shows an IDE window with a Java file named `TeacherTest.java`. The code is as follows:

```

1 package ec.edu.espe.decc.test;
2
3 import org.junit.Assert;
4
5 @RunWith(SpringJUnit4ClassRunner.class)
6 @ContextConfiguration(classes = { TestConfiguration.class })
7 public class TeacherTest {
8
9     @Autowired
10    private TeacherDAO teacherDAO;
11
12    @Test
13    public void getTeachers() {
14        try {
15            System.out.println("***** getTeachers()");
16
17            Pageable<Teacher> teachers = teacherDAO.getPageableTeachers(0, 18);
18            Assert.assertNotNull("La lista de docentes no puede ser nulo", teachers);
19
20            System.out.println("***** Resultado: ");
21            System.out.println(teachers);
22
23        } catch (DeccLabsException ex) {
24            ex.printStackTrace();
25        }
26    }
27 }

```

The console output shows the result of the test execution:

```

*terminated> TeacherTest [AHR] C:\Program Files\Java\jdk1.8.0_111\bin\j...
**** Resultado:
{ 'content' : [ {
  'document' : '10341112',
  'bannerId' : 'L00051351',
  'names' : 'ALVARO DAHELO',
  'lastName' : 'UYAGUARI UYAGUARI',
  'emailInstitutional' : 'edyaguari@espe.edu.ec',
  'emailPersonal' : 'ayaguari@hotmail.com',
  'photoUrl' : 'null',
  'fullName' : 'ALVARO DAHELO UYAGUARI UYAGUARI'
}, {
  'document' : '1303120514',
  'bannerId' : 'L00007345',
  'names' : 'EDDIE EGBERTO',
  'lastName' : 'GALARZA ZAMBRANO',
  'emailInstitutional' : 'egalarza@espe.edu.ec',
  'emailPersonal' : 'egalarzaz@gmail.com',
  'photoUrl' : 'null',
  'fullName' : 'EDDIE EGBERTO GALARZA ZAMBRANO'
}, {
  'document' : '1306053720',
  'bannerId' : 'L00056405',
  'names' : 'SANG GUAN',
  'lastName' : 'YOO PARK',
  'emailInstitutional' : 'yysang@espe.edu.ec',
  'emailPersonal' : 'sangguan@hotmail.com',
  'photoUrl' : 'null',
  'fullName' : 'SANG GUAN YOO PARK'
}, {
  'document' : '1702379023',
  'bannerId' : 'L00007563',
  'names' : 'LUIS ALBERTO',
  'lastName' : 'ESCOBAR QUIÑA',

```

Figura 34 Prueba de caja blanca para Consulta de Docentes

The screenshot shows a web application interface with a table of teachers. The table has the following columns: **Cédula**, **Nombres**, **Apellidos**, **Email Institucional**, and **Email Personal**. The data is as follows:

Cédula	Nombres	Apellidos	Email Institucional	Email Personal
10341112	ALVARO DAHELO	UYAGUARI UYAGUARI	edyaguari@espe.edu.ec	ayaguari@hotmail.com
1303120514	EDDIE EGBERTO	GALARZA ZAMBRANO	egalarza@espe.edu.ec	egalarzaz@gmail.com
1306053720	SANG GUAN	YOO PARK	yysang@espe.edu.ec	sangguan@hotmail.com
1702379023	LUIS ALBERTO	ESCOBAR QUIÑA	escobar@espe.edu.ec	escobarqui@espe.edu.ec
170294403	PAUSTO HONORATO	MENDES DECEPRA	pmendes@espe.edu.ec	pmendes@gmail.com
1704299747	MARIO BERNABE	ROMEGAS	romega@espe.edu.ec	
170448818	CARLOS ESTEBAN	MONTEGRO ARBAS	cmontegro@espe.edu.ec	cmontegro@gmail.com
1704538075	MYRIAM	NOUVEY PURVA	mnouvey@espe.edu.ec	mnouvey@gmail.com
1704837758	CESAR JAVIER	VILLACI SILVA	cvillaci@espe.edu.ec	cvillaci@hotmail.com
170525449	CARLOS TERCIO	BROCEL SILVA	cbrocel@espe.edu.ec	carlosbrocel@hotmail.com
170535373	ADRIAN FERNANDEZ	DUARTE FLORES	aduarte@espe.edu.ec	aduartefernandez@gmail.com
170547527	LUIS ALBERTO	QUERRA CRUZ	lquerra@espe.edu.ec	albertquerra20@yahoo.es
170560612	FREDY ABELARDO	QUEJAGA HORA	fquejaga@espe.edu.ec	fquejaga@gmail.com
170588099	CECILIA MELINA	HERNANDEZ RAZA	cmhernandez@espe.edu.ec	cmhernandez710@gmail.com
170670191	JOSE ALBERTO	SANCHO ARBAS	jsancho@espe.edu.ec	jsanchoarbas@yahoo.es
1707057701	WALTER MARCELO	PUERTES DIAZ	wpuertes@espe.edu.ec	wpuertes@hotmail.com
170729178	KAMIRO RAMIRO	DELGADO RODRIGUEZ	krdelgado@espe.edu.ec	
1707383614	GELMA JENETH	TOAZA MARRANCO	gtocaza@espe.edu.ec	gtocaza@yahoo.es

Figura 35 Prueba de caja negra para visualización de Docentes

La figura 36 muestra la prueba de caja blanca que está relacionada a la consulta de asignaturas, y la figura 37 presenta la caja negra que implica la misma funcionalidad en el Administrador Web.

```

1 package ec.edu.espe.decc.test;
2
3 import org.junit.Assert;
4
5 @RunWith(SpringJUnit4ClassRunner.class)
6 @ContextConfiguration(classes = { TestConfiguration.class })
7 public class SubjectTest {
8
9     @Autowired
10    private SubjectDAO subjectDAO;
11
12
13    @Test
14    public void getSubjects() {
15        try {
16            System.out.println("***** getSubjects()");
17
18            Pageable<Subject> subjects = subjectDAO.getPageableSubjects(0, 10);
19            Assert.assertNotNull("La lista de asignaturas no puede ser nulo", subjects);
20
21            System.out.println("***** Resultado: ");
22            System.out.println(subjects);
23
24        } catch (DeccLabException ex) {
25            ex.printStackTrace();
26        }
27    }
28 }

```

```

<terminated> SubjectTest [10m] C:\Program Files\Java\
[**** Resultado:
{ 'content' : [
  {
    'id' : '22',
    'nrc' : '1285',
    'period' : '201710',
    'weekHours' : '0.0',
    'teacherDocument' : '502166796'
  }, {
    'id' : '174',
    'nrc' : '1297',
    'period' : '201710',
    'weekHours' : '0.0',
    'teacherDocument' : '1708856781'
  }, {
    'id' : '96',
    'nrc' : '1341',
    'period' : '201710',
    'weekHours' : '0.0',
    'teacherDocument' : '1707249072'
  }, {
    'id' : '15',
    'nrc' : '1304',
    'period' : '201710',
    'weekHours' : '0.0',
    'teacherDocument' : '502023275'
  }, {
    'id' : '89',
    'nrc' : '4389',
    'period' : '201710',
    'weekHours' : '0.0',
    'teacherDocument' : '1711659456'
  }, {
    'id' : '106',

```

Figura 36 Prueba de caja blanca para Consulta de Asignaturas

NRC	Nombre	Cédula	Nombre(s)	Apellido(s)	# Créditos
1289	APLICACIONES MOVILES	502366796	PIWARLIN JAVIER	MONTALUISA YUGLA	00
1397	ARQUITECTURAS DE SOFTWARE	1788894701	EDUARDO MAURICIO	CAMIANA ORTEGA	00
1341	BASES DE DATOS I	1707249072	MAGI PAUL	DAZ DUFGA	00
1304	BASES DE DATOS I	502020275	MILTON PATRICIO	NAVALEMOA	00
4289	BASES DE DATOS I	1718878454	TATIANA KARENA	NECEGA MORALES	00
3991	BASES DE DATOS II	400652500	OSWALDO EFRAN	DAZ RODRIGUEZ	00
4349	BASES DE DATOS II	1707249072	MAGI PAUL	DAZ DUFGA	00
1305	BASES DE DATOS II	502020275	MILTON PATRICIO	NAVALEMOA	00
1285	CERTIFICACION I	501940988	EDGAR FABIAN	MONTALUISA PRUTARRI	00
3814	CERTIFICACION II	502403883	Diego ISMAEL	VISGALZO SANCHEZ	00
3010	CLOUD COMPUTING	1738802925	DIEGO MIGUEL	MARCIO PARRA	00
3516	COMERCIO ELECTRONICO	1712330512	ALEXANDER OMAR	SALCEDO ANDRADE	00
4245	COMERCIO ELECTRONICO	1712330512	ALEXANDER OMAR	SALCEDO ANDRADE	00
1178	COMERCIO ELECTRONICO	600911588	CARLOS OSWALDO	CAIZACUANO CHIBEO	00
4304	COMERCIO ELECTRONICO	400911588	CARLOS OSWALDO	CAIZACUANO CHIBEO	00
3009	COMERCIO ELECTRONICO	180092094	CESAR ARTURO	DE LA TORRE DAVALOS	00
3543	COMERCIO ELECTRONICO	1702379233	LUIS ALBERTO	ESCORBAR OLIVERA	00
1341	COMERCIO ELECTRONICO	1713484834	JUAN FERNANDO	GALARRAGA HURTADO	00

Figura 37 Prueba de caja negra para Visualización de Asignaturas

La figura 38 y figura 39 muestran las pruebas de caja blanca que está relacionada a la consulta, creación y actualización de la asignación de materias a Laboratorios y la figura 40 y figura 41 presenta la caja negra que implica la misma funcionalidad en el Administrador Web.

```

aplicacionLaborComputacionEspe - Java EE - decs-laboratorios/src/test/java/ec/edu/espe/decc/test/LaboratoryTest.java - Eclipse
File Edit Refactor Source Navigate Search Project Run Window Help
LaboratoryTest.java
1 package ec.edu.espe.decc.test;
2
3 import java.util.List;
15
16 @RunWith(SpringJUnit4ClassRunner.class)
17 @ContextConfiguration(classes = { TestConfiguration.class })
18 public class LaboratoryTest {
19
20     @Autowired
21     private LaboratoryDAO laboratoryDAO;
22
23     @Test
24     public void getLaboratories() {
25         try {
26             System.out.println("***** getLaboratories()");
27
28             List<Laboratory> laboratories = laboratoryDAO.getWorkload("G201");
29
30             System.out.println("***** Resultado: ");
31             System.out.println(laboratories);
32
33         } catch (DeccLabsException ex) {
34             ex.printStackTrace();
35         }
36     }
37
38     public void addLaboratory() {}
39
40     public void updateLaboratory() {}
41
42 }
43
<terminated> LaboratoryTest.getLaboratories [
***** Resultado:
[[
    {
        'id' : 186,
        'floor' : 'G200',
        'code' : 'G201',
        'subjectId' : 109,
        'dayOfWeek' : 'F',
        'timeStart' : '15:00:00',
        'timeEnd' : '17:00:00',
        'description' : 'null'
    }, {
        'id' : 204,
        'floor' : 'G200',
        'code' : 'G201',
        'subjectId' : 84,
        'dayOfWeek' : 'M',
        'timeStart' : '15:00:00',
        'timeEnd' : '17:00:00',
        'description' : 'null'
    }, {
        'id' : 205,
        'floor' : 'G200',
        'code' : 'G201',
        'subjectId' : 84,
        'dayOfWeek' : 'W',
        'timeStart' : '15:00:00',
        'timeEnd' : '17:00:00',
        'description' : 'null'
    }, {
        'id' : 79,
        'floor' : 'G200',
        'code' : 'G201',
        'subjectId' : 91,

```

Figura 38 Prueba de caja blanca para Consultar Laboratorios

The screenshot shows an IDE window titled 'LaboratoryTest.java'. The code defines a test class with a method `updateLaboratory()` that creates a `Laboratory` object, sets its attributes, and calls `laboratoryDAO.updateSchedule(lab)`. The console output shows a series of debug messages from Spring Framework, followed by the execution of `updateLaboratory()` and the output of `rowsAffected` as 1.

```

1 package ec.edu.espe.decc.test;
2
3 import java.util.List;
4
5 @RunWith(SpringJUnit4ClassRunner.class)
6 @ContextConfiguration(classes = { TestConfiguration.class })
7 public class LaboratoryTest {
8
9     @Autowired
10    private LaboratoryDAO laboratoryDAO;
11
12    public void getLaboratories() {}
13
14    public void addLaboratory() {}
15
16    @Test
17    public void updateLaboratory() {
18        try {
19            System.out.println("***** updateLaboratory()");
20
21            Laboratory lab = new Laboratory();
22            lab.setSubjectId(128L);
23            lab.setCode("0201");
24            lab.setDayOfWeek("F");
25            lab.setTimeStartStr("10:30:00");
26            Integer rowsAffected = laboratoryDAO.updateSchedule(lab);
27            Assert.assertEquals("Se debe realizar un update del registro", 1, rowsAffected);
28
29            System.out.println("***** Filas actualizadas: ");
30            System.out.println(rowsAffected);
31
32        } catch (DeccLabsException ex) {
33            ex.printStackTrace();
34        }
35    }
36 }

```

```

<terminated>- LaboratoryTest.updateLaboratory [JUnit] C:\Program Files\Ja
03:16:19.171 [main] DEBUG org.springframework.beans.factory
03:16:19.172 [main] DEBUG org.springframework.beans.factory
03:16:19.172 [main] DEBUG org.springframework.beans.factory
03:16:19.172 [main] DEBUG org.springframework.beans.factory
03:16:19.172 [main] DEBUG org.springframework.beans.factory
03:16:19.173 [main] DEBUG org.springframework.beans.factory
03:16:19.175 [main] DEBUG org.springframework.beans.factory
03:16:19.180 [main] DEBUG org.springframework.beans.factory
03:16:19.180 [main] DEBUG org.springframework.beans.factory
03:16:19.181 [main] DEBUG org.springframework.beans.factory
03:16:19.222 [main] DEBUG org.springframework.context.sug
03:16:19.222 [main] DEBUG org.springframework.beans.factory
03:16:19.225 [main] DEBUG org.springframework.core.env.Pr
03:16:19.231 [main] DEBUG org.springframework.test.conte
03:16:19.231 [main] DEBUG org.springframework.test.conte
03:16:19.235 [main] DEBUG org.springframework.beans.factory
03:16:19.246 [main] DEBUG org.springframework.beans.factory
03:16:19.247 [main] DEBUG org.springframework.beans.factory
03:16:19.292 [main] DEBUG org.springframework.test.conte
***** updateLaboratory()
03:16:19.357 [main] DEBUG org.springframework.jdbc.core.J
03:16:19.359 [main] DEBUG org.springframework.jdbc.core.J
03:16:19.365 [main] DEBUG org.springframework.jdbc.datas
03:16:19.365 [main] DEBUG org.springframework.jdbc.datas
03:16:20.206 [main] DEBUG org.springframework.jdbc.core.J
03:16:20.223 [main] DEBUG org.springframework.jdbc.datas
***** Filas actualizadas: 1
03:16:20.226 [main] DEBUG org.springframework.test.conte
03:16:20.228 [main] DEBUG org.springframework.test.conte
03:16:20.288 [Thread-0] INFO org.springframework.context.
03:16:20.298 [Thread-0] DEBUG org.springframework.beans.f
03:16:20.291 [Thread-0] DEBUG org.springframework.beans.f

```

Figura 39 Prueba de caja blanca para Actualización de Laboratorios

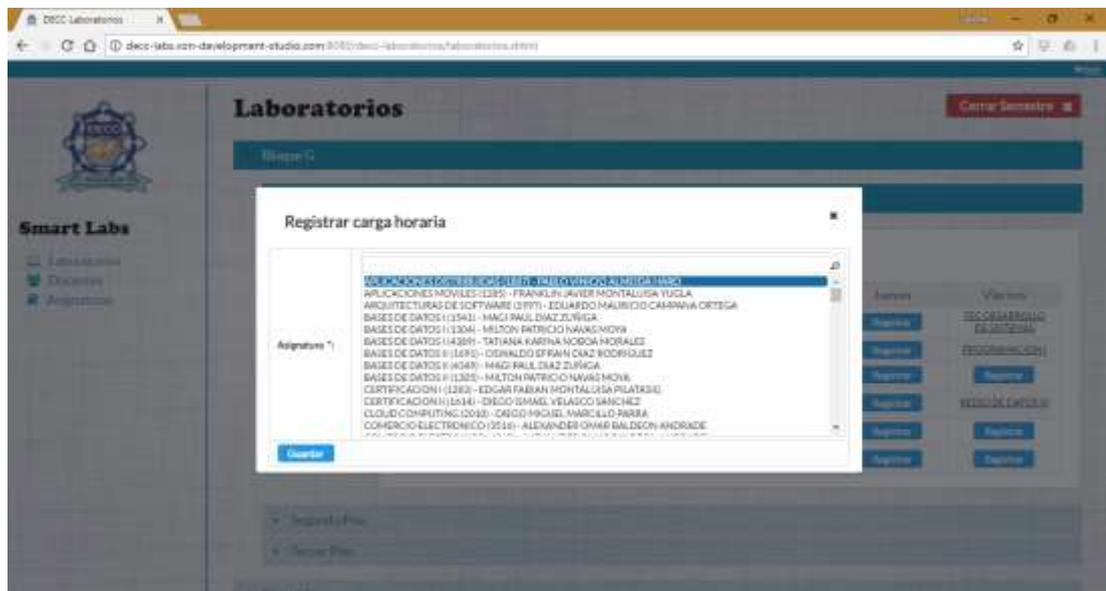


Figura 40 Prueba de caja negra para Asignación de carga horaria a Laboratorio



Figura 41 Prueba de caja negra para Actualización de Laboratorios

La figura 42 y figura 43 muestran las pruebas de caja blanca que está relacionada a la eliminación de carga horaria en los Laboratorios del período actual y la figura 44 presenta la caja negra que implica la misma funcionalidad en el Administrador Web.

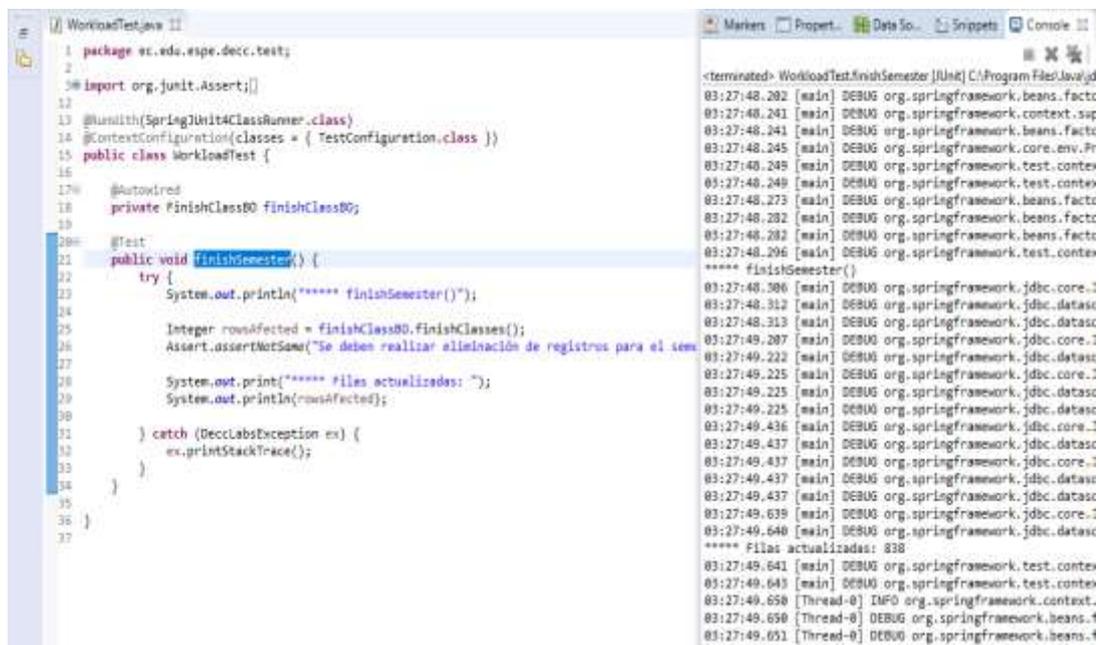


Figura 42 Prueba de caja blanca para eliminación de carga horaria en los laboratorios

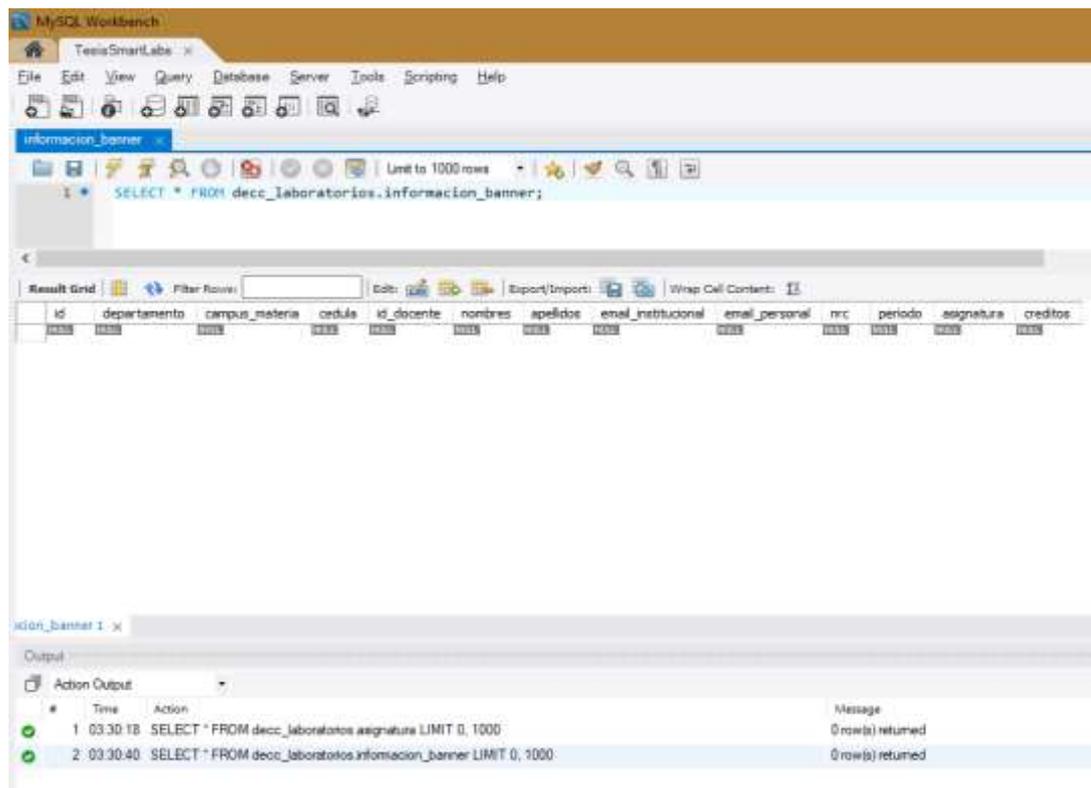


Figura 43 Registro de datos eliminados en los laboratorios



Figura 44 Prueba de caja negra para eliminación de carga horaria en los laboratorios

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones

- El manejo de las metodologías SCRUM y XP durante el desarrollo del sistema tanto del Administrador Web como de la aplicación móvil permitieron que cada requerimiento se cumpla a cabalidad en los tiempos establecidos.
- Los requerimientos especificados en el documento IEEE 830 fueron detallados de manera clara y oportuna para el desarrollo óptimo del sistema.
- El análisis y diseño de la base de datos permitió que la aplicación móvil cumpla con el objetivo de proyecto de grado lo cual consiste en mostrar la información en tiempo real de la carga horaria de los laboratorios de computación de la Universidad de las Fuerzas Armadas – ESPE MATRIZ, haciendo uso de tecnología en realidad aumentada.
- La creación de los servicios RESTFul fue la mejor solución encontrada para el proyecto debido a su baja complejidad tanto en la creación como en la exposición de la información, por lo que la aplicación móvil interactúa con la base de datos por medio de una capa intermedia y recolectando los datos a través de estos servicios Web.
- Al trabajar con tecnología de realidad aumentada es necesario utilizar motores específicos para un buen manejo de la misma, como lo es Unity 3D; motor dedicado a videojuegos que permite interactuar directamente con el hardware del equipo. Para el presente proyecto fue necesario usar la cámara la cual ayudó a capturar las imágenes en tiempo real y desplegar la información requerida.
- El manejo de las pruebas de caja blanca y negra hicieron posible la detección de posibles fallos y errores en el desarrollo del sistema, lo que permitió la solución de estos sin contratiempos.

- El administrador Web está diseñado para que los señores Laboratoristas hagan uso de ella para el manejo de la carga horaria de los Laboratorios de Computación. Es importante recalcar que la aplicación móvil está disponible para el público en general, estudiantes y docentes que deseen hacer uso de la misma a través de un Smartphone.
- Con la evolución de la tecnología, surge la necesidad de crear soluciones a través de dispositivos electrónicos que faciliten el manejo de información y esta nos pueda orientar a tomar decisiones acertadas en el lugar que nos encontremos.

6.2 Recomendaciones

- Incrementar la información de cada laboratorio en cuanto a la capacidad de alumnos que pueden ingresar en ellas.
- Detallar el número de computadores que se encuentran en mantenimiento por cada laboratorio, permitiendo brindar información oportuna a los docentes que deseen impartir clases y hagan uso de las mismas.
- Brindar esta solución tecnológica a los diferentes Laboratorios que dispone la Universidad de las Fuerzas ESPE – MATRIZ para uso de todo el alumnado y docentes en general.
- Debido al avance de nuevas tecnologías que se suman continuamente en el mercado en relación a la Realidad Aumentada, es importante que se incluya una materia que aplique estos conocimientos para que las nuevas generaciones estén preparadas para afrontar cualquier tipo de proyecto.
- La tecnología está en constante cambio, existen plataformas o herramientas que no se ajustan en su totalidad por diferencias de versiones como fue el caso de Spring Framework que se conecta con el servidor de aplicaciones Glassfish 4, requerimiento proporcionado por la institución. Debido a la experiencia integrando estas dos

tecnologías es recomendable actualizar a un servidor de aplicaciones más actual como lo es WildFly, permitiendo realizar una integración con el entorno de desarrollo de forma eficiente.

- Disponer en su totalidad de uno o varios servidores de desarrollo para una mayor facilidad en la configuración y despliegue de los aplicativos realizados para la institución. Además de proporcionar las herramientas configuradas para el desarrollo del mismo, debido a que puede llevar un mayor tiempo replicar la configuración y se tiene un alto riesgo de que la configuración que se realice no cumpla con los estándares de la Institución.

REFERENCIAS BIBLIOGRAFICAS

(8 de Octubre de 2013). Recuperado el 24 de Mayo de 2016, de Extreme Programming: <http://www.extremeprogramming.org/>

Álvarez García, A., de las Heras del Dedo, R., & Lasa Gómez, C. (2012). *Métodos Ágiles y Scrum*. Madrid: Anaya Multimedia.

Anónimo. (s.f.). *Metodología SCRUM*. Recuperado el 15 de Mayo de 2016, de Procesos de Software: <https://procesosdesoftware.wikispaces.com/METODOLOGIA+SCRUM>

Bustamente, D. (2014). *Metodología Actual - Metodología XP*. Barinas.

Contenedor EJB. (7 de Julio de 2012). Obtenido de <http://glosarios.servidor-alicante.com/java/contenedor-ejb>

Cruz, A. (14 de Enero de 2014). *Realidad Aumentada con Vuforia*. Recuperado el 24 de Mayo de 2016, de Desarrollo Libre: <http://www.desarrollolibre.net/blog/tema/73/android/realidad-aumentada-con-vuforia>

Editor de Unity. (2016). Recuperado el 24 de Mayo de 2016, de Unity: <https://unity3d.com/es/unity/editor>

Herrera Uribe, E., & Valencia Ayala, L. (2007). *Del Manifiesto Ágil sus valores y Principios*. Colombia.

IBM. (s.f.). *Modelos Físicos de datos*. Obtenido de https://www.ibm.com/support/knowledgecenter/es/SSRTLW_8.5.1/com.ibm.datatools.core.ui.doc/topics/cphysmod.html

IEEE. (10 de 10 de 2014). *IEEE Standards*. Obtenido de www.ieee.org

Letelier, P., & Penádes, C. (15 de Enero de 2006). Recuperado el 24 de Mayo de 2016, de *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*: <http://www.cyta.com.ar/ta0502/v5n2a1.htm>

Microsoft. (2017). *IDE de Visual Studio*. Obtenido de <https://www.visualstudio.com/es/vs/?rr=https%3A%2F%2Fwww.google.com.ec%2F>

Microsoft. (2017). *Lenguaje Visual C#*. Obtenido de [https://msdn.microsoft.com/es-es/library/aa287558\(v=vs.71\).aspx](https://msdn.microsoft.com/es-es/library/aa287558(v=vs.71).aspx)

Mocholi, A. (28 de Julio de 2014). *Desarrollo de juegos con Unity 3D*. Recuperado el 24 de Mayo de 2016, de YeePLY: <https://www.yeeply.com/blog/desarrollo-de-juegos-con-unity-3d/>

Oracle. (2014). *The Java EE 7 Tutorial*. Obtenido de <https://docs.oracle.com/javaee/7/tutorial/doc/overview003.htm>

Oracle. (2014). *The Java EE 7 Tutorial* . Obtenido de <https://docs.oracle.com/javaee/7/tutorial/doc/overview003.htm>

Piñeiro Gómez, J. M. (2013). *Bases de datos relacionales y modelado de datos*. España: Paraninfo S.A.

Pressman, R. (2010). *Ingeniería del Software*. México: McGraw-Hill.

Pressman, R. (2010). *Ingeniería del Software* . México, México, México: McGraw-Hill,.

Realidad Aumentada. (s.f.). Recuperado el 5 de Mayo de 2016, de Definición .DE: <http://definicion.de/realidad-aumentada/#ixzz48lO3pS9r>

Realidad Virtual. (s.f.). Recuperado el 5 de Mayo de 2016, de Retro Informatica: <http://www.fib.upc.edu/retro-informatica/avui/realitatvirtual.html>

Realidad Virtual. (s.f.). Recuperado el 5 de Mayo de 2016, de <http://sabia.tic.udc.es/gc/Contenidos%20adicionales/trabajos/3D/Realidad%20Virtual/web/definicion.html>

Reinoso, R., Muñoz, J., Carballo, I., Sáenz, J., Nadal, F., Nin, P., . . . Cervera, M. (19 de Agosto de 2011). *Tipos de Realidad Aumentada*. Recuperado el 21 de Mayo de 2016, de [aumentame: http://www.aumenta.me/node/36](http://www.aumenta.me/node/36)

Sprunck, M. (s.f.). *Some Basics about Product Burndown Charts*. Obtenido de <http://www.sw-engineering-candies.com/blog-1/howtouseproduct-burndown-chartsandsprint-burndown-chartsnotonlyinscrumprojects>

Torras, I. (2015). *Que es una Metodología Ágil: Scrum.org*. Obtenido de Scrum: <http://metodologiascrum.readthedocs.io/en/latest/Scrum.html>

Torras, I. (2015). *Scrum.org*. Obtenido de <http://metodologiascrum.readthedocs.io/en/latest/Scrum.html>

Wikipedia. (27 de junio de 2017). *Java (lenguaje de programación)*. Obtenido de [https://es.wikipedia.org/wiki/Java_\(lenguaje_de_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n))
)