



FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

DESARROLLO E IMPLEMENTACIÓN DE UN PROTOTIPO DE UN
SISTEMA PARA REGISTRO Y CONTROL DE EQUIPOS DE
LABORATORIO, MEDIANTE APLICACIONES MÓVIL Y WEB

AUTOR

Juan José Bahamonde Suasnavas

AÑO

2018



FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

DESARROLLO E IMPLEMENTACIÓN DE UN PROTOTIPO DE UN SISTEMA
PARA REGISTRO Y CONTROL DE EQUIPOS DE LABORATORIO,
MEDIANTE APLICACIONES MÓVIL Y WEB

Trabajo de Titulación presentado en conformidad con los requisitos
establecidos para optar por el título de Ingeniero en Electrónica y Redes de
Información

Profesor guía:

MSc. Carlos Andrés Muñoz Cueva

Autor

Juan José Bahamonde Suasnavas

Año

2018

DECLARACIÓN DEL PROFESOR GUÍA

“Declaro haber dirigido el trabajo, Desarrollo e implementación de un prototipo de un sistema para registro y control de equipos de laboratorio, mediante aplicaciones móvil y web, a través de reuniones periódicas con el estudiante Juan José Bahamonde Suasnavas, en el semestre 2018-2, orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación”.

Carlos Andrés Muñoz Cueva
Master en Gerencia de Sistemas
C.I.: 1712981511

DECLARACIÓN DEL PROFESOR CORRECTOR

"Declaro haber revisado este trabajo, Desarrollo e implementación de un prototipo de un sistema para registro y control de equipos de laboratorio, del estudiante Juan José Bahamonde Suasnavas, en el semestre 2018-2, dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación".

Santiago Ramiro Villareal Narvaez

Master Science Technologies, Sante A Finalite Recherche Mention

Informatique Specialite Web Intelligence

C.I.: 1713980074

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

“Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes.”

Juan José Bahamonde Suasnavas

C.I.: 1723308290

AGRADECIMIENTOS

A la Universidad de las Américas por darme todas las facilidades. A mi tutor de tesis, Carlos Muñoz, por su exigencia y enseñanzas; y sobre todo, a mi familia, por su amor y apoyo incondicional.

DEDICATORIA

A Lilo, mi ángel, te extrañare por siempre. A mi abuelita, Lila, mi luz. A mi hermano Oscar, mi ídolo. A mi padre, mi héroe y modelo a seguir. Y a mi madre santa, mi vida y mi todo.

RESUMEN

La Universidad de las Américas cuenta con varios laboratorios en sus diferentes facultades. Los estudiantes pueden hacer uso de estos para su desarrollo académico y profesional. La Facultad de Ingeniería y Ciencias Aplicadas (FICA), específicamente, posee varios laboratorios en los que se realizan actividades con equipos de redes, lo que facilita las prácticas de los estudiantes de las carreras como Redes y Telecomunicaciones, Tecnologías de la Información entre otras. En la actualidad, no se maneja un sistema de registro y control adecuado, tanto del inventario como del uso de los equipos. Por esta razón, se elaboró el prototipo de un sistema que optimizará y facilitará dichas tareas. Se empleó la metodología de desarrollo Scrum para potenciar la agilidad y efectividad en la creación del sistema. Este consiste en una base de datos relacional en el servidor MySQL, en el cual se tendrá el registro de todos los equipos, las actividades que se realicen con ellos y los usuarios responsables de los mismos. Para la administración de la información y generación de reportes se creó una aplicación web, desarrollada en la plataforma ASP.NET. También, se desarrollaron aplicaciones móviles para los sistemas operativos iOS y Android, las cuales tienen como función la creación de actividades, en las cuales se registra la fecha y hora de préstamo y devolución de los equipos utilizados mediante un lector de código de barras. Además, es posible ingresar observaciones durante o fuera de las actividades. Toda esta información puede ser recuperada o enviada a la base de datos mediante *Web Services*. En conjunto forman un sistema eficiente de registro y control de equipos de laboratorio. El prototipo fue implementado en el Data Center experimental de la FICA, mediante una máquina virtual con sistema operativo Windows Server 2012, en la cual se aloja el servidor de base de datos, y de la aplicación y servicios web. Con esto es posible que los docentes utilicen el sistema de registro de equipos (SREQ) por medio de las redes LAN y WLAN del campus Queri.

ABSTRACT

Universidad de las Américas has several laboratories in its different faculties. Students can make use of these for their academic and professional development. The Facultad de Ingeniería y Ciencias Aplicadas (FICA), has certain laboratories in which activities are carried out with network equipment, which facilitates the practices of the students of the careers like Redes y Telecomunicaciones, Tecnologías de la Información, among others. Currently, an adequate registration and control system is not managed, both for the inventory and for the use of the equipment. For this reason, the prototype of a system that will optimize and facilitate said tasks was created. The Scrum software development methodology was used to enhance the agility and effectiveness in the implementation of the system. It consists of a relational database in the MySQL server, which will have the record of all the equipment, the activities carried out with them and the users responsible for them. For the management of information and generation of reports, a web application was created, developed on the ASP.NET platform. Also, mobile applications were developed for the iOS and Android operating systems, which have the function of creating activities, in which the date and time of loan and return of the equipment used are recorded by a bar code reader. In addition, it is possible to enter observations during or outside of the activities. All this information can be retrieved or sent to the database through Web Services. Together, they form an efficient system for the registration and control of laboratory equipment. The prototype was implemented in the experimental Data Center of the FICA, where there is a virtual machine with Windows Server 2012 operating system, in which the database server, the application and web services are hosted. It makes possible for teachers to use the equipment registration system (SREQ) through the LAN and WLAN networks of the Queri campus.

ÍNDICE

INTRODUCCIÓN.....	1
Antecedentes.....	1
Alcance	2
Justificación.....	3
Objetivo General.....	3
Objetivos Específicos	3
1. Marco referencial y herramientas.....	4
1.1 Microsoft .NET.....	4
1.2 Integrated Development Environments, IDEs	17
1.3 MySQL.....	21
1.4 Scrum	22
2. Desarrollo de la metodología.....	38
2.1 Definición del Scrum Team.....	38
2.2 Generación de las historias de usuario	38
2.3 Sprint 0	48
2.4 Sprint Planing y Sprint Backlog	52
2.5 Cronograma de desarrollo	59
2.6 Primer Sprint	59
2.7 Segundo Sprint.....	85
2.8 Tercer Sprint	101
2.9 Cuarto Sprint.....	116
2.10 Quinto Sprint.....	121
2.11 Sexto Sprint	133
2.12 Séptimo Sprint.....	138
2.13 Estimación de costos e implementación.....	142

2.14	Prueba de rendimiento	145
3.	Conclusiones y Recomendaciones	153
3.1	Conclusiones.....	153
3.2	Recomendaciones	154
	Referencias	155
	ANEXOS.....	157

INTRODUCCIÓN

Antecedentes

La Universidad de las Américas facilita a sus estudiantes espacios como laboratorios, talleres, estudios, entre otros; en donde puedan realizar actividades académicas como prácticas, tareas, experimentaciones, investigaciones y demás trabajos. Esto, con el fin de dar apoyo a la labor de los docentes y que los alumnos desarrollen un alto grado de aprendizaje y manejo de tecnologías. Los laboratorios cuentan con una excelente infraestructura y estado de equipos que permiten que los estudiantes puedan llevar a cabo la ejecución de las actividades antes mencionadas.

Entre las facultades que gozan de este tipo de beneficios está la Facultad de Ingeniería y Ciencias Agropecuarias, en la cual se ha integrado, asimismo, laboratorios para todas sus carreras. Es así como los estudiantes pueden trabajar los conocimientos adquiridos en clases y llevarlos a la práctica. Así, se permite obtener una visión más real para su aplicación en el futuro profesional.

Para una mejor formación académica de los estudiantes de las nuevas carreras de Tecnologías de la Información y Redes y Telecomunicaciones, específicamente, se implementó el Data Center experimental, el cual cuenta con dispositivos para almacenar información y proveer servicios de diferentes índoles. De esta manera, este recurso puede ser aprovechado para que los estudiantes propongan iniciativas innovadoras. (Universidad de las Américas, s.f).

A esto, se suma el laboratorio de Redes, en donde podemos encontrar equipos Cisco como *switches*, *routers*, *firewalls*, además de *access point*, cables Ethernet, seriales, etc. El préstamo de estos es regulado por profesores designados que gestionarán y velarán por el uso adecuado de los materiales.

Alcance

El alcance de este proyecto de titulación es desarrollar el prototipo de un sistema de registro y control de equipos prestados, orientado a profesores y encargados de los mismos. Estos pueden ser *switches*, *routers*, *access points*, *firewalls*, fuentes de alimentación PoE, cables seriales, de consola y demás.

Dicho sistema contará con una aplicación móvil para el registro de los equipos entregados. También se elaborará una aplicación web para administrar la base de datos y poder visualizar reportes.

En cuanto a su manejo, se podrá acceder a la aplicación móvil una vez que los profesores inicien sesión con sus credenciales. Después se procederá a registrar el equipo, que tomarán prestado, con la ayuda de un lector de código de barras y QR, debido a que los equipos ya cuentan con un código identificativo. Finalmente, se deberá notificar el inicio y fin de la actividad.

Como ya se mencionó anteriormente, la aplicación web permitirá que el docente encargado visualice el reporte de uso de los equipos que proporcionará la aplicación. Asimismo, de ser necesario se podrá ingresar, eliminar o modificar elementos ya existentes en la base de datos.

Para esto, se implementará un servidor web y de bases de datos en el Data Center experimental, ya establecido en la Universidad. Los servicios serán instalados en una máquina virtual que ejecuta Windows Server, con el uso de IIS y MySQL Server. En cuanto a la aplicación móvil será desarrollada tanto para la plataforma Android como para iOS.

Para cumplir con lo planteado anteriormente, se aplicará el conocimiento adquirido en las materias de programación, bases de datos, redes LAN y WAN, administración de servicios de internet, seminario de redes, certificación de sistemas operativos y calidad de servicio.

Justificación

Actualmente, no se tiene un registro formal y completo de todos los equipos que utilizan los profesores en sus sesiones de práctica, por lo que no se tiene un control minucioso del correcto uso de estos. Esto puede generar malentendidos e incluso pérdidas materiales. Por tal razón, esto es una necesidad latente para la Facultad de Ingeniería y Ciencias Agropecuarias.

Al plantear el prototipo se busca automatizar el sistema para hacerlo más sencillo, rápido y eficaz; tanto para profesores como para los encargados de los equipos. Con un inventario en la base de datos se tendrá constancia de todos los equipos existentes, evitando pérdidas. Además, se podrá asegurar el buen uso del material; verificar su tiempo de uso y quien es el responsable de los equipos en cada actividad realizada.

En conclusión, el sistema se desarrollará en base a la información de las carreras de Ingeniería en Redes y Telecomunicaciones y Tecnologías de la Información, debido a que estas hacen uso de los equipos frecuentemente. El mismo sistema podría escalar y ser útil para el resto de las facultades que lo necesiten y usen el mismo procedimiento, simplemente modificando la información requerida en la base de datos.

Objetivo General

Desarrollar e implementar un prototipo de un sistema para registro y control de equipos de laboratorio, mediante aplicaciones móvil y web.

Objetivos Específicos

- Aplicar la metodología de desarrollo de software Scrum para el desarrollo del prototipo.

- Diseñar e implementar una base de datos para almacenar el inventario, los registros de los equipos de laboratorio y reportes respectivos.
- Realizar pruebas funcionalidad del prototipo.
- Implementar el sistema de registro y control de equipos en el Data Center experimental.

1. Marco referencial y herramientas

1.1 Microsoft .NET

Microsoft .NET provee un ambiente de programación orientada a objetos y otro de ejecución utilizado para el desarrollo de aplicaciones de escritorio o web.

.NET Framework se conforma de dos componentes principales:

- Common Language Runtime o CLR, es una máquina virtual que actúa como motor de ejecución, administrando el código de las aplicaciones que se están ejecutando. Este motor se puede también alojar en servidores como Internet Information Services, IIS o Microsoft SQL Server.
- Biblioteca de clases de .NET Framework, ésta es orientada a objetos y permite realizar tareas tradicionales de programación, como la gestión de cadenas, captura de datos, conexión de bases de datos y acceso a archivos, al igual que desarrollar diferentes tipos de aplicaciones y servicios como aplicaciones Windows Forms, WPF, ASP.NET, entre otros. (Ceballos, 2013, p.3,4).

1.1.1 Plataforma .NET

“Es un conjunto de herramientas de desarrollo y lenguajes de programación, de propósito general, orientados a objetos, de tercera generación, de alto nivel, de compilación a código intermedio, que nos permiten utilizar todos los recursos disponibles en la computadora” (Ramírez, 2012, p.8).

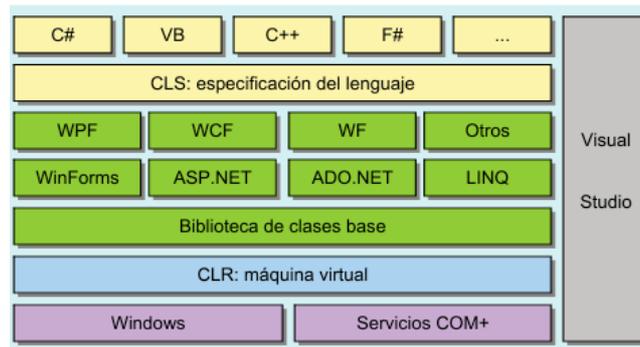


Figura 1. Arquitectura de .NET Framework.

Tomado de Ceballos, 2013, p 6

1.1.1.1 Capa de lenguajes

Esta capa la compone el Common Language Specification, y los lenguajes de programación que son compatibles con el mismo. El CLS es un conjunto de especificaciones que deben tener los lenguajes para ser considerados como lenguaje de .NET. En un inicio Microsoft contaba con C#, Managed C++ y Visual Basic.NET, los cuales cumplían con los requisitos de CLS. En la actualidad se integró J# y otros lenguajes están siendo desarrollados para unirse a .NET. (Ramírez, 2012, p. 9).

1.1.1.2 .NET Framework

En la arquitectura de .NET es la capa conformada por los recursos de .NET y el núcleo de servicios, incluyendo la biblioteca de clases base, los compiladores y los servicios que transforman lo codificado a lenguaje de máquina para los diferentes sistemas operativos.

Servicios

Ramírez (2012, p. 10) señala que lo conforman los servicios que permiten la intercomunicación entre los programas compatibles con .NET, el resto de los

componentes de .NET Framework, y los elementos del sistema operativo. Los dos principales servicios son:

- Windows Application Services, que permite a las aplicaciones comunicarse con el sistema operativo con la ayuda de .NET Framework.
- ASP.NET Application Services, que permite a las aplicaciones comunicarse con el sistema operativo con la ayuda de .NET Framework y además con IIS, el motor de servicios web.

Base Class Library

Conforma un grupo de aproximadamente 6000 clases, las cuales se distribuyen en más de 80 ficheros de la librería DLL. El usuario puede hacer uso de estas librerías al desarrollar sus aplicaciones. El desarrollador solo debe codificar los procedimientos de negocio, el resto será realizado por estas librerías. (Ramírez, 2012, p.10).

1.1.1.3 Common Language Runtime

El CLR es la plataforma donde se ejecutan los programas de .NET. Éste agrupa líneas de comandos que permiten crear código intermedio conocido como assembly. También tiene compiladores Just In Time, los cuales generan el lenguaje de máquina a partir del código assembly o ensamblador. Además, es el encargado del uso de recursos, la gestión de errores y la comunicación con el sistema operativo al momento de la ejecución. (Ramírez, 2012, p. 10).

1.1.1.4 Soporte operativo

Esta capa lo componen los sistemas operativos y sus herramientas, así como productos independientes de back end. Tiene como fin explotar al máximo los

beneficios del desarrollo de aplicaciones en .NET y generar un producto robusto y confiable. (Ramírez, 2012, p.10).

1.1.2 Aplicaciones para Internet

Uno de los modelos tradicionales utilizados era el de cliente-servidor, en el cual mediante una aplicación cliente se accede al servidor. Se realiza una comunicación y se hacen solicitudes al servidor y éste retorna una respuesta al requerimiento solicitado.

Una aplicación del modelo cliente-servidor consta de tres elementos principales: interfaz de usuario, lógica de negocio y administración de datos. Estos componentes se pueden distribuir tanto en el servidor como en el cliente, creando diferentes formas de desarrollar la aplicación obteniendo una arquitectura de software de dos capas.

En la actualidad el componente de negocio también puede ser un servicio, por lo que puede alojarse en cualquier otro servidor, llamado servidor de aplicaciones, lo cual da como resultado una arquitectura de tres capas.



Figura 2. Arquitectura de tres capas.

Tomado de Ceballos, 2013, p 742

Este modelo tres capas se divide en presentación, lógica de negocio y gestión de datos. La primera es el conjunto de elementos que componen la interfaz gráfica, un ejemplo de la capa de presentación es el explorador refiriéndose a

aplicaciones de internet. La capa de negocio se encarga de modelar el comportamiento del sistema dependiendo de la información entregada por la capa de datos y los actualiza si es necesario. También indica qué procesos se deben hacer con los datos. Por último, la capa de datos proporciona el acceso y almacenamiento de los datos. Es un gestor de bases de datos o de objetos. Este tipo de aplicaciones tienen como características principales la escalabilidad y su facilidad en el mantenimiento, gracias a que es modular. (Ceballos, 2013, p. 742).

1.1.2.1 ASP.NET

ASP.NET proporciona un ambiente multicapa y de ejecución distribuida que solventa varios de los problemas con los que se encuentran los desarrolladores que crean sistemas distribuidos a gran escala centrados en la Web, utilizando diferentes lenguajes de programación compatibles con .NET.

Las aplicaciones ASP.NET siempre son compiladas y además .NET Framework permite que las aplicaciones interactúen con el sistema operativo. Estas se implementan gracias a un mecanismo en el cual un explorador pueda acceder a una página web mediante su URL. IIS pone a disposición todos los ficheros y páginas que forman parte de una aplicación ASP.NET. Si ésta contiene código de la lógica de negocio, es ejecutado por .NET Framework, y usa IIS para entregar el resultado al cliente web como código HTML. De otra forma IIS accede al sistema operativo para presentar el resultado en HTML.

Las principales características de ASP.NET según Ceballos (2013, p. 744) son las siguientes:

- Capa de presentación y lógica de negocio separadas.
- Acceso a servicios de .NET Framework.
- Una página se compila al acceder por primera vez y su resultado es guardado para utilizarse en los siguientes accesos esa página.

- Inclusión de diferentes mecanismos que la información del estado de una aplicación para administrarlo.
- Programación de la aplicación con diferentes lenguajes.
- Actualización de ficheros que forman parte de la aplicación.

1.1.2.2 Formularios Web

Los Web Forms forman parte de ASP.NET, permitiendo hacer interfaces de usuario, las cuales son páginas web programables para las aplicaciones web.

Se pueden utilizar controles en el lado del servidor para crear elementos de la interfaz de usuario y programarlos para que hagan los procesos requeridos. Al usar estos formularios se simplifica el desarrollo gracias a las siguientes ventajas:

- Modelo de programación basado en eventos en el servidor.
- Separación entre el formato HTML y la lógica de la aplicación.
- Diseño sin escribir líneas de código utilizando el IDE Visual Studio.
- Compatible con componentes y controles .NET

Un formulario web es ejecutado en el servidor donde se aloja, una vez que lo requiere un cliente o explorador web, donde muestra la interfaz. Al ejecutarse en el servidor evita que se desarrollen versiones de la interfaz para cada explorador, debido a que el estándar HTML lo puede reconocer cualquier navegador. Los controles encapsulan el mecanismo para mantener el estado mientras se hacen los recorridos al servidor de ida y vuelta. (Ceballos, 2013, p.792).

Las páginas web ASP.NET proveen una interfaz de usuario para las aplicaciones web. La plataforma ASP.NET contiene controles y objetos que pueden ser añadidos a la interfaz de usuario y un contexto de ejecución y desarrollo para las aplicaciones en un servidor de aplicaciones web. El código ejecutado en el servidor genera dinámicamente la salida HTML que da lugar a la página web que

se muestra en un navegador. Ceballos (2013, p. 744) señala que estos controles pueden ser de los siguientes tipos:

- Controles de servidor HTML: están relacionados con los elementos HTML que procesan.
- Controles de servidor web: diseñados para proveer funcionalidad a una página web sin importar el explorador que se utilice.
- Controles de validación: asociados a los controles de entrada para comprobar las entradas del usuario.
- Controles de usuario: son creados como formularios web y se adjuntan en otras páginas web ASP.NET. Es una forma fácil de crear elementos reutilizables como menús, barras de herramientas, entre otros

1.1.2.3 Aplicación WEB ASP.NET

La plataforma ASP.NET provee los servicios que se necesitan para administrar y producir formularios web de forma dinámica. Las formas web adicional a otros elementos conforman las aplicaciones web, que son ejecutadas en un servidor web con las peticiones de los usuarios. Visual Studio ofrece un diseñador de web forms, controles, una herramienta de depuración y un editor, con el fin de facilitar la creación de aplicaciones, a las que se podrá acceder desde un explorador una vez que se encuentran alojadas en un servidor.

ASP.NET hace más fácil la creación de aplicaciones web gracias a que proporciona una abstracción de la comunicación cliente-servidor web tradicional la cual permite programar aplicaciones por medio de herramientas de programación orientada a objetos y RAD. También proporciona un modelo unificado que da respuesta a los eventos de los clientes en el código que es ejecutado en el servidor. Y finalmente mantiene de forma automática el estado de la página web, y de sus controles. (Ceballos, 2013, p. 792).

1.1.2.4 Servicios WEB

Los servicios web son elementos que son ejecutados en el servidor y usualmente implementan la capa de lógica de negocio. Muestran una interfaz con la que otras aplicaciones pueden acceder a los servicios proporcionados. Un servicio web está disponible mediante protocolos web, por lo que es compatible con programas que son ejecutados en diferentes lenguajes, dispositivos o sistemas operativos. Es decir, es posible acceder desde cualquier aplicación capaz de interpretar y generar mensajes escritos en SOAP, un protocolo basado en XML, que usa el estándar HTTP para su transporte, y proporciona una comunicación basada en mensajes.

Un servicio web es una aplicación que es instalada en un servidor que ofrece sus servicios a las aplicaciones cliente que lo pidan. Un proveedor de servicios genera un servicio web, para luego instalarlo en un servidor, lo define mediante WSDL y crea un registro de este en el directorio de servicios. Un usuario hace una aplicación cliente que hace una petición de un servicio determinado, localiza el servicio web en el directorio de servicios, vincula la aplicación con el servicio y la aplicación se comunica con el servicio utilizando mensajes SOAP e intercambiando información en XML. (Ceballos, 2013, p. 889).

1.1.3 Internet Information Server

El motor de servicios web, IIS provee una plataforma confiable, fácil de gestionar, modular y escalable donde se pueden alojar aplicaciones, servicios y páginas web, de forma segura.

Es posible compartir datos con usuarios en intranet, extranet o en el internet. IIS 8 es una plataforma que la conforman servicios de PHP, FTP y WCF, además de IIS y ASP.NET. Según la página oficial de IIS de Microsoft (s.f a). dentro de las ventajas se tienen las siguientes:

- La seguridad web mejorada gracias al aislamiento automático de aplicaciones.
- Un mismo servidor soporta aplicaciones web de ASP.NET, PHP y ASP.
- El aislamiento de aplicaciones es conseguido al proporcionar una identidad única a los procesos de trabajo
- Se puede agregar, reemplazar y eliminar componentes IIS integrados para que puedan adaptarse a las necesidades del cliente.
- Al usar una mejor compresión y almacenamiento caché dinámico se incrementa la velocidad del sitio web.

Cuando se realizan páginas web con ASP.NET deben ser traducidas a código HTML para que se puedan acceder a ellas mediante el cliente web. Esta labor es realizada por IIS para que los usuarios puedan visualizar el contenido de la aplicación web. En el caso de que la aplicación no cuente con código en la lógica de negocio para que IIS realice esta acción, éste accede al sistema operativo para desplegar el resultado en HTML.

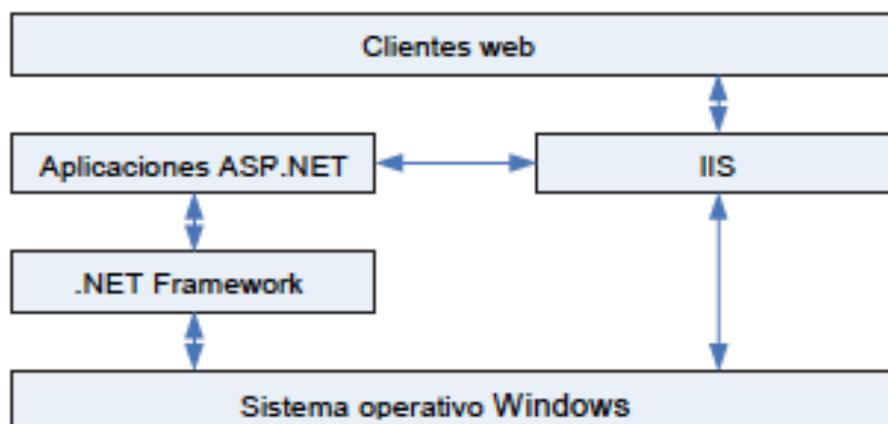


Figura 3. Función de IIS.

Tomado de Ceballos, 2013, p 743

1.1.3.1 Publicación de una aplicación web

Una vez que se instala IIS en el sistema operativo se crea automáticamente la ruta C: \inetpub\wwwroot, que es donde se ubicarán los directorios de las aplicaciones. Una buena práctica es crear una nueva carpeta o directorio para cada aplicación, pero dicho directorio debe ser un directorio virtual con un vínculo al directorio físico del sistema de archivos que posee la aplicación web. Éste es utilizado para incluir el contenido de la aplicación en IIS, sin la necesidad de copiar o mover los archivos desde el directorio físico al directorio raíz del servidor de aplicaciones web.

IIS cuenta con herramientas administrativas para crear el directorio virtual, donde simplemente se indica la dirección física de los archivos de la aplicación web. Muchas veces se suele hacer ambas cosas, copiar o mover el contenido dentro del directorio raíz y marcar esa misma dirección como directorio virtual.

Al usar directorios virtuales se pueden publicar aplicaciones web que no se encuentren en la carpeta raíz del servidor web o en un equipo remoto. Esta es una de las funciones que necesitan operaciones de administración del servidor, proporcionadas por el complemento Internet Information Services, el cual cuenta con un conjunto de servicios para Windows y un servidor web. También permite controlar el contenido del sitio web y el acceso al mismo. (Ceballos, 2013, p.759, 760).

Finalmente, una aplicación debe ser convertida en una aplicación de IIS. Como requisito debe formar parte de un sitio web y debe ser asignada a un grupo de aplicaciones para que tenga recursos reservados para que pueda funcionar correctamente, y así optimizar el rendimiento del servidor. Una vez que se han realizado estos pasos la aplicación ya ha sido publicada y se puede acceder a ella con su URL.

Arquitectura de IIS

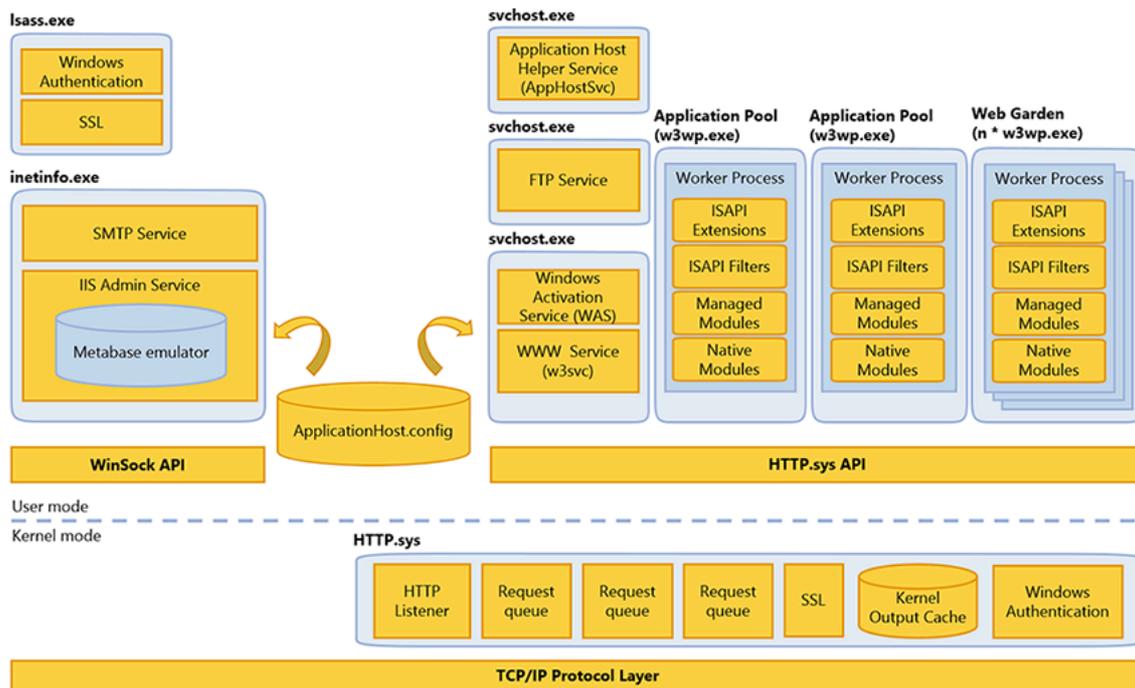


Figura 4. Arquitectura de IIS 8.

Tomado de Marín, 2014

IIS posee varios componentes que hacen importantes funciones tanto para la aplicación como para el servidor Web. Entre ellos se tienen los Protocol Listeners como HTTP.sys y los servicios como el WAS y WWW Service.

Los *protocol listeners* reciben una petición de un protocolo específico, lo envía a IIS para que sea procesado y devuelve la respuesta al cliente que lo solicitó. Por defecto, el *protocol listener* que da IIS es HTTP.sys, que escucha las peticiones HTTP y HTTPS.

El WWW Service lee la información de la configuración de la metabase de IIS y la usa para configurar y actualizar el HTTP.sys. Y notifica al WAS cuando una petición entra en cola. Además, inicia, para, monitorea y gestiona los work processes que producen las peticiones HTTP.

WAS o Windows Process Activation Service gestiona los grupos de aplicaciones y work processes de las peticiones HTTP y non-HTTP. Cuando un Protocol

Listener recoge una petición del cliente, WAS determina si un work process está ejecutándose o no. Si lo está haciendo pasa la solicitud al work process para su procesamiento y luego envía la respuesta al cliente y al HTTP.sys. Si no lo está haciendo, WWW Service hace una petición al WAS para que inicialice y configure un nuevo work process. (Marín, 2014).

1.1.4 Modelo de desarrollo Multicapa

El modelo multicapa, tiene como fin en diferenciar las operaciones de base de datos, de lógica de negocios, y de presentación, para que trabajen juntas y así llegar al resultado esperado. El aislamiento de las operaciones es beneficioso, debido a que de esta manera los profesionales pueden trabajar sobre su capa respectiva de una manera más eficiente y de manera simultánea sobre una misma aplicación. Lo ideal es que cada capa se aloje en un servidor diferente, con el fin de distribuir la carga de las operaciones del aplicativo entre las distintas capas. (Ramírez, 2012, p. 213).

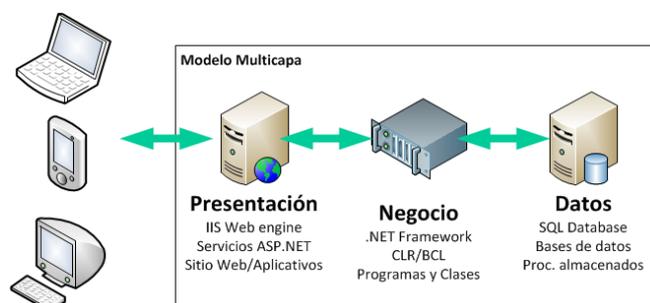


Figura 5. Modelo multicapa.

Tomado de Ramírez, 2012, p 213

1.1.4.1 Capa de Presentación

Es la capa que proporciona las interfaces gráficas con las que el usuario final puede interactuar con la aplicación. Provee los servicios al usuario final y recopila

datos robustos y confiables para que las otras capas puedan manejar las funciones de procesamiento, explotación de datos y almacenamiento sin problemas. Ramirez (2012, p. 214) señala que las aplicaciones web constan de lo siguiente:

1. Un Servidor que tiene instalado IIS, el motor de ejecución Web de Microsoft que permite recibir peticiones HTTP y responderlas.
2. ASP.NET Web Services habilitado por .NET Framework, controla el *rendering* y el *parsing* de las páginas web, gestionan los estados, manejan el caché, y soportan el diálogo entre páginas.
3. Las aplicaciones web, conformadas por páginas .aspx, web.config, archivos complementarios, *master*, entre otros.

1.1.4.2 Capa de Negocio

La capa de negocio proporciona los procesos y la lógica de negocio de la aplicación. Esta capa realiza todos los procesos que son solicitados por la capa de presentación, toma en cuenta los datos previstos por la capa de presentación, y se encarga de retornar los resultados a la capa de negocio, luego de haber hecho las tareas solicitadas en la capa de datos. Para aplicaciones web consta de lo siguiente:

1. Un servidor que tiene instalado la plataforma .NET Framework, que habilita los Windows Forms Services, que se usan para interactuar con las API de Windows, y conseguir el control de la utilización y de los recursos del servidor.
2. .NET Framework proporciona el modelo de ejecución, CLR, la compilación JIT, y el acceso a las clases de BCL, lo que aprovecha todas las funcionalidades de los lenguajes de .NET.

3. Las librerías DLL que tienen la programación de lógica de negocios, programada por los desarrolladores del proyecto, que realizan funciones de escritura, lectura y actualización de datos, además del vínculo con la base de datos. (Ramírez, 2012, p.214).

1.1.4.3 Capa de Datos

Esta capa proporciona los servicios del administrador de bases de datos. Recibe peticiones de escritura y lectura de datos, debe hacerlas de forma eficaz, confiable e integrada. Hace la lectura y procesamiento de datos usando arquitecturas especializadas y modelos optimizados. Para aplicaciones web consta de lo siguiente:

1. Un servidor con el motor de bases de datos SQL Server instalado, el que provee varios servicios, entre ellos Database Engine y Analysis Services.
2. Proporciona los objetos de base de datos como relaciones, tablas, vistas, entre otros, y los datos contenidos en los objetos, que se encuentran disponibles gracias al adaptador de datos y el Data Provider.
3. Posee soporte a programación, debido a que cuenta con un intérprete de instrucciones Transact SQL, en caso de que se codifique manualmente, y también tiene y ejecuta procedimientos almacenados que proporcionan un mejor desempeño. (Ramírez, 2012, p.215).

1.2 Integrated Development Environments, IDEs

1.2.1 Visual Studio

NET Framework en conjunto con otros elementos de desarrollo, como la biblioteca .NET, el compilador C#, ASP.NET y ADO.NET, conforman el paquete de desarrollo Microsoft Visual Studio que es utilizado para desarrollar

aplicaciones web y aplicaciones de Windows. Este paquete cuenta con diseñadores de interfaces de usuario, editor de código avanzado, depurador, entre otras utilidades que aportan al desarrollo eficaz y eficiente de aplicaciones. (Ramírez, 2012, p. XXIII).

Dentro de sus tareas más importantes según la documentación de Visual Studio (Microsoft, s.f b) se tiene las siguientes:

- El desarrollo de aplicaciones, donde se puede escribir y administrar el código con un editor de código. Dentro de los lenguajes que maneja Visual Studio está C#, Visual Basic, C + +, F#, JavaScript, R y TypeScript.
- La compilación y generación de código fuente, la cual crea aplicaciones y ensamblados ejecutables en cualquier momento del ciclo de desarrollo.
- Depuración, con la cual se visualiza el comportamiento al momento de ejecutarse y se pueden encontrar problemas para después solucionarlos. El proceso de depuración puede interrumpir la ejecución de un programa para examinar el código, modificar variables, visualizar registros, etc.
- Se pueden realizar procesos de pruebas para mantener en el código altos estándares de excelencia. Se pueden hacer pruebas unitarias que facilitan el análisis de los métodos de clases de proyectos para buscar en ellos errores lógicos con mucha más rapidez.
- Se puede realizar la publicación de las aplicaciones ASP.NET
- Controlar las versiones de un programa en Visual Studio, esto gracias a que el IDE realiza instantáneas de los archivos en un tiempo determinado.
- Creación de perfiles, con los cuales se puede diagnosticar tanto problemas a nivel de aplicación como el uso de memoria y del CPU.
- Conexión a cualquier base de datos de manera pública, privada, de forma local, en la nube, etc.

C#

Es un lenguaje de programación orientado a objetos compatible con CLS de .NET Framework que permite desarrollar una serie de aplicaciones robustas y

confiables. Actualmente es uno de los lenguajes de programación más utilizados para el desarrollo de aplicaciones web, gracias a los beneficios de la biblioteca de .NET

Permite a los desarrolladores la implementación de aplicaciones complejas en poco tiempo y con más facilidad, tomado como base las ventajas de Visual Basic y el control y potencia que da C++. Gracias a esto se tienen las siguientes ventajas:

- Cuenta con un Garbage Collector que evita el manejo manual de la memoria por parte de los programadores.
- Los objetos como matrices que son creados dinámicamente son inicializados a cero.
- Permite ver a los sistemas de tipos como un objeto en el lenguaje, debido a que los unifica.

1.2.2 Android Studio

Es el entorno de desarrollo integrado o IDE oficial para Android, el cual tiene las herramientas más apropiadas para crear aplicaciones para todos los dispositivos Android como tablets, smartphones, smartwatches, etc. Está basado en IntelliJ IDEA, según la página oficial de Android Studio (Android Studio, s.f a), este IDE cuenta con un potente editor de códigos y herramientas de desarrollo de IntelliJ, ofreciendo funciones que mejoran la productividad de *apps* como las siguientes:

- Sistema flexible de compilación basado en Gradle que permite configurar fácilmente el proyecto para que integre bibliotecas de códigos y genera distintas variantes de compilación desde un mismo proyecto. Al usar Gradle se tiene una automatización de compilaciones de alto rendimiento, configuraciones personalizables de la compilación y una administración de dependencias sólida. (Android Studio, s.f b).

- Emulador multifunción y cargado de funciones, que permiten instalar y ejecutar las aplicaciones mucho más rápido en los dispositivos físicos.
- Entorno unificado para desarrollo en todas las plataformas Android como smartphones, smartwatches, tablets, Android Auto, y Android TV.
- Instant Run que permite realizar cambios mientras corre una aplicación sin tener que compilar un APK nuevo, por lo que la edición, ejecución y compilación se la hace mucho más rápida.
- Plantillas de código y GitHub integradas para facilitar la compilación de funciones. Gracias a los asistentes de proyectos hace que agregar código en un proyecto sea mucho más sencillo.
- Frameworks y herramientas de prueba.
- Herramientas Lint encontrar problemas de usabilidad, compatibilidad, rendimiento, etc. Estas verifican los archivos de origen de los proyectos Android para detectar errores y posibles mejoras de optimización en cuanto a la seguridad, la corrección, el rendimiento, la internacionalización y la accesibilidad. (Android Studio, s.f b).

1.2.3 Xcode

Xcode es el IDE para el desarrollo de aplicaciones de Apple, tiene integrado los frameworks de Cocoa y Cocoa Touch, por lo que se pueden hacer aplicaciones tanto para computadores Mac, iPad, Apple Watch, iPhone y Apple TV. Tiene la característica de hacer el trabajo mucho más natural, debido a que al desarrollar la interfaz de una aplicación su Assistant Editor presenta las fuentes de código relacionadas a lo que se está haciendo permitiendo crear una conexión entre la interfaz y el código fuente. (Apple Inc., 2018).

Cuenta además con una Interface Builder que permite realizar prototipos de interfaces de usuario sin tener que escribir código. Esta interfaz es fácilmente enlazada gráficamente con el código fuente de Xcode Editor, estableciendo ventanas, sliders y botones para crear una interfaz de usuario completamente funcional. Se puede trabajar al mismo tiempo en la interfaz gráfica y en el código fuente gracias al Assistant Editor que proporciona una vista dividida, mostrando los dos elementos.

Posee con un Version Editor que permite comparar las versiones de los archivos, visualizar que usuarios realizaron los cambios, etc. Se muestran las versiones a comparar al mismo tiempo, resaltando los cambios. También tiene un Source Control que facilita la realización de operaciones de separación y unión para equipos en los que se distribuye el trabajo. (Apple Inc., 2018).

Gracias a la utilización de esquemas se puede personalizar la forma en la que se ejecutan y compilan las aplicaciones, dependiendo de la tarea que se esté haciendo como *debugging*, análisis de código, *profiling*, etc. En Xcode el desarrollo basado en pruebas es manejado por el Test Navigator, que permite ejecutar cualquier tipo de prueba individual o grupo de pruebas en el proyecto. El Assistant Editor puede rastrear automáticamente qué pruebas están empleando el código que se está editando, manteniendo de esa manera el código y las pruebas sincronizadas. (Apple Inc., 2018).

1.3 MySQL

Según la página principal de Oracle MySQL (Oracle, s.f) MySQL es la base de datos open source más popular del mundo gracias a su confiabilidad, facilidad de uso y rendimiento. Es la principal opción al momento de realizar aplicaciones basadas en la web y además es una alternativa como base de datos integrada.

El MySQL Workbench es una herramienta para arquitectos, desarrolladores y DBA de bases de datos. Proporciona modelamiento de datos, desarrollo SQL y

herramientas para la administración de usuarios, respaldos, configuraciones del servidor, etc.

Es posible diseñar, modelar, generar y gestionar bases de datos de forma visual gracias al MySQL Workbench. Tiene los requisitos necesarios para crear un modelo ER complejo, ingeniería directa y reversa y características clave para realizar gestión de cambios y tareas de documentación. De la misma manera cuenta con herramientas visuales para crear, ejecutar y optimizar consultas SQL. Una de ellas es el SQL Editor que provee sintaxis de resaltado por colores, autocompletado, etc. El Database Connections Panel permite gestionar las conexiones de bases de datos. Por último, el Object Browser provee acceso instantáneo a los objetos y esquemas de la base de datos. (Oracle Corporation, 2018).

En cuanto a la administración, se tiene una consola visual para manejar los ambientes de MySQL y obtener una mejor visibilidad de las bases de datos. Con esta herramienta se pueden configurar servidores, administrar usuarios, realizar respaldos, recuperación de datos, auditar los datos y ver el estado de la base de datos. Posee también un tablero para visualizar el rendimiento de la base de datos, además de un grupo de herramientas para mejorar el funcionamiento de las aplicaciones MySQL. Otra característica importante es la migración de bases de datos la cual puede realizarse desde MS SQL Server, MS Access, PostgreSQL, entre otros, además de la conversión de aplicaciones para diferentes plataformas. (Oracle Corporation, 2018).

1.4 Scrum

Es un marco de trabajo de procesos, con el que las personas pueden englobar problemas complicados, y al mismo tiempo entregar productos con el máximo valor creativo y productivo. En este marco se pueden emplear varias técnicas y procesos. Scrum demuestra la efectividad de las técnicas de trabajo y de gestión de producto de tal forma que es posible mejorar el producto, el entorno de trabajo

y el equipo continuamente. Scrum cuenta con reglas que relacionan los eventos, artefactos y roles que manejan las relaciones entre ellos.

Scrum ha sido utilizado para desarrollar hardware, software, redes interactivas, automóviles autónomos, gobiernos, programas de gestión para la operación de organizaciones y muchas de las cosas de la vida cotidiana. La utilidad de Scrum está a prueba diariamente debido a la complejidad del mercado, del entorno, la tecnología y sus interacciones, las cuales incrementan rápidamente. En especial Scrum es efectivo para transferencia incremental e iterativa de conocimiento.

La base de Scrum es un equipo pequeño de personas que tiene como características la adaptabilidad y la flexibilidad, las cuales se mantienen operando en uno o varios equipos que operan, desarrollan, liberan, y continúan el trabajo y los productos del trabajo de muchas personas que colaboran mediante ambientes finales de liberación y arquitecturas de desarrollo sofisticadas. (Schwaber y Sutherland, 2017, p. 4).

Según Monte (2016, p. 22, 23) las premisas de Scrum son las siguientes:

- Satisfacción del usuario final
- Flexibilidad frente a cambios de requerimientos
- Trabajo enfocado en el producto
- Desarrollo sustentable
- Participación abierta entre desarrollo y negocio.
- Intercomunicación entre personas
- Participantes motivados e integrados
- Orientado a la excelencia
- Simplicidad
- Equipos organizados
- Adaptabilidad

1.4.1 Valores Scrum

Los valores de Scrum son cuatro:

- **Compromiso:** Necesario para el trabajo en equipo, entregarse al proyecto.
- **Enfoque:** Segmentar el problema en pequeñas partes que permitan centrarse en la solución de un único problema.
- **Organización abierta:** Comunicación y apoyo entre las personas que conforman el equipo de trabajo.
- **Respeto:** con el trabajo en equipo y el compromiso se respeta el trabajo de cada uno de los integrantes.
- **Coraje:** Ayuda a afrontar los retos que se presentan en proyectos complejos con la ayuda del trabajo en equipo y el respeto.

Además de los valores, Scrum tiene tres pilares principales según Monte (2016, p. 21):

- **Transparencia:** los elementos significativos del proceso tienen que ser conocidos por todos los participantes, por lo que deben estar definidos mediante un estándar común.
- **Inspección:** Los integrantes deben evaluar continuamente el proceso y los resultados para hallar posibles desviaciones lo más rápido posible.
- **Adaptación:** Se deben adoptar de acciones ayuden a corregir una desviación detectada.

1.4.2 Ciclo de vida de Scrum

El ciclo de un proyecto Scrum comienza una vez que el Product Owner informa y crea el Product Backlog, el cual está conformado por historias de usuario que muestran una funcionalidad o requerimiento del producto. El Product Owner

tiene la autoridad de priorizar dichas historias de usuario para formar una lista ordenada.

El equipo Scrum toma las decisiones respecto a las funcionalidades del Product Backlog para cada Sprint, para luego juntarse al Sprint Backlog actual y entonces el Development Team separa las historias de usuario en tareas con el esfuerzo respectivo que tomará cada una de ellas. Desde ese instante, en lo que dure el Sprint, se realizan reuniones diarias conocidas como Daily Scrum, donde el equipo se organizará internamente para luego avanzar con el desarrollo de las tareas.

El proceso termina con el Sprint Review, una reunión donde se presentan los resultados al usuario, quien decidirá si se acepta o no el producto. Por último, el Scrum Master y el Development Team se reúnen para verificar aspectos que se deben mejorar en el proceso interno y luego comenzar un nuevo Sprint, a esta reunión se la conoce como Scrum Retrospective. (Monte, 2016, p. 49, 50).

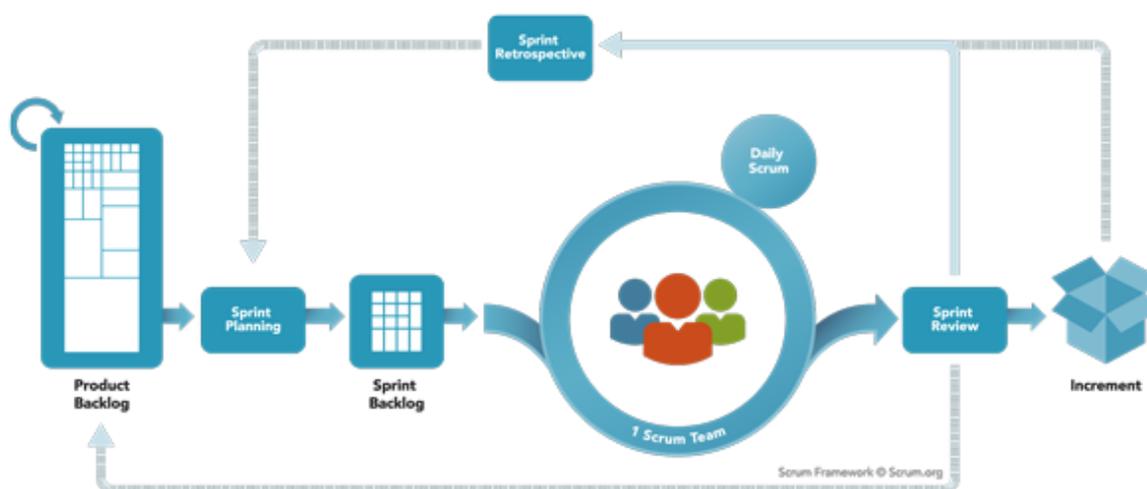


Figura 6. Marco de trabajo de Scrum.

Tomado de Schwaber y Sutherland, 2017, p. 4.

1.4.3 Scrum Team

El equipo de Scrum permanece organizado para elegir la mejor manera de realizar su trabajo y es dirigido por personas dentro del equipo. Deben ser competentes al hacer el trabajo para que no dependan de personas que no sean parte del equipo.

El equipo debe poder optimizar la creatividad, la productividad y la flexibilidad, con el fin de entregar productos de manera incremental e iterativa, generando más oportunidades de retroalimentación. (Schwaber y Sutherland, 2017, p. 6)

1.4.3.1 Product Owner

Es una única persona encargada de incrementar el valor del producto que realiza el Development Team. Es necesario que toda la organización respete las decisiones del dueño del producto para que haga bien su trabajo. Estas decisiones se ven reflejadas en la priorización y en el contenido del Product Backlog. Es el enlace directo con el usuario por lo que debe tener conocimiento del negocio.

Según Schwaber y Sutherland (2017, p. 6) el PO es la única persona a cargo de gestionar el Product Backlog. Esta tarea incluye lo siguiente:

- Expresar los elementos del Product Backlog
- Alcanzar misiones y objetivos de la mejor forma posible ordenando los elementos del Product Backlog
- Optimizar el valor de las actividades que hace el Development Team
- Cerciorarse de que el Product Backlog sea transparente para todo el equipo y que muestre el siguiente trabajo del equipo.
- Asegurar que el Development Team comprenda los elementos del Product Backlog a un nivel necesario.

1.4.3.2 Development Team

Este equipo cuenta con profesionales que hacen su trabajo con el fin de proporcionar un Incremento de producto “terminado” que pueda utilizarse en producción al finalizar cada Sprint. Deben tener la capacidad de organizar y gestionar su propio trabajo. En el Sprint Review es obligatorio un Incremento “terminado” y solo los miembros de este equipo de desarrollo pueden crear el Incremento.

Monte (2016, p. 53) explica que el Development Team tiene las siguientes características:

- Flexibilidad: las personas pueden ocupar distintos roles en el equipo.
- Autoorganización: Ninguna persona que no sea del equipo puede indicar como se debe hacer un Incremento, el mismo equipo debe definir su método de trabajo y los roles.
- Multifuncional: Todo el equipo debe tener las capacidades para garantizar la ejecución del proyecto.

1.4.3.3 Scrum Master

El Scrum Master es un líder que tiene como responsabilidad apoyar y promover Scrum, mediante las capacitaciones todo el Scrum Team de valores, reglas, teoría y prácticas. También ayuda a personas que no pertenezcan al equipo a comprender las interrelaciones que son útiles con el equipo Scrum.

De acuerdo con Schwaber y Sutherland (2017, p. 8) el Scrum Master proporciona varios servicios al equipo de desarrollo, al dueño del producto y a la organización, entre los más importantes se tiene:

Al Product Owner:

- Asegurar el entendimiento del dominio, el alcance y los objetivos del producto por parte del equipo de Scrum
- Generar técnicas de gestión efectivas del Product Backlog
- Demostrar la necesidad de contar con elementos de forma clara en el Product Backlog
- Comprender la planificación del producto en de forma práctica
- Asegurar que el PO pueda ordenar el Product Backlog correctamente para incrementar el valor
- Practicar y entender la agilidad
- Facilitar los eventos Scrum necesarios

Al Development Team:

- Ayudar a conseguir la autoorganización y la multifuncionalidad
- Ayudar a generar productos con un valor alto
- Eliminar limitaciones para el progreso del equipo
- Facilitar los eventos Scrum necesarios

A la organización:

- Guiar y liderar la incorporación de Scrum
- Planear la ejecución de Scrum
- Promover cambios para maximizar la productividad del Scrum Team.
- Cooperar con otros Scrum Master para maximizar la efectividad de la integración de Scrum en la organización

1.4.3.4 Stakeholders

Son las personas que reciben el producto acabado, por lo que son ellos quien lo aceptan o no. Obligatoriamente deben asistir a los Sprint Reviews. Además, son los responsables de detallar criterios de aceptación para cada una de las

funcionalidades del Product Backlog y de dar información y resolver dudas que tenga el Product Owner, incluso por parte del equipo de desarrollo. (Monte, 20016, p. 54,55).

1.4.4 Artefactos de Scrum

Según Schwaber y Sutherland (2017, p. 15) “Los artefactos definidos por Scrum están diseñados específicamente para maximizar la transparencia de la información clave, necesaria para asegurar que todos tengan el mismo entendimiento del artefacto.”

1.4.4.1 Product Backlog

Es una lista ordenada que contiene lo necesario para el producto y la única referencia de requerimientos en el caso que se deba hacer un cambio en el producto. El Product Backlog o lista de producto no se completa en absoluto, debido a que si un producto existe la lista también lo hace. Esta lista evoluciona a medida que el ambiente en el que se lo utilizará y el producto lo hacen, por lo que puede cambiar continuamente para reconocer lo que necesita el producto para que sea útil, competitivo y apropiado.

El Product Backlog enlista todas las funcionalidades, mejoras, correcciones y características que conforman los cambios que deben realizarse en el producto para futuras entregas. Los atributos de esta lista son el valor, la descripción, el orden y la estimación. El refinamiento del Product Backlog es la acción de incluir orden, detalle y estimaciones a los elementos de la lista. Es un proceso constante donde el PO y el DT se examinan, refinan y cooperan con especificaciones de los elementos de la lista. Los elementos del Product Backlog de mayor orden generalmente son más detallados que los de menos orden. Los elementos en los que trabaja el DT tienen mayor granularidad para que cada parte pueda ser “terminado” en un tiempo determinado. (Schwaber y Sutherland, 2017, p. 15)

Se puede sumar el trabajo total que resta para conseguir el objetivo, el PO analiza este trabajo y lo compara con Sprint Reviews anteriores, para evaluar el progreso necesario para finalizar el trabajo planeado en el tiempo esperado para el objetivo.

Historias de usuario

Las historias de usuario son descripciones de las funciones que tendrá el software. Son el resultado de la cooperación entre el cliente y el Scrum Team, y éstas evolucionarán durante la vida del proyecto.

Se componen de los siguientes elementos:

- *Card*: es una descripción breve que sirve de recordatorio.
- *Conversation*: Es una conversación cuya función es asegurarse de que todo se haya entendido correctamente y definir el objetivo.
- *Confirmation*: Pruebas funcionales para fijar detalles que sean destacados y fijar cuál será su límite.

Tabla 1.

Anverso de la historia de usuario

Título		Identificación	
Descripción			
Notas			
Story Points		Prioridad (1-5)	
ID user story padre		Roles	

Adaptado de Monte (2016, p. 116).

- Título: nombre de la historia de usuario, puede ser “Pantalla de inicio”
- Identificación: conjunto de caracteres único, diferente al resto de historias.
- Descripción: Descripción de la historia de usuario.
- Notas: Información importante para el DT.
- Story points: valoración de la historia en cuanto al esfuerzo necesario.
- Prioridad: es prioridad respecto a las demás historias determinadas en el Product Backlog. Se mide con una valoración o utilizando un gradiente.
- Riesgo: el riesgo es opcional que muestra la complejidad del desarrollo al DT.
- ID user story padre: se utiliza en el caso que sean divididas en partes más pequeñas para identificar el padre.
- Rol(es): Determina los roles de los usuarios principales que se responsabilizarán de la funcionalidad mostrada en la historia de usuario al momento de estar en producción. (Monte, 2016, p. 117,118).

Tabla 2.

Reverso de la historia de usuario

Criterios de aceptación			
Notas			
Número de tareas			
Estimación de horas en tareas		Horas reales de la tarea	

Adaptado de Monte (2016, p. 116).

- Criterios de aceptación: definición de los criterios de aceptación que deben conseguirse para marcar esta historia como “terminada”.
- Notas: Información sobre la calidad y la aceptación que sea relevante para el DT.

- Número de tareas: Número de tareas en las que se dividió una historia durante el Sprint.
- Estimación del tiempo de las tareas: Suma de horas de las estimaciones iniciales de todas las tareas
- Horas reales de las tareas: Tiempo que tomo llevar a cabo todas las tareas de la historia.

Una historia de usuario debe estar estimada y priorizada siempre. En el Sprint 0 ya se hace una primera estimación y priorización de toda la lista de producto. Después, en cada Sprint, esta valoración y priorización puede variar, conforme se va incorporando más información en cada historia de usuario por parte del Development.

1.4.4.2 Sprint Backlog

Esta lista es elaborada durante el Sprint Planning, donde se asignan tareas a cada una de las personas y se determina el tiempo para terminarlas. Así, el proyecto se divide en unidades más pequeñas y es posible ver o determinar las tareas donde no se está avanzando como se espera y se intenta erradicar los problemas.

El Sprint Backlog está ordenado por prioridades para el usuario, donde puede haber dependencias entre tareas, por lo que se las debe diferenciar de una forma u otra. El formato de la lista puede ser en pizarras, herramientas colaborativas u hojas de cálculo. Se debe incluir las listas de tareas asignando una persona responsable, determinar su estado y el tiempo para finalizar cada una de ellas. También debe mostrar una referencia diaria del tiempo restante por cada tarea. (Trigas, s.f, p. 40).

1.4.4.3 Incremento

Según La Guía de Scrum de Schwaber y Sutherland (2017, p. 15) “El Incremento es la suma de todos los elementos de la Lista de Producto completados durante un Sprint y el valor de los incrementos de todos los Sprints anteriores.” Para que un Sprint se considere como finalizado debe contar con un nuevo incremento, el cual debe ser completamente funcional y aceptado por el dueño del producto. De esta forma se puede seguir con el proceso.

1.4.5 Eventos de Scrum

Son eventos que están predefinidos para lograr estabilidad y reducir reuniones que estén definidas en Scrum. Los eventos son bloques de tiempo o time-box por lo que cuentan con un límite de duración. Al comenzar un Sprint se fija su duración y no puede ser alterada. Todos los eventos de Scrum comprenden una oportunidad formal para la revisión y ajuste de algún aspecto. (Schwaber y Sutherland, 2017, p. 9).

1.4.5.1 Sprint

Es el evento más importante de Scrum, y consiste en un time-box de máximo un mes, donde se produce un Incremento de producto “terminado” potencialmente despegable.

El Sprint está conformado por los eventos Sprint Planning, Daily Scrums, Sprint Review y Sprint Retrospective, además del trabajo de desarrollo. En esta etapa no se deben hacer cambios que afecten al Sprint Goal, ni disminuir los objetivos de calidad. También se puede renegociar el alcance entre el PO y el DT una vez que se aprenda más. Un Sprint debe tener un diseño, plan de construcción flexible, trabajo en equipo, el incremento, y una meta de lo que se va a construir. Al finalizar un Sprint se debe comenzar inmediatamente con un Sprint nuevo. (Schwaber y Sutherland, 2017, p. 9).

Un Sprint puede ser cancelado únicamente por el Product Owner si el Sprint Goal queda obsoleto, y debe cancelarse antes de que se acabe el time-box. Al cancelarlo se deben revisar los elementos del Product Backlog que hayan sido “terminados” y se acepta el trabajo que sea potencialmente entregable, de otro modo los elementos incompletos se vuelven a estimar e ingresan nuevamente al Product Backlog.

1.4.5.2 Sprint Planning

En este evento se planifica detalladamente el Sprint. Entre sus funciones está definir la funcionalidad que se desarrollará, solventar dudas, crear las historias de usuario y dividir las en tareas con su respectivo esfuerzo, establecer los criterios de aprobación del Sprint y de cada historia de usuario. Esta actividad se la debe realizar al inicio de cada Sprint. También se puede hacer una revaloración del esfuerzo de las funcionalidades integradas en el sprint. Si la revaloración es leve la puede asumir el DT, si es que no se debe renegociar las funcionalidades al sprint con el PO. (Monte, 2016, p. 79).

La planeación del Sprint responde a dos preguntas fundamentales:

- ¿Qué se puede hacer en este Sprint?

El Development Team plantea la funcionalidad que será desarrollada en el Sprint. El dueño del producto analiza el objetivo que el Sprint debe alcanzar y los elementos del Product Backlog, si estos son completados, se cumple el Sprint Goal. Todo el Scrum Team aporta en la comprensión del trabajo del Sprint y define el Sprint Goal.

La entrada a esta reunión está conformada por el Product Backlog, el último incremento de producto, el rendimiento pasado y la capacidad proyectada del Development Team para el Sprint.

- ¿Cómo completar el trabajo elegido?

El DT define cómo construir una funcionalidad para generar un Incremento de producto “terminado” en lo que dura el Sprint. Los elementos del Product Backlog elegidos para el Sprint y el plan para finalizarlos, se conoce como Sprint Backlog. El Development Team empieza diseñando el sistema y el trabajo que se necesita para convertir el Product Backlog en un Incremento de producto funcional. El Product Owner puede ayudar a explicar los elementos del Product Backlog seleccionados y hacer aprobaciones. Al finalizar el Sprint Planning, el DT debe tener la capacidad de explicar al PO y al Scrum Master cómo aspira trabajar como un equipo autoorganizado para alcanzar el Sprint Goal y producir el Incremento deseado. (Schwaber y Sutherland, 2017, p. 11).

1.4.5.3 Sprint Goal

Es una meta que se establece para el Sprint que pueda lograrse por medio de la implementación del Product Backlog. Es definido en el Sprint Planning y provee una guía al Development Team referente a por qué se está creando el incremento, al igual que cierta flexibilidad en cuanto a la funcionalidad ejecutada en el Sprint. El objetivo del Sprint pueden ser elementos del Product Backlog que proveen una función coherente.

El Sprint Goal puede representar otro vínculo de unión que haga que el DT trabaje en equipo y con iniciativas separadas, manteniendo en mente el objetivo del Sprint. El equipo de desarrollo implementa tecnología y funcionalidades para poder cumplir el Sprint Goal. (Schwaber y Sutherland, 2017, p. 12).

1.4.5.4 Daily Scrum

El Daily Scrum es una reunión diaria con un time-box de 15 minutos donde el Development team planifica su trabajo para las próximas 24 horas optimizando el desempeño y revisando el trabajo avanzado desde el último Daily Scrum.

En este evento el DT evalúa el progreso hacia el Sprint Goal y optimiza las posibilidades para cumplir el mismo. También se encarga de establecer la estructura de la reunión y la cual se puede manejar de diferentes maneras. Estas reuniones mejoran la comunicación, evitan reuniones extras, reconocen impedimentos que deben eliminarse para proceder con el desarrollo, toma de decisiones rápida y nivel de conocimiento del DT. (Schwaber y Sutherland, 2017, p. 12,13).

1.4.5.5 Sprint Review

Este evento se lleva a cabo al final de cada Sprint para mostrar el Incremento y fomentar la colaboración, facilitar la retroalimentación de información y adecuar el Product Backlog si se necesitara. En la reunión los Stakeholders y el Scrum Team comunica lo que se hizo durante el Sprint, para después determinar aspectos que pueden ser aplicados para optimizar el valor tomando en cuenta cambios en el mercado y el uso potencial del producto

El PO determina qué elementos del Product Backlog están “terminados” y cuáles no, para después planear objetivos y fechas de entrega. El DT comunica los problemas que aparecieron durante el Sprint, sus soluciones y detalles que destacaron. El resultado del Sprint Review es un Product Backlog examinado que detalla los posibles elementos para el Sprint que siguen. La lista también puede ser adecuada para centrarse en nuevas oportunidades. (Schwaber y Sutherland, 2017, p. 13,14).

1.4.5.6 Sprint Retrospective

Es una reunión donde el Development Team y el Scrum Master discuten sobre los problemas que puede solucionar el Scrum Team y de problemas cuya solución sobrepasa la capacidad del equipo. Se proponen soluciones para los aspectos percibidos durante el sprint que puedan haber frenado la productividad. Se atienden temas que faciliten el cierre del sprint y el inicio de uno nuevo con el equipo fortalecido en todos los sentidos.

Según Monte (2016, p. 90) esta reunión debe atender los siguientes temas:

- Mantener: reconocer acciones o hábitos que se hacen actualmente y es conveniente que se sigan haciendo.
- Aumentar: identificar acciones o hábitos que se hacen pero que podrían potenciarse.
- Disminuir: identificar acciones o hábitos que no contribuyan al funcionamiento del Development Team y que se deben disminuir.
- Empezar a hacer: reconocer acciones o hábitos que no se hacen actualmente y que sería conveniente comenzar a hacer.
- Dejar de hacer: identificar acciones o hábitos que se hacen actualmente y sería conveniente dejar de hacer.

1.4.6 *Definition of “done”* o definición de “terminado”

Cuando un elemento del Product Backlog o un Incremento es descrito como “terminado” todo el equipo debe entender su significado. Para asegurar la transparencia, los miembros del Equipo Scrum deben tener conocimiento compartido del significado de trabajo completo. Esta definición se emplea para estimar cuándo se finaliza el trabajo sobre el Incremento de producto. Esta definición también encamina al equipo de desarrollo para conocer cuántos aspectos del Product Backlog es posible optar durante el Sprint Planning. (Schwaber y Sutherland, 2017, p. 18).

2. Desarrollo de la metodología

El presente proyecto se llevó a cabo con la metodología de desarrollo de software SCRUM. Una vez que se tienen claro los objetivos y el alcance del proyecto, el primer paso es definir el Scrum Team y precisar las responsabilidades que tiene cada uno.

2.1 Definición del Scrum Team

Tabla 3.

Scrum Team

Rol	Encargado	Responsabilidad
Product Owner	Ángel Jaramillo	Definir criterios de aceptación de cada Sprint.
Development Team	Juan José Bahamonde	Desarrollar las aplicaciones móviles y de la aplicación web completamente funcionales.
Scrum Master	Juan José Bahamonde	Gestionar y promover el correcto funcionamiento de Scrum.
Stakeholders	Docentes de Ingeniería	Colaborar con el Product Owner sobre dudas y aportar con información.

2.2 Generación de las historias de usuario

Para la recolección de los requerimientos, tanto de la aplicación web como de las aplicaciones móviles, se mantuvo una reunión con el Product Owner y con los Stakeholders, de la cual se pudo generar las historias de usuario respectivas.

El prototipo cuenta con dos sistemas: una página web para administrar la base de datos y visualizar reportes; y una aplicación móvil para el registro de equipos y actividades. Por lo tanto, cumple con dos funciones.

Tabla 4.

Roles del sistema

Rol	Funciones
Administrador	<ul style="list-style-type: none"> • Administrar la base de datos en la aplicación web. • Visualización de reportes • Visualización de tablas
Docente	<ul style="list-style-type: none"> • Registrar equipos • Registrar observaciones • Registrar actividades

Historia de usuario HU001

Tabla 5.

Anverso historia de usuario HU001

Título	Inicio de sesión en aplicación web	Identificación	HU001
Descripción	<ul style="list-style-type: none"> • El usuario administrador ingresa las credenciales para ingresar al sistema y se muestra la pantalla de inicio. 		

Notas	<ul style="list-style-type: none"> Validar el nombre de usuario y su contraseña en la base de datos 		
Story Points	8	Prioridad (1-5)	3
ID user story padre	-	Roles	Administrador

Tabla 6.

Reverso historia de usuario HU001

Criterios de aceptación	<ul style="list-style-type: none"> Si los datos son correctos el usuario debe ingresar sin problema a la aplicación web. Si los datos son erróneos el usuario no podrá ingresar a la aplicación. Solo los usuarios con rol de administrador pueden ingresar a la aplicación web. 		
Notas	<ul style="list-style-type: none"> Mostrar mensajes de error y de éxito. Se debe capturar los datos del usuario en cookies. 		
Número de tareas	2		
Estimación de horas en tareas	24	Horas reales de la tarea	20

Historia de usuario HU002

Tabla 7.

Anverso historia de usuario HU002

Título	CRUD de la base de datos	Identificación	HU002
Descripción	<ul style="list-style-type: none"> El usuario administrador crea, lee, elimina o edita la información de la base de datos 		

Notas	<ul style="list-style-type: none"> • Validar si la información ingresada para leer, eliminar o actualizar sí están en la base de datos. • No se pueden ingresar datos duplicados en las tablas con registros únicos. • Solo se deben eliminar y actualizar los datos de usuarios y equipos. • El ingreso de actividades, observaciones y préstamo de equipos se lo realiza solamente desde la aplicación móvil. 			
	Story Points	13	Prioridad (1-5)	5
	ID user story padre	HU001	Roles	Administrador

Tabla 8.

Reverso historia de usuario HU002

Criterios de aceptación	<ul style="list-style-type: none"> • El ingreso de usuarios y equipos se debe realizar correctamente en la base de datos. • La eliminación de usuarios y equipos se debe realizar correctamente. • La actualización de usuarios y equipos se debe realizar correctamente. • La lectura debe retornar los datos solicitados de todas las tablas de la base de datos. 			
	Notas	<ul style="list-style-type: none"> • Mostrar mensajes de error y de éxito. • La eliminación debe ser en cascada. 		
		Número de tareas	5	
	Estimación de horas en tareas	60	Horas reales de la tarea	48

Historia de usuario HU003

Tabla 9.

Anverso historia de usuario HU003

Título	Generación de reportes	de	Identificación	HU003
Descripción	<ul style="list-style-type: none"> • El usuario administrador selecciona los elementos que desea visualizar en un reporte. 			
Notas	<ul style="list-style-type: none"> • El administrador ingresa los parámetros para la generación del reporte. • Mostrar reportes de actividades, observaciones y equipos mensuales. • Se deben mostrar los datos requeridos a manera de información y no en un datagrid. 			
Story Points	8	Prioridad (1-5)	4	
ID user story padre	HU001	Roles	Administrador	

Tabla 10.

Reverso historia de usuario HU003

Criterios de aceptación	<ul style="list-style-type: none"> • Se muestra el reporte mensual de los equipos utilizados y los responsables de los mismos. • Se muestra reporte de las actividades en un mes y año determinado con sus respectivas observaciones. 			
--------------------------------	---	--	--	--

	<ul style="list-style-type: none"> Se muestra un reporte mensual de las observaciones realizadas durante o fuera de las actividades. 		
Notas	<ul style="list-style-type: none"> Mostrar mensajes de error y de éxito. 		
Número de tareas	4		
Estimación de horas en tareas	48	Horas reales de la tarea	24

Historia de usuario HU004

Tabla 11.

Anverso historia de usuario HU004

Título	Inicio de sesión en aplicación móvil	Identificación	HU004
Descripción	<ul style="list-style-type: none"> El usuario ingresa las credenciales para ingresar al sistema y se muestra la pantalla de inicio. 		
Notas	<ul style="list-style-type: none"> Validar el nombre de usuario y contraseña con la información en la base de datos. 		
Story Points	8	Prioridad (1-5)	3
ID user story padre	-	Roles	Docente, Administrador

Tabla 12.

Reverso historia de usuario HU004

Criterios de aceptación	<ul style="list-style-type: none"> El usuario ingresa a la aplicación sin problemas.
--------------------------------	---

	<ul style="list-style-type: none"> El inicio de sesión debe desarrollarse para iOS y Android. 		
Notas	<ul style="list-style-type: none"> Mostrar mensajes de error y de éxito. Capturar los datos del usuario en una variable. 		
Número de tareas	2		
Estimación de horas en tareas	48	Horas reales de la tarea	15

Historia de usuario HU005

Tabla 13.

Anverso historia de usuario HU005

Título	Ingreso de nueva actividad.	Identificación	HU005
Descripción	<ul style="list-style-type: none"> El usuario ingresa los datos de la actividad al sistema. 		
Notas	<ul style="list-style-type: none"> Se debe corroborar la información ingresada con un mensaje. Una vez ingresada la práctica el docente no podrá editarla. 		
Story Points	8	Prioridad (1-5)	4
ID user story padre	HU004	Roles	Docente, Administrador

Tabla 14.

Reverso historia de usuario HU005

Criterios de aceptación	<ul style="list-style-type: none"> El usuario ingresa a la actividad sin problema y ésta se registra en la base de datos. 		
Notas	<ul style="list-style-type: none"> Mostrar mensajes de error y de éxito. Capturar la información del ID de la última práctica ingresada en una variable. 		
Número de tareas	4		
Estimación de horas en tareas	24	Horas reales de la tarea	72

Historia de usuario HU006

Tabla 15.

Anverso historia de usuario HU006

Título	Ingreso y devolución de equipo.	Identificación	HU006
Descripción	<ul style="list-style-type: none"> El usuario reconoce el equipo con el lector de código de barras y lo ingresa a la base de datos. 		
Notas	<ul style="list-style-type: none"> Se debe corroborar la información capturada con el lector de código de barras. Una vez ingresado el equipo el usuario no puede eliminarlo de la base de datos. 		

	<ul style="list-style-type: none"> • Se debe llevar un contador de equipos ingresados. • Se deben devolver todos los equipos para que pueda finalizar la práctica. 		
Story Points	13	Prioridad (1-5)	3
ID user story padre	HU005	Roles	Docente, Administrador

Tabla 16.

Reverso historia de usuario HU006

Criterios de aceptación	<ul style="list-style-type: none"> • El usuario registra el ingreso de un equipo a la actividad actual. • El usuario registra la devolución de un equipo en la actividad actual. • El contador muestra la información de los equipos ingresados y a su vez equipos por devolver. • Los registros se almacenan en la base de datos. 		
Notas	<ul style="list-style-type: none"> • Mostrar mensajes de error y de éxito. 		
Número de tareas	4		
Estimación de horas en tareas	60	Horas reales de la tarea	48

Historia de usuario HU007

Tabla 17.

Anverso historia de usuario HU007

Título	Ingreso de observaciones	Identificación	HU007
Descripción	<ul style="list-style-type: none"> El usuario ingresa la descripción de una observación sobre los equipos. 		
Notas	<ul style="list-style-type: none"> Una vez ingresada la observación el usuario no podrá editarla. La observación no debe pertenecer a una sesión de práctica. 		
Story Points	5	Prioridad (1-5)	3
ID user story padre	HU004	Roles	Docente, Administrador

Tabla 18.

Reverso historia de usuario HU007

Criterios de aceptación	<ul style="list-style-type: none"> El usuario ingresa una observación sin problema durante una actividad. El usuario ingresa una observación sin problema fuera de una actividad. 		
Notas	<ul style="list-style-type: none"> Mostrar mensajes de error y de éxito. 		
Número de tareas	2		
Estimación de horas en tareas	24	Horas reales de la tarea	20

2.3 Sprint 0

En el Sprint 0 se implementaron las herramientas esenciales y se prepararon todos los ambientes necesarios para comenzar con el primer Sprint. Además, se preparó el Product Backlog en base a las historias de usuario.

2.3.1 Creación del Product Backlog

Para la elaboración del Product Backlog se consideraron dos aspectos principales de las historias de usuario:

- Story Points: Este parámetro indica la complejidad que representa la actividad para el equipo de desarrollo. En este proyecto se utilizó el método de estimación Planning poker, donde su escala de dificultad es representada con la serie de Fibonacci (1, 2, 3, 5, 8, 13).
- Prioridad: Indica la preferencia que deben tener las actividades.

Una vez que se analizaron las prioridades y su grado de dificultad se obtuvo como Product Backlog el siguiente resultado:

Tabla 19.

Product Backlog

Orden	Actividad	Story Points	Prioridad	Historia de usuario
1	Diseño y creación de la base de datos.	13	5	Determinado por el autor
2	Definir arquitecturas de las aplicaciones web y móvil.	8	5	Determinado por el autor

3	Diseño de la interfaz gráfica de las aplicaciones web y móvil.	8	5	Determinado por el autor
4	CRUD de la base de datos	13	4	HU002
5	Generación de reportes.	8	4	HU003
6	Ingreso de nueva actividad.	8	4	HU005
7	Ingreso y devolución de equipo	13	3	HU006
8	Inicio de sesión en aplicación web	8	3	HU001
9	Inicio de sesión en aplicación móvil.	8	3	HU004
10	Ingreso de observaciones.	5	3	HU007

2.3.2 Preparación de entornos

Para el desarrollo de las aplicaciones móviles se utilizó un solo computador donde se tienen instalados los dos entornos de desarrollo integrados y una máquina virtual con sistema operativo Windows Server 2012 Standard R2. En la máquina virtual se instalará Visual Studio 2013 y también el servidor de base de datos MySQL y el MySQL Workbench para su administración.

Una vez que se finalice el desarrollo del prototipo, se realizará la implementación de los servidores en el Data Center experimental de la FICA, donde también se publicará la aplicación web y el servicio web para las aplicaciones móviles con IIS.

Los prerequisites de los programas que se utilizarán para el desarrollo del sistema son las siguientes:

Tabla 20.

Requerimientos del sistema

Software	Prerrequisitos mínimos de Hardware	Prerrequisitos mínimos de Software
Xcode 9.2	MacBook Pro Mid 2010 o más reciente	Mac OS 10.12 Sierra
Android Studio 2.3.1	<ul style="list-style-type: none"> • 3GB de memoria RAM • 2GB de espacio en disco • Pantalla de 1280x800 	Mac OS 10.10 o versiones más recientes, hasta macOS Sierra.
Microsoft Visual Studio 2013 Community	<ul style="list-style-type: none"> • Procesador 1.6 Ghz • 1.5 GB de memoria RAM • 5 GB de espacio en el disco • Pantalla de 1024 x 768 	Windows Server 2012 R2 (x64)
MySQL Community Server	<ul style="list-style-type: none"> • Procesador de 2 núcleos <ul style="list-style-type: none"> • 2 GB de RAM • Subsistema de I/O de disco aplicable a una base de datos de escritura intensiva 	Windows x86 64-bit

Teniendo en cuenta los requerimientos mínimos se instalaron los IDEs con éxito. Para el desarrollo de la aplicación web se utilizó el IDE Visual Studio 2013 Community debido a que es una función gratuita para estudiantes. Ideal para el desarrollo de este prototipo.

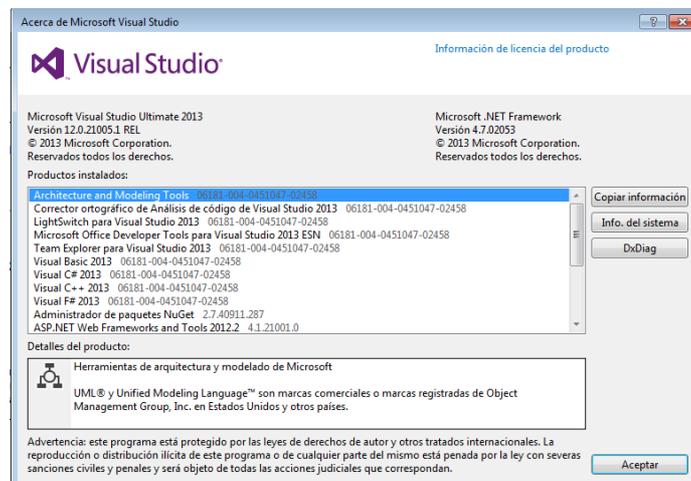


Figura 7. Visual Studio 2013 Instalado

El IDE para el desarrollo de la aplicación en iOS es la versión Xcode 9.2, la cual cuenta con Swift 4.0. La aplicación podrá ejecutarse en los dispositivos que tengan la versión 10.2 como mínima del sistema operativo iOS.



Figura 8. Xcode 9 Instalado

Para la aplicación de Android se utilizó el IDE Android Studio versión 2.3.1. Este es el IDE oficial para el desarrollo de las aplicaciones Android.



Figura 9. Android Studio 2.3.1 Instalado

2.4 Sprint Planing y Sprint Backlog

Los Sprint Planing permitieron dividir cada uno de los requerimientos en tareas más pequeñas, a las cuales se les asigna un tiempo determinado para realizarse y una persona responsable. Como producto final se tiene el Sprint Backlog:

Tabla 21.

Sprint Backlog General

Sprint	Actividad	Tareas	Tiempo (Sem)	Responsable
1	Diseño y creación de la base de datos.	1.1. Diseño Conceptual de la base de datos.	1	Juan José Bahamonde
		1.2. Diseño lógico estándar de la base de datos.		
		1.3. Diseño lógico específico de la base de datos.		

		1.4. Crear procedimientos almacenados.	
		1.5. Diseñar arquitectura para aplicación web.	
	Definir arquitecturas de las aplicaciones web y móvil.	1.6. Diseñar arquitectura para aplicación en Android.	
		1.7. Diseñar arquitectura para aplicación en iOS.	
		2.1. Diseño de interfaz para inicio de sesión.	
	Diseño de la interfaz gráfica	2.2. Diseño de web forms de ingreso de usuarios y equipos.	
2	de las aplicaciones web y móvil.	2.3. Diseño de web forms para actualizar usuarios y equipos	1
		2.4. Diseño de web forms para eliminar usuarios y equipos.	

Juan José Bahamonde

2.5. Diseño de web forms para reportes de prácticas, observaciones y equipos

2.6. Diseño de web forms para visualización de toda la información de la base de datos.

2.7. Diseño de inicio de sesión para iOS y Android.

2.8. Diseño de pantalla de inicio para IOS y Android

2.9. Diseño de pantallas de ingreso y devolución de equipos para IOS y Android

2.10. Diseño de pantallas para ingreso de actividades y observaciones para iOS y Android.

		3.1. Crear capas de negocios y de datos con conexión a la base de datos en la aplicación web.	
		3.2. Implementar métodos para ingresar datos de la aplicación web.	
3	CRUD de la base de datos	3.3. Implementar métodos para eliminar información de las tablas.	1
		3.4. Implementar métodos para editar las tablas.	
		3.5. Implementar métodos para leer todas las tablas.	
		4.1. Crear método para mostrar reportes de observaciones.	
4	Generación de reportes		1

Juan José Bahamonde

Juan José Bahamonde

4.2. Crear método para mostrar reportes de actividades.

4.3. Crear método para mostrar reportes de equipos.

4.4. Crear métodos para exportar datos a PDF.

5.1. Crear servicios web para conectarse a la base de datos para iOS y Android.

5 Ingreso de nueva actividad. 5.2. Implementar métodos para ingresar una actividad en Android y en iOS. 1 Juan José Bahamonde

5.3. Implementar métodos para recuperar la actividad actual.

5.4. Implementar método para finalizar una actividad.

5.5. Implementar lector de código de barras en Android y en iOS.

5.6. Implementar método para leer equipo por código ingresado en Android y en iOS.

Ingreso y devolución de equipo

5.7. Implementar método para ingresar la información del equipo escaneado en Android y en iOS.

5.8. Implementar métodos para devolver equipos.

6.1. Implementar métodos para iniciar sesión en aplicación web. 1

6

Inicio de sesión en aplicación web

Juan José Bahamonde

		6.2. Implementar métodos para guardar los datos del usuario ingresado.	
		6.3. Implementar métodos para iniciar sesión en Android y en iOS.	
	Inicio de sesión en aplicación móvil.	6.4. Implementar métodos para guardar los datos en una variable en Android y en iOS.	
		7.1. Implementar métodos para ingresar observaciones fuera de actividad en iOS y Android.	
7	Ingreso de observaciones	7.2. Implementar métodos para ingresar observaciones dentro de actividad en iOS y Android.	1
			Juan José Bahamonde
Total	7 Sprints	40 TAREAS	7 SEMANAS

2.5 Cronograma de desarrollo

Una vez definidos los Sprint con sus respectivas tareas y tiempo de ejecución, se puede definir el cronograma de desarrollo del sistema. Para este proyecto se determinó que cada Sprint durará una semana, sin embargo, éste puede terminar antes y se iniciaría el siguiente sprint. Cabe recalcar que en teoría no se tiene un tiempo de Sprint establecido. El tiempo y el número de Sprints que haya en un proyecto dependen de cada equipo Scrum.

Tabla 22.

Cronograma de desarrollo inicial

Sprint	Tiempo (Semanas)	Fecha Inicio	Fecha Fin
1	1	11 de abril de 2018	17 de abril de 2018
2	1	18 de abril de 2018	24 de abril de 2018
3	1	25 de abril de 2018	1 de mayo de 2018
4	1	2 de mayo de 2018	8 de mayo de 2018
5	1	9 de mayo de 2018	15 de mayo de 2018
6	1	16 de mayo de 2018	22 de mayo de 2018
7	1	23 de mayo de 2018	29 de mayo de 2018

2.6 Primer Sprint

El primer Sprint fue desarrollado del 11 de abril de 2018 al 17 de abril de 2018, como se tenía planteado en el cronograma de desarrollo inicial. Se definió la arquitectura para las tres aplicaciones que se desarrollarán y se realizó el diseño de la base de datos que se va a utilizar.

Tabla 23.

Primer Sprint

Sprint	Actividad	Tareas	Tiempo	Actividad	
1	Diseño y creación de la base de datos.	1.1. Diseño Conceptual de la base de datos.	1	Diseño y creación de la base de datos.	
		1.2. Diseño lógico estándar de la base de datos.			
		1.3. Diseño lógico específico de la base de datos.			
	Definir arquitecturas de las aplicaciones web y móvil.	1.4. Crear procedimientos almacenados.		1	Definir arquitecturas de las aplicaciones web y móvil.
		1.5. Diseñar arquitectura para aplicación web.			
		1.6. Diseñar arquitectura para aplicación en Android.			

1.7. Diseñar
arquitectura para
aplicación en iOS.

2.6.1 Diseño y creación de la base de datos

A continuación, se detalla el desarrollo de cada una de las tareas definidas en el Sprint Backlog.

2.6.1.1 Diseño Conceptual de la base de datos

En el diseño conceptual se debe tomar en cuenta los requerimientos del sistema que se pretende ejecutar. El primer paso es identificar las entidades, las cuales serán la base del funcionamiento del prototipo. En este caso se tienen 4 entidades: Usuario, Equipo, Observación y Práctica.

Una vez que se definió bien las funcionalidades de las aplicaciones, se debe normalizar la base de datos para evitar la redundancia de datos innecesarios, el uso ineficiente de la memoria y también para asegurar la integridad de la información de la base de datos. En este caso, se empleó la tercera forma normal o 3FN para el modelo final.

El modelo conceptual fue desarrollado con la herramienta Dia, la cual permite realizar diagramas exclusivos para base de datos. A continuación, se muestra el modelo de la base de datos con sus respectivas entidades, atributos, y relaciones entre sí.



Figura 10. Modelo Conceptual Entidad-Relación

2.6.1.2 Diseño Lógico estándar de la base de datos

Con el modelo Entidad-Relación se puede conseguir que el usuario visualice la base de datos como una estructura lógica uniforme y simple, la cual representa el conjunto de sus relaciones. El esquema estándar es independiente del sistema de gestión de datos o DBMS.

Una vez que se desarrolló el modelo Entidad-Relación, se requiere implementar el esquema relacional base a dicho modelo. Las entidades forman una tabla, y sus atributos son sus columnas. En cuanto a las claves foráneas, al tener relaciones entre entidades 1:1 o 1:M, migran las claves de la tabla padre a la tabla hija formando su clave foránea. Si las relaciones son M:N, se crea una nueva tabla para representar su relación. Este modelo permite definir claramente la estructura de la base de datos.

Para el diseño de este esquema lógico estándar se utilizó la herramienta de ingeniería reversa de MySQL Workbench. Las relaciones 1:M se distinguen por las “patas de cuervo” que indica a dónde debe migrar la clave de la entidad. Entre estas relaciones se tienen de Usuario > Observación y Usuario > Práctica. La relación M:N que se muestra entre práctica y equipo crea una nueva tabla entre ambas entidades, por lo que las claves primarias migran a ésta nueva tabla.

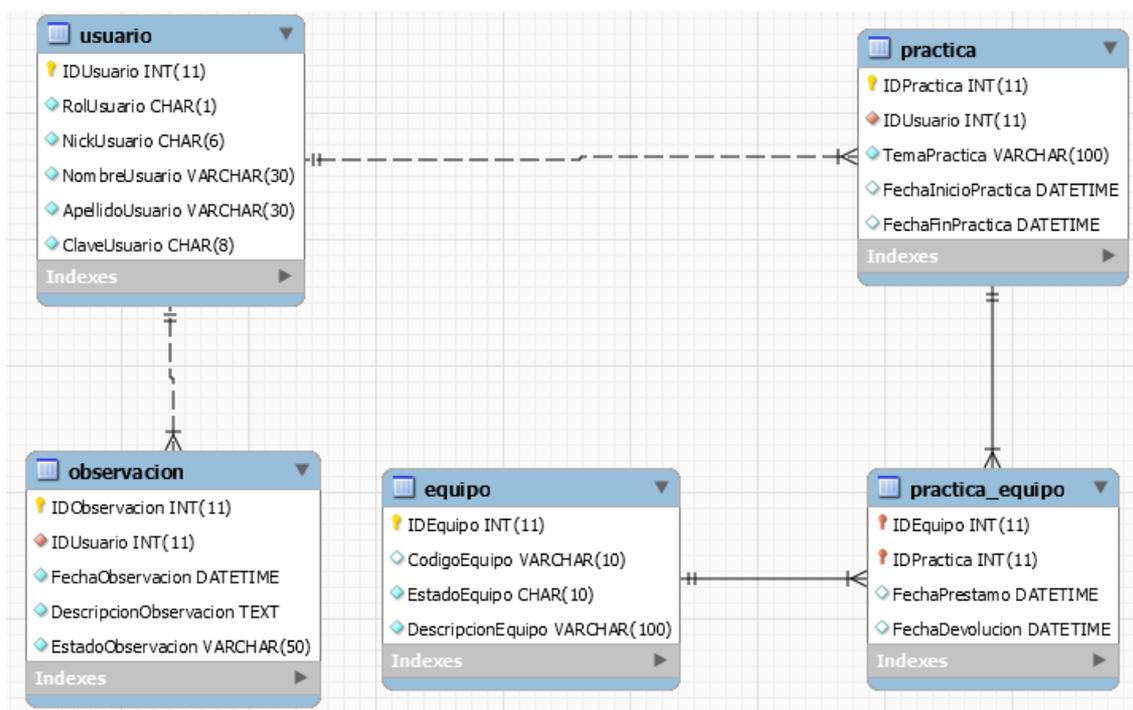


Figura 11. Esquema lógico estándar E-R

2.6.1.3 Diseño Lógico específico de la base de datos

El esquema lógico específico consiste en la creación de las tablas en el DBMS, mediante un *script*. Se escriben instrucciones para que luego sean ejecutadas, se deben especificar el tipo de datos para cada uno de los atributos, claves primarias y claves foráneas. MySQL no soporta la creación de tablas con restricciones en los atributos como lo hace SQL Server, por lo que deben ser manejadas en procedimientos almacenados.

Uno de los aspectos importantes para la creación de las tablas con relaciones entre entidades es la eliminación en cascada. Esta permite que se eliminen todos los registros, de todas las tablas que contengan el ID del elemento eliminado. En este caso, se debe realizar en las tablas Observacion, Practica y Practica_Equipo. De esta forma, si se elimina un usuario, se eliminarán en cascada todas las observaciones y prácticas que haya realizado.

Para el desarrollo del modelo lógico específico se utilizó el DBMS de Oracle MySQL Workbench. A manera de ejemplo se muestra el script de la creación de la base de datos y de una de las tablas, el diseño lógico específico completo se encuentra adjuntado en el CD en la carpeta de anexos.

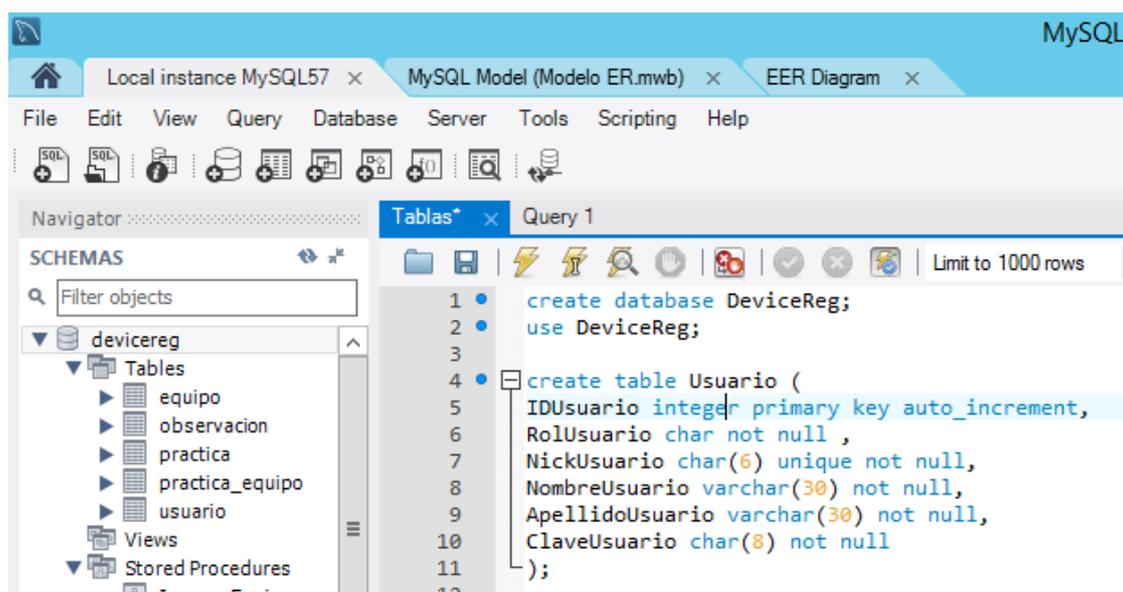


Figura 12. Esquema lógico específico, creación de tabla usuario

En la siguiente tabla se muestran todos los atributos con su respectiva clave foránea y tipo de dato:

Tabla 24.

Atributos y entidades

Atributo	Entidad	Key	Tipo de dato	Not NULL
ApellidoUsuario	usuario		VARCHAR(30)	Yes
ClaveUsuario	usuario		CHAR(8)	Yes
CodigoEquipo	equipo		VARCHAR(10)	No
DescripcionEquipo	equipo		VARCHAR(100)	Yes
DescripcionObservacion	observacion		TEXT	Yes
EstadoEquipo	equipo		CHAR(10)	Yes
FechaDevolucion	practica_equipo		DATETIME	No
FechaFinPractica	practica		DATETIME	No
FechaInicioPractica	practica		DATETIME	No
FechaObservacion	observacion		DATETIME	Yes
FechaPrestamo	practica_equipo		DATETIME	No
IDEquipo	equipo	PK	INTEGER(11)	Yes
IDEquipo	practica_equipo	PK, FK	INTEGER(11)	Yes
IDObservacion	observacion	PK	INTEGER(11)	Yes
IDPractica	practica	PK	INTEGER(11)	Yes
IDPractica	practica_equipo	PK, FK	INTEGER(11)	Yes
IDUsuario	observacion	FK	INTEGER(11)	Yes
IDUsuario	practica	FK	INTEGER(11)	Yes
IDUsuario	usuario	PK	INTEGER(11)	Yes
NickUsuario	usuario		CHAR(6)	Yes
NombreUsuario	usuario		VARCHAR(30)	Yes
RolUsuario	usuario		CHAR(1)	Yes
TemaPractica	practica		VARCHAR(100)	Yes
EstadoObservacion	observacion		VARCHAR(50)	Yes

Con el fin de mostrar un término más general en la interfaz gráfica de las aplicaciones, se manejará el término “Actividad” que representa la tabla Practica en la base de datos.

2.6.1.4 Creación de procedimientos almacenados

Los procedimientos almacenados fueron creados para ayudar a facilitar las transacciones que se realizarán. Con estos procedimientos la programación de las aplicaciones será más sencilla, debido a que no se tiene que programar toda la lógica, sino, solo llamar al método y enviar las variables necesarias para correr el procedimiento almacenado. También, ayuda a controlar las restricciones que tiene cada uno de los atributos de las tablas, manteniendo así la integridad de los datos.

Se crearon en total nueve procedimientos almacenados, los cuales fueron diseñados para enviar la menor cantidad de datos posibles en la aplicación. Esto quiere decir que el procedimiento mismo se encarga de realizar los cálculos necesarios para cada transacción.

Este sistema tendrá una arquitectura de tres capas, por lo tanto, al tener procedimientos almacenados, se mantiene gran parte de la capa de datos en la misma base de datos y no en la aplicación en sí.

En el primer procedimiento se hace el ingreso del equipo. Se deben ingresar solamente dos datos: el código y la descripción. Al ingresar un equipo a la base de datos automáticamente su estado es disponible, por lo que no es necesario ingresar ese dato.

Tabla 25.

Procedimiento para ingreso del equipo

Procedimiento para ingreso del equipo	
Nombre	IngresoEquipo
Código	<pre> create definer=`root`@`localhost` procedure `ingresoequipo`(in codigo varchar(10), in descripcion varchar(100)) begin insert into equipo (codigoequipo,estadoequipo,descripcionequipo) values (codigo,'d',descripcion); end </pre>

La fecha y hora tanto de inicio como fin de la actividad se la toma automáticamente del primer equipo que se toma prestado y hasta que se devuelve el último equipo. Por lo que solo es necesario ingresar el identificador de la actividad o practica para que se realice esta acción. Este procedimiento solo funciona una vez que la actividad ha finalizado y se ha devuelto el último equipo.

Tabla 26.

Procedimiento para el ingreso automático de las fechas de la actividad

Procedimiento para el ingreso automático de las fechas	
Nombre	IngresoFechaPractica
Código	<pre> create definer=`root`@`localhost` procedure `ingresofechapractica`(in practica int) begin declare fechainicio datetime; </pre>

```

declare fechafin datetime;
set fechainicio=(select fechaprestamo from practica_equipo
where idpractica=practica order by fechaprestamo limit 1) ;
set fechafin=(select fechadevolucion from practica_equipo where
idpractica=practica order by fechadevolucion desc limit 1);

update practica set fechainiciopractica = fechainicio where
idpractica=practica;
update practica set fechafinpractica = fechafin where
idpractica=practica;
end

```

Para el ingreso de las observaciones se necesita el identificador de la tabla Usuario, su descripción y el identificador de la tabla Practica en el caso de que se ingrese una observación durante una actividad. La fecha y hora del ingreso de la observación se la toma automáticamente. El estado tiene dos valores, dentro y fuera de la actividad. En el caso que sea dentro se muestra el identificador de la actividad.

Tabla 27.

Procedimiento para el ingreso de una observación

Procedimiento para ingreso de una observación

Nombre	IngresoObservacion
---------------	--------------------

Código	<pre> CREATE DEFINER=`root`@`localhost` PROCEDURE `IngresoObservacion`(in usuario int, in observacion text, in practica int) BEGIN declare estado varchar(50); </pre>
---------------	---

```

if (practica!=0)
then
set estado=concat('Durante actividad: ', practica);
end if;
if (practica=0)
then
set estado='Fuera de actividad';
end if;
insert into observacion
(idusuario,fechaobservacion,descripcionobservacion,
estadoobservacion)
values (usuario,now(),observacion,estado);
END

```

Al ingresar una actividad, solo se debe indicar el usuario que lo está haciendo y el tema de esta. Las fechas del inicio y fin son ingresadas automáticamente como se mostró en la tabla 25.

Tabla 28.

Procedimiento para ingreso de la actividad

Procedimiento para ingreso de la actividad

Nombre	IngresoPractica
Código	<pre> CREATE DEFINER='root'@'localhost' PROCEDURE `IngresoPractica`(in usuario int, in tema varchar(100)) BEGIN insert into practica (idusuario,temapractica) values(usuario,tema); END </pre>

El procedimiento para ingreso de usuarios está diseñado para generar automáticamente el *nick* del usuario. Este consiste en seis caracteres, los cuales están conformados por las dos primeras letras del nombre, las dos primeras del apellido y el número 10. En el caso de que haya un usuario con las mismas cuatro letras se incrementa en uno el número, hasta que el usuario tenga un *nick* único.

Tabla 29.

Procedimiento para ingreso de usuarios

Procedimiento para ingreso de usuarios	
Nombre	IngresoUsuario
Código	<pre> CREATE DEFINER=`root`@`localhost` PROCEDURE `IngresoUsuarios`(in rol char(1), in nombre varchar(30), in apellido varchar(30), in clave char(8)) BEGIN declare nom varchar(2); declare ape varchar(2); declare nick char(6); declare cont char(2); set cont=10; set nom=substring(nombre from 1 for 2); set ape=substring(apellido from 1 for 2); set nick = concat(nom,ape,cont); while nick=(select nickusuario from usuario where nickusuario=nick) do </pre>

```

set cont = cont +1;
set nick=concat(nom,ape,cont);
end while;

insert into usuario
(rolusuario,nickusuario,nombreusuario,apellidousuario,claveusuario)
values (rol,nick,nombre,apellido,clave);
END

```

En el siguiente procedimiento se abarca tanto el préstamo como la devolución del equipo. Al ingresar un préstamo se toma como fecha de préstamo la actual en ese momento, junto con la hora y automáticamente el estado del equipo cambia a “Prestado”. Al volver a registrar el mismo equipo, se toma en cuenta como una devolución, por lo que se actualiza la hora y fecha de devolución en el mismo campo y el estado del equipo vuelve a estar “Disponible”.

Tabla 30.

Procedimiento para préstamo de equipo

Procedimiento para préstamo de equipo

Nombre PrestamoEquipo

```

CREATE DEFINER=`root`@`localhost` PROCEDURE
`PrestamoEquipo`(
in ipractica int,
in iequipo int
)

```

Código BEGIN

```

declare estado char(1);
declare fechaprestamo datetime;
declare fechadevolucion datetime;

```

```
set estado = (select EstadoEquipo from equipo where
IdEquipo=iequipo);

if (estado='D') then
insert into practica_equipo (idequipo,idpractica,fechaprestamo)
values (iequipo,ipractica, now());
update equipo set estadoequipo='P' where idequipo=iequipo;

end if;

if (estado='P') then
update equipo set estadoequipo='D' where idequipo=iequipo;
update practica_equipo set fechadevolucion=now() where
idequipo=iequipo and idpractica=ipractica;

end if;

END
```

En cuanto a los reportes, estos podrán generarse mensualmente. En el procedimiento del reporte mensual de los equipos se calcula cuántos equipos se tomaron y su respectivo tiempo de uso. Esto se lo hace sumando el tiempo que fueron utilizados en cada una de las actividades realizadas en un mes determinado. Para esto simplemente se deben ingresar el mes y el año que se desee analizar.

Tabla 31.

Procedimiento para el reporte mensual de equipos

Procedimiento para el reporte mensual de equipos

Nombre	ReporteMensualEquipos
Código	<pre> CREATE DEFINER=`root`@`localhost` PROCEDURE `ReporteMensualEquipos`(in mes int, in ano int) BEGIN select @rownum:=@rownum+1 AS Nro, DescripcionEquipo, CodigoEquipo, sec_to_time((sum(time_to_sec ((timediff(FechaDevolucion,FechaPrestamo)))))) as Tiempo from (SELECT @rownum:=0) r, equipo join practica_equipo on Equipo.idequipo=practica_equipo.IDEquipo and month(FechaDevolucion)=mes and year(fechadevolucion)=ano group by equipo.IDEquipo; END </pre>

Con el fin de filtrar la información por el usuario responsable de los equipos, se realizó un procedimiento para que se muestren todos los equipos que utilizó un docente durante un mes y año determinado.

Tabla 32.

Procedimiento almacenado para mostrar responsable de equipo

Procedimiento para el reporte mensual del responsable de los equipos	
Nombre	reporteResponsable
Código	<pre> CREATE DEFINER=`root`@`localhost` PROCEDURE `reporteResponsable`(</pre>

```

in mes int,
in ano int
)
BEGIN

select concat(nombreusuario , ' ',apellidousuario) as
Responsable, descripcionequipo as Equipo, codigoequipo as
Codigo, fechaprestamo as Prestamo, fechadevolucion as
Devolucion
from practica_equipo,equipo, usuario, practica
where Equipo.IDEquipo=practica_equipo.IDEquipo and
practica.IDUsuario=usuario.idusuario
and practica.idpractica=practica_equipo.IDPractica
and month(fechadevolucion)=mes and
year(fechadevolucion)=ano ;

END

```

En el reporte mensual de las observaciones se muestran todas las que se hicieron en un determinado mes y año. Se muestra la cantidad de observaciones que se realizaron, la persona que las hizo, el estado y su hora y fecha de ingreso. También se creó un procedimiento para filtrar las observaciones por usuario.

Tabla 33.

Procedimiento para el reporte mensual de observaciones

Procedimiento para el reporte mensual de observaciones

Nombre ReporteMensualObservacion

Código CREATE DEFINER=`root`@`localhost` PROCEDURE
`ReporteMensualObservacion`(

```

in mes int,
in ano int
)
BEGIN

SELECT @rownum:=@rownum+1 AS Nro,
DescripcionObservacion as Descripcion, FechaObservacion as
Fecha, EstadoObservacion as Estado , concat(NombreUsuario, '
',ApellidoUsuario) as Docente

FROM (SELECT @rownum:=0) r, usuario,observacion
where month(Observacion.Fechaobservacion)=mes
and year(Observacion.Fechaobservacion)=ano and
usuario.IDUsuario=observacion.IDUsuario;

END

```

Tabla 34.

Procedimiento para el reporte mensual de observaciones por usuario

Procedimiento para el reporte mensual de observaciones `por usuario

Nombre	ReporteMensualObservacionIndividual
---------------	-------------------------------------

Código	<pre> CREATE DEFINER=`root`@`localhost` PROCEDURE `ReporteMensualObservacionIndividual`(in usuario int, in mes int, in ano int) BEGIN </pre>
---------------	--

```
SELECT @rownum:=@rownum+1 AS Nro,
DescripcionObservacion as Descripcion, FechaObservacion as
Fecha, EstadoObservacion as Estado , concat(NombreUsuario, '
',ApellidoUsuario) as Docente
```

```
FROM (SELECT @rownum:=0) r, usuario,observacion
where month(Observacion.Fechaobservacion)=mes
and year(Observacion.Fechaobservacion)=ano and
usuario.IDUsuario=observacion.IDUsuario and
usuario.IDUsuario=usuario;
```

```
END
```

Para mostrar las observaciones que se realizaron durante una actividad se realizaron dos procedimientos almacenados, uno que muestra todas las observaciones en un determinado mes y año; y otro que cuenta con un filtro por docente.

Tabla 35.

Procedimiento para observaciones durante actividad en general

Procedimiento para el reporte mensual de observaciones durante actividad en general

Nombre ObservacionPracticaTodos

```
CREATE DEFINER=`root`@`localhost` PROCEDURE
`ObservacionPracticaTodos`(
in mes int,
Código in ano int
)
BEGIN
```

```

select idobservacion as Observacion, idusuario as Usuario,
fechaobservacion as Fecha, DescripcionObservacion as
Descripcion, estadoobservacion as Estado
from observacion
where estadoobservacion like 'Durante actividad: %'
and month(fechaobservacion)=mes and
year(fechaobservacion)=ano ;

END

```

Tabla 36 .

Procedimiento para observaciones durante actividad individual

Procedimiento para el reporte mensual de observaciones durante actividad individual

Nombre ObservacionPracticaUsuario

```

CREATE DEFINER='root'@'localhost' PROCEDURE
`ObservacionPracticaUsuario`(
in usuario int,
in mes int,
in ano int
)
BEGIN

```

Código

```

select idobservacion as Observacion, idusuario as Usuario,
fechaobservacion as Fecha, DescripcionObservacion as
Descripcion, estadoobservacion as Estado
from observacion
where estadoobservacion like 'Durante actividad: %'
and month(fechaobservacion)=mes and
year(fechaobservacion)=ano and idusuario=usuario;

```

END

Para proveer un reporte mensual por profesor, el procedimiento almacenado, muestra todas las actividades realizadas en un mes y año determinados, con su tiempo de duración por cada una de ellas. También se realizó un procedimiento para mostrar un reporte general de todos los profesores.

Tabla 37.

Procedimiento para el reporte mensual por profesor

Procedimiento para el reporte mensual por profesor

Nombre ReporteMensualProfesor

```
CREATE DEFINER=`root`@`localhost` PROCEDURE
`ReporteMensualProfesor`(
in usuario int,
in mes int,
in ano int
)
BEGIN
```

```
declare tiempopractica time;
```

Código

```
SELECT @rownum:=@rownum+1 AS Nro, NombreUsuario,
ApellidoUsuario,TemaPractica,
FechaIniciopractica, FechaFinPractica,
timediff(Fechafinpractica,fechainiciopractica)
as Tiempo
```

```
FROM (SELECT @rownum:=0) r, usuario,practica
where Usuario.IDusuario=usuario and
month(Practica.FechainicioPractica)=mes
```

```
and year(Practica.FechainicioPractica)=ano;
```

```
END
```

Tabla 38.

Procedimiento para el reporte mensual de todos los profesores

Procedimiento para el reporte mensual de todos los profesores

Nombre	ReporteMensualTodosProfesor
--------	-----------------------------

```
CREATE DEFINER=`root`@`localhost` PROCEDURE
```

```
`ReporteMensualTodosProfesor` (
```

```
in mes int,
```

```
in ano int
```

```
)
```

```
BEGIN
```

```
declare tiempopractica time;
```

Código

```
SELECT @rownum:=@rownum+1 AS Nro, NombreUsuario as
Nombre, ApellidoUsuario as Apellido, TemaPractica as Practica,
FechaIniciopractica as Inicio , FechaFinPractica as Fin,
timediff(FechaFinPractica, fechaIniciopractica)
as Duracion
```

```
FROM (SELECT @rownum:=0) r, usuario, practica
```

```
where usuario.idusuario=practica.idusuario and
```

```
month(Practica.FechainicioPractica)=mes
```

```
and year(Practica.FechainicioPractica)=ano;
```

```
END
```

Se creó un procedimiento para poder visualizar los préstamos que se han realizado a un usuario determinado. Como resultado, se tiene los identificadores de los equipos, identificador de la actividad, el código, el responsable, y la fecha de préstamo y devolución.

Tabla 39.

Procedimiento para mostrar préstamo de equipos a profesores

Procedimiento para mostrar préstamo de equipos			
Nombre	verPrestamo		
Código	CREATE	DEFINER=`root`@`localhost`	PROCEDURE
	`verPrestamo`(in id int) BEGIN		
	select practica_equipo.idequipo, practica_equipo.idpractica, equipo.codigoequipo, concat(nombreusuario,'',apellidousuario) as Responsable, practica_equipo.fechaprestamo, practica_equipo.fechadevolucion from practica_equipo,practica,equipo,usuario where equipo.idequipo=practica_equipo.idequipo and practica.idpractica=practica_equipo.idpractica and usuario.idusuario=practica.idusuario and usuario.idusuario=id order by idpractica;		
	END		

Finalmente, se creó una vista para mostrar el préstamo de equipos, pero en este caso se recuperan todos los datos sin ningún filtro.

Tabla 40.

Vista para mostrar préstamos

Vista para mostrar préstamo de equipos general	
Nombre	vistaprestamo
Código	<pre> CREATE ALGORITHM = UNDEFINED DEFINER = `root`@`localhost` SQL SECURITY DEFINER VIEW `vistaprestamo` AS SELECT `practica_equipo`.`IDEquipo` AS `idequipo`, `practica_equipo`.`IDPractica` AS `idpractica`, `equipo`.`CodigoEquipo` AS `codigoequipo`, CONCAT(`usuario`.`NombreUsuario`, ' ', `usuario`.`ApellidoUsuario`) AS `Responsable`, `practica_equipo`.`FechaPrestamo` AS `fechaprestamo`, `practica_equipo`.`FechaDevolucion` AS `fechadevolucion` FROM (((`practica_equipo` JOIN `practica`) JOIN `equipo`) JOIN `usuario`) WHERE (`equipo`.`IDEquipo` = `practica_equipo`.`IDEquipo`) AND (`practica`.`IDPractica` = `practica_equipo`.`IDPractica`) AND (`usuario`.`IDUsuario` = `practica`.`IDUsuario`) ORDER BY `practica_equipo`.`IDPractica` </pre>

2.6.2 Definición de las arquitecturas para las aplicaciones móviles y web.

Es necesario definir la arquitectura que se va a utilizar en este proyecto con el fin de facilitar su entendimiento y posterior implementación.

2.6.2.1 Arquitectura de la aplicación web

La arquitectura que se utilizará en este sistema es de tres capas, las cuales son:

- Capa de Presentación
- Capa de Negocio
- Capa de Datos

La capa de presentación contiene los elementos que componen la interfaz gráfica, en este caso son las páginas web aspx, aspx.cs, páginas maestras, entre otras. Actúa como intermediario entre el usuario final y el sistema. Es el Front-End de la aplicación, por lo que permite proveer servicios al usuario y obtener datos para que el resto de las capas lo puedan procesar. La capa de negocio contiene el comportamiento del sistema, dependiendo de la información que sea entregada por la capa de datos y también puede actualizarla si se necesita. También, indica los procesos que se deben hacer con los datos. Finalmente, la capa de datos provee el acceso y almacenamiento de los datos. Consiste en un gestor de bases de datos o de objetos.

En este proyecto se implementaron las tres capas en un mismo servidor físico, por lo que para que sea considerado como una arquitectura multicapa se debe aislar las operaciones de distinta naturaleza. De esta manera se facilita el mantenimiento y se optimiza el rendimiento del sistema. Por lo tanto, la arquitectura es la siguiente:

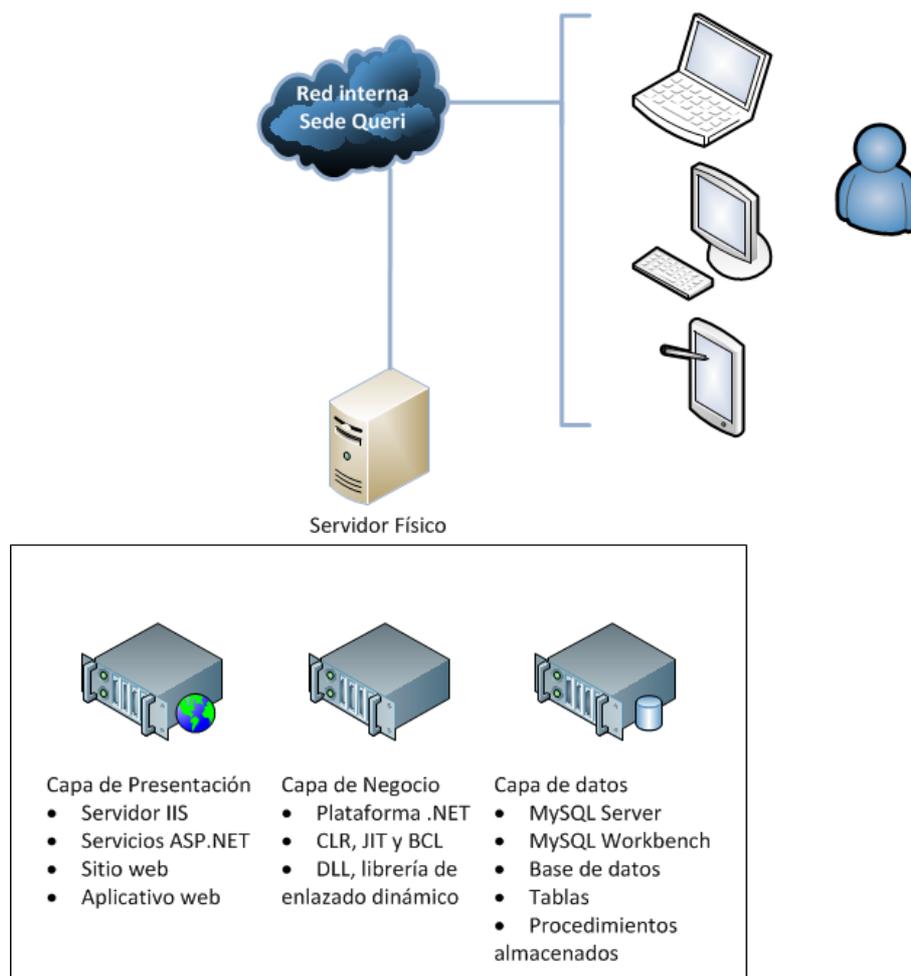


Figura 13. Diagrama de la arquitectura de aplicación web

2.6.2.2 Arquitectura de la aplicación móvil para Android

Para conseguir que la aplicación móvil se conecte con la base de datos se debe conectar con la tecnología de Web Services. Para lo cual se crea un aplicativo en PHP que contiene todos los procedimientos necesarios para conectarse a la base de datos, se lo publica con ayuda del Internet Information Services y con la aplicación móvil se conecta a dicho aplicativo por medio de la red inalámbrica.

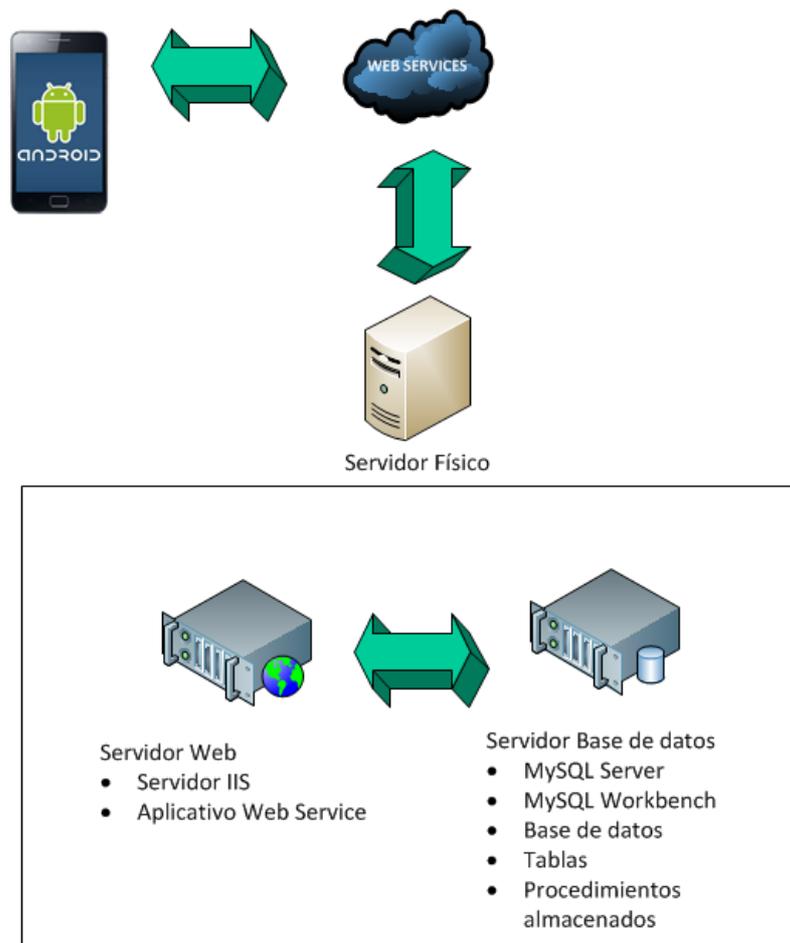


Figura 14. Arquitectura de la aplicación móvil en Android

2.6.2.3 Arquitectura de la aplicación móvil para iOS

La arquitectura para que la aplicación en iOS pueda comunicarse y enviar información a la base de datos tiene el mismo concepto de la aplicación para Android. La aplicación móvil funciona con Web Services, por lo que necesita conectarse al aplicativo por medio del internet. Este aplicativo contiene el código necesario para conectarse a la base de datos MySQL y para realizar las consultas necesarias. La aplicación se desarrollará con Swift 4.0.

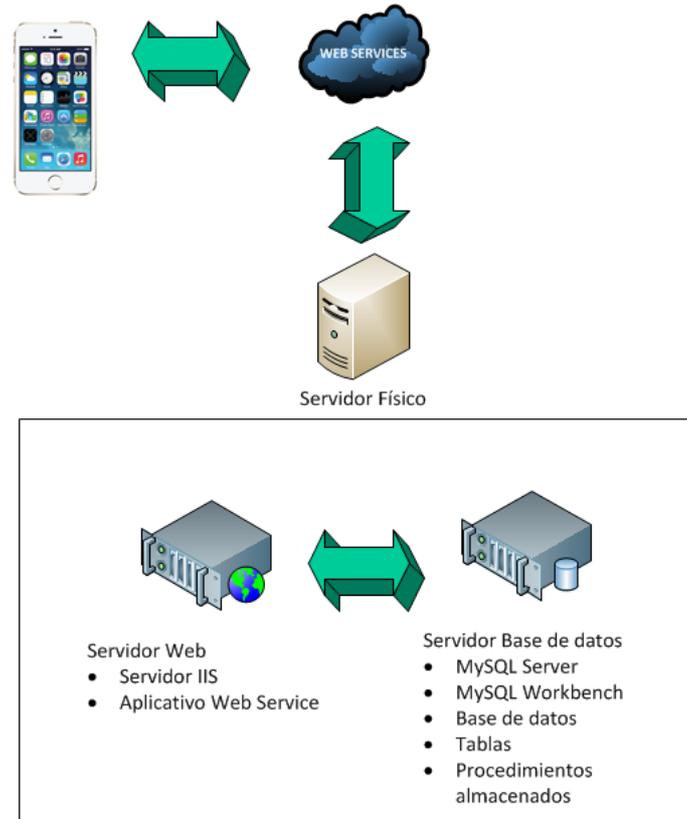


Figura 15. Arquitectura de aplicación móvil en iOS

2.7 Segundo Sprint

El segundo Sprint se desarrolló desde el 25 de abril de 2018 al 1 de mayo de 2018, siguiendo con lo programado en el cronograma de desarrollo. Se diseñaron todas las interfaces gráficas tanto de la aplicación web como de las aplicaciones móviles.

Tabla 41.

Segundo Sprint

Sprint	Actividad	Tareas	Tiempo (Sem)	Responsable
--------	-----------	--------	-----------------	-------------

	2.1. Diseño de interfaz para inicio de sesión.	
	2.2. Diseño de web forms de ingreso de usuarios y equipos.	
	2.3. Diseño de web forms para actualizar usuarios y equipos.	
	2.4. Diseño de web forms para eliminar usuarios y equipos.	
2	Diseño de la interfaz gráfica de las aplicaciones web y móvil.	1
	2.5. Diseño de web forms para reportes de prácticas, observaciones y equipos.	Juan José Bahamonde
	2.6. Diseño de web forms para visualización de toda la información de la base de datos.	
	2.7. Diseño de inicio de sesión para iOS y Android.	
	2.8. Diseño de pantalla de inicio para IOS y Android	

2.9. Diseño de pantallas de ingreso y devolución de equipos para IOS y Android

2.10. Diseño de pantallas para ingreso de actividades y observaciones para iOS y Android.

2.7.1 Diseño de la interfaz gráfica de las aplicaciones web y móvil.

A continuación, se describe el desarrollo de las tareas definidas en el Sprint Backlog para el segundo sprint. Con el fin de darle una identidad propia al sistema se diseñaron tres logotipos con las siglas SERQ que corresponde a Sistema de Registro de Equipos.



Figura 16. Logotipos del sistema

Para darle el mismo estilo y facilitar los cambios de la aplicación web se utilizaron varias clases CSS utilizadas para definir el formato de tablas, cuadros de texto, botones y fuentes.

2.7.1.1 Diseño de interfaz para inicio de sesión.

Una vez que se ingresa a la aplicación web se muestra una página maestra con un menú desplegable, el campo para mostrar los usuarios que están conectados y el botón para iniciar o cerrar sesión.

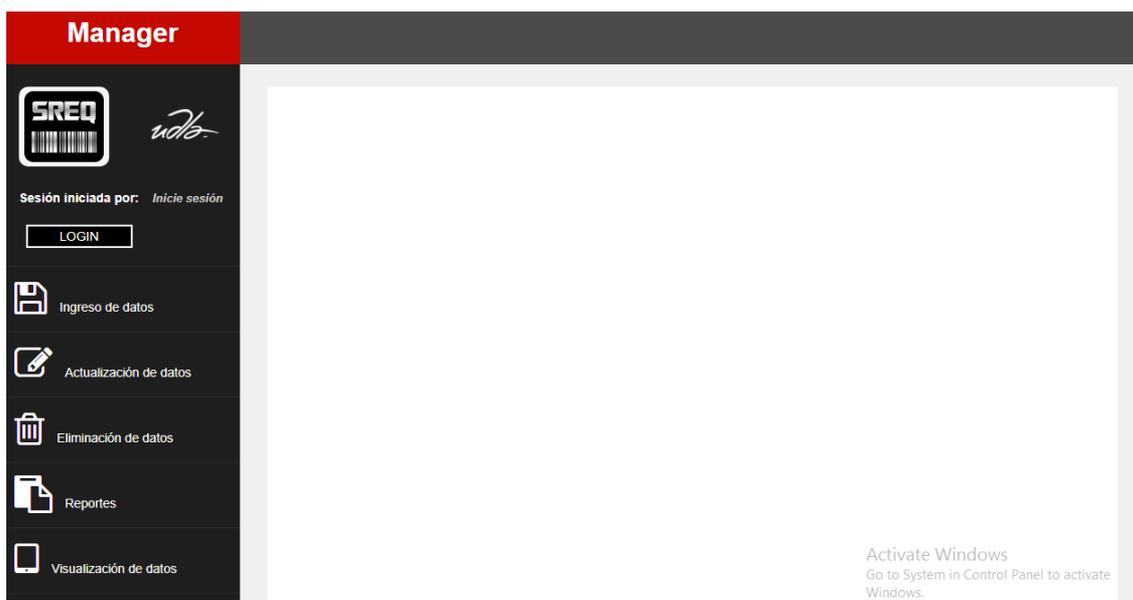


Figura 17. Página de Inicio

Al dirigirse a la pantalla de inicio de sesión, se carga una nueva página maestra donde no existe ningún menú desplegable y contiene la forma web correspondiente con los campos necesarios para iniciar sesión.



Figura 18. Pantalla de Login

2.7.1.2 Diseño de web forms de ingreso de usuarios y equipos

Para la pantalla de ingreso de usuarios se muestran cuadros de texto para el ingreso del nombre, apellido y contraseña, mientras que para el ingreso del rol se tiene una lista desplegable donde se puede escoger entre administrador o docente.



The screenshot shows a web interface for user registration. On the left is a dark sidebar with the 'Manager' title, the 'SREQ' logo, and a signature. It displays the user's session information: 'Sesión iniciada Angel Gabriel por: Jaramillo A.' and a 'LOGOUT' button. Below are three menu items: 'Ingreso de datos' (with a document icon), 'Actualización de datos' (with a pencil icon), and 'Usuario' (with a person icon). The main content area is titled 'Ingreso de usuarios' and contains the heading 'Ingrese los datos de un nuevo usuario'. The form includes a dropdown menu for 'Rol usuario' (set to 'Administrador'), and text input fields for 'Nombre', 'Apellido', and 'Contraseña'. An 'Ingresar' button is positioned at the bottom of the form.

Figura 19. Página para ingreso de usuarios

En cuanto al ingreso de los equipos, solo se requieren los cuadros de texto para el ingreso del código del equipo y su descripción.



The screenshot shows a web interface for equipment registration. The sidebar is identical to the previous form, but the 'Usuario' menu item is not visible. The main content area is titled 'Ingreso de equipos' and contains the heading 'Ingrese los datos de un nuevo equipo'. The form includes two text input fields: 'Código del equipo' and 'Descripción del equipo'. An 'Ingresar' button is positioned at the bottom of the form.

Figura 20. Página para ingreso de equipos

2.7.1.3 Diseño de web forms para actualizar usuarios y equipos.

Con respecto a la actualización de los usuarios, la forma web contiene varios filtros de búsqueda, entre ellos está la búsqueda por nombre, por apellido y por Nick. Los datos se mostrarán en una tabla. Ésta posee una columna con el botón para poder editar los registros.

Manager

SREQ *web*

Sesión iniciada *Angel Gabriel*
por: *Jaramillo A.*

LOGOUT

Ingreso de datos

Actualización de datos

Usuario

Equipo

Actualización de usuarios

Presione editar en el usuario que desea actualizar:

Mostrar todo

Buscar por nombre:

Buscar

Buscar por apellido:

Buscar

Buscar por nick:

Buscar

Rol	Nick	Nombre	Apellido	Clave	Acción
A	AnJa10	Angel Gabriel	Jaramillo A.	12345678	Editar

Figura 21. Página de actualización de usuarios

La página para la actualización de equipos es bastante parecida a la de los usuarios. Se ingresaron cuadros de texto, botones y una lista desplegable para filtrar la información por código, descripción y estado. Además, se pueden mostrar todos los equipos.

Manager

SREQ *Wala*

Sesión iniciada *Angel Gabriel Jaramillo A.*
por: *Jaramillo A.*

LOGOUT

- Ingreso de datos
- Actualización de datos
- Eliminación de datos
- Reportes
- Visualización de datos

Actualización de equipos

Presione editar en el equipo que desea actualizar:

Mostrar todo

Buscar por código: Buscar

Buscar por descripción: Buscar

Buscar por estado: Buscar

Código	Estado	Descripción	
12274	D	ROUTERS CISCO 2901	Editar
12275	D	ROUTERS CISCO 2901	Editar
12820	D	ROUTERS CISCO 2901	Editar
12821	D	ROUTER	Editar
12827	D	FIREWALL	Editar
12828	D	ANTENA WIRELESS	Editar
12834	D	GATEWAY	Editar
12854	D	WIRELESS EXTERNO	Editar
12856	D	TARJETA DE RED	Editar

Figura 22. Página de Actualización de equipos

2.7.1.4 Diseño de web forms para eliminar usuarios y equipos.

Tanto para la eliminación de los usuarios como de los equipos se utilizan exactamente los mismos componentes que se usaron en la actualización de datos, lo único que se modifica es la columna de la acción, que en este caso se muestra un botón para eliminar el usuario o el equipo de la fila seleccionada. A continuación, se muestran pantallas de ambas páginas.

Manager

SREQ *Wala*

Sesión iniciada *Angel Gabriel Jaramillo A.*
por: *Jaramillo A.*

LOGOUT

- Ingreso de datos
- Actualización de datos
- Eliminación de datos
- Usuario
- Equipo

Eliminación de usuarios

Presione eliminar en el usuario que desea borrar:

Mostrar todo

Buscar por nombre: Buscar

Buscar por apellido: Buscar

Buscar por nick: Buscar

Rol	Nick	Nombre	Apellido	Clave	
A	AnJa10	Angel Gabriel	Jaramillo A.	12345678	Eliminar

Figura 23. Página para eliminar usuarios

Manager

SREQ

Sesión iniciada por: *Angel Gabriel Jaramillo A.*

LOGOUT

Ingreso de datos

Actualización de datos

Eliminación de datos

Eliminacion de equipos

Presione eliminar en el equipo que desea borrar:

Mostrar todo

Buscar por código:

Buscar

Buscar por descripción:

Buscar

Buscar por estado:

Buscar

Código	Estado	Descripción	
12274	D	ROUTERS CISCO 2901	Eliminar
12275	D	ROUTERS CISCO 2901	Eliminar
12820	D	ROUTERS CISCO 2901	Eliminar
12821	D	ROUTER	Eliminar
12827	D	FIREWALL	Eliminar

Figura 24. Página para eliminar equipos

2.7.1.5 Diseño de web forms para reportes de prácticas, observaciones y equipos.

En cuanto a los reportes que se generarán, se crearon tres páginas. La primera mostrará las actividades realizadas en un determinado mes y año. Por lo que se implementó un cuadro de texto con formato de fecha, donde se selecciona el mes y año directamente. Para el filtro individual se utilizó una lista desplegable donde sus elementos serán los nombres de todos los usuarios. Con ayuda de los botones se puede mostrar el reporte general e individual una vez que se ingresaron los parámetros. Las prácticas que no hayan sido finalizadas no se mostrarán. En esta forma web se tendrá una tabla para la información de las actividades y otra destinada a las observaciones ingresadas durante estas. Por último, se implementó un botón para que se pueda exportar los datos de las tablas a un archivo PDF.



Figura 25. Página de reporte de actividades.

Con respecto al reporte de las observaciones, se tiene un cuadro de texto para ingresar el mes y año que se desea, y una lista desplegable con los nombres de los usuarios para filtrar las observaciones. En este caso se muestran todas las observaciones sin importar el estado que tengan. Se añadió de la misma forma un botón para exportar la información a un archivo PDF.



Figura 26. Página de reporte de observaciones.

En interfaz para el reporte de equipos se colocó un cuadro de texto para el ingreso de mes y año requeridos. Los patrones de búsqueda en este caso son: el reporte de uso por equipo y los responsables de los equipos utilizados. El resultado de las búsquedas se muestra en una tabla. También, cuenta con un botón para exportar la tabla a PDF.



Figura 27. Página de reporte de equipos

2.7.1.6 Diseño de web forms para visualización de toda la información de la base de datos.

Para la visualización de la información de la base de datos se muestran los registros tal como se encuentran en las tablas. Debido a que la información en la tabla resultante de la relación M:N no tiene datos completamente entendibles, estos fueron adaptados para su comprensión. Cada uno tendrá la opción de visualizar todos los datos o individualmente.

En cuanto a la visualización de los usuarios se generó una lista desplegable para filtrar la información y mostrar individualmente. Los resultados se cargan en una tabla como se muestra en la siguiente figura.



Figura 28. Página de visualización de usuarios

Para la visualización de actividades se muestra también una lista desplegable con los nombres de los usuarios como elementos. Se implementaron los botones para mostrar reportes generales e individuales. La información solicitada se muestra en una tabla. En la siguiente figura se puede apreciar también los elementos del menú en el apartado de visualización de datos.



Figura 29. Página para visualización de actividades.

En la visualización de observaciones se utiliza el mismo concepto de filtrado por usuario o se pueden mostrar todos los datos. La información es desplegada en una tabla y contiene todos los atributos de la entidad.

Manager

SREQ *WOL*

Sesión iniciada por: Angel Gabriel Jaramillo A.

LOGOUT

Visualización observaciones

Para visualizar todas las observaciones: **Mostrar todas**

¡Seleccione el nombre del docente: Angel Gabriel Jaramillo A. ▾ **Mostrar individual**

IDObservacion	IDUsuario	FechaObservacion	DescripciónObservacion	EstadoObservacion
1	1	31/05/2018 22:27:34	Observacion de prueba	Durante actividad: 1

Figura 30. Página de visualización de observaciones.

Por otro lado, la página de visualización de equipos cuenta con un cuadro de texto para filtrar la información por el código. Al igual que todas las visualizaciones, se muestra toda la información en una tabla.

Manager

SREQ *WOL*

Sesión iniciada por: Angel Gabriel Jaramillo A.

LOGOUT

Visualización equipos

Para visualizar todos los equipos: **Mostrar todos**

Ingrese el código del equipo: **Mostrar individual**

idequipo	codigoequipo	estadoequipo	descripcionequipo
1	12274	D	ROUTERS CISCO 2901
2	12275	D	ROUTERS CISCO 2901

Figura 31. Página de visualización de equipos

Finalmente, la información de la página de visualización de los equipos utilizados puede ser filtrada por el usuario que utilizó los equipos, o se pueden mostrar todos los equipos. Para el filtrado por persona responsable se utilizó una lista desplegable con los nombres de los usuarios. Al igual que todas las visualizaciones, la información requerida se muestra en una tabla.

Manager

SREQ *WOL*

Sesión iniciada por: Angel Gabriel Jaramillo A.

LOGOUT

Visualización equipos utilizados

Para visualizar todos los equipos utilizados: **Mostrar todos**

Seleccione el nombre del docente que utilizó los equipos: Angel Gabriel Jaramillo A. ▾ **Mostrar individual**

idequipo	idpractica	codigoequipo	Responsable	fechaprestamo	fechadevolucion
1	1	12274	Angel Gabriel Jaramillo A.	31/05/2018 12:48:50	31/05/2018 12:48:51

Figura 32. Página para visualización de equipos utilizados.

2.7.1.7 Diseño de inicio de sesión para iOS y Android.

Las pantallas para el inicio de sesión que se diseñaron para ambas plataformas cuentan con dos cuadros de texto para el ingreso del *nick* y su contraseña. En el campo de contraseña se oculta los caracteres ingresados.

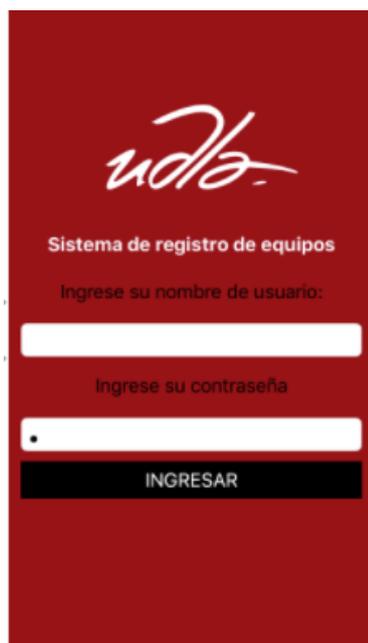
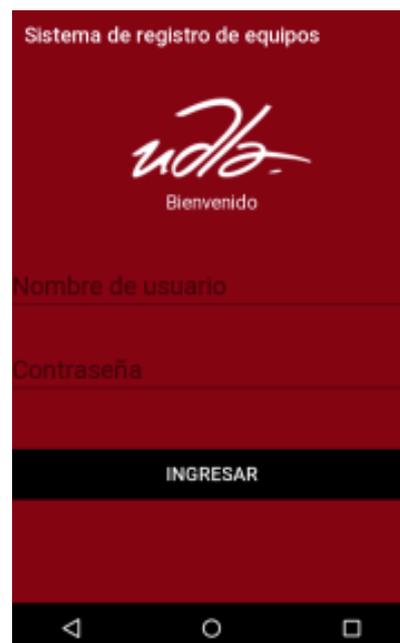


Figura 33. Pantalla Login iOS



34. Pantalla Login Android

2.7.1.8 Diseño de pantalla de inicio para iOS y Android.

En las pantallas de inicio de la aplicación se muestra el nombre y apellido del usuario y un botón para cerrar la sesión. A continuación, se muestra un campo donde se indicará la actividad actual. Consta de cinco botones, los cuales se ocultarán y aparecerán dependiendo de las acciones que se realicen. Cumplen las funciones de ingreso de equipos, devolución de equipos, ingreso de una nueva actividad, finalizar la actividad actual e ingresar una nueva observación.

Para evitar que el usuario vuelva a la pantalla de inicio sin salir de su sesión se bloqueó la navegación hacia la pantalla de *Login*.

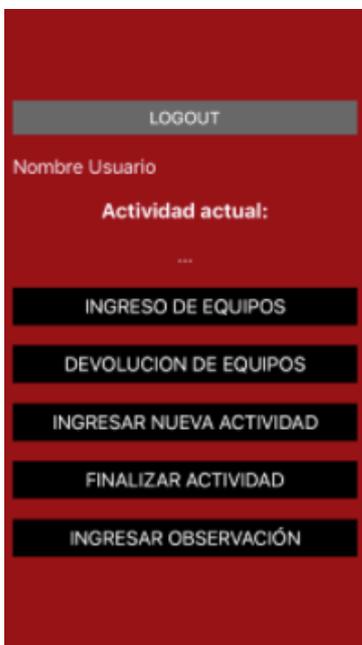


Figura 35. Pantalla de inicio iOS

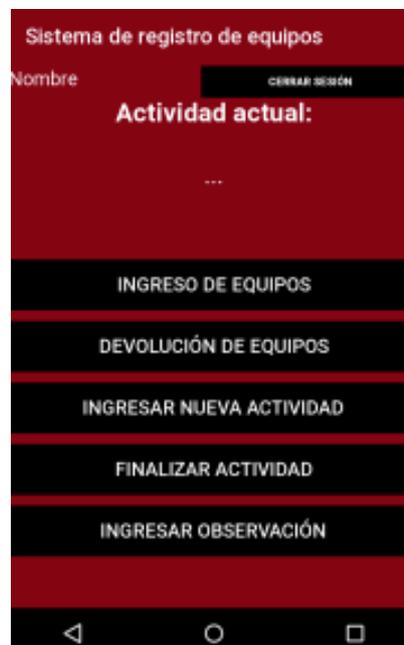


Figura 36. Pantalla de inicio Android

2.7.1.9 Diseño de pantallas de ingreso y devolución de equipos para iOS y Android.

Tanto para el ingreso como para la devolución de equipos se utilizan los mismos elementos, los cuales consisten en una vista donde se mostrará la cámara para leer el código de barras, etiquetas para el código escaneado, su descripción; y una para el número de equipos ingresados en el caso del ingreso, o equipos por devolver en el caso de la devolución. Por último, la pantalla de ingreso cuenta con dos botones, uno para ingresar el equipo y otro para finalizar el proceso de ingreso. Por otro lado, en la página de devolución se tienen dos botones para devolver el equipo y para finalizar la devolución.



Figura 37. Pantalla de ingreso de equipos iOS



Figura 39. Pantalla de devolución de equipos iOS



Figura 38. Pantalla de ingreso de equipos Android



Figura 40. Pantalla de devolución de equipos Android

2.7.1.10 Diseño de pantallas para ingreso de actividades y observaciones para iOS y Android.

Las pantallas para el ingreso de una nueva actividad son simples, solo contienen un cuadro de texto para escribir el tema y un botón para ingresar, una vez que se ingresa se regresa a la pantalla de inicio.



Figura 41. Pantalla de ingreso de actividad iOS



Figura 42. Pantalla de ingreso de actividad Android

Para las pantallas de observación se tiene, de la misma forma, un cuadro de texto para la descripción de la observación y un botón para ingresarla. En este caso la navegación entre vistas o pantallas se habilita debido a que el ingreso de observaciones no afecta en nada el proceso completo.



Figura 43. Pantalla de ingreso de observación iOS



Figura 44. Pantalla de ingreso de observación Android.

Una vez que se finalizaron las tareas planteadas en este Sprint y revisado por todo el equipo Scrum, se determinó con la ayuda del dueño del producto que el segundo Sprint finalizó y se entregó el incremento.

2.8 Tercer Sprint

El tercer Sprint se llevó a cabo del 25 de abril al 1 de mayo de 2018, cumpliendo con lo programado en el cronograma de desarrollo. Se crearon los métodos para realizar las operaciones de ingreso, actualización, eliminación y visualización de las tablas.

Tabla 42.

Tercer Sprint

Sprint	Actividad	Tareas	Tiempo (Sem)	Responsable
---------------	------------------	---------------	---------------------	--------------------

		3.1. Crear capa de negocio y de datos con conexión a la base de datos en la aplicación web.	
		3.2. Implementar métodos para ingresar datos de la aplicación web.	
3	CRUD de la base de datos	3.3. Implementar métodos para eliminar información de las tablas.	1
		3.4. Implementar métodos para editar las tablas.	
		3.5. Implementar métodos para leer todas las tablas.	

Juan José Bahamonde

2.8.1 Crear capa de negocio y de datos con conexión a la base de datos en la aplicación web

De acuerdo con la teoría detallada en el Capítulo I, con el fin de mejorar la seguridad de la aplicación web y aislar las operaciones para un mejor desarrollo y mantenimiento, se sumaron las capas de presentación, negocio y datos.

En la capa de negocio se implementaron cinco clases, correspondientes a cada una de las tablas que existe en la base de datos. En cada una se definieron las variables correspondientes a los atributos de las entidades, y se las encapsularon para poder establecer y recuperar la información. En estas clases se creó el vínculo entre la capa de datos y la capa de presentación, el mismo que permite el manejo de la información recuperada desde la base de datos y enviarla a la capa de presentación para mostrar los datos solicitados en la interfaz.

En cuanto a la capa de datos se crearon cinco clases, una por cada tabla de la base de datos. Cada una de ellas cuenta con métodos para conectarse de forma local. Se utilizó el controlador para MySQL Connector/NET, el cual permite emplear ADO.NET Entity Framework, proporcionando una conectividad de alto rendimiento y segura con la base de datos en MySQL Server. Una vez que se recuperan los datos solicitados son enviados a la capa de negocio.

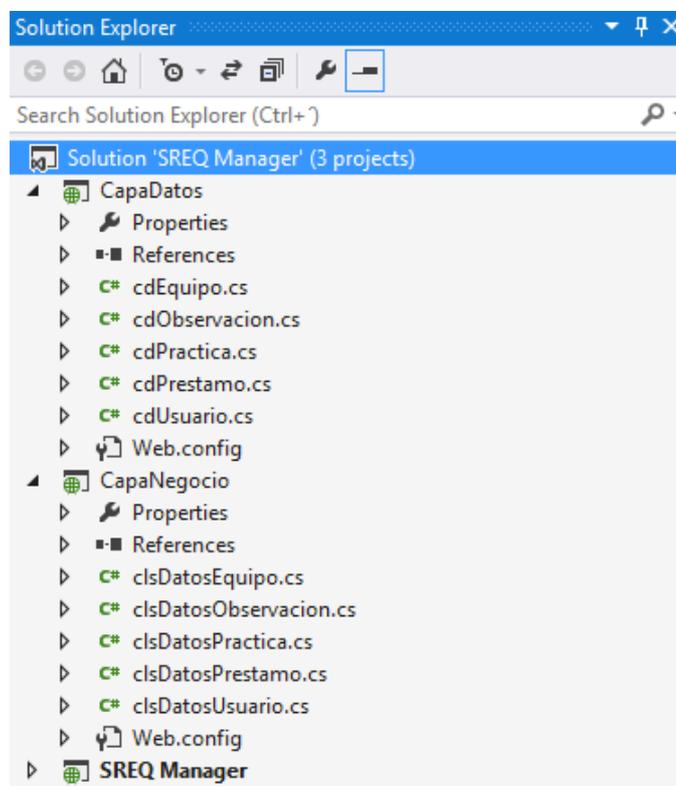


Figura 45. Capas de la aplicación web

2.8.2 Implementar métodos para ingresar datos de la aplicación web

Como se definieron en las historias de usuario, el ingreso que se realizará en la aplicación web solo será de usuarios y de equipos. Para esto se recuperan los datos de los cuadros de texto definidos en la capa de presentación y gracias a la capa de datos se utilizan los procedimientos almacenados para el ingreso de usuarios y de equipos.

Una vez implementados los métodos en las tres capas se obtuvo el siguiente resultado:

Manager

SREQ *udla*

Sesión iniciada Angel Gabriel
por: Jaramillo A.

LOGOUT

Ingreso de datos

Actualización de datos

Ingreso de usuarios

Ingrese los datos de un nuevo usuario

Rol usuario: Administrador

Nombre: Juan

Apellido: Bahamonde

Contraseña: 1234qwer

Ingresar

Figura 46. Ingreso de usuario

```
1 • select * from usuario;
```

IDUsuario	RolUsuario	NickUsuario	NombreUsuario	ApellidoUsuario	ClaveUsuario
1	A	AnJa10	Anael Gabriel	Jaramillo A.	12345678
2	A	JuBa10	Juan	Bahamonde	1234qwer
NULL	NULL	NULL	NULL	NULL	NULL

Figura 47. Registro de usuario ingresado

Los cuadros de texto para el ingreso del usuario cuentan con validadores para que no se ingresen campos en blanco, y para la contraseña debe tener exactamente ocho caracteres alfanuméricos. Las siguientes figuras muestran el ingreso de un nuevo equipo.

Manager

SREQ

Sesión iniciada por: Angel Gabriel Jaramillo A.

LOGOUT

Ingreso de datos

Ingreso de equipos

Ingrese los datos de un nuevo equipo

Código del equipo: 12345

Descripción del equipo: Equipo ingreso

Ingresar

Figura 48. Ingreso de equipo

```
select * from usuario; select * from equipo;
```

IDEquipo	CodigoEquipo	EstadoEquipo	DescripcionEquipo
213	62151	D	UPS
214	12345	D	Equipo ingreso

Figura 49. Registro de ingreso de equipo

Adicionalmente, si un administrador desea realizar el ingreso de equipos mediante el lector de código de barras lo puede hacer con la aplicación móvil. Una vez que ingresa con sus credenciales, en la pantalla de inicio se muestra un botón extra que permite registrar un nuevo equipo en el inventario. Esta función permite leer y capturar un código de barras, para después asignar una descripción al equipo y guardarlo en la base de datos mediante servicios web y

el procedimiento almacenado para el ingreso de equipos. Este proceso se muestra a continuación en las figuras.

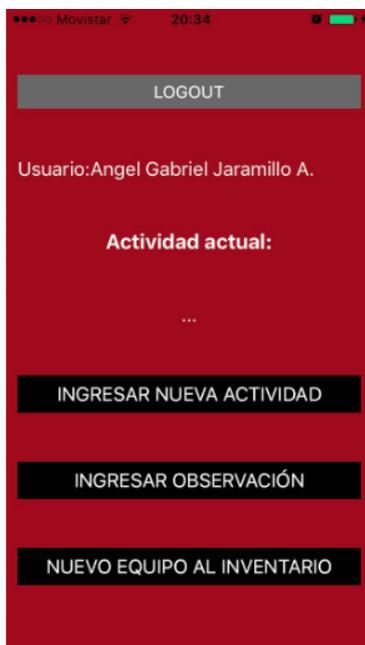


Figura 50. Pantalla de inicio de aplicación móvil para administradores.



Figura 51. Ingreso de equipos al inventario mediante aplicación móvil



Figura 52. Mensaje de confirmación de ingreso de equipo



Figura 53. Verificación de ingreso de equipo mediante aplicación móvil

Con el fin de mantener la integridad de la información de la base de datos, se realizaron ciertas validaciones al momento de ingresar un nuevo equipo al inventario. En primer lugar, al leer un código de barras se verifica que éste no se encuentre en la base de datos, para eso se utiliza un servicio web que realiza una búsqueda del código leído. En el caso de que el código ya exista, no se puede ingresar el nuevo registro como se muestra en la figura 55.

Finalmente, en el caso de que se lea un código y no se ingrese su respectiva descripción, no es posible ingresar el nuevo equipo, debido a que este campo no puede estar vacío o ser nulo.



Figura 54. Verificación de ingreso de descripción



Figura 55. Verificación de código de barras existente

2.8.3 Implementar métodos para eliminar información de las tablas

De la misma manera que el ingreso, la eliminación de tablas se la podrá realizar solo de las entidades Usuario y Equipo. En este caso, se utilizó el botón de la columna de acciones de la tabla para eliminar. El primer paso fue codificar los métodos para los filtros de información que en el caso de los usuarios se puede mostrar todos los registros, buscar por nombre, apellido o *Nick*. Una vez que se aplican los filtros se carga la tabla con la información solicitada. A continuación, se presenta el resultado de haber filtrado por apellido y eliminado el registro.



Manager

Sesión iniciada por: Angel Gabriel Jaramillo A.

LOGOUT

Ingreso de datos

Actualización de datos

Eliminación de usuarios

Presione eliminar en el usuario que desea borrar:

Mostrar todo

Buscar por nombre:

Buscar

Buscar por apellido: Bahamonde

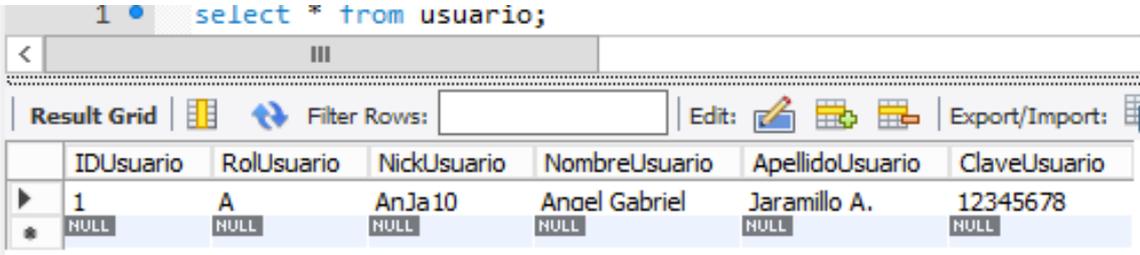
Buscar

Buscar por nick:

Buscar

Rol	Nick	Nombre	Apellido	Clave	
A	JuBa10	Juan	Bahamonde	1234qwer	Eliminar

Figura 56. Eliminación de usuario



```
1 • select * from usuario;
```

Result Grid

Filter Rows:

Edit:

Export/Import:

IDUsuario	RolUsuario	NickUsuario	NombreUsuario	ApellidoUsuario	ClaveUsuario
1	A	AnJa10	Angel Gabriel	Jaramillo A.	12345678
NULL	NULL	NULL	NULL	NULL	NULL

Figura 57. Registro de usuario eliminado

Para la eliminación de los equipos se puede realizar una búsqueda por código, descripción o por estado. En este caso los equipos no pueden ser eliminados del

registro si aún no se han devuelto, es decir, su estado es “P”. En las siguientes figuras se muestra el proceso de eliminación realizando un filtrado por código.

Manager

SREQ

Sesión iniciada por: *Angel Gabriel Jaramillo A.*

LOGOUT

Ingreso de datos

Eliminacion de equipos

Presione eliminar en el equipo que desea borrar: **Mostrar todo**

Buscar por código: 12345 **Buscar**

Buscar por descripción: **Buscar**

Buscar por estado: Seleccione... **Buscar**

Código	Estado	Descripción	Acción
12345	D	Equipo ingreso	Eliminar

Figura 58. Eliminación de equipo

```
1 • select * from equipo;
```

Result Grid | Filter Rows: | Edit:

IDEquipo	CodigoEquipo	EstadoEquipo	DescripcionEquipo
213	62151	D	UPS
NULL	NULL	NULL	NULL

Figura 59. Registro de eliminación de equipo

2.8.4 Implementar métodos para editar las tablas

Para la actualización de los datos de usuarios y equipos se emplearon los mismos filtros de búsqueda. La diferencia está en la columna de acción, la cual tiene dos estados. El primero muestra un botón para seleccionar el registro a editar; el segundo transforma cada uno de los campos en cuadros de texto para la edición y muestra botones para guardar los cambios o para cancelar la edición y volver al primer estado.

Para el usuario se empleará un filtro por nombre y se modificará la contraseña del usuario.

Manager

SREQ *ndo*

Sesión iniciada por: Angel Gabriel Jaramillo A.

LOGOUT

Ingreso de datos

Actualización de usuarios

Presione editar en el usuario que desea actualizar: **Mostrar todo**

Buscar por nombre: Angel **Buscar**

Buscar por apellido: **Buscar**

Buscar por nick: **Buscar**

Rol	Nick	Nombre	Apellido	Clave	Acción
A	AnJa10	Angel Gabriel	Jaramillo A.	12345678	Editar

Figura 60. Búsqueda para actualización de usuarios

Una vez que se inicia la edición, los cuadros de texto cuentan con validadores, por lo que no se puede dejar los campos en blanco. En el caso del rol de usuario solo se puede ingresar la letra “A” para administrador y “D” para docente. En cuanto a la contraseña se debe ingresar exactamente los ocho caracteres alfanuméricos requeridos.

Manager

SREQ *ndo*

Sesión iniciada por: Angel Gabriel Jaramillo A.

LOGOUT

Ingreso de datos

Actualización de datos

Actualización de usuarios

Presione editar en el usuario que desea actualizar: **Mostrar todo**

Buscar por nombre: Angel **Buscar**

Buscar por apellido: **Buscar**

Buscar por nick: **Buscar**

Rol	Nick	Nombre	Apellido	Clave	Acción
A	AnJa10	Angel Gabriel	Jaramillo A.	qwertyu	Actualizar Cancelar

*8 caracteres alfanumericos

Figura 61. Actualización de usuario

Una vez hecha la actualización se vuelve a cargar la información desde la base de datos y se puede visualizar el cambio inmediato y un mensaje que muestra el resultado de la acción.

Rol	Nick	Nombre	Apellido	Clave	Acción
A	AnJa10	Angel Gabriel	Jaramillo A.	qwertyui	Editar

Figura 62. Registro de actualización de usuario

El procedimiento es bastante parecido para la actualización de los equipos como se a continuación:

Código	Estado	Descripción	Acción
12274	D	ROUTERS CISCO 2901	Editar

Figura 63. Búsqueda para actualización de equipo

En edición los cuadros de texto poseen igual validadores de datos, en el caso del estado del equipo solo se puede ingresar si está disponible "D" o si fue prestado "P".

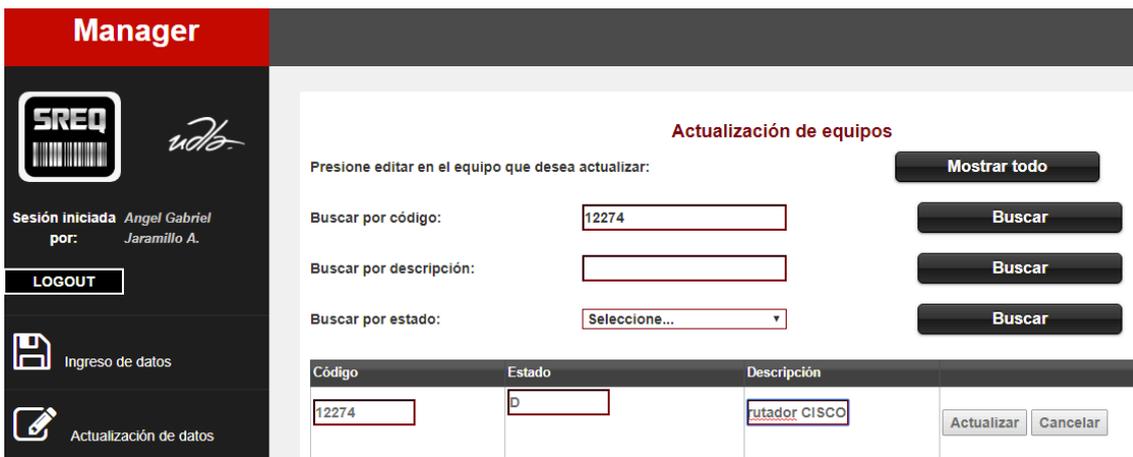


Figura 64. Actualización de equipo

Una vez que se actualiza se vuelve a cargar la información de la base de datos y se muestra el mensaje de actualización.



Figura 65. Registro de equipo actualizado

2.8.5 Implementar métodos para leer todas las tablas

Para realizar esta acción no hizo falta el uso de procedimientos almacenados a excepción de la visualización de los equipos prestados. Para cada una de las tablas se implementó un método para recuperar todos los registros y un solo filtro.

La visualización de los usuarios cuenta con un filtro para mostrar registros individualmente, para esto se cargan todos los usuarios en la lista desplegable desde la base de datos.

Manager

SREQ *udla*

Sesión iniciada por: Angel Gabriel Jaramillo A.

LOGOUT

Visualización usuarios

Para visualizar todos los usuarios: **Mostrar todos**

Seleccione el nombre del docente: **Angel Gabriel Jaramillo A.** **Mostrar individual**

IDUsuario	RolUsuario	NickUsuario	NombreUsuario	ApellidoUsuario	ClaveUsuario
1	A	AnJa10	Angel Gabriel	Jaramillo A.	qwertyui

Figura 66. Visualización de usuarios

1 • `Select * from usuario;`

Result Grid | Filter Rows: | Edit: | Export/Import:

IDUsuario	RolUsuario	NickUsuario	NombreUsuario	ApellidoUsuario	ClaveUsuario
1	A	AnJa10	Angel Gabriel	Jaramillo A.	qwertyui

Figura 67. Verificación de tabla Usuario

La visualización de las actividades y observaciones se la realiza de la misma manera, se pueden mostrar todos los registros o se emplea el mismo filtro, es decir, se busca las actividades u observaciones realizadas por un usuario determinado.

Manager

SREQ *udla*

Sesión iniciada por: Angel Gabriel Jaramillo A.

LOGOUT

Visualización actividades

Para visualizar todas las prácticas: **Mostrar todas**

Seleccione el nombre del docente: **Angel Gabriel Jaramillo A.** **Mostrar individual**

Actividad	Usuario	Tema	Inicio	Fin
1	1	Prueba	31/05/2018 12:48:50	31/05/2018 12:48:51

Figura 68. Visualización de actividades

The screenshot shows a database query window with the SQL statement `Select * from practica;`. Below the query, there is a toolbar with options like 'Result Grid', 'Filter Rows', 'Edit', and 'Export/Import'. The main area displays a table with the following data:

IDPractica	IDUsuario	TemaPractica	FechaInicioPractica	FechaFinPractica
1	1	Prueba	2018-05-31 12:48:50	2018-05-31 12:48:51
NULL	NULL	NULL	NULL	NULL

Figura 69. Verificación de tabla Práctica

The screenshot shows the 'Manager' web application interface. On the left, there is a sidebar with the 'SREQ' logo, a user profile for 'Angel Gabriel Jaramillo A.', and a 'LOGOUT' button. The main content area is titled 'Visualización observaciones' and includes a dropdown menu for selecting a teacher, currently set to 'Angel Gabriel Jaramillo A.'. Below this, there is a table of observations:

IDObservacion	IDUsuario	FechaObservacion	DescripcionObservacion	EstadoObservacion
1	1	31/05/2018 22:27:34	Observacion de prueba	Durante actividad: 1

Figura 70. Visualización de observaciones

The screenshot shows a database query window with the SQL statement `Select * from observacion;`. Below the query, there is a toolbar with options like 'Result Grid', 'Filter Rows', 'Edit', and 'Export/Import'. The main area displays a table with the following data:

IDObservacion	IDUsuario	FechaObservacion	DescripcionObservacion	EstadoObservacion
1	1	2018-05-31 22:27:34	Observacion de prueba	Durante actividad: 1
NULL	NULL	NULL	NULL	NULL

Figura 71. Verificación de tabla Observacion

Para la visualización de los equipos se pueden mostrar todos, o realizar el filtrado por el código del equipo.

Manager

SREQ

Sesión iniciada por: *Angel Gabriel Jaramillo A.*

LOGOUT

Ingreso de datos

Actualización de datos

Visualización equipos

Para visualizar todos los equipos: **Mostrar todos**

Ingrese el código del equipo: **Mostrar individual**

idequipo	codigoequipo	estadoequipo	descripcionequipo
1	12274	D	Enrutador CISCO 2901
2	12275	D	ROUTERS CISCO 2901
3	12820	D	ROUTERS CISCO 2901
4	12821	D	ROUTER
5	12827	D	FIREWALL
6	12828	D	ANTENA WIRELESS

Figura 72. Visualización de equipos

```
Select * from equipo;
```

Result Grid | Filter Rows: | Edit:

IDEquipo	CodigoEquipo	EstadoEquipo	DescripcionEquipo
1	12274	D	Enrutador CISCO 2901
2	12275	D	ROUTERS CISCO 2901
3	12820	D	ROUTERS CISCO 2901
4	12821	D	ROUTER
5	12827	D	FIREWALL
6	12828	D	ANTENA WIRELESS

Figura 73. Verificación de tabla Equipo

Finalmente, para mostrar los equipos que han sido prestados se utiliza la vista para mostrar todos y el procedimiento almacenado para mostrar individualmente el préstamo de equipos. Por lo que se tiene información adicional sobre el equipo y el usuario que lo tomó prestado. Para el filtro de búsqueda se emplea la lista desplegable cargando los nombres de los usuarios desde la base de datos.

The screenshot shows the 'Manager' application interface. On the left, there is a sidebar with the 'SREQ' logo, a signature, and user information: 'Sesión iniciada Angel Gabriel Jaramillo A.' and a 'LOGOUT' button. The main content area is titled 'Visualización equipos utilizados'. It includes a button 'Mostrar todos' and a dropdown menu for selecting the teacher, currently set to 'Angel Gabriel Jaramillo A.', with a 'Mostrar individual' button. Below this is a table with the following data:

idequipo	idpractica	codigoequipo	Responsable	fechaprestamo	fechadevolucion
1	1	12274	Angel Gabriel Jaramillo A.	31/05/2018 12:48:50	31/05/2018 12:48:51

Figura 74. Visualización de equipos utilizados

The screenshot shows a SQL query execution tool. The query entered is 'Select * from practica_equipo;'. The result grid displays the following data:

IDEquipo	IDPractica	FechaPrestamo	FechaDevolucion
1	1	2018-05-31 12:48:50	2018-05-31 12:48:51
NULL	NULL	NULL	NULL

Figura 75. Verificación de tabla Practica_Equipo

Después de culminar con las tareas del tercer Sprint, y tener el Sprint Review con el equipo se presentó el incremento. De acuerdo con los criterios de aceptación de la historia de usuario HU002 que se muestran en la Tabla 7, se determinó que se cumplieron por completo. Por lo tanto, el tercer Sprint es considerado como terminado y el incremento es completamente funcional.

2.9 Cuarto Sprint

El cuarto Sprint fue desarrollado del 2 de mayo de 2018 al 8 de mayo de 2018, como se planteó en el cronograma de desarrollo inicial. Se implementaron los métodos para la generación de reportes mensuales de equipos, actividades y observaciones.

Tabla 43.

Cuarto Sprint

Sprint	Actividad	Tareas	Tiempo (Sem)	Responsable
4	Generación de reportes	4.1. Crear método para mostrar reporte de observaciones.	1	Juan José Bahamonde
		4.2. Crear método para mostrar reportes de actividades.		
		4.3. Crear método para mostrar reporte de equipos.		
		4.4. Crear métodos para exportar datos a PDF.		

2.9.1 Crear método para mostrar reporte de observaciones

Para el reporte de observaciones se utilizarán los procedimientos almacenados determinados en el primer Sprint, para buscar por un mes y año determinado. En la capa de datos se implementaron los métodos necesarios para llamar a los procedimientos y recuperar la información para después mostrarla en la tabla. Se muestra el número de observaciones que se hicieron, su fecha y hora, el estado y el usuario que la realizó. En este reporte se visualizan las observaciones con estado tanto fuera como dentro de la actividad. A continuación, se exponen los resultados de generar un reporte individual.

Manager

SREQ *udla*

Sesión iniciada Angel Gabriel Jaramillo A.
por: Jaramillo A.

LOGOUT

Reporte de observaciones

Ingrese el mes y año: **Reporte general**

Seleccione el nombre del docente: **Reporte individual**

Nro	Descripción	Fecha	Estado	Docente
1	Observacion de prueba	31/05/2018 22:27:34	Durante actividad: 1	Angel Gabriel Jaramillo A.

Figura 76. Reporte de observaciones

2.9.2 Crear método para mostrar reportes de actividades.

En cuanto al reporte de las actividades, se mostrarán dos tablas, una de las acciones y otra con las observaciones que se realizaron durante esas actividades. Se presentará un reporte general o individual de acuerdo con el mes y año determinados. La implementación fue sencilla debido a que solo se tuvo que llamar a los procedimientos almacenados desde la capa de datos y posteriormente enviar la información a la capa de presentación. A continuación, se observa el resultado de la generación de reportes.

Manager

SREQ *udla*

Sesión iniciada Angel Gabriel Jaramillo A.
por: Jaramillo A.

LOGOUT

Ingreso de datos

Actualización de datos

Eliminación de datos

Reportes

Reporte de actividades

Ingrese el mes y año: **Reporte general**

Seleccione el nombre del docente: **Reporte individual**

Nro	Actividad	Nombre	Apellido	Tema	Inicio	Fin	Duracion
1	1	Angel Gabriel	Jaramillo A.	Prueba	31/05/2018 12:48:50	31/05/2018 12:48:51	00:00:01

Observacion	Usuario	Fecha	Descripción	Estado
1	1	31/05/2018 22:27:34	Observacion de prueba	Durante actividad: 1

Figura 77. Reporte de actividades

2.9.3 Crear método para mostrar reporte de equipos.

Al igual que los otros reportes, se llamaron a los procedimientos almacenados de reporte general e individual desde la capa de datos y se envió la información ya procesada a la capa de presentación. Se tienen dos tipos de reporte: reporte de uso y reporte de responsable. El primero, indica los equipos y el tiempo de uso que se le dieron en un determinado mes y año. El otro, muestra los usuarios que tomaron prestados los equipos en el mes especificado, la fecha, hora de préstamo, devolución y el código.

Manager

SREQ

Sesión iniciada por: Angel Gabriel Jaramillo A.

LOGOUT

Ingreso de datos

Reporte de equipos

Ingrese el mes y año:

Generar reporte de uso de equipos:

Generar reporte de los responsables de los equipos:

Nro	Descripción	Codigo	Tiempo_Uso
1	Enrutador CISCO 2901	12274	00:00:01

Figura 78. Reporte de uso de equipos

Manager

SREQ

Sesión iniciada por: Angel Gabriel Jaramillo A.

LOGOUT

Ingreso de datos

Reporte de equipos

Ingrese el mes y año:

Generar reporte de uso de equipos:

Generar reporte de los responsables de los equipos:

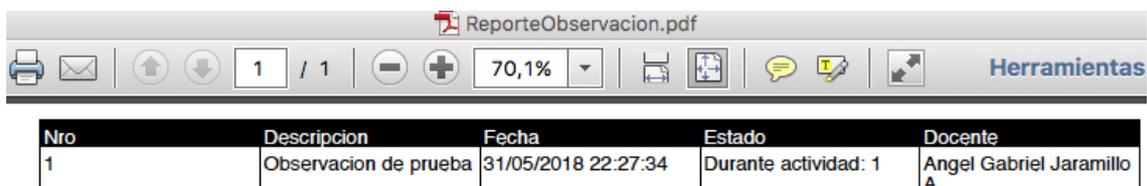
Responsable	Equipo	Codigo	Prestamo	Devolucion
Angel Gabriel Jaramillo A.	Enrutador CISCO 2901	12274	31/05/2018 12:48:50	31/05/2018 12:48:51

Figura 79. Reporte de responsables de equipos

2.9.4 Crear métodos para exportar datos a PDF

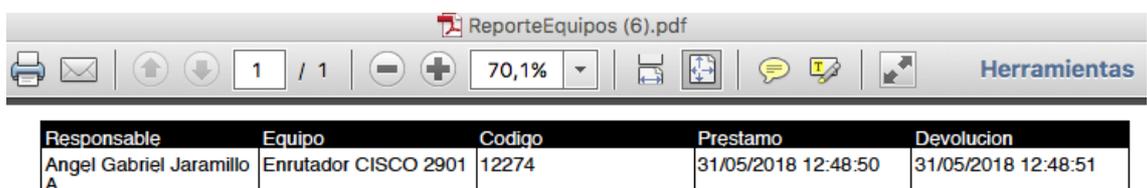
Finalmente, para exportar los datos de los reportes en cada uno de los apartados se elaboró un método para generar un documento PDF con las tablas que se

muestran en la aplicación web. Para esto se importó la librería iTextSharp que funciona con la plataforma .NET y permite generar archivos en varios formatos de forma sencilla. Una vez implementados los métodos necesarios para actividades, observaciones y usuarios se obtuvieron los siguientes resultados:



Nro	Descripcion	Fecha	Estado	Docente
1	Observacion de prueba	31/05/2018 22:27:34	Durante actividad: 1	Angel Gabriel Jaramillo A.

Figura 80. Reporte en PDF de observaciones



Responsable	Equipo	Codigo	Prestamo	Devolucion
Angel Gabriel Jaramillo A.	Enrutador CISCO 2901	12274	31/05/2018 12:48:50	31/05/2018 12:48:51

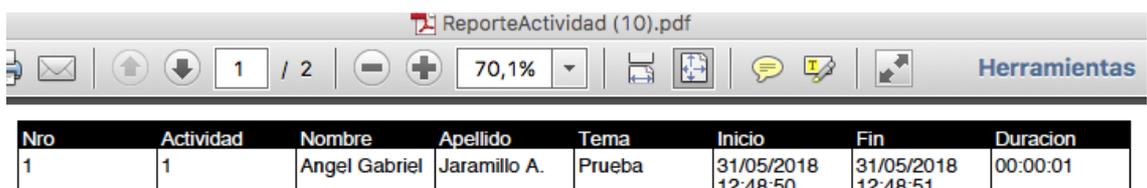
Figura 81. Reporte en PDF de responsable de equipos



Nro	Descripcion	Codigo	Tiempo Uso
1	Enrutador CISCO 2901	12274	00:00:01

Figura 82. Reporte en PDF de uso de equipos

En el caso del reporte de actividades, al ser dos tablas, se ingresa primero la tabla de las actividades y con un salto de página se ingresa la tabla de las observaciones realizadas en la o las actividades.



Nro	Actividad	Nombre	Apellido	Tema	Inicio	Fin	Duracion
1	1	Angel Gabriel	Jaramillo A.	Prueba	31/05/2018 12:48:50	31/05/2018 12:48:51	00:00:01

Figura 83. Reporte en PDF de actividades, tabla de actividades.

Observacion	Usuario	Fecha	Descripcion	Estado
1	1	31/05/2018 22:27:34	Observacion de prueba	Durante actividad: 1

Figura 84. Reporte en PDF de actividades, tabla de observaciones

Una vez finalizadas las tareas planeadas para el cuarto Sprint se tuvo la revisión del Sprint y la entrega del incremento respectivo. Gracias a que se cumplieron los criterios de aceptación descritos en el reverso de la historia de usuario HU003 de la tabla 9, el cuarto Sprint es considerado como terminado y se implementa en ambiente de producción con una funcionalidad completa.

2.10 Quinto Sprint

El quinto Sprint fue desarrollado del 9 de mayo de 2018 al 15 de mayo de 2018 como se estableció en el cronograma de desarrollo inicial. Se crearon la mayoría de las funcionalidades de las aplicaciones móviles en iOS y Android. Las actividades realizadas se detallan en la siguiente tabla.

Tabla 44.

Quinto Sprint

Sprint	Actividad	Tareas	Tiempo (Sem)	Responsable
5	Ingreso de nueva actividad.	5.1. Crear servicios web para conectarse a la base de datos para iOS y Android. 5.2. Implementar métodos para ingresar	1	Juan José Bahamonde

una actividad en
Android y en iOS.

5.3. Implementar
métodos para
recuperar la actividad
actual.

5.4. Implementar
método para finalizar
una actividad.

5.5. Implementar lector
de código de barras en
Android y en iOS.

Ingreso y
devolución
de equipo

5.6. Implementar
método para leer
equipo por código
ingresado en Android
y en iOS.

5.7. Implementar
servicio web y método
para ingresar la
información del equipo
escaneado en Android
y en iOS.

5.8. Implementar
métodos para
devolver equipos.

Las funcionalidades de ambas aplicaciones son exactamente las mismas, por lo que en este documento se evidenciará únicamente de las funcionalidades de iOS. El código de ambas aplicaciones se encuentra adjunto en el CD en la carpeta de anexos.

2.10.1 Crear servicios web para conectarse a la base de datos para iOS y Android.

Para la creación de servicios web se utilizó la versión 7.2.4 de PHP, y se emplearon las funcionalidades de la extensión *mysqli*. Esta extensión fue desarrollada para explotar al máximo las nuevas funcionalidades de MySQL Server.

Se desarrollaron servicios web independientes para las dos plataformas, con el fin de acoplarse y facilitar el desarrollo en cada una de ellas. Se utilizaron los métodos de envío GET y POST para realizar las peticiones a la base de datos desde las aplicaciones.

Gracias a que los procedimientos almacenados creados realizan todas las funciones necesarias, simplemente fueron invocados en cada servicio web dependiendo de su funcionalidad. En total se crearon siete servicios web para cada aplicación cuyas funcionalidades son las siguientes:

- Inicio de sesión
- Ingreso de actividad
- Ingreso de observación

- Préstamo y devolución de equipos
- Recuperar datos de equipo
- Recuperar datos de la actividad actual
- Finalizar actividad

Todos los servicios web se encuentran adjuntos en el CD en la carpeta de anexos.

2.10.2 Implementar métodos para ingresar una actividad en Android y en iOS

Al presionar el botón ingresar actividad se muestra una pantalla con un cuadro de texto donde se ingresa el tema. Una vez introducido el texto, al seleccionar el botón de ingreso se conecta con el servicio web y se ingresa en la base de datos.

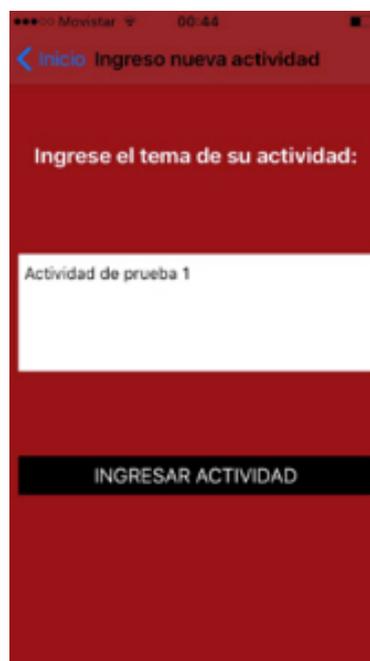


Figura 85. Ingreso de actividad

The screenshot shows a database query interface with the SQL statement `select * from practica`. Below the query, there is a toolbar with options like 'Result Grid', 'Filter Rows', 'Edit', and 'Export/Import'. The main area displays a table with the following data:

IDPractica	IDUsuario	TemaPractica	FechaInicioPractica	FechaFinPractica
1	1	Actividad de prueba 1	NULL	NULL
NULL	NULL	NULL	NULL	NULL

Figura 86. Actividad ingresada en base de datos

2.10.3 Implementar métodos para recuperar la práctica actual

Una vez que se ingresa la actividad y se almacena en la base de datos, automáticamente se ejecuta un método mediante un servicio web para recuperar la información de la actividad, es decir, su identificador y el tema. Esta información se almacena en las variables de la base de datos del teléfono. En el caso de Android se utilizaron SharedPreferences y en iOS, UserDefaults. Ambos cumplen la misma función para almacenar y recuperar la información, la cual es permanente a menos que se elimine la aplicación.

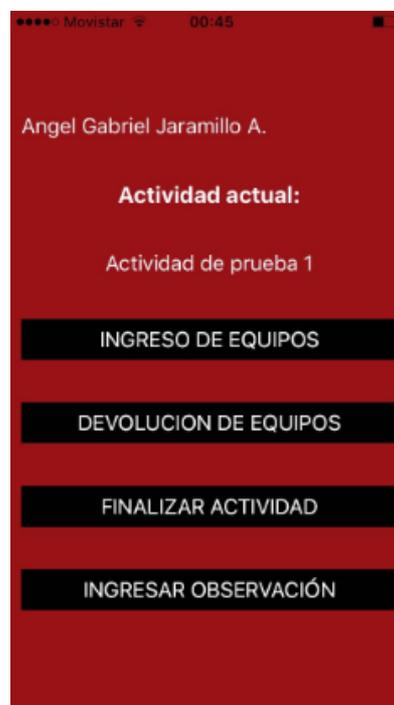


Figura 87. Recuperación de información de la actividad

2.10.4 Implementar lector de código de barras en Android y en iOS.

El primer paso para crear el lector de código de barras en las aplicaciones es solicitar los permisos necesarios para acceder a la cámara. Sin estos permisos que son solicitados al abrirse la aplicación por primera vez, no se puede ejecutar ninguna acción.

Para la implementación del lector de código de barras en iOS se importó la librería de AVFoundation Framework. Esta permite crear sesiones de captura de video y añadir la captura de objetos de metadatos, que en este caso es el código de barras. Una vez implementado, es posible leer todos los tipos de código de barras existentes y además código QR.

En el caso de Android se utilizó la API Google Play Services versión 11.0.4. Esta permite agregar una fuente de cámara, la vista o superficie para la captura de video y un detector de código de barras. Al igual que iOS permite la lectura de todos los formatos de código de barras.

En ambos casos, el detector de código de barras tiene enfoque automático. A continuación, se muestra un ejemplo de la lectura de un código:



Figura 88. Lector de código de barras

2.10.5 Implementar método para leer equipo por código ingresado en Android y en iOS.

Una vez implementado el lector de código de barras se crearon los métodos para recuperar la información del código ingresado. Para esto se emplearon los servicios web creados, donde se envía el código del equipo y retorna los datos de este. Esta información es su identificador, el código, la descripción y el estado.

Después de haber leído el código, si se encuentra en la base de datos, se hace una captura de imagen y se habilita el botón para ingresar el equipo. La metodología para la devolución de equipos es exactamente la misma en cuanto a la lectura de códigos.



Figura 89. Lectura del código desde la base de datos

2.10.6 Implementar método para ingresar la información del equipo escaneado en Android y en iOS.

Para registrar el préstamo del equipo, una vez que ya se ha leído el código de barras, se ha verificado que se encuentra en la base de datos y que además está disponible, se utiliza el botón de ingreso de equipo. La función que cumple este botón es utilizar el servicio web de préstamo de equipos y envía el identificador del equipo y el identificador de la actividad actual, almacenando la información en la base de datos. Por último, la variable del sistema que lleva la cuenta de los equipos ingresados aumenta en una unidad.



Figura 90. Ingreso de equipo a actividad

```
1 • select * from practica_equipo
```

	IDEquipo	IDPractica	FechaPrestamo	FechaDevolucion
▶	7	1	2018-06-05 01:29:06	NULL
*	NULL	NULL	NULL	NULL

Figura 91. Préstamo de equipo ingresado en la base de datos

2.10.7 Implementar métodos para devolver equipos.

Para garantizar que el equipo es devuelto, en la página de inicio se bloquea el uso del botón para finalizar la práctica hasta que el contador de equipos ingresados llegue a cero nuevamente. Tampoco es posible que el usuario cierre sesión. Por lo que se deben devolver todos los equipos prestados.



Figura 92. Pantalla de inicio con equipos por devolver

Para la devolución de los equipos se emplea el mismo reconocimiento del código de barras, pero en este caso no se puede regresar un equipo que se encuentre disponible, solo los que han sido prestados. Si cumple con requisitos, se hace una captura y se habilita el botón para devolver el equipo.



Figura 93. Lectura de código en devolución de equipos

Al presionar el botón se utiliza el mismo servicio web del ingreso, debido a que el procedimiento almacenado cumple ambas funciones. Una vez que se devuelve, se actualiza la fecha y hora de devolución del equipo y el contador de equipos ingresados disminuye en uno.



Figura 94. Devolución de equipo

The screenshot shows a database query result grid. The query is 'select * from practica_equipo'. The result grid has the following data:

IDEquipo	IDPractica	FechaPrestamo	FechaDevolucion
7	1	2018-06-05 01:29:06	2018-06-05 01:31:45
NULL	NULL	NULL	NULL

Figura 95. Verificación de equipo devuelto en base de datos

2.10.8 Implementar método para finalizar actividad.

En la pantalla de inicio, una vez que se han devuelto todos los equipos, y la variable del sistema que lleva la cuenta de equipos ingresados llega a cero y se activa el botón para finalizar la práctica.



Figura 96. Pantalla de inicio sin equipos por devolver

Al finalizar la actividad, los botones vuelven a la configuración de inicio donde solo puede ingresar una nueva actividad o una observación. Además, el usuario ya tiene permitido cerrar su sesión. Por último, mediante el servicio web para finalizar la actividad se ingresan las fechas de inicio y fin correspondientes, tomando en cuenta el ingreso del primer equipo y la devolución del último equipo.



Figura 97. Actividad finalizada

1 • `select * from practica`

IDPractica	IDUsuario	TemaPractica	FechaInicioPractica	FechaFinPractica
1	1	Actividad de prueba 1	2018-06-05 01:29:06	2018-06-05 01:31:45
NULL	NULL	NULL	NULL	NULL

Figura 98. Verificación de practica finalizada en base de datos

Después de terminar las tareas planteadas para este Sprint y tener la reunión donde se presenta el incremento y se ofrece una revisión de este, se concluyó que, gracias a que cumple con los criterios de aceptación tanto de la historia de usuario HU005 como de la HU006, descritos en las tablas 14 y 16 el incremento puede ponerse en funcionamiento y por lo tanto el quinto Sprint se considera finalizado.

2.11 Sexto Sprint

El sexto Sprint se llevó a cabo del 16 de mayo de 2018 al 22 de mayo de 2018 tal como indica el cronograma de desarrollo inicial. Se implementaron los métodos para el inicio de sesión para la aplicación web y para las plataformas móviles.

Tabla 45.

Sexto Sprint

Sprint	Actividad	Tareas	Tiempo (Sem)	Responsable
6	Inicio de sesión en aplicación web	6.1. Implementar métodos para iniciar sesión en aplicación web.	1	Juan José Bahamonde
		6.2. Implementar métodos para guardar los datos del usuario ingresado.		
	Inicio de sesión en aplicación móvil.	6.3. Implementar métodos para iniciar sesión en Android y en iOS.		
	6.4. Implementar métodos para guardar los datos en una variable en Android y en iOS.			

2.11.1 Implementar métodos para iniciar sesión en aplicación web.

En la aplicación web solo podrán ingresar los usuarios que tengan rol de administrador. Una vez que se ingresa a la página principal del sistema se bloquean todas las formas web hasta que se inicie sesión.



Figura 99. Página de inicio antes de iniciar sesión

Para iniciar sesión deben presionar en el botón "LOGIN", y se cargará la página maestra exclusiva para el inicio de sesión. El usuario ingresa su *Nick* de usuario y su contraseña. Al iniciar sesión se envía una consulta a la base de datos para recuperar los datos del usuario que está intentando ingresar. Si los datos son correctos, el usuario puede ingresar al sistema.

Figura 100. Inicio de sesión en aplicación web

2.11.2 Implementar métodos para guardar los datos del usuario ingresado.

Una vez que los datos de usuario han sido verificados, la información pasa por la capa de datos y negocio para llegar a la de presentación. Se toma el nombre y apellido del usuario y se lo coloca en la etiqueta, para indicar quién inició sesión. También se activan todas las formas web y es posible utilizar el sistema.

Figura 101. Página de inicio con usuario ingresado

Al iniciar sesión se almacena en memoria cache toda la información de los usuarios. Estos datos tienen una duración de 30 minutos. Una vez transcurrido ese tiempo la sesión se cierra.

2.11.3 Implementar métodos para iniciar sesión en Android y en iOS.

El inicio de sesión en las aplicaciones móviles utiliza el procedimiento almacenado respectivo mediante servicios web. El usuario ingresa su *Nick* de usuario y su contraseña. Si la información es correcta, ingresa a la pantalla de inicio de la aplicación.

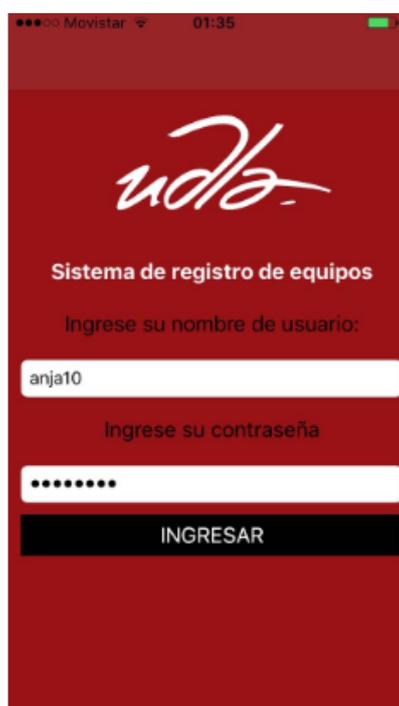


Figura 102. Inicio de sesión

2.11.4 Implementar métodos para guardar los datos en una variable en Android y en iOS.

Una vez que se inició sesión, los datos del usuario como identificador, nombre, apellido, *Nick*, y contraseña son guardados en las variables que se almacenan en la base de datos del teléfono. Esto con el fin de que el usuario no tenga que iniciar sesión cada vez que ingresa a la aplicación. Para esto, apenas se abre la aplicación se ejecuta el método de inicio de sesión. Si tiene datos almacenados, ingresa al sistema de inmediato. Si no los tiene, debe iniciar sesión nuevamente.

Una vez que aparece la pantalla de inicio, se carga una etiqueta con el nombre y apellido del usuario que inició sesión. Además, se muestra un mensaje, que solicita el ingreso de una nueva actividad para comenzar.

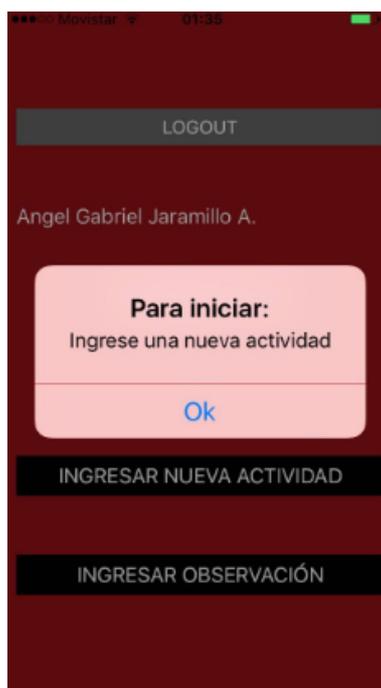


Figura 103. Pantalla de inicio después de iniciar sesión

Al finalizar las tareas planteadas para este Sprint, y realizar las reuniones respectivas para la revisión de este, se definió que el Sprint fue terminado, debido a que cumplió con los criterios de aceptación de las historias de usuario HU001 y HU004, mostrados en las tablas 6 y 12 respectivamente. Por lo tanto, el incremento es completamente funcional y puede ser utilizado.

2.12 Séptimo Sprint

El séptimo Sprint se llevó a cabo del 23 de mayo de 2018 al 29 de mayo de 2018 siguiendo con el cronograma de desarrollo inicial. Se desarrollaron los métodos

en las aplicaciones móviles para ingresar observaciones dentro y fuera de actividad.

Tabla 46.

Séptimo Sprint

Sprint	Actividad	Tareas	Tiempo (Sem)	Responsable
7	Ingreso de observaciones	7.1. Implementar métodos para ingresar observaciones fuera de actividad en iOS y Android. <hr/> 7.2. Implementar métodos para ingresar observaciones dentro de actividad en iOS y Android.	1	Juan José Bahamonde

2.12.1 Implementar métodos para ingresar observaciones fuera de actividad en iOS y Android.

Una vez que se ingresa a la pantalla de observación desde el inicio, se muestra un cuadro de texto donde se ingresa la descripción. Al presionar el botón de ingreso de observación se conecta al servicio web que llama al procedimiento almacenado respectivo. En las siguientes figuras, se muestra el ingreso de una observación fuera de actividad.

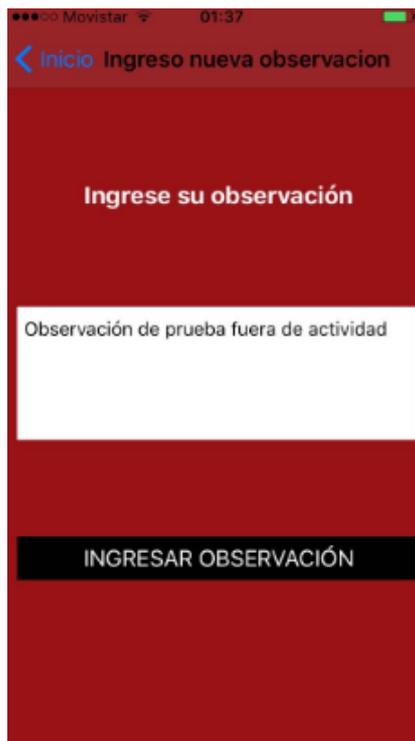


Figura 104. Ingreso de observación fuera de actividad

1 select * from observacion
2

IDObservacion	IDUsuario	FechaObservacion	DescripcionObservacion	EstadoObservacion
1	1	2018-06-05 01:37:19	Observación de prueba fuera de actividad	Fuera de actividad

Figura 105. Validación en base de datos e ingreso de observación

Al concluir, se visualiza un mensaje que indica el ingreso de la observación y permanece en la misma pantalla, de forma que se puede añadir una nueva observación o volver al inicio.



Figura 106. Mensaje de observación ingresada

2.12.2 Implementar métodos para ingresar observaciones dentro de actividad en iOS y Android.

Por último, para el ingreso de una observación durante una actividad lo único que se necesita es tener una actividad actual y se realiza el mismo procedimiento de ingreso de observación. En la base de datos se muestra el estado de la observación y la actividad durante la que se realizó.

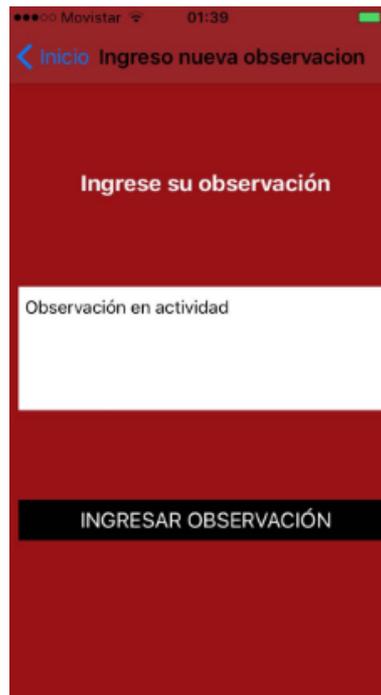


Figura 107. Ingreso de observación durante actividad

1 • `select * from observacion`
2

IDObservacion	IDUsuario	FechaObservacion	DescripcionObservacion	EstadoObservacion
1	1	2018-06-05 01:37:19	Observación de prueba fuera de actividad	Fuera de actividad
2	1	2018-06-05 01:39:32	Observación en actividad	Durante actividad: 2
*	NULL	NULL	NULL	NULL

Figura 108. Verificación en base de datos de ingreso de observación durante actividad

Después de finalizar todas las tareas que se plantearon para el séptimo Sprint, se tuvieron las reuniones respectivas para mostrar el incremento y realizar su revisión. Gracias a que este cumplió con los criterios de aceptación descritos en el reverso de la historia de usuario HU007 de la tabla 18, se puede determinar que el séptimo y último sprint ha concluido, y con ello el desarrollo del sistema.

2.13 Estimación de costos e implementación

2.13.1 Implementación en ambiente real

Una vez que se ha desarrollado todo el sistema, se implementó en el Data Center experimental que se encuentra en la FICA. Para esto, se migró la máquina virtual donde se encontraba el servidor de base de datos y de aplicaciones web. Para esto se creó un archivo con extensión .OVF y uno .VMDK para que tenga compatibilidad con el hipervisor que se maneja en el Data Center que es VMWare ESXi. Las características de cómputo que se le otorgaron fueron las siguientes y se calcularon dependiendo de los requerimientos mínimos descritos en la tabla 20:

View basic information about your computer

Windows edition

Windows Server 2012 R2 Standard

© 2013 Microsoft Corporation. All rights reserved.

System

Processor: Intel(R) Xeon(R) CPU E5-2609 v3 @ 1.90GHz 1.90 GHz (4 processors)
 Installed memory (RAM): 8,00 GB
 System type: 64-bit Operating System, x64-based processor
 Pen and Touch: Limited Touch Support with 10 Touch Points

Figura 109. Propiedades de la máquina virtual

Tabla 47.

Características de cómputo de la máquina virtual

	Capacidad	Velocidad
Almacenamiento	125 GB	
Memoria RAM	8 GB	
Procesadores	4	1.9 GHz

El acceso a la aplicación web y a los servicios web se lo realiza mediante el puerto 80, debido a que estos fueron publicados por IIS como un solo sitio web, esto evita inconvenientes en el caso de que existan puertos bloqueados que no permitan que el sistema funcione correctamente. El Data Center permite conexiones desde las redes LAN como WLAN a las que pueden acceder los docentes, desde cualquier parte del campus Queri.

2.13.2 Estimación de costos

Para el desarrollo de este prototipo se utilizaron en su mayoría herramientas gratuitas, como es el caso de Android Studio y Xcode. En cuanto al servidor de bases de datos, se utilizó la versión gratuita de MySQL, es decir MySQL Community Server. De la misma manera, para la ejecución de la aplicación web, se utilizó la versión gratuita Visual Studio Community 2013. Estas herramientas fueron seleccionadas debido a que son suficientemente efectivas para que el desarrollo y el funcionamiento del prototipo sean completamente viables y correctos.

El único precio que genera el desarrollo de este prototipo es la licencia para el sistema operativo Windows Server 2012. La edición utilizada fue Windows Server 2012 R2 Standard, cuya licencia fue proporcionada gratuitamente por Microsoft Virtual Academy. Para la estimación de costos se tomó en cuenta el valor real de esta licencia. Sin embargo, se podría implementar con la versión Essentials sin problema y con un menor costo.

Tabla 48.

Estimación de costos

Elemento	Costo
Android Studio 2.3.1	\$ 0
Xcode 9	\$ 0
Visual Studio Community 2013	\$ 0

MySQL Community Server	\$ 0
Windows Server 2012 R2 Standard	\$ 882
TOTAL	\$882

Por lo tanto, el costo del proyecto sería de \$ 882 al utilizar la versión Standard del sistema operativo y \$ 501 si se implementa con la versión Essential.

2.14 Prueba de rendimiento

Con el fin de determinar el rendimiento del servidor web instalado para el funcionamiento de la aplicación como de los servicios web, así como su escalabilidad ante cargas grandes de tráfico se realizaron pruebas con ayuda del software StresStimulus. Éste ayuda a ejecutar pruebas de carga para páginas web. Permite la simulación de usuarios físicos, los que crean una carga para al sitio web, por lo que es posible reconocer los cuellos de botella que se generan en el servidor, afectando el rendimiento de las aplicaciones.



Figura 110. Software para pruebas de rendimiento

El primer paso es grabar las acciones que se van a simular. Para esto se ejecutaron todas las funciones que tiene la aplicación web. Además, debido a que los servicios web son accesibles mediante la misma dirección IP, se ejecutaron todos ellos, tanto para la aplicación móvil en iOS como en Android.

Para esto se provee un nombre a la prueba y se selecciona el tipo de acceso a la aplicación. En este caso se seleccionó el buscador web Google Chrome,

debido a que es el más utilizado en la facultad. Después se especifica el URL de la aplicación web como se muestra en las figuras.

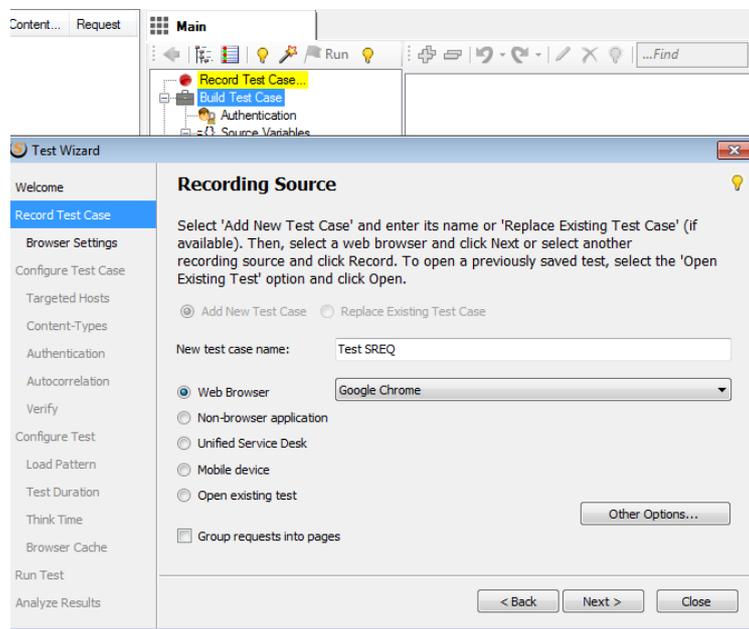


Figura 111. Nombre de prueba de rendimiento

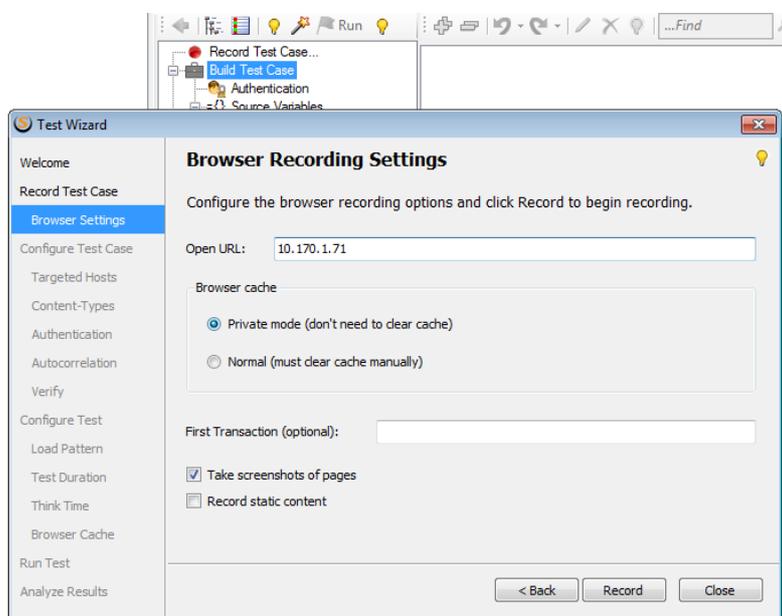


Figura 112. Dirección URL para la prueba de rendimiento

Una vez que se grabaron todas las acciones que se quieren simular, se seleccionan todos los *hosts* que se van a tomar en cuenta y también el tipo de contenido como se muestra en las siguientes figuras. Se tomaron en cuenta todos, debido a que se busca analizar todo el tráfico generado durante el uso del sistema.

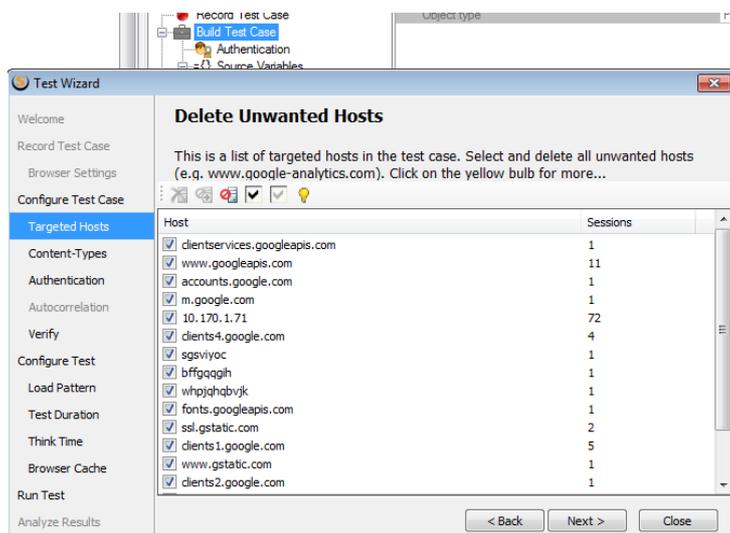


Figura 113. Hosts de la prueba de rendimiento

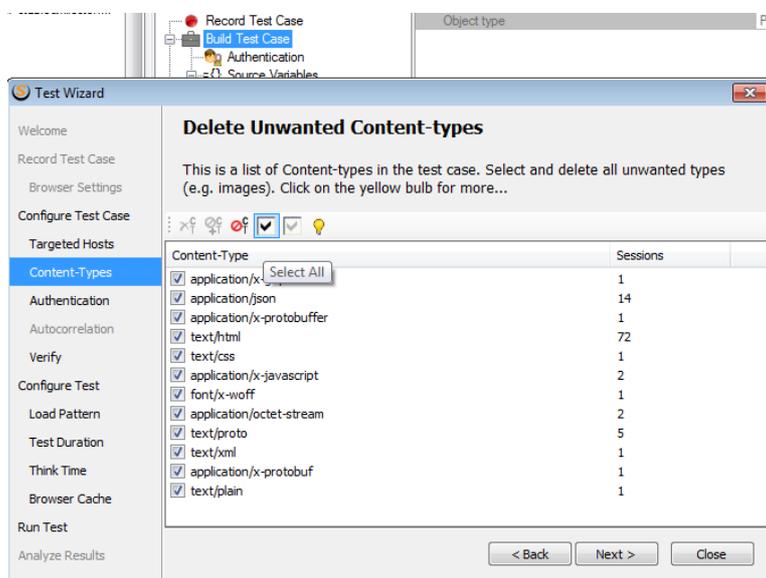


Figura 114. Tipos de contenido para la prueba de rendimiento

Después, se asigna el patrón de carga. Se realizaron dos pruebas, una con un valor constante de treinta usuarios, y otra con un patrón que incrementa un

usuario cada diez segundos, donde se puede obtuvo un máximo de veinte usuarios.

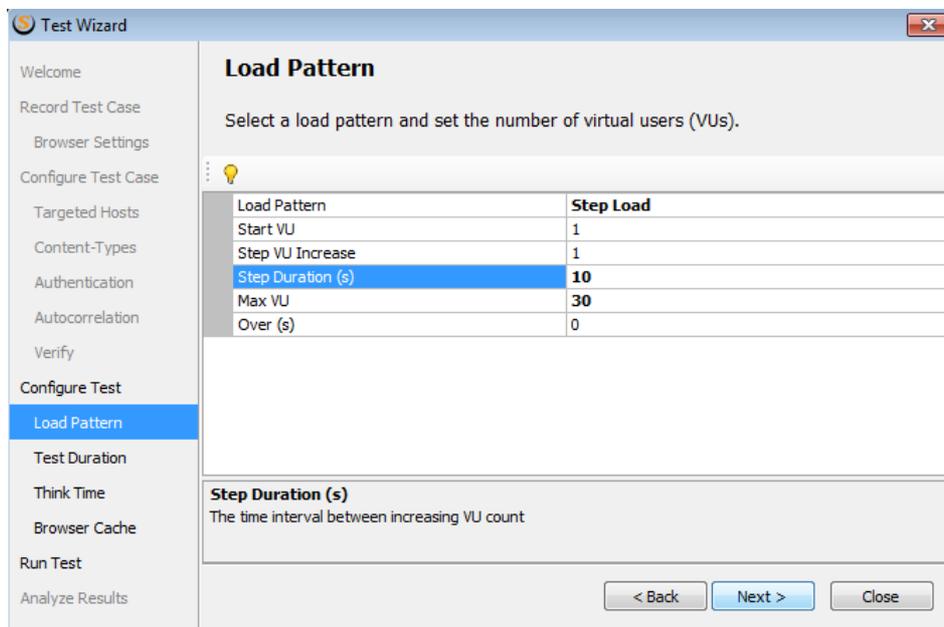


Figura 115. Patrón de carga

Por último, se detalló el tiempo que los usuarios virtuales van a estar activos en cada una de las páginas. En este caso se especificó un tiempo aleatorio entre dos y tres minutos como se muestra en la figura 116.

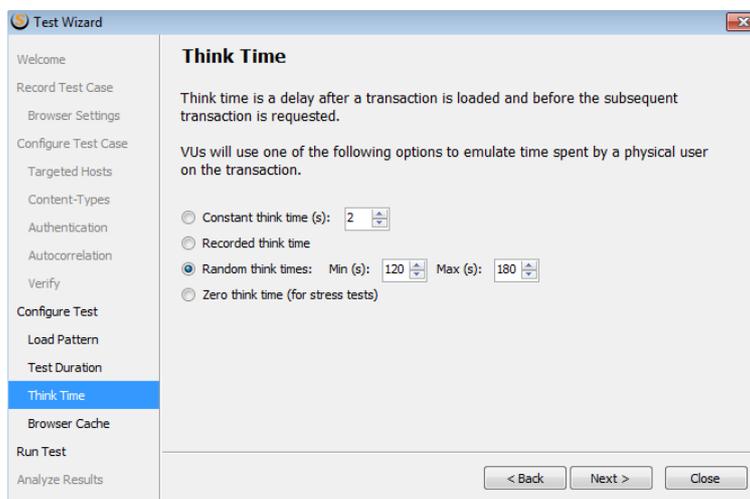


Figura 116. Tiempo de permanencia

2.14.1 Resultados

Los equipos que constan en el inventario y serían utilizados por los docentes se encuentran en un total de seis aulas, por lo que los servicios web serían utilizados por seis usuarios aproximadamente, salvo casos excepcionales. Por otro lado, los administradores de los equipos y de la aplicación web serían asignados según se necesite. Tomando esto en cuenta, se realizaron las pruebas en casos extremos, donde el número de usuarios activos y el número de peticiones es muy alto.

En la prueba con usuarios ascendentes se comenzó con un usuario virtual y se terminó con veinte. Se considera que cada uno de ellos realiza todas las acciones posibles en ambas aplicaciones. En la siguiente figura se muestra el indicador clave de rendimiento.

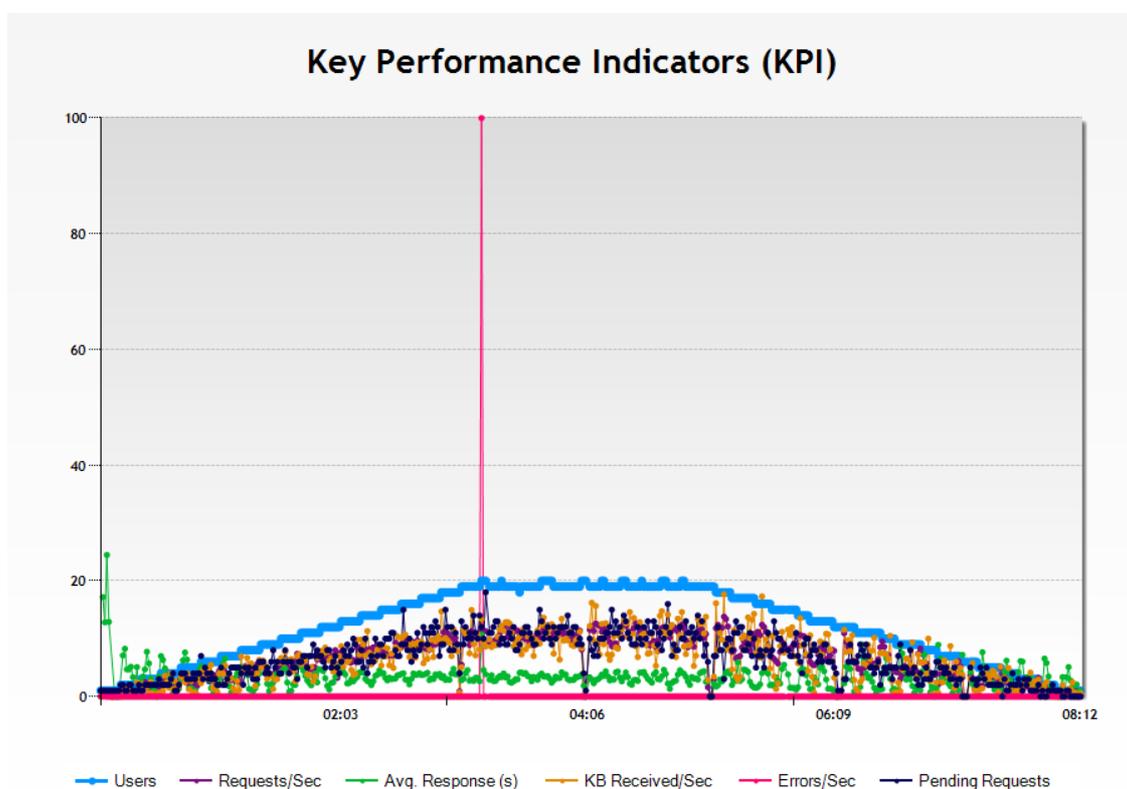


Figura 117. Indicador clave de rendimiento, usuarios ascendentes

En la tabla 48 se muestran las estadísticas de la prueba realizada, contiene las variables del indicador clave de rendimiento. Ésta muestra un mínimo, un máximo, el promedio y el último valor registrado. En este caso solo existió un error de todas las peticiones realizadas y el promedio de respuesta es de 0,033 segundos, lo cual indica que para este caso el servidor está respondiendo correctamente. Además, el promedio del tráfico generado es bastante bajo considerando todos los usuarios virtuales conectados. El informe completo generado por StresStimulus se encuentra en el Anexo 3.

Tabla 49.

Indicador clave de rendimiento para usuarios ascendentes

Curva	Rango	Min	Max	Promedio	Último
Usuarios	100	1	20	11,953	1
Peticiones/Seg	1.000	0	138	64,63	10
Prom. Respuesta (s)	1	0	0,245	0,033	0,009
KB Recibidos/Seg	10.000	0	1.765,964	622,69	57,259
Errors/Seg	1	0	1	0,002	0
Peticiones pendientes	100	0	18	6,557	0

La prueba con usuarios virtuales constantes simula una carga de un total de treinta, los cuales se conectan al mismo tiempo y utilizan tanto la aplicación web como las aplicaciones móviles. En la figura 118 se muestra la gráfica del indicador clave de rendimiento.

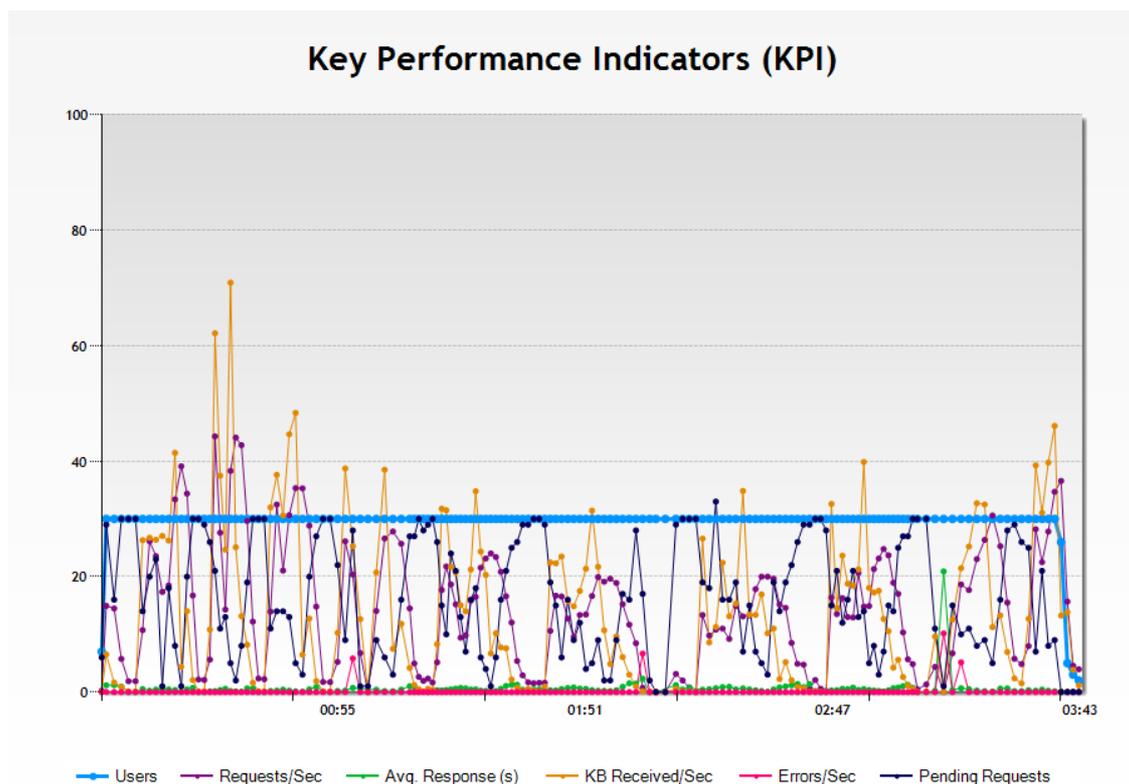


Figura 118. Indicador clave de rendimiento, usuarios constantes

En este caso se obtuvieron cinco errores, que en relación a las peticiones realizadas por segundo es un valor mínimo y aceptable. Sin embargo, estos errores provienen de las API de Google mas no del servidor web. La taza promedio del tráfico que genera la aplicación se duplica en este caso, sin embargo, 1 363,49 KB/s es bastante bajo considerando que son treinta usuarios virtuales ejecutando todas las funciones del sistema al mismo tiempo. El informe completo de la prueba realizada con usuarios virtuales constantes se encuentra en el anexo 4.

Tabla 50.

Indicador clave de rendimiento para usuarios constantes

Curva	Rango	Min	Max	Promedio	Último
Usuarios	100	2	30	29,327	2

Peticiones/Seg	1.000	0	443,085	142,803	39,491
Prom. Respuesta (s)	10	0	2,092	0,04	0,003
KB Recibidos/Seg	10.000	0	7.094,988	1.363,49	114,026
Errors/Seg	10	0	1,016	0,022	0
Peticiones pendientes	100	0	33	16,654	0

3. Conclusiones y Recomendaciones

3.1 Conclusiones

Las herramientas de desarrollo Xcode y Android Studio empleadas para la creación de las aplicaciones móviles en iOS y Android, permitieron el uso de las librerías y APIs necesarias para generar el lector de código de barras y las conexiones a la base de datos mediante servicios web. De la misma manera, el conector Connector/Net facilitó la conexión de la aplicación web ASP.NET a la base de datos del servidor MySQL. Estos elementos en conjunto con la metodología de desarrollo conllevaron a una adecuada implementación, instalación y funcionamiento del sistema de registro y control de equipos.

La metodología Scrum, utilizada para crear el prototipo, facilitó el desarrollo ágil y eficiente del sistema. Esta permitió dividir el proyecto en tareas y mostrar al dueño del producto el avance progresivo del mismo.

La creación de la base de datos del sistema con MySQL permite tener un registro mucho más confiable de los equipos y las personas que lo utilizan. Las entidades creadas y sus relaciones garantizan una correcta gestión e integridad de la información. Además, con los procedimientos almacenados se pueden generar reportes, necesarios para una buena administración de los equipos.

Con las pruebas realizadas de funcionamiento del prototipo, tanto de las aplicaciones móviles como de la aplicación web, se pudo determinar a través de los criterios de aceptación de cada Sprint, que el desarrollo del sistema se considera terminado. Esto debido a que cumple con todos los requerimientos solicitados por el usuario. Las pruebas de rendimiento demuestran que el sistema puede ser utilizado por grandes cantidades de usuarios, sin afectar al funcionamiento de éste, ni al rendimiento en general de la red. Además, los tiempos de respuesta, ancho de banda ocupado y tasa de errores son adecuados para su correcta utilización.

La instalación del sistema en el Data Center experimental de la FICA fue realizada con éxito. Con la publicación de la aplicación y servicios web mediante IIS y la generación de los instaladores de las aplicaciones móviles, el prototipo está habilitado y operativo para uso exclusivo de los docentes competentes y administradores de los equipos. Los usuarios podrán acceder a este tanto desde la red LAN como desde la red WLAN de la sede Queri de la Universidad de las Américas.

3.2 Recomendaciones

Este prototipo puede ser mejorado y ampliado, creando un producto final que pueda aplicarse en otras carreras y facultades de la Universidad de las Américas. Esto ayuda a la automatización de procesos que tengan funciones similares.

Para el desarrollo este sistema una sola persona cumplió con todos los roles que indica la metodología Scrum. En caso de mejorarlo y crear un sistema más complejo, lo recomendable es que cada uno de los roles sean ejecutados por diferentes personas. El tamaño del equipo de desarrollo debe manejarse para mantener la agilidad y debe tener la capacidad para completar un trabajo en su totalidad en menor tiempo.

Para garantizar la operación y disponibilidad de este sistema se puede crear redundancia tanto de la información almacenada en la base de datos como de la aplicación y servicios web. Además, para potenciar el rendimiento es posible implementar un balanceador de carga.

Para implementar funcionalidades más complejas al sistema, es recomendable adquirir las versiones completas tanto de los entornos de desarrollo, como del servidor de base de datos. Asimismo, para el funcionamiento legal del sistema operativo y explotación de todas sus características se debe obtener el licenciamiento respectivo.

Referencias

- Universidad de las Américas. (s.f). *Carreras*. Recuperado el 7 de diciembre de 2017 de: <http://www.udla.edu.ec/carreras/programas-academicos/pregrados/facultad-de-ingenieria-y-ciencias-aplicadas/ingenieria-en-tecnologias-de-la-informacion/>
- Ramírez, F. (2012). *Aprenda Practicando ASP.NET usando Visual Studio 2012*. México D.F., México: Alfaomega Grupo Editor, S.A. de C.V.
- Ceballos, J. (2013). *Enciclopedia de Microsoft Visual C#*. España: RA-MA Editorial
- Microsoft. (s.f a). *Introducción al servidor web (IIS)*. Recuperado el 28 de diciembre de 2017 de: [https://msdn.microsoft.com/es-es/library/hh831725\(v=ws.11\).aspx](https://msdn.microsoft.com/es-es/library/hh831725(v=ws.11).aspx)
- Schwaber, K y Sutherland, J. (2017). *La Guía de Scrum [archivo PDF]*. Recuperado el 3 de enero de 2018 de: <http://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Spanish-SouthAmerican.pdf#zoom=100>
- Monte, J. (2016). *Implantar Scrum con éxito*. Barcelona, España: Editorial UOC
- Trigas, M. (s.f). *Metodología Scrum [archivo PDF]*. Recuperado el 5 de enero de 2018 de: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf>
- Marín, C. (2014). *Arquitectura de IIS 7*. Recuperado el 2 de abril de 2018 de: <https://blogs.msdn.microsoft.com/cmarin/2014/10/10/arquitectura-de-iis-7/>
- Mircosoft. (s.f b). *Documentación de Visual Studio*. Recuperado el 2 de abril de 2018 de: <https://docs.microsoft.com/es-es/visualstudio/#pivot=features&panel=features1>

Android Studio. (s.f a). *Conoce Android Studio*. Recuperado el 2 de abril de 2018 de: <https://developer.android.com/studio/intro/index.html>

Android Studio. (s.f b). *Android Studio*. Recuperado el 2 de abril de 2018 de: <https://developer.android.com/studio/index.html?hl=es-419#features>

Apple Inc. (2018). *Xcode*. Recuperado el 2 de abril de 2018 de: <https://developer.apple.com/xcode/ide/>

Oracle. (s.f). *Oracle MySQL*. Recuperado el 2 de abril de 2018 de: <https://www.oracle.com/lad/mysql/index.html>

Oracle Corporation. (2018). *MySQL Workbench*. Recuperado el 2 de abril de 2018 de: <https://www.mysql.com/products/workbench/>

ANEXOS

Anexo 1. Requerimientos aplicación en Android

Requerimiento	Mínimo	Recomendado
Sistema Operativo	Android Jelly Bean 4.1	Android Marshmallow 6.01 o posterior
Dispositivo	Dispositivos que soporten Android 4.1	Dispositivos que soporten Android 6.0.1 (Samsung, Nokia, Huawei, Sony, LG, HTC, etc)
Espacio de almacenamiento libre	49.27 MB	60 MB
Características		Cámara posterior
Permisos		Cámara y Wi-Fi

Anexo 2. Requerimientos aplicación en iOS

Requerimiento	Mínimo	Recomendado
Sistema Operativo	iOS 10.3	iOS 11.2 o posteriores
Dispositivo	iPhone 5	iPhone 6 o posteriores
Espacio de almacenamiento libre	47.6 MB	60 MB
Características		Cámara posterior
Permisos		Cámara y Wi-Fi

Anexo 3. Resultados prueba de rendimiento con usuarios ascendientes

Test Summary



▼ Test Name

Test Run Name ?	
Test run description ?	
Test result name ?	2018_07_30_13_50_42_526_Untitled
Test File name ?	
Script last modified ?	lunes, 30 de julio de 2018 13:58:56

▼ Test Settings

Load pattern ?	Step Load - Start with 1 VUs, increase by 1 every 10s until 30 VUs
Complete after (hh:mm:ss) ?	10 Per VU iterations
Warm-up time (s) ?	0

▼ Test Run Information

Start time ?	jul-30 2018 13:50:45
End time ?	jul-30 2018 13:58:54
Test run duration ?	00:08:09

▼ Overall Result

Completion Status ?	Completed
Pass/Fail Status ?	Passed
Max User Load ?	30
Total sent (KB) ?	112.451,542
Total received (KB) ?	306.376,822
KB sent/sec ?	229,815
KB received/sec ?	626,135

▼ Test Iterations

Avg. Iteration time (s) ?	11,126
Iterations started ?	300
Iterations passed ?	300

▼ Requests

Avg. response time (s) ?	0,033
Requests/sec ?	64,989
Number of URLs ?	106
Total requests issued ?	31800
Request errors ?	1
Request timeouts ?	0

▼ Top Errors

T.C.	Request Id	Error Description
Test SREQ	# 10. www.googleapis.com/oauth2/v1/userinfo	Response code 503

▼ VU details

VU	Agent	T.C.	Iterations Started	Iterations Passed	Iterations Failed	Iteration Time (s)	Requests	Errors	Timeouts	Browser	Network	Cache
1	Local	Test SREQ	10	10		10,866	1060			Chrome 64.0	Gigabit Ethernet	New
2	Local	Test SREQ	10	10		10,667	1060			Chrome 64.0	Gigabit Ethernet	New
3	Local	Test SREQ	10	10		10,416	1060			Chrome 64.0	Gigabit Ethernet	New
4	Local	Test SREQ	10	10		11,111	1060	1		Chrome 64.0	Gigabit Ethernet	New
5	Local	Test SREQ	10	10		10,509	1060			Chrome 64.0	Gigabit Ethernet	New
6	Local	Test SREQ	10	10		10,833	1060			Chrome 64.0	Gigabit Ethernet	New
7	Local	Test SREQ	10	10		10,769	1060			Chrome 64.0	Gigabit Ethernet	New
8	Local	Test SREQ	10	10		10,956	1060			Chrome 64.0	Gigabit Ethernet	New
9	Local	Test SREQ	10	10		10,631	1060			Chrome 64.0	Gigabit Ethernet	New
10	Local	Test SREQ	10	10		10,902	1060			Chrome 64.0	Gigabit Ethernet	New
11	Local	Test SREQ	10	10		10,841	1060			Chrome 64.0	Gigabit Ethernet	New
12	Local	Test SREQ	10	10		11,32	1060			Chrome 64.0	Gigabit Ethernet	New
13	Local	Test SREQ	10	10		11,113	1060			Chrome 64.0	Gigabit Ethernet	New
14	Local	Test SREQ	10	10		11,263	1060			Chrome 64.0	Gigabit Ethernet	New
15	Local	Test SREQ	10	10		11,089	1060			Chrome 64.0	Gigabit Ethernet	New
16	Local	Test SREQ	10	10		11,423	1060			Chrome 64.0	Gigabit Ethernet	New
17	Local	Test SREQ	10	10		10,969	1060			Chrome 64.0	Gigabit Ethernet	New
18	Local	Test SREQ	10	10		11,225	1060			Chrome 64.0	Gigabit Ethernet	New
19	Local	Test SREQ	10	10		11,292	1060			Chrome 64.0	Gigabit Ethernet	New
20	Local	Test SREQ	10	10		11,667	1060			Chrome 64.0	Gigabit Ethernet	New
21	Local	Test SREQ	10	10		11,266	1060			Chrome 64.0	Gigabit Ethernet	New
22	Local	Test SREQ	10	10		11,553	1060			Chrome 64.0	Gigabit Ethernet	New
23	Local	Test SREQ	10	10		11,281	1060			Chrome 64.0	Gigabit Ethernet	New
24	Local	Test SREQ	10	10		11,489	1060			Chrome 64.0	Gigabit Ethernet	New
25	Local	Test SREQ	10	10		11,397	1060			Chrome 64.0	Gigabit Ethernet	New
26	Local	Test SREQ	10	10		11,477	1060			Chrome 64.0	Gigabit Ethernet	New
27	Local	Test SREQ	10	10		11,283	1060			Chrome 64.0	Gigabit Ethernet	New
28	Local	Test SREQ	10	10		11,506	1060			Chrome 64.0	Gigabit Ethernet	New
29	Local	Test SREQ	10	10		11,114	1060			Chrome 64.0	Gigabit Ethernet	New
30	Local	Test SREQ	10	10		11,559	1060			Chrome 64.0	Gigabit Ethernet	New

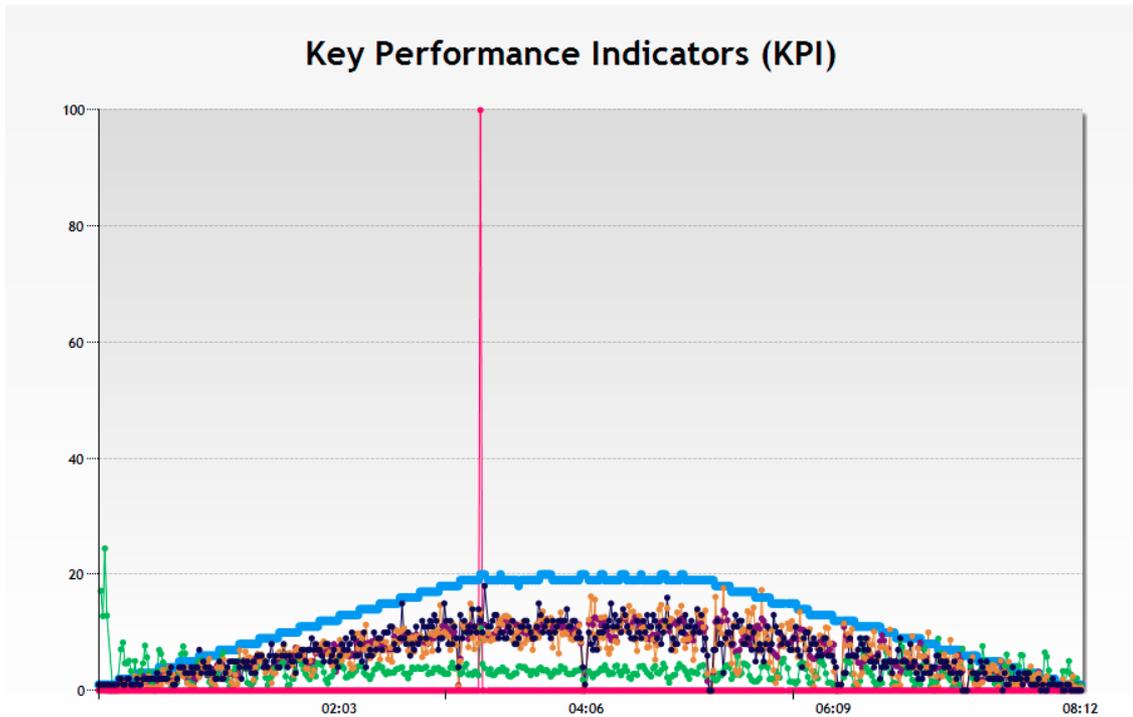
▼ Test Case details

T.C.	VUs	Iterations Started	Iterations Passed	Iterations Failed	Incomplete Iteration's Requests	Requests	Avg. (s)	Errors	Timeouts	Page Think time	Transaction Think time	Test case delay
Test SREQ	30	300	300			31800	11,126	1		Constant 2s	Rnd. b/t 120s-180s	Zero

▼ Agent details

Agent	Aggregate (s)	Max VUs	Sent (KB)	Received (KB)	Sent (KB/s)	Received (KB/s)	Errors	Request Timeouts	Iterations Started	Iterations Passed	Iterations Failed	Iteration Time (s)	Requests	Request /sec	Response (s)
Local	1.038,672	30	112.451,542	306.376,822	229,815	626,137	1		300	300		11,126	31800	64,989	0,033

▼ Key Performance Indicators (KPI)



— Users
 — Requests/Sec
 — Avg. Response (s)
 — KB Received/Sec
 — Errors/Sec
 — Pending Requests

Curve	Range	Min	Max	Avg	Last	Warnings	Errors
Users	100	1	20	11,953	1		
Requests/Sec	1.000	0	138	64,63	10		
Avg. Response (s)	1	0	0,245	0,033	0,009		
KB Received/Sec	10.000	0	1.765,964	622,69	57,259		
Errors/Sec	1	0	1	0,002	0		
Pending Requests	100	0	18	6,557	0		

Anexo 4. Resultados prueba de rendimiento con usuarios constantes

Test Summary



▼ Test Name

Test Run Name ?	
Test run description ?	
Test result name ?	2018_07_30_13_43_26_577_Untitled
Test File name ?	
Script last modified ?	lunes, 30 de julio de 2018 13:47:13

▼ Test Settings

Load pattern ?	Steady Load - 30 VUs
Complete after (hh:mm:ss) ?	10 Per VU iterations
Warm-up time (s) ?	0

▼ Test Run Information

Start time ?	jul-30 2018 13:43:29
End time ?	jul-30 2018 13:47:10
Test run duration ?	00:03:40

▼ Overall Result

Completion Status ?	Completed
Pass/Fail Status ?	Passed
Max User Load ?	30
Total sent (KB) ?	110.308,536
Total received (KB) ?	303.576,842
KB sent/sec ?	499,305
KB received/sec ?	1.374,123

▼ Test Iterations

Avg. Iteration time (s) ?	14,049
Iterations started ?	300
Iterations passed ?	300

▼ Requests

Avg. response time (s) ?	0,04
Requests/sec ?	143,941
Number of URLs ?	106
Total requests issued ?	31800
Request errors ?	5
Request timeouts ?	0

▼ Top Errors

T.C.	Request Id	Error Description
Test SREQ	#11. www.googleapis.com/oauth2/v1/userinfo	Response code 503
Test SREQ	#8. www.googleapis.com/oauth2/v1/userinfo	Response code 500
Test SREQ	#10. www.googleapis.com/oauth2/v1/userinfo	Response code 500
Test SREQ	#12. www.googleapis.com/oauth2/v1/userinfo	Response code 500

▼ VU details

VU	Agent	T.C.	Iterations Started	Iterations Passed	Iterations Failed	Iteration Time (s)	Requests	Errors	Timeouts	Browser	Network	Cache
1	Local	Test SREQ	10	10		13,866	1060	1		Chrome 64.0	Gigabit Ethernet	New
2	Local	Test SREQ	10	10		14,066	1060			Chrome 64.0	Gigabit Ethernet	New
3	Local	Test SREQ	10	10		13,961	1060			Chrome 64.0	Gigabit Ethernet	New
4	Local	Test SREQ	10	10		13,947	1060			Chrome 64.0	Gigabit Ethernet	New
5	Local	Test SREQ	10	10		14,042	1060			Chrome 64.0	Gigabit Ethernet	New
6	Local	Test SREQ	10	10		14,042	1060			Chrome 64.0	Gigabit Ethernet	New
7	Local	Test SREQ	10	10		14,109	1060	2		Chrome 64.0	Gigabit Ethernet	New
8	Local	Test SREQ	10	10		14,416	1060	1		Chrome 64.0	Gigabit Ethernet	New
9	Local	Test SREQ	10	10		14,017	1060			Chrome 64.0	Gigabit Ethernet	New
10	Local	Test SREQ	10	10		14,196	1060			Chrome 64.0	Gigabit Ethernet	New
11	Local	Test SREQ	10	10		14,034	1060			Chrome 64.0	Gigabit Ethernet	New
12	Local	Test SREQ	10	10		14,031	1060			Chrome 64.0	Gigabit Ethernet	New
13	Local	Test SREQ	10	10		14,05	1060			Chrome 64.0	Gigabit Ethernet	New
14	Local	Test SREQ	10	10		13,845	1060			Chrome 64.0	Gigabit Ethernet	New
15	Local	Test SREQ	10	10		14,075	1060			Chrome 64.0	Gigabit Ethernet	New
16	Local	Test SREQ	10	10		14,2	1060			Chrome 64.0	Gigabit Ethernet	New
17	Local	Test SREQ	10	10		13,915	1060			Chrome 64.0	Gigabit Ethernet	New
18	Local	Test SREQ	10	10		14,005	1060			Chrome 64.0	Gigabit Ethernet	New
19	Local	Test SREQ	10	10		13,979	1060			Chrome 64.0	Gigabit Ethernet	New
20	Local	Test SREQ	10	10		13,978	1060			Chrome 64.0	Gigabit Ethernet	New
21	Local	Test SREQ	10	10		14,387	1060	1		Chrome 64.0	Gigabit Ethernet	New
22	Local	Test SREQ	10	10		14,129	1060			Chrome 64.0	Gigabit Ethernet	New
23	Local	Test SREQ	10	10		14,17	1060			Chrome 64.0	Gigabit Ethernet	New
24	Local	Test SREQ	10	10		13,882	1060			Chrome 64.0	Gigabit Ethernet	New
25	Local	Test SREQ	10	10		13,939	1060			Chrome 64.0	Gigabit Ethernet	New
26	Local	Test SREQ	10	10		14,077	1060			Chrome 64.0	Gigabit Ethernet	New
27	Local	Test SREQ	10	10		14,029	1060			Chrome 64.0	Gigabit Ethernet	New
28	Local	Test SREQ	10	10		14,027	1060			Chrome 64.0	Gigabit Ethernet	New
29	Local	Test SREQ	10	10		14,014	1060			Chrome 64.0	Gigabit Ethernet	New
30	Local	Test SREQ	10	10		14,039	1060			Chrome 64.0	Gigabit Ethernet	New

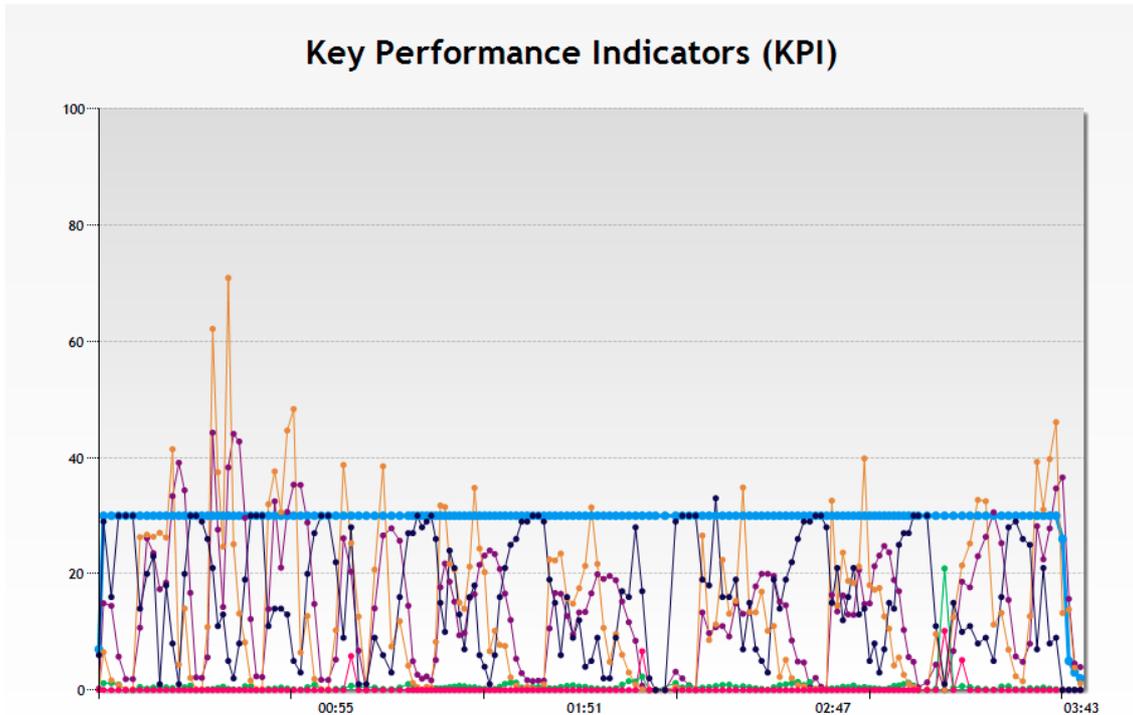
▼ Test Case details

T.C.	VUs	Iterations Started	Iterations Passed	Iterations Failed	Incomplete Iteration's Requests	Requests	Avg. (s)	Errors	Timeouts	Page Think time	Transaction Think time	Test case delay
Test SREQ	30	300	300			31800	14,049	5		Constant 2s	Rnd. b/t 120s-180s	Zero

▼ Agent details

Agent	Aggregate (s)	Max VUs	Sent (KB)	Received (KB)	Sent (KB/s)	Received (KB/s)	Errors	Request Timeouts	Iterations Started	Iterations Passed	Iterations Failed	Iteration Time (s)	Requests	Request /sec	Response (s)
Local	1.273,719	30	110.308,536	303.576,842	499,307	1.374,127	5		300	300		14,049	31800	143,941	0,04

▼ Key Performance Indicators (KPI)



— Users
 — Requests/Sec
 — Avg. Response (s)
 — KB Received/Sec
 — Errors/Sec
 — Pending Requests

Curve	Range	Min	Max	Avg	Last	Warnings	Errors
Users	100	2	30	29,327	2		
Requests/Sec	1.000	0	443,085	142,803	39,491		
Avg. Response (s)	10	0	2,092	0,04	0,003		
KB Received/Sec	10.000	0	7.094,988	1.363,49	114,026		
Errors/Sec	10	0	1,016	0,022	0		
Pending Requests	100	0	33	16,654	0		

Anexo 5. Contenido de la carpeta de anexos del CD

El CD contiene una carpeta de anexos con los siguientes elementos:

- Script de la base de datos
- Proyecto de la aplicación web en Visual Studio 2013
- Proyecto de la aplicación para dispositivos Android desarrollada en Android Studio.
- Proyecto de la aplicación para dispositivos iOS desarrollada en Xcode 9.2
- Archivo de instalación APK de la aplicación para Android
- Archivo de instalación IPA de la aplicación para iOS

