



UTMACH

UNIDAD ACADÉMICA DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

DESARROLLO DE UN SISTEMA WEB Y MÓVIL BASADO EN LA
TECNOLOGÍA CLIENTE SERVIDOR PARA LA GESTIÓN DE
ASISTENCIA LABORAL

CASTILLO CRESPÍN JIMMY FERNANDO
INGENIERO DE SISTEMAS

MACHALA
2018



UTMACH

UNIDAD ACADÉMICA DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

DESARROLLO DE UN SISTEMA WEB Y MÓVIL BASADO EN LA
TECNOLOGÍA CLIENTE SERVIDOR PARA LA GESTIÓN DE
ASISTENCIA LABORAL

CASTILLO CRESPÍN JIMMY FERNANDO
INGENIERO DE SISTEMAS

MACHALA
2018



UTMACH

UNIDAD ACADÉMICA DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

EXAMEN COMPLEXIVO

DESARROLLO DE UN SISTEMA WEB Y MÓVIL BASADO EN LA TECNOLOGÍA
CLIENTE SERVIDOR PARA LA GESTIÓN DE ASISTENCIA LABORAL

CASTILLO CRESPÍN JIMMY FERNANDO
INGENIERO DE SISTEMAS


MOLINA RIOS JIMMY ROLANDO

MACHALA, 04 DE JULIO DE 2018

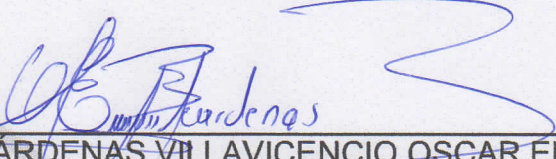
MACHALA
04 de julio de 2018

Nota de aceptación:

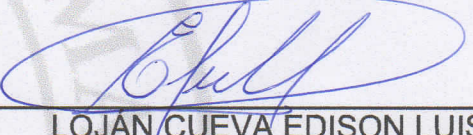
Quienes suscriben, en nuestra condición de evaluadores del trabajo de titulación denominado Desarrollo de un sistema web y móvil basado en la tecnología cliente servidor para la gestión de asistencia laboral, hacemos constar que luego de haber revisado el manuscrito del precitado trabajo, consideramos que reúne las condiciones académicas para continuar con la fase de evaluación correspondiente.



MOLINA RIOS JIMMY ROLANDO
0703691980
TUTOR - ESPECIALISTA 1



CÁRDENAS VILLAVICENCIO OSCAR ÉFRÉN
0703935312
ESPECIALISTA 2



LOJAN CUEVA EDISON LUIS
0703249698
ESPECIALISTA 3

Fecha de impresión: miércoles 04 de julio de 2018 - 10:38

Urkund Analysis Result

Analysed Document: TrabajoComplexivo-Castillo Jimmy .docx (D40217812)
Submitted: 6/16/2018 9:12:00 PM
Submitted By: jmolina@utmachala.edu.ec
Significance: 0 %

Sources included in the report:

Instances where selected sources appear:

0

CLÁUSULA DE CESIÓN DE DERECHO DE PUBLICACIÓN EN EL REPOSITORIO DIGITAL INSTITUCIONAL

El que suscribe, CASTILLO Crespín Jimmy Fernando, en calidad de autor del siguiente trabajo escrito titulado Desarrollo de un sistema web y móvil basado en la tecnología cliente servidor para la gestión de asistencia laboral, otorga a la Universidad Técnica de Machala, de forma gratuita y no exclusiva, los derechos de reproducción, distribución y comunicación pública de la obra, que constituye un trabajo de autoría propia, sobre la cual tiene potestad para otorgar los derechos contenidos en esta licencia.


El autor declara que el contenido que se publicará es de carácter académico y se enmarca en las disposiciones definidas por la Universidad Técnica de Machala.

Se autoriza a transformar la obra, únicamente cuando sea necesario, y a realizar las adaptaciones pertinentes para permitir su preservación, distribución y publicación en el Repositorio Digital Institucional de la Universidad Técnica de Machala.

El autor como garante de la autoría de la obra y en relación a la misma, declara que la universidad se encuentra libre de todo tipo de responsabilidad sobre el contenido de la obra y que asume la responsabilidad frente a cualquier reclamo o demanda por parte de terceros de manera exclusiva.

Aceptando esta licencia, se cede a la Universidad Técnica de Machala el derecho exclusivo de archivar, reproducir, convertir, comunicar y/o distribuir la obra mundialmente en formato electrónico y digital a través de su Repositorio Digital Institucional, siempre y cuando no se lo haga para obtener beneficio económico.

Machala, 04 de julio de 2018



CASTILLO Crespín Jimmy Fernando
0706829116



DEDICATORIA

Primeramente, a Dios, por brindarme la salud y fuerza necesaria para lograr cumplir todas mis metas propuestas durante la duración del periodo de mi carrera.

A mis padres, aquellos que me dieron la vida y siempre están ahí cuando se los necesita, tanto en momentos malos como en los buenos, resaltando todo su apoyo, consejos y ánimos entregados hacia mí día tras día.

A mis hermanos, porque al igual que mis padres, me entregaron todo su apoyo, ánimos y comprensión, lo cual me motivaron mucho para el cumplimiento de mis objetivos.

A mi tutor, Ing. Jimmy Molina, que, a través de sus consejos, orientaciones y amplios conocimientos, supo cómo prepararme y superar muchos obstáculos durante el proceso de mi trabajo de titulación.

Finalmente, pero no menos importante, a la Universidad Técnica de Machala, que en conjunto con la Unidad Académica de Ingeniería Civil y la carrera de Ingeniería de Sistemas, agradecerles por brindarme la oportunidad de forjarme como profesional y prepararme para desenvolverme en el campo laboral.

Sr. Castillo Crespín Jimmy Fernando.

AGRADECIMIENTO

Agradezco, primeramente, ante todo a Dios, el cual durante todo el transcurso de mi vida me ha dado fuerza, salud y me ha guiado por el camino del bien tanto en las cosas que me he propuesto realizar y en las decisiones que se me han presentado en mi convivir diario.

Agradezco a mi familia, los cuales son los seres más importantes en mi vida, ellos supieron criarme con los mejores valores y me han brindado sus apoyos tantos emocionales como económicos.

Finalmente agradezco a la Universidad Técnica de Machala por darme la oportunidad de cursar mis estudios en una buena institución educativa, al igual agradezco a mis profesores que me acompañaron durante todo el lapso de mi formación como profesional, especialmente a mi tutor Ing. Jimmy Molina, por su dedicación, conocimientos y apoyo hacia mí durante sus tutorías.

Sr. Castillo Crespín Jimmy Fernando

RESUMEN

DESARROLLO DE UN SISTEMA WEB Y MÓVIL BASADO EN LA TECNOLOGÍA CLIENTE SERVIDOR PARA LA GESTIÓN DE ASISTENCIA LABORAL

Castillo Crespín Jimmy Fernando, 0706829116

En la actualidad existe una gran demanda por parte de las empresas en utilizar aplicaciones web y móviles, promocionando sus productos o servicios o utilizándolos en ambientes laborales, sea cual sea el caso, lograr que muchos usuarios accedan a sus aplicaciones de manera remota y en cualquier lugar a través de la red, ha sido uno de los requerimientos más habituales por parte de las empresas y una de las muchas maneras de realizar esto es utilizando la arquitectura cliente-servidor, dándole la oportunidad a muchos usuarios de utilizar las aplicaciones, todos al mismo tiempo. Por ello, en el presente informe, se desarrollará un sistema web para la gestión de asistencia del personal docente de la carrera de Ingeniería de Sistemas, el mismo que será basado en una arquitectura cliente-servidor, haciendo uso del lenguaje de programación PHP bajo el Framework Laravel, permitiendo al administrador del sitio, gestionar las asistencias del personal docente y a su vez, hacer uso de una aplicación móvil realizada en el sistema operativo Android, visualizando de mejor manera los horarios de trabajo, marcaciones, atrasos de los empleados entre otras opciones y dado a que este caso de estudio está orientado bajo la asignatura de Programación III, se hará uso de estructuras de datos como listas, pilas o colas. Cabe recalcar que para el desarrollo de cada fase de este proyecto se tomará como guía la metodología ágil ICONIX.

PALABRAS CLAVES: Aplicación Web, Aplicación Móvil, Arquitectura Cliente-Servidor, Metodología Ágil ICONIX, Framework Laravel, Sistema Operativo Android.

ABSTRACT

DEVELOPMENT OF A WEB AND MOBILE SYSTEM BASED ON TECHNOLOGY CLIENT SERVER FOR LABOR ASSISTANCE MANAGEMENT

Castillo Crespín Jimmy Fernando, 0706829116

Currently, there is a great demand from companies to use web and mobile applications, promoting their products or services or using them in work environments, whatever the case, getting many users to access their applications remotely and in any place through the network, has been one of the most common requirements by companies and one of the many ways to achieve this is using the client-server architecture, giving many users the opportunity to use the applications, all at Same time. Therefore, in this report, a web system will be developed for the attendance management of the teaching staff of the Systems Engineering career, which will be based on a client-server architecture, making use of the PHP programming language under the Framework Laravel, allowing the administrator of the site, manage the assistance of teaching staff and in turn, make use of a mobile application made in the Android operating system, to better visualize work schedules, markings, employee delays among other options and given that this case study is oriented under the subject of Programming III, it will make use of data structures such as lists, stacks or tails. It should be noted that for the development of each phase of this project will be taken as a guide the agile ICONIX methodology.

KEYWORDS: Web Application, Mobile Application, Client-Server Architecture, Agile ICONIX Methodology, Laravel Framework, Android Operating System.

CONTENIDO

	Pág.
DEDICATORIA	1
AGRADECIMIENTO	2
ABSTRACT	4
CONTENIDO	5
LISTA DE TABLAS	7
LISTA DE ILUSTRACIONES	8
LISTA DE ANEXOS	9
1. INTRODUCCIÓN	11
1.1 Marco Contextual	12
1.2 Problema	12
1.3 Objetivo General	12
2. DESARROLLO	13
2.1 Marco teórico	13
2.1.1 Arquitectura Física: Cliente-Servidor.	13
2.1.2 Arquitectura Lógica: Modelo, Vista, Controlador (MVC).	13
2.1.3 Sistema Operativo Centos 7.	13
2.1.4 Servidor Web Apache.	13
2.1.5 Gestor de Base de Datos Relacional Mysql.	14
2.1.6 Lenguaje de Programación Backend: Framework Laravel 5.	14
2.1.7 Lenguaje de Programación Frontend: Javascript.	14
2.1.8 Framework Web: Framework Bootstrap 4.	14
2.1.9 Lenguaje de Marcado para Hipertextos (HTML).	14
2.1.10 Protocolo de Comunicación HTTP.	15
2.1.11 Protocolo de Comunicación ApisRest.	15
2.1.12 Arquitectura cliente: App Android.	15
2.1.13 Arquitectura cliente: Aplicación en Java 8.	15
2.1.14 Estructura de datos.	15
2.1.15 Modelo de Proceso de Software: Metodología Ágil ICONIX.	15
2.2 Solución del problema	16
2.2.1 Materiales	16
2.2.2 Desarrollo de la Metodología ICONIX	17

2.2.2.1 Fase 1: Análisis de requerimientos	17
2.2.2.2 Fase 2: Análisis y diseño preliminar	17
2.2.2.3 Fase 3: Diseño	18
2.2.2.4 Fase 4: Implementación	18
2.3 Resultados	18
3. CONCLUSIONES	21
Bibliografía	22
ANEXOS	24

LISTA DE TABLAS

	Pág.
Tabla 1: Materiales utilizados	16
Tabla 2: RF 01 -Gestión de usuarios	24
Tabla 3: RF 02 - Gestión de roles	24
Tabla 4: RF 03 - Login	25
Tabla 5: RF 04 - Gestión de planificación del calendario de clases	25
Tabla 6: RF 05 - Gestión de asistencia laboral	26
Tabla 7: RF 06 -Gestión de laboratorios.	26
Tabla 8: RF 07 - Gestión de asignaturas	27
Tabla 9: RF 08 - Gestión de semestres y años lectivos.	27
Tabla 10: RF 09 - Reportes de asistencias mensuales	27
Tabla 11: RNF 01 - Diseñar la solución utilizando una arquitectura n-thiers	28
Tabla 12: RNF 02 - Diseñar la solución utilizando una arquitectura cliente-servidor	28
Tabla 13: RNF 03 - Utilizar estructura de datos	29
Tabla 14: RNF 04 - Validaciones	29
Tabla 15: RNF 05 - Aplicación móvil	30
Tabla 16: RNF 06- Diseño responsivo.	30
Tabla 17: RNF 07 - Control de permisos	30
Tabla 18: Caso de Uso Expandido: CRUD Usuarios	36
Tabla 19: Caso de Uso Expandido: CRUD ROLES	38
Tabla 20: Caso de Uso Expandido: Login	40
Tabla 21: Caso de Uso Expandido: Gestionar calendario	41
Tabla 22: Caso de Uso Expandido: Asistencia del Personal	42
Tabla 23: Caso de Uso Expandido: Gestión de laboratorios	43
Tabla 24: Caso de Uso Expandido: Gestión de asignaturas	45
Tabla 25: Caso de Uso Expandido: Gestión de semestres	47
Tabla 26: Caso de Uso Expandido: Reportes	48

LISTA DE ILUSTRACIONES

	Pág.
Ilustración 1: Caso de Uso para RF-01	31
Ilustración 2: Caso de Uso para RF-02	31
Ilustración 3: Caso de Uso para RF-03	32
Ilustración 4: Caso de Uso para RF-04	32
Ilustración 5: Caso de Uso para RF-05	32
Ilustración 6: Caso de Uso para RF-06	33
Ilustración 7: Caso de Uso para RF-07	33
Ilustración 8: Caso de Uso para RF-08	33
Ilustración 9: Caso de Uso para RF-09	34
Ilustración 10: Modelo de dominio	34
Ilustración 11: Login	34
Ilustración 12: Panel de administración	35
Ilustración 13: Listado de Usuarios	35
Ilustración 14: Aplicación móvil	36
Ilustración 15: Diagrama de clases	50
Ilustración 16: Diagrama relacional base de datos	51
Ilustración 17: Diagrama de robustez - Caso de uso 01	51
Ilustración 18: Diagrama de robustez - Caso de uso 02	52
Ilustración 19: Diagrama de robustez - Caso de uso 03	52
Ilustración 20: Diagrama de robustez - Caso de uso 04	52
Ilustración 21: Diagrama de robustez - Caso de uso 05	53
Ilustración 22: Diagrama de robustez - Caso de uso 06	53
Ilustración 23: Diagrama de robustez - Caso de uso 07	53
Ilustración 24: Diagrama de robustez - Caso de uso 08	54
Ilustración 25: Diagrama de robustez - Caso de uso 09	54
Ilustración 26: Diagrama de secuencia - Caso de usos 01-02-04-05-06-07-08	54
Ilustración 27: Diagrama de secuencia - Caso de uso 03	56
Ilustración 28: Diagrama de secuencia - Caso de uso 09	57
Ilustración 29: Diseño arquitectónico	57
Ilustración 30: Ingreso al sistema web	58

Ilustración 31: Dashboard Administrador	58
Ilustración 32: Menú dashboard administrador	58
Ilustración 33: Dashboard docente	59
Ilustración 34: Menú dashboard docente	60
Ilustración 35: Estructura de datos (TAD)	60
Ilustración 36: Creación de calendarios	60
Ilustración 37: Generación de reportes	61
Ilustración 38: Aplicación escritorio – lector huellas	61
Ilustración 39: Registro de huella digital	61
Ilustración 40: Registro de asistencia	62
Ilustración 41: Visualización de asistencias	63
Ilustración 42: Login en android	63
Ilustración 43: Dashboard administrador en android	64

LISTA DE ANEXOS

	Pág.
Anexo A. Requisitos funcionales y no funcionales	24
Anexo B. Modelo de caso de usos	31
Anexo C. Modelo de dominio	34
Anexo D. Prototipación rápida	34
Anexo E. Caso de uso expandido	36
Anexo F. Diagrama de clases	50
Anexo G. Diagrama relacional	51
Anexo H. Diagrama de robustez	51
Anexo I. Diagrama de secuencias	54
Anexo J. Diagrama arquitectónico	57
Anexo K. Ingreso al sistema web	58
Anexo L. Dashboard Administrador	58
Anexo M. Menú dashboard administrador	58
Anexo N. Dashboard docente	59
Anexo O. Menú dashboard docente	60
Anexo P. Estructura de datos (TAD)	60

Anexo Q. Creación de calendarios	60
Anexo R. Generación de reportes	61
Anexo S. Aplicación escritorio – lector huellas	61
Anexo T. Registro de huella digital	61
Anexo U. Registro de asistencia	62
Anexo V. Visualización de asistencias	63
Anexo W. Login Android	63
Anexo X. Dashboard administrador en android	64

1. INTRODUCCIÓN

Desde el surgimiento de las primeras computadoras, todas las aplicaciones que se encontraban instaladas eran de arquitectura stand-alone, es decir, programas que funcionaban en la misma computadora con propósitos que solamente un usuario y una máquina podía realizar, pero desde el surgimiento del internet y las redes, fueron apareciendo nuevas arquitecturas llegando hasta tal punto de encontrar aplicaciones alojadas en la nube de las cuales millones de usuarios interactúan a la vez desde distintas partes del mundo, sin necesidad que el usuario conozca dicha infraestructura [1].

A medida que se fueron desarrollando todas las redes de comunicación, surgió la posibilidad de conectar múltiples clientes (ordenadores y móviles) a distintos servidores, permitiendo así, la realización de transferencias de datos a muy altas velocidades y partir de aquí nació el concepto de los sistemas distribuidos siendo la arquitectura cliente-servidor una de las más utilizadas.

La arquitectura cliente servidor es una tecnología distribuida la cual consiste en repartirse las tareas solicitadas por muchos clientes entre distintos ordenadores, donde cada ordenador puede funcionar de manera distinta como servidor de correo, web, base de datos entre otros, otorgándole al usuario final la información solicitada que conforman los sitios web (archivos e imágenes) alojadas en servidores para su preservación en ambiente seguro [2], de aquí radica su importancia y del porque muchas empresas implementan dicha arquitectura.

El objetivo de este proyecto es desarrollar un sistema web y móvil basado en la tecnología cliente-servidor para la gestión de asistencia laboral, por ende, se necesitó recopilar información sobre diversos artículos científicos que estén relacionados con la temática a resolver, una vez realizado esto se procedió a analizar las diferentes tecnologías encontradas y escoger las más óptimas para resolver el caso práctico propuesto dando como resultado final un sistema informático y una aplicación móvil.

El presente trabajo de titulación se ha organizado de la siguiente manera:

Capítulo 1: Introducción, marco contextual, problema a resolver y el objetivo general.

Capítulo 2: Marco teórico, solución al problema y resultados.

Capítulo 3: Conclusiones, bibliografía y anexos.

1.1 Marco Contextual

Para cualquier empresa, el supervisar la asistencia de sus empleados, siempre ha sido una de las mayores preocupaciones que los gerentes o jefes de departamento, día tras día desean conocer, debido a que, llevando un registro de entradas y salidas de los empleados, la empresa puede monitorear el desempeño laboral y conocer si en verdad los empleados están cumpliendo con las labores de la cuales fueron contratados.

Para este caso de estudio, el lugar donde se desarrollará el problema de la investigación será en las instalaciones de los laboratorios de la carrera de Ingeniería de Sistema, en donde los docentes imparten sus asignaturas, para lo cual deben firmar una hoja de registro de asistencia laboral por cada asignatura impartida por día y a pesar de que ya existe un control de asistencia laboral, este mismo actualmente no se encuentra automatizado.

1.2 Problema

La carrera de Ingeniería de Sistemas requiere gestionar la asistencia del personal docente de sus horas de clases en laboratorios de manera automatizada, dado a que esta gestión se lo realiza de manera manual en una hoja impresa, donde cada profesor debe firmar por día, dependiendo de la asignatura que imparte, pero eso no asegura que en verdad el docente haya impartido sus clases a las horas establecidas por el calendario de clases, por lo cual, la carrera necesita de un sistema web que gestione las asistencias por hora de entrada y salida de cada clase, mediante el uso de un lector de huellas para garantizar que el docente estuvo presente en su día laboral y además, una aplicación móvil que garantice la visualización de las asistencias, marcaciones, retrasos, entre otras opciones.

1.3 Objetivo General

Desarrollar un sistema web y móvil basado en la tecnología cliente – servidor para la gestión de asistencia laboral.

2. DESARROLLO

2.1 Marco teórico

2.1.1 *Arquitectura Física: Cliente-Servidor.*

Es un modelo de diseño en donde varias aplicaciones se ejecutan dentro de una misma red, claro ejemplo se encuentra las redes sociales en donde las empresas implementan muchas infraestructuras de servidores facilitando la comunicación entre usuarios [3].

La ventaja de usar esta arquitectura es que se pueden dividir las tareas de procesamientos de forma eficaz entre los clientes y servidores, dando la perspectiva al usuario de percibir todo como un solo sistema.

2.1.2 *Arquitectura Lógica: Modelo, Vista, Controlador (MVC).*

Es un patrón de diseño la cual consiste en la separación de sus componentes en tres capas distintas las cuales son la capa del modelo, vista y el controlador [4], aumentando la posibilidad de programar y diseñar software escalables, de fácil mantenimiento y reutilización del código.

2.1.3 *Sistema Operativo Centos 7.*

Es una de las múltiples distribuciones de Linux, destinado enteramente a servir como sistema operativo para servidores, también se caracteriza por ser gratuito y seguro [5], siendo el preferido en grandes empresas.

2.1.4 *Servidor Web Apache.*

Es el más popular, usado y preferido servidor web a nivel mundial, es un proyecto de código abierto, multiplataforma, dinámico y escalable diseñado especialmente para la transferencia de archivos, utilizando el protocolo de comunicación http [6].

2.1.5 Gestor de Base de Datos Relacional Mysql.

Base de datos relacional open-source, multiusuario y multiplataforma, es una de las 10 DBMS más utilizada a nivel mundial en diversas áreas científicas como telecomunicaciones, cloud computing, big data etc., debido a su gran capacidad de almacenar mucha información y de su alta velocidad de respuesta [7].

2.1.6 Lenguaje de Programación Backend: Framework Laravel 5.

Framework del lado del servidor, es de código abierto y sirve para desarrollar aplicaciones web utilizando el lenguaje de programación PHP 5 o 7, entre sus características principales están la alta reutilización, estabilidad, seguridad, concentrándose primordialmente en la lógica del negocio [8].

2.1.7 Lenguaje de Programación Frontend: Javascript.

Lenguaje de programación orientado a objetos, altamente dinámico, funciona en el lado del cliente y a diferencia de otros lenguajes, este es interpretado y no compilado utilizado principalmente para mejorar las interacciones del usuario con las páginas web [9].

2.1.8 Framework Web: Framework Bootstrap 4.

Framework del lado del cliente conocido también con el nombre de responsive web design, abarca una variedad de técnicas y pautas de diseño facilitando a los programadores el diseño de sus sitios web y así adaptarlos a distintos dispositivos entre móviles, tablets y TV. [10].

2.1.9 Lenguaje de Marcado para Hipertextos (HTML).

Lenguaje simple y general utilizado para desarrollar páginas web, se fundamenta en el protocolo HTTP el cual permite enviar los archivos desde el servidor y mostrarlo en un documento HTML desde un navegador web [11].

2.1.10 Protocolo de Comunicación HTTP.

Protocolo de comunicación de respuesta de solicitud simple basado en el paradigma petición-respuesta, utilizado principalmente para la transferencia de forma eficiente de la información solicitada entre servidores web y clientes [12].

2.1.11 Protocolo de Comunicación ApisRest.

Arquitectura web que trabaja en conjunto con el protocolo HTTP, facilitando la creación de servicios web y también la obtención de datos del servidor en formato XML y JSON [13].

2.1.12 Arquitectura cliente: App Android.

Sistema operativo open-source para dispositivos móviles, creado por Google, basado en Linux y estructurado con el lenguaje de programación Java [14].

2.1.13 Arquitectura cliente: Aplicación en Java 8.

Es la versión más reciente de Java, la cual incluye mejoras a su predecesora de java 7, es un lenguaje basado en el paradigma orientado a objetos permitiéndole tener una sintaxis limpia y elegante haciéndola una de las más utilizadas en la educación e investigación [15].

2.1.14 Estructura de datos.

Forma en la cual una computadora puede organizar datos complejos utilizando distintos tipos de estructuras como las pilas, colas, listas, árboles o grafos con el objetivo de mejorar la eficiencia en el manejo de los datos [16].

2.1.15 Modelo de Proceso de Software: Metodología Ágil ICONIX.

Las metodologías de desarrollo ágil enfatizan en mejorar los aspectos negativos de la metodología tradicional, por lo cual seleccionar una

metodología ágil es más adecuada para el desarrollo de aplicaciones web dado a sus múltiples ventajas [17], por ende, ICONIX es iterativo e incremental, utiliza fases de la metodología RUP y XP, convirtiéndolo así, en una metodología pesada-ligera abarcando gran parte del ciclo clásico del desarrollo de un sistema [18].

Esta metodología no se diferencia mucho de las distintas metodologías ya existentes para el desarrollo web como son la OOHDM, SOHDM, WSDM, UWE, IEWD entre otros, debido a que estas metodologías contienen fases para el desarrollo del software muy similares a la metodología ICONIX, las cuales son fase de análisis de requisitos, fase análisis y diseño preliminar, fase de diseño y fase de implementación [19].

2.2 Solución del problema

2.2.1 Materiales

Entre los materiales utilizados para la elaboración del caso práctico son:

Tabla 1: Materiales utilizados

Materiales/Recursos	
Lenguaje de programación	PHP 5.6.30
Framework	Laravel 5.4
Gestor de base de datos	MySQL 5.7
Metodología de desarrollo	ICONIX
Aplicaciones Clientes	Aplicación de escritorio: Java 8, Aplicación Móvil: Android
Servidor	S.O: Centos 7; Servidor web: Apache
Otros	Router, Lector de Huellas

Fuente: Elaboración propia

2.2.2 Desarrollo de la Metodología ICONIX

2.2.2.1 Fase 1: Análisis de requerimientos

En esta fase como primer paso se realizó una serie de entrevistas no formales al cliente para identificar todos los requisitos funcionales y no funcionales que servirán como punto de partida para el desarrollo del sistema, el resultado puede verse en el siguiente anexo (Ver Anexo A).

Una vez que se haya obtenido una lista de los requisitos del sistema, el siguiente paso consistió en elaborar los respectivos casos de uso para cada uno de los requisitos funcionales planteados anteriormente, dando a conocer como el usuario interactúa con los diferentes módulos del sistema (Ver Anexo B).

El siguiente paso consistió en elaborar un modelo de dominio, este modelo servirá como punto de partida para la elaboración del diagrama de clases y el relacional, además ayudará a tener una idea básica de cómo funcionará el sistema (Ver Anexo C).

Como último paso, se diseñó una serie de prototipos de interfaz de usuarios y aunque estos prototipos pueden variar con el paso del tiempo, dependiendo de las indicaciones del usuario, este servirá para que el cliente tenga una idea cómo se verá el sistema a futuro (Ver Anexo D).

2.2.2.2 Fase 2: Análisis y diseño preliminar

En esta fase se procedió a la elaboración de los casos de uso extendido, las cuales consisten en una serie de fichas detallando de mejor manera cada caso de uso (Ver Anexo E).

Haciendo uso del modelo de dominio elaborado en la fase anterior, se continuó con la elaboración del diagrama de clases (Ver Anexo F) y del diagrama relacional de la base de datos (Ver Anexo G).

Finalmente se procedió con la elaboración de los diagramas de robustez (Ver Anexo H), estos diagramas se elaboran por cada caso de uso y su finalidad es conocer cuáles serán las actividades

que realizará el usuario dentro del sistema, siendo el diagrama de robustez un reemplazo del diagrama de actividades.

2.2.2.3 Fase 3: Diseño

En la fase de diseño, ICONIX hace uso del diagrama de secuencia, para lo cual se elaboró un diagrama por cada caso de uso (Ver Anexo I), con estos diagramas se consigue conocer cuál será el proceso del funcionamiento normal (de inicio a fin) de cada proceso y también conocer las posibles incidencias que puedan ocurrir con el mismo.

2.2.2.4 Fase 4: Implementación

En esta fase final se diseñó el diagrama de despliegue (Ver Anexo J) el cual permite modelar la arquitectura física del sistema, es decir, la manera en cómo se distribuirán de forma física los componentes (hardware) que conforman el software final.

Finalmente se comenzó con la parte de codificación, como su palabra lo indica, consiste en tipiar el código de programación necesario para la elaboración del software.

2.3 Resultados

Después de la aplicación de los diversos temas detallados en este informe, se obtuvieron los siguientes resultados:

En primera instancia se encuentra la realización del sistema web, este mismo será utilizado para el gestionamiento administrativo del sitio web, la cual fue desarrollado con el framework Laravel en su versión 5 utilizando una arquitectura de 3 capas (MVC).

Para la implementación de la arquitectura cliente-servidor, se utilizó el sistema operativo Centos 7 como servidor, en la cual se configuró el servidor web Apache y la base de datos MariaDB.

El sistema web cuenta con un login (Ver anexo K), en donde deberá ingresar su usuario y contraseña permitiéndole así el ingreso al sitio web y dependiendo de

su rol, se re direccionará a un dashboard con funciones distintas, los roles por defecto son administrador y docentes.

En caso de que sea un usuario administrador le aparecerá el siguiente dashboard (ver Anexo L), el mismo que consta de las siguientes opciones (Ver anexo M):

- Gestión de roles.
- Gestión de usuarios.
- Gestión de semestres.
- Gestión de laboratorios.
- Gestión de asignaturas.
- Gestión de años lectivos.
- Gestión de calendarios de clases.
- Gestión de asistencias de docentes.
- Gestión de reportes.

Para el usuario docente, le aparecerá el siguiente dashboard (Ver Anexo N) la cual constará de las siguientes opciones (Ver anexo O):

- Gestión de perfil de usuario.
- Visualización de calendario de clases.
- Visualización de asistencias.

También se empleó las estructuras de datos, para eso se hizo uso de los TAD (ver anexo P) facilitando así el uso de los algoritmos de las pilas, colas y listas. La estructura de datos tipo lista se empleó en los módulos de gestión de roles, usuarios, semestres, laboratorios, asignaturas, años lectivos, debido a que esta estructura facilita la rapidez en el manejo de los datos, la estructura tipo pila se la aplicó en la gestión de calendarios, ya que cada calendario tiene un máximo de 30 asignaturas por hora, entonces se tomó al calendario como una pila de tamaño 30 y por último la estructura tipo cola se utilizó en la gestión de la asistencia de los docentes, ya que por su característica FIFO, el primer docente que llegue a marcar su asistencia será el primero en ser registrado en el sistema.

Dentro de las funciones principales del sistema web se encuentra la realización de calendarios (Ver anexo Q) el cual permite al administrador tener un mejor manejo en cuestión de crear los calendarios de clases y también la realización

de reportes (Ver anexo R) permitiéndole así crear reportes desde un intervalo de tiempo.

Otro de los resultados obtenidos fue de una aplicación de escritorio realizado en java (Ver anexo S) la cual permite registrar la huella digital del docente (Ver anexo T) para posteriormente utilizarla en la toma de asistencia (Ver anexo U), para ello el docente deberá seleccionar su asignatura y laboratorio donde dará su cátedra luego dar clic en verificar huella y luego dar clic en guardar asistencia registrando así la fecha y hora de su entrada a clases.

Luego el administrador podrá revisar en el sitio web todas las asistencias realizadas en los días de clases respectivos (Ver anexo V).

Como último resultado se elaboró una aplicación móvil que fue desarrollada en Android utilizando web services y el enfoque de progressive web app, la cual consiste en crear aplicaciones móviles híbridas, es decir, reutilizar los recursos usados en la elaboración del sistema web (código html, css, javascript) y llevarlo a una aplicación nativa de Android, reduciendo así el tiempo de elaboración y conforme se realice cambios en el sistema web, la aplicación móvil irá cambiando progresivamente.

Las funcionalidades de la aplicación móvil son las mismas con las que cuenta el sistema web, incluyendo un login (Ver anexo W) y los dashboard de administrador (Ver anexo X).

3. CONCLUSIONES

En base a la ayuda de los artículos científicos investigado y a la aplicación de la metodología de desarrollo de software escogida, se logró cumplir con cada uno de los requerimientos dados en el caso práctico propuesto, ofreciendo como producto final un sistema web y móvil que ayude en la gestión de las asistencias del personal docente de la carrera de Ingeniería de Sistemas.

Se empleó correctamente la arquitectura cliente-servidor, haciendo que el sistema no sea stand-alone, dando la posibilidad de que el software sea usado por muchos clientes, a cualquier hora y en cualquier lugar siempre y cuando el usuario tenga acceso a la red.

Al ser este un caso práctico perteneciente a la asignatura de programación III, se empleó el uso de estructuras de datos como fueron las listas, pilas y colas, con el fin de mejorar la eficiencia en el manejo de los datos traídos desde el servidor.

El desarrollo del sistema web permite llevar un control acerca de la asistencia de los docentes, y el tener una aplicación móvil desarrollada en Android, facilita al administrador del sitio web un mejor control de las asistencias de los docentes diariamente.

Bibliografía

- [1] A. S. F. F. Nelly Lisbeth Hernández, «Computación en la nube,» *Mundo FESC*, vol. 2, nº 8, pp. 46-51, 2014.
- [2] O. T. Mercy y C. L. Luna, «Una arquitectura basada en software libre para archivos web,» *Revista Venezolana de Información, Tecnología y Conocimiento*, vol. 10, nº 1, pp. 53 - 72, 2013.
- [3] T. Duong-Ba, B. Bose y T. Nguyen, «Distributed Client-Server Assignment for Online Social Network Applications,» in *IEEE Transactions on Emerging Topics in Computing*, vol. 2, nº 4, pp. 422 - 435, 2014.
- [4] J. R. M. Ríos, N. M. L. Mora, M. P. Z. Ordóñez y E. L. L. Sojos, «Evaluación de los Frameworks en el Desarrollo de Aplicaciones Web con Python,» *Revista Latinoamericana de Ingeniería de Software*, vol. 4, nº 4, pp. 201-207, 2016.
- [5] M. R. L. Vallejo, «Hacking ético. Vulnerabilidad de Sistemas Operativos en el acceso por contraseñas,» *Revista Publicando*, vol. 4, nº 10, pp. 31-51, 2017.
- [6] M. B. Seyyar, F. Ö. Çatak y E. Gül, «Detection of attack-targeted scans from the Apache HTTP Server access logs,» *Applied Computing and Informatics*, vol. 14, nº 1, pp. 28-36, 2017.
- [7] J. F. C. Altamirano, C. E. Terán, H. F. V. Ballesteros y R. C. A, «Análisis e implementación de una solución de automatización de configuración de enlaces de datos en una Red MPLS para un ambiente de Cloud Computing,» *Dominio de las Ciencias*, vol. 3, nº 4, pp. 381-399, 2017.
- [8] L. Chen, W. Huang, A. Sui, D. Chen y C. Sun, «The online education platform using Proxmox and noVNC technology based on Laravel framework,» *IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*, pp. 487-491, 2017.
- [9] D. Hedin, L. Bello y A. Sabelfeld, «Value-Sensitive Hybrid Information Flow Control for a JavaScript-Like Language,» in *IEEE 28th Computer Security Foundations Symposium*, pp. 351-365, 2015.
- [10] D. González y M. C. Marcos, «Responsive web design: diseño multidispositivo para mejorar la experiencia de usuario,» *BiD: Textos universitarios de biblioteconomía i documentación*, nº 31, 2013.
- [11] J. M. Ríos y M. Z. Ordóñez, «Metodologías de Desarrollo en Aplicaciones WEB,» *ARJÉ. Revista de Postgrado FaCE-UC*, vol. 11, nº 21, pp. 245-270, 2017.
- [12] P. Wei y M. S. Zhiguo Hong, «Performance Analysis of HTTP and FTP Based on,» in *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*, pp. 1-4, 2016.
- [13] O. A. Leandro, R. Verborgh y J. C. Corrales, «A topic modeling approach for web service annotation,» *Revista Ingenierías Universidad de Medellín*, vol. 13, nº 24, pp. 147-164, 2014.
- [14] U. L. Christian y A. N. Cadavid, «Framework for malware analysis in Android,» *Sistemas & Telemática*, vol. 14, nº 37, pp. 45-56, 2016.

- [15] L. Gajardo y L. Mateu, «Análisis semántico de programas escritos en java,» *Theoria*, nº 13, pp. 39-49, 2004.
- [16] L. Joyanes, *Estructura de datos en Java*, Madrid: MCGRAW-HILL / INTERAMERICANA DE ESPAÑA, 2008, p. 5.
- [17] J. Molina, M. Zea, J. Honores y A. Gómez, «Analysis Methodologies Web Application Development,» *International Journal of Applied Engineering Research*, vol. 11, nº 16, pp. 9070-9078, 2016.
- [18] ICONIX, «Manual Introductorio de Iconix,» 2013. [En línea]. Available: <http://ima.udg.edu/~sellares/EINF-ES2/Present1011/MetodoPesadesICONIX.pdf>.
- [19] J. Molina, M. Zea, M. Contenido y F. García, «Comparación de metodologías en aplicaciones web,» *3C Tecnología*, vol. 7, nº 1, pp. 1-19, 2018.

ANEXOS

Anexo A. Requisitos funcionales y no funcionales

Tabla 2: RF 01 -Gestión de usuarios

Número de requisito	RF 01
Nombre de requisito	Gestión de usuarios.
Nivel	Dominio Sistema
Tipo	Funcional No funcional
Descripción	El administrador del sistema podrá realizar un CRUD de usuarios con distintos roles.
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/ Opcional

Fuente: Elaboración propia.

Tabla 3: RF 02 - Gestión de roles

Número de requisito	RF 02
Nombre de requisito	Gestión de roles.
Nivel	Dominio Sistema
Tipo	Funcional No funcional
Descripción	El administrador del sistema podrá realizar un CRUD de roles para posteriormente asignarles a los usuarios registrados en el sistema.
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/ Opcional

Fuente: Elaboración propia.

Tabla 4: RF 03 - Login

Número de requisito	RF 03
Nombre de requisito	Login.
Nivel	Dominio Sistema
Tipo	Funcional No funcional
Descripción	Este control se da por seguridad, para ingreso de usuarios autorizados al sistema informático el cual consta de nombre de usuario y contraseña..
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/ Opcional

Fuente: Elaboración propia.

Tabla 5: RF 04 - Gestión de planificación del calendario de clases

Número de requisito	RF 04
Nombre de requisito	Gestión de planificación del calendario de clases
Nivel	Dominio Sistema
Tipo	Funcional No funcional
Descripción	Este control permite realizar el calendario de clases, establecer los horarios, como días laborales, no laborales e imprevistos.
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/ Opcional

Fuente: Elaboración propia.

Tabla 6: RF 05 - Gestión de asistencia laboral

Número de requisito	RF 05
Nombre de requisito	Gestión de asistencia laboral
Nivel	Dominio Sistema
Tipo	Funcional No funcional
Descripción	Esta opción permitirá controlar la asistencia de los docentes a sus horarios de clases, trabajará en conjunto con el lector de huellas.
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/ Opcional

Fuente: Elaboración propia.

Tabla 7: RF 06 -Gestión de laboratorios.

Número de requisito	RF 06
Nombre de requisito	Gestión de laboratorios.
Nivel	Dominio Sistema
Tipo	Funcional No funcional
Descripción	Esta opción permitirá gestionar los distintos laboratorios, con sus respectivos nro. de salón y disponibilidad.
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/ Opcional

Fuente: Elaboración propia.

Tabla 8: RF 07 - Gestión de asignaturas

Número de requisito	RF 07
Nombre de requisito	Gestión de asignaturas
Nivel	Dominio Sistema
Tipo	Funcional No funcional
Descripción	Esta opción permitirá gestionar las distintas asignaturas impartidas por semestre.
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/ Opcional

Fuente: Elaboración propia.

Tabla 9: RF 08 - Gestión de semestres y años lectivos.

Número de requisito	RF 08
Nombre de requisito	Gestión de semestres y años lectivos.
Nivel	Dominio Sistema
Tipo	Funcional No funcional
Descripción	Esta opción permitirá gestionar los semestres de cada año lectivo.
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/ Opcional

Fuente: Elaboración propia.

Tabla 10: RF 09 - Reportes de asistencias mensuales

Número de requisito	RF 09
Nombre de requisito	Reportes de asistencias mensuales

Nivel	Dominio Sistema
Tipo	Funcional No funcional
Descripción	Esta opción permitirá al administrador realizar reportes mensuales de las asistencias de los docentes.
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/ Opcional

Fuente: Elaboración propia.

Tabla 11: RNF 01 - Diseñar la solución utilizando una arquitectura n-thiers

Número de requisito	RNF 01
Nombre de requisito	Diseñar la solución utilizando una arquitectura n-thiers.
Nivel	Dominio Sistema
Tipo	Funcional No funcional
Descripción	El software deberá ser realizado utilizando una arquitectura n-capas.
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/ Opcional

Fuente: Elaboración propia.

Tabla 12: RNF 02 - Diseñar la solución utilizando una arquitectura cliente-servidor

Número de requisito	RNF 02
Nombre de requisito	Diseñar la solución utilizando una arquitectura cliente-servidor.
Nivel	Dominio Sistema
Tipo	Funcional No funcional

Descripción	El software deberá ser realizado utilizando una arquitectura cliente-servidor.		
Prioridad del requisito	Alta/Esencial	Media/Deseado	Baja/ Opcional

Fuente: Elaboración propia.

Tabla 13: RNF 03 - Utilizar estructura de datos

Número de requisito	RNF 03		
Nombre de requisito	Utilizar estructura de datos.		
Nivel	Dominio	Sistema	
Tipo	Funcional	No funcional	
Descripción	El software deberá ser realizado utilizando estructura de datos como listas, pilas y colas.		
Prioridad del requisito	Alta/Esencial	Media/Deseado	Baja/ Opcional

Fuente: Elaboración propia.

Tabla 14: RNF 04 - Validaciones

Número de requisito	RNF 04		
Nombre de requisito	Validaciones		
Nivel	Dominio	Sistema	
Tipo	Funcional	No funcional	
Descripción	El software deberá realizar las respectivas validaciones de ingreso y salida de datos.		
Prioridad del requisito	Alta/Esencial	Media/Deseado	Baja/ Opcional

Fuente: Elaboración propia.

Tabla 15: RNF 05 - Aplicación móvil

Número de requisito	RNF 05
Nombre de requisito	Aplicación móvil
Nivel	Dominio Sistema
Tipo	Funcional No funcional
Descripción	Implementar una aplicación móvil
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/ Opcional

Fuente: Elaboración propia.

Tabla 16: RNF 06- Diseño responsivo.

Número de requisito	RNF 06
Nombre de requisito	Diseño responsivo.
Nivel	Dominio Sistema
Tipo	Funcional No funcional
Descripción	El software debe adaptarse a cualquier dispositivo móvil.
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/ Opcional

Fuente: Elaboración propia.

Tabla 17: RNF 07 - Control de permisos

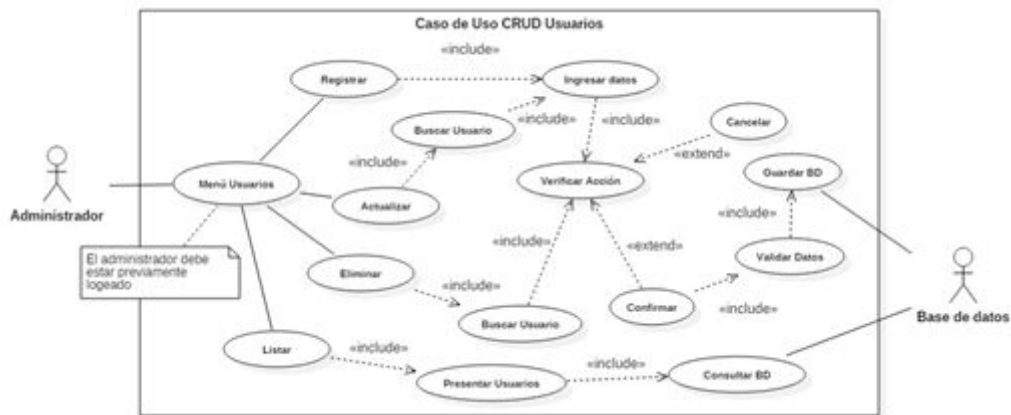
Número de requisito	RNF 07
Nombre de requisito	Control de permisos
Nivel	Dominio Sistema

Tipo	Funcional	No funcional	
Descripción	El sistema otorgará permisos a los usuarios para que acceda o no a distintas funcionalidades del software.		
Prioridad del requisito	Alta/Esencial	Media/Deseado	Baja/ Opcional

Fuente: Elaboración propia.

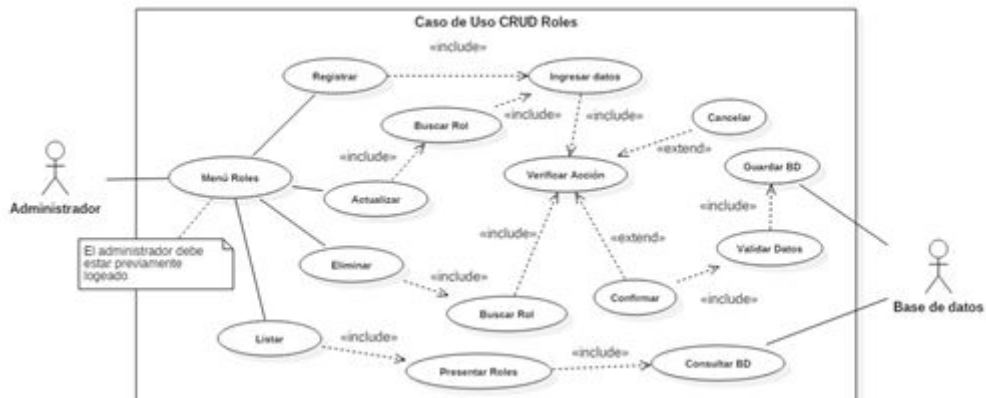
Anexo B. Modelo de caso de usos

Ilustración 1: Caso de Uso para RF-01



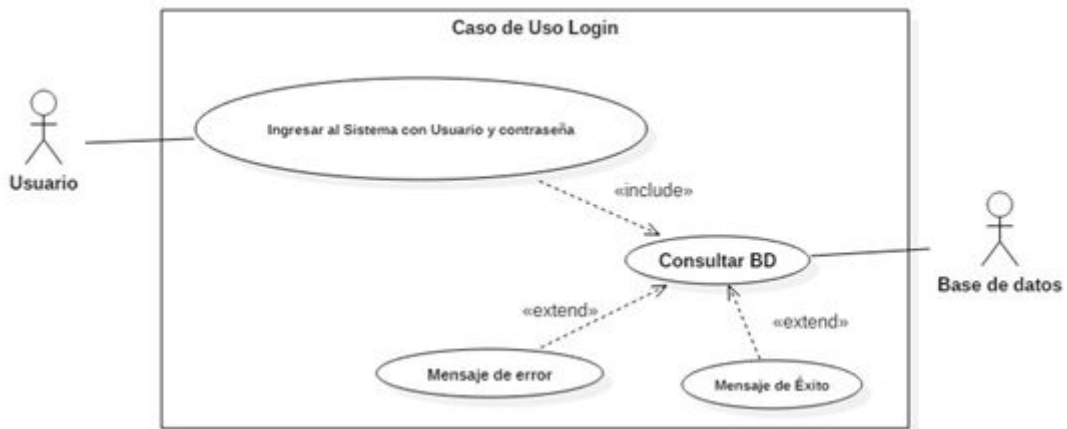
Fuente: Elaboración propia

Ilustración 2: Caso de Uso para RF-02



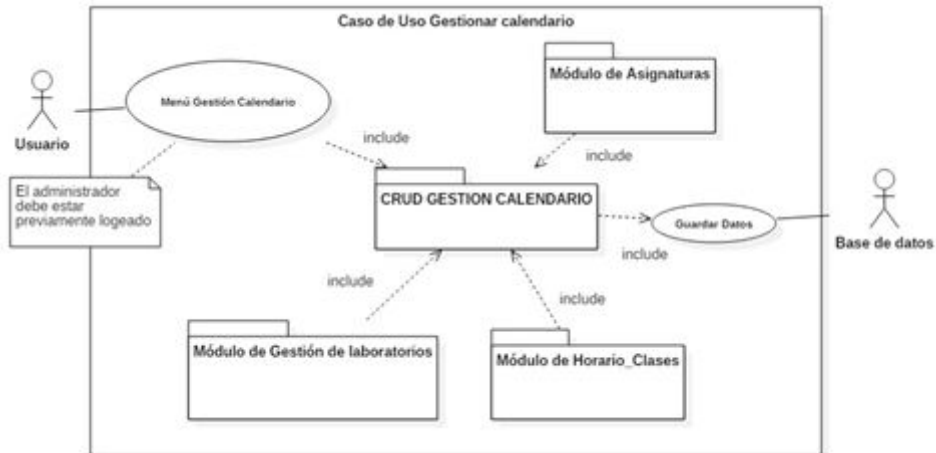
Fuente: Elaboración propia

Ilustración 3: Caso de Uso para RF-03



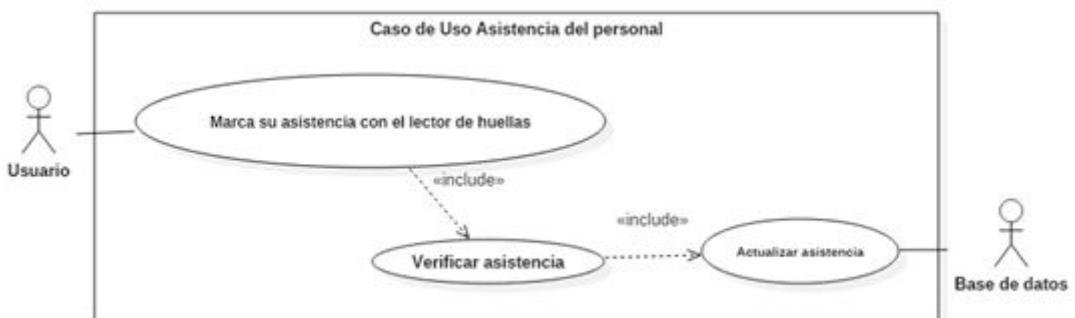
Fuente: Elaboración propia

Ilustración 4: Caso de Uso para RF-04



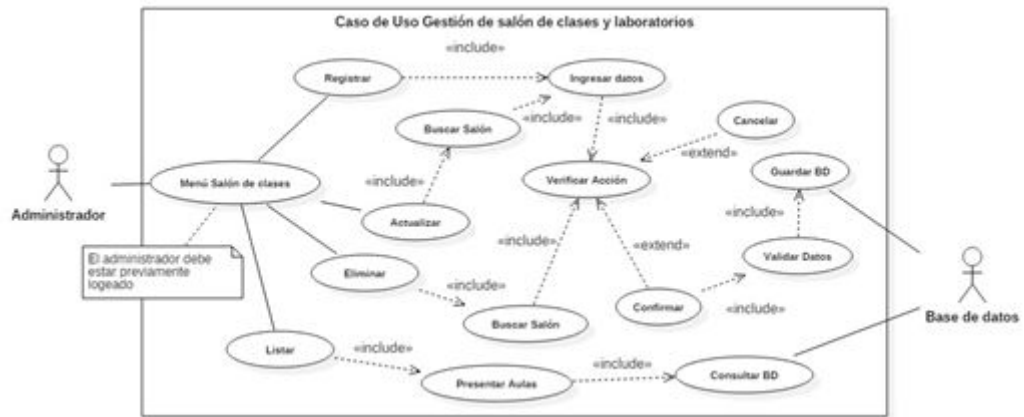
Fuente: Elaboración propia

Ilustración 5: Caso de Uso para RF-05



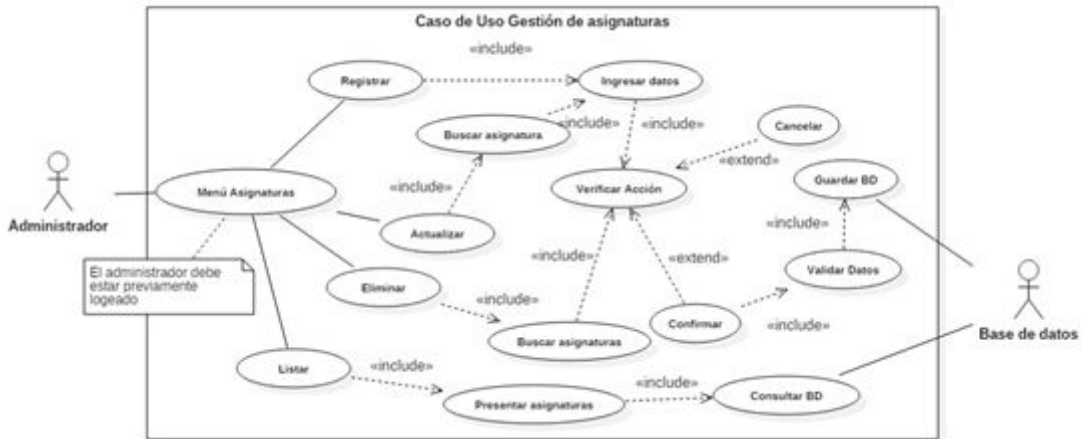
Fuente: Elaboración propia

Ilustración 6: Caso de Uso para RF-06



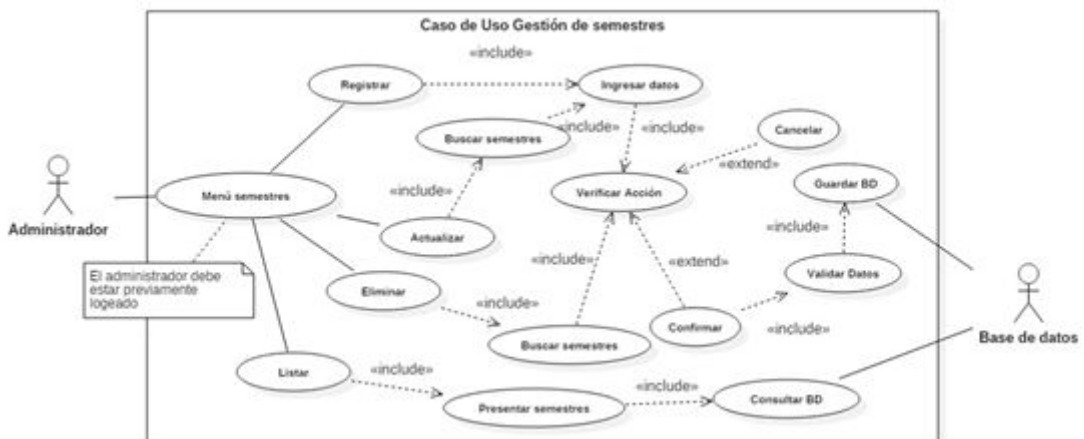
Fuente: Elaboración propia

Ilustración 7: Caso de Uso para RF-07



Fuente: Elaboración propia

Ilustración 8: Caso de Uso para RF-08



Fuente: Elaboración propia

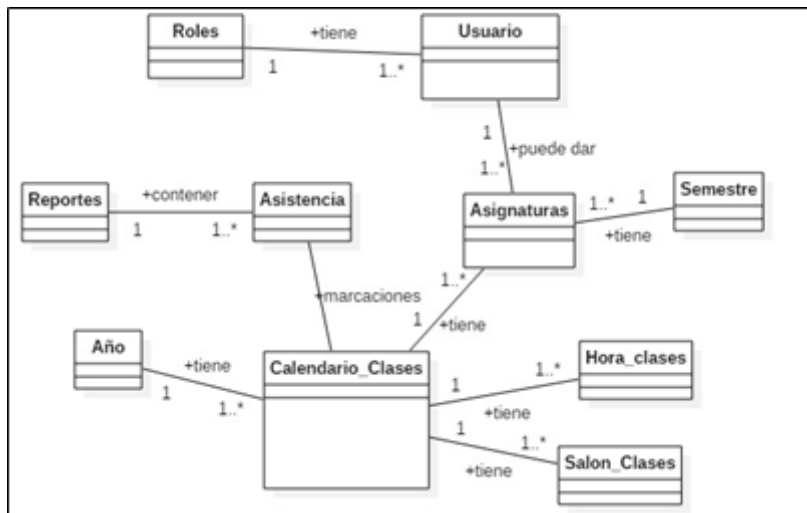
Ilustración 9: Caso de Uso para RF-09



Fuente: Elaboración propia

Anexo C. Modelo de dominio

Ilustración 10: Modelo de dominio



Fuente: Elaboración propia

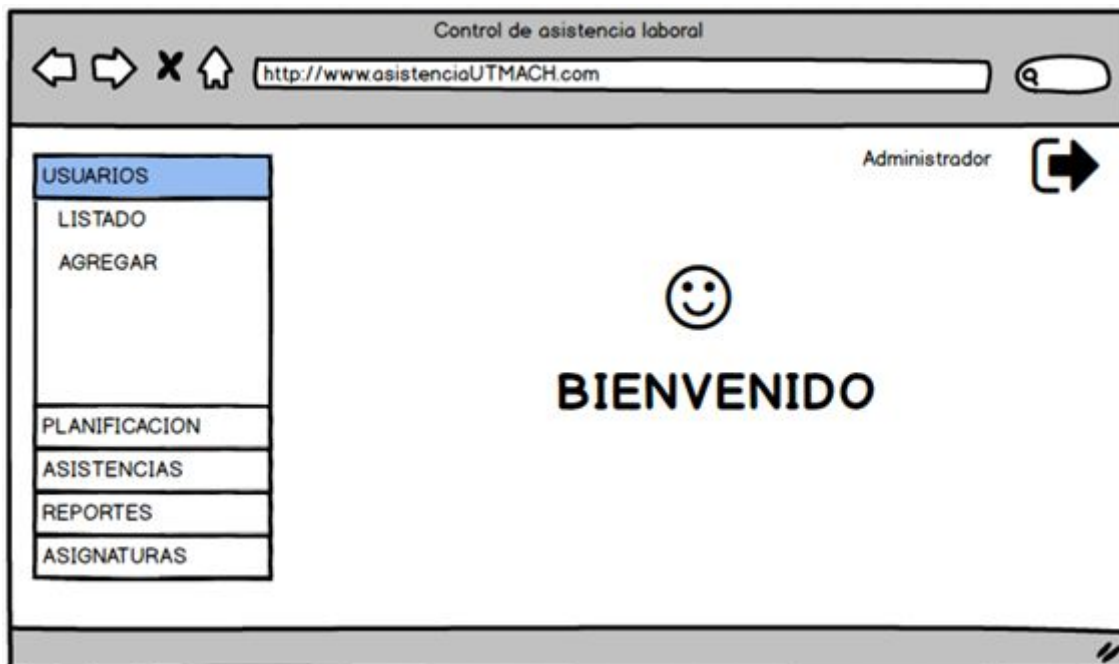
Anexo D. Prototipación rápida

Ilustración 11: Login



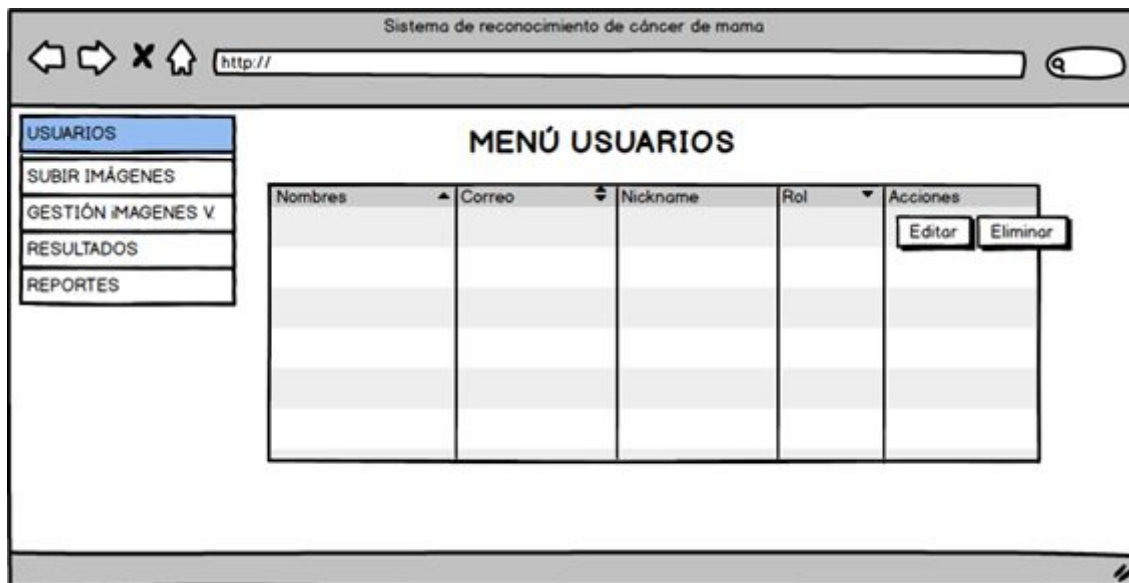
Fuente: Elaboración propia

Ilustración 12: Panel de administración



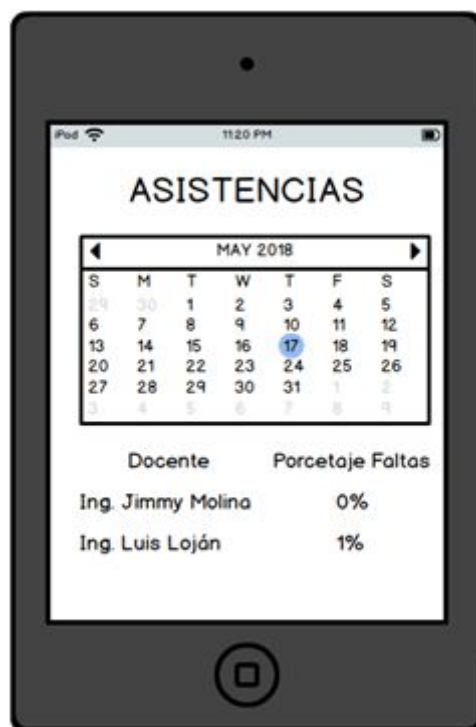
Fuente: Elaboración propia

Ilustración 13: Listado de Usuarios



Fuente: Elaboración propia

Ilustración 14: Aplicación móvil



Fuente: Elaboración propia

Anexo E. Caso de uso expandido

Tabla 18: Caso de Uso Expandido: CRUD Usuarios

Caso de Uso: CRUD USUARIOS	
Descripción	Permite gestionar los usuarios del sistema
Actores	Administrador (Iniciador), Base de datos.
Propósito	Permitir que el sistema gestione los usuarios que van a utilizar el sistema informático.
Visión General	El administrador podrá crear, buscar, eliminar o editar la información de los usuarios que van hacer uso del sistema.
Tipo	Primario y Esencial
Referencias	Requisito 1

Curso Típico de Eventos

Secuencia Normal (acción del actor)

1. Este caso empieza cuando el administrador selecciona menú usuario, y escoge una de las siguientes opciones:
 - 1.1. Registrar usuario
 - 1.2. Actualizar usuario
 - 1.3. Eliminar usuario
 - 1.4. Listar Usuario
2. En caso de escoger registrar usuario, el administrador deberá ingresar los datos solicitados en el formulario de registro, luego deberá dar clic en confirmar o cancelar acción, en caso de elegir confirmar el software validará si los datos están correctamente ingresados, si lo están se procederá a guardar el registro en la base de datos.
3. En caso de elegir actualizar usuario, el sistema le pedirá ingresar el código de usuario para realizar la búsqueda del mismo y luego seguirá los procedimientos vistos en el punto 2.
4. En caso de elegir eliminar usuario, el sistema le pedirá ingresar el código de usuario para realizar la búsqueda del mismo, una vez encontrado al usuario le preguntará si desea o no eliminar el usuario.
5. En el caso de elegir listar usuarios, se presentará una lista de todos los usuarios registrados en el sistema para esto el sistema previamente deberá hacer una consulta en la base de datos.

Secuencia(s) Alternativas(s)	<p>En caso de:</p> <p>Ocurrir algún error en la validación de los datos ingresados, se le notificará al administrador de los campos erróneos para que proceda a corregir.</p> <p>Ocurrir algún error en la búsqueda del usuario, se le notificará al administrador que dicho usuario no se encuentra registrado en el sistema.</p>
Precondiciones	El administrador debió haber iniciado sesión en el sistema.
Post condiciones	Usuarios registrados, modificados o eliminados.
Comentarios	Ninguno

Fuente: Elaboración propia

Tabla 19: Caso de Uso Expandido: CRUD ROLES

Caso de Uso: CRUD ROLES	
Descripción	Permite gestionar los roles dentro del sistema
Actores	Administrador (Iniciador), Base de datos.
Propósito	Permitir que el sistema gestione los roles que van a utilizar los usuarios dentro del sistema informático.
Visión General	El administrador podrá crear, buscar, eliminar o editar los roles de los usuarios que van hacer uso del sistema.
Tipo	Primario y Esencial
Referencias	Requisito 2
Curso Típico de Eventos	

<p style="text-align: center;">Secuencia Normal (acción del actor)</p>	<ol style="list-style-type: none"> 1. Este caso empieza cuando el administrador selecciona menú roles, y escoge una de las siguientes opciones: <ol style="list-style-type: none"> 1.1. Registrar rol 1.2. Actualizar rol 1.3. Eliminar rol 1.4. Listar rol 2. En caso de escoger registrar rol, el administrador deberá ingresar los datos solicitados en el formulario de registro, luego deberá dar clic en confirmar o cancelar acción, en caso de elegir confirmar el software validará si los datos están correctamente ingresados, si lo están se procederá a guardar el registro en la base de datos. 3. En caso de elegir actualizar rol, el sistema le pedirá ingresar el código de rol para realizar la búsqueda del mismo y luego seguirá los procedimientos vistos en el punto 2. 4. En caso de elegir eliminar rol, el sistema le pedirá ingresar el código de rol para realizar la búsqueda del mismo, una vez encontrado el rol le preguntará si desea o no eliminar el rol. 5. En el caso de elegir listar roles, se presentará una lista de todos los roles registrados en el sistema para esto el sistema previamente deberá hacer una consulta en la base de datos.
<p>Secuencia(s) Alternativas(s)</p>	<p>En caso de:</p> <ul style="list-style-type: none"> · Ocurrir algún error en la validación de los datos ingresados, se le notificará al administrador de los campos erróneos para que proceda a corregir. · Ocurrir algún error en la búsqueda del rol, se le notificará al administrador que dicho rol no se encuentra registrado en el sistema.

Precondiciones	El administrador debió haber iniciado sesión en el sistema.
Post condiciones	Roles registrados, modificados o eliminados.
Comentarios	Ninguno

Fuente: Elaboración propia

Tabla 20: Caso de Uso Expandido: Login

Caso de Uso: Login	
Descripción	Permite ingresar al panel de administración.
Actores	Administrador (Iniciador), Base de datos.
Propósito	Permitir que el usuario inicie sesión dentro del sistema informático.
Visión General	El administrador o el usuario registrado escribirá su usuario y contraseña para acceder al sistema.
Tipo	Primario y Esencial
Referencias	Requisito 3
Curso Típico de Eventos	
Secuencia Normal (acción del actor)	<ol style="list-style-type: none"> 1. Este caso empieza cuando el usuario digita su usuario y contraseña en el formulario Login, luego se realizará una consulta en la base de datos para comparar si existe un registro del usuario, al final se le dará un mensaje de éxito o fracaso.

Secuencia(s) Alternativas(s)	En caso de: Ocurrir algún error en la consulta con la base de datos de los datos ingresados, se le notificará al usuario si el usuario o contraseña están mal ingresados.
Precondiciones	Ingresar a la URL del sitio web.
Post condiciones	Ingreso al sistema web.
Comentarios	Ninguno

Fuente: Elaboración propia

Tabla 21: Caso de Uso Expandido: Gestionar calendario

Caso de Uso: Gestionar calendario	
Descripción	Permite gestionar el calendario de clases.
Actores	Administrador (Iniciador), Base de datos.
Propósito	Permitir que el administrador gestione los calendarios de clases.
Visión General	El administrador podrá crear, buscar, eliminar o editar los calendarios de clases.
Tipo	Primario y Esencial
Referencias	Requisito 4
Curso Típico de Eventos	

Secuencia Normal (acción del actor)	<ol style="list-style-type: none"> 1. Este caso empieza cuando el administrador ingresa al menú de gestión de calendarios el cual podrá elegir entre las opciones de ingresar, buscar, eliminar o listar los calendarios, para eso se hará uso de los registros de los módulos de asignaturas, horarios de clases y gestión de laboratorios. 2. Una vez terminado la gestión de los calendarios se procederá a guardar los cambios en la base de datos.
Secuencia(s) Alternativas(s)	<p>En caso de:</p> <p>Ocurrir algún error en la consulta con la base de datos de los datos ingresados, se le notificará al administrador de dicho error.</p>
Precondiciones	El administrador debió haber iniciado sesión.
Post condiciones	Gestión de los calendarios finalizado.
Comentarios	Ninguno

Fuente: Elaboración propia

Tabla 22: Caso de Uso Expandido: Asistencia del Personal

Caso de Uso: Asistencia del Personal	
Descripción	Permite tomar la asistencia del personal docente.
Actores	Usuario (Iniciador), Base de datos.
Propósito	Permitir que los docentes puedan registrar su asistencia en el sistema.
Visión General	Los usuarios registrados marcarán su asistencia con un lector de huellas.
Tipo	Primario y Esencial

Referencias	Requisito 5
Curso Típico de Eventos	
Secuencia Normal (acción del actor)	<ol style="list-style-type: none"> 1. El docente se acercará hacia el lector de huellas donde colocará su dedo y el sistema comenzará a verificar su huella digital. 2. Una vez terminado la verificación se procederá a registrar su asistencia en la base de datos.
Secuencia(s) Alternativas(s)	En caso de: Ocurrir algún error con la verificación de su asistencia se notificará el error al usuario.
Precondiciones	El usuario debe estar registrado en el sistema.
Post condiciones	Actualización de su asistencia laboral en el sistema.
Comentarios	Ninguno

Fuente: Elaboración propia

Tabla 23: Caso de Uso Expandido: Gestión de laboratorios

Caso de Uso: Gestión de salón de clases y laboratorios	
Descripción	Permite gestionar los salones de clases y laboratorios.
Actores	Administrador (Iniciador), Base de datos.
Propósito	Permitir que el sistema gestione los salones de clases y laboratorios que van a utilizar el sistema informático.
Visión General	El administrador podrá crear, buscar, eliminar o editar la información de los salones de clases y laboratorios que van a hacer uso del sistema.
Tipo	Primario y Esencial

Referencias	Requisito 6
Curso Típico de Eventos	
Secuencia Normal (acción del actor)	<ol style="list-style-type: none"> 1. Este caso empieza cuando el administrador selecciona menú salón de clases, y escoge una de las siguientes opciones: <ol style="list-style-type: none"> 1.1. Registrar laboratorio 1.2. Actualizar laboratorio 1.3. Eliminar laboratorio 1.4. Listar laboratorio 2. En caso de escoger registrar laboratorio, el administrador deberá ingresar los datos solicitados en el formulario de registro, luego deberá dar clic en confirmar o cancelar acción, en caso de elegir confirmar el software validará si los datos están correctamente ingresados, si lo están se procederá a guardar el registro en la base de datos. 3. En caso de elegir actualizar laboratorio, el sistema le pedirá ingresar el código de laboratorio para realizar la búsqueda del mismo y luego seguirá los procedimientos vistos en el punto 2. 4. En caso de elegir eliminar laboratorio, el sistema le pedirá ingresar el código de laboratorio para realizar la búsqueda del mismo, una vez encontrado al usuario le preguntará si desea o no eliminar el laboratorio. 5. En el caso de elegir listar laboratorio, se presentará una lista de todos los laboratorios registrados en el sistema para esto el sistema previamente deberá hacer una consulta en la base de datos.

Secuencia(s) Alternativas(s)	<p>En caso de:</p> <p>Ocurrir algún error en la validación de los datos ingresados, se le notificará al administrador de los campos erróneos para que proceda a corregir.</p> <p>Ocurrir algún error en la búsqueda del laboratorio, se le notificará al administrador que dicho laboratorio no se encuentra registrado en el sistema.</p>
Precondiciones	El administrador debió haber iniciado sesión en el sistema.
Post condiciones	Laboratorios registrados, modificados o eliminados.
Comentarios	Ninguno

Fuente: Elaboración propia

Tabla 24: Caso de Uso Expandido: Gestión de asignaturas

Caso de Uso: Gestión de asignaturas	
Descripción	Permite gestionar las asignaturas.
Actores	Administrador (Iniciador), Base de datos.
Propósito	Permitir que el sistema gestione las asignaturas que van a utilizar el sistema informático.
Visión General	El administrador podrá crear, buscar, eliminar o editar la información de las asignaturas que van hacer uso del sistema.
Tipo	Primario y Esencial
Referencias	Requisito 7
Curso Típico de Eventos	

**Secuencia Normal
(acción del actor)**

1. Este caso empieza cuando el administrador selecciona menú asignaturas, y escoge una de las siguientes opciones:
 - 1.1. Registrar asignaturas
 - 1.2. Actualizar asignaturas
 - 1.3. Eliminar asignaturas
 - 1.4. Listar asignaturas
2. En caso de escoger registrar asignaturas, el administrador deberá ingresar los datos solicitados en el formulario de registro, luego deberá dar clic en confirmar o cancelar acción, en caso de elegir confirmar el software validará si los datos están correctamente ingresados, si lo están se procederá a guardar el registro en la base de datos.
3. En caso de elegir actualizar asignaturas, el sistema le pedirá ingresar el código de asignatura para realizar la búsqueda del mismo y luego seguirá los procedimientos vistos en el punto 2.
4. En caso de elegir eliminar asignatura, el sistema le pedirá ingresar el código de asignatura para realizar la búsqueda del mismo, una vez encontrado la asignatura le preguntará si desea o no eliminar la asignatura.
5. En el caso de elegir listar asignaturas, se presentará una lista de todas las asignaturas registradas en el sistema para esto el sistema previamente deberá hacer una consulta en la base de datos.

Secuencia(s) Alternativas(s)	<p>En caso de:</p> <p>Ocurrir algún error en la validación de los datos ingresados, se le notificará al administrador de los campos erróneos para que proceda a corregir.</p> <p>Ocurrir algún error en la búsqueda de las asignaturas, se le notificará al administrador que dichas asignaturas no se encuentra registrado en el sistema.</p>
Precondiciones	El administrador debió haber iniciado sesión en el sistema.
Post condiciones	Asignaturas registradas, modificadas o eliminadas.
Comentarios	Ninguno

Fuente: Elaboración propia

Tabla 25: Caso de Uso Expandido: Gestión de semestres

Caso de Uso: Gestión de semestres	
Descripción	Permite gestionar los semestres de clases.
Actores	Administrador (Iniciador), Base de datos.
Propósito	Permitir que el sistema gestione los semestres que van a utilizar el sistema informático.
Visión General	El administrador podrá crear, buscar, eliminar o editar la información de los semestres que van hacer uso del sistema.
Tipo	Primario y Esencial
Referencias	Requisito 8
Curso Típico de Eventos	

**Secuencia Normal
(acción del actor)**

1. Este caso empieza cuando el administrador selecciona menú semestres, y escoge una de las siguientes opciones:
 - 1.1. Registrar semestres
 - 1.2. Actualizar semestres
 - 1.3. Eliminar semestres
 - 1.4. Listar semestres
2. En caso de escoger registrar semestres, el administrador deberá ingresar los datos solicitados en el formulario de registro, luego deberá dar clic en confirmar o cancelar acción, en caso de elegir confirmar el software validará si los datos están correctamente ingresados, si lo están se procederá a guardar el registro en la base de datos.
3. En caso de elegir actualizar semestres, el sistema le pedirá ingresar el código de semestre para realizar la búsqueda del mismo y luego seguirá los procedimientos vistos en el punto 2.
4. En caso de elegir eliminar semestres, el sistema le pedirá ingresar el código de semestre para realizar la búsqueda del mismo, una vez encontrado el semestre le preguntará si desea o no eliminar el semestre.
5. En el caso de elegir listar semestres, se presentará una lista de todos los semestres registrados en el sistema para esto el sistema previamente deberá hacer una consulta en la base de datos.

Secuencia(s) Alternativas(s)	<p>En caso de:</p> <p>Ocurrir algún error en la validación de los datos ingresados, se le notificará al administrador de los campos erróneos para que proceda a corregir.</p> <p>Ocurrir algún error en la búsqueda del semestre, se le notificará al administrador que dicho semestre no se encuentra registrado en el sistema.</p>
Precondiciones	El administrador debió haber iniciado sesión en el sistema.
Post condiciones	Semestres registrados, modificados o eliminados.
Comentarios	Ninguno

Fuente: Elaboración propia

Tabla 26: Caso de Uso Expandido: Reportes

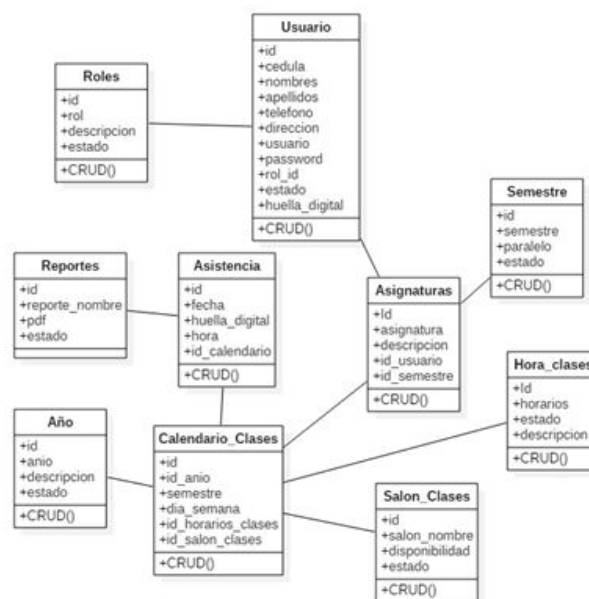
Caso de Uso: Reportes	
Descripción	Permite realizar reportes de asistencias
Actores	Administrador (Iniciador), Base de datos.
Propósito	Permitir que el administrador realice reportes.
Visión General	El administrador podrá crear reportes de asistencia laboral.
Tipo	Primario y Esencial
Referencias	Requisito 9
Curso Típico de Eventos	

Secuencia Normal (acción del actor)	<ol style="list-style-type: none"> 1. Este caso empieza cuando el administrador ingresa al menú de reportes, después deberá elegir el tipo de reportes, es decir, mensual o semanal. 2. Luego se obtendrá los datos necesarios de la base de datos y se procederá a generar el reporte.
Secuencia(s) Alternativas(s)	<p>En caso de:</p> <p>Ocurrir algún error en la consulta con la base de datos de buscar los registro de asistencias, se le notificará al administrador de dicho error.</p>
Precondiciones	El administrador debió haber iniciado sesión.
Post condiciones	Reporte generado.
Comentarios	Ninguno

Fuente: Elaboración propia

Anexo F. Diagrama de clases

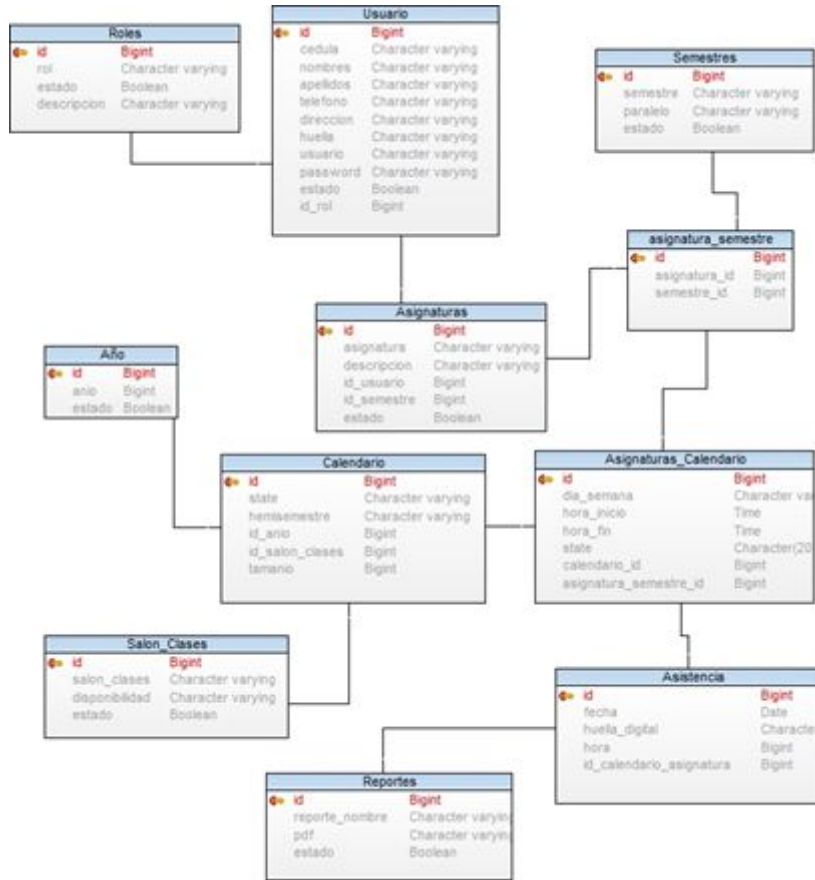
Ilustración 15: Diagrama de clases



Fuente: Elaboración propia

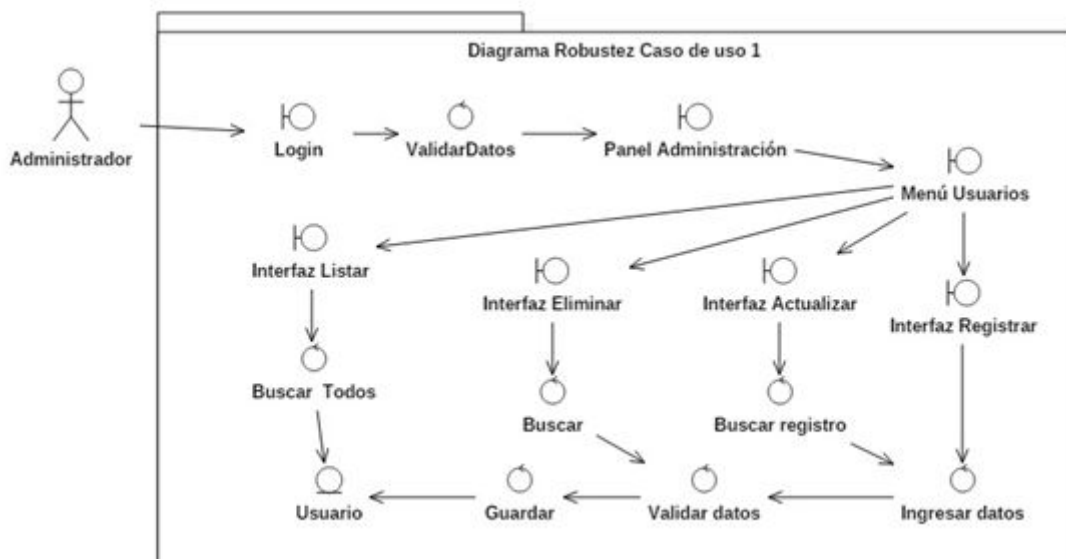
Anexo G. Diagrama relacional

Ilustración 16: Diagrama relacional base de datos



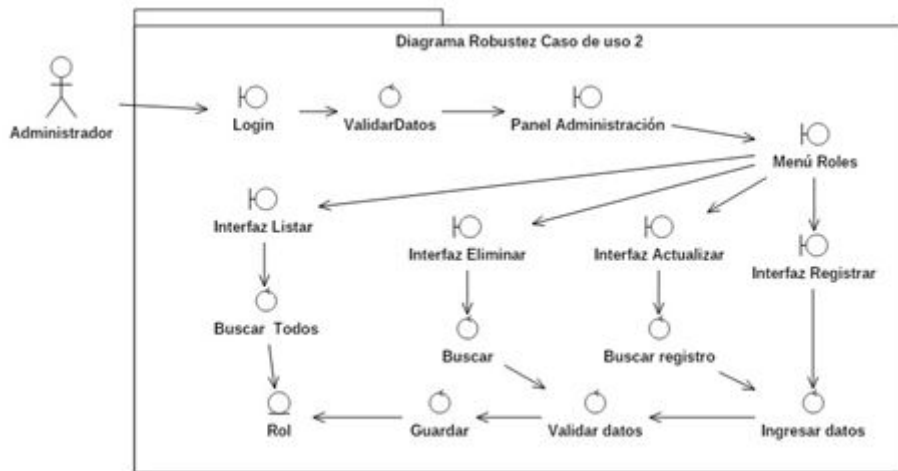
Anexo H. Diagrama de robustez

Ilustración 17: Diagrama de robustez - Caso de uso 01



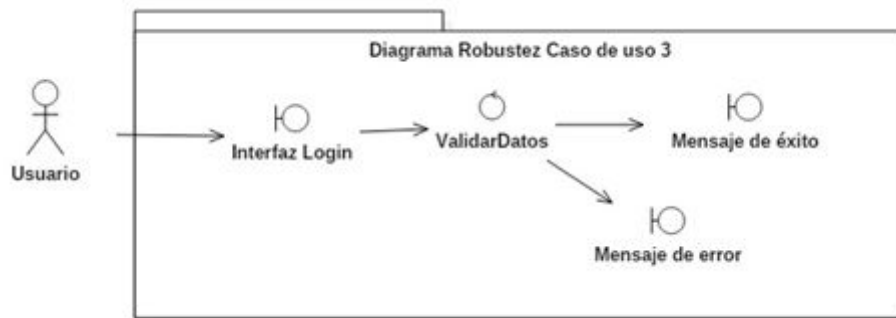
Fuente: Elaboración propia

Ilustración 18: Diagrama de robustez - Caso de uso 02



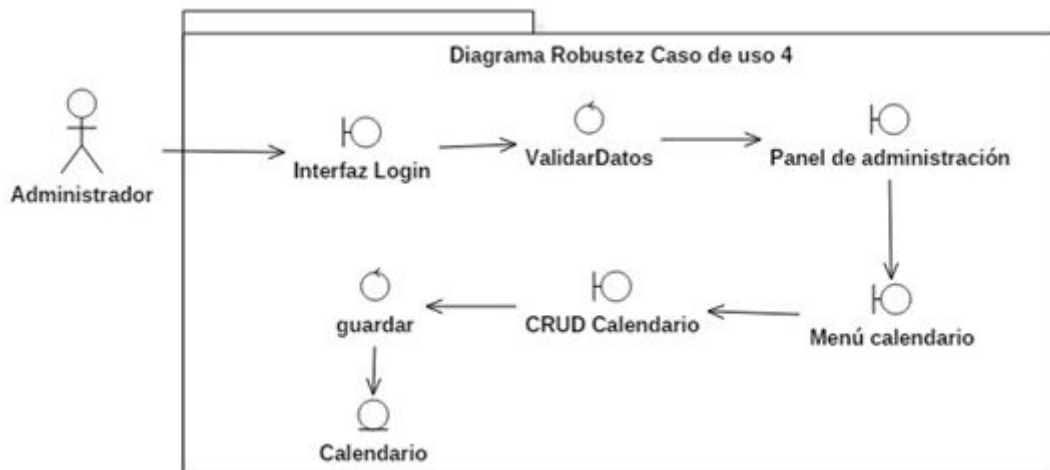
Fuente: Elaboración propia

Ilustración 19: Diagrama de robustez - Caso de uso 03



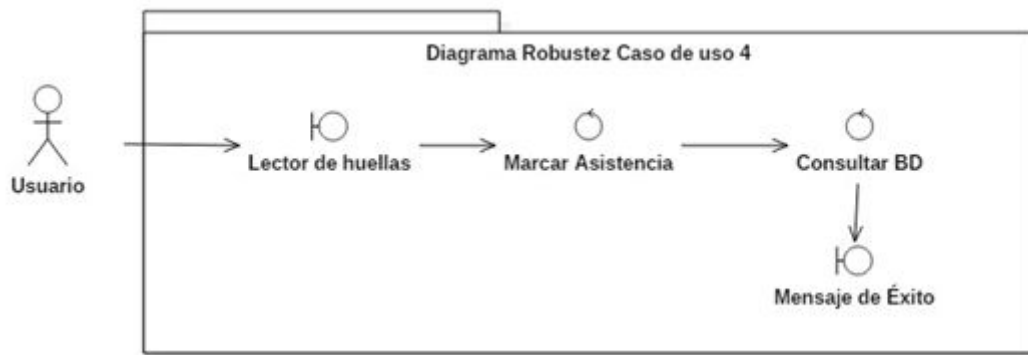
Fuente: Elaboración propia

Ilustración 20: Diagrama de robustez - Caso de uso 04



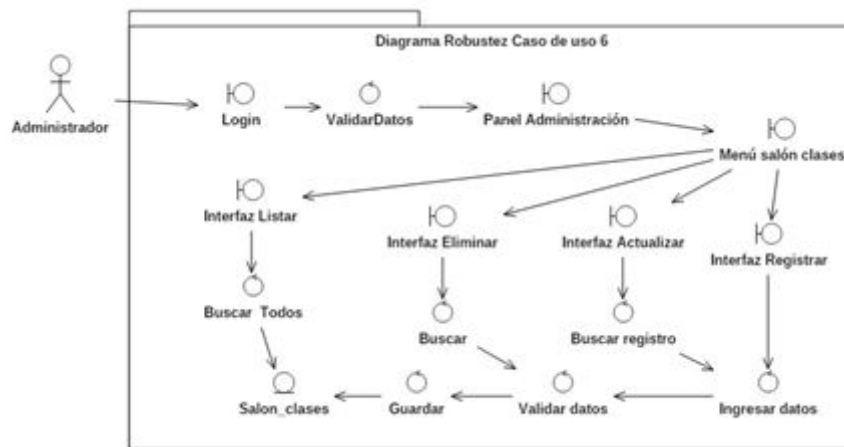
Fuente: Elaboración propia

Ilustración 21: Diagrama de robustez - Caso de uso 05



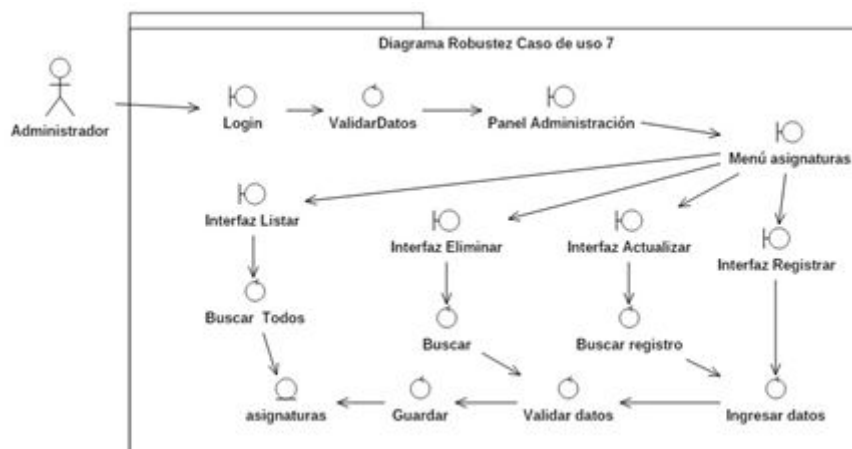
Fuente: Elaboración propia

Ilustración 22: Diagrama de robustez - Caso de uso 06



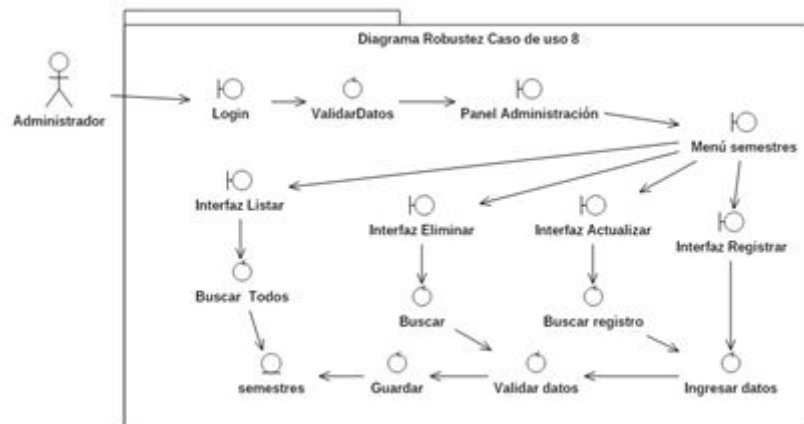
Fuente: Elaboración propia

Ilustración 23: Diagrama de robustez - Caso de uso 07



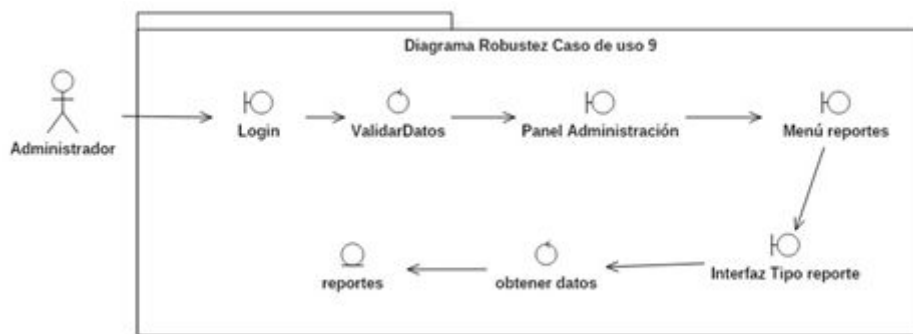
Fuente: Elaboración propia

Ilustración 24: Diagrama de robustez - Caso de uso 08



Fuente: Elaboración propia

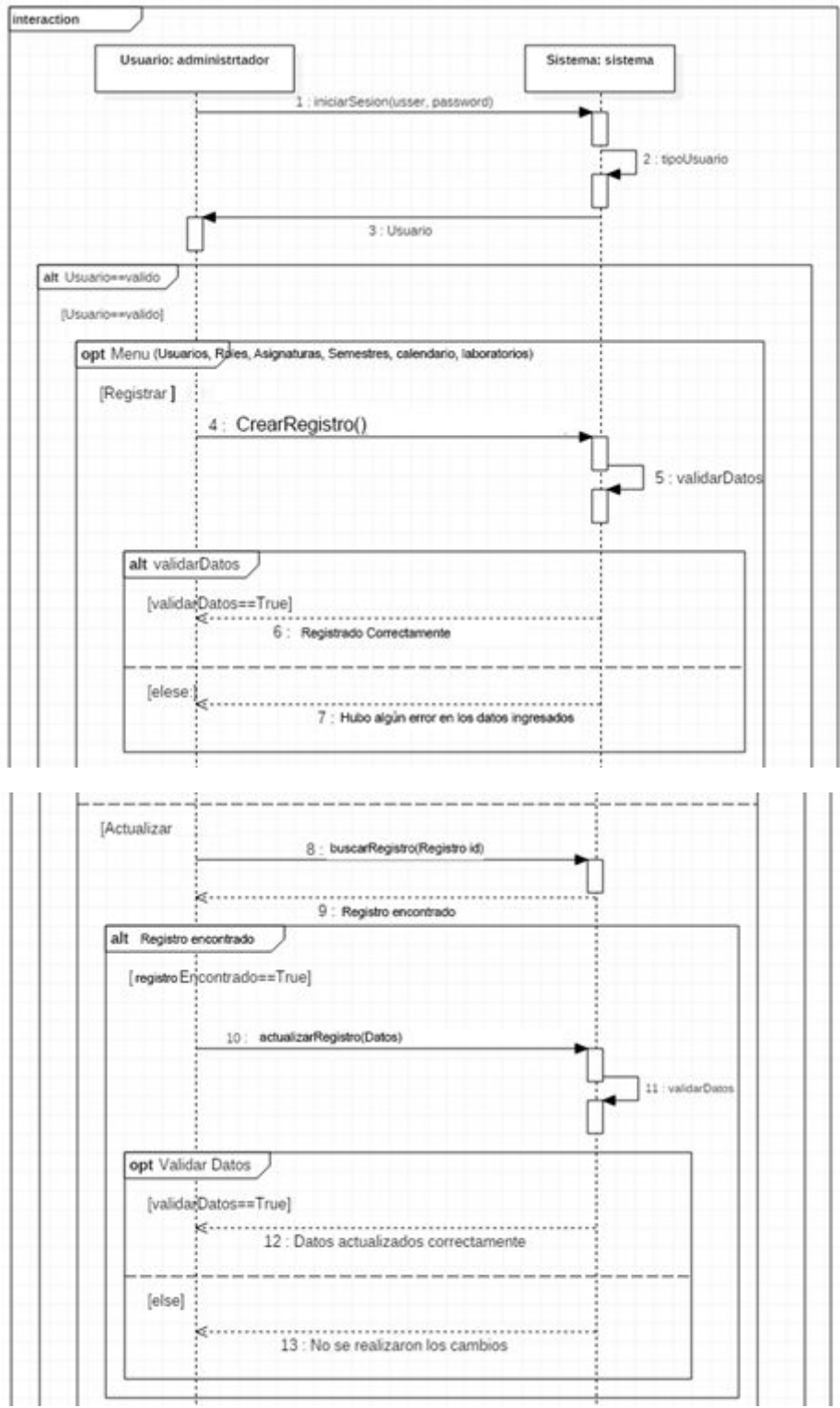
Ilustración 25: Diagrama de robustez - Caso de uso 09

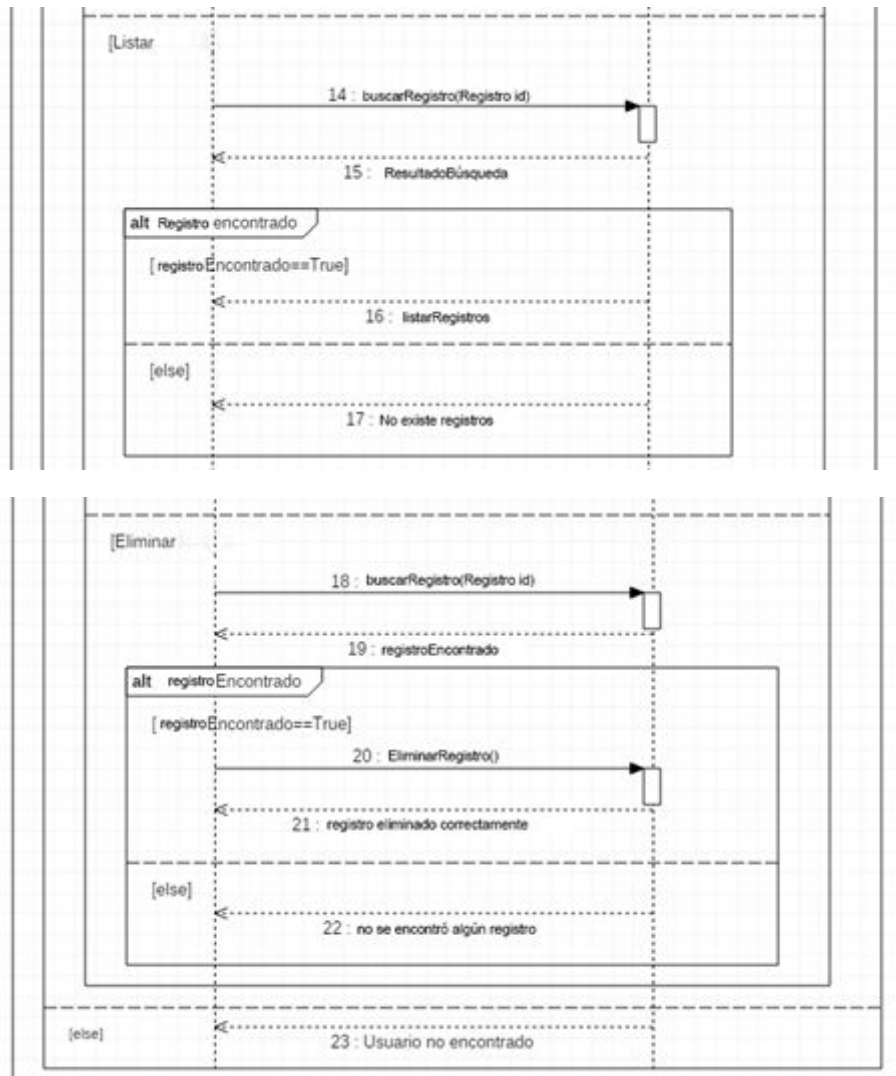


Fuente: Elaboración propia

Anexo I. Diagrama de secuencias

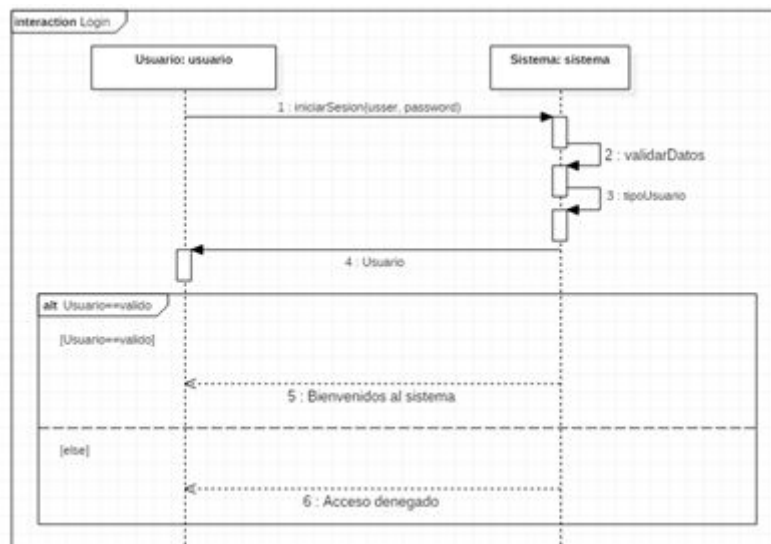
Ilustración 26: Diagrama de secuencia - Caso de usos 01-02-04-05-06-07-08





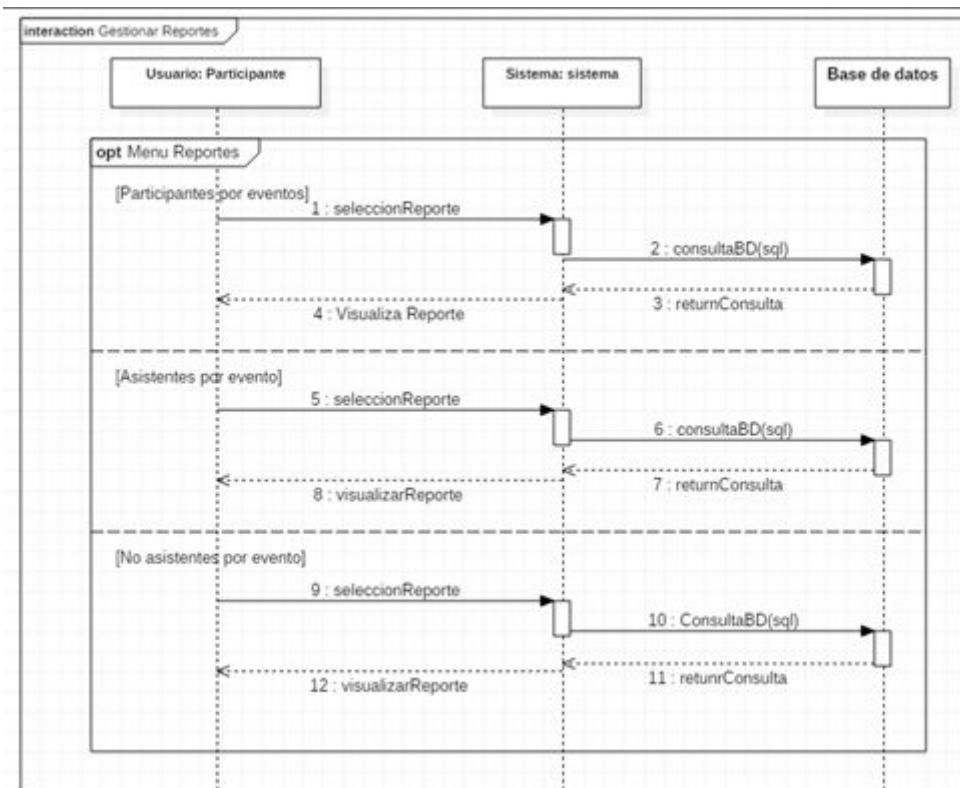
Fuente: Elaboración propia

Ilustración 27: Diagrama de secuencia - Caso de uso 03



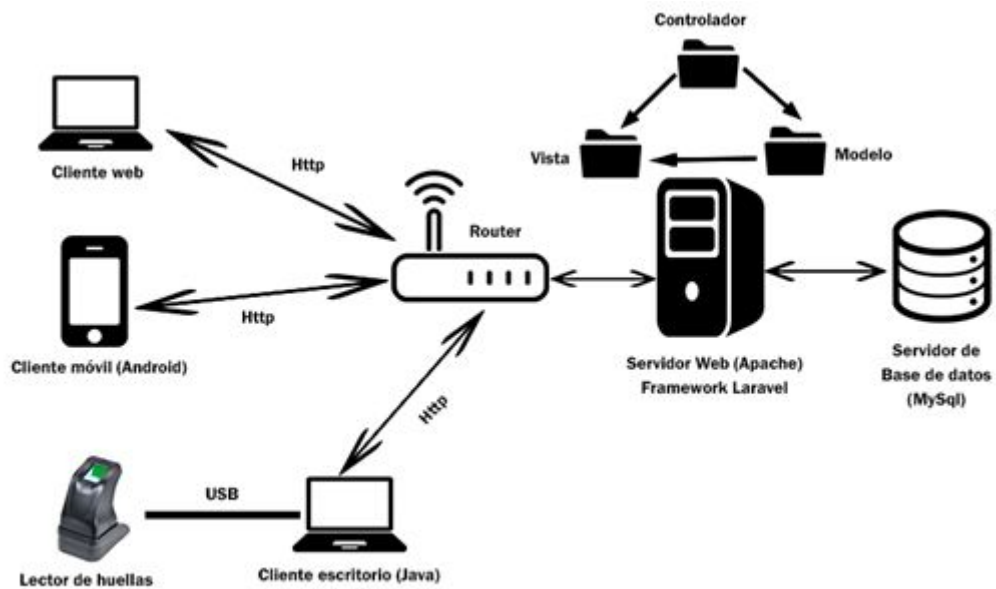
Fuente: Elaboración propia

Ilustración 28: Diagrama de secuencia - Caso de uso 09



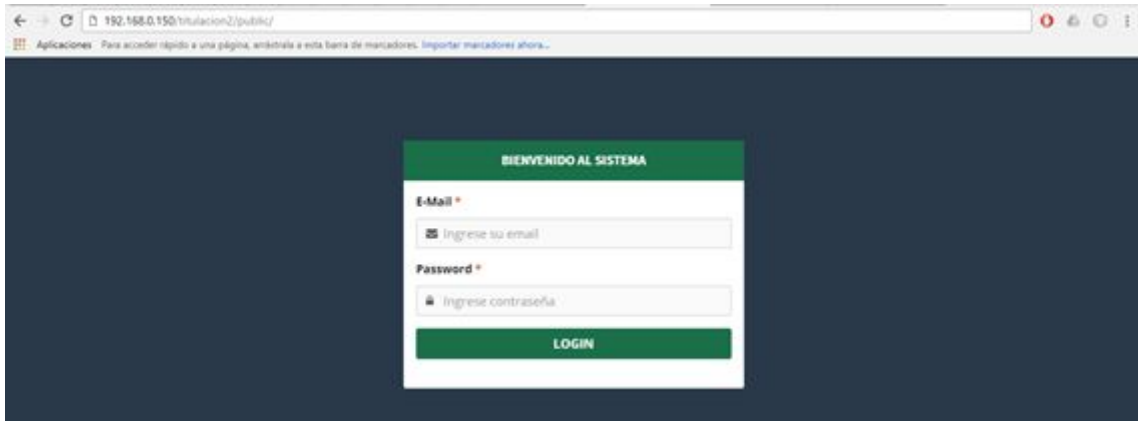
Anexo J. Diagrama arquitectónico

Ilustración 29: Diseño arquitectónico

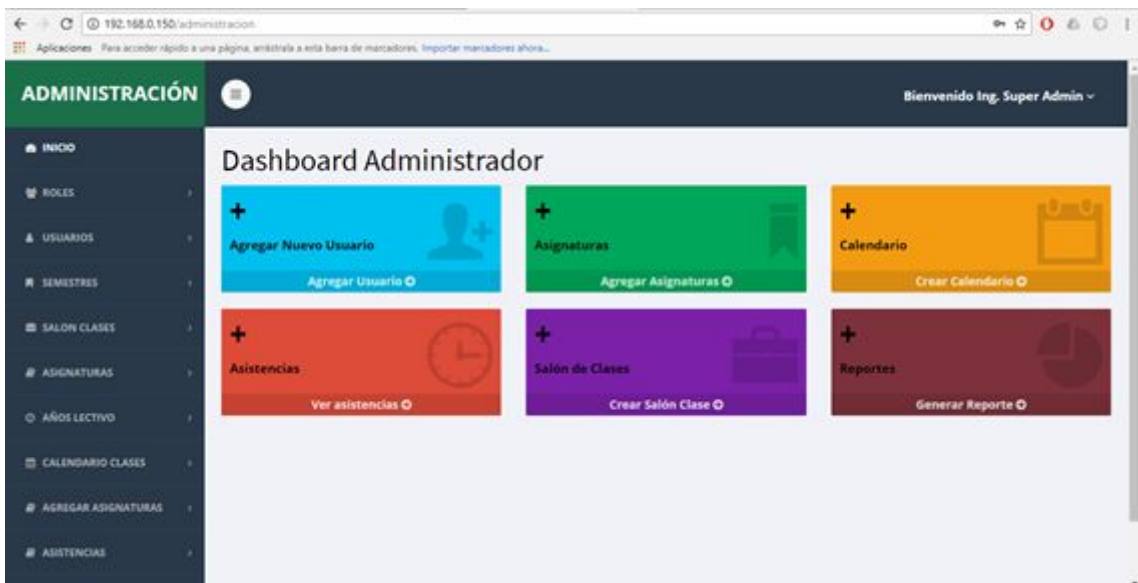


Fuente: Elaboración propia

Anexo K. Ingreso al sistema web
Ilustración 30: Ingreso al sistema web



Fuente: Elaboración propia
Anexo L. Dashboard Administrador
Ilustración 31: Dashboard Administrador



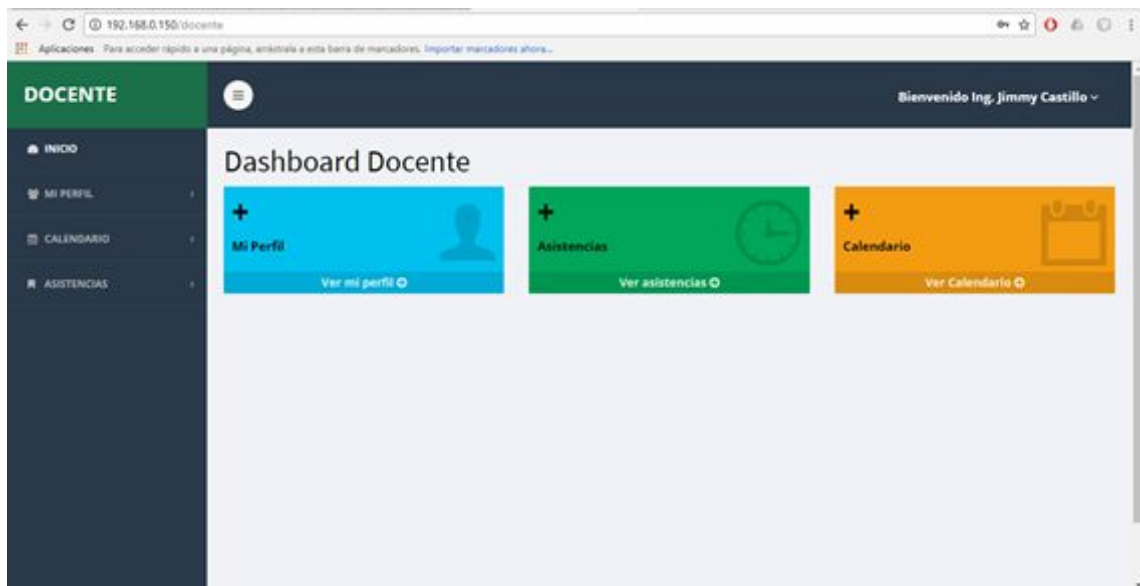
Fuente: Elaboración propia
Anexo M. Menú dashboard administrador
Ilustración 32: Menú dashboard administrador



Fuente: Elaboración propia

Anexo N. Dashboard docente

Ilustración 33: Dashboard docente



Fuente: Elaboración propia

Anexo O. Menú dashboard docente

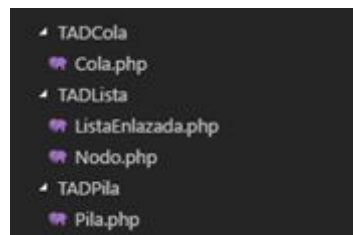
Ilustración 34: Menú dashboard docente



Fuente: Elaboración propia

Anexo P. Estructura de datos (TAD)

Ilustración 35: Estructura de datos (TAD)



Fuente: Elaboración propia

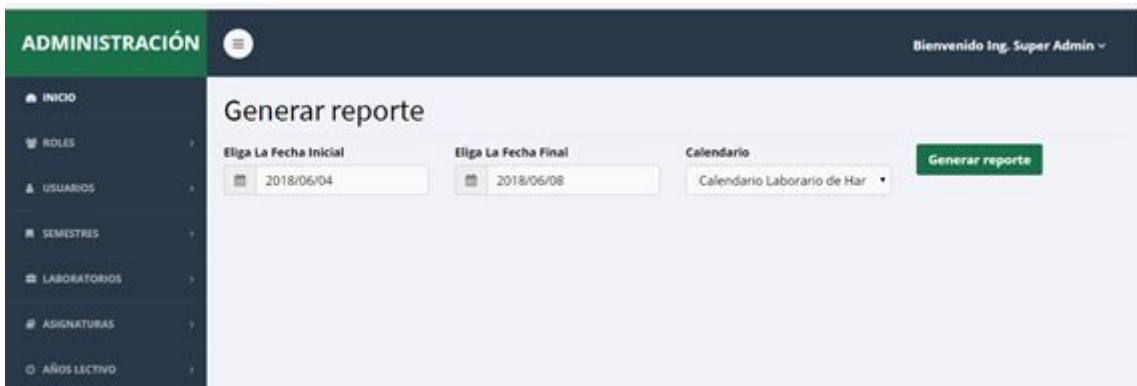
Anexo Q. Creación de calendarios

Ilustración 36: Creación de calendarios



Fuente: Elaboración propia

Anexo R. Generación de reportes
Ilustración 37: Generación de reportes



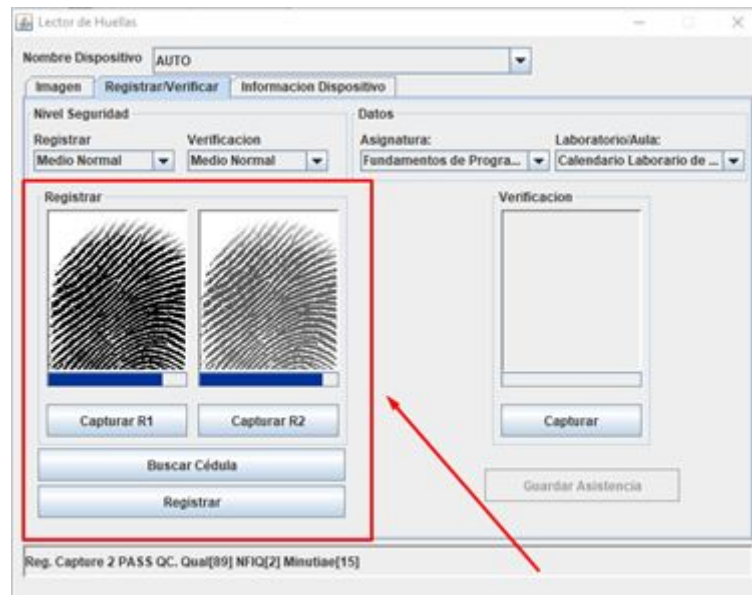
Fuente: Elaboración propia

Anexo S. Aplicación escritorio – lector huellas
Ilustración 38: Aplicación escritorio – lector huellas



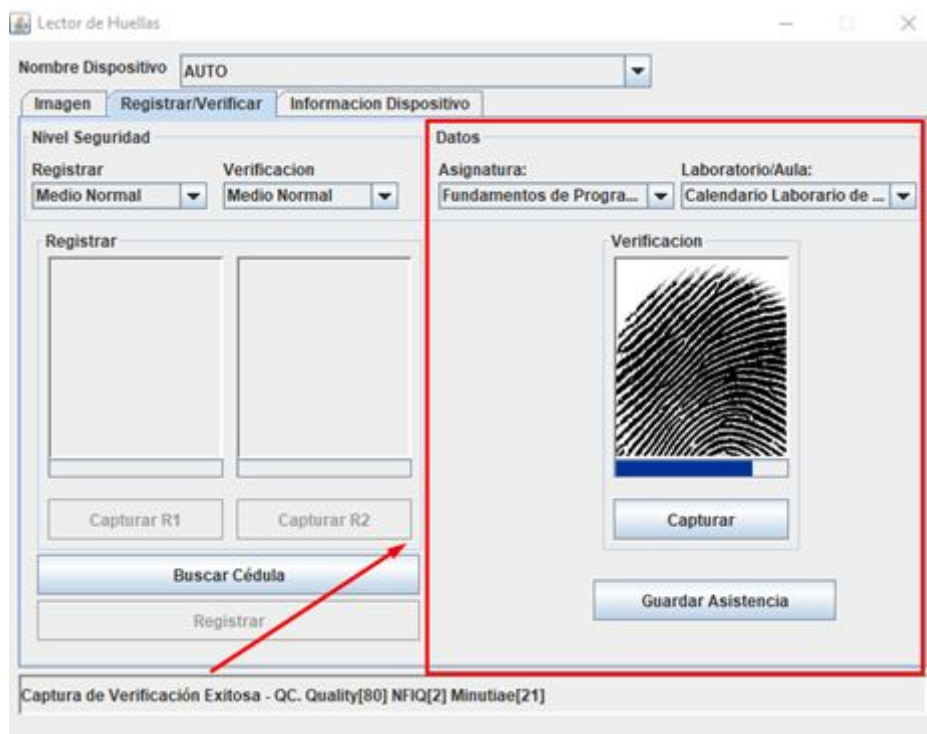
Fuente: Elaboración propia

Anexo T. Registro de huella digital
Ilustración 39: Registro de huella digital



Fuente: Elaboración propia

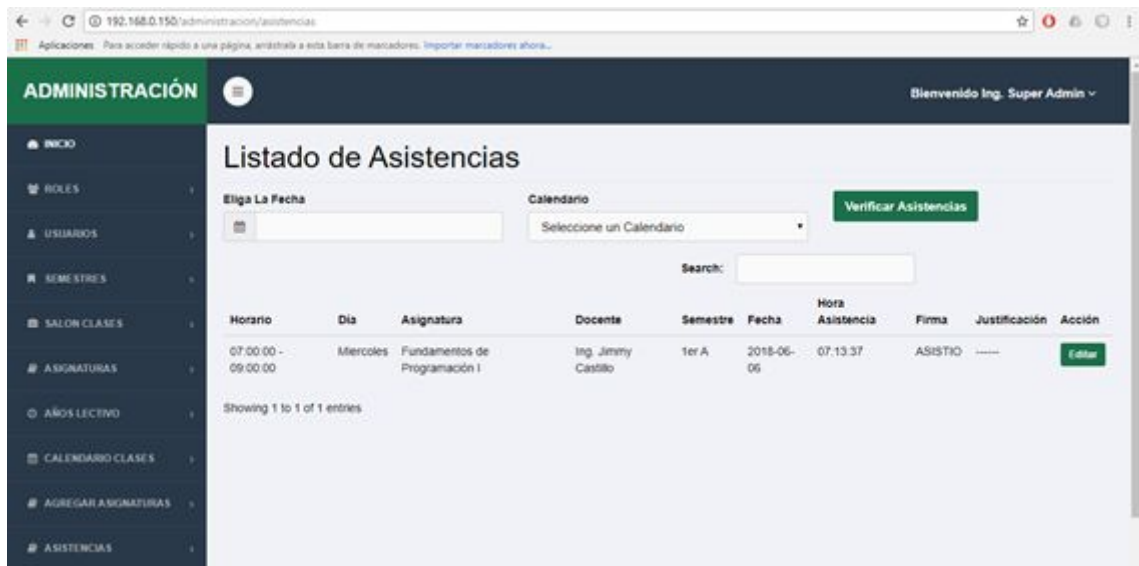
Anexo U. Registro de asistencia
 Ilustración 40: Registro de asistencia



Fuente: Elaboración propia

Anexo V. Visualización de asistencias

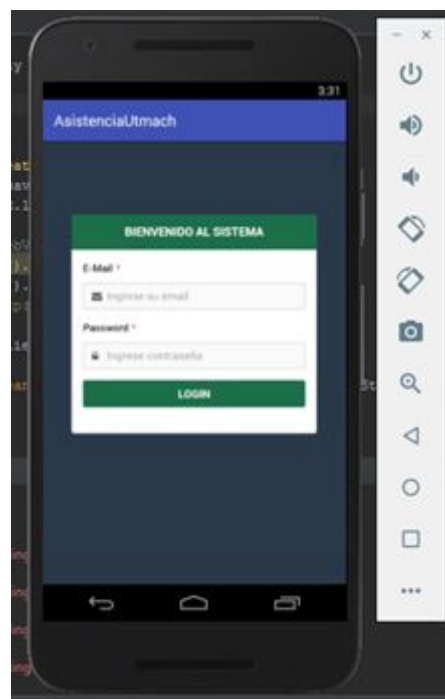
Ilustración 41: Visualización de asistencias



Fuente: Elaboración propia

Anexo W. Login Android

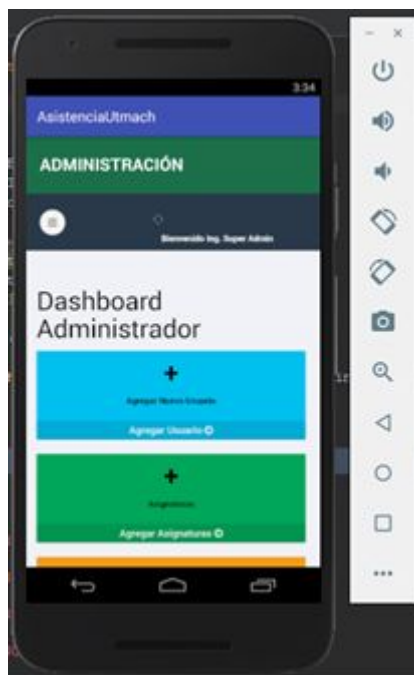
Ilustración 42: Login en android



Fuente: Elaboración propia

Anexo X. Dashboard administrador en android

Ilustración 43: Dashboard administrador en android



Fuente: Elaboración propia