UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO

CARRERA:
INGENIERÍA DE SISTEMAS

Trabajo de titulación previo a la obtención del título de:
Ingeniera e Ingeniero de Sistemas

TEMA:
AUTOMATIZACIÓN DEL PROCESO DE RESERVACIÓN DE LABORATORIOS, PRÉSTAMOS Y MANTENIMIENTO DE EQUIPOS TECNOLÓGICOS DEL CECASIS A TRAVÉS DE UNA APLICACIÓN WEB

AUTORES:
JESENIA NATALY MAIGUA JIMÉNEZ
HENRY DAVID QUILUMBA TOAPANTA

TUTOR:
GUSTAVO ERNESTO NAVAS RUILOVA

Quito, agosto del 2018

CESIÓN DE DERECHOS DE AUTOR

Nosotros, JESENIA NATALY MAIGUA JIMÉNEZ, con documento de identificación N° 1721137584 y, HENRY DAVID QUILUMBA TOAPANTA, con documento de identificación N° 1720758224, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación con el tema: AUTOMATIZACIÓN DEL PROCESO DE RESERVACIÓN DE LABORATORIOS, PRÉSTAMOS Y MANTENIMIENTO DE EQUIPOS TECNOLÓGICOS DEL CECASIS A TRAVÉS DE UNA APLICACIÓN WEB, mismo que ha sido desarrollado para optar por el título de INGENIERA E INGENIERO DE SISTEMAS en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada.

En concordancia, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.

JESENIA NATALY
MAIGUA JIMÉNEZ
CI: 1721137584

HENRY DAVID
QUILUMBA TOAPANTA
CI: 1720758224

Quito, agosto del 2018

DECLARATORIA DE COAUTORÍA DEL TUTOR

Yo declaro que bajo mi dirección y asesoría fue desarrollado el proyecto técnico, con el tema: AUTOMATIZACIÓN DEL PROCESO DE RESERVACIÓN DE LABORATORIOS, PRÉSTAMOS Y MANTENIMIENTO DE EQUIPOS TECNOLÓGICOS DEL CECASIS A TRAVÉS DE UNA APLICACIÓN WEB, realizado por JESENIA NATALY MAIGUA JIMÉNEZ y HENRY DAVID QUILUMBA TOAPANTA, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Quito, agosto del 2018

GUSTAVO ERNESTO NAVAS RUILOVA
CI: 1705675625

DEDICATORIA

Dedico este trabajo en primer lugar a Dios por darme salud y fortaleza para lograr mis objetivos.

A mis padres Carlos y Jacqueline por ser el pilar fundamental en todo lo que soy, por su paciencia, sus consejos, sus valores, por creer en mí, por la motivación y apoyo que me han brindado a lo largo de toda mi vida y por su amor incondicional.

A mis hermanos Juan Carlos y Dilan por el cariño que me tiene y por estar a mi lado en todo momento con su apoyo y palabras de ánimo.

Y a todas las personas que me apoyaron y creyeron en mí.

JESENIA NATALY MAIGUA JIMÉNEZ

Este logro se lo dedico a toda mi familia porque han sabido ayudarme en todas las etapas de la vida, con sus consejos, principios y en general sus enseñanzas que me han permitido ir creciendo tanto en mi vida personal y profesional.

HENRY DAVID QUILUMBA TOAPANTA

AGRADECIMIENTO

Agradecemos a la Universidad Politécnica Salesiana que ha contribuidito en nuestra formación tanto profesional como humana, a los maestros que marcaron cada etapa de nuestro camino universitario, y que nos ayudaron con asesorías y dudas presentadas en la elaboración de nuestra tesis.

JESENIA NATALY MAIGUA JIMÉNEZ

HENRY DAVID QUILUMBA TOAPANTA

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

Resumen

El proyecto que se presenta a continuación permite plasmar los conocimientos adquiridos durante la permanencia en la carrera. El objetivo del proyecto es desarrollar una aplicación web que ayude a mejorar de forma notable los procesos de reservación de laboratorios, alquiler de máquinas, préstamo de elementos, reporte de incidencias y manejo de inventario, en el Centro de Capacitación y Servicios Informáticos (CECASIS) del Campus Sur de la Universidad Politécnica Salesiana.

El diseño de la aplicación fue realizado en base a la información y requerimientos proporcionados por el personal del CECASIS. Fue necesario realizar reuniones de forma periódica al inicio del proyecto hasta llegar al entendimiento total del negocio.

En la etapa de desarrollo y con la ayuda de la información obtenida se buscó automatizar los procesos sin alterar la lógica y el orden con los que se venían trabajando.

Después de varias presentaciones y correcciones el producto final es una aplicación web que cumple con los requerimientos inicialmente establecidos.

Abstract

The following project allow us to inquire all the knowledge learned during the permanence of the carrier. The main objective is to develop a web application that can help improve notably the processes in the reservations of laboratories, machines renting, loan of elements, incidents reports and the inventory management in the Training Center and Informatics Services (CECASIS) from the south campus of the Universidad Politécnica Salesiana.

The application designed was made according to the information and requirements given by the CECASIS staff. It was necessary to arrange meetings periodically at the beginning of this project until we reach the full understanding of the business logic.

During the development phase and with the help of the information given, we sought to automate the processes without altering the logic and order with which they had been working.

After a few presentations and corrections, the final product is a web application that follows the requirements stablished.

INTRODUCCIÓN

Antecedentes

El CECASIS del Campus Sur actualmente no tiene a su disposición algún tipo de software que, de soporte a los procesos de reservación de laboratorios, préstamo y mantenimiento de equipos tecnológicos, los registros se encuentran en documentos físicos y digitales ya sean hojas de Word o Excel.

La información se encuentra dispersa y al necesitar realizar una consulta se debe buscar por los diferentes archivos, generando un consumo de tiempo en ocasiones y obtener la información deseada pueden existir registros duplicados.

Su inventario de partes y piezas también se encuentran en archivos dispersos corriendo el riesgo de pérdida, además cuando se necesita reportes se toma un tiempo considerable hasta realizar un solo consolidado.

Problema

Los servicios prestados por el CECASIS generan una gran cantidad de información que es necesaria registrar y posteriormente procesarla lo cual consume el tiempo de la persona asignada. Por este motivo para realizar una consulta se requiere buscar por los archivos físicos y manuales donde puede llegar a detectarse registros duplicados.

Un problema latente es la posibilidad de pérdida de la información de forma total o parcial al estar de forma dispersa. Al tener algún inconveniente en algunos de los servicios que brinda el CECASIS, no se podrá tener evidencia de la persona que solicito el servicio.

Por cada periodo académico el CECASIS registra un gran volumen de información que tiene dificultad en la administración, de forma temporal la manera de registro actual solventa las necesidades, pero en tema de tiempo de respuestas en consultas a nivel de reporte hay un consume considerable.

Para los usuarios también se genera un consumo de tiempo debido a que necesitan acceder a un servicio y al llegar al CECASIS pueden tener la noticia que no hay disponibilidad de laboratorios, equipos. Permitirle visualizar información ayudaría a las partes involucradas a mejorar en la calidad de servicio.

Justificación

Con los problemas identificados se propone el desarrollo de una aplicación web, que automatice el proceso de reservación de laboratorios, préstamo de elementos tecnológicos y ayude a gestionar la información del proceso de mantenimiento de equipos.

Mediante esta aplicación se entregará información oportuna, concreta y precisa de los servicios que brinda el CECASIS y que además aporta al siguiente objetivo estratégico: "Las dependencias universitarias tienen acceso a información relevante, consistente, congruente y oportuna a través de las tecnologías de la información y comunicación."

Con la implementación de un almacén de datos digital, se busca promover la disminución del consumo de papel y ayudará a asegurar la información, manteniendo la integridad de datos y mejorando los tiempos de respuesta invertidos en las diferentes tareas de cada proceso.

El docente, estudiante podrá tener la información sobre cada servicio actualizada y así planificar las actividades académicas de una mejor manera.

Objetivo General

Automatizar el proceso de reservación de laboratorios, préstamos y mantenimiento de equipos de cómputo del CECASIS, a través del desarrollo de una aplicación web.

Objetivos Específico

Mejorar la gestión de la información de los servicios prestados por el CECASIS disminuyendo los tiempos que se utilizan en realizar las diferentes actividades.

Permitir a estudiantes y docentes manejar información actualizada sobre la disponibilidad de laboratorios y equipos tecnológicos para poder realizar una reservación, sin tener que acudir a la oficina del CECASIS.

Entregar una aplicación modular utilizando el patrón Modelo-Vista-Controlador (MVC), para que la codificación sea realizada por capas y que las futuras versiones sean más fáciles de realizar.

Alcance del proyecto

Para definir el alcance del proyecto se hizo uso de la Estructura de Desglose del Producto o Product Breakdown Structure (PBS), donde se realiza una descripción macro del proyecto.

PBS del proyecto

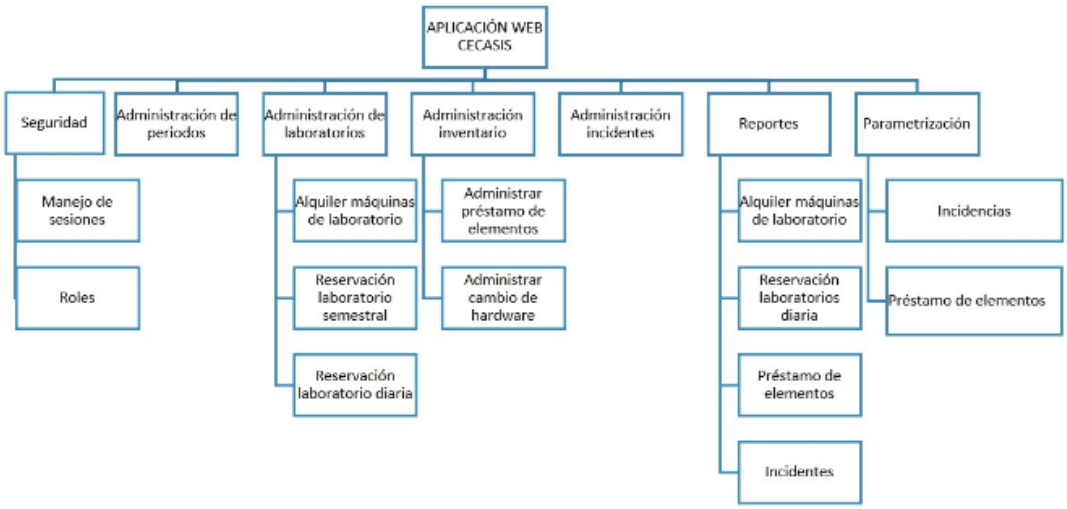


Figura 1. PBS que modela los componentes del proyecto
Elaborado por: Maigua Jesenia & Quilumba Henry (2018).

Seguridad

Manejo de sesiones. Iniciar y cerrar sesión con uso de credenciales en este caso correo electrónico institucional y contraseña.

Roles. La seguridad será manejada mediante roles definidos para los usuarios: estudiante, docente y personal de CECASIS, y acorde a su rol podrá acceder a las funcionalidades permitidas dentro de la aplicación.

Administración de períodos

Para mantener información actual de estudiantes y docentes, los usuarios de CECASIS podrán actualizar esta información al activar un periodo en el sistema, previamente el periodo debe ser ingresado y la nueva información debe ser recopilada en archivos csv para su lectura y almacenamiento automático.

Administración laboratorios

Alquiler máquinas de laboratorio. El personal de CECASIS podrá registrar el alquiler de máquinas de laboratorio que tengan habilitada esta funcionalidad, ingresando la cedula del estudiante o docente para registrar el uso de las máquinas y podrá finalizar el alquiler una vez concluido es uso de esta.

Reservación laboratorios semestral. El personal de CECASIS podrá realizar la reservación de laboratorios para un uso continuo en el día y hora indicados, la finalización de reserva será automática o manual.

Reservación laboratorios diario. El personal de CECASIS y el usuario docente podrá realizar la reservación de laboratorios hasta 8 días desde la fecha actual, el usuario técnico registrara su reservación como mantenimiento, las reservaciones serán finalizadas automáticamente una vez concluido el horario escogido.

Administración inventario

Administrar préstamo de elementos. Usuarios del CECASIS podrán registrar el préstamo de los elementos disponibles en el centro, registrando la cedula del estudiante o docente, la finalización del préstamo se realizará una vez el elemento sea recibido, el plazo establecido por defecto es de un día o el usuario podrá registrar la fecha deseada, el cambio de estado si el equipo no es devuelto será de forma automática.

Administrar cambio de hardware. El usuario de CECASIS podrá administrar el cambio de hardware de las máquinas de los laboratorios mediante el ingreso de solicitudes, las mismas de deberán ser aprobadas por el/la coordinador/a del centro.

Administración de incidentes

Los usuarios podrán registrar incidentes en laboratorios, que serán asignados a técnicos de forma automática o manual para su respectiva solución.

Reportes

Los usuarios de CECASIS podrán visualizar los reportes que se mencionan a continuación ingresando una fecha inicio y fecha fin, o seleccionando un periodo.

- Alquiler máquinas de laboratorio
- Reservación laboratorios diaria
- Préstamo de elementos
- Incidentes

Parametrización

Los usuarios de CECASIS podrán parametrizar valores en los siguientes casos:

- Número máximo de incidencias ingresadas por un mismo usuario y que aún no han sido solucionadas.

- Porcentaje máximo según el número de máquinas para el ingreso de incidencias en un mismo laboratorio y que aún no han sido solucionadas.
- Registrar el valor de la multa para el préstamo de elementos que no han sido devueltos dentro de la fecha indicada.

CAPÍTULO 1

1 MARCO TEÓRICO

1.1 Programación orientada a objetos

La programación Orientada a objetos nos permite tener un enfoque más cercano al pensamiento del ser humano, además nos permite tener un equilibrio éntrelos procesos y los datos. (Bernal Bermúdez & Fernández Muñoz , 2012, pág. 7)

1.2 Java Server Faces (JSF)

La tecnología JSF constituye un marco de trabajo (Framework) de interfaces de usuario del lado de servidor para aplicaciones web basadas en tecnología Java y en el patrón MVC (Modelo Vista Controlador). (SGI-Consulting, 2018, pág. 4)

Al utilizar JSF en un proyecto de software, se busca mantener un estándar, normalizar el desarrollo de la aplicación web. Es un conjunto de API's permitiéndonos hacer uso de sus componentes de una manera más fácil para el desarrollador. La arquitectura como se encuentra desarrollado JSF ayuda a separar toda la lógica de la aplicación y de la interfaz de usuario. Cada miembro del equipo de desarrollo puede enfocarse en sus actividades y cada cierto tiempo se puede integrar todo el sistema sin tener inconvenientes.

1.3 Primefaces

Es una librería que incorpora un conjunto enriquecido de componentes JSF, permitiendo una mayor facilidad al momento de desarrollar una aplicación web. Los componentes de Primefaces facilitan el desarrollo de interfaces de usuario, así como también brindan una mejor apariencia a nivel de programación de las diferentes funcionalidades.

1.3.1 Característica de Primefaces

A continuación, se mencionan varias características que Primefaces ofrece al desarrollador:

- Contiene varios componentes como: autocompletado, entrada de texto, cuadros de diálogos, mensajes, tablas, etc.
- La librería no necesita ninguna dependencia adicional.
- Por defecto trae consigo varios temas para la interfaz del usuario, siendo muy simple su uso.
- De la página oficial de Primefaces se puede descargar la documentación para el uso de todos los componentes, donde explica con ejemplos su funcionamiento.

1.4 Base de datos relacional

Una base de datos relacional consiste en una colección de datos interrelacionados y un conjunto de aplicaciones que nos permitan el acceso a dichos datos. Las bases de datos

relacionales nos permiten el almacenamiento de grandes cantidades de información relevante para nuestra empresa. (Silberschatz, Korth, & Sudarshan, 2002, pág. 1)

Entendiendo los requerimientos del cliente, el paso a seguir es generar un diagrama entidad–relación, representando todas las entidades identificadas, sus atributos e interconexiones entre las mismas. Se requiere tener en cuenta los principios que garanticen la facilidad de consulta de datos, integridad y consistencia de la información.

1.5 Structured Query Lenguaje (SQL)

Es un lenguaje que se soporta en el modelo relacional. Contiene estándares que permiten incluir construcciones orientadas a objetos. (Sheldon & Oppel, 2010, pág. 17)

Permite especificar diferentes tipos de sentencias las cuales pueden estas compuestas de comandos, operadores, clausulas. Los elementos mencionados pueden combinarse para crear, eliminar, actualizar y recuperar la información de una base de datos.

En el lenguaje SQL se pueden utilizar dos tipos de comandos, como se describen a continuación:

1.5.1 Lenguaje de definición de datos (DDL)

El DDL es parte del lenguaje SQL, sirve para definir datos y sus respectivas relaciones una base de datos. Se puede generar un script DDL que contengas los diferentes objetos de una base de datos.

1.5.2 Lenguaje de manipulación de datos (DML)

Este lenguaje es provisto por el sistema gestor de base de datos (SGDB), para poder consultar, actualizar, eliminar y crear nuevos registrar en el modelo de datos. Acorde al SGDB elegido las sentencias pueden variar.

1.6 PostgreSQL

PostgreSQL es un gestor de bases de datos orientadas a objetos (SGBDOO) muy usado, ofrece un conjunto de funcionalidades avanzadas que soporta lenguaje SQL y lo sitúa al mismo o a un mejor nivel que muchos SGBD comerciales. (Perez Mora & Gibert Ginestà, 2009, pág. 5)

Puede ser ejecuta tanto en sistemas operativos Linux como en Windows, además soporta el trabajo cliente–servidor, atendiendo las peticiones del usuario que necesita conectarse a la base de datos.

1.7 Arquitectura

El diseño arquitectónico es un proceso creativo en el que se intenta establecer una organización del sistema que satisfaga las necesidades de los requerimientos funcionales y no funcionales del propio sistema. (Pressman, 2014, págs. 217-218)

1.8 Patrón Modelo-Vista-Controlador

Sus características principales están dadas por el hecho de que, el Modelo, Vistas y los Controladores se tratan como entidades separadas; esto hace que cualquier cambio

producido en el Modelo se refleje automáticamente en cada una de las Vistas. (Fernández, 2012, pág. 48)

Utilizar un patrón de diseño para el desarrollo de una aplicación web permite un mejor mantenimiento para versiones futuras.

Explicación de Modelo-Vista-Controlador:

- **Modelo:** Datos y reglas de negocio
- **Vista:** Son las diferentes interfaces con las cuales el usuario interactuar con el sistema.
- **Controlador:** Gestiona las operaciones del usuario.

1.9 Diseño responsivo

Se describe como una técnica de diseño y desarrollo web que, mediante el uso de estructuras e imágenes fluidas, así como de media-queries en la hoja de estilo CSS, consigue adaptar el sitio web al entorno del usuario. (Labrada, 2013, pág. 4)

El contenido de la aplicación web se puede visualizar tanto en un computador de escritorio y en los diferentes dispositivos móviles. Está características se logra utilizando HTML y CSS. No es necesario desarrollar diferentes aplicaciones para cada versión de dispositivo en la que se utilizara.

1.10 Prototipado

Como en la mayoría de los proyectos de software los requerimientos son cambiantes, el prototipo permite establecer un bosquejo inicial de la solución a entregar, además ayuda

a establecer los requerimientos funcionales que el cliente necesita. Al final de cada entrega se puede ir mejorando funcionalidades de los nuevos módulos.

1.11 Metodología

1.11.1 Scrum

Scrum se basa en la teoría de control de procesos empírica o empirismo. El empirismo asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce. Scrum emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo. (Schwaber & Sutherland, 2013, pág. 4).

El trabajo que se realiza en equipo comprende iteraciones de corto tiempo con entregas parciales donde se priorizan los requerimientos de cliente para obtener resultados sobre el producto desea.

1.11.2 Sprint

Consiste en un tiempo máximo de 4 semanas para la construcción de un incremento del producto que para el cliente debe encontrarse totalmente operativo.

1.11.3 Product Backlog

Son un conjunto de requerimientos establecidos por el dueño del producto, quien es responsable de su priorización y contenido para agregar valor funcional al producto.

1.11.4 Sprint Planning

Es la reunión donde el dueño del producto presenta las funcionalidades de forma priorizada y se establece el orden para ser resueltas.

1.11.5 Daily Sprint Meeting

Son las reuniones diarias no mayores a 15 minutos donde el equipo de desarrollo expone las dificultades que se presentan durante el desarrollo del producto.

1.11.6 Scrum Master

En la metodología es la persona que cumple el rol de intermediario entre el product owner y el equipo de desarrollo, haciendo cumplir los acuerdos definidos.

1.11.7 Product Owner

Es la persona encargada de agregar el mejor valor al producto en cuanto a funcionamiento para los usuarios, cliente y todos los demás implicados.

1.11.8 Development Team

El equipo que posee el mayor conocimiento técnico para desarrollo del proyecto, dando solución a la priorización del product backlog.

CAPÍTULO 2

2 MARCO METODOLÓGICO

2.1 Utilización de SCRUM

Debido a que la naturaleza de los proyectos de software es altamente cambiante, existe una creciente tendencia del uso de metodologías ágiles.

En este proyecto se optó por utilizar la metodología SCRUM la misma que maneja un modelo incremental e iterativo que permite desarrollar una base funcional mínima y probada durante un periodo de tiempo definido para cada sprint.

El cliente cuenta con un amplio conocimiento de las funcionalidades con las cuales el software debe contar, por ende, al presentar cada incremento tendrá una idea más clara de lo que recibe y si es en realidad lo que necesita caso contrario brindara una retroalimentación más precisa de los cambios a realizar.

2.2 Roles de SCRUM para el proyecto

2.2.1 Scrum Master

En la ejecución de cada sprint el scrum master se fue alternando por los integrantes del equipo de desarrollo, para la planificación del siguiente sprint.

2.2.2 Dueño del producto

En un inicio del proyecto se definió a la coordinadora de CECASIS como dueña del producto la cual aprobaría cada uno de los sprints realizados.

Dentro de la metodología scrum el dueño del producto el encargado de la entrega de requerimientos, pero en este proyecto se identificó que la mayor parte de actividades son ejecutadas por el personal de CECASIS (técnicos y secretarias), por esta razón se tuvo que adaptar este concepto y optar por una práctica distinta en la cual el personal de CECASIS fue quien nos facilitó con toda la información necesaria para definir requerimientos del software.

2.3 Lista de sprints

Los sprints que se detallan a continuación permiten dividir el alcance de aplicación para generar entregables a corto plazo que aporten a las funcionalidades basadas en los requerimientos.

- **Sprint 1:** Con el fin de tener una idea clara de los requerimientos obtenidos se modelaron los procesos principales del CECASIS se los puede encontrar en el Anexo 1 y se realizó un prototipo general de toda la aplicación en el Anexo 2 se puede observar ejemplos del prototipo realizado.
- **Sprint 2:** Se generó el diagrama físico de la base de datos, se trabajó en la arquitectura del software y en el desarrollo de la funcionalidad para la administración de periodos, creación y administración de roles.

- **Sprint 3:** Se trabajó en el desarrollo de la funcionalidad para la administración de laboratorios, concluyendo con los módulos de reservación semestral y reservación diaria, obteniendo una nueva versión del software, obteniendo una nueva versión del software.
- **Sprint 4:** Se trabajó en el desarrollo de la funcionalidad para la administración de inventario, con los módulos de préstamo de elementos y cambio de hardware, obteniendo una nueva versión del software, obteniendo una nueva versión del software.
- **Sprint 5:** Se trabajó en el desarrollo de la funcionalidad el ingreso de incidencias, obteniendo una nueva versión del software.
- **Sprint 6:** Se trabajó en el desarrollo para la generación de reportes en el alquiler de máquinas, reservación diaria, prestamos de elementos e ingreso de incidentes además de la parametrización de datos necesarios en el ingreso de incidencias y préstamo de elementos, obteniendo una nueva versión del software.

2.4 Planificación del sprint

Debido a la jornada laboral de los integrantes del equipo de desarrollo se tuvo una gran dificultad para mantener reuniones presenciales por lo cual la mayoría de las reuniones se las hizo vía video conferencia con la ayuda de la herramienta Zoom, en estas reuniones se daba seguimiento al trabajo realizado y los avances del sprint en curso.

Al terminar cada sprint se planifico reuniones con el dueño del producto (personal de CECASIS), en estas reuniones se evaluaba el cumplimiento de los requerimientos especificados y se recibía una retroalimentación sobre lo presentado.

2.4.1 Product Backlog

La priorización de las funcionalidades que el cliente requiere para ir validando el cumplimiento y funcionalidad en cada presentación realizada.

Tabla 1. Product backlog del proyecto

Sprint	Actividades	Avance
Seguridad, administración de periodos	Administración periodo	100%
	Administración campus	100%
	Administración carrera	100%
	Administración materia	100%
	Administración de usuarios	100%
	Administración permisos por usuarios	100%
Administración laboratorios	Administración laboratorios	100%
	Asignación de un técnico por laboratorio	100%
	Administración software	100%
	Asignación software a laboratorio	100%
	Reservación laboratorios semestral	100%
	Reservación laboratorios diaria	100%
	Alquiler de Máquinas	100%
Administración inventarios	Administración tipo elementos	100%
	Administración elementos	100%
	Préstamo elementos	100%
	Administración tipo hardware	100%
	Administración hardware	100%
	Visualizar hardware por máquina	100%
	Visualizar hardware por laboratorio	100%
	Visualizar kardex hardware	100%
	Administrar solicitudes	100%
	Aprobación solicitud	100%
	Visualizar solicitudes	100%
Administración de incidentes	Ingreso Incidencias	100%
	Asignación Incidencias	100%
	Solución Incidencias	100%
Reportes	Reporte alquiler de maquinas	100%
	Reporte reservación laboratorios	100%
	Reporte préstamo de elementos	100%
	Reporte incidencias	100%
Parametrización	Multas en préstamo de elementos	100%
	Problemas comunes en incidencias	100%
	Ingreso incidencias por usuario	100%
	Ingreso Incidencias en laboratorio	100%

Nota: La tabla contiene el detalle del product backlog o requerimientos funcionales.
Fuente: Maigua Jesenia & Quilumba Henry (2018).

CAPÍTULO 3

3 ANÁLISIS Y DISEÑO

Para iniciar con el desarrollo del software se realiza un análisis de los procesos que van a ser automatizados. Los requerimientos funcionales se obtienen a partir de un prototipo, el cual permite al cliente aprobar los requerimientos o brindar retroalimentación en los cambios que sean necesarios realizarlos. Las historias de usuario también fueron parte de la toma de requerimientos, las cuales en ocasiones fueron tomadas registradas de forma manual y en otras ocasiones fue necesaria grabarlas por la complejidad de las funcionalidades. Adicional el personal del CECASIS proporciono los archivos digitales y físicos en que se registra las informaciones de los procesos, lo que permite aclarar las funcionalidades e ir definiendo el diseño de las pantallas y el modelo de la base de datos.

En la tabla 2, se describen algunos de los requerimientos funcionales que el sistema de cumplir, las siguientes páginas presentan algunos diagramas de caso de uso, diagrama de secuencia que permiten mejorar el análisis previo para el desarrollo del software.

Tabla 2.Especificación de requerimientos funcionales

Requerimiento	Especificación
El sistema debe tener roles.	Los roles que el sistema debe manejar son: coordinador/a, secretaria, técnico, docente y estudiante.
Actualizar datos de estudiantes y docentes en cada período.	Todos los usuarios serán cargados mediante archivos que contengan la información personal y sus credenciales, en cada período. Estudiantes: nombre, apellido, cedula, email, campus, carrera, nivel. Docentes: nombre, apellido, cedula, email, cedula, carrera, materia.
El coordinador/a es quien otorga los permisos de acceso a las funcionalidades.	El rol coordinador es el único que puede establecer los permisos a los diferentes módulos del sistema.
Permitir el préstamo de elementos.	La secretaria podrá realizar el préstamo de los elementos disponibles, la devolución por defecto será el mismo día o se le podrá ingresar una fecha límite.
Administrar los elementos que el centro posee.	Se debe realizar el ingreso de nuevos elementos, identificar los no devueltos y eliminados. Debe ser posible la parametrización del valor a cobrar por multa de elementos no devueltos.
Las reservaciones de laboratorios pueden ser de los siguientes tipos: semestral, diarias y mantenimiento.	Las reservaciones semestrales se podrán ingresar o no una fecha límite de reserva, las reservas diarias se las podrá hacer con 8 días de anticipación. Las reservaciones deberán finalizar automáticamente o podrán ser canceladas por los usuarios que las crearon.
Ingreso de incidentes por laboratorio	A los técnicos se les asignara laboratorios para la resolución de incidentes.
Realizar el alquiler de máquinas para los laboratorios indicados.	Se debe indicar previamente los laboratorios hábiles para el alquiler de máquinas, el rol secretaria realizara el alquiler a estudiantes o docentes ingresando el número de cedula.
Identificar el hardware en las máquinas de laboratorios.	Se podrá consultar el hardware que tienen las máquinas de los laboratorios indicando la serie, tipo y marca.
Generación de reportes por cambios de hardware.	El reporte deberá ser ingresado por un técnico, autorizado por el coordinador e indicar la ejecución del cambio de pieza.
Generar reportes.	Reportes en la reservación de laboratorios, préstamo de elementos, alquiler de máquinas e incidencias. Filtrar por fecha inicio y fecha fin o periodo.

Nota: Contiene algunos de los requerimientos funcionales del sistema.
Fuente: Maigua Jesenia & Quilumba Henry (2018).

3.1 Casos de uso

Los diagramas de casos de uso ayudaron a la identificación de los actores de cada proceso e identificar las funcionalidades que cada uno necesita en el software.

3.1.1 Caso de uso para el rol coordinador

Diagrama de caso de uso rol coordinador

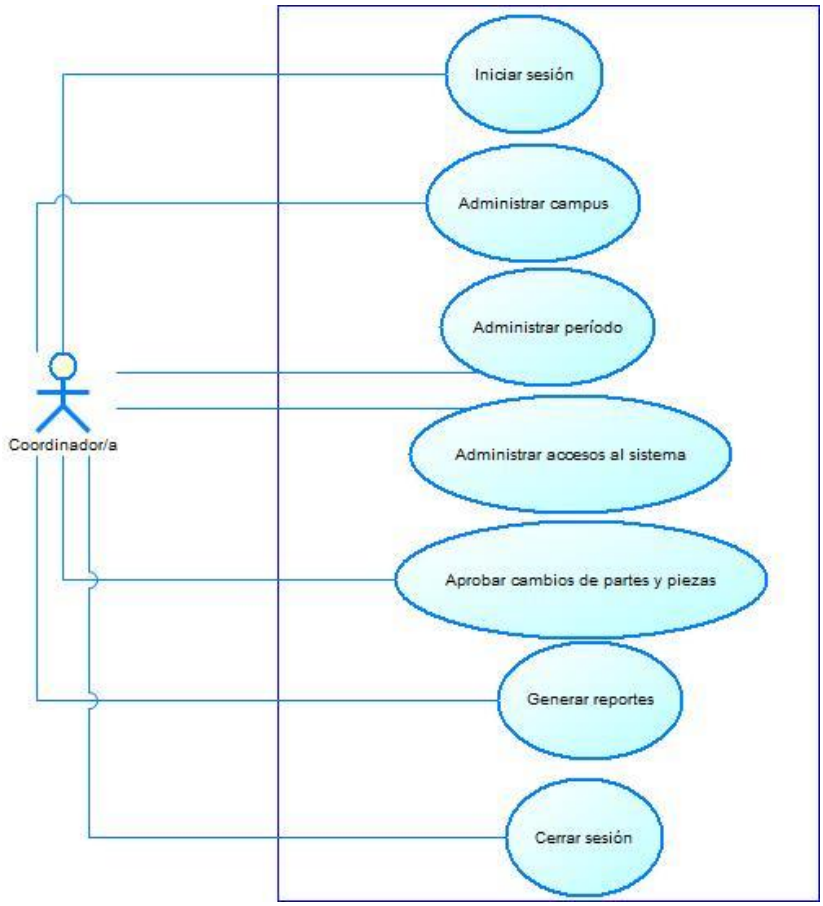


Figura 2. Actividades para el rol coordinador
Elaborado por: Maigua Jesenia & Quilumba Henry (2018).

En la figura 2, se muestra como el rol coordinador interactúa con la aplicación, siempre debe iniciar sesión para poder acceder a las funcionalidades del sistema las cuales están definidas mediante su rol. Después de un inicio de sesión exitoso tendrá disponible las

siguientes opciones: administrar campus, administrar periodo donde puede activar un nuevo periodo académico al finalizar el anterior, administrar accesos al sistema para los otros roles, aprobar cambios de piezas cuando un técnico lo solicita en un mantenimiento de los equipos, generar reportes y por último cerrar sesión para salir del sistema.

3.1.2 Caso de uso del rol secretaría

Diagrama de caso de uso rol secretaría

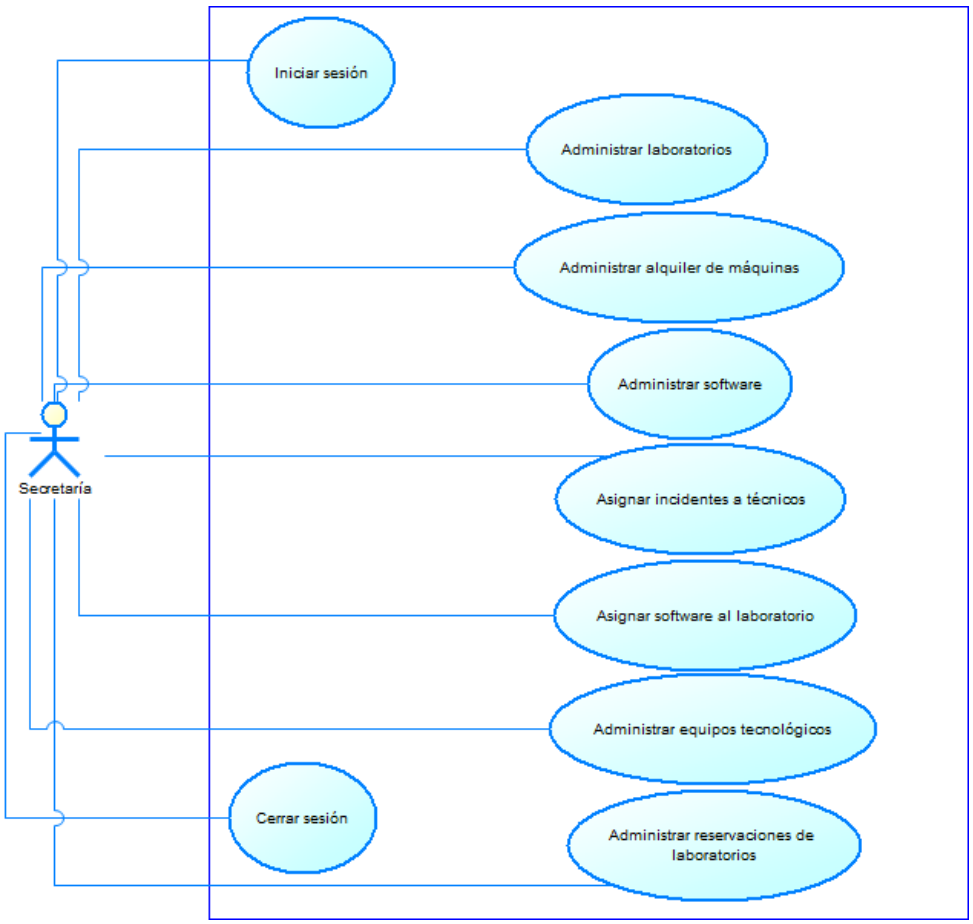


Figura 3. Flujo de actividades para el rol secretaría
Elaborado por: Maigua Jesenia & Quilumba Henry (2018).

En la figura 3, se puede visualizar como el rol secretaría puede hacer uso de las diferentes funcionalidades, el primer paso es el inicio de sesión, después de cumplir con el prerrequisito puede hacer uso de las opciones acorde al rol las cuales son: administrar

laboratorios, administrar software, asignar software a un laboratorios, administrar equipos tecnológicos, administrar las reservaciones de laboratorios, administrar préstamos de equipos tecnológicos, administrar alquiler de máquinas, asignar incidentes a los diferentes técnicos y cerrar sesión.

3.1.3 Caso de uso del rol técnico

Diagrama de caso de uso rol técnico

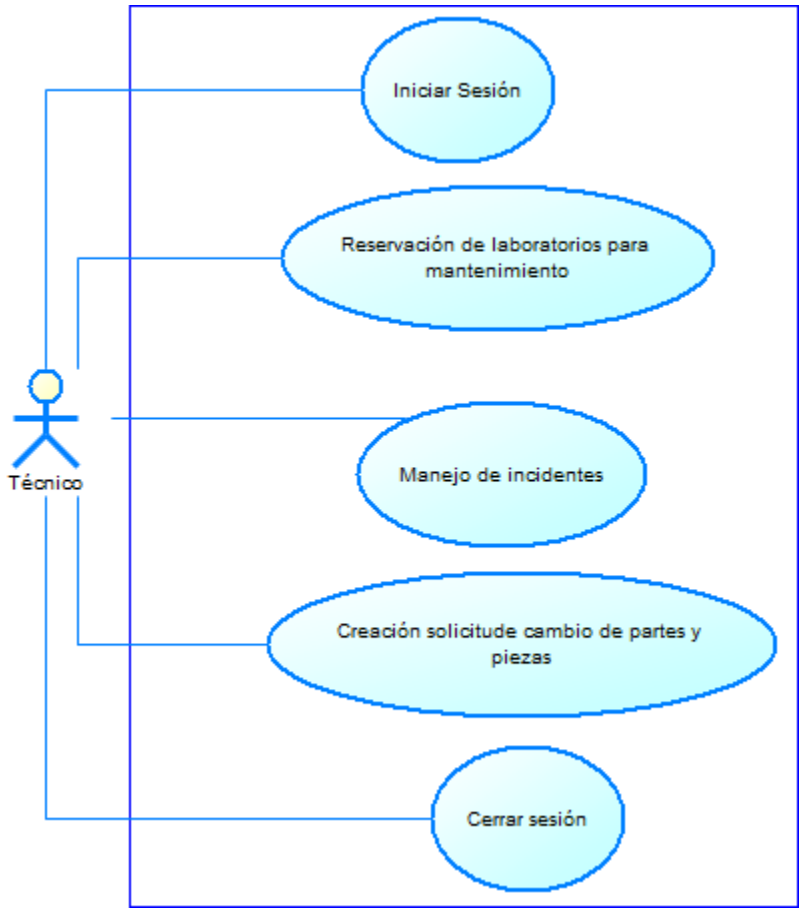


Figura 4. Actividades que el rol técnico puede realizar en el sistema
Elaborado por: Maigua Jesenia & Quilumba Henry (2018).

La figura 4, muestra como el rol técnico se relaciona con el sistema, en primera instancia es necesario el inicio de sesión, después de completar este paso se habilitan las siguientes funcionalidades: reservación de laboratorios para mantenimiento, manejo de incidentes

asignados, creación de solicitudes de cambios de partes y piezas de los equipos tecnológicos y cerrar sesión.

3.1.4 Caso de uso del rol docente

Diagrama de caso de uso rol docente

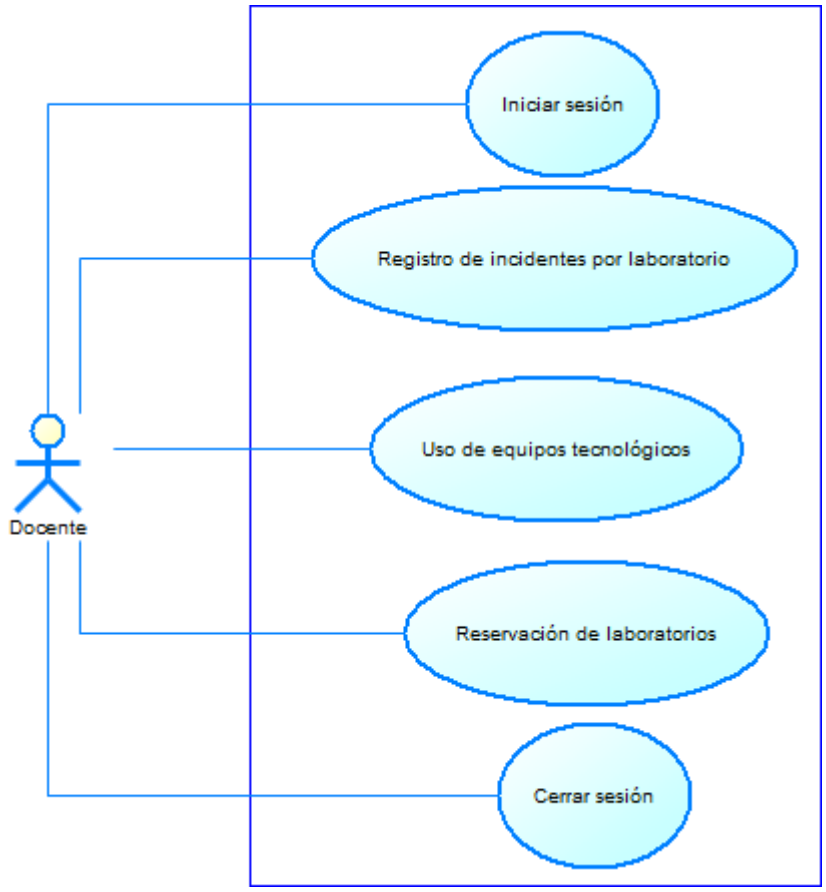


Figura 5. Actividades permitidas para el rol docente
Elaborado por: Maigua Jesenia & Quilumba Henry (2018).

El rol docente puede hacer uso de las funcionalidades como se muestra en la figura 5, después de iniciar sesión en el sistema, se presenta las siguientes opciones: registro de incidentes por laboratorios, uso de equipos tecnológicos, reservación de laboratorios y cerrar sesión finalizar el uso del sistema.

3.1.5 Caso de uso del rol estudiante

Diagrama de caso de uso rol estudiante

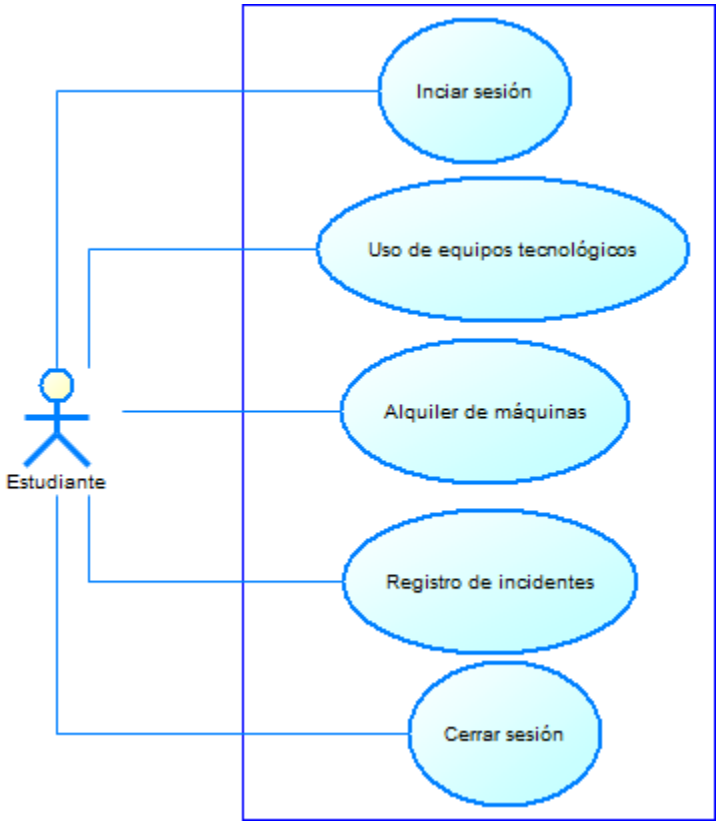


Figura 6. Trabajos que el rol estudiante está permitido realizar en la aplicación
Elaborado por: Maigua Jesenia & Quilumba Henry (2018).

Para el rol estudiante en la figura 6, se muestra las opciones de funcionalidades que tiene en el sistema. Como primer paso es el inicio de sesión, una vez que ingresa al sistema tiene habilitado las opciones que son: uso de equipos tecnológicos, registro de incidentes, alquiler de máquinas y cerrar sesión.

3.2 Diagramas de secuencia

Un diagrama de secuencia permite establecer la interacción dinámica que va a existir entre los diferentes objetos de la aplicación y la forma en que se van a comunicar.

3.2.1 Diagrama de secuencia para el inicio de sesión

Diagrama de secuencia inicio de sesión

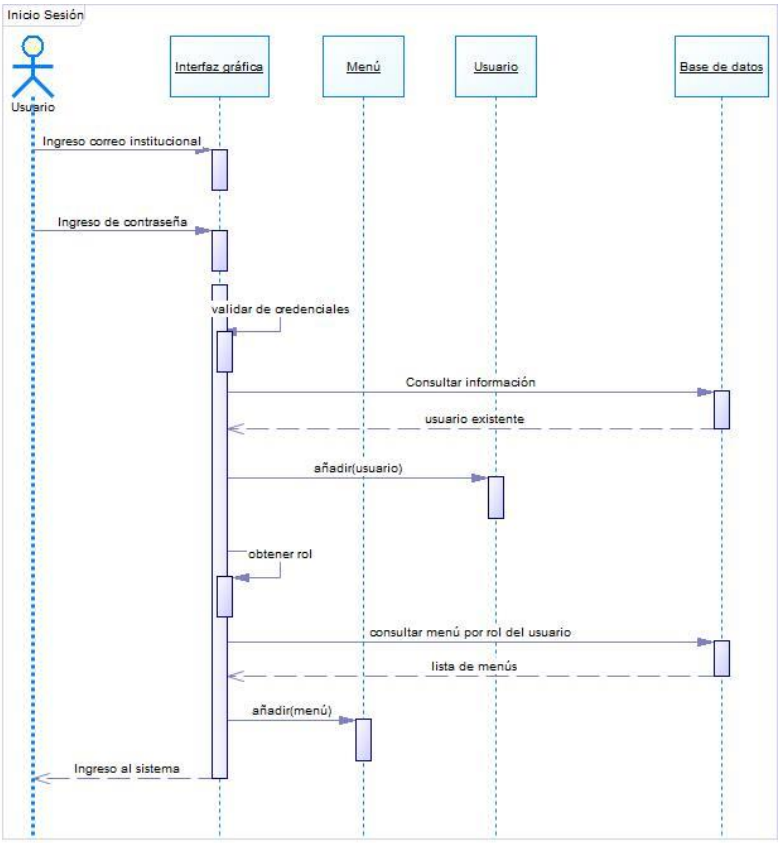


Figura 7. Secuencia que el sistema cumple para el inicio de sesión
Elaborado por: Maigua Jesenia & Quilumba Henry (2018).

En la figura 7, se muestra la secuencia y los diferentes objetos que permiten a un usuario iniciar sesión y posteriormente presentar el menú disponible según su rol. La secuencia se inicia cuando el usuario mediante la interfaz gráfica ingresa sus credenciales institucionales, se requiere realizar una validación de las credenciales, el siguiente paso es recuperar la información del usuario desde la base de datos, con esta información tenemos un nuevo objeto usuario, se utiliza el dato rol para nuevamente consultar a la base de datos y obtener el menú para la navegación a las diferentes páginas en el sistema.

3.2.2 Diagrama de secuencia reservación de laboratorios

Diagrama de secuencia reservación laboratorios

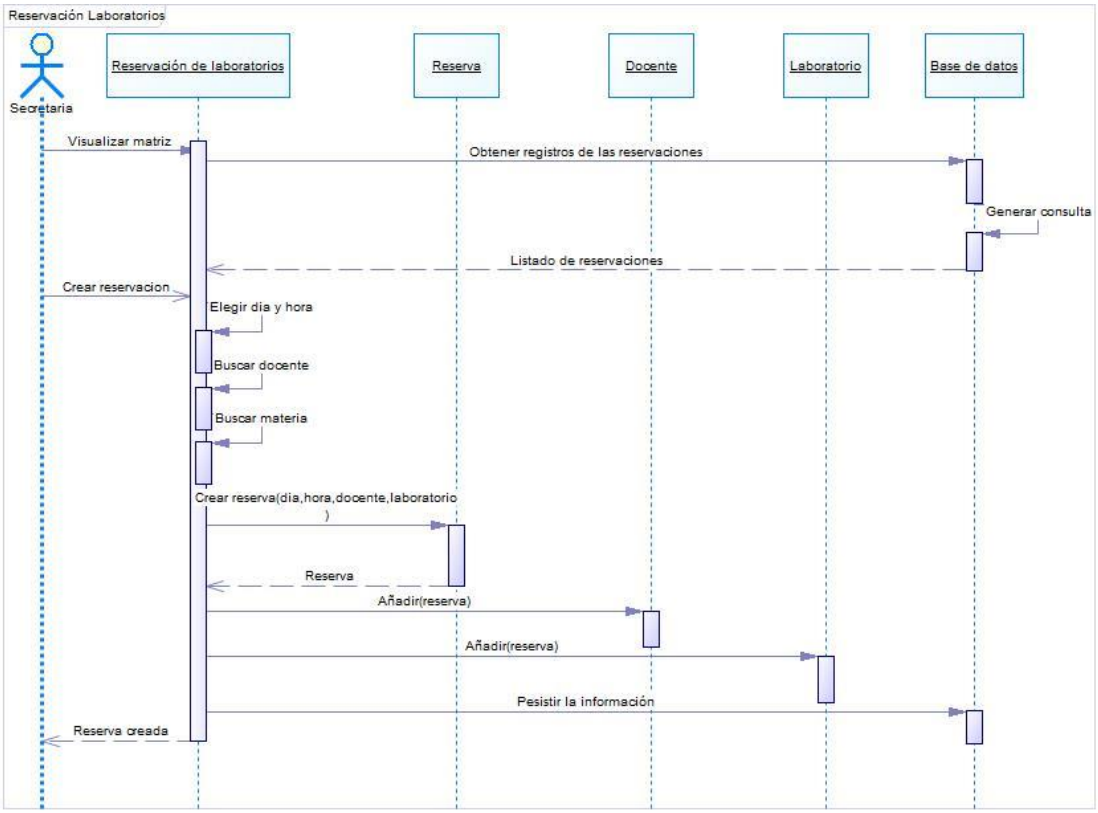


Figura 8. Secuencia para la reservación de laboratorios
Elaborado por: Maigua Jesenia & Quilumba Henry (2018).

En la figura 8, se visualiza la secuencia para la reservación de un laboratorio que es realizada por el rol secretaria. Una matriz similar a un horario de clases es presentada y contiene la información sobre las reservaciones realizas para un día y hora específico, toda la información es recuperada de la base de datos según el laboratorio. Para crear una nueva reservación se requiere elegir un día y hora disponible en la matriz, seleccionar un docente y las materias disponibles, la nueva reservación es asociada a un docente y laboratorio. Posteriormente se registra la reservación en la base de datos.

3.2.3 Diagrama de secuencia registro de incidentes

Diagrama de secuencia registro de incidentes

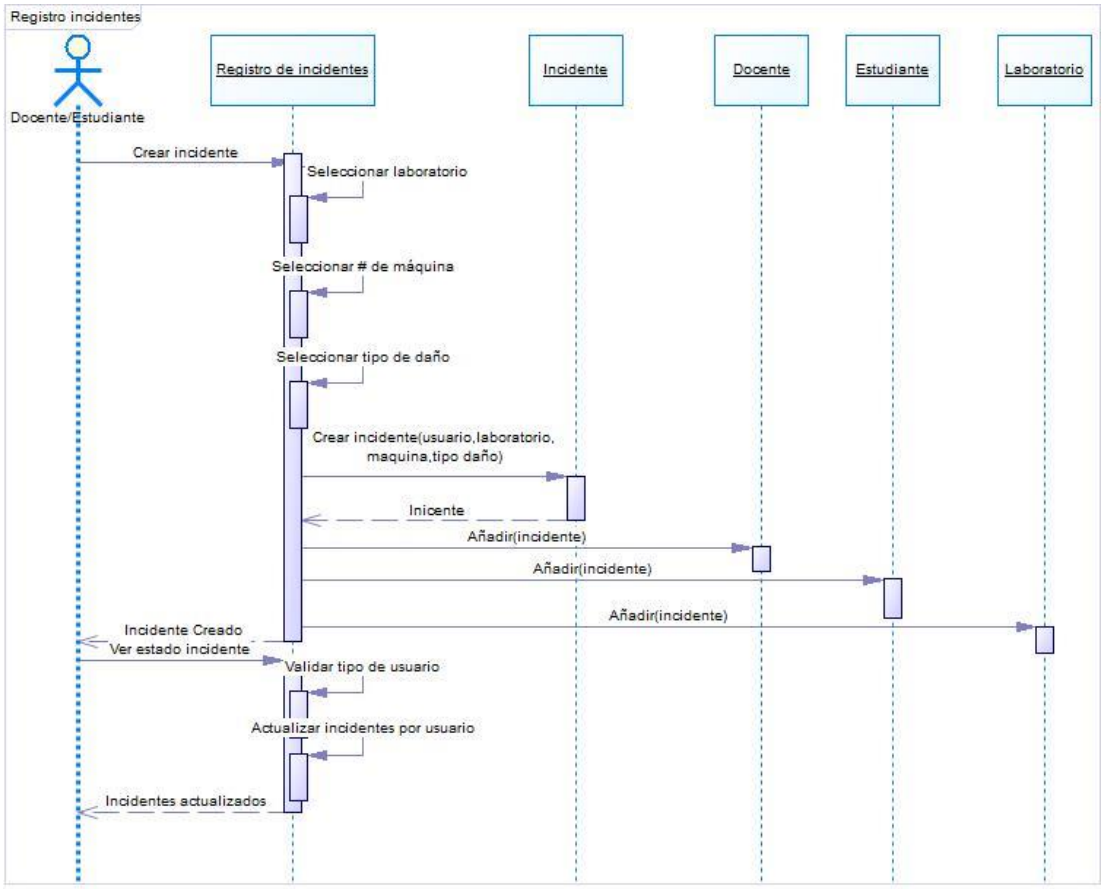


Figura 9. Secuencia para registro de incidentes
Elaborado por: Maigua Jesenia & Quilumba Henry (2018).

En la figura 9, presenta la secuencia necesaria que debe cumplir un rol docente o estudiante para registrar los incidentes de tipo hardware o software en un laboratorio. Primero hay que elegir un laboratorio y el número de máquina a la cual se va a reportar el incidente. Se crea un nuevo incidente que es registrado por un docente o estudiante y asociado a un laboratorio, toda la información para a ser almacenada en la base de datos.

3.3 Archivos digitales del CECASIS

Actualmente la información del CECASIS se encuentra guardada en archivos digitales de tipo Excel, donde mantienen información de los procesos que se automatizaron en el presente proyecto. Como parte del análisis los archivos sirvieron para tener un conocimiento más a fondo de la información que sería primordial que persista en la base de datos.

3.3.1 Archivo para la reservación de laboratorios

Formato reservación de laboratorios

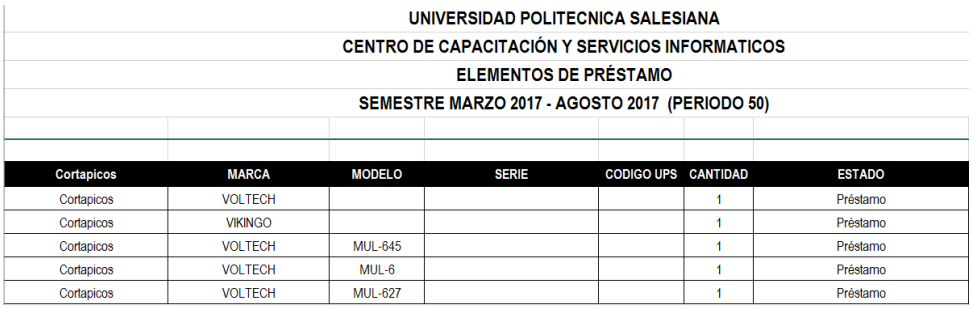
	VIERNES	21								
HORA	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10
07H00 - 08H00	Estudiantes	Carrera	Carrera	Italo	L. Fernández		Estudiantes	Seminario Amb.		
08H00 - 09h00	Estudiantes	Computación	Computación	Jumbo	Mecatro			M. Araque		
09H00 - 10H00	Estudiantes	Carrera	Carrera			I. Jumbo	Estudiantes	Alumnos		Carrera
10H00 - 11H00	Estudiantes	Computación	Computación			Sist	Estudiantes	Alumnos		Computación
11H00 - 12H00	Estudiantes	Alumnos	Carrera				Estudiantes	Alumnos		
12H00 - 13H00	Estudiantes	Alumnos	Computación				Estudiantes	Alumnos		
13H00 - 14H00	Estudiantes	Alumnos		Iván Montalvo		R. Pérez		Alumnos		
14H00 - 15H00	Estudiantes	Alumnos	Carrera	I. Montalvo			Carrera	Alumnos	Carrera	
15H00 - 16H00	Estudiantes	Alumnos	Computación	I. Montalvo			Computación	Alumnos	Computación	
16H00 - 17H00	Estudiantes	Alumnos	Carrera					Llerena	Carrera	
17H00 - 18H00	Estudiantes	Alumnos	Computación			17:30-19:30		Sistemas sig	Computación	
18H00 - 19H00	Estudiantes	Carrera					Carrera	Alumnos	Carrera	
19H00 - 20H00	Estudiantes	Computación					Computación	Alumnos	19:30-21:30	
20H00 - 21H00	Estudiantes							Alumnos		

Figura 10. Matriz de reservación de laboratorios
Elaborado por: Maigua Jesenia & Quilumba Henry (2018).

En la figura 10, se presenta la información y el formato que actualmente el CECASIS mantiene para realizar la reservación de un laboratorio. La información presentada en el archivo permite definir entidades, atributos y relaciones en la base de datos, además se puede ir teniendo una mejor visión de cómo se puede desarrollar la funcionalidad para la reservación de laboratorios.

3.3.2 Archivo para el préstamo de elementos

Inventario de elementos para préstamo



UNIVERSIDAD POLITECNICA SALESIANA
CENTRO DE CAPACITACIÓN Y SERVICIOS INFORMATICOS
ELEMENTOS DE PRÉSTAMO
SEMESTRE MARZO 2017 - AGOSTO 2017 (PERIODO 50)

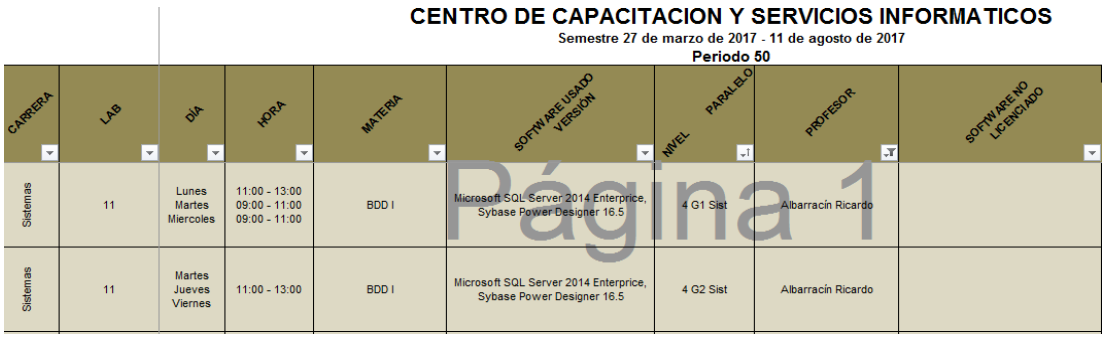
Cortapicos	MARCA	MODELO	SERIE	CODIGO UPS	CANTIDAD	ESTADO
Cortapicos	VOLTECH				1	Préstamo
Cortapicos	VIKINGO				1	Préstamo
Cortapicos	VOLTECH	MUL-645			1	Préstamo
Cortapicos	VOLTECH	MUL-6			1	Préstamo
Cortapicos	VOLTECH	MUL-627			1	Préstamo

Figura 11. Formato actual para el préstamo de elementos
Elaborado por: Maigua Jesenia & Quilumba Henry (2018).

La figura 11 presenta una fracción del documento donde se guarda la información del inventario que tiene el CECASIS de los elementos destinados a préstamo a estudiantes y docentes. La información que se proporciona en este documento ayuda a la definición sobre cómo ir modelando la base de datos para almacenar la información de los elementos. Es importante destacar que cada elemento maneja un estado para saber si está disponible para un préstamo, dañado o perdido.

3.3.3 Archivo software por docente

Asignación de software a docente



CENTRO DE CAPACITACION Y SERVICIOS INFORMATICOS
Semestre 27 de marzo de 2017 - 11 de agosto de 2017
Periodo 50

CARRERA	LAB	DIA	HORA	MATERIA	SOFTWARE USADO VERSIÓN	PARALELO NIVEL	PROFESOR	SOFTWARE NO LICENCIADO
Sistemas	11	Lunes Martes Miercoles	11:00 - 13:00 09:00 - 11:00 09:00 - 11:00	BDD I	Microsoft SQL Server 2014 Enterprice, Sybase Power Designer 16.5	4 G1 Sist	Albarracín Ricardo	
Sistemas	11	Martes Jueves Viernes	11:00 - 13:00	BDD I	Microsoft SQL Server 2014 Enterprice, Sybase Power Designer 16.5	4 G2 Sist	Albarracín Ricardo	

Figura 12. Software que el docente pedio para el uso semestral
Elaborado por: Maigua Jesenia & Quilumba Henry (2018).

La figura 12 presenta la petición de software que realiza un docente acorde a la materia a dictar. En el análisis del archivo se determina que es necesario mantener un inventario de software, un laboratorio debe tener software asignado para que puede ser filtrado al momento de realizar una reservación en el caso de ser necesario.

3.4 Diagrama de la base de datos

El diagrama entidad-relación se utilizó para diseñado de la base de datos, las entidades, atributos y relaciones fueron obtenidas después de realizar el análisis de los archivos proporcionados por el CECASIS.

En la figura siguiente, se presenta el diagrama final de la base datos que se utiliza en el proyecto y además permite guardar la información necesaria para que el sistema cumpla con las diferentes funcionalidades.

La primera versión se generó en base a los documentos proporcionados por el CECASIS. Esta versión no contenía todas las tablas necesarias debido a que el primer entregable se enfocó en la funcionalidad de registro, actualización y eliminación de la información requerida para las demás funcionalidades posteriores. En base a la retroalimentación brindada por el personal del CECASIS se obtiene una segunda versión.

La segunda versión incremento las tablas para la funcionalidad de préstamo de equipos tecnológicos, incidentes de laboratorios. Con una nueva revisión del sistema, se generan cambios en el diagrama, generando una tercera versión.

Diagrama de base de datos

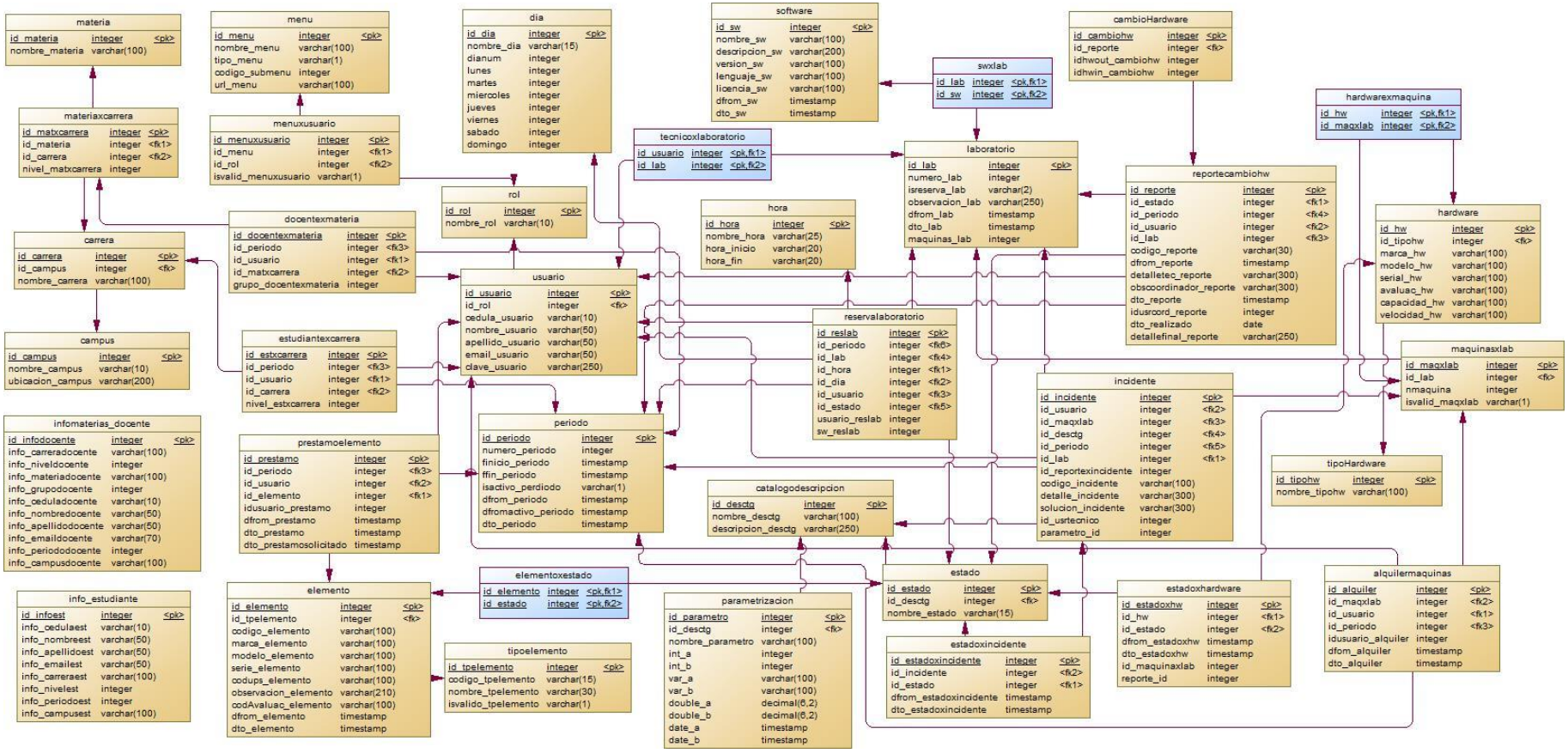


Figura 13. Modelo físico de la base de datos
Elaborado por: Maigua Jesenia & Quilumba Henry (2018).

CAPÍTULO 4

4 CONSTRUCCIÓN Y PRUEBAS

En el CECASIS trabajan usuarios que poseen diferentes conocimientos sobre la parte tecnológica, por este motivo se enfocó en construir una aplicación que se dé fácil uso, diseño responsivo y visualmente agradable. Actualmente los Frameworks proporcionan componentes que ayudan y permiten la construcción de aplicaciones de fácil usabilidad y adaptables ya que incorporan las librerías de Bootstrap, cual se enfoca en el diseño de los componentes y en que estos se adapten sin problemas a los diferentes tipos de dispositivos donde se puede ejecutar la aplicación.

La separación del modelo de datos, lógica de negocio y presentación en la interfaz, en diferentes capaz permite llevar un desarrollo ordenado, fácil de entender y robusto. Esto se debe a la utilización de la arquitectura MVC.

4.1 Arquitectura MVC

El empleo de los diferentes conceptos de la arquitectura MVC, permite llevar un flujo ordenado de las funcionalidades codificadas como se especifica en la siguiente figura.

Diagrama Modelo-Vista-Controlador

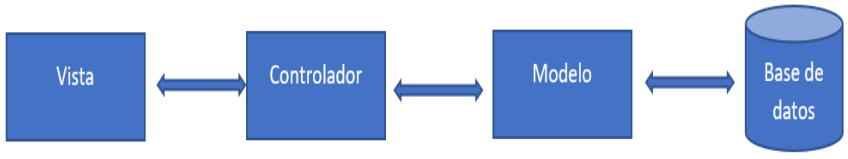


Figura 14. Diagrama Modelo-Vista-Controlador
Elaborado por: Maigua Jesenia & Quilumba Henry (2018).

En base a la figura 14, se describe el flujo como trabaja la arquitectura utilizada.

La vista permite presentar al usuario un interfaz, por la cual este realiza alguna acción para interactuar con la aplicación.

El controlador se encuentra a la espera de alguna petición realizada por el usuario, utiliza el modelo para enviar la petición a la base de datos.

La base de datos responde a las peticiones con un resultado que contiene la información almacenada.

El modelo de datos recibe la respuesta de la base de datos, la información es enviada al controlador para que realice la lógica de negocio y enviar su respuesta a la vista.

4.2 Diagrama de componentes

Diagrama de componente del sistema

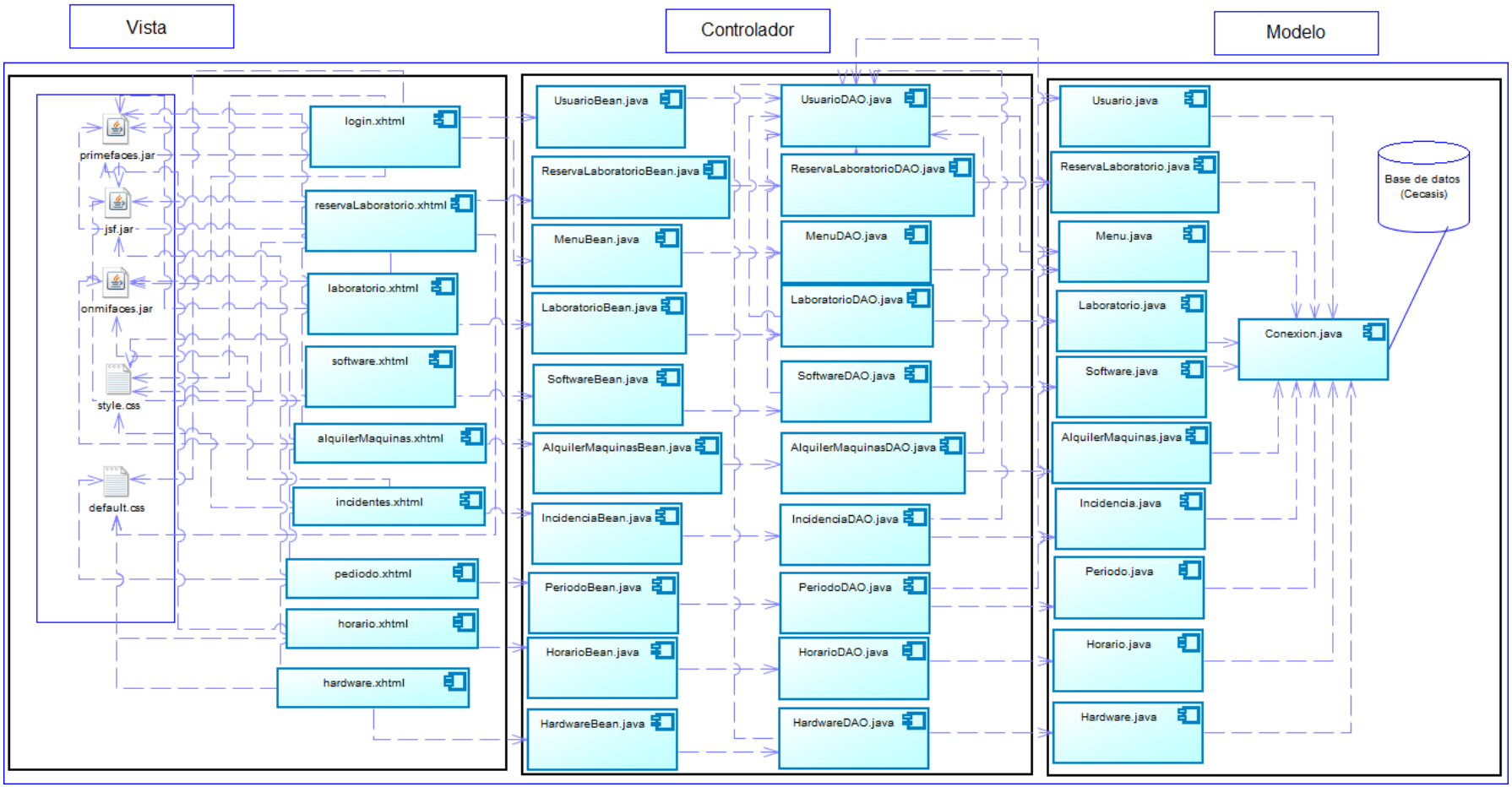


Figura 15. Diagrama de componentes
Elaborado por: Maigua Jesenia & Quilumba Henry (2018).

La figura 15, permite visualizar como el software se encuentra constituido por diferentes componentes que se encuentran interrelaciones para permitir obtener como resultado las funcionalidades emitidas en los requerimientos.

De izquierda a derecha se va a proceder a la explicación del diagrama.

En la capa de las vistas se encuentran las páginas web con extensión .xhtml, las cuales hacen uso de las librerías como: primefaces, jsf, omnifaes, además se utilizas archivos CSS para agregar un estilo personalizado a ciertos componentes html. Las páginas web contienen la interfaz gráfica para el usuario que le permite mantener una vista agradable y de fácil entendimiento para el uso.

La siguiente capa denominada controlador, se encuentra a la espera de cualquier evento que genera el usuario al usar las interfaces gráficas. Todo el código se encuentra guardado en un archivo .java, para procesar las peticiones conectándose a la base de datos y mostrar las respuestas.

4.3 Estándares de programación

El manejo de un estándar de programación ayuda a mantener una organización adecuada del código, las clases y paquetes que se utilizarán en el proyecto. El nombre de los paquetes se encuentra definido por el nombre de la institución separado por puntos como por ejemplo: ec.edu.ups.cecasis.

Paquetes del proyecto web

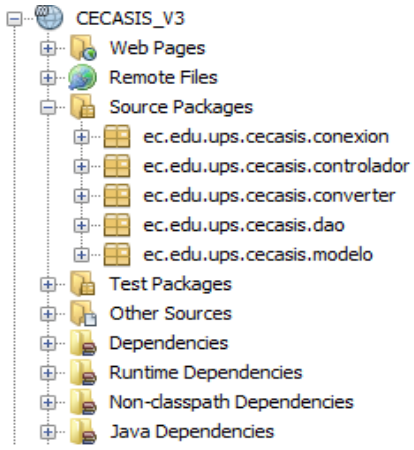


Figura 16. Estructura de paquetes para organizar el código
Elaborado por: Maigua Jesenia & Quilumba Henry (2018).

Las clases que se encuentran en la capa del controlador tienen el sufijo Bean, para distinguirlas de la capa del modelo de datos.

Clases de la capa controlador

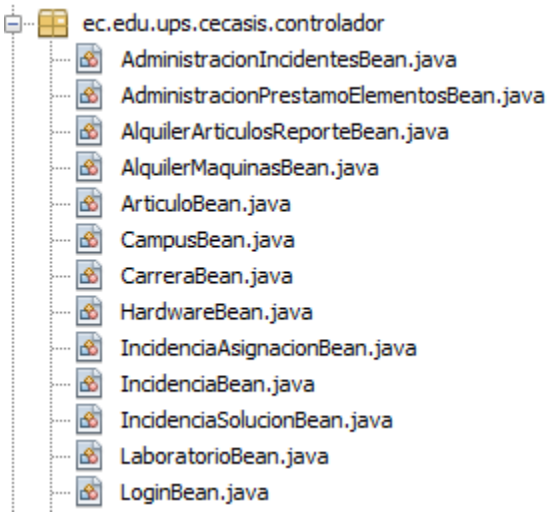


Figura 17. Clases que se encuentran en el paquete controlador
Elaborado por: Maigua Jesenia & Quilumba Henry (2018).

Para la comunicación con la base de datos las clases que contienen las sentencias SQL contienen el sufijo DAO.

Clases para el acceso a la base de datos

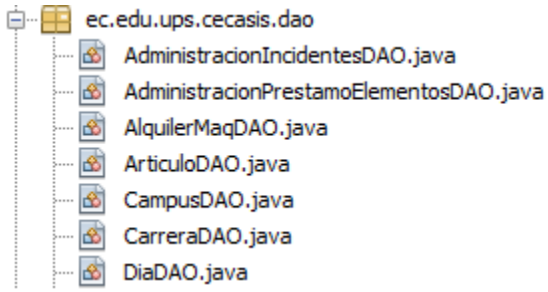


Figura 18. Clases DAO para realizar las transacciones a la base de datos
Elaborado por: Maigua Jesenia & Quilumba Henry (2018).

Todas las clases de la capa del modelo de datos utilizan clases POJO, que contienen los métodos getter y setter de los atributos para poder modificar los valores al momento de instanciar la clase.

Clase POJO para el modelo de datos

```
public class Horario {

    private int id_hora;
    private String nombre_hora;
    private String hora_inicio;
    private String hora_fin;
    private Boolean bloqueado;

    public int getId_hora() {
        return id_hora;
    }

    public void setId_hora(int id_hora) {
        this.id_hora = id_hora;
    }

    public String getNombre_hora() {
        return nombre_hora;
    }

    public void setNombre_hora(String nombre_hora) {
        this.nombre_hora = nombre_hora;
    }
```

Figura 19. Ejemplo clase POJO, con los métodos getter y setter
Elaborado por: Maigua Jesenia & Quilumba Henry (2018).

4.4 Código principal

Cada línea de código del proyecto aporta a la funcionalidad global de la aplicación web, se pone a consideración diferentes fragmentos del código para evidenciar como se encuentra estructurado.

4.4.1 Clase conexión

La clase conexión cumple la función de realizar el enlace entre la aplicación web y la base de datos, permite persistir la información generada mediante solicitudes por el usuario, para posteriormente recuperar la información realizando peticiones de tipo consulta a la base de datos.

Clase conexión a la base de datos

```
public void Conectar() throws Exception {
    try {
        Class.forName("org.postgresql.Driver");
        conexion = DriverManager
                .getConnection("jdbc:postgresql://localhost:5432/cecasis1?user=postgres&password=123456");
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public void CerrarConexion() {
    try {
        if (conexion != null) {
            if (conexion.isClosed() == false) {
                conexion.close();
            }
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Figura 20. Métodos de conexión a la base de datos
Elaborado por: Maigua Jesenia & Quilumba Henry (2018).

El método Conectar es el encargado de generar una instancia de conexión cuando se lo llame de un controlador. Aquí se encuentra el tipo de base de datos, puerto, nombre de base de datos y la clave respectiva para la conexión.

Para cerrar la conexión se crea el método CerrarConexion después de cada llamado a la base de datos.

4.4.2 Lectura archivo CSV

El método presentado en la figura 21, permite realizar la lectura de archivo de tipo .csv, para realizar la carga masiva de usuarios tanto para docentes y estudiantes, esta carga se debe realizar al activar un nuevo período académico. La funcionalidad de leer archivos facilita poblar la tabla usuarios sin tenerlo que hacer de forma manual, puesto que en el campus sur existen cientos de estudiantes y realizar el registro manual conllevaría mucho tiempo.

Método para leer archivos CSV

```
public void ingresarEstudiante(Periodo pr) throws Exception {
    EstudianteDAO estDAO;
    try {
        estDAO = new EstudianteDAO();
        List<Estudiante> estLista = new ArrayList<Estudiante>();

        CsvReader estudiantes_import = new CsvReader("C:/Users/USUARIO/Documents/Proyecto cecasis/adminPeriodo/estudiantes/Periodo" + pr.
        getNumero() + ".csv");
        estudiantes_import.readHeaders();

        while (estudiantes_import.readRecord()) {
            String cedula = estudiantes_import.get(0);
            String nombre = estudiantes_import.get(1);
            String apellido = estudiantes_import.get(2);
            String email = estudiantes_import.get(3);
            String campus = estudiantes_import.get(4);
            String carrera = estudiantes_import.get(5);
            int nivel = Integer.parseInt(estudiantes_import.get(6));
            int periodo = Integer.parseInt(estudiantes_import.get(7));
            estLista.add(new Estudiante(cedula, nombre, apellido, email, campus, carrera, nivel, periodo));
        }
        estudiantes_import.close();
        for (Estudiante est : estLista) {
            estDAO.registrarEstudiantes(est);
        }
    } catch (FileNotFoundException e) {
        conEstudiante = 1;
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

Figura 21. Lectura de un archivo CSV
Elaborado por: Maigua Jesenia & Quilumba Henry (2018).

4.4.3 Creación de vistas estudiantes y docentes

El código presentado en la figura 22, se definió crear una vista para el rol docente donde se almacenarán todos los usuarios, incluyendo las materias asignadas para el semestre, información del grupo, campus y la carrera.

La vista estudiante permite almacenar el campus, carrera y nivel al que pertenece.

Estas vistas facilitaran la consulta de la información de un docente o estudiante en las funcionalidades que sean requeridas.

Crear una vista para estudiante y docente

```
CREATE OR REPLACE VIEW docentes AS
SELECT t1.id_docentexmateria,
(SELECT t7.nombre_campus FROM campus t7
 WHERE t7.id_campus = (( SELECT t8.id_campus FROM carrera t8
                         WHERE t8.id_carrera = t2.id_carrera))) AS campus,
(SELECT t4.nombre_carrera FROM carrera t4 WHERE t4.id_carrera = t2.id_carrera) AS carrera,
t2.nivel_matxcarrera AS nivel,
(SELECT t6.nombre_materia FROM materia t6 WHERE t6.id_materia = t2.id_materia) AS materia,
 t1.grupo_docentexmateria AS grupo, t3.id_usuario, t3.cedula_usuario,
 pg_catalog.concat(t3.nombre_usuario, ' ', t3.apellido_usuario) AS nombre,
 t3.email_usuario AS email
 FROM docentexmateria t1, materiaxcarrera t2, usuario t3
WHERE t1.id_matxcarrera = t2.id_matxcarrera
AND t1.id_usuario = t3.id_usuario
AND t1.id_periodo = ((SELECT t5.id_periodoFROM periodo t5 WHEREt5.isactivo_perdiodo = 'Y'))
ORDER BY t2.id_materia;

CREATE OR REPLACE VIEW estudiantes AS
SELECT t1.id_usuario, t1.cedula_usuario,
pg_catalog.concat(t1.nombre_usuario, ' ', t1.apellido_usuario) AS nombre,
 t1.email_usuario AS email,
 ( SELECT t3.nombre_campus FROM campus t3 WHERE t3.id_campus = t4.id_campus) AS campus,
 t4.nombre_carrera AS carrera, t2.nivel_estxcarrera AS nivel
 FROM usuario t1, estudiantexcarrera t2, carrera t4
 WHERE t1.id_usuario = t2.id_usuario
 AND t4.id_carrera = t2.id_carrera
 AND t2.id_periodo = (( SELECT t5.id_periodo FROM periodo t5 WHERE t5.isactivo_perdiodo = 'Y'))
ORDER BY t1.id_usuario;
```

Figura 22. Las vistas ayudan a clasificar la información de docentes y estudiantes
Elaborado por: Maigua Jesenia & Quilumba Henry (2018).

4.4.4 Función elementos perdidos

Función elementos perdidos

```
CREATE OR REPLACE FUNCTION elementos_perdidos()
  RETURNS character AS
  $BODY$
DECLARE
    loc_result    CHAR(50);
    CUR_PERDIDO elementoxestado%ROWTYPE;
BEGIN

    FOR CUR_PERDIDO IN (select t2.id_elemento,t2.id_estado
                        from elemento t1, elementoxestado t2, prestamoelemento t3
                        where t1.id_elemento= t2.id_elemento
                        and t3.id_elemento = t1.id_elemento
                        and t3.dto_prestamo is null
                        and t2.id_estado = 4
                        and t3.dfrom_prestamo < CURRENT_DATE)
    LOOP
        update elementoxestado set id_estado = 5
        where id_elemento = CUR_PERDIDO.id_elemento;
    END LOOP;
        loc_result = 'success';
    RETURN loc_result ;
END;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
ALTER FUNCTION elementos_perdidos()
OWNER TO postgres;
```

Figura 23. Función para actualizar el estado de un elemento de forma automática
Elaborado por: Maigua Jesenia & Quilumba Henry (2018).

En la figura 23, presenta una función creada en postgresql la cual permite al sistema actualizar de forma automática el estado ha perdido de un elemento en préstamo. La función consiste en verificar de forma diaria la fecha en que un elemento fue prestado, por política del CECASIS un elemento se establece como perdido si no es devuelto el mismo día del préstamo.

4.4.5 Función finalizar reservaciones de laboratorios

Función finalizar reservaciones

```
CREATE OR REPLACE FUNCTION FINALIZARRESERVAS() RETURNS character  AS $BODY$
DECLARE
loc_result    CHAR(50);
CUR_RESERVA reservalaboratorio%ROWTYPE;
BEGIN

FOR CUR_RESERVA IN (select * from reservalaboratorio t1
    where t1.dtouso_reslab is not null
    and t1.dtouso_reslab<current_timestamp
    and dto_reslab is null)
LOOP
    update reservalaboratorio set dto_reslab = current_timestamp
    where id_reslab = CUR_RESERVA.id_reslab;
END LOOP;
    loc_result = 'success';
RETURN loc_result ;
END;
$BODY$
LANGUAGE 'plpgsql' VOLATILE;
```

Figura 24. Función para la finalización de una reservación de forma automática
Elaborado por: Maigua Jesenia & Quilumba Henry (2018).

La figura 24, permite visualizar como las reservaciones pueden ser realizadas para un lapso de una hora o más, la función verifica la hora de finalización de la reservación y automáticamente la finaliza, lo estable como disponible para poder ser reservado. La funcionalidad ayuda cuando la secretaria no finaliza una reservación.

4.5 Interfaces del sistema

El sistema se encuentra constituido por diferentes interfaces de usuario, permitiendo una fácil interacción con las funcionalidades que fueron requeridas.

4.5.1 Interfaces para la reservación de laboratorios

Matriz reservación semestral

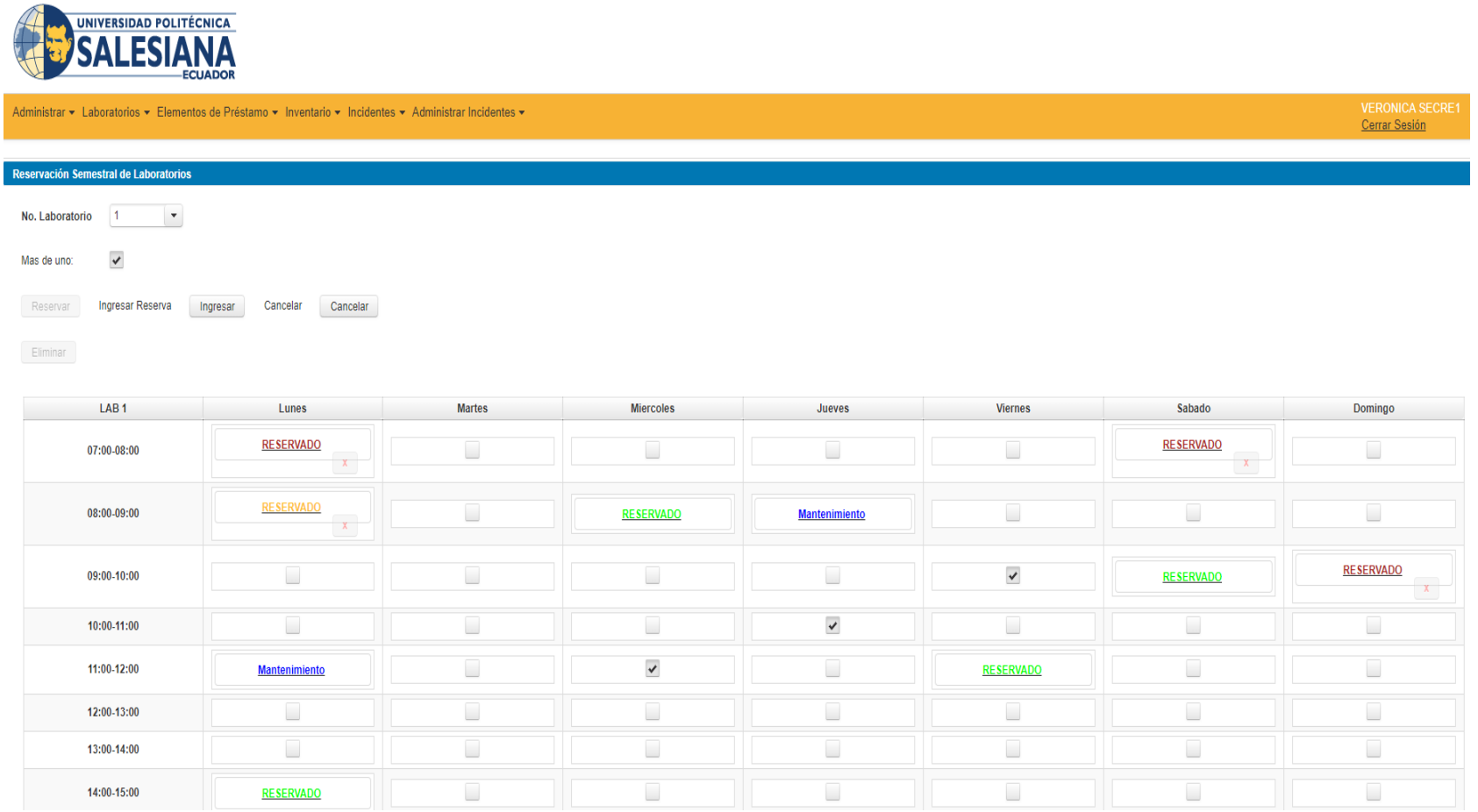


Figura 25. Matriz para la reservación de laboratorios
Elaborado por: Maigua Jesenia & Quilumba Henry (2018).

La figura 25, muestra la matriz que se desarrolló para poder visualizar y realizar una reservación semestral. Se presentan las reservaciones para uso diaria, y mantenimiento, para el usuario es una forma más fácil de poder ubicarse en la celda disponible para poder realizar la reservación. Se utiliza varios colores para el tipo de letra dependiendo el tipo de reservación. El color verde es una reservación realizada por un docente, el azul indica que para esa hora y día se encuentra en mantenimiento, los otros colores son reservaciones semestrales realizadas por la secretaria.

La siguiente figura presenta la reservación que realiza una secretaria para el uso diaria. En la matriz se presenta la información correspondiente al día actual y los próximos ocho días donde pueden existir reservaciones para un seminario de ser el caso. De igual manera se visualiza las reservaciones, además al pasar el mouse sobre la palabra Reservado se despliega la información de la persona quien solicito el servicio.

Matriz reservación de un laboratorio para uso diario

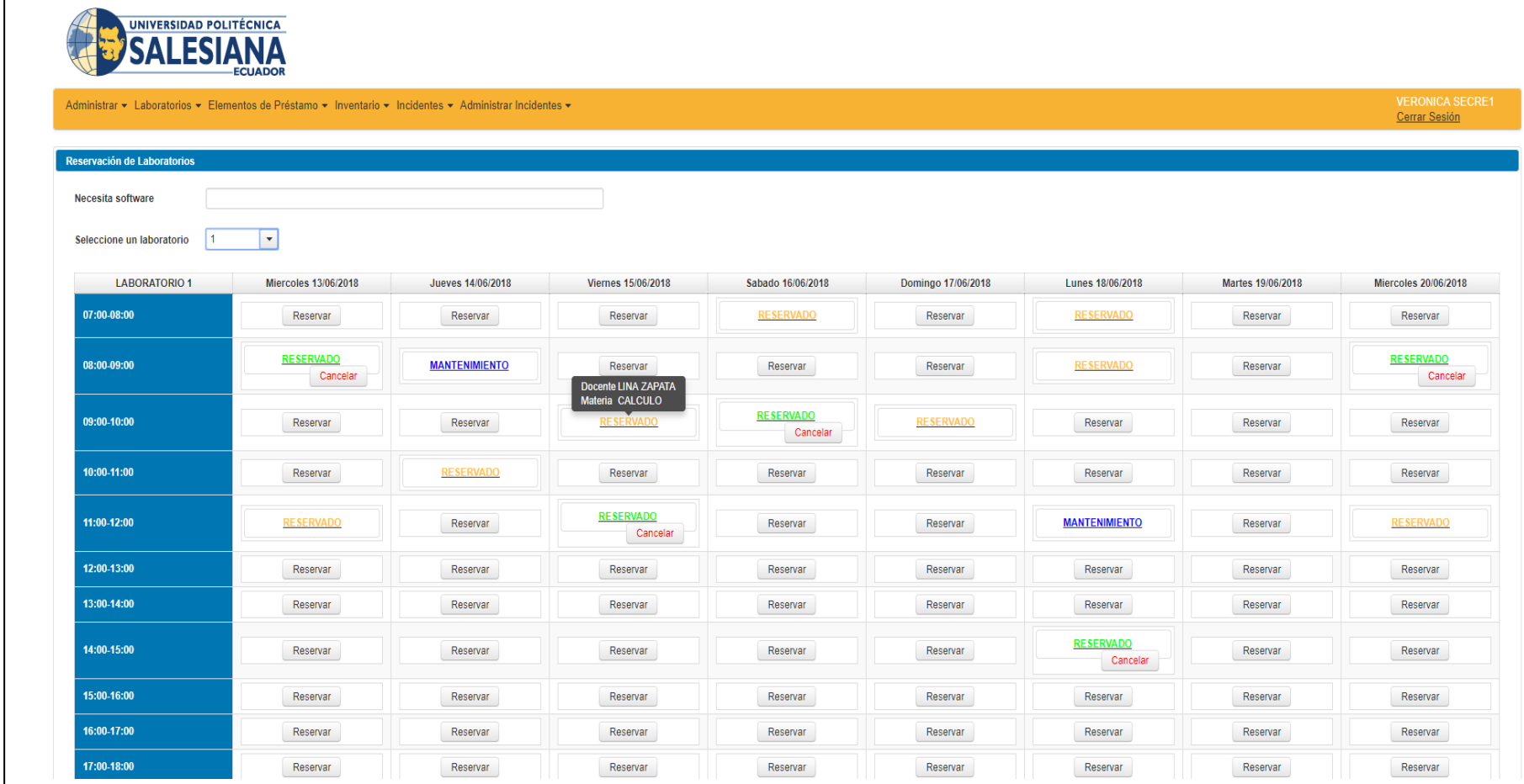


Figura 26. Reservación para uso diario de un laboratorio
Elaborado por: Maigua Jesenia & Quilumba Henry (2018).

4.5.2 Interfaz para el alquiler de máquinas

El rol secretaría es quien puede hacer uso de esta funcionalidad, como se muestra en la siguiente figura, según el laboratorio seleccionado se va a presentar el número de máquinas que contiene. Cada máquina puede ser alquila para realizar trabajos académicos, no existe un tiempo de uso. Cuando la máquina se encuentra disponible se presenta un botón llamado Alquilar para abrir un cuadro de dialogo y buscar la información del usuario que va a hacer uso de la máquina. Cuando la reservación se genera se presenta el siguiente texto En Uso y al pasar el mouse por la palabra se puede visualizar la información del usuario.

Alquiler de máquinas por laboratorio

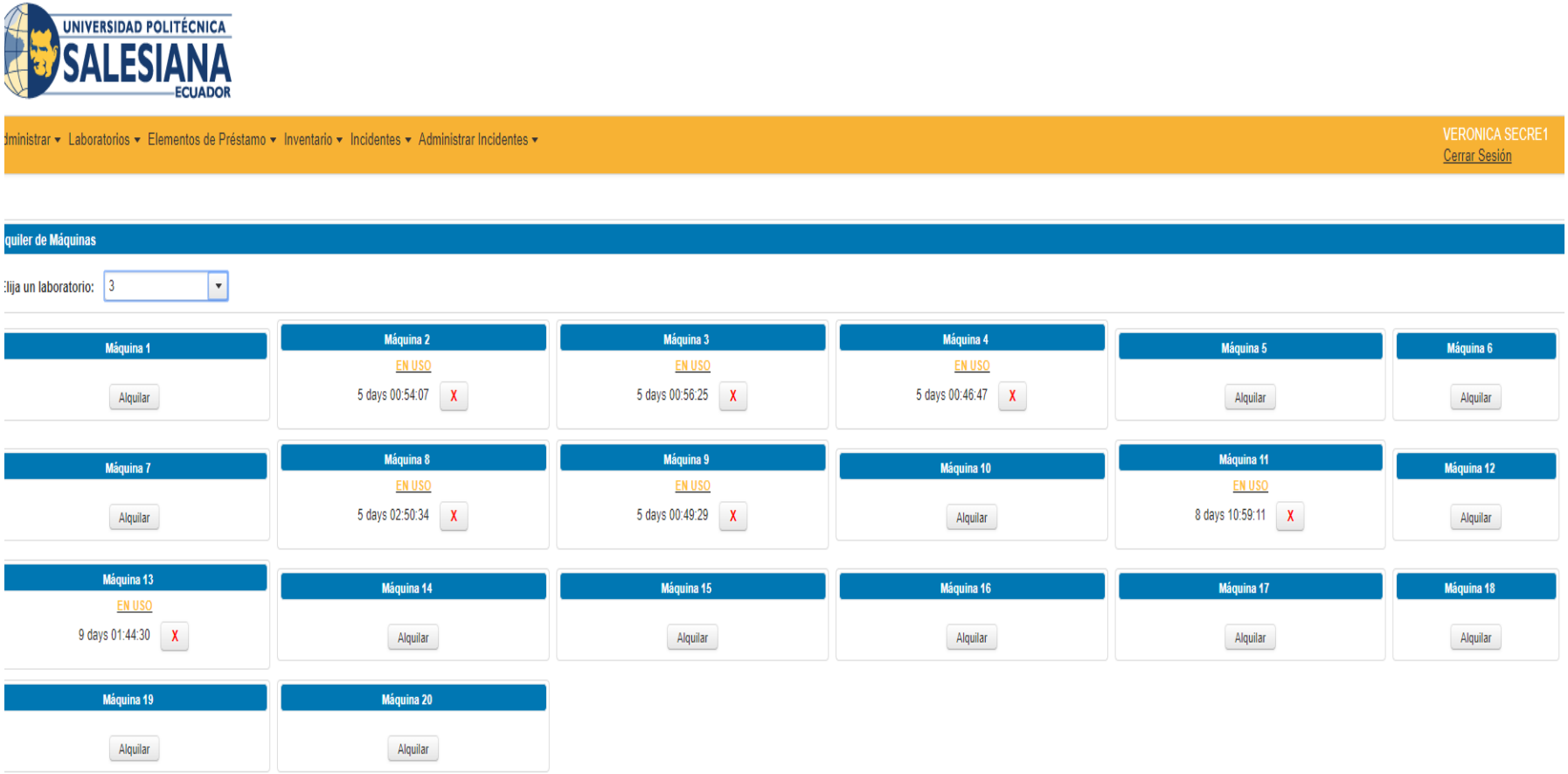


Figura 27. Matriz para el alquiler de máquinas por laboratorio
Elaborado por: Maigua Jesenia & Quilumba Henry (2018).

4.6 Pruebas de rendimiento

Las pruebas de rendimiento se realizaron haciendo uso de la herramienta "JMETER" de código abierto, entre sus funcionalidades permite definir un número de usuarios en un tiempo determinado para obtener resultados de comportamiento del sistema. La herramienta al finalizar las pruebas entrega reportes según se realicen las configuraciones.

Para realizar las pruebas, en la herramienta se requiere previamente configurar un grupo de hilos y el tipo de petición a utilizar.

Configuración del grupo de hilos

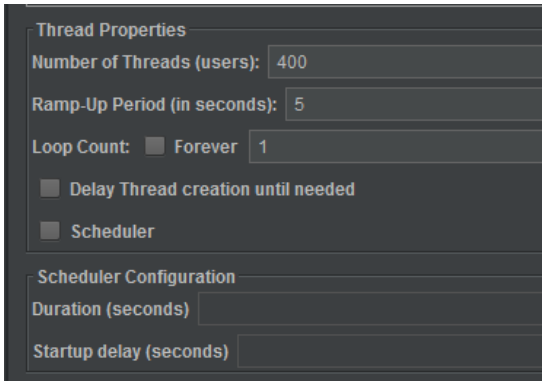


Figura 28. Configuración del grupo de hilos e intervalo de ejecución
Elaborado por: Maigua Jesenia & Quilumba Henry (2018).

Caso 1: Pruebas al módulo de autenticación para 400 usuarios en intervalo de 5 segundos.

Configuración de la petición HTTP

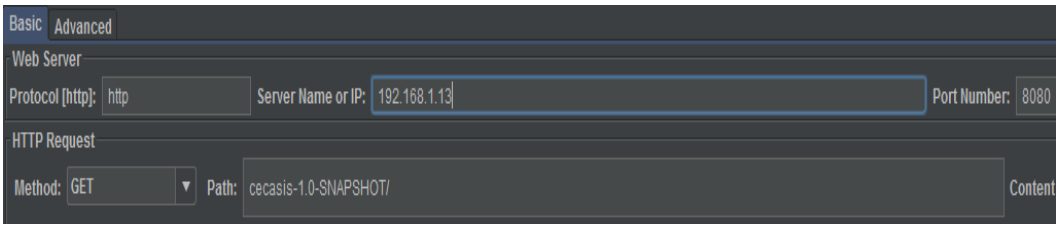


Figura 29. Petición HTTP para el módulo de autenticación
Elaborado por: Maigua Jesenia & Quilumba Henry (2018).

La figura 29, presenta la configuración realizada para la prueba del módulo de autenticación.

Árbol de resultados login

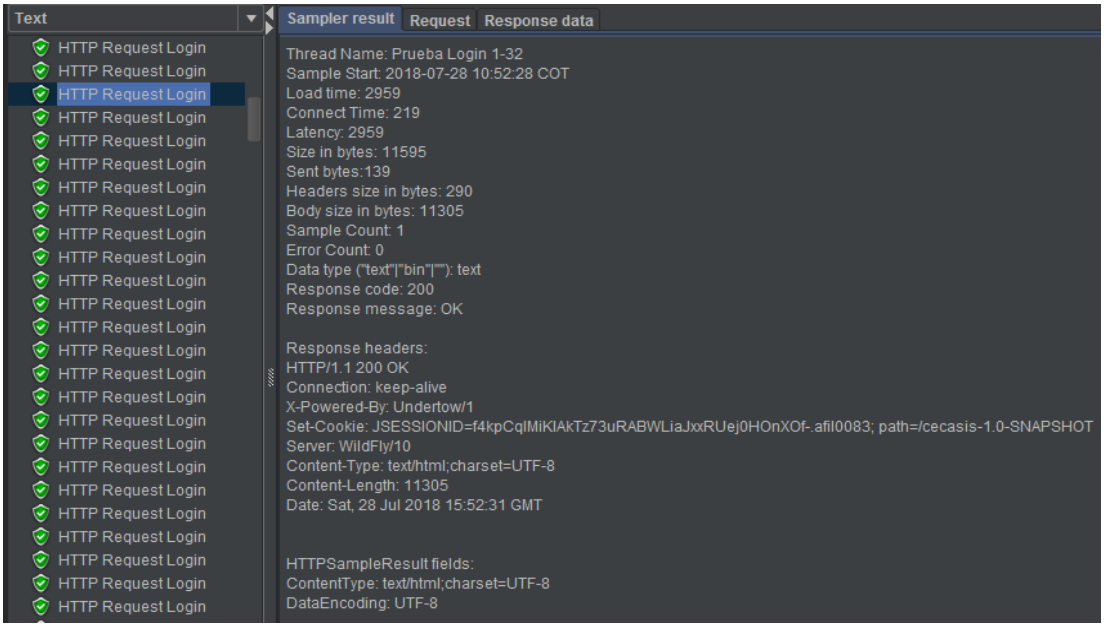


Figura 30. Resultados de las pruebas al módulo de autenticación
Elaborado por: Maigua Jesenia & Quilumba Henry (2018).

El árbol de resultados de la figura 30 muestra que las peticiones enviadas al sistema tuvieron respuesta exitosa

Informe resumido

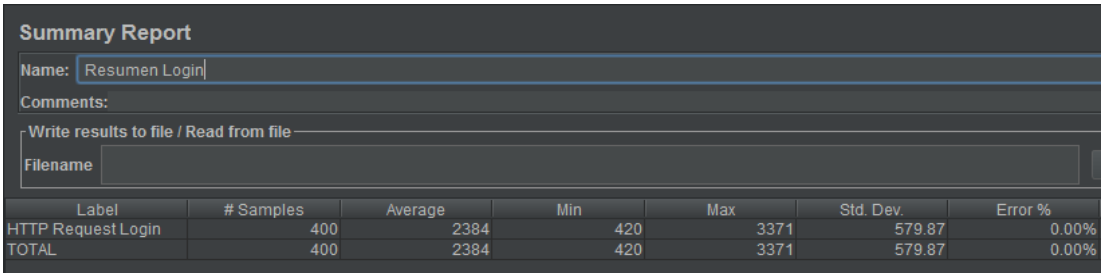


Figura 31. Informe resumido de la prueba realizada a login.
Elaborado por: Maigua Jesenia & Quilumba Henry (2018).

El informe resumido de la figura 31 presenta los resultados con detalles específicos.

Gráfica de la prueba a login

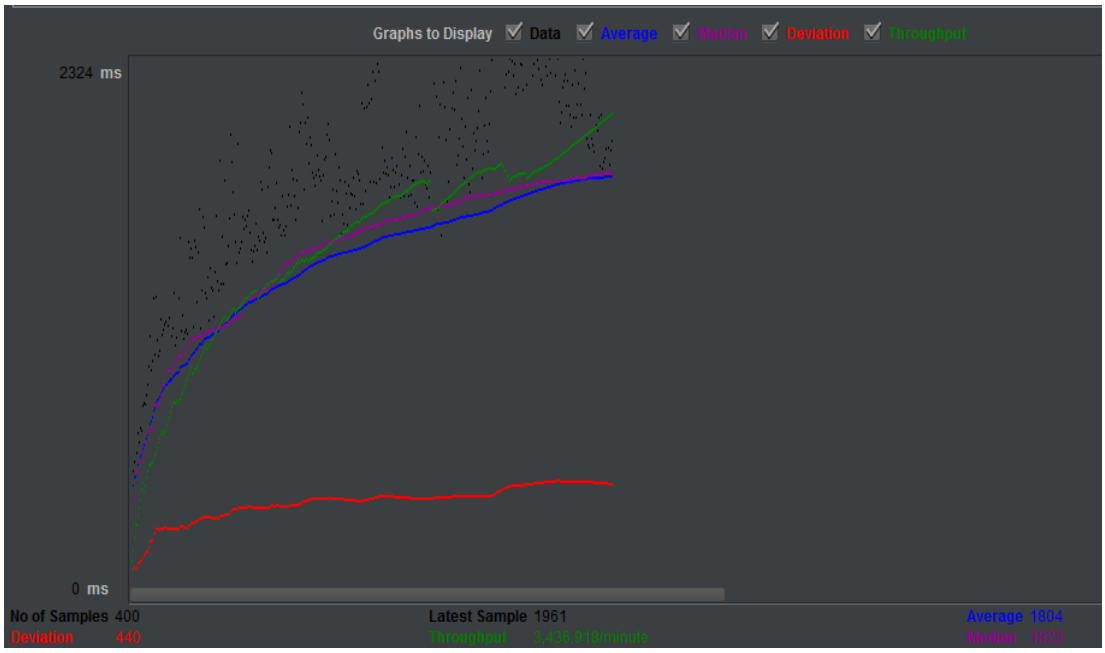


Figura 32. Gráfico con los resultados de las peticiones realizadas a login
Elaborado por: Maigua Jesenia & Quilumba Henry (2018).

La figura 32 muestra de forma gráfica como el sistema de comporta en relación con las peticiones enviadas.

Caso 2: Pruebas al módulo de reservación de laboratorios para 400 usuarios en intervalo de 5 segundos.

Petición HTTP al módulo de reservación de laboratorios

Figura 33. Configuración de la petición HTTP para la reservación de laboratorios
Elaborado por: Maigua Jesenia & Quilumba Henry (2018).

La figura 33, presenta la configuración realizada para la prueba de la reservación de laboratorios.

Árbol de resultados prueba reservación de laboratorios.

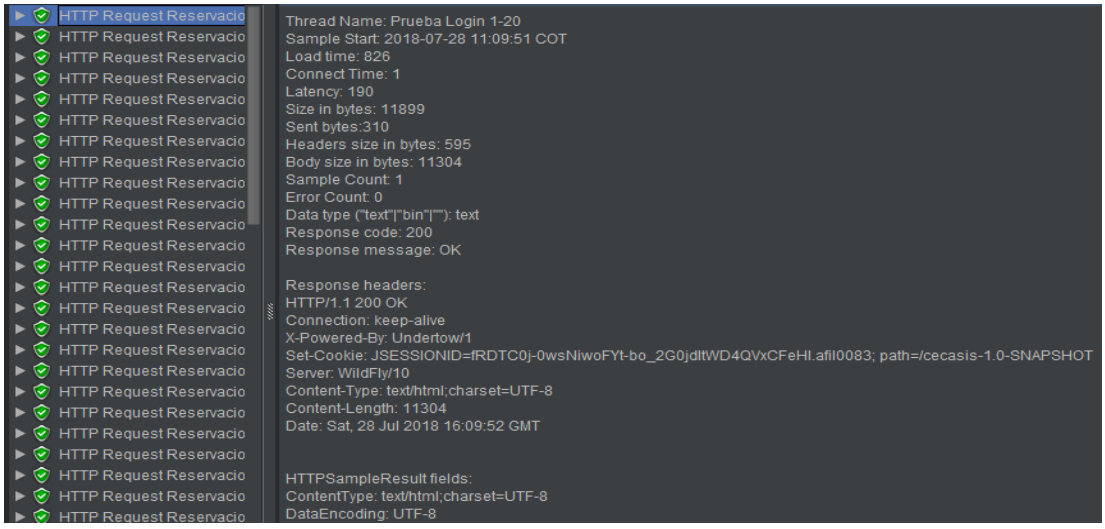


Figura 34 . Resultados de las peticiones enviadas a la reservación de laboratorios
Elaborado por: Maigua Jesenia & Quilumba Henry (2018).

El árbol de resultados para el módulo de reservación de laboratorios indica que todas las peticiones fueron atendidas con éxito.

Informe resumido de la prueba a la reservación de laboratorios

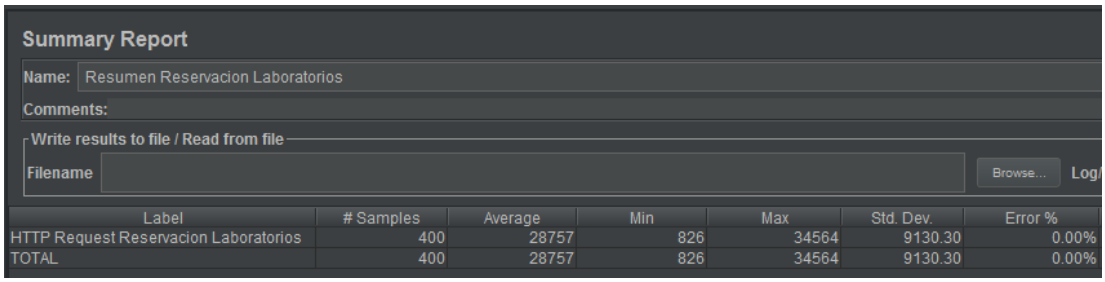


Figura 35. Resumen de resultados de las peticiones enviadas
Elaborado por: Maigua Jesenia & Quilumba Henry (2018).

La figura 35, presenta un resumen detallado de las pruebas realizadas al módulo de reservación de laboratorios.

Gráfica de la prueba a la reservación de laboratorios

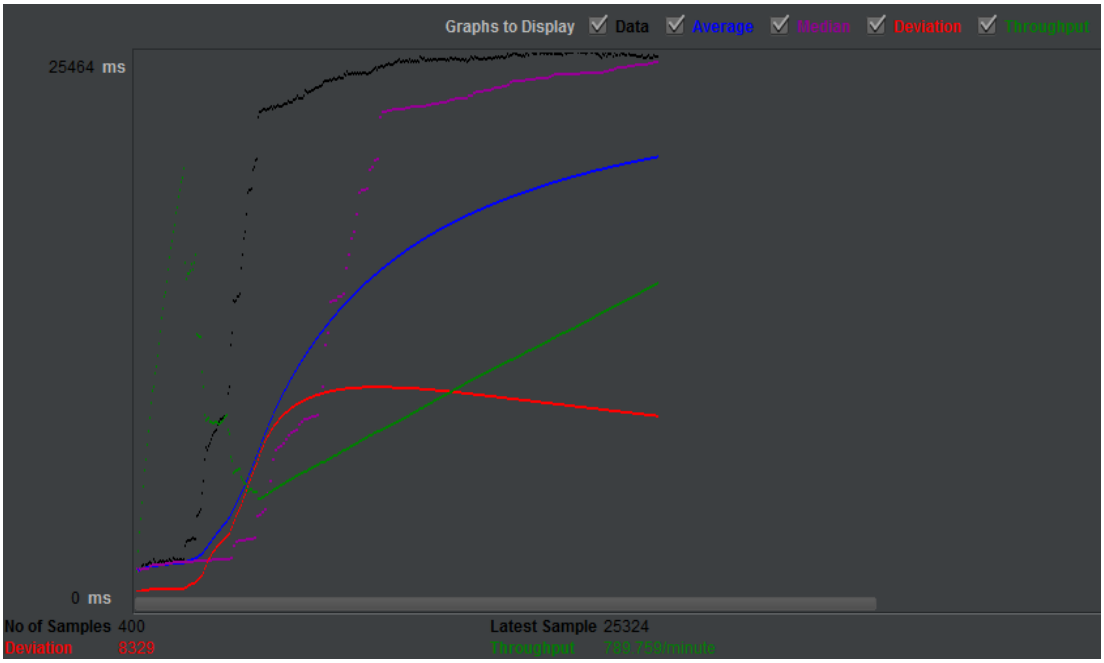


Figura 36. Árbol de resultados módulo reservación de laboratorios
Elaborado por: Maigua Jesenia & Quilumba Henry (2018).

La figura 36, muestra la una gráfica de los resultados de las pruebas realizadas a la reservación de laboratorios.

Las pruebas de rendimiento se realizaron a los módulos que más concurrencias tendrán en el sistema, para el módulo de seguridad la respuesta fue de 23,24 segundos y el porcentaje de error fue el 0%, para el módulo de reservaciones de laboratorios el tiempo de respuesta fue de 26,12 s con un porcentaje de error del 0%. Los resultados obtenidos se encuentran dentro de un rango aceptable acorde a las peticiones realizadas y el intervalo de tiempo. La utilización de los Frameworks y tecnologías para el desarrollo del sistema ayudan a obtener resultados positivos en las pruebas.

Un punto que destacar en las pruebas es que todas las peticiones enviadas a los dos módulos fueron atendidas con éxito manteniendo un buen rendimiento del sistema.

4.7 Pruebas funcionales

Una vez finalizado el desarrollo del software se realizó la toma de pruebas funcionales, para lo cual se contó con la ayuda de la coordinadora del centro, con esta práctica se pudo garantizar al dueño del producto que los requerimientos inicialmente propuestos, fueron correctamente atendidos.

Se mantuvo dos reuniones, en la primera se indicaron cambios a realizar en la segunda reunión se llegó al cumplimiento total de los requerimientos propuestos para el sistema.

En la siguiente tabla se tiene un resumen de las pruebas realizadas.

Tabla 3. Resumen de las pruebas funcionales del sistema

Código prueba	Resultado satisfactorio		Observaciones	
	Si	No	Primera reunión	Segunda reunión
PRUEBA01	X		N/A	N/A
PRUEBA02	X		N/A	N/A
PRUEBA03	X		N/A	N/A
PRUEBA04	X		N/A	N/A
PRUEBA05	X		N/A	N/A
PRUEBA06	X		N/A	N/A
PRUEBA07	X		Añadir tiempo de uso de la maquina	N/A
PRUEBA08	X		N/A	N/A
PRUEBA09	X		N/A	N/A
PRUEBA10	X		N/A	N/A
PRUEBA11	X		En la reservación del laboratorio validar que la fecha final sea mayor a la fecha más próxima del día seleccionado	N/A
PRUEBA12	X		N/A	N/A
PRUEBA13	X		N/A	N/A
PRUEBA14	X		N/A	N/A

PRUEBA15	X		N/A	N/A
PRUEBA16	X		N/A	N/A
PRUEBA17	X		N/A	N/A
PRUEBA18	X		N/A	N/A
PRUEBA19	X		N/A	N/A
PRUEBA20	X		Permitir identificar marca y modelo del hardware en reportes	N/A
PRUEBA21	X		N/A	N/A
PRUEBA22	X		N/A	N/A
PRUEBA23	x		N/A	N/A

Nota: La tabla contiene el resumen de las pruebas funcionales realizas por la coordinadora del CECASIS.

El detalle de todas las pruebas funcionales se la puede encontrar en el Anexo 3.

Tabla 4. Prueba funcional número 1

Prueba funcional	
Código	**PRUEBA01**
Modulo	Seguridad
Precondiciones	
• **Ingreso usuario (email institucional) correcto e incorrecto** • **Ingreso clave correcta e incorrecta**	
Descripción	
• **Verificar la validación en autentificación en el sistema**	
Resultado	
Exitoso, se verifica el correcto funcionamiento	
Estado de la prueba	Satisfactoria
Observaciones	N/A
Responsable	Yadira Jerez (Dueña del producto)

Nota: Prueba funcional al módulo de seguridad.

CONCLUSIONES

- La clave principal para el desarrollo del presente proyecto fue la recopilación de información, datos, permiso de acceso a los formatos actuales que el CECASIS mantiene para almacenar la información. Después del análisis de la información proporcionada, se procedió a las reuniones con del personal del CECASIS para entender más a detalle las funcionalidades que el sistema debe tener y los roles que están permitidos acceder a la aplicación.
- Utilizar una técnica como el prototipado ayudó tanto al cliente como al equipo de desarrollo a plasmar las ideas iniciales de como la aplicación se estructura en cuanto a la interfaz de usuario, además se descubren funcionalidades, definen parametrizaciones y aclara las reglas de negocio para el correcto funcionamiento de los procesos automatizados.
- La arquitectura Modelo-Vista-Controlador, para el proyecto ayudó a separarlo en diferentes capas, para el equipo de desarrollo facilita la implementación y la integración de la aplicación al finalizar cada entregable.

RECOMENDACIONES

- Se recomienda para la activación de un período académico, crear un archivo Excel tanto para estudiantes y docentes, que contiene la información que se guardara en la tabla de usuario en la base de datos, en caso de no considerar los archivos el sistema no tendrás usuarios para poder tener un correcto funcionamiento.
- Utilizar Frameworks que brinden facilidad de uso, documentación de clara sobre el funcionamiento y permitan optimizar tiempo en el desarrollo.
- Definir una arquitectura para la aplicación que brinde una fácil comprensión de la estructuración del código fuente a personas que requieran revisar la codificación o realizar mejoras.
- Establecer lapsos de tiempo para realizar un respaldo de la base de datos, para en un caso de extremo de pérdida o daño del sistema gestor de base de datos restaurar la base de datos y poder tener un funcionamiento normal de la aplicación.
- Para entregar un software de calidad en un proyecto y dependiendo la magnitud de mismo. optar por el prototipado es una buena estrategia para al inicio del proyecto aclara dudas sobre funcionalidades y al final de cada entregable garantizar que el cliente está satisfecho con lo requerido.

GLOSARIO DE TÉRMINOS

CECASIS: Centro de Capacitación y Servicios Informáticos del Campus Sur.

MVC: Modelo Vista Controlador.

DML: Lenguaje de Manipulación de Datos.

DDL: Lenguaje de Manipulación de Datos.

PBS: Product Breakdown Structure.

CSV: Comma-Separated Values.

JSF: Java Server Faces.

API: Application Programming Interface

CSS: Cascading Style Sheets

HTML: HyperText Markup Language

DAO: Data Acces Object

POJO: Plain Old Java Object

HTTP: Hypertext Transfer Protocol

SGDB: Sistema Gestor de Bases de Datos

REFERENCIAS BIBLIOGRÁFICAS

Bernal Bermúdez, J., & Fernández Muñoz , L. (2012). *Programación Orientada a objetos con Java (POOJ).* Madrid: Universidad Politécnica de Madrid.

Çivici, Ç. (2018). *User Guide 6.2.*

Deemer, P. &. (2012). *Una introducción básica a la teoría y práctica de Scrum.*

Fakhroutdinov, K. (junio de 2018). *OMG™ Unified Modeling Language™*. Obtenido de OMG™ Unified Modeling Language™: https://www.uml-diagrams.org/

González, R. (junio de 2018). *Ragose*. Obtenido de Ragose: https://ragose.com/que-es-un-diseno-responsivo-y-en-que-afecta-tu-web/

IBM Corporation. (junio de 2018). *IBM Corporation*. Obtenido de IBM Corporation: https://www.ibm.com/support/knowledgecenter/es/SS6RBX_11.4.3/com.ibm.sa.tutorial.doc/topics/Mod2_CreateSeqColl.html

Mateu, C. (2004). *Desarrollo de aplicaciones web.* Universidad de Oberta Catalunya.

Menzinsky, A. &. (2016). *Scrum Manager.* Iubaris Info 4 Media SL.

Oracle Corporation. (junio de 2018). *Oracle Corporation*. Obtenido de Oracle Corporation: http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html

Ortega, J. A. (2011). *Desarrollo de proyectos de software.* Estado de México.

Perez Mora, O., & Gibert Ginestà, M. (2009). *Bases de datos en PostgreSQL.* Catalunya: Editorial de la Universitat Oberta de Catalunya.

PostgreSQL. (junio de 2018). *PostgreSQL*. Obtenido de PostgreSQL: https://www.postgresql.org/docs/10/static/sql-commands.html

Pressman, R. (2014). *Ingeniería del software.* MCGRAW-HILL.

Recalde, T. (2018). *Administración de centros de computo.* Tecnológico ECOMUNDO.

Schwaber, K., & Sutherland, J. (2013). *La Guía Definitiva de Scrum: Las Reglas del Juego.*

Sheldon, R., & Oppel, A. (2010). *Fundamentos de SQL "Tercera edición"*. México: McGRAW-HILL INTERAMERICANA EDITORES, S.A. DE C.V.

Silberschatz, A., Korth, H. F., & Sudarshan, S. (2002). *FUNDAMENTOS DE BASES DE DATOS. Cuarta edición.* Madrid: McGRAW-HILL/INTERAMERICANA DE ESPAÑA, S. A. U.

Suárez, E. (junio de 2018). *Dispersium*. Obtenido de Dispersium: http://dispersium.es/prototipos-para-diseno-web/

Wildfly. (junio de 2018). *Wildfly*. Obtenido de Wildfly: http://docs.wildfly.org/12/Getting_Started_Guide.html

ANEXOS

Anexo 1. Diagramas de los procesos

Anexo 2. Prototipo de la aplicación

Anexo 3. Pruebas funcionales

Anexo 4. Diccionario de datos

Anexo 5. Préstamo-Elementos

Anexo 6. Horarios de laboratorios

Anexo 7. Software semestre

Anexo 8. Carta de aceptación del software