**ESCUELA POLITÉCNICA NACIONAL**

**ESCUELA DE FORMACIÓN DE TECNÓLOGOS**

**UNIDAD DE TITULACIÓN**

**DESARROLLO DEL SISTEMA AUTOMATIZADO PARA EL CONTROL – SEGUIMIENTO DE SÍLABOS Y PRÉSTAMO DE PROYECTORES DE LA ESFOT**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN ANÁLISIS DE SISTEMAS INFORMÁTICOS**

**CARRILLO MOROCHO JESSICA VANESSA**

Jessica.carrillo@epn.edu.ec

**DIRECTOR: ING. JUAN PABLO ZALDUMBIDE PROAÑO, M.Sc**

juan.zaldumbide@epn.edu.ec

**CODIRECTORA: ING. LUZ MARINA VINTIMILLA JARAMILLO, M.Sc**

marina.vintimilla@epn.edu.ec

**Quito, marzo 2020**

# **DECLARACIÓN DE AUTORÍA**

Yo, Jessica Vanessa Carrillo Morocho, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que o ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

Sin perjuicio de los derechos reconocidos en el primer párrafo del artículo 114 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación - COESC-, soy titular de la obra en mención y otorgo una licencia gratuita, intransferible y no exclusiva de uso con fines académicos a la Escuela Politécnica Nacional. Entregaré toda la información técnica pertinente. En el caso de que hubiese una explotación comercial de la obra por parte de la EPN, se negociará los porcentajes de los beneficios conforme lo establece la normativa nacional vigente.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Jessica Vanessa Carrillo Morocho**

# **CERTIFICACIÓN**

Como director del trabajo de titulación DESARROLLO DEL SISTEMA AUTOMATIZADO PARA EL CONTROL - SEGUIMIENTO DE SÍLABOS Y PRÉSTAMO DE PROYECTORES DE LA ESFOT. desarrollado por Carrillo Morocho Jessica Vanessa, estudiante de la carrera de Tecnología en Análisis de Sistemas Informáticos, habiendo supervisado la realización de este trabajo y realizado las correcciones correspondientes, doy por aprobada la redacción final del documento escrito para que prosiga con los trámites correspondientes a la sustentación de la Defensa oral.

|  |  |
| --- | --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| **Ing. Juan Pablo Zaldumbide Proaño, M.Sc** | **Ing. Luz Marina Vintimilla Jaramillo, M.Sc** |
| **DIRECTOR DE PROYECTO** | **CODIRECTORA DE PROYECTO** |

# **DEDICATORIA**

El presente proyecto va dedicado a familia, en especial a mi madre por brindarme su cariño, apoyo, paciencia y confianza de manera incondicional, siendo un pilar importante a lo largo de mi vida y por ende en la culminación de la carrera, ya que ha venido alentándome día a día a cumplir mis metas, ha estado presente en cada uno de mis triunfos y por hacer de mi una mejor persona con sus consejos.

**JESSICA VANESSA CARRILLO MOROCHO**

# **AGRADECIMIENTO**

Agradezco a mi madre por su amor incondicional, creer siempre en mí y ser mi guía, gracias a ella me he convertido en la persona que soy ahora, siendo para mi un gran ejemplo de vida ya que sola y con gran esfuerzo a logrado sacar a delante a mí, mis hermanas y hermano creando personas de bien con muchos valores morales, demostrando además su fortaleza en momentos difíciles y que todo se puede lograr.

A mis hermanas por brindarme su apoyo infinito y palabras de aliento durante cada etapa de mi vida, más aún en este proceso del desarrollo del presente proyecto, que mas que mis hermanas son mis mejores amigas.

A la Escuela Politécnica Nacional por acogerme, permitiéndome desarrollar y ampliar conocimientos útiles que serán usados a lo largo de etapa profesional.

A la Escuela de Formación de Tecnólogos por brindarme los mejores años de mi vida universitaria ya que, además, del amplio conocimiento adquirido no solo en lo académico sino también en lo personal, me ha permitido conocer personas especiales y generar un sin número de experiencias que siempre serán recordadas. Entre ellas a mi mejor amiga Katherine Lasluisa, la cual hasta el día de hoy me sigue brindando su apoyo, cariño y se ha convertido en mi máximo.

Al Ingeniero Juan Pablo Zaldumbide por la confianza que ha depositado en mí durante toda la carrera y más aún en el desarrollo del presente proyecto de titulación, destacando la calidad de persona y su gran profesionalismo.

**Jessica Vanessa Carrillo Morocho**

# **ÍNDICE DE CONTENIDO**

[**DECLARACIÓN DE AUTORÍA** 1](#_Toc38491297)

[**CERTIFICACIÓN** 2](#_Toc38491298)

[**DEDICATORIA** 3](#_Toc38491299)

[**AGRADECIMIENTO** 4](#_Toc38491300)

[**ÍNDICE DE CONTENIDO** 5](#_Toc38491301)

[**INDICE DE FIGURAS** 6](#_Toc38491302)

[**INDICE DE TABLAS** 7](#_Toc38491303)

[**LISTA DE ANEXOS** 7](#_Toc38491304)

[**RESUMEN** 8](#_Toc38491305)

[**ABSTRACT** 9](#_Toc38491306)

[1. INTRODUCCIÓN 10](#_Toc38491307)

[1.1. Planteamiento del problema 10](#_Toc38491308)

[1.2. Objetivo General 12](#_Toc38491309)

[1.3. Objetivo Específicos 12](#_Toc38491310)

[1.4. Alcance 12](#_Toc38491311)

[2. METODOLOGÍA 13](#_Toc38491312)

[2.1. Metodología de desarrollo 13](#_Toc38491313)

[2.2. Análisis y levantamiento de requerimientos 15](#_Toc38491314)

[2.3. Diseño de la base de datos y arquitectura del sistema 23](#_Toc38491315)

[3. RESULTADOS Y DISCUSIÓN 25](#_Toc38491316)

[3.1. Implementación del sistema 25](#_Toc38491317)

[3.2. Pruebas del sistema 43](#_Toc38491318)

[4. CONCLUSIONES Y RECOMENDACIONES 51](#_Toc38491319)

[5. REFERENCIAS BIBLIOGRÁFICAS 54](#_Toc38491322)

[6. ANEXOS 56](#_Toc38491323)

# **INDICE DE FIGURAS**

[Figura 1. Fases Extreme Programming (XP) 14](#_Toc38491442)

[Figura 2. Elementos de casos de uso 22](#_Toc38491443)

[Figura 3. Estructura de la base de datos del sistema 24](#_Toc38491444)

[Figura 4. Arquitectura del sistema 25](#_Toc38491445)

[Figura 5. Base de datos en PhpMyAdmin 28](#_Toc38491446)

[Figura 6. Creación del proyecto en Laravel 28](#_Toc38491447)

[Figura 7. Arquitectura de archivos del proyecto creado 29](#_Toc38491448)

[Figura 8. Parámetros de conexión a la base de datos 29](#_Toc38491449)

[Figura 9. Módulo inicio de sesión por credenciales 30](#_Toc38491450)

[Figura 10. Módulo reestablecer contraseña 31](#_Toc38491451)

[Figura 11. Email para reestablecer contraseña 31](#_Toc38491452)

[Figura 12. Barra con Menú de opciones (Administrador) 31](#_Toc38491453)

[Figura 13. Registro de usuario al sistema 33](#_Toc38491454)

[Figura 14. Registro de usuario al dispositivo biométrico 33](#_Toc38491455)

[Figura 15. Módulo importar sílabo 34](#_Toc38491456)

[Figura 16. Módulo de dispositivos de proyección 35](#_Toc38491457)

[Figura 17. Código de conexión y captura de datos del biométrico (Python) 36](#_Toc38491458)

[Figura 18. Registro de clases o tema del sílabo 37](#_Toc38491459)

[Figura 19. Gestión de dispositivos de proyección 38](#_Toc38491460)

[Figura 20. Reserva de dispositivo de proyección 39](#_Toc38491461)

[Figura 21. Devolución de dispositivo de proyección 39](#_Toc38491462)

[Figura 22. Código de conexión serial al Arduino (Python) 40](#_Toc38491463)

[Figura 23. Email de no devolución proyector/kit 40](#_Toc38491464)

[Figura 24. Email de no registro de tema 41](#_Toc38491465)

[Figura 25. Módulo de reporte de clases 42](#_Toc38491466)

[Figura 26. Script para ejecución de dos procesos simultáneamente 42](#_Toc38491467)

[Figura 27. Pruebas en el navegador Google Chrome 44](#_Toc38491468)

[Figura 28. Pruebas en el navegador Mozilla Firefox 44](#_Toc38491469)

[Figura 29. Pruebas en el navegador Opera 45](#_Toc38491470)

# **INDICE DE TABLAS**

[Tabla 1. Modelo de historia de usuario 15](#_Toc38491495)

[Tabla 2. Resumen de las historias de usuario de la primera iteración 16](#_Toc38491496)

[Tabla 3. Roles de usuario del sistema 17](#_Toc38491497)

[Tabla 4. Resumen de las historias de usuario de la segunda iteración 17](#_Toc38491498)

[Tabla 5. Resumen de las historias de usuario de la tercera iteración 18](#_Toc38491499)

[Tabla 6. Resumen de las historias de usuario de la cuarta iteración 19](#_Toc38491500)

[Tabla 7. Requerimientos funcionales del sistema 20](#_Toc38491501)

[Tabla 8. Requerimientos no funcionales del sistema 21](#_Toc38491502)

[Tabla 9. Resultados de las pruebas de compatibilidad 43](#_Toc38491503)

[Tabla 10. Prueba de aceptación 1 46](#_Toc38491504)

[Tabla 11. Prueba de aceptación 2 46](#_Toc38491505)

[Tabla 12. Prueba de aceptación 3 47](#_Toc38491506)

[Tabla 13. Prueba de aceptación 4 48](#_Toc38491507)

[Tabla 14. Prueba de aceptación 5 48](#_Toc38491508)

[Tabla 15. Prueba de aceptación 6 49](#_Toc38491509)

# **LISTA DE ANEXOS**

[I. Anexo A. Historias de usuario 56](#_Toc44518325)

[II. Anexo B. Diagramas de caso de uso 56](#_Toc44518326)

[III. Anexo C. Diagrama de base de datos del sistema 56](#_Toc44518327)

[IV. Anexo D. Tablas y atributos de la base de datos 56](#_Toc44518328)

[V. Anexo E. Prototipos de las interfaces del Sistema 56](#_Toc44518329)

[VI. Anexo F. Interfaces del sistema 56](#_Toc44518330)

# **RESUMEN**

El presente proyecto de titulación consiste en el desarrollo e implementación de un sistema web para el control-seguimiento de sílabos correspondientes a docentes de la Escuela de Formación de Tecnólogos (ESFOT) y la gestión de dispositivos de proyección, donde las acciones de reserva / devolución se realizan de la forma más eficiente posible, siempre que alguno de los 20 dispositivos se encuentre disponibles. Docentes y administradores tendrán acceso a los módulos mencionados al autenticarse por medio de un dispositivo biométrico.

Programación extrema (extreme programming, xp) fue la metodología ágil seleccionada para controlar los avances realizados en el proyecto, reduciendo el tiempo el entrega, errores y ejecución de pruebas, ya que cada una de las iteraciones cumplen un ciclo completo de las 4 fases establecidas por la metodología (Planificación, Diseño, Codificación, Pruebas).

Para el desarrollo del sistema se hizo uso de varias tecnologías multiplataforma como: MySQL para el almacenamiento de datos, Framework Laravel, librerias como JavaScript, Bootstrap, Jquery, etc. en la codificación, Python para la realizar la conexión serial al Arduino y toma de datos en tiempo real del dispositivo biométrico necesaria en la autenticación.

A continuación, se describen las secciones en las que se encuentra divido el documento, y en las cuales se distribuyó el proceso de desarrollo del sistema web:

En la sección I se detalla el planteamiento del problema, objetivo general, objetivos específicos y alcance del proyecto.

En la sección ll se describe la metodología ágil Programación Extrema (XP), su implementación en el proyecto, la identificación y análisis de requerimientos del sistema, el diseño de la base de datos y arquitectura del sistema.

En la sección lll se detalla el proceso de desarrollo del sistema web ejecutado a través de las diferentes Iteraciones, las herramientas utilizadas en su implementación, así como los resultados obtenidos de las pruebas realizadas en el sistema

Finalmente, la sección lV presenta las conclusiones y recomendaciones obtenidas tras el desarrollo del proyecto

**Palabras clave:** XP, automatización, biométrico, reserva, devolución

# **ABSTRACT**

This degree project consists of the development and implementation of a web system for the control and monitoring of syllables corresponding to teachers at the “Escuela de Formación de Tecnólogos (ESFOT)” and the management of projection devices, where the actions of reservation / return are made as efficiently as possible, provided that one of the 20 devices is available. Teachers and administrators will have access to the mentioned modules when authenticating themselves through a biometric device.

Extreme programming (xp) was the agile methodology selected to control the progress of the project, reducing the time of delivery, errors and execution of tests, since each of the iterations fulfills a complete cycle of the 4 phases established by the methodology (Planning, Design, Coding, Testing).

For the development of the system we used several multiplatform technologies such as: MySQL for data storage, Laravel Framework, libraries such as JavaScript, Bootstrap, Jquery, etc. in the coding, Python for the serial connection to the Arduino and real-time data collection of the biometric device required in the authentication.

Next, we describe the sections in which the document is divided, and in which the web system development process was distributed:

Section I details the approach to the problem, general objective, specific objectives and scope of the project.

Section II describes the Extreme Programming (XP) agile methodology, its implementation in the project, the identification and analysis of system requirements, the database design and system architecture.

Section III details the development process of the web system executed through the different Iterations, the tools used in its implementation, as well as the results obtained from the tests performed on the system.

Finally, section IV presents the conclusions and recommendations obtained after the development of the project

**Keywords**: XP, automatization, biometrics, reservation, devolution

# **INTRODUCCIÓN**

“La disciplina tarde o temprano vencerá a la inteligencia”

*Yokoi Kenji*

La automatización de procesos acelera el tiempo de ejecución de actividades, tras sustituir los procesos manuales por aplicaciones generando una serie de beneficios, convirtiéndose en una de las practicas más implementadas actualmente por organizaciones. Dicho término complementa y se encuentra combinado con el concepto IoT internet de las cosas considerado un sistema de dispositivos IoT de computación interrelacionados, máquinas mecánicas y digitales, objetos, animales o personas que tienen identificadores únicos con la capacidad de transferir datos a través de una red, sin requerir de interacciones humano a humano o humano a computadora y el cual ha tenido una considerable evolución en el mundo actual ya que fusiona estos procesos automatizados y los interconecta por medio de redes (Instituto Nacional de Ciberseguridad, 2018).

La serie de características de dispositivos IoT permiten la comunicación y cooperación con otros dispositivos, ya que son de fácil configuración remota y localizables tanto físicamente, como en la red de la que forman parte.

## **Planteamiento del problema**

Actualmente la Escuela de Formación de Tecnólogos (ESFOT) alberga alrededor de 700-900 estudiantes matriculados en cada semestre, distribuidos en 4 carreras de nivel tecnológico superior, por lo que, aproximadamente 60 docentes se encuentran encargados de cumplir con las diferentes actividades académicas. La ESFOT tiene una jornada laboral de 7:00 am a 20:00 pm (ESFOT, 2019).

Dicha jornada maneja una planificación de clases con sus horarios basados en un sílabo planteado al inicio de cada semestre académico, por cada docente responsable de cada materia y paralelo; en él se detallan los temas y las actividades académicas a desarrollarse en cada curso por cada período; cumpliendo, de esta manera, con horas semanales establecidas por la Subdirección de la ESFOT, en armonía con el Reglamento de Carrera y Escalafón Expedido por el CES (CES, 2012).

El área administrativa de la ESFOT labora en horario de 08:00 am a 17:00 pm y está encargada de gestionar tramites académicos y operativos institucionales. Una de las tareas realizadas por el personal administrativo es el préstamo de dispositivos de proyección (infocus) para las clases; los infocus se

encuentran físicamente en la dirección y la gestión de préstamos se lo hace en forma manual a los docentes o estudiantes autorizados (ESFOT, 2019).

Teniendo en cuenta la diferencia de horarios al inicio y fin de las jornadas entre docentes y administrativos, se presentan dos problemas. La primera problemática está asociada al seguimiento de los temas y actividades descritos en los sílabos por parte del personal docente, por lo que es indispensable al inicio o final de cada clase, se realice un registro del tema y las actividades desarrolladas avalado por la firma del docente en un documento físico, el cual permite respaldar y validar el cumplimiento de dictado de clases. Tomando en cuenta que algunos docentes dictan clases en diferentes carreras, se mantienen los registros de cada una de estas.

La segunda problemática está relacionada con el préstamo de proyectores, ya que actualmente es necesario que el docente se dirija hacia la dirección y realice un registro manual en un documento físico, para que posteriormente, algún administrativo o delegado suministre el dispositivo. El encargado de entrega-recepción debe cumplir con la jornada de clases, lo que significa que se requieren 2 personas que cubran la jornada de 7:00 a 20:00 horas.

Los procesos que permiten la adquisición de proyectores y seguimiento de sílabos llevados actualmente, pueden ser automatizados, optimizando el tiempo de los docentes al ejecutar dichas acciones a través de herramientas tecnológicas basadas en el concepto IoT (Internet de las cosas), lo que permitirá conocer en tiempo real la disponibilidad de los dispositivos, por lo que se plantea con estudiantes de la carrera de “Tecnología en Electrónica y Telecomunicaciones” el proyecto integrador “Desarrollo del sistema automatizado para el control – seguimiento de sílabos y préstamo de proyectores”.

Sistema en el cual, el adquirir y devolver un dispositivo de proyección se lo realizará de manera rápida y sencilla, evitando depender de alguna persona intermediaria. En el caso del control - seguimiento de sílabos se generará un reporte cada mes, en base a los temas registrados diariamente por cada docente, mediante una aplicación desarrollada para el caso; cada docente deberá firmar mensualmente el reporte para su legalización. Se tiene como ventajas la reducción del consumo de papel, digitalización del proceso lo cual conlleva a eliminar el manejo de una gran cantidad de documentos por parte del área administrativa.

Se deja explícito que no se intenta reemplazar el proceso actual, el alcance del proyecto es automatizarlo, usando herramientas de punta, implementando la solución basados en una metodología y cumpliendo con estándares de calidad para el manejo de la información

## **Objetivo General**

Desarrollar el sistema automatizado para el control - seguimiento de sílabos y préstamo de proyectores de la ESFOT

## **Objetivo Específicos**

* Determinar los requerimientos correspondientes a los módulos control-seguimiento de sílabos y préstamo de proyectores del sistema.
* Diseñar la base de datos y arquitectura de los módulos control-seguimiento de sílabos y préstamo de proyectores del sistema.
* Implementar el sistema basado en los requerimientos definidos.
* Probar el funcionamiento del sistema.

## **Alcance**

A través del sistema se automatizará procesos relacionados con el préstamo/devolución de dispositivos de proyección y el seguimiento de sílabos correspondientes a docentes, permitiendo llevar un control de su cumplimiento durante el transcurso del periodo académico, esto para evitar totalmente el papeleo y trámites manuales que se generan actualmente, mismo que se encuentra dirigido únicamente a personal administrativo y docencia de la ESFOT, ayudando además a tener un mejor y mayor control de los recursos.

El sistema implementó dos tipos de autenticación para poder controlar de mejor manera cada uno de sus módulos: el ingreso a través de credenciales (correo electrónico - contraseña) lo realizan únicamente administrativos con el fin de manipular funciones relacionadas con el ingreso, modificación, eliminación sea de usuarios, dispositivos de proyección, cronogramas, carreras, materias y períodos académicos.

Para el otro mecanismo de autenticación se hace uso de un dispositivo biométrico, por el que, tanto administradores o docentes tiene control de las funciones de reserva/devolución de los dispositivos de proyección tomando en cuenta que se tiene a disposición la cantidad de 20, así como la integración de un módulo para el registro de los temas de clase manipulado solamente por docentes. Cabe mencionar que dicha autenticación solo podrá ser ejecutada a través del navegador de la Tablet instalada junto a los casilleros o lockers que contienen los dispositivos de proyección, y en el que además se encontrará el biométrico.

Para garantizar el cumplimiento de sílabo correspondiente de cada docente, cada fin de mes se deben exportar e imprimir los reportes relacionados con dicha actividad, el cual será posible realizarlo al acceder al módulo reportes encontrado en las funciones administrativas del sistema, para su posterior validación a través de la firma del docente respectivo.

# **METODOLOGÍA**

## **Metodología de desarrollo**

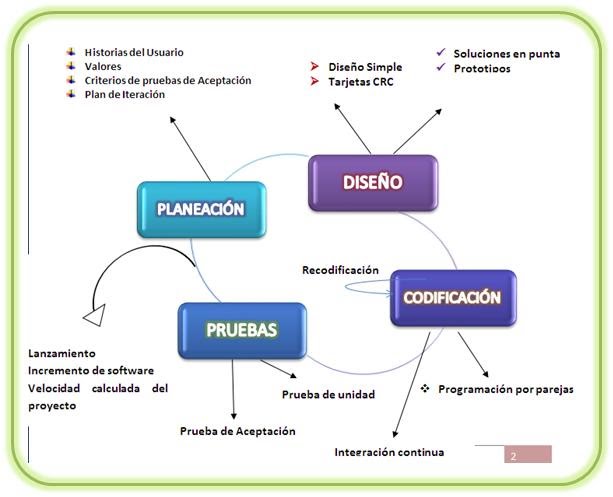
Para el desarrollo del sistema del presente Proyecto Integrador se utilizó la metodología ágil Extreme Programming (Programación Extrema o XP), la cual es adecuada en el desarrollo de sistemas grandes o pequeños con un tiempo corto de entrega, siendo capaz de adaptarse a cambios en cualquier etapa del proyecto. Incluye al cliente como parte importante del equipo de desarrollo ya que su enfoque principal se basa en satisfacer completamente sus necesidades y garantizar la calidad en el software. La metodología XP implementa un ciclo de vida dinámico que integra las siguientes fases: diseño, codificación y pruebas, las cuales se cumplen de manera simultánea en base a los requerimientos o historias de usuario generadas en la fase de planificación (Flores, 2020). Razones por las cuales se ha considerado a esta metodología como la mejor opción para el desarrollo del sistema.

**Programación extrema (extreme programming, xp)**

Metodología ágil para el desarrollo de software orientada en potenciar la comunicación e interacción entre el cliente y equipo de desarrollo, cuenta con la capacidad de adaptabilidad a través de la simplicidad en las soluciones implementadas para manejar los cambios imprevistos presentados por los clientes, lo que además minoriza la taza de errores, haciéndola idónea para proyectos con requerimientos poco definidos o cambiantes y que deban ser entregados en un periodo corto de tiempo. Por lo que cumple con sus 5 valores esenciales: comunicación, simplicidad, retroalimentación, coraje y respeto (Veracruzana, 2012).

Xp genera entregables funcionales a través de ciclos cortos de desarrollo conocidos como iteraciones, ejecutados en base a historias de usuarios: tarjetas en las que el cliente hace una breve descripción de las características que debe poseer el sistema, siendo requisitos funcionales como no funcionales (Grau, 2016). Para añadir o modificar características en las iteraciones las historias de usuario reciben un tratamiento flexible y dinámico, permitiendo ser implementadas en unas semanas por los programadores. Cada historia de usuario debe contener datos como: código de historia, usuario, nombre de historia, prioridad técnica y descripción de los requerimientos que dependiendo de la complejidad del sistema se basara su granularidad. Cabe mencionar que no existe una plantilla definida para las historias de usuario por lo que es posible usar la que más se adapte a las necesidades.

Esta metodología consta de 4 fases fundamentales que se ejecutan de manera paralela en cada una de las iteraciones y que a su vez utilizan un conjunto de reglas y prácticas específicas que deben ser ejecutadas de manera disciplinada para la obtención de buenos resultados (Letelier & Penadés, 2006). Por lo que sí surge un nuevo requerimiento se crea una nueva historia de usuario y se lo implementa.



**Figura 1.** Fases Extreme Programming (XP)

**Fuente:** (García, 2015)

* **Planificación**: Se establecen las historias de usuario y su prioridad junto con el cliente, generando las iteraciones que debe cumplir el programador, lo que permite estructurar un plan de entrega que será presentado en la siguiente reunión.
* **Diseño**: Se busca implementar soluciones de diseños simples, de manera que se realizan prototipos del sistema que una vez aceptados por el cliente y tras pasar por varios rediseños se codifican. Adicional, se estructura el diseño de la base de datos y el diagrama de clases.
* **Codificación**: El equipo desarrollador inicia con la codificación del sistema en base a estándares establecidos, lo que mantiene un código consistente, escalable y de fácil comprensión.
* **Pruebas**: Se crean y ejecutan pruebas unitarias por parte de los desarrolladores para asegurar la calidad del software, así como las pruebas de aceptación en conjunto con los clientes, para garantizar que el sistema cumpla con los requerimientos expuestos en las historias de usuario.

## **Análisis y levantamiento de requerimientos**

Los requerimientos para implementar el sistema fueron levantados a través de las historias de usuario junto con el cliente, según lo determinado en la metodología XP, por lo que se realizaron varias reuniones con personal docente, administrativo de la ESFOT y estudiantes de ET, en las cuales se especificaron funcionalidades principales y objetivos del sistema, identificando los requerimientos funcionales, no funcionales y de interfaces para cada una de las iteraciones, así como los datos correspondientes para la estructura lógica de base de datos. Además, se hizo un análisis de los procesos manuales realizados en el control de sílabos y gestión de proyectores para ser automatizados de forma eficiente en el desarrollo del sistema.

El formato que se estableció para las historias de usuario consta de elementos descritos en la siguiente tabla:

**Tabla 1.** Modelo de historia de usuario

|  |  |
| --- | --- |
| **Historia de Usuario** | |
| **Código:** HU-01-01 | **Usuario:** (Administrador / Docente) |
| **Nombre historia:** | |
| **Prioridad:** (Alta / Media / Baja) | **Estimación:** 48 |
| **Descripción:** | |
| **Observaciones:** | |

**Elaborado por:** Jessica Carrillo

1. **Código:** Identificador único de la historia de usuario, constituida por las siglas “HU”, numero de iteración y número de historia.
2. **Usuario:**  Personas que proporcionan los requisitos del sistema para generar las historias de usuario, identificado a través de un rol.
3. **Nombre historia:** Nombre de la historia de usuario formado por una descripción corta de cada uno de los requerimientos.
4. **Prioridad:** Se presenta el grado de prioridad de las historias de usuario, el cual consta de 3 niveles: definiendo a 1 como alta, 2 media y 3 baja.
5. **Estimación:** Total de horas estimadas para el desarrollo de la historia de usuario
6. **Descripción:** Detalle de las acciones a realizar para cumplir con el requerimiento.
7. **Observaciones (Opcional):** Explicación de procesos que se deben tomar en cuenta al realizar alguna acción en particular.

El proyecto se dividió en 4 iteraciones permitiendo identificar un total de 33 historias de usuario, descritas en el ***anexo A***, es decir, se generaron cuatro entregas con partes completamente funcionales del sistema.

**Primera iteración**

En la primera iteración a través de la entrevista realizada con docentes de la ESFOT, se expusieron una serie de requerimientos referentes a los módulos: inicio de sesión normal (Autenticación usuario/contraseña) y biométrico (Autenticación huella dactilar), administración de carreras, materias, períodos académicos, usuarios, sílabos, proyectores, reportes, reserva/devolución proyectores/kits y registro de tema de sílabo, notificaciones los cuales fueron distribuidos entre las 4 iteraciones para su desarrollo y mejor organización, realizando entregas funcionales probadas directamente por el cliente. Para la presente iteración se definieron las historias de usuario resumidas y descritas en la Tabla 2:

**Tabla 2.** Resumen de las historias de usuario de la primera iteración

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Código** | **Título** | **Historia de usuario** | **Usuario** | **Prioridad** |
| HU-01-01 | Módulo inicio de sesión normal | Inicio sesión normal | Administrador | 1 |
| HU-01-02 | Módulo administración de carreras | Visualizar carreras | Administrador | 1 |
| HU-01-03 | Registrar nuevas carreras | Administrador | 1 |
| HU-01-04 | Modificar carreras | Administrador | 2 |
| HU-01-05 | Eliminar carreras | Administrador | 1 |
| HU-01-06 | Módulo administración de materias | Visualizar materias | Administrador | 1 |
| HU-01-07 | Registrar nuevas materias | Administrador | 1 |
| HU-01-08 | Modificar materias | Administrador | 2 |
| HU-01-09 | Eliminar materias | Administrador | 1 |
| HU-01-10 | Módulo administración de períodos académicos | Visualizar períodos académicos | Administrador | 1 |
| HU-01-11 | Registrar períodos académicos | Administrador | 1 |
| HU-01-12 | Modificar períodos académicos | Administrador | 2 |
| HU-01-13 | Eliminar períodos académicos | Administrador | 1 |

**Elaborado por:** Jessica Carrillo

Además, entre los requerimientos iniciales se identificaron los roles de usuario del sistema, descritos a continuación en la Tabla 3.

**Tabla 3.** Roles de usuario del sistema

|  |  |
| --- | --- |
| **Rol** | **Descripción** |
| Super Administrador | Acceso total al sistema, responsable de la creación de roles de usuario, así como de un usuario administrador |
| Administrador | Encargado de la administración de carreras, materias, períodos académicos, usuarios (Administradores/ Docentes), sílabos y proyectores |
| Docente | Puede registrar tema de clases de acuerdo al sílabo respectivo, así como reservar y devolver proyectores o kits |

**Elaborado por:** Jessica Carrillo

**Segunda iteración**

A la segunda iteración se le asignó el desarrollo de los módulos administración de usuarios y sílabos, por lo que se crearon sus respectivas historias de usuario, resumidas en la Tabla 4.

**Tabla 4.** Resumen de las historias de usuario de la segunda iteración

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Código** | **Título** | **Historia de usuario** | **Usuario** | **Prioridad** |
| HU-02-01 | Módulo administración de usuarios | Visualizar usuarios | Administrador | 1 |
| HU-02-02 | Registrar nuevos usuarios | Administrador | 1 |
| HU-02-03 | Modificar usuarios | Administrador | 1 |
| HU-02-04 | Habilitar/Inhabilitar usuarios | Administrador | 2 |
| HU-02-05 | Módulo administración de sílabo | Visualizar sílabo | Administrador | 1 |
| HU-02-06 | Registrar nuevo sílabo | Administrador | 1 |
| HU-02-07 | Eliminar sílabo | Administrador | 2 |

**Elaborado por:** Jessica Carrillo

**Tercera iteración**

A la tercera iteración se incluyó un nuevo requerimiento por parte del cliente a las historias de usuario, el cual permitirá a los administradores abrir el locker de un proyector/kit iniciando con la integración a la parte física del sistema, la cual se encuentra a cargo de estudiantes de ETE.

Las historias de usuario se encuentran resumidas en la presente Tabla 5:

**Tabla 5.** Resumen de las historias de usuario de la tercera iteración

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Código** | **Título** | **Historia de usuario** | **Usuario** | **Prioridad** |
| HU-03-01 | Módulo administración de proyectores/kits | Visualizar proyectores/kits | Administrador | 1 |
| HU-03-02 | Registrar proyectores/kits | Administrador | 1 |
| HU-03-03 | Modificar proyectores/kits | Administrador | 1 |
| HU-03-04 | Habilitar/Inhabilitar proyectores/kits | Administrador | 2 |
| HU-03-05 | Abrir locker de proyector/kit | Administrador | 1 |
| HU-03-06 | Módulo autenticación biométrico | Inicio sesión biométrico | Administrador /Docente | 1 |
| HU-03-07 | Módulo registro clase | Registrar tema de sílabo | Docente | 1 |

**Elaborado por:** Jessica Carrillo

**Cuarta iteración**

Las historias de usuario asignadas a la cuarta iteración se resumen en la Tabla 6, a las cuales se realizaron varias modificaciones, así como la adición de un nuevo requerimiento, esto tras la última reunión entre el cliente y desarrollador.

**Tabla 6.** Resumen de las historias de usuario de la cuarta iteración

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Código** | **Título** | **Historia de usuario** | **Usuario** | **Prioridad** |
| HU-04-01 | Módulo reserva proyector/kit | Reservar proyector | Administrador /Docente | 1 |
| HU-04-02 | Módulo devolver proyector/kit | Devolver proyector | Administrador /Docente | 1 |
| HU-04-03 | Notificaciones | Notificación de no registro de tema | Docente | 1 |
| HU-04-04 | Notificación no devolución de proyector/kit | Docente | 1 |
| HU-04-05 | Módulo reportes | Reporte de registro de temas de sílabo | Administrador | 2 |
| HU-04-06 | Reporte de préstamo/devolución proyectores/kits | Administrador | 2 |

**Elaborado por:** Jessica Carrillo

**Análisis de requerimientos**

Con el fin de disminuir inconvenientes en el desarrollo del sistema y en base a las necesidades mencionadas en las historias de usuario, se determinan los requerimientos funcionales y no funcionales.

**Requerimientos funcionales**

Describen la interacción entre el sistema y su entorno, es decir, la manera en que este se comporta sistema tras cumplirse o ejecutarse ciertas actividades o condiciones (funciones específicas) (Blog, 2018).

Los requerimientos funcionales presentados en la Tabla 7, tienen asignados un código que consta de las siglas RF (Requerimiento funcional), número de requerimiento y número de iteración.

**Tabla 7.** Requerimientos funcionales del sistema

|  |  |
| --- | --- |
| **Código** | **Funcionalidad** |
| RF-01-01 | El sistema permitirá al usuario administrador autenticarse a través de las credenciales (Email y contraseña) e interfaz correspondiente, teniendo en cuenta que este maneja dos tipos de inicio de sesión (Inicio de sesión normal) |
| RF-02-01 | El sistema permitirá al usuario administrador el registro, modificación y eliminación de carreras (Inicio de sesión normal) |
| RF-03-01 | El sistema permitirá al usuario administrador el registro, modificación y eliminación de materias (Inicio de sesión normal) |
| RF-04-01 | El sistema permitirá al usuario administrador el registro, modificación y eliminación de períodos académicos (Inicio de sesión normal) |
| RF-05-02 | El sistema permitirá al usuario administrador el registro, modificación, habilitar e inhabilitar usuarios (Inicio de sesión normal) |
| RF-06-02 | El sistema permitirá al usuario administrador importar y eliminar sílabos correspondientes a cada docente (Inicio de sesión normal) |
| RF-07-03 | El sistema permitirá al usuario administrador el registro, modificación, habilitar e inhabilitar proyectores/kits (Inicio de sesión normal) |
| RF-08-04 | El sistema permitirá al usuario administrador generar reportes relacionados con el préstamo/devolución de proyectores/kits y registro de temas de docentes (Inicio de sesión normal) |
| RF-09-03 | El sistema permitirá al usuario docente y administrador autenticarse a través su huella dactilar por medio de un dispositivo biométrico e interfaz correspondiente, teniendo en cuenta que este maneja dos tipos de inicio de sesión (Inicio de sesión biométrico).  Cabe mencionar que si en un lapso de 20 segundos no hay interacción en el sistema la sesión se cerrara automáticamente |
| RF-12-03 | El sistema permitirá al usuario docente registrar los temas correspondientes a los sílabos cargados previamente y en base a su cumplimiento (Inicio de sesión biométrico). |
| RF-10-04 | El sistema permitirá al usuario docente y administrador reservar proyectores o kits.  El usuario docente solo podrá reservar un proyector o kit a la vez  El usuario administrador podrá reservar las veces que sea necesarias siempre y cuando los dispositivos se encuentren disponibles (Inicio de sesión biométrico). |
| RF-11-04 | El sistema permitirá al usuario docente y administrador devolver proyectores o kits (Inicio de sesión biométrico). |
| RF-13-04 | El sistema se encargará de enviar notificaciones de tipo email ante dos situaciones: no se cumpla con la devolución de proyectores/kit o no exista registro de los temas de clase por parte de docentes |
| RF-14-03 | El sistema permitirá al usuario administrador modificar, habilitar e inhabilitar proyectores o kits, así como abrir las puertas correspondientes a los dispositivos que se desee tener acceso para su mantenimiento. (Inicio de sesión biométrico). |

**Elaborado por:** Jessica Carrillo

**Requerimientos no funcionales**

Describen las propiedades del sistema: rendimiento, seguridad y disponibilidad, así como sus restricciones: capacidad de dispositivos de entrada/salida y representación de datos.

Los requerimientos funcionales descritos en la Tabla 8 tienen asignados un código que consta de las siglas RNF (Requerimiento no funcional) y el número de requerimiento,

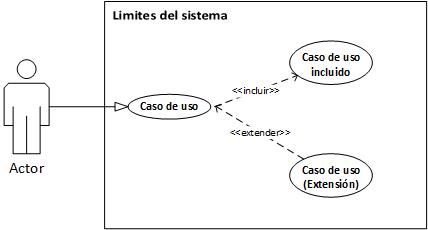
**Tabla 8.** Requerimientos no funcionales del sistema

|  |  |
| --- | --- |
| **Código** | **Funcionalidad** |
| RNF-01 | Las interfaces del sistema deberán ser amigables e intuitivas |
| RNF-02 | El tiempo de respuesta de los módulos del sistema, y más aún de la autenticación por medio de la huella digital, reservas y devoluciones de los diferentes dispositivos deberá ser rápido |
| RNF-03 | El sistema emitirá los correspondientes mensajes de error e informativos hacia los usuarios correspondientes |
| RNF-04 | La contraseña de los usuarios deberá ser cifrada antes de ser registrada en la base de datos |
| RNF-05 | Todos los usuarios creados tendrán un id único en relación con el biométrico |

**Elaborado por**: Jessica Carrillo

**Diagrama de Casos de uso**

Los diagramas de casos de uso permiten especificar la interacción entre usuario final y el sistema de manera gráfica, lo que ayuda a definir la arquitectura, pautas de diseño y ejecución de pruebas funcionales (Pérez, 2013). Esto tras el uso de elementos como:



**Figura 2.** Elementos de casos de uso

**Elaborado por:** Jessica Carrillo

* **Actor**: algo o alguien (rol de usuario) que interactúa con el sistema
* **Sistema**: Representado por un rectángulo que limita la función del sistema y contiene los casos de uso
* **Casos de uso**: Representado por un ovalo que contiene las funciones del sistema
* **Relaciones:** Representado por líneas o flechas muestran la relación entre actores y casos de uso, se tiene 4 tipos de relaciones de comportamiento comunica, incluye, extiende y generaliza, de los cuales los más utilizados son:
  + **Incluir**: Es necesario de otro caso de uso para cumplir una tarea o acción
  + **Extender**: El caso de uso posee varias opciones o alternativas

Los diagramas de casos de uso, ver ***anexo B***, se elaboraron en base a los requerimientos funcionales del sistema previamente identificados, permitiendo tener una visión más simple del funcionamiento tras la interacción con los usuarios, la estructura y flujo de información del sistema.

## **Diseño de la base de datos y arquitectura del sistema**

En esta fase se estableció la arquitectura del sistema, modelo de base de datos y los prototipos de las pantallas principales, para lo que se hizo uso de las siguientes herramientas:

**Power Designer**

Herramienta para visualizar, analizar, diseñar y manipular de manera sencilla metadatos ya que combinas técnicas de modelamiento, lo que facilita la construcción de una base de datos sólida generando una arquitectura de información eficaz para aplicaciones de alto rendimiento. Tiene la capacidad de acoplarse a los cambios sin afectar su estructura (Ecured, 2019).

**MySQL**

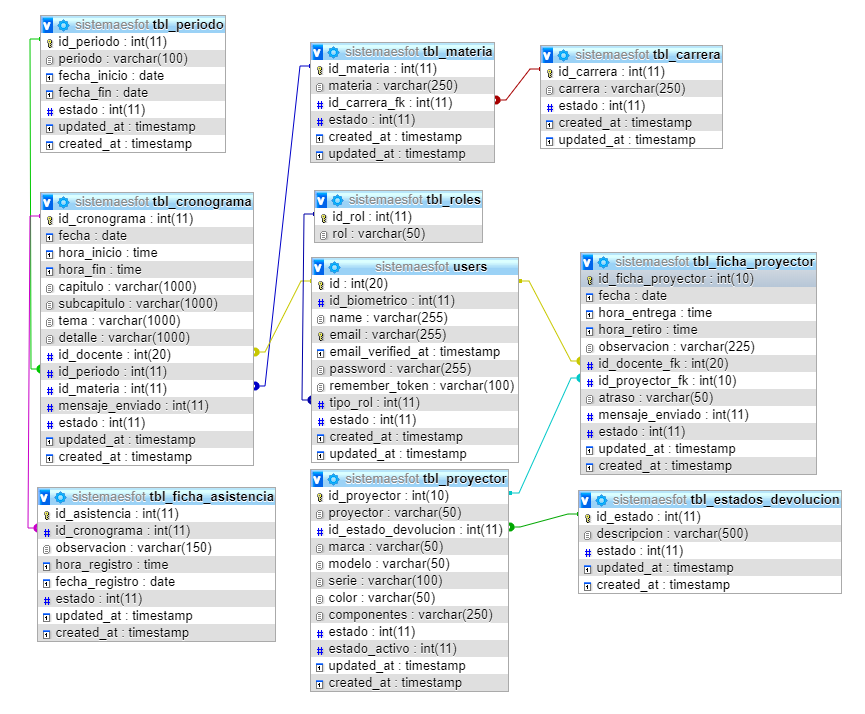
Sistema de administración de bases de datos de código abierto, que permite almacenar y manipular datos en bases relacionales, considerada una de las más populares en el desarrollo web se ejecuta en cualquier sistema operativo, así como, junto con otros programas de entorno de programación Wamp, Lamp o Xampp (Navia, 2018).

**Balsamiq Mockups**

Herramienta wireframing que permite diseñar prototipos y maquetas de las interfaces de usuario para webs o aplicaciones móviles, de manera que digitaliza los distintos escenarios y funcionalidades incluyendo la estructura de navegación (Balsamiq, 2020). Lo que servirá de base y guía para el desarrollo del frontend del sistema.

**Modelo de base de datos**

El diseño de la estructura de base de datos se ha elaborado utilizando Power Designer tras las reuniones realizadas en las iteraciones junto con el cliente, lo que ha permitido identificar datos relevantes del sistema y sus relaciones. Es necesario mencionar que una base de datos sólida y correctamente estructurada permite desarrollar un sistema muy eficiente. De manera que se ha generado el modelo entidad relación de la base de datos del sistema, **ver anexo C**, con dicha herramienta, permitiendo identificar las principales tablas y atributos para luego ser implementadas a través de MySQL.



**Figura 3.** Estructura de la base de datos del sistema

**Elaborado por:** Jessica Carrillo

La descripción de las principales tablas y campos de la estructura de la base de datos de la Figura 3, se encuentran en el ***anexo D.***

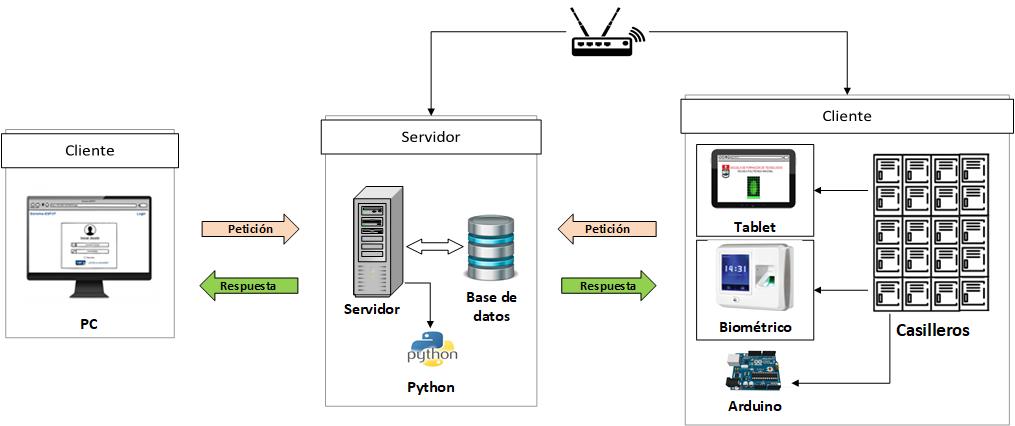
**Diseño de los prototipos del sistema**

En base a los requerimientos funcionales del sistema y con la ayuda de Balsamiq Mockup se diseñaron los prototipos que simulan las interfaces graficas del sistema, esenciales para determinar la estructura o pilar de codificación y que a su vez son presentados al cliente para su previa aprobación. Todos los prototipos para más detalle se encuentran en el ***anexo E***. a continuación, se presentan las interfaces principales del sistema.

**Diseño de la arquitectura del sistema**

La arquitectura del sistema se basa en un modelo Cliente-Servidor constituido por dos elementos claramente diferenciables: el servidor en el que se alojan el sistema, base de datos, script python para la comunicación con arduino y biométrico, servicios que son consumidos por el cliente a través de los métodos de petición HTTP por la URL correspondiente, es decir, interactúan a través del envío y recepción de mensajes. En relación con el elemento cliente, se han determinado dos tipos:

* **Cliente 1:** Acceso mediante PC, autenticación por credenciales (email, contraseña), control funciones administrativas
* **Cliente 2:** Acceso mediante Tablet y uso de dispositivo biométrico para la autenticación (huella digital/id-biométrico), control de casilleros/lockers



**Figura 4**. Arquitectura del sistema

**Elaborado por**: Jessica Carrillo

# **RESULTADOS Y DISCUSIÓN**

## **Implementación del sistema**

**Herramientas de desarrollo**

**Laravel 5.8**

Framework que permite desarrollar en PHP de forma simple y segura, trabaja bajo el patrón de diseño MVC (Modelo-Vista-Controlador) por lo que separa la parte lógica, datos e interfaces de usuario facilitando su mantenimiento, además integra una serie de funciones y herramientas sumamente útiles en el desarrollo de aplicaciones web (Nube, 2019).

Por estas y otras características esta herramienta fue seleccionada para la construcción del sistema ya que ayudo a mantener un código organizado, y al ser de fácil uso se optimizo el tiempo en el desarrollo

**HTML5**

Lenguaje de marcas de hipertexto de fácil interpretación por los navegadores (Jiménez, 2019). Permitió definir la arquitectura y contenido de las páginas del sistema a través del uso de etiquetas que facilitaron la administración de contenido multimedia y texto, por lo que se mantuvo una correcta organización de cada uno de estos elementos dentro de las interfaces

**Bootstrap V4.4.1**

Framework de código abierto basado en HTML5, CSS y JavaScript que ayudó en el desarrollo de las interfaces del sistema permitiendo que se adapten a cualquier tipo de dispositivos, ya cuentan con un diseño responsive que engloba una amplia gama de herramientas y funciones, lo que mejora la experiencia de usuario en cuanto a la visualización (team, 2019).

**JavaScript**

Lenguaje de programación tipo Cliente/Servidor, el cual fue utilizado para crear contenido dinámico, validar formularios, manejar y controlar eventos en las páginas web del sistema. Se integra fácilmente a documentos HTML lo que hace posible el manejo de sus elementos de manera oculta al usuario final (MDN, 2019).

**Jquery**

Biblioteca de código abierto de JavaScript con capacidad multiplataforma ofrece una variedad de funciones y métodos comprimidos en muy pocas líneas de código para desarrollar sitios interactivos (Domingues, 2018). Esta herramienta facilitó la manipulación de elementos del DOM (textos, imágenes, enlaces) de las interfaces del sistema, a través de Ajax, realizar peticiones al servidor actualizando únicamente una porción de la página y no la página entera evitando recargas innecesarias y mediante el plugin datatable darle una mejor apariencia a las tablas que presentan la información e incluyendo componentes de paginación, búsqueda y filtrado.

**Sweet Alert**

Plugin Jquery usado para mejorar el aspecto de los mensajes, notificaciones o alertas lanzadas al realizarse alguna acción por parte de los usuarios en el sistema. Integra una serie de parámetros que permiten configurar el plugin de manera sencilla adaptándose a cualquier tipo de dispositivos (Edwards, 2019).

**Librería pyzk 0.9**

Biblioteca no oficial del dispositivo biométrico de huellas digitales zkzteco (zksoftware) (pyzk, 2019). Con la cual fue posible crear una API de comunicación segura en python entre el sistema y el dispositivo biométrico permitiendo ingresar, extraer y autenticar usuarios.

**Phyton 3.6**

Lenguaje de programación interpretado de código abierto, multiplataforma y multiparadigma que contiene una gran cantidad de librerías y funciones para el desarrollo de todo tipo de programas, como la automatización de procesos y ejecución de tareas en entornos tanto cliente como servidor (Robledano, 2019).

Esta herramienta permitió establecer la comunicación serial necesaria para controlar los microcontroladores Arduino a través del sistema de manera sencilla y segura

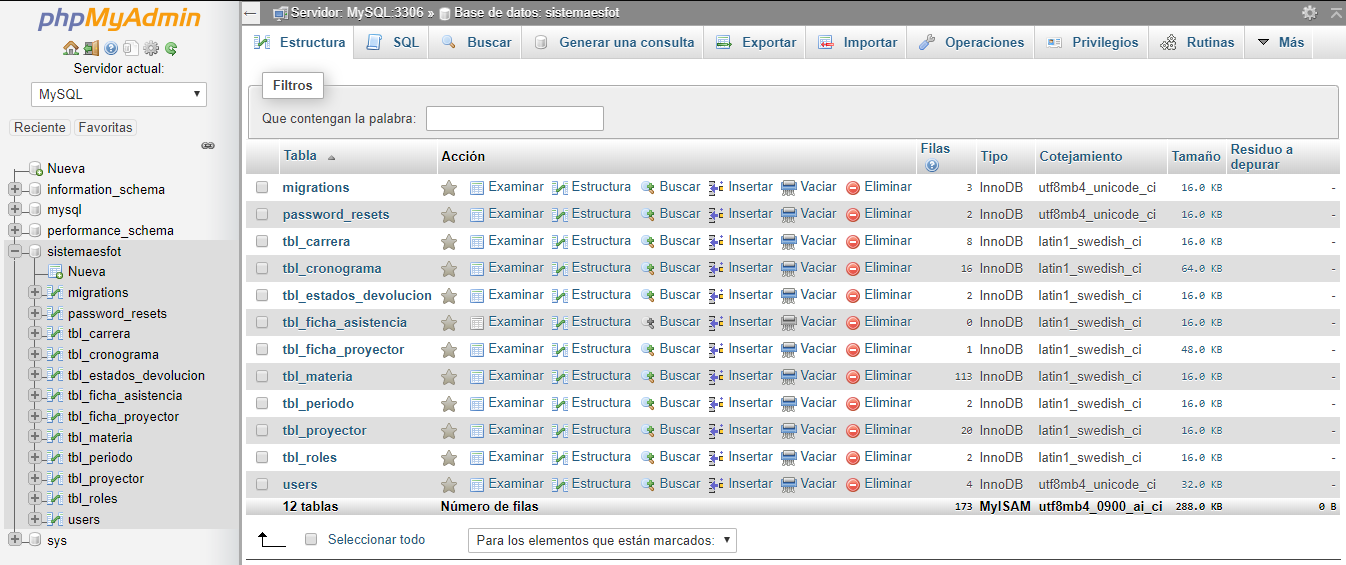
**Iteración 1**

Para la primera iteración ser realizaron las configuraciones correspondientes al entorno de desarrollo que involucraba la creación del proyecto (Laravel) y la base de datos (MySQL), seguido de la implementación del módulo inicio de sesión (normal) o autenticación por credenciales (correo electrónico/ contraseña) y los módulos administrativos de carreras, materias y períodos académicos

Las interfaces de todos los módulos se crearon en base a los mockups o prototipos previamente realizados junto con sus respectivos mensajes de validación.

**Creación de la base de datos MySQL**

La base de datos se creó y alojo en el servidor virtual WAMPSERVER, bajo el servicio MySQL mediante el entorno phpMyAdmin que a través de su sencilla interfaz facilito su administración, cabe mencionar que WAMPP además integra Apache y PHP que en combinación crean un ambiente web completo permitiendo desarrollar, alojar y subir al internet aplicaciones web.



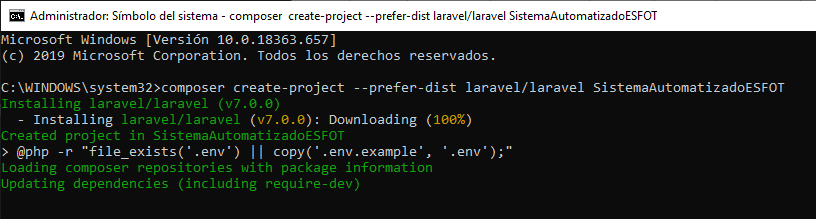
**Figura 5.** Base de datos en PhpMyAdmin

**Elaborado por**: Jessica Carrillo

**Creación del proyecto en Laravel**

Para el desarrollo del sistema se utilizó el Framework Laravel versión 5.8, previamente fue necesaria la instalación de composer que de acuerdo con la documentación de Laravel <https://laravel.com/docs/5.8> es necesario para gestionar dependencias y librerías.

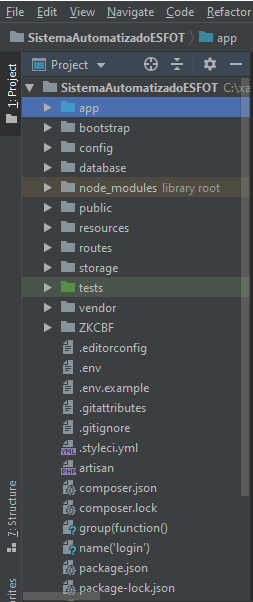
Para la creación del proyecto se ejecutó el comando composer create-project --prefer-dist laravel/laravel SistemaAutomatizadoESFOT en el CMD



**Figura 6.** Creación del proyecto en Laravel

**Elaborado por:** Jessica Carrillo

Finalizado el proceso de creación y mediante el editor de código PHPStorm se visualizará la arquitectura de directorios y se codificará el sistema

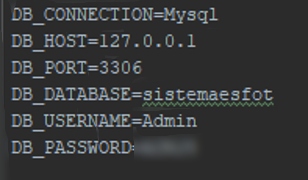


**Figura 7.** Arquitectura de archivos del proyecto creado

**Elaborado por:** Jessica Carrillo

**Conexión del proyecto con la Base de Datos**

La conexión de base de datos se realizó mediante la edición del archivo ENV de la estructura de archivos del sistema, estableciendo parámetros tales como: usuario, contraseña, host y puerto



**Figura 8.** Parámetros de conexión a la base de datos

**Elaborado por:** Jessica Carrillo

**Implementación de la aplicación web**

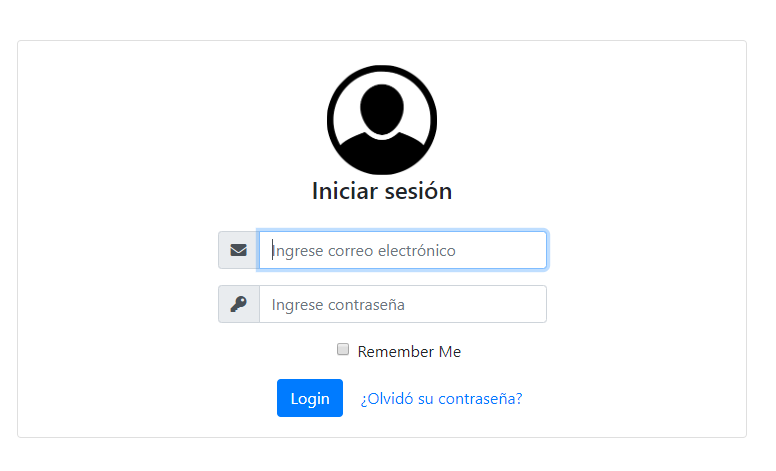
El proyecto fue desarrollado bajo la modalidad de aplicación web, lo que evita que usuarios descarguen alguna apk de acceso en sus dispositivos ocupando almacenamiento, en su lugar se hace uso de una red local y un navegador sea de la Tablet o computador para manipularlo, y gracias a la característica responsive este se acopla a cualquier dispositivo.

La conexión de red que utiliza el sistema es independiente e interna de las redes EPN, lo que garantiza su disponibilidad y seguridad.

El sistema maneja dos tipos de autenticación que dan acceso a módulos específicos: por credenciales (correo electrónico-contraseña) control de la parte administrativa y dispositivo biométrico (Id-biométrico-huella dactilar) control de casilleros y gestión de clases, por lo que su uso está destinado únicamente a personal administrativo y docentes de la ESFOT

**Módulo inicio de sesión (credenciales)**

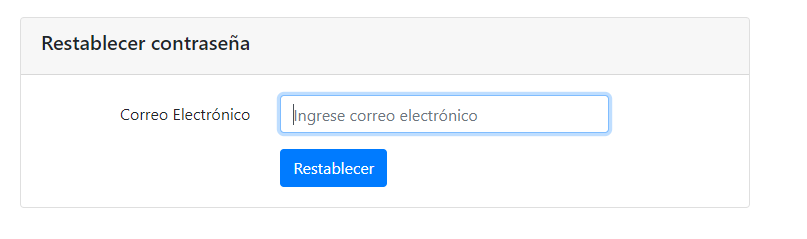
En primer lugar, se desarrolló el módulo inicio de sesión por credenciales con sus respectivas validaciones, como se visualiza en la Figura 9 y el anexo F, el cual permite el acceso al sistema únicamente a usuarios administradores al ingresar correctamente su email y contraseña.



**Figura 9.** Módulo inicio de sesión por credenciales

**Elaborado por:** Jessica Carrillo

El administrador, además, tiene la opción de restaurar la contraseña o cambiarla, una vez que este ingrese el correo electrónico en el módulo correspondiente se le enviará un mensaje junto con un link que corresponde al formulario para validar la nueva contraseña, como se visualiza en las Figuras 10 y 11.



**Figura 10.** Módulo reestablecer contraseña

**Elaborado por:** Jessica Carrillo

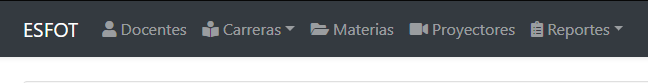


**Figura 11.** Email para reestablecer contraseña

**Elaborado por:** Jessica Carrillo

**Módulo administración de carreras, materias, períodos académicos**

El administrador al ingresar al sistema tiene la capacidad de manejar varios módulos, ver Figura 12 que cumplen una serie de actividades relevantes, para esta la primera iteración se desarrollaron los relacionados con la administración de carreras, materias y periodos académicos. En cada una de las secciones se manejan listas para presentar información importante, las cuales a su vez implementan métodos CRUD para facilitar su uso, los formularios para el registro y edición de datos contienen campos específicos y mensajes de validación.



**Figura 12.** Barra con Menú de opciones (Administrador)

**Elaborado por:** Jessica Carrillo

Se hace uso de del botón tipo switch para habilitar o inhabilitar los registros, ya que no es indispensable que estos sean eliminados de la base de datos por lo que se manejan estados entre 0 y 1.

Además, se cuenta con filtros y un buscador que permiten manejar de una manera sencilla los registros presentados en las listas.

Las interfaces correspondientes a dichos módulos pueden ser visualizados a gran detalle en el ***anexo F***

**Iteración 2**

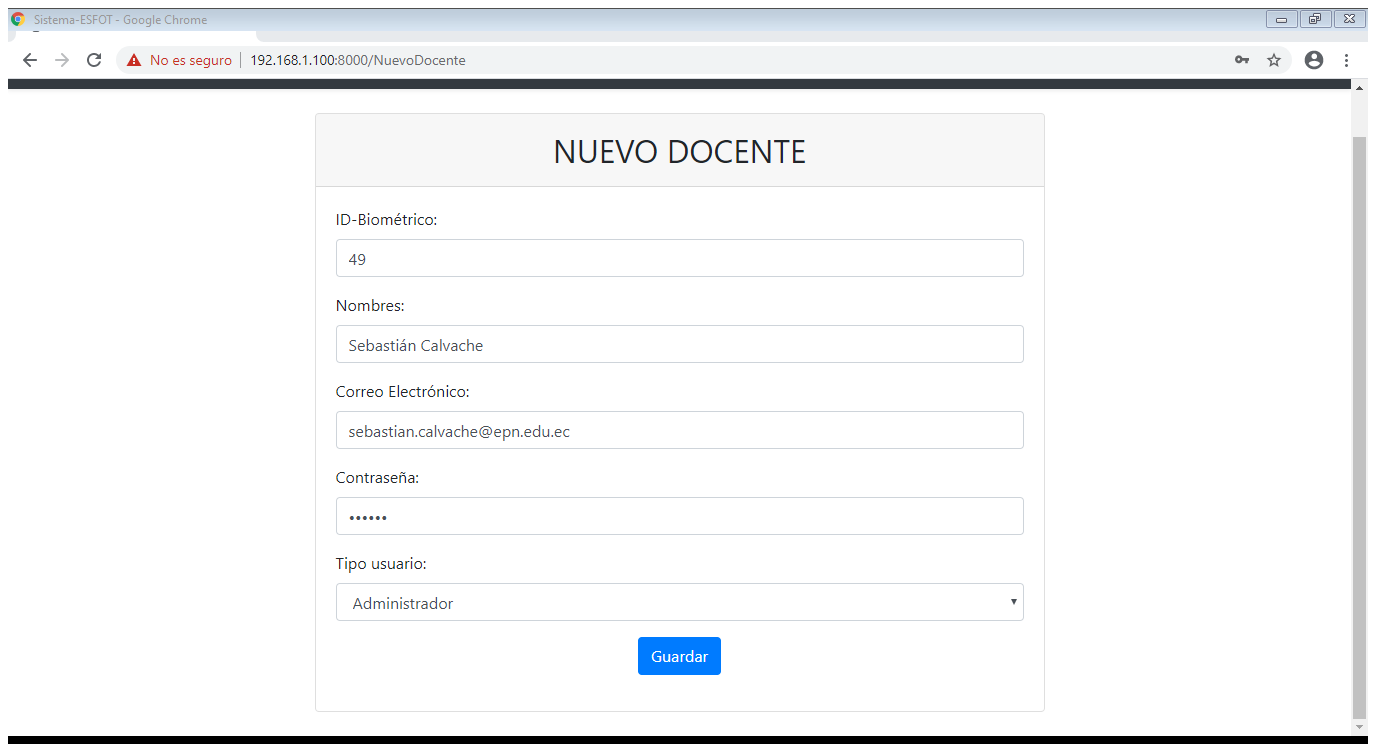
En la segunda iteración se realizó el módulo para la administración de usuarios y sílabos, implementando en primer lugar las interfaces gráficas diseñadas para después desarrollar el código respectivo.

**Módulo usuario**

El ítem usuarios hace referencia a todos los usuarios que tendrán interacción con el sistema ya sea para el inicio de sesión por credenciales o dispositivo biométrico. Todos los registros serán presentados a manera de lista y con acciones de editar y eliminar como el de los módulos anteriores

El formulario de registro de usuarios debe ser llenado en su totalidad ya que son campos obligatorios, es necesario asignar un identificador único al campo id-biométrico, ya que este mantendrá la relación con la huella digital cargada en el dispositivo biométrico y por ende tener control de los casilleros

Al momento de presionar el botón guardar el registro será insertado tanto en la base de datos como en el dispositivo biométrico y una vez en el dispositivo se procederá a la toma de huellas correspondientes, ver Figura 13 y 14.



**Figura 13.** Registro de usuario al sistema

**Elaborado por:** Jessica Carrillo



**Figura 14.** Registro de usuario al dispositivo biométrico

**Módulo sílabos**

Todos los usuarios que cuenten con el rol Docente tendrán un botón denominado Sílabo, que permitirá cargar en formato xls los sílabos correspondientes a cada docente, seleccionado antes el período académico, carrera y materia para una mejor organización y por ende visualización, como se visualiza en la Figura 14, teniendo en cuenta que dicha información será utilizada en otros módulos. Se tendrá la posibilidad de eliminar ya sea de manera masiva o individual los temas cargados



**Figura 15.** Módulo importar sílabo

**Elaborado por:** Jessica Carrillo

**Iteración 3**

En la tercera iteración se realizaron los módulos correspondientes a la administración de proyectores, autenticación a través del dispositivo biométrico y registro de los temas de clase.

**Módulo administración de proyectores/kits**

El módulo proyector cuenta únicamente con el registro de 20 dispositivos de los cuales 16 son proyectores y 4 son kits, lo que inhabilita la función de nuevos ingresos. Se mantiene el formato de los módulos posteriores, ya que los datos son visualizados a través de una lista que incluye una sección de acciones (editar, habilitar/inhabilitar) para su posible manipulación.

El formulario de edición permite al administrador seleccionar de una manera sencilla los componentes que vienen junto con el dispositivo ya que estos son presentados a través de checkboxs, ver ***anexo F.***

La función habilitar/inhabilitar debe ser ejecutada únicamente cuando el dispositivo se encuentre en mantenimiento para evitar inconvenientes en su uso y por ende no pueda estar disponible

El buscador integrado en la interfaz permite que el administrador digite cualquier característica de los registros presentados en la tabla o lista y este sea encontrado fácilmente

El módulo administración de proyectores/kits podrá ser controlado al acceder por cualquiera de los dos inicios de sesión a diferencia de que, si el administrador ingresa a través de la autenticación biométrica tendrá la opción de abrir cualquiera de las puertas de los casilleros únicamente para mantenimiento de los dispositivos



**Figura 16.** Módulo de dispositivos de proyección

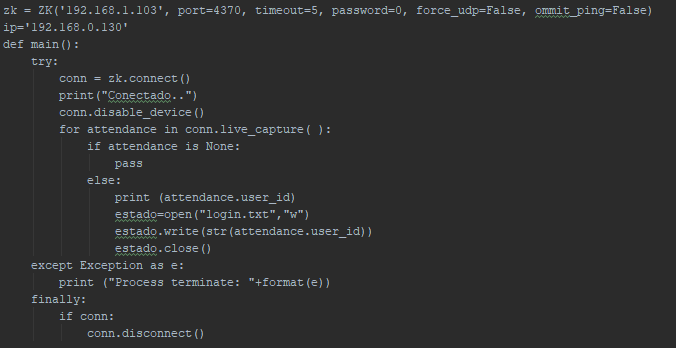
**Elaborado por**: Jessica Carrillo

**Módulo autenticación biométrico**

El inicio de sesión o autenticación biométrica es usado tanto por administradores y docentes de la ESFOT, los cuales se deben identificar ante el dispositivo biométrico colocando su huella digital, una vez esta sea escaneada se extrae el paramétrico único id-biométrico y se relaciona con el almacenado en la base de datos del sistema para posteriormente y de acuerdo al rol asignado ser redirigido al módulo correspondiente, todo el proceso es realizado en cuestión de segundos e imperceptible al usuario, desarrollado tras el uso de la librería ZK y Python para realizar la conexión directa al dispositivo biométrico capturando datos en tiempo real como se muestra en la Figura 17.

Solos los usuarios que este registrados en el sistema y dispositivo biométrico podrán hacer uso del presente módulo.

La sesión caducara a los 20 segundos si no se tiene interacción del usuario garantizando aún más su seguridad, evitando el mal uso del sistema por personas externas.



**Figura 17.** Código de conexión y captura de datos del biométrico (Python)

**Elaborado por:** Jessica Carrillo

**Módulo registro clase**

Al momento en que los docentes se autentiquen a través del dispositivo biométrico visualizaran una pantalla con dos opciones: gestión proyector y gestión clases

El docente al seleccionar la opción gestión clases será dirigido a dicho módulo visualizando el cronograma o sílabo completo de acuerdo al periodo académico vigente, en base al cual se realizará su seguimiento, ya que cada vez que el docente tenga que impartir o imparta su clase debe registrar en el sistema el tema correspondiente. La lista contiene únicamente temas que aún no han sido registrados lo que facilita y hace más sencilla su manipulación.

El docente tiene la opción de registrar un tema bajo el termino permiso, solo cuando las clases choquen con eventos de la ESFOT y se disponga del comunicado oficial por parte de las autoridades.



**Figura 18.** Registro de clases o tema del sílabo

**Elaborado por:** Jessica Carrillo

**Iteración 4**

En la cuarta iteración se desarrollaron los módulos reserva proyector/kit, devolver proyector/kit, reportes y la función automática de notificaciones

**Módulo reservar y devolver proyector/kits**

En este módulo los administradores y docentes autenticados mediante el dispositivo biométrico realizaran la reserva/devolución de los dispositivos de proyección siempre y cuando se encuentren disponibles.

La interfaz consta de un total de 20 botones que representan a cada uno de los dispositivos, como se visualiza en la figura… y en base a su estado de disponibilidad toman un color: verde (disponible), rojo (no disponible), gris(mantenimiento).

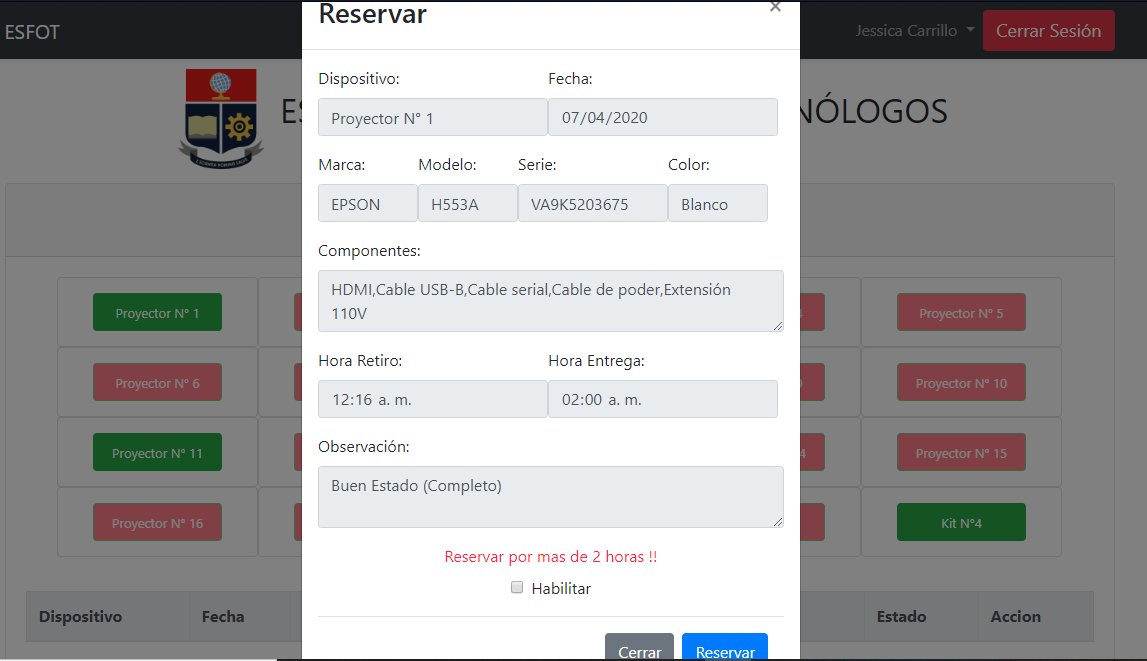


**Figura 19.** Gestión de dispositivos de proyección

**Elaborado por:** Jessica Carrillo

Al momento en que el usuario oprima el botón del proyector seleccionado podrá visualizar un formulario con toda la información del dispositivo, la hora de entrega marcada se generará de forma automática aproximada de 2 horas basada en la hora de retiro, se tiene la opción de extender el tiempo de entrega al seleccionar el checkbox habilitar y colocar la hora manualmente.

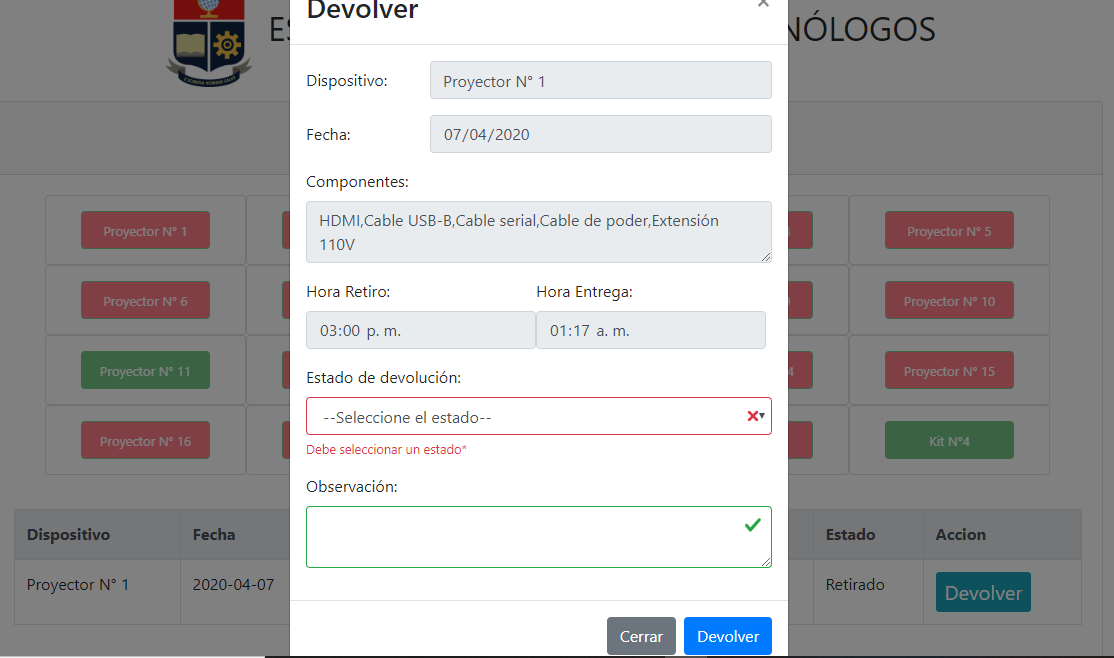
Realizado el registro correctamente el color del botón cambiará a rojo y todos los demás se bloquearán ya que el docente solo tiene la opción de reservar un dispositivo mientras no realice su devolución no podrá tener acceso a otro, mientras que administradores podrán reservar varios dispositivos.



**Figura 20.** Reserva de dispositivo de proyección

**Elaborado por:** Jessica Carrillo

El proceso de devolución del dispositivo es similar a la reserva, ya que al oprimir el botón devolver este mostrará el respectivo formulario de información con excepción de que, el usuario deberá seleccionar uno de los estados de devolución para finalizar con la acción.

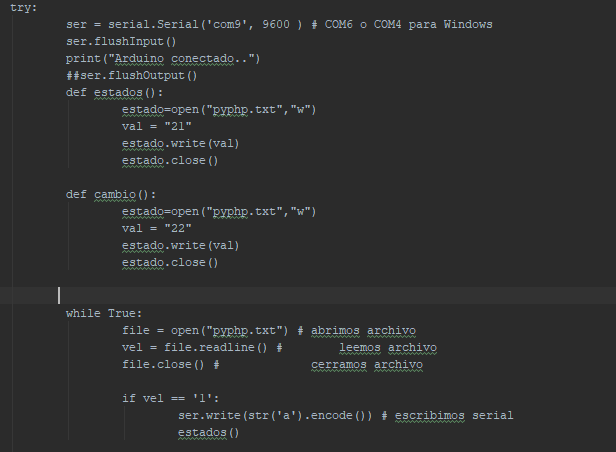


**Figura 21**. Devolución de dispositivo de proyección

**Elaborado por:** Jessica Carrillo

Los mensajes manifestados en el módulo corresponden a: reserva en proceso, reserva correcta del dispositivo, reserva no realizada (dispositivo no retirado), devolución en proceso, devolución realizada, devolución no realizada (dispositivo no entregado).

Para el desarrollo del módulo fue necesario realizar una conexión de tipo serial en Python al Arduino principal que controla los casilleros, de manera que cada una de las puertas puedan ser manipuladas en tiempo real por la interfaz del sistema

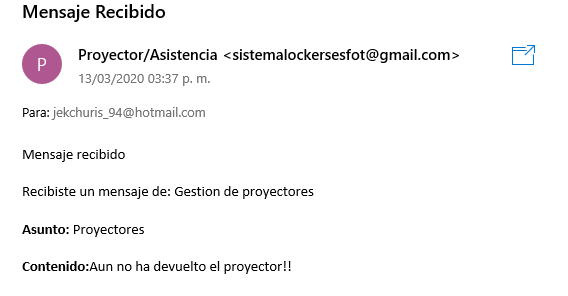


**Figura 22**. Código de conexión serial al Arduino (Python)

**Elaborado por:** Jessica Carrillo

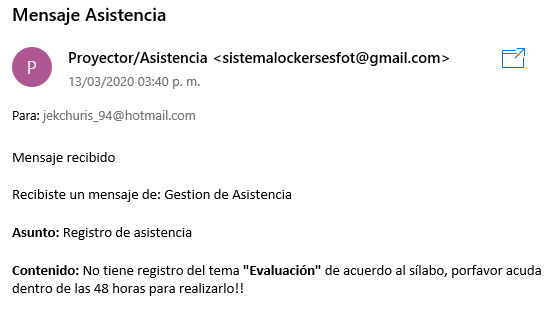
**Notificaciones**

Mensajes de email generados y enviados automáticamente por el sistema tras el uso de las tareas programadas (CRON) de Laravel, dirigidos a docentes que no cumplan con la devolución de los dispositivos de proyección en el tiempo estipulado o no realicen el registro de su tema de clase.



**Figura 23**. Email de no devolución proyector/kit

**Elaborado por:** Jessica Carrillo



**Figura 24.** Email de no registro de tema

**Elaborado por:** Jessica Carrillo

**Modulo reportes**

El usuario administrador tiene la capacidad de generar dos tipos de reportes: proyectores y registros de temas, al ingresar a través del inicio de sesión por credenciales en el módulo reportes como se visualiza en la Figura 25.

Para facilidad de administrador la interfaz tanto del reporte de proyectores y registro de temas hacen uso de filtros de búsqueda como: rango de fechas o cualquier propiedad que sea visible en la tabla de información.

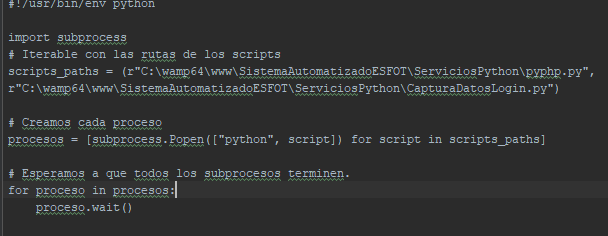
Además, se tiene la opción de exportar los reportes a PDF o EXCEL, al dar clic en el icono que los representa se procede con su inmediata descarga. Es importante mencionar que los reportes de clases deben ser generados y firmados mensualmente por cada uno de los docentes para validar el seguimiento de lo sílabos correspondientes, ver ***anexo E***.



**Figura 25**. Módulo de reporte de clases

**Elaborado por:** Jessica Carrillo

##### Cabe mencionar que los scripts de conexión desarrollados en Python tanto al Arduino como el dispositivo biométrico se ejecutan en segundo plano, siendo transparente a la vista y de manera simultánea, esto tras la ayuda de un tercer script y la librería subprocess que permite ejecutar varios procesos al mismo tiempo sin ningún problema, por lo que cada proceso trabaja de manera independiente sin interferir en el otro, como se muestra en la Figura 26



**Figura 26**. Script para ejecución de dos procesos simultáneamente

**Elaborado por:** Jessica Carrillo

Todas las interfaces implementadas en el sistema se encuentran detalladas en el ***anexo F.***

## **Pruebas del sistema**

Para verificar y garantizar que cada módulo de sistema funcione correctamente y cumpla con los requerimientos establecidos, se efectuaron distintas pruebas, las cuales fueron ejecutadas al final de cada iteración funcional junto con algunos usuarios, lo que disminuyo la cantidad de errores en la prueba general implementada en la entrega final del proyecto. A continuación, se presenta en detalle cada una de las pruebas realizadas al sistema.

**Pruebas de compatibilidad**

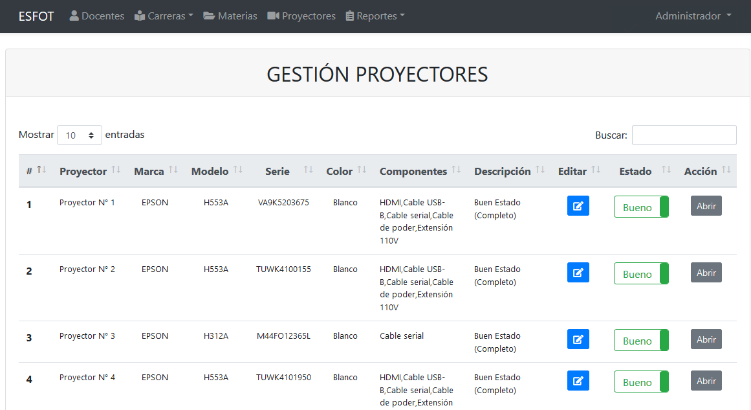
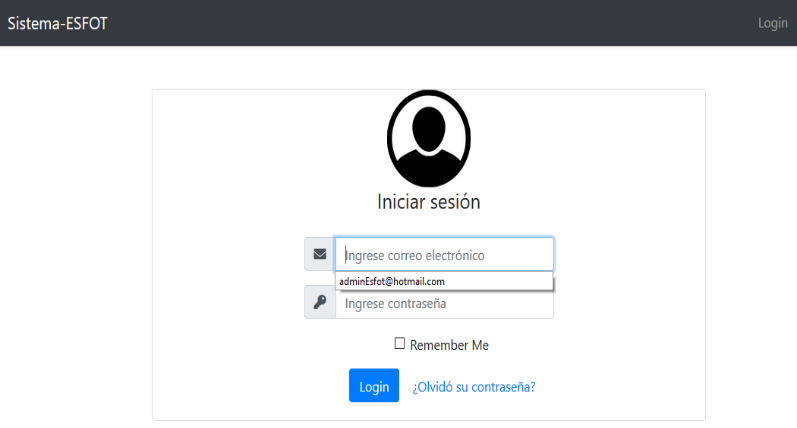
Las pruebas de compatibilidad permitieron probar al sistema web en diferentes entornos de navegación, comprobando su correcto funcionamiento y visualización, haciendo uso de 3 de los navegadores más populares en sus últimas versiones presentados en la Tabla 9

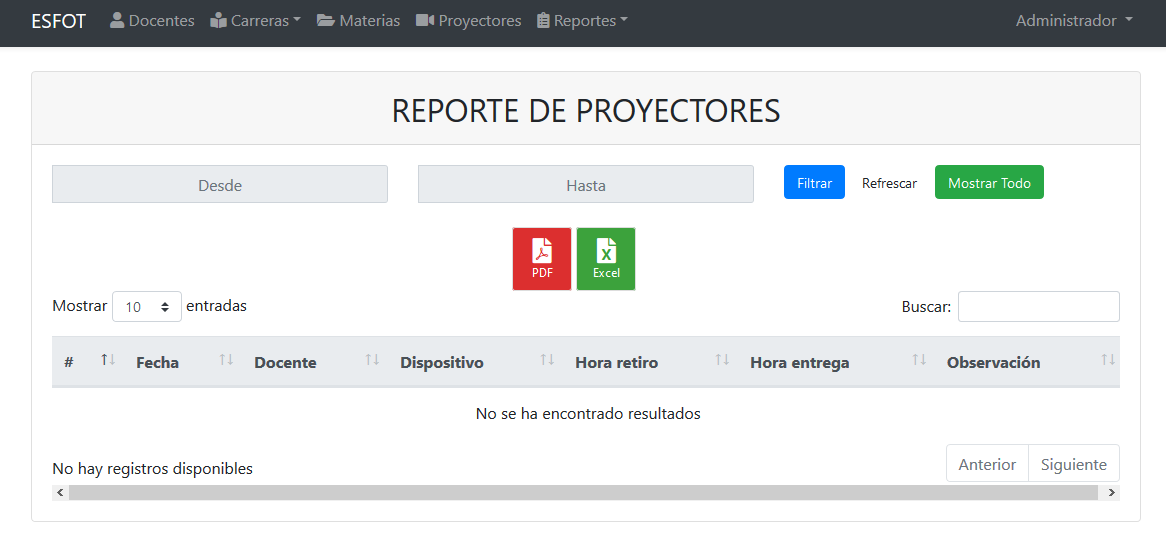
**Tabla 9.** Resultados de las pruebas de compatibilidad

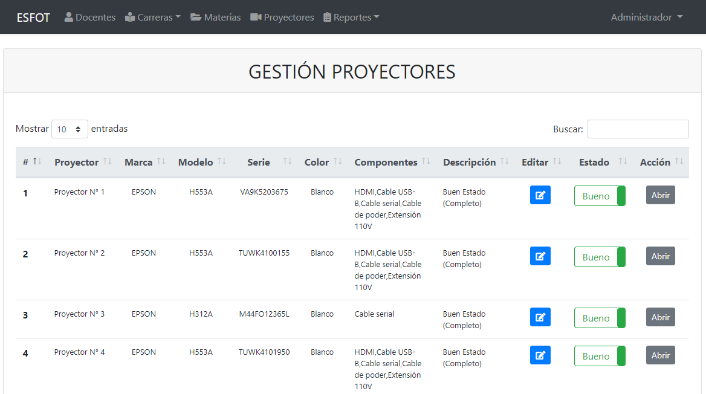
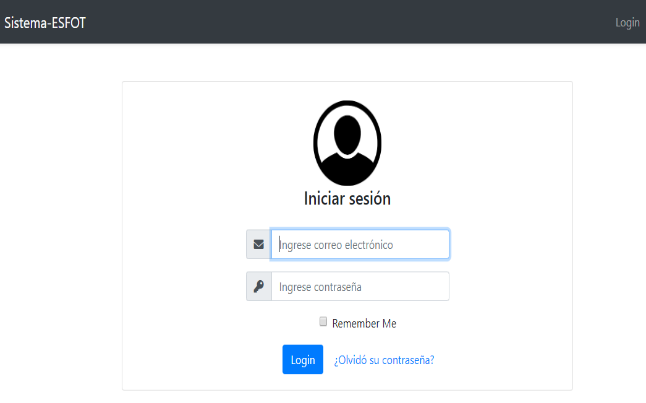
|  |  |  |  |
| --- | --- | --- | --- |
|  | **Google Chrome** | **Mozilla Firefox** | **Opera** |
| **Versión** | 80.0.3987.132 | 75.0 | 67.0.3575.137 |
| **Botones** | Excelente | Excelente | Excelente |
| **Tablas** | Excelente | Bueno | Excelente |
| **Modales** | Excelente | Excelente | Excelente |
| **Imágenes** | Excelente | Excelente | Excelente |
| **Notificaciones** | Excelente | Bueno | Excelente |
| **Animaciones** | Excelente | Bueno | Excelente |
| **Jquey, Javascript** | Excelente | Malo | Excelente |

**Elaborado por:** Jessica Carrillo

**Figura 27**. Pruebas en el navegador Google Chrome







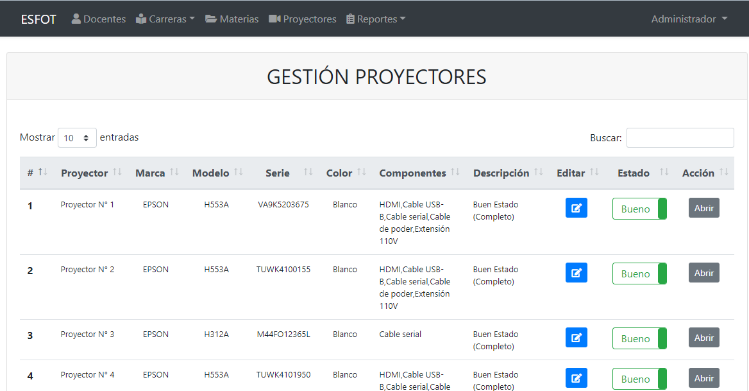
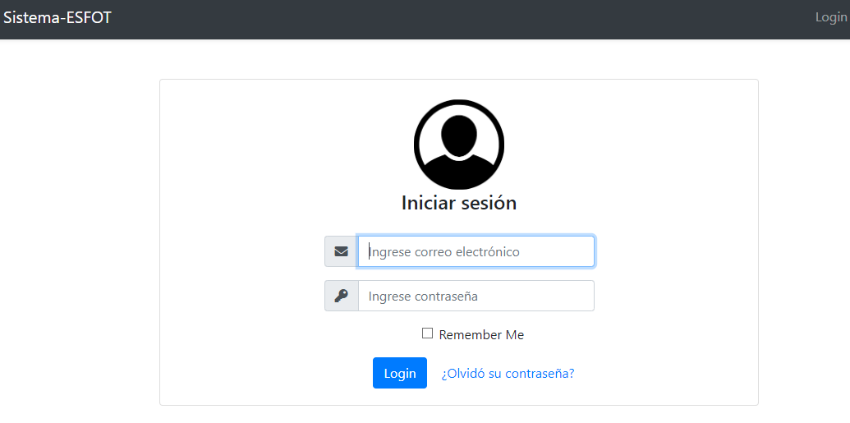


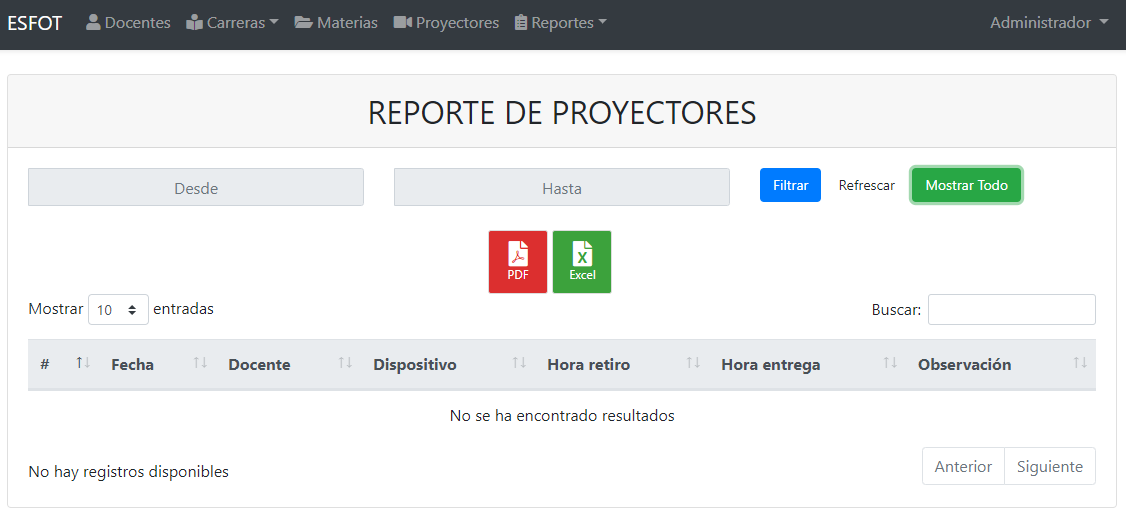
**Elaborado por:** Jessica Carrillo

**Figura 28**. Pruebas en el navegador Mozilla Firefox

**Elaborado por:** Jessica Carrillo

**Figura 29**. Pruebas en el navegador Opera





**Elaborado por:** Jessica Carrillo

Para una mejor experiencia de usuario y buen funcionamiento del sistema es indispensable que las versiones de los navegadores no sean inferiores al año 2015, ya que varios componentes desarrollados en Javascript y Jquery no son reconocidos y por ende varias funciones no se ejecutan. Como principal opción se propone el uso de Opera o Chrome para controlar el acceso a los dispositivos de proyección y registro de temas ya que Mozilla Firefox no reconoce varias funciones Query por ende no las ejecuta.

**Pruebas de aceptación**

Las pruebas de aceptación fueron diseñadas por el equipo de desarrollo con el objetivo de verificar que el sistema cumpla con los requerimientos del cliente, las cuales a su vez fueron ejecutadas y aprobadas por el usuario para su puesta en producción.

Estas pruebas se realizaron a partir de la segunda iteración en los entregables funcionales generados, dando como resultado la modificación de algunos de los módulos del sistema y por ende las historias de usuario, priorizando las necesidades del cliente.

**Tabla 10.** Prueba de aceptación 1

|  |  |
| --- | --- |
| **Caso de prueba de aceptación** | |
| **Código:** PA-01 | **Historia de usuario:** HU-01-01 y HU-03-06 |
| **Nombre:** Inicio de sesión normal y biométrico | |
| **Descripción:** El usuario administrador/docente debe identificarse ya sea por sus credenciales o el dispositivo biométrico en el login correspondiente para utilizar sus respectivas funciones. | |
| **Pasos de ejecución:**  **Inicio sesión normal:**  El administrador ingresa las credenciales (usuario-contraseña) en los campos correspondientes de la interfaz de inicio de sesión normal  Presionar el botón enviar  **Inicio sesión biométrico:**  El usuario administrador/docente coloca su huella dactilar en el dispositivo biométrico | |
| **Resultado esperado:**  El sistema reconoce tanto las credenciales y huella dactilar en las interfaces de inicio de sesión respectivas, y redirige a las páginas y funciones correspondientes de acuerdo a su perfil de usuario.  Los usuarios administradores pueden autenticarse tanto por credenciales como ´pr su huella dactilar. | |
| **Evaluación de la prueba**: Prueba satisfactoria | |

**Elaborado por:** Jessica Carrillo

**Tabla 11**. Prueba de aceptación 2

|  |  |
| --- | --- |
| **Caso de prueba de aceptación** | |
| **Código:** PA-02 | **Historia de usuario:** HU-04-01 |
| **Nombre:** Reserva de proyectores/kits | |
| **Descripción:** El usuario docente/Administrador puede realizar la reserva de proyectores/kits disponibles | |
| **Pasos de ejecución:**  Inicio de sesión biométrico  Seleccionar la opción “Gestión Proyectores” (Docentes)  Elegir una de las opciones disponibles presentadas en color verde  Presionar el botón reservar | |
| **Resultado esperado:**  El dispositivo es retirado del casillero correspondiente y al momento de cerrar la puerta el sistema debe presentar información de la reserva realizada, la opción seleccionada cambia su color al rojo y se deshabilita haciéndola no disponible ante otros usuarios.  El docente solo podrá realizar una reserva a la vez, mientras que administradores reservarán varios dispositivos siempre y cuando se encuentren disponibles.  Si el dispositivo no ha sido retirado y la puerta del casillero es cerrada no se hará efectiva la reserva.  El tiempo de reserva puede extenderse a más de dos horas o acortarse a 1 si se lo requiere, tras activar la opción habilitar e ingresar manualmente la hora de devolución | |
| **Evaluación de la prueba**: Prueba satisfactoria | |

**Elaborado por:** Jessica Carrillo

**Tabla 12.** Prueba de aceptación 3

|  |  |
| --- | --- |
| **Caso de prueba de aceptación** | |
| **Código:** PA-03 | **Historia de usuario:**  HU-04-02 |
| **Nombre: D**evolución de proyectores/kits | |
| **Descripción:** El usuario docente/Administrador puede realizar la devolución de proyectores/kits pendientes | |
| **Pasos de ejecución:**  Inicio sesión biométrico  Seleccionar la opción “Gestión Proyectores” (Docentes)  Seleccionar la opción “Devolución” ubicada en una tabla en la parte inferior que muestra información de la reserva realizada previamente  Presionar el botón “Devolver” | |
| **Resultado esperado:**  La devolución se efectuará una vez que el dispositivo se ingrese en el casillero correspondiente y se cierre de forma correcta la puerta, quedando vacía la tabla de devoluciones pendientes  Si el dispositivo no ha sido ingresado y la puerta del casillero es cerrada no se hará efectiva la devolución | |
| **Evaluación de la prueba**: Prueba satisfactoria | |

**Elaborado por**: Jessica Carrillo

**Tabla 13.** Prueba de aceptación 4

|  |  |
| --- | --- |
| **Caso de prueba de aceptación** | |
| **Código:** PA-04 | **Historia de usuario:** HU-03-07 |
| **Nombre:** Registro tema del sílabo | |
| **Descripción:** El usuario docente puede registrar los temas de clase que se imparten en el día de acuerdo a su sílabo | |
| **Pasos de ejecución:**  Inicio sesión biométrico  Seleccionar la opción “Gestión Clases”  Elegir la “Materia” correspondiente en el dropdowlist  Elegir el “Tema” de clase a impartir o impartido del segundo dropdowlist  Seleccionar opción “Permiso” si hay decreto por parte de las autoridades de la ESFOT  Presionar botón “Guardar” | |
| **Resultado esperado:**  Esperar que el sistema realice el registro correspondiente del tema en la base de datos y lo muestre en la tabla de registros de la parte inferior  Por cada registro efectuado la lista de temas pendientes irá disminuyendo | |
| **Evaluación de la prueba**: Prueba satisfactoria | |

**Elaborado por:** Jessica Carrillo

**Tabla 14.** Prueba de aceptación 5

|  |  |
| --- | --- |
| **Caso de prueba de aceptación** | |
| **Código:** PA-05 | **Historia de usuario:** HU-04-03 y HU-04-04 |
| **Nombre:** Notificaciones email a usuarios | |
| **Descripción:** El usuario docente recibirá una notificación automática a su email si ha incumplido el tiempo de entrega del dispositivo, y en caso de docentes si además no ha realizado el registro de temas al final del día | |
| **Pasos de ejecución:**  Acceder al correo registrado en el sistema | |
| **Resultado esperado:**  Esperar que el correo automático sea enviado a docentes sin registro de temas o Atraso en la devolución de los dispositivos con el contenido correspondiente a la situación | |
| **Evaluación de la prueba**: Prueba satisfactoria | |

**Elaborado por:** Jessica Carrillo

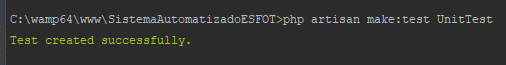
**Tabla 15.** Prueba de aceptación 6

|  |  |
| --- | --- |
| **Caso de prueba de aceptación** | |
| **Código:** PA-06 | **Historia de usuario:** HU-04-05 y HU-04-06 |
| **Nombre:** Reportes generales | |
| **Descripción:** El usuario Administrador puede generar reportes con con información relacionada al registro de temas de clase, proyectores y su devolución. | |
| **Pasos de ejecución:**  Inicio sesión normal  Ingresar al módulo reportes  Seleccionar entre las opciones “Reporte proyectores” o “Reporte clases”  Descarga opción Excel presionar botón con imagen respectiva  Descarga opción Pdf presionar botón con imagen respectiva | |
| **Resultado esperado:**  Esperar la descarga de un archivo tipo EXCEL o PDF que contenga el reporte correspondiente | |
| **Evaluación de la prueba**: Prueba satisfactoria | |

**Elaborado por:** Jessica Carrillo

**Pruebas Unitarias**

Las pruebas unitarias fueron implementadas en la fase de desarrollo o codificación del proyecto, las cuales ayudaron a verificar el correcto funcionamiento de cada unidad de código, asegurando que dicha unidad funcione como debería en el sistema. PHPUnit complemento de Laravel 5.8 fue la herramienta que se usó para efectuar las pruebas correspondientes



Culminado el proceso de ejecución de las pruebas en las funciones o métodos considerados relevantes y de importancia para el correcto funcionamiento del sistema, es posible generar un reporte general de todas las pruebas realizadas.

# **CONCLUSIONES Y RECOMENDACIONES**

## **Conclusiones**

Actualmente innumerables herramientas y dispositivos tecnológicos pueden ser controlados por sistemas web o aplicaciones móviles, dicha combinación de tecnologías fueron utilizadas para el desarrollo del sistema, el cual, que permite realizar la reserva y devolución de los recursos de proyección de la ESFOT como son proyectores o kits, así como el control y seguimiento de los temas de clases correspondientes a los sílabos de cada docente. Analizando los requerimientos establecidos y procesos manuales llevados a cabo por cada actividad, se determinó la manera de optimizar su automatización para lograr hacer más eficientes dichas actividades, minimizando a su vez el uso del papel ya que toda la información es almacenada digitalmente, además se han identificado datos relevantes para la realización de los módulos correspondientes.

Se implemento la metodología ágil de programación XP (Extremme Programming) durante todo el desarrollo del sistema, llevando una planificación incremental, en la que se realizaron entregas funcionales y parciales al cliente al final de cada una de las 5 iteraciones establecidas, su ciclo de vida al ejecutar simultáneamente sus 4 fases permitió añadir y modificar requerimientos del sistema presentados mediante historias de usuario sin afectar el proceso de desarrollo ni el tiempo de entrega del proyecto.

El diseño de base de datos y estructura del sistema fueron desarrollados en base a los requerimientos obtenidos de las entrevistas realizadas al cliente, convirtiéndose en pilares fundamentales para su posterior codificación. La herramienta Power Designer se utilizó para modelar la base de datos estableciendo tablas y atributos necesarios para el ingreso de información. Culminado el proceso de modelado se replicó mediante el gestor MySQL que almacena grandes cantidades de información y facilita la administración de manera local.

Las interfaces prototipo del sistema web se elaboraron mediante la herramienta Balsamic, el cual permitió agilizar el proceso de desarrollo y disminuir el tiempo que toma diseñar interfaces, ya que solo se imitó el boceto previamente presentado y aceptado por cliente.

Para la codificación del sistema web se utilizó el Framework de Laravel, por ser una plataforma altamente flexible, escalable y de fácil programación, basado en el lenguaje PHP y modelo MVC que simplifica el proceso de mantenimiento el proyecto, así como la reutilización de métodos o funciones entre los diferentes módulos. Se crearon dos tipos de autenticación que trabajan de manera independiente, por un lado, se mantiene la autenticación por credenciales (email, contraseña) usado únicamente por personal administrativo a través del navegador de un computador. Por otro lado, está la autenticación biométrica ejecutada mediante la huella dactilar usada tanto por docentes y administradores accedida únicamente por el navegador de la Tablet ubicada en los casilleros que alojan a los dispositivos de proyección.

El control de acceso a los dispositivos de proyección alojados en los casilleros correspondientes y la captura de datos en tiempo real del dispositivo biométrico para la autenticación se desarrollaron mediante el lenguaje de programación Python y el uso de librerías como pyzk, serial, time a manera de scripts, lo que permitió levantar en el servidor de alojamiento del sistema servicios de transferencia de datos en tiempo real, esto debido su sintaxis multiplataforma y multiparadigma que facilita la comunicación entre dispositivos electrónicos (biométrico, arduino) y el sistema web, recogiendo información cada vez que se realicen peticiones, por lo que él tiempo de respuesta es corto. Al no ser posible ejecutar o levantar dos scripts al mismo tiempo se hizo uso de la librería subprocess de Python para invocar de manera simultánea dichos procesos.

Se realizaron diferentes pruebas al sistema con el objetivo de comprobar su correcto funcionamiento, así como el cumplimiento de todos y cada uno de los requerimientos planteados por el cliente. Las pruebas unitarias realizadas durante el proceso de desarrollo ayudaron a verificar que los métodos principales funcionen correctamente, detectando errores que fueron rápidamente solucionados, cabe recalcar que dicha prueba fue implementada únicamente a funciones y métodos considerados importantes, debido al tiempo que conllevaban realizarlas.

Las pruebas de compatibilidad permitieron verificar que las interfaces de usuario y todos sus complementos se acoplen y funcionen bien en diferentes navegadores web, estableciendo además que la versión mínima de Chrome para el módulo control de acceso a los dispositivos de proyección y registro de temas debe ser superior o igual a 60.0.3112 ya que muchas de las funciones Javascript no son reconocidas.

Las pruebas de aceptación permitieron verificar que los módulos principales del sistema funcionen de acuerdo a los requisitos establecidos al inicio del proyecto, cumpliendo las expectativas del cliente, definiendo casos de prueba que agrupaban las historias de usuario que contenían las funcionalidades principales.

## **Recomendaciones**

Se sugiere revisar el manual de usuario que se encuentra en el Anexo G para un mejor entendimiento al hacer uso del sistema evitando inconvenientes, ya que se presenta de manera detallada las acciones a realizar en cada módulo.

Es necesario que se delegue a una persona para ejecutar las acciones administrativas relacionadas con el CRUD de los módulos de usuarios, dispositivos de proyección, carreras, materias, períodos académicos e importación de sílabos.

El administrador al momento de ingresar un nuevo usuario deberá tomar las huellas dactilares correspondientes desde el dispositivo biométrico, teniendo en cuenta que desde el sistema web solo es posible el ingreso de datos como nombre, id-biométrico, contraseña y rol, los cuales son registrados simultáneamente tanto en la base de datos del sistema como el dispositivo biométrico.

Al cargar los sílabos o cronogramas correspondientes a cada docente se debe subir solamente en formato .xlsx, ya que es el único que recepta el sistema web, además se tiene la opción de eliminar de manera masiva los datos importados.

Para el mantenimiento o actualización del sistema se recomienda seguir el patrón MVC, en el que se basa el Framework Laravel y la guía de actualización encontrada en el sitio web styde.net.

# **REFERENCIAS BIBLIOGRÁFICAS**

Balsamiq, S. (2020). *balsamiq.* Obtenido de https://balsamiq.com/

Blog, R. (20 de Abril de 2018). *Requerimientos Funcionales y No Funcionales, ejemplos y tips.* Obtenido de https://medium.com/@requeridosblog/requerimientos-funcionales-y-no-funcionales-ejemplos-y-tips-aa31cb59b22a

CES. (2012). *CES*. Obtenido de http://www.ces.gob.ec/index.php?option=com\_phocadownload&view=category&id=12&Itemid=303

Domingues, P. (27 de Marzo de 2018). *¿Qué es jQuery?* Recuperado el 04 de Marzo de 2020, de https://openclassrooms.com/en/courses/4309491-simplifica-tus-proyectos-con-jquery/4309498-que-es-jquery

Ecured. (06 de Julio de 2019). *PowerDesigner.* Obtenido de https://www.ecured.cu/PowerDesigner

Edwards, T. (28 de Abril de 2019). *sweetalert.* Recuperado el 04 de Marzo de 2020, de https://sweetalert.js.org/guides/

ESFOT. (2019). *ESFOT.* Obtenido de https://esfot.epn.edu.ec/

Flores, L. E. (2020). *Ingenieria de software*. Obtenido de http://ingenieriadesoftware.mex.tl/52753\_xp---extreme-programing.html

García, M. (27 de Noviembre de 2015). *Extreme Programming (XP).* Obtenido de Funcionamiento: https://iswugxp.wordpress.com/funcionamiento/

Grau, J. L. (8 de Julio de 2016). *proagilist*. Obtenido de https://proagilist.es/blog/agilidad-y-gestion-agil/agile-scrum/sabes-como-funciona-xp/

Instituto Nacional de Ciberseguridad. (27 de 06 de 2018). *Osi.es.* Obtenido de IoT, el universo conectado | Oficina de Seguridad del Internauta: https://www.osi.es/es/actualidad/blog/2018/06/27/iot-el-universo-conectado

Jiménez, J. D. (20 de 01 de 2019). *Qué es HTML5.* Recuperado el 03 de 04 de 2019, de https://openwebinars.net/blog/que-es-html5/

Letelier, P., & Penadés, M. (15 de 01 de 2006). *Métodologías ágiles para el desarrollo de software.* Obtenido de https://www.cyta.com.ar/ta0502/v5n2a1.htm

MDN. (11 de 05 de 2019). *¿Qué es JavaScript?* Recuperado el 04 de 03 de 2020, de https://developer.mozilla.org/es/docs/Learn/JavaScript/First\_steps/Qu%C3%A9\_es\_JavaScript

Navia, F. R. (06 de Marzo de 2018). *itsoftware.* Recuperado el 26 de 02 de 2020, de https://itsoftware.com.co/content/que-es-y-para-que-sirve-mysql/

Nube, C. (28 de Mayo de 2019). *Aplicación con Laravel.* Recuperado el 04 de Marzo de 2020, de https://blog.nubecolectiva.com/que-es-laravel-tu-primera-aplicacion-con-laravel/

Pérez, A. (5 de Junio de 2013). Obtenido de https://sites.google.com/site/alfonsoperezr/investigacion/estructuracin-y-especificacin-de-casos-de-uos

*pyzk.* (13 de Agosto de 2019). Recuperado el 04 de Marzo de 2020, de https://libraries.io/pypi/pyzk

Robledano, Á. (23 de Septiembre de 2019). *Qué es Python: Características, evolución y futuro.* Recuperado el 04 de Marzo de 2020, de https://openwebinars.net/blog/que-es-python/

team, B. (01 de 08 de 2019). *Bootstrap.* Obtenido de https://getbootstrap.com/

Veracruzana, U. (11 de Junio de 2012). *Universo*. Obtenido de https://www.uv.mx/universo/486/infgral/infgral\_15.html

# **ANEXOS**

## **MANUAL DE USUARIO**

## **MANUAL TÉCNICO**

1. Anexo A. Historias de usuario

Documento Adjunto: Documento Word con las historias de usuario (formato doc.)

1. Anexo B. Diagramas de caso de uso

Documento Adjunto: Diagramas de caso de uso del sistema (formato JPG.)

1. Anexo C. Diagrama de base de datos del sistema

Documento Adjunto: Diagrama modelo entidad relación de la base de datos del sistema (formato JPG.)

1. Anexo D. Tablas y atributos de la base de datos

Documento Adjunto: Documento Word que contiene la descripción de las tablas principales, secundarias, así como cada uno de sus atributos o campos del sistema (formato docx.)

1. Anexo E. Prototipos de las interfaces del Sistema

Documento Adjunto: Documento de tipo PDF que contiene los prototipos correspondientes a las interfaces del sistema (formato PDF.)

1. Anexo F. Interfaces del sistema

Documento Adjunto: Documento Word que incluye todas las interfaces del sistema a manera de imagen (formato docx)