TECNOLÓGICO DE ESTUDIOS SUPERIORES DE COACALCO

Módulo de Detección de Materiales del Sistema Integral de Dispositivos de Reciclaje con JAVA PHYTON y MYSQL para el Tecnológico de Querétaro

Proyecto Integral de Educación Dual

Para obtener el título de,

Ingeniería en Sistemas Computacionales

Especialidad

Ingeniería de Software

Presenta(n),

Fuentes Rivero Mario Alberto

Matricula 201820886

Asesor Interno

Ing. Fabiola Fuentes Herrera

Coacalco de Berriozábal, Marzo/ 2025

AGRADECIMIENTOS

Primeramente, agradecemos a INTGEN TECHNOLOGIES por darnos la oportunidad de poder laborar en un proyecto tan extenso que claramente nos brindó la oportunidad de crecer laboralmente, aplicando los conocimientos que adquirimos previamente.

También a la Ing. Fabiola Fuentes Herrera por impulsarnos a buscar más y mejores maneras para crecer en nuestra vida laboral, apoyándonos en dudas y cualquier cosa que se nos pudiese presentar

RESUMEN

En el presente proyecto se documenta todo el proceso de desarrollo e implementación del proyecto “Módulo de Detección de Materiales del Sistema Integral de Dispositivos de Reciclaje con JAVA PHYTON y MYSQL para el Tecnológico de Querétaro”, el cual tendrá como propósito de desarrollar un sistema de reciclaje en donde se busca ofrecer una sistema web, donde se pueden consultar todos los módulos de detección que estarán en el Tecnológico Nacional de México Plantel Querétaro, los usuarios puedan realizar movimientos internos consultas de saldo obtenido entre otros.

El desarrollo del presente proyecto se realizó para que nosotros como ingenieros de la carrera en sistemas computacionales, podamos adentrarnos al mundo laboral, y de este modo tener facilidades para crecer en nuestra vida profesional. Por lo tanto, el desarrollo de este sistema, se ajusta a nuestras aptitudes y conocimientos previamente adquiridos durante nuestra etapa de aprendizaje, donde podremos aplicar lenguajes y herramientas para poder satisfacer las necesidades y expectativas del cliente.

El contenido del presente documento consta de cuatro capítulos, en donde el capítulo I corresponde a las generalidades del proyecto, en este apartado se da a conocer el planteamiento de la problemática correspondiente a identificar necesidades y problemas a resolver, se declara nuestro objetivo general y objetivos específicos a cumplir durante el desarrollo del proyecto con su respectiva justificación del porque se realiza el mismo.

En el capítulo II se definen los métodos para poder gestionar el proyecto, obteniendo como beneficio una mejor organización para poder asignar distintas tareas y actividades de manera equitativas en el equipo, a la vez se da a conocer un marco teórico, con todos los temas técnicos como son los lenguajes de programación, frameworks, tipos de bases de datos, conceptos y utilidades que son utilizadas en el desarrollo del proyecto.

En el capítulo III se realizó un estudio de la situación actual respecto al mercado, se declara el cronograma de actividades que se deben realizar especificando el nombre de la actividad, y en que rango de fechas establecidas se deben de cumplir, además se presenta todo el desarrollo del proyecto colocando diagramas de caso de uso.

En el capítulo IV corresponde al apartado de los resultados obtenidos, esto implica el termino de todas las actividades previamente establecidas en el cronograma, concluyendo el

desarrollo del proyecto, adjuntando imágenes como evidencia correspondiente a todas las páginas y funcionalidades que conforman el sistema web, como es la utilización de los frameworks de primefaces, la confirmación de envió de correos electrónicos, los módulos de reciclaje en línea, gestión de tarjetas y usuarios, consultas de movimientos. Para finalizar se dan a conocer los respectivos resultados, la conclusión, los anexos y las referencias bibliográficas.

CONTENIDO

1.1 INTRODUCCIÓN 15
1.2 DESCRIPCIÓN DE LA EMPRESA 16
1.2.1 RAZÓN SOCIAL, UBICACIÓN, ORGANIGRAMA Y SECTOR. 16
1.2.2 MISIÓN 17
1.2.3 VISIÓN 17
1.2.4 VALORES 17
1.2.5 BREVE HISTORIA DE LA EMPRESA 17
1.3 DESCRIPCIÓN DEL PUESTO O ÁREA DE TRABAJO 19
1.5 PROBLEMA A RESOLVER 20
1.6 OBJETIVOS 21
1.6.1 GENERAL 21
1.6.2 ESPECÍFICOS 21
1.7 JUSTIFICACIÓN 22
MARCO TEÓRICO 23
2.1 INTRODUCCION 24
2.2 APLICACIÓN WEB 24
2.2.1 ANTECEDENTES DE LA APLICACIÓN WEB 24
2.2.2 INTERFAZ 25
2.2.3 CONSIDERACIONES TÉCNICAS 25
2.2.4 ESTRUCTURA DE LAS APLICACIONES WEB 26
2.2.5 USO EMPRESARIAL 26
2.2.6 VENTAJAS 27
2.2.7 DESVENTAJAS 28
2.2.8 DIFERENCIA ENTRE APLICACIÓN WEB Y APLICACIÓN DE INTERNET ENRIQUECIDA (RIA) 28
2.3 ARQUITECTURA CLIENTE SERVIDOR 29
2.3.1 BACKEND 29
2.3.2 FRONTEND 29

2.3.3 WEB SERVICES **30**
2.3.4 AUTENTICACIÓN **30**
2.3.5 AUTORIZACIÓN **30**
2.3.6 REST **30**
2.4 ARQUITECTURA MVC. **31**
2.4.1 MODELO **32**
2.5.2 VISTA **32**
2.5.3 CONTROLADOR **33**
2.7 SPRING SECURITY **34**
2.7.1 INCLUIR SPRING SECURITY EN UNA APLICACIÓN WEB **35**
2.7.2 PROTEGER SECCIONES DE LA APLICACIÓN **37**
2.7.3 ADMINISTRAR LOS USUARIOS DEL PROGRAMA **38**
2.8 HIBERNATE **41**
2.8.1 HISTORIA HIBERNATE **41**
2.8.2 FUNCIONAMIENTO DE HIBERNATE **41**
2.8.2.1 Objeto de Configuración Hibernate **43**
2.8.2.2 Objetivo SessionFactory **43**
2.8.2.3 Objeto Session **43**
2.8.2.4 Objeto Transaction **43**
2.8.2.4 Objeto Query **43**
2.8.2.4Objeto Criteria **43**
2.9 JPA 44
2.9.1 UNIDAD DE PERSISTENCIA **45**
2.9.2 ENTITYMANAGERFACTORY **45**
2.9.3 CONTEXTO DE PERSISTENCIA **45**
2.9.4 ENTITYMANAGER **46**
2.9.5 OBJETOS DE ENTIDAD **46**
2.10 LENGUAJES DE DESARROLLO IMPLEMENTADOS **46**

2.10.1 JAVA **46**
2.10.2 JSP: JAVA SERVER PAGES **47**
2.10.3 XHTML **48**
2.10.3.1 características **48**
2.11 MYSQL **50**
2.12 FRAMEWORKS **52**
2.12.1 JSF JAVA SERVER FACES **52**
2.12.2 BOOTSFACES **52**
2.12.3 PRIMEFACES **53**
2.13 SISTEMAS DE GESTIÓN DE BASES DE DATOS (SGBD) **53**
DESARROLLO **55**
3.1 INTRODUCCIÓN **56**
3.2 ESTUDIO SITUACIÓN ACTUAL **56**
3.3 ANÁLISIS **57**
3.3.1 CRONOGRAMA DE ACTIVIDADES **57**
3.4 ARQUITECTURA **60**
3.5 REQUISITOS DEL SISTEMA **62**
3.5.1 REQUERIMIENTOS FUNCIONALES **63**
3.4.1.1 Iniciar Sesión en el Sistema **63**
3.4.1.2 Gestión de Usuarios para el Administrador (CRUD) **64**
3.4.1.3 Gestión de Tarjetas para el Administrador (CRUD) **65**
3.4.1.4Consulta y búsqueda de Movimientos para el administrador **67**
3.4.1.5 Consulta y búsqueda de errores para el Administrador **68**
3.4.1.5 Cambio de estatus para los errores surgidos para el Administrador **70**
3.4.1.6 Gestión de máquinas para los Administradores **72**
3.4.1.6 Cambio de Contraseña para todos los usuarios **74**
3.4.1.6 Consulta y búsqueda de los movimientos de cuenta propia para un usuario Regular **75**

4.1 INTRODUCCIÓN **78**
4.2 RESULTADO DE INTERFACES **79**
4.2.1 INICIO DE SESIÓN EN LA PLATAFORMA **79**
4.2.2 CAMBIO DE CONTRASEÑA EN LA PÁGINA DE PERFIL **81**
4.2.3 USUARIO ADMINISTRADOR **82**
4.2.3.1 Página de Inicio del Administrador **82**
4.2.3.2 Gestión de usuarios para el Administrador **84**
4.2.3.3 Gestión de tarjetas de los usuarios para el Administrador **87**
4.2.3.4 Consulta y filtro de búsqueda de los movimientos para el Administrador **88**
4.2.3.5 Consulta y filtro de búsqueda de los errores para el Administrador **90**
4.2.4 USUARIO REGULAR **92**
4.2.4.1 Página de Inicio de un Usuario Regular **92**
4.2.3.3 Consulta y filtro de búsqueda de los movimientos del usuario Regular **94**
CONCLUSIONES **96**
COMPETENCIAS DESARROLLADAS **97**
FUENTES DE INFORMACIÓN **98**

INDICE DE FIGURAS

FIG. 1 UBICACIÓN DE LA EMPRESA 16
FIG. 2 ORGANIGRAMA INTGEN TECHNOLOGIES 16
FIG. 3 MODELO VISTA CONTROLADOR 34
FIG. 4 CODIFICACIÓN DE LAS PROPIEDADES DE SPRING SECURITY 36
FIG. 5 CODIFICACIÓN DE LA CONFIGURACIÓN PARA LA PROTECCIÓN DE SECCIONES 38
FIG. 6 INTERFAZ USERDETAILSSERVICE 39
FIG. 8 CODIFICACIÓN PARA REGISTRAR UNA CLASE 40
FIG. 9 MODELO DE LAS CAPAS QUE MANEJA HIBERNATE 42
FIG. 10 EJEMPLO DE UN CÓDIGO DE XHTML 49
FIG. 11 IMPORTACIÓN DE LIBRERÍAS 60
FIG. 12 ARCHIVO PERSISTENCE 61
FIG. 13 CONFIGURACIÓN DE SEGURIDAD 62
FIG. 14 CASO DE USO DE INICIAR SESIÓN EN EL SISTEMA 63
FIG. 15 CASO DE USO DE GESTIÓN DE USUARIOS PARA EL ADMINISTRADOR (CRUD) 64
FIG. 16 CASO DE USO DE INICIAR SESIÓN EN EL SISTEMA 66
FIG. 17 CASO DE USO DE CONSULTA Y BÚSQUEDA DE MOVIMIENTOS PARA EL ADMINISTRADOR 67
FIG. 18 CASO DE USO DE CONSULTA Y BÚSQUEDA DE ERRORES PARA EL ADMINISTRADOR 68
FIG. 19 CASO DE USO DE CAMBIO DE ESTATUS PARA LOS ERRORES SURGIDOS PARA EL ADMINISTRADOR 70
FIG. 20 CASO DE USO DE GESTIÓN DE MÁQUINAS PARA LOS ADMINISTRADORES 72
FIG. 21 CASO DE USO DE CAMBIO DE CONTRASEÑA PARA TODOS LOS USUARIOS 74
FIG. 22 CASO DE USO DE CONSULTA Y BÚSQUEDA DE MOVIMIENTOS PARA EL USUARIO REGULAR 76
FIG. 23 PAGINA DE INICIAR SESIÓN 79

FIG. 24 PÁGINA DE INICIO ADMINISTRADOR 80
FIG. 25 PAGINA DE PERFIL 81
FIG. 26 VENTANA EMERGENTE DE CONFIRMACIÓN DE CAMBIO DE CONTRASEÑA 81
FIG. 27 MENÚ PARA EL ADMINISTRADOR 82
FIG. 28 SECCIÓN DE DATOS DEL USUARIO 82
FIG. 29 SECCIÓN DE DATOS GENERALES DEL SISTEMA 83
FIG. 30 SECCIÓN DE GRAFICA DE PUNTOS GENERADOS 83
FIG. 31 PAGINA DE GESTIÓN DE USUARIOS PARA EL ADMINISTRADOR .. 84
FIG. 32 FORMULARIO PARA REGISTRAR UN NUEVO USUARIO 84
FIG. 33 EJEMPLO DE CORREO CON CONTRASEÑA PARA EL REGISTRO DE UN USUARIO NUEVO (PROPIA AUTORÍA, 2022) 85
FIG. 34 TABLA DE LISTA DE USUARIO REGISTRADOS 85
FIG. 35 VENTANA DE CONFIRMACIÓN PARA ELIMINAR UN USUARIO..... 86
FIG. 36 VENTANA DE RESTAURAR LA CONTRASEÑA DE UN USUARIO 86
FIG. 37 PAGINA DE EDICIÓN DE LOS USUARIOS PARA EL ADMINISTRADOR 87
FIG. 38 TABLA DE TARJETAS DE LOS USUARIOS..... 87
FIG. 39 PAGINA DE CONSULTA Y FILTRO DE BÚSQUEDA DE MOVIMIENTOS PARA EL ADMINISTRADOR 88
FIG. 40 FILTRO DE BÚSQUEDA PARA LA CONSULTA DE MOVIMIENTOS PARA EL ADMINISTRADOR 88
FIG. 41 TABLA DE LISTA DE MOVIMIENTOS 89
FIG. 42 VENTANA EMERGENTE CON EL DETALLE DEL MOVIMIENTO..... 89
FIG. 43 PAGINA DE CONSULTA Y FILTRO DE BÚSQUEDA DE ERRORES PARA EL ADMINISTRADOR 90
FIG. 44 FILTRO DE BÚSQUEDA PARA LA CONSULTA DE ERRORES PARA EL ADMINISTRADOR 90
FIG. 45 TABLA DE LISTA DE ERRORES PARA EL ADMINISTRADOR 91
FIG. 46 VENTANA EMERGENTE PARA CAMBIAR EL ESTATUS DEL ERROR 91

FIG. 47 VENTANA EMERGENTE CON EL DETALLE DEL ERROR 91
FIG. 48 PÁGINA DE INICIO DE UN USUARIO REGULA 92
FIG. 49 MENÚ PARA UN USUARIO REGULAR..... 92
FIG. 50 SECCIÓN DE DATOS DEL USUARIO..... 93
FIG. 51 SECCIÓN DE DATOS DE LA CUENTA DEL USUARIO..... 93
FIG. 52 SECCIÓN DE GRAFICA DE PUNTOS GENERADOS POR EL USUARIO 93
FIG. 53 PAGINA DE CONSULTA Y FILTRO DE BÚSQUEDA DE MOVIMIENTOS PARA EL USUARIO REGULAR..... 94
FIG. 54 FILTRO DE BÚSQUEDA PARA LA CONSULTA DE MOVIMIENTOS PARA UN USUARIO REGULAR 94
FIG. 55 TABLA DE LISTA DE MOVIMIENTOS DE UN USUARIO REGULAR .. 95

INDICE DE TABLAS

TABLA 1. CRONOGRAMA DE ACTIVIDADES 58
TABLA. 2 TABLA DE REQUERIMIENTO INICIAR SESIÓN EN EL SISTEMA .. 63
TABLA. 3 TABLA CASO DE USO INICIAR SESIÓN EN EL SISTEMA 64
TABLA. 4 TABLA DE REQUERIMIENTO GESTIÓN DE USUARIOS PARA EL ADMINISTRADOR (CRUD) 64
TABLA. 5 TABLA CASO DE USO GESTIÓN DE USUARIOS PARA EL ADMINISTRADOR (CRUD) 65
TABLA. 6 TABLA DE REQUERIMIENTO GESTIÓN DE USUARIOS PARA EL ADMINISTRADOR (CRUD) 65
TABLA. 7 CASO DE USO GESTIÓN DE TARJETAS PARA EL ADMINISTRADOR (CRUD) 66
TABLA. 8 TABLA DE REQUERIMIENTO CONSULTA Y BÚSQUEDA DE MOVIMIENTOS PARA EL ADMINISTRADOR 67
TABLA. 9 TABLA CASO DE USO CONSULTA Y BÚSQUEDA DE MOVIMIENTOS PARA EL ADMINISTRADOR 67
TABLA. 10 TABLA DE REQUERIMIENTO CONSULTA Y BÚSQUEDA DE ERRORES PARA EL ADMINISTRADOR 68
TABLA. 12 TABLA DE REQUERIMIENTO CAMBIO DE ESTATUS PARA LOS ERRORES SURGIDOS PARA EL ADMINISTRADOR 70
TABLA. 13. TABLA CASO DE USO CAMBIO DE ESTATUS PARA LOS ERRORES SURGIDOS PARA EL ADMINISTRADOR 71
TABLA. 14 TABLA DE REQUERIMIENTO GESTIÓN DE MÁQUINAS PARA LOS ADMINISTRADORES 72
TABLA. 15. TABLA CASO DE USO GESTIÓN DE MÁQUINAS PARA LOS ADMINISTRADORES 73
TABLA. 16 TABLA DE REQUERIMIENTO CAMBIO DE CONTRASEÑA PARA TODOS LOS USUARIOS 74
TABLA. 17 TABLA CASO DE USO CAMBIO DE CONTRASEÑA PARA TODOS LOS USUARIOS 75

TABLA. 18 TABLA DE REQUERIMIENTO CONSULTA Y BÚSQUEDA DE MOVIMIENTOS PARA EL USUARIO REGULAR 75

TABLA. 19. TABLA CASO DE USO CONSULTA Y BÚSQUEDA DE MOVIMIENTOS PARA EL USUARIO REGULAR 76

CAPITULO I

GENERALIDADES DEL PROYECTO

1.1 INTRODUCCIÓN

En el siguiente reporte de residencias profesionales, se dará a conocer el proyecto con el título de *"Módulo de Detección de Materiales del Sistema Integral de Dispositivos de Reciclaje con JAVA PHYTON y MYSQL para el Tecnológico de Querétaro"* para la empresa *"INTGEN TECHNOLOGIES DE MÈXICO SA DE CV."* Para así adquirir un beneficio propio, el cual será la experiencia laboral en las tecnologías a utilizar, del mismo poder aplicar todo el conocimiento adquirido durante nuestra vida escolar.

El reciclaje de materiales es un tema de gran importancia en la sociedad actual, ya que ayuda a reducir la huella de carbono y a preservar los recursos naturales. Con el objetivo de mejorar los procesos de reciclaje, este trabajo presenta el desarrollo de un módulo de gestión de usuarios para un sistema integral de dispositivos de reciclaje. Este módulo está desarrollado utilizando las tecnologías Java, Python y MySQL, las cuales son las tecnologías líderes en el desarrollo de aplicaciones.

Java es utilizado para el Backend de la aplicación, se utilizó Java Enterprise Edition (Java EE) para desarrollar la lógica de negocio, la estructura interna, los registros de usuarios y módulos, así como la conexión con la base de datos. Se utilizó también para desarrollar una aplicación web Frontend, para facilitar la visualización de los resultados obtenidos.

Por último, MySQL se utilizó para almacenar los datos recabados en las dos etapas anteriores, lo que permite tener acceso a ellos para su análisis posterior. Esto permite tener un registro de todo el proceso y poder hacer mejoras continuas en el sistema.

Este trabajo busca contribuir a mejorar la gestión de procesos, datos y puntos generados por los dispositivos de reciclaje para tener un buen control de los usuarios que hacen uso de estos dispositivos para contribuir en el reciclaje.

1.2 DESCRIPCIÓN DE LA EMPRESA

1.2.1 Razón social, ubicación, organigrama y sector.

La empresa fue fundada en 2014 por el Mtro. Allan Ronier Diez Barroso Agraz con la razón social de INTGEN TECHNOLOGIES DE MÈXICO SA DE CV.con el giro Industrial, con la ubicación en la matriz en Calle San Francisco, Los Héroes Coacalco, 55712 San Francisco Coacalco, Méx como se muestra en la Fig 1., la cual cuenta con un organigrama conformado por 12 áreas como se muestra en la Fig. 2

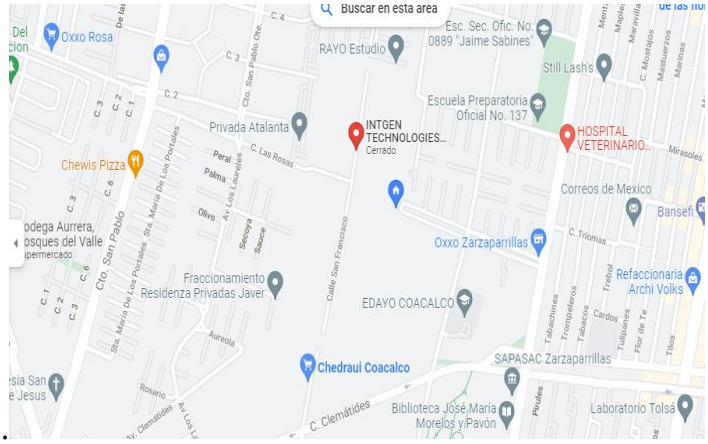


Fig. 1 Ubicación de la Empresa (Fuente: Google Maps,2022)

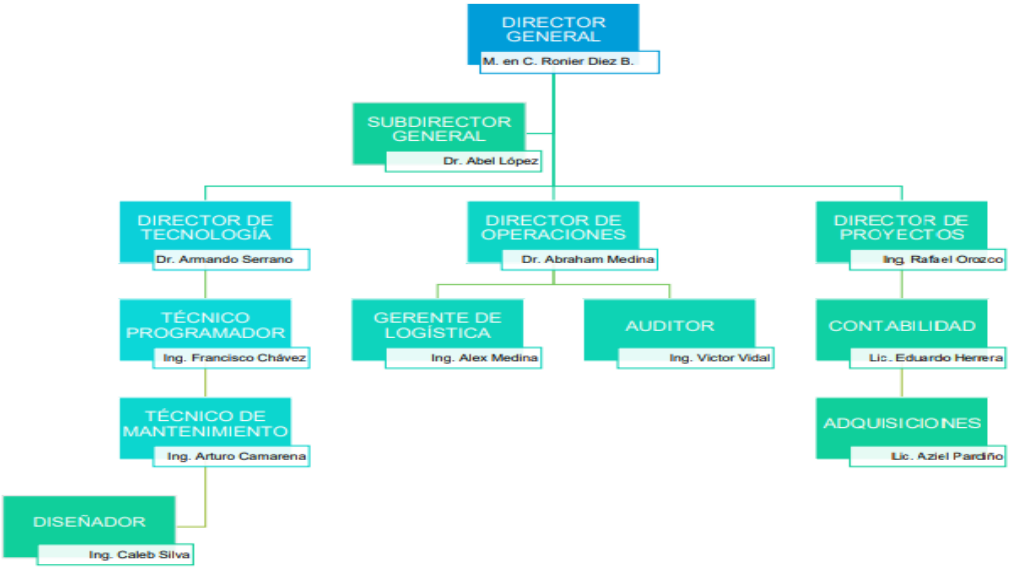


Fig. 2 Organigrama INTGEN TECHNOLOGIES (Fuente: INTGEN TECHNOLOGIES, 2022)

1.2.2 Misión

Proveer a la sociedad mexicana tecnologías, productos y diseños innovadores, sustentables y funcionales, que generen nuevas experiencias y buenas expectativas que enriquezcan sus operaciones junto a un servicio de calidad.

1.2.3 Visión

Liderar el desarrollo e innovación tecnológica en el territorio nacional evolucionando continuamente para generar una oferta de valor a la industria manufacturera que se adecue a sus necesidades y exigencias asegurando productos diversos, de calidad y con un suministro confiable.

1.2.4 Valores

- Liderazgo
- Trabajo en equipo
- Honestidad
- Responsabilidad
- Creatividad
- Respeto
- Comunicación asertiva

1.2.5 Breve historia de la empresa

INTGEN TECHNOLOGIES DE MÉXICO S.A DE C.V se crea en 2014 atendiendo principalmente a proyectos de domótica, cableado eléctrico en baja tensión, redes de cableado estructurado e infraestructura civil. Esta empresa tuvo un declive económico para inicios de 2015 y desafortunadamente tuvo que suspender sus obligaciones fiscales.

En septiembre de 2020 reinician nuevamente actividades operativas y se añaden socios capitalistas, con lo que comienza una nueva etapa en la empresa y comienzan a dar servicios a instituciones como son: la Secretaría de la Defensa Nacional, el Instituto Politécnico Nacional y el Tecnológico Nacional de México Campus Querétaro, actualmente la empresa se encuentra ampliando su plantilla en recursos humanos y realizando nuevos

proyectos, ejemplo de ello es el desarrollo de nuevas tecnologías, las cuales se encuentran en trámite de protección intelectual.

Actualmente INTGEN se dedica a comercializar, administrar, supervisar y ejecutar proyectos de software y hardware en el desarrollo de ingeniería en sistemas, integrando servicios profesionales, tales como servicios de consultoría y asesoría. Así mismo comercializar con materias primas, productos terminados o semiterminados para la industria petrolera, manufacturera, metal, mecánica, extractiva, militar, naval y de construcción.

Todo lo anteriormente mencionado, como se exhibe en el acta constitutiva de la empresa, con escritura pública 59017, fecha de escritura del 9 de abril de 2014, en la notaría pública número 93, en Cuautitlán Izcalli, Estado de México, bajo el registro público de la licenciada Liliana Castañeda Salinas, en la notaría 93, bajo el registro público de la propiedad y comercio con fecha de 10 de julio de 2014 en Cuautitlán Izcalli, Estado de México, con número de registro 23710-10.

1.3 DESCRIPCIÓN DEL PUESTO O ÁREA DE TRABAJO

Puesto a que realizan los residentes son de programadores junior los cuales están coordinados por el Lic. Ezequiel Ramon Cedillo Cruz para realizar las siguientes actividades:

- Desarrollar y mantener aplicaciones y sistemas utilizando Java, Python y MySQL.
- Participar en la investigación y el análisis de problemas técnicos y en la propuesta de soluciones.
- Trabajar en estrecha colaboración con otros programadores y equipos de desarrollo de software.
- Escribir y mantener documentación técnica.
- Asegurar que el código cumpla con los estándares de calidad y se mantenga actualizado
- Los resultados esperados incluyen la entrega oportuna y de alta calidad de aplicaciones y sistemas, la resolución eficiente de problemas técnicos y la contribución a la mejora continua de procesos y herramientas de desarrollo de software.

1.5 PROBLEMA A RESOLVER

La empresa INTGEN TECHNOLOGIES DE MÈXICO SA DE CV se encarga de la elaboración de dispositivos de reciclaje los cuales son máquinas identificadoras y separadoras de materiales inorgánicos, dichos materiales corresponden a envases plásticos, envases de aluminio (lata en general) y latas de metal. Posteriormente en base a el material ingresado calcula el peso en cada contenedor y se generará un puntaje en función de los kilogramos introducidos. Dicho puntaje se almacenará en una base de datos relacionada con las tarjetas de usuario vía RFID/NFC. Sin embargo, no hay un sistema que pueda gestionar todos los datos almacenados en la base de datos para así llevar un buen control y manejo de datos. Por otra parte, no hay manera en la cual los administradores puedan gestionar los usuarios, maquinas, tarjetas y puntos generados de los dispositivos de reciclaje.

La empresa requiere un sistema donde un administrador sea capaz de gestionar los usuarios, dispositivos de reciclaje y tarjetas RFID/NFC registradas dentro del sistema. Se busca que el sistema sea capaz de mostrar los movimientos o puntos generados totales gracias a los dispositivos de reciclaje. Por otra parte, existe el riesgo de que surjan errores dentro de los dispositivos de reciclaje por lo cual el sistema debe ser capaz de administrar dichos errores para llevar un buen control de estos mismos. Para que los usuarios que generen puntos dentro de los dispositivos se propone la opción de agregar un tipo de usuario el cual será capaz de consultar los puntos que el usuario ha generado dentro del sistema, junto con los puntos que contiene su cuenta en tiempo real.

1.6 OBJETIVOS

1.6.1 General

Desarrollar sistema de gestión de usuarios en el Sistema Integral de Dispositivos de Reciclaje por medio de lenguaje JAVA y manejo de datos con MYSQL.

1.6.2 Específicos

- Desarrollar base de datos que sea capaz de cumplir con los requisitos solicitados por el cliente.
- Diseñar las interfaces de las vistas cumpliendo los requisitos del cliente (Iniciar sesión, de la gestión de usuarios, de Inicio, de gestión de movimientos, de gestión de Errores, de gestión de Maquinas, de Perfil).
- Establecer una arquitectura estable y segura para el sistema
- Un sistema capaz de gestionar los usuarios, tarjetas y maquinas registrados dentro del mismo.
- Desarrollar un sistema de fácil acceso y entendimiento para usuarios generales.
- Implementar un diseño responsivo para que la plataforma se adapte a los tamaños de distintos dispositivos
- Un diseño atractivo de la web, que sea dinámico y agradable para el usuario.

1.7 JUSTIFICACIÓN

Los dispositivos de reciclaje de materiales inorgánicos es una solución innovadora que utiliza tecnologías de vanguardia para reciclar plástico, aluminio y latón de manera eficiente y sostenible. Uno de los aspectos clave para mejorar el uso y gestión de estos dispositivos es un sistema el cual corre bajo una aplicación web desarrollada con Java EE, lo que permite una interfaz intuitiva y fácil de usar para los operadores.

Por último, todos los datos generados por la máquina son almacenados y gestionados en una base de datos en red hecha con MySQL. Esto permite una gestión eficiente de la información y un análisis detallado de los datos para optimizar el funcionamiento de la máquina.

En resumen, el sistema utiliza Java EE y MySQL de manera complementaria para proporcionar un sistema eficiente y efectivo que permite gestionar los usuarios, dispositivos de reciclaje y tarjetas registradas dentro de la base de datos.

CAPÍTULO II

MARCO TEÓRICO

2.1 INTRODUCCION

En este capítulo se presentarán los conocimientos, con el fin de percibir las tecnologías usadas e implementadas dentro del proyecto y darles la importancia que requieren dentro del mismo. Inicialmente se exhiben los conceptos para comprender la arquitectura de la aplicación web. Seguidamente los lenguajes de programación usados con la relevancia dentro del campo del desarrollo web y del desarrollo del Backend. Igualmente se tocan temas de técnicas del análisis de la información. Para concluir se muestra la vinculación de la arquitectura con los lenguajes de programación y conceptos como el análisis de información para formar el producto.

2.2 Aplicación Web

Según Luján Mora, S. (2002), en la ingeniería de software se denomina aplicación web a aquellas herramientas que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador.

Las aplicaciones web son populares debido a lo práctico del navegador web como cliente ligero, a la independencia del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales. Existen aplicaciones como los web mails, wikis, web logs, tiendas en línea, etc.

Es importante mencionar que una página Web puede contener elementos que permiten una comunicación activa entre el usuario y la información. Esto permite que el usuario acceda a los datos de modo interactivo, gracias a que la página responderá a cada una de sus acciones, como por ejemplo rellenar y enviar formularios, participar en juegos diversos y acceder a gestores de base de datos de todo tipo.

2.2.1 Antecedentes de la Aplicación Web

Según Luján Mora, S. (2002), en los primeros tiempos de la computación cliente-servidor, cada aplicación tenía su propio programa cliente que servía como interfaz de usuario que tenía que ser instalado por separado en cada ordenador personal de cada usuario. El cliente realizaba peticiones a otro programa -el servidor- que le daba respuesta. Una

mejora en el servidor, como parte de la aplicación, requería normalmente una mejora de los clientes instalados en cada ordenador personal, añadiendo un coste de soporte técnico y disminuyendo la productividad.

A diferencia de lo anterior, las aplicaciones web generan dinámicamente una serie de páginas en un formato estándar, como HTML o XHTML, soportados por los navegadores web comunes. Se utilizan lenguajes interpretados en el lado del cliente, directamente o a través de plugins tales como JavaScript, Java, Flash, etc., para añadir elementos dinámicos a la interfaz de usuario. Generalmente cada página web en particular se envía al cliente como un documento estático, pero la secuencia de páginas ofrece al usuario una experiencia interactiva. Durante la sesión, el navegador web interpreta y muestra en pantalla las páginas, actuando como cliente para cualquier aplicación web.

2.2.2 Interfaz

De acuerdo con Luján Mora, S. (2002). Las interfaces web tienen ciertas limitaciones en las funcionalidades que se ofrecen al usuario. Hay funcionalidades comunes en las aplicaciones de escritorio como dibujar en la pantalla o arrastrar-y-soltar que no están soportadas por las tecnologías web estándar. Los desarrolladores web generalmente utilizan lenguajes interpretados (scripts) en el lado del cliente para añadir más funcionalidades, especialmente para ofrecer una experiencia interactiva que no requiera recargar la página cada vez (lo que suele resultar molesto a los usuarios). Recientemente se han desarrollado tecnologías para coordinar estos lenguajes con las tecnologías en el lado del servidor. Como ejemplo, AJAX es una técnica de desarrollo web que usa una combinación de varias tecnologías.

2.2.3 Consideraciones técnicas

De acuerdo con Ramos Martín, A., & Ramos Martín, M. J. (2014). Una ventaja significativa es que las aplicaciones web deberían funcionar igual independientemente de la versión del sistema operativo instalado en el cliente. En vez de crear clientes para Windows, Mac OS X, GNU/Linux y otros sistemas operativos, la aplicación web se escribe una vez y se ejecuta igual en todas partes. Sin embargo, hay aplicaciones inconsistentes escritas con HTML, CSS, DOM y otras especificaciones estándar para navegadores web que pueden causar problemas en el desarrollo y soporte de estas aplicaciones, principalmente debido a la falta de adhesión de los navegadores a dichos estándares web (especialmente versiones de

Internet Explorer anteriores a la 7.0). Adicionalmente, la posibilidad de los usuarios de personalizar muchas de las características de la interfaz (tamaño y color de fuentes, tipos de fuentes, inhabilitar JavaScript) puede interferir con la consistencia de la aplicación web.

Otra aproximación es utilizar Adobe Flash Player o Java applets para desarrollar parte o toda la interfaz de usuario. Como casi todos los navegadores incluyen soporte para estas tecnologías (usualmente por medio de plugins), las aplicaciones basadas en Flash o Java pueden ser implementadas con aproximadamente la misma facilidad. Dado que ignoran las configuraciones de los navegadores, estas tecnologías permiten más control sobre la interfaz, aunque las incompatibilidades entre implementaciones Flash o Java puedan crear nuevas complicaciones, debido a que no son estándares. Por las similitudes con una arquitectura cliente-servidor, con un cliente "no ligero", existen discrepancias sobre el hecho de llamar a estos sistemas “aplicaciones web”; un término alternativo es “Aplicación Enriquecida de Internet”.

2.2.4 Estructura de las aplicaciones web

De acuerdo con Ramos Martín, A., & Ramos Martín, M. J. (2014). Aunque existen muchas variaciones posibles, una aplicación web está normalmente estructurada como una aplicación de tres-capas. En su forma más común, el navegador web ofrece la primera capa, y un motor capaz de usar alguna tecnología web dinámica, por ejemplo: PHP, Java Servlets o ASP, ASP.NET, CGI, ColdFusion, embPerl, Python o Ruby on Rails que constituye la capa intermedia. Por último, una base de datos constituye la tercera y última capa.

El navegador web manda peticiones a la capa intermedia que ofrece servicios valiéndose de consultas y actualizaciones a la base de datos y a su vez proporciona una interfaz de usuario.

2.2.5 Uso empresarial

Según De acuerdo con Ramos Martín, A., & Ramos Martín, M. J. (2014). Una estrategia que está emergiendo para las empresas proveedoras de software consiste en proveer acceso vía web al software. Para aplicaciones previamente distribuidas, como las aplicaciones de escritorio, se puede optar por desarrollar una aplicación totalmente nueva o

simplemente por adaptar la aplicación para ser usada con una interfaz web. Estos últimos programas permiten al usuario pagar una cuota mensual o anual para usar la aplicación, sin necesidad de instalarla en el ordenador del usuario. A esta estrategia de uso se la denomina Software como servicio y a las compañías desarrolladoras se les denomina Proveedores de Aplicaciones de Servicio (ASP por sus siglas en inglés), un modelo de negocio que está atrayendo la atención de la industria del software.

2.2.6 Ventajas

- Ahorra tiempo: Se pueden realizar tareas sencillas sin necesidad de descargar ni instalar ningún programa.
- No hay problemas de compatibilidad: Basta tener un navegador actualizado para poder utilizarlas.
- No ocupan espacio en nuestro disco duro.
- Actualizaciones inmediatas: Como el software lo gestiona el propio desarrollador, cuando nos conectamos estamos usando siempre la última versión que haya lanzado.
- Consumo de recursos bajo: Dado que toda (o gran parte) de la aplicación no se encuentra en nuestro ordenador, muchas de las tareas que realiza el software no consumen recursos nuestros porque se realizan desde otro ordenador.
- Multiplataforma: Se pueden usar desde cualquier sistema operativo porque sólo es necesario tener un navegador.
- Portables: Es independiente del ordenador donde se utilice (un PC de sobremesa, un portátil...) porque se accede a través de una página web (sólo es necesario disponer de acceso a Internet). La reciente tendencia al acceso a las aplicaciones web a través de teléfonos móviles requiere sin embargo un diseño específico de los ficheros CSS para no dificultar el acceso de estos usuarios.
- La disponibilidad suele ser alta porque el servicio se ofrece desde múltiples localizaciones para asegurar la continuidad de este.

- Los virus no dañan los datos porque éstos están guardados en el servidor de la aplicación.
- Colaboración: Gracias a que el acceso al servicio se realiza desde una única ubicación es sencillo el acceso y compartición de datos por parte de varios usuarios. Tiene mucho sentido, por ejemplo, en aplicaciones online de calendarios u oficina.
- Los navegadores ofrecen cada vez más y mejores funcionalidades para crear aplicaciones web ricas (Risas).

2.2.7 Desventajas

Según Ramos Martín, A., & Ramos Martín, M. J. (2014). Habitualmente ofrecen menos funcionalidades que las aplicaciones de escritorio. Se debe a que las funcionalidades que se pueden realizar desde un navegador son más limitadas que las que se pueden realizar desde el sistema operativo.

La disponibilidad depende de un tercero, el proveedor de la conexión a internet o el que provee el enlace entre el servidor de la aplicación y el cliente. Así que la disponibilidad del servicio está supeditada al proveedor.

2.2.8 Diferencia entre aplicación web y aplicación de internet enriquecida (RIA)

De acuerdo con Ramos Martín, A., & Ramos Martín, M. J. (2014). Las aplicaciones web se ejecutan nativamente desde el navegador. Pero existen algunas aplicaciones que funcionan desde el navegador, pero además requieren la instalación de un software en el ordenador para poder utilizarse. Estas aplicaciones se denominan Aplicaciones de Internet Ricas. El motivo de usar este software adicional es que hay muchas funcionalidades que los navegadores no pueden ofrecer, y él enriquece a las aplicaciones web ofreciendo dichas funcionalidades.

Ejemplos de funcionalidades que pueden ofrecer los programas online gracias al uso de software instalado:

- Procesamiento de imágenes
- Captura de imágenes
- Uso de webcam / Captura de video

2.3 Arquitectura Cliente Servidor

Lizama Kindley (2016) menciona la arquitectura cliente y el servidor y lo describe como dos computadores donde el cliente pide la información y el servidor es quien se la suministra, siempre y cuando la primera tengo los permisos. El proceso cuando el cliente solicita información o recursos al servidor se llama petición, en el desarrollo web tradicional cada petición recargaba toda la página y había una mayor carga en el servidor. En la nueva web se aplican tecnologías como AJAX donde la página no necesita recargar si no hace una petición de manera Asíncrona haciendo que la web sea mucho más fluida.

Existía una web tradicional donde todo se hacía con PHP y por eso no existía o no había una división entre Backend y Frontend pues todo se presentaba en el servidor y el cliente no hacía proceso alguno, pero eso cambio alrededor del 2008 cuando HTML5 da una nueva visión, y se empieza a usar los términos Backend y Frontend.

2.3.1 Backend

De acuerdo con Alvarado Díaz (2010), son todo aquello que se maneja en el servidor sin mostrar al usuario final nada de información. En desarrollo web, hace referencia a la parte de la web que visualiza el administrador encargado de gestionar los contenidos que les aparecerán a los usuarios en la parte del Frontend. También es la encargada, de la funcionalidad del sitio, la seguridad, la autorización y la autenticación de toda la aplicación web.

2.3.2 Frontend

De la misma forma Alvarado Díaz (2010) lo describe como todo aquello que se maneja en la parte del cliente (el usuario final), es decir todas aquellas tecnologías que corren en el navegador web. Generalizando tres tecnologías HTML5, CSS3, JavaScript.

También es la encargada de proporcionar al usuario comodidad al ocupar el sitio web, para un mejor consumo. Actualmente, tiene más responsabilidades que solo presentar algo agradable al usuario final. Hoy maneja una complejidad que puede hacer fácil la producción de cualquier sitio web.

2.3.3 Web Services

De acuerdo con Ferris, C., & Farrell, J. (2003), define como; un sistema de software diseñado para admitir interacción interoperable de máquina a máquina a través de una red. Tiene una interfaz descrita en un formato procesable por máquina. Otros sistemas interactúan con el servicio web de la manera prescrita por su descripción utilizando mensajes SOAP, que normalmente de utilizan mediante HTTP con una serialización XML junto otros estándares relacionados con la web.

2.3.4 Autenticación

Santiago Cely (2006) lo referiré a la confirmación que se realiza a través de los medios electrónicos de la identidad de un individuo o de un organismo, así como de todas sus operaciones, transacciones y documentos además de las autorías de estos. se refiere al procedimiento a través del cual la persona coloca sus datos de identificación sin equivocarse

2.3.5 Autorización

De acuerdo con Santiago Cely (2006) es un permiso, que da el consentimiento para que otros hagan o dejen de hacer una acción en cierto sistema. Esto viene después de una autenticación y por el cual el proceso del sistema aprueba el acceso de la autenticación dada.

2.3.6 REST

Según Barreiro, A. (2012) Transferencia de representación de estado, (sus siglas en Ingles REpresentational State Transfer). Como lo describe Roy Thomas en: Architectural Styles and the Design of Network-based Software Architectures un servicio REST no es una arquitectura software, sino un conjunto de restricciones que tener en cuenta en la arquitectura software que usaremos para crear aplicaciones web respetando HTTP

2.4 Arquitectura MVC.

Como Reenskaug, T. (2003) define, esta arquitectura como <El módulo que se ocupa de la entrada> de forma similar a como la vista <se preocupa de la salida>. Trygve Reenskaug fue el que implemento el patrón Modelo Vista Controlador. MVC fue introducido en el año 1976 en por Xerox y su aceptación fue creciendo, dando una adaptación a los sistemas que iban saliendo.

La arquitectura Modelo-Vista-Controlador es la aplicación de esta factorización de tres vías, donde los objetos de diferentes clases se hacen cargo de las operaciones relacionadas con el dominio de la aplicación (el modelo), la visualización del estado de la aplicación (la vista) y la interacción del usuario con el modelo y la vista (el controlador). En las interfaces de usuario tendía a consistir en arreglos de cuatro modismos básicos de visualización: párrafos de texto, listas de texto (menús), "botones" de elección y formas gráficas (mapas de bits o píxeles). Estas herramientas también tendían a utilizar tres paradigmas básicos de interacción con el usuario: navegación, inspección y edición.

También imaginamos que la metodología MVC permitiría a los programadores escribir un modelo de aplicación definiendo primero nuevas clases que incorporarían la información específica del dominio de la aplicación especial. A continuación, diseñarían una interfaz de usuario al diseñar una vista compuesta (ventana) para él "conectando" instancias tomadas de las clases de interfaz de usuario predefinidas.

Esta "capacidad de conexión" era deseable no solo para ver modismos, sino también para implementar los paradigmas de control (edición). Aunque ciertamente está relacionado en una aplicación interactiva, existe la ventaja de poder separar la funcionalidad entre cómo se muestra el modelo y los métodos para interactuar con él. El uso de menús emergentes frente a menús fijos, el significado asociado a el teclado y el mouse / teclas de función y la programación de múltiples vistas deben ser opciones que se puedan hacer independientemente del modelo o de sus vistas. Son opciones que pueden dejarse en manos del usuario final cuando corresponda.

Para abordar los problemas descritos anteriormente, se desarrolló la metáfora Modelo-Vista-Controlador y su paradigma de estructuración de aplicaciones para pensar (e implementar) componentes de aplicaciones interactivas. Los modelos son aquellos componentes de la aplicación del sistema que realmente hacen la simulación de trabajo del

dominio de la aplicación). Se mantienen bastante distintos de las vistas, que muestran aspectos de los modelos. Los controladores se utilizan para enviar mensajes al modelo y proporcionar la interfaz entre el modelo con sus vistas asociadas y los dispositivos de interfaz de usuario interactivos (por ejemplo, teclado, mouse). Se puede pensar que cada vista está estrechamente asociada con un controlador, cada uno con exactamente un modelo, pero un modelo puede tener muchos pares de vista / controlador.

2.4.1 Modelo

Como Reenskaug, T. (2003) define. El modelo de una aplicación es la simulación de software específico del dominio o la implementación de la estructura central de la aplicación. Esto puede ser tan simple como un número entero (como el modelo de un contador) o una cadena (como el modelo de un editor de texto), o puede ser un objeto complejo que es una instancia de una subclase.

Debe haber una correspondencia uno a uno entre el modelo y sus partes, por un lado, y el mundo representado tal como lo percibe el propietario del modelo, por otro lado. Por lo tanto, los nodos de un modelo deben representar una parte identificable del problema Los nodos de un modelo deben estar todos en el mismo nivel de problema, es confuso y se considera de mala forma mezclar nodos orientados a problemas (por ejemplo, citas de calendario) con implementación detalles (por ejemplo, párrafos).

2.5.2 Vista

Reenskaug, T. (2003) indica que las vistas se refieren a todo lo gráfico; solicitan datos de su modelo y muestran los datos. No solo contienen los componentes necesarios para la visualización, sino que también pueden contener subvistas y estar incluidas en supervistas. La supervistas proporciona la capacidad de realizar transformaciones gráficas, ventanas y recortes entre los niveles de esta jerarquía de subvistas / supervistas. Los mensajes de pantalla a menudo se transmiten desde la vista de nivel superior (la vista estándar del sistema de la ventana de la aplicación) hasta las subvistas (los objetos de vista utilizados en las subvistas de la vista de herramientas).

Una vista se adjunta a su modelo (o parte del modelo) y obtiene los datos necesarios para la presentación del modelo haciendo preguntas. También puede actualizar el modelo enviando los mensajes correspondientes. Todas estas preguntas y mensajes tienen que estar en la terminología del modelo, por lo que la vista deberá conocer la semántica de los atributos del modelo que representa. (Eso puede, por ejemplo, solicitar el identificador del modelo y esperar una instancia de Text, no puede asumir que el modelo es de la clase Text).

2.5.3 Controlador

Reenskaug, T. (2003) define que los controladores contienen la interfaz entre sus modelos y vistas asociados y los dispositivos de entrada (teclado, dispositivo señalador, hora). Los controladores también se encargan de programar interacciones con otros pares de vista-controlador: rastrean el movimiento del mouse entre las vistas de la aplicación e implementan mensajes para la actividad del botón del mouse y la entrada del sensor de entrada. Aunque los menús se pueden considerar como pares de controlador de vista, normalmente se consideran dispositivos de entrada y, por lo tanto, pertenecen al ámbito de los controladores.

También podemos verlo como el enlace entre un usuario y el sistema. Proporciona al usuario información mediante hacer arreglos para que las vistas relevantes se presenten en los lugares apropiados de la pantalla. Proporciona medios para la salida del usuario al presentarle menús u otros medios para dar comandos y datos. El controlador recibe dicha salida de usuario, la traduce en los mensajes apropiados y pasa estos mensajes a una o más de las vistas.

Un controlador nunca debe complementar las vistas; por ejemplo, nunca debe conectar las vistas de los nodos dibujando flechas entre ellos.

Por el contrario, una vista nunca debe conocer la entrada del usuario, como las operaciones del mouse y pulsaciones de teclas. Siempre debería ser posible escribir un método en un controlador que envíe mensajes a vistas que reproduzcan exactamente cualquier secuencia de comandos de usuario.

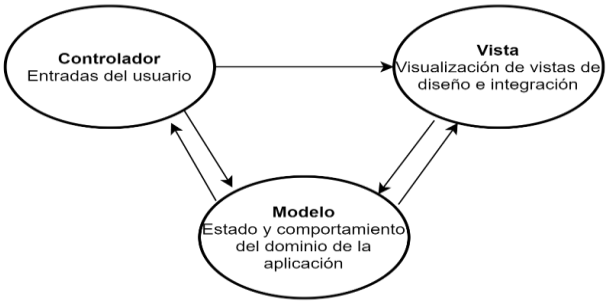


Fig. 3 Modelo Vista Controlador (Fuente: Glenn E. Krasner and Stephen T. Pope 2003)

2.7 Spring Security

Como lo indica Mularien, P. (2010), Spring Security es el módulo del proyecto Spring para incorporar seguridad de acceso a las aplicaciones hechas con Spring Boot. Permite controles de acceso por URL entre otras muchas opciones y es más que suficiente para proteger tu programa.

Spring Security es una librería que forma parte del paraguas del proyecto Spring. Spring tiene más de 25 subproyectos o módulos que aportan funcionalidad que las aplicaciones pueden utilizar si lo creen conveniente. En este caso Spring Security trata de agrupar todas las funcionalidades de control de acceso de usuarios sobre proyectos Spring.

El control de acceso permite limitar las opciones que pueden ejecutar un determinado conjunto de usuarios o roles sobre la aplicación. En esta dirección, Spring Security controla las invocaciones a la lógica de negocios o limita el acceso de peticiones HTTP a determinadas URLs.

Para ello, el programador debe realizar una configuración sobre la aplicación indicando a Spring Security cómo debe comportarse la capa de seguridad. Y aquí es una de las grandes ventajas de Spring Security ya que permite realizar toda una serie de parametrizaciones y ajustes para un gran abanico de posibilidades, permitiendo que el módulo se adapte bien a casi cualquier escenario de aplicaciones realizadas con Spring IoC o Spring Boot.

Si la implementación de Spring Security no fuera suficiente, aun así, proporciona toda una serie de interfaces Java que el programador puede implementar para cambiar el modo de comportarse de una determinada funcionalidad, como el ir a buscar los usuarios a una fuente de datos muy particulares.

Siempre que se pueda, conviene usar las implementaciones provenientes en la librería de Spring Security, ya que es código muy maduro ampliamente testado y además se mantiene actualizado por la comunidad cuando se publica un aviso de seguridad o ante cualquier publicación un nuevo estándar como pasó al liberarse el protocolo HTTP 2.

En definitiva, Spring Security es el método más conveniente para incorporar una capa de seguridad en donde se desea que sólo algunos usuarios tengan acceso a métodos y controladores de una aplicación Spring en base al uso de roles de usuario.

Además, al ser un proyecto maduro y ampliamente utilizado, es posible encontrar implementaciones incluidas de serie para los principales sistemas de seguridad de usuarios más desplegados en el mundo del desarrollo de software o en la administración de sistemas de TI, como LDAP o Kerberos entre otros.

2.7.1 Incluir Spring Security en una aplicación web

De acuerdo con Mularien, P. (2010). Añadir Spring Security en cualquier proyecto Spring de Java es muy sencillo, sobre todo si se ha utilizado la capa de autoconfiguración de Spring MVC y Spring IoC denominada Spring Boot. Si deseas incorporar Spring Security en una aplicación Java o Spring MVC conviene que consideres la migración a Spring Boot antes de proceder con los siguientes pasos, ya que el uso de Spring Boot simplifica enormemente las acciones de configuración que debe realizar el desarrollador para poner en marcha cualquier aplicación Spring. Puedes consultar el artículo tutorial de Spring Boot y Thymeleaf para ver los pasos necesarios en la configuración de una aplicación de este tipo.

Suponiendo que tienes una aplicación Spring Boot en marcha gestionado por Maven, en donde el fichero pom.xml tiene como parent el artefacto spring-boot-starter-parent, lo primero que se debe hacer es incorporar el conjunto de dependencias provenientes de spring-boot-starter-security. Ésta traerá de forma transitiva el resto de las librerías JARs que Spring necesitará para aplicar los mecanismos de seguridad requeridos.

Por sólo incluir la dependencia, si el programador no realiza ninguna configuración adicional, Spring Boot de manera predeterminada protegerá todo el acceso a la aplicación, impidiendo que ningún usuario no identificado pueda invocar a cualquier controlador. Este

mecanismo tan restrictivo suele ser suficiente para pequeñas aplicaciones donde sólo se necesita restringir el acceso de forma general, pero queda algo reducido en aplicaciones donde hay secciones accesibles y otras protegidas, dependiendo de si el usuario está identificado o de si éste tiene un rol determinado.

Como el resto de los aspectos configurables de Spring Boot, en el fichero de configuración de la aplicación, **application.properties** o **application.yml**, hay varias propiedades que pueden ajustarse para controlar el comportamiento base de Spring Security. Las propiedades más importantes de Spring Security se muestra en la Fig. 4

```
spring.ldap.\* = ...                    # propiedades correspondientes a integración con LDAP
spring.security.oauth2.\* = ...         # parámetros para OAuth2 y JWT
spring.session.\* = ...                 # configuraciones para la sesión HTTP, con persistencia SQL opcional

spring.security.user.name = user        # usuario por defecto
spring.security.user.password =         # password por defecto
spring.security.user.roles =            # roles por defecto
```

Fig. 4 Codificación de las propiedades de Spring Security (Fuente: Mularien, P. 2010)

De esta manera, sólo por incluir la dependencia, en el fichero de propiedades de la aplicación puede indicarse un usuario y clave por defecto que podrá usarse para tener acceso a los controladores, y como consecuencia a las diferentes pantallas HTML de la aplicación. Cuando el usuario intente acceder a cualquier URL de la aplicación, Spring Boot y el Security Filter de HTTP redirigirá al usuario al formulario de identificación, donde solicitará al usuario a insertar el nombre y password para proceder. En ese formulario se debe indicar los valores fijados en **spring.security.user.name** y **spring.security.user.password.** Si el usuario ha introducido los valores correctos, se considera al usuario autenticado, y sin tener en cuenta el rol, dejará continuar al usuario con una navegación normal.

Se debe tener en cuenta que la configuración por defecto no incluye el auto registro de usuarios. En el caso de necesitar que los usuarios puedan registrarse automáticamente, por ejemplo, con un email, se deberá programar el mecanismo por el cual se confía en la información proporcionada por el usuario procediendo a la creación de éste en la tabla de datos o fuente de usuarios confiables usada por la aplicación.

2.7.2 Proteger secciones de la aplicación

Según Mularien, P. (2010), Si la protección de forma global a toda la aplicación no se adapta a tus necesidades, probablemente será porque hay secciones que deseas que sean accesibles mientras que otras estén protegidas para un determinado número de usuarios. Si este es el caso, al paso anterior de incluir la dependencia spring-boot-starter-security se debe acompañar con la creación de una clase de configuración de Spring anotada con **@Configuration**, en donde se podrá especificar qué secciones estarán habilitadas y cuáles no en función de los patrones de URL. Por comodidad, conviene que esta clase a crear extienda a **WebSecurityConfigurerAdapter** ya que simplifica la implementación.

Por ejemplo, supóngase que la aplicación web que se está desarrollando tiene una parte pública compuesta por páginas HTML, otra parte protegida y accesible para usuarios autenticados (sin importar su rol) y por último otra sección accesible únicamente para el o los usuarios con el rol de administrador. Para poder llevar a cabo esta configuración se debe segmentar los accesos por URLs de tal manera que permita realizar la siguiente distribución:

/admin/* Patrón de URL que agrupa las funcionalidades del administrador

/user/* Grupo de funcionalidades de los usuarios autenticados

/* Resto de accesos públicos a la aplicación

Para que el mecanismo de control de acceso por URL funcione correctamente conviene hacer esta división por espacios de URL e incluir las secciones más restrictivas primero, dejando al final las más generales. Así se simplifica el número de configuraciones necesarias a realizar. La configuración de Spring Security necesaria para este supuesto quedaría como se muestra en la Figura 5.

```
@Configuration
public class ArtecoCmsSecurityConfig extends WebSecurityConfigurerAdapter {

    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http.authorizeRequests()
                .regexMatchers("/admin/.\*").hasRole("ADMIN")
                .regexMatchers("/user/.\*").authenticated()
            .anyRequest()
                .permitAll()
            .and()
                .formLogin();
    }

}
```

Fig. 5 Codificación de la configuración para la protección de secciones (Fuente: Mularien, P. 2010)

Con esta simple configuración se ha separado el espacio de direcciones que pueden llegar a la web en base a tres conjuntos, los que contienen **/admin** que requerirá que el usuario esté identificado y además disponga del rol de ADMIN como mínimo. La segunda sección delimitada por direcciones que comiencen por **/user** que bastará que el usuario haya pasado por éxito a través del formulario de identificación. Y por último el resto de URLs quedarán abiertas para cualquier usuario identificado o no.

La última línea de configuración correspondiente a **formLogin** habilita el formulario de identificación de usuario, que se interpondrá en el usuario automáticamente si trata de acceder a alguna URL que requiera autenticación o algún ROL.

2.7.3 Administrar los usuarios del programa

Como indica Mularien, P. (2010). Tener un sólo usuario en la aplicación puede ser útil en ciertas ocasiones, aunque desde luego es un escenario algo limitado para aplicaciones abiertas en Internet. Por norma general se usarán varios usuarios, presumiblemente con roles de acceso diferentes para limitar las acciones que pueden realizar cada tipo de usuario.

Para ello necesitamos añadir una fuente de usuarios administrador por la aplicación y que serán usados en el proceso de autenticación por Spring Security. De esta manera, nuestra aplicación puede incorporar un mecanismo de registro de usuarios que puedan autenticarse.

Para que Spring pueda obtener la información de un usuario durante el proceso de login, se solicitará al conjunto de clases Java registradas en el contexto de Spring, de alguna que implemente la interfaz UserDetailsService:

```
package org.springframework.security.core.userdetails;

public interface UserDetailsService {

        UserDetails loadUserByUsername(String username) throws UsernameNotFoundException;

}
```

Fig. 6 Interfaz UserDetailsService (Fuente: Mularien, P. 2010)

Esta interfaz tiene un único método de obligada implementación que es aquel que devuelve los detalles de un usuario UserDetails dado un nombre de usuario, el que trata de ingresar en la aplicación. Por tanto, será obligatorio implementar un bean con esa interfaz dando cuerpo a ese método, por ejemplo, como se muestra en la Figura 7.

```
@Override
public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {
    Optional<MyUser> userDb = userRepository.findByUsername(username);
    if (userDb.isPresent()) {
        MyUser user = userDb.get();
        List<Permission> permissions = user.getPermissions();
        Set<String> roles = new HashSet<>();
        if (!CollectionUtils.isEmpty(permissions)) {
            for (Permission p : permissions) {
                roles.add(p.getRol().name());
            }
        }
        return org.springframework.security.core.userdetails.User
                .withUsername(username)
                .roles(roles.toArray(new String\[0\]))
                .password(user.getPassHash())
                .build();
    } else {
        throw new UsernameNotFoundException("Usuario no encontrado!");
    }
}
```

Fig. 7 Implementación del método UserDetails (Fuente: Mularien, P. 2010)

La implementación de este bean dependerá de dónde tenemos registrados los usuarios. Normalmente éstos estarán en una tabla de la base de datos, así que ya sea con SQL nativo o usando JPA con EntityManager o a través de Spring Data Repositories deberemos consultar la fila asociada según el nombre de usuario. Una vez localizado sólo restará

convertir o mapear ese objeto a uno esperado por Spring, del tipo UserDetails. Por último, en la clase de configuración de Spring Security registraremos nuestra clase recién creada como se muestra en la Figura 8.

```
@Configuration
public class ArtecoCmsSecurityConfig extends WebSecurityConfigurerAdapter {

    @Override
    protected void configure(HttpSecurity http) throws Exception {
        // ... código omitido
    }

    @Autowired
    public void configureGlobal(AuthenticationManagerBuilder auth) throws Exception {
        auth.userDetailsService(myUserDetails()).passwordEncoder(new BCryptPasswordEncoder());
    }

    @Autowired
    public MyUserDetails myUserDetails(UserRepository userRepository) throws Exception {
        return new MyUserDetails(userRepository);
    }

}
```

Fig. 8 Codificación para registrar una clase (Fuente: Mularien, P. 2010)

2.8 Hibernate

Bauer, C., & King, G. (2005) indica que Hibernate es una herramienta de mapeo objeto-relacional (ORM) bajo licencia GNU LGPL para Java, que facilita el mapeo de atributos en una base de datos tradicional, y el modelo de objetos de una aplicación mediante archivos declarativos o anotaciones en los beans de las entidades que permiten establecer estas relaciones. Todo lo dicho, que suena a vendedor de seguros, se resume en que agiliza la relación entre la aplicación y nuestra base de datos SQL, de un modo que optimiza nuestro flujo de trabajo evitando caer en código repetitivo.

Imagina un programa sencillo. Necesitas un método que permita dar de alta, baja, o modificar los datos de usuarios. Estos datos se almacenan en una base de datos, y cada objeto tiene diferentes clases. Para cada objeto debemos crear una clase que al menos permita crear, insertar, eliminar, consultar o modificar la información contenida en sus atributos. Esto, con excepción de alguna consulta un poco especial, es siempre lo mismo (SELECT, UPDATE, CREATE, FROM, WHERE, GROUP BY, ORDER BY y etc.).

2.8.1 Historia Hibernate

Hibernate comenzó en el año 2001 por Gavin King con colegas de Cirrus Technologies, como una alternativa al estilo de programación de EJB2, que usa beans como entidades. El objetivo original era ofrecer mejores capacidades de persistencia que las ofrecidas por EJB2, simplificar las complejidades, y complementar algunas características necesarias.

A principios de 2003, el equipo de desarrollo de Hibernate lanzó Hibernate 2 y sus subversiones sucesivas, que ofrecieron muchas mejoras significativas sobre el primer lanzamiento. JBoss, Inc. (ahora parte de Red Hat) contrató a los líderes de desarrollo de Hibernate para continuar su desarrollo.

2.8.2 Funcionamiento de Hibernate

Como indica Bauer, C., & King, G. (2005). En la mayoría de programas actuales existen dos modelos de datos que coexisten: el usado en la memoria de la computadora (orientación a objetos), y el usado en las bases de datos (modelo relacional). Hibernate busca solucionar el problema de la diferencia entre ambos modelos. Para lograr esto, nos permite detallar cómo es su modelo de datos, qué relaciones existen y qué forma tienen mediante un

documento XML, o mediante anotaciones donde corresponde un atributo de una clase, con una columna de una tabla. En la actualidad es una tarea simple, pues existen herramientas que lo hacen por nosotros.

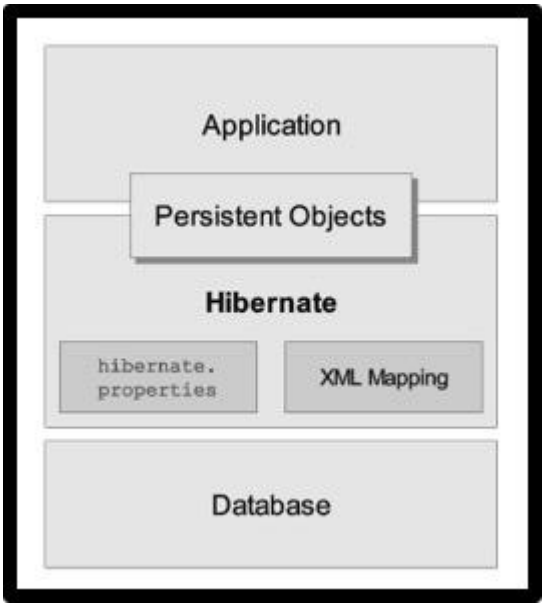


Fig. 9 Modelo de las capas que maneja Hibernate (Fuente: Bauer, C., & King, G., 2005)

Con esta información, Hibernate permite a la aplicación manipular los datos en la base de datos operando sobre objetos, con todas las características de la POO. Hibernate convertirá los datos entre los tipos utilizados por Java y los definidos por SQL. Hibernate genera las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias, manteniendo la portabilidad entre todos los motores de bases de datos con un ligero incremento en el tiempo de ejecución.

A la hora de programar con Hibernate, se debe tener en cuenta unas particularidades clave pues a la hora de trabajar con datos; este framework hace uso de los siguientes objetos

2.8.2.1 Objeto de Configuración Hibernate

El objeto de configuración es el primer objeto de Hibernate y generalmente se crea una sola vez durante la inicialización de la aplicación. Representa un archivo de configuración o propiedades requeridas y proporciona dos componentes claves:

- Conexión de base de datos
- Configuración de Mapeo Clase

2.8.2.2 Objetivo SessionFactory

Objeto de configuración se utiliza para crear un objeto SessionFactory para la aplicación que utiliza el archivo de configuración suministrada, y permite un objeto Session ser ejecutado.

2.8.2.3 Objeto Session

Se utiliza para obtener una conexión física con una base de datos. Los objetos de sesión no deben mantenerse abiertas durante mucho tiempo, ya que no suelen ser seguros para subprocesos y deben ser creados y destruidos, según sea necesario.

2.8.2.4 Objeto Transaction

Este es un objeto opcional y las aplicaciones de Hibernate puede optar por no utilizar esta interfaz, en lugar gestionar las transacciones en su propio código de la aplicación.

2.8.2.4 Objeto Query

Objetos de consulta utilizan SQL o Hibernate Query Language (HQL) cadena para recuperar datos de la base de datos y crear objetos. Una instancia de consulta se utiliza para enlazar los parámetros de consulta, limitar el número de resultados devueltos por la consulta, y finalmente, para ejecutar la consulta.

2.8.2.4Objeto Criteria

Los objetos Criteria se utilizan para crear y ejecutar consultas con objetos y recuperar objetos.

2.9 JPA

Según Keith, J. (2011). La persistencia de datos es un medio mediante el cual una aplicación puede recuperar información desde un sistema de almacenamiento no volátil y hacer que esta persista. La persistencia de datos es vital en las aplicaciones empresariales debido al acceso necesario a las bases de datos relacionales. Las aplicaciones desarrolladas para este entorno deben gestionar por su cuenta la persistencia o utilizar soluciones de terceros para manejar las actualizaciones y recuperaciones de las bases de datos con persistencia. JPA (Java™ Persistence API) proporciona un mecanismo para gestionar la persistencia y la correlación relacional de objetos y las funciones desde las especificaciones EJB 3.0.

La especificación JPA define la correlación relacional de objetos internamente, en lugar de basarse en implementaciones de correlación específicas del proveedor. JPA se basa en el modelo de programación Java que se aplica a entornos Java Enterprise Edition (Java EE), pero JPA puede funcionar dentro de un entorno Java SE para probar funciones de aplicación.

JPA representa una simplificación del modelo de programación de persistencia. La especificación JPA define explícitamente la correlación relacional de objetos, en lugar de basarse en implementaciones de correlación específicas del proveedor. JPA crea un estándar para la importante tarea de la correlación relacional de objetos mediante la utilización de anotaciones o XML para correlacionar objetos con una o más tablas de una base de datos. Para simplificar aún más el modelo de programación de persistencia:

La API EntityManager puede actualizar, recuperar, eliminar o aplicar la persistencia de objetos de una base de datos.

La API EntityManager y los metadatos de correlación relacional de objetos manejan la mayor parte de las operaciones de base de datos sin que sea necesario escribir código JDBC o SQL para mantener la persistencia.

JPA proporciona un lenguaje de consulta, que amplía el lenguaje de consulta EJB independiente, conocido también como JPQL, el cual puede utilizar para recuperar objetos sin grabar consultas SQL específicas en la base de datos con la que está trabajando.

JPA está diseñado para funcionar tanto dentro como fuera de un contenedor Java Enterprise Edition (Java EE). Cuando se ejecuta JPA dentro de un contenedor, las

aplicaciones pueden utilizar el contenedor para gestionar el contexto de persistencia. Si no hay ningún contenedor para gestionar JPA, la aplicación debe manejar ella misma la gestión del contexto de persistencia. Las aplicaciones diseñadas para la persistencia gestionada por contenedor no requieren tanta implementación de código para manejar la persistencia, pero estas aplicaciones no se pueden utilizar fuera de un contenedor. Las aplicaciones que gestionan su propia persistencia pueden funcionar en un entorno de contenedor o en un entorno Java SE.

Los contenedores Java EE que dan soporte al modelo de programación EJB 3.x deben dar soporte a una implementación JPA, también denominada proveedor de persistencia. Un proveedor de persistencia JPA utiliza los elementos siguientes para habilitar una gestión de persistencia más fácil en un entorno EJB 3.x

2.9.1 Unidad de persistencia

Como indica Keith, J. (2011), la unidad de persistencia define un modelo relacional de objetos completo que correlaciona clases Java (entidades + estructuras de soporte) con una base de datos relacional. EntityManagerFactory utiliza estos datos para crear un contexto de persistencia al que se puede acceder mediante EntityManager.

2.9.2 EntityManagerFactory

Según Keith, J. (2011) Se utiliza para crear un EntityManager para las interacciones de base de datos. Los contenedores del servidor de aplicaciones normalmente proporcionan esta función, pero es necesario EntityManagerFactory si se utiliza la persistencia gestionada por las aplicaciones JPA. Una instancia de EntityManagerFactory representa un contexto de persistencia.

2.9.3 Contexto de persistencia

Keith, J. (2011) Define que el contexto de persistencia es el conjunto de las instancias activas que la aplicación está manipulando actualmente. Puede crear el contexto de persistencia manualmente o mediante inyección.

2.9.4 EntityManager

De igual manera Keith, J. (2011) indica que es Gestor de recursos que mantiene la colección activa de objetos de entidad que está utilizando la aplicación. EntityManager maneja la interacción y metadatos de bases de datos para las correlaciones relacionales de objetos. Una instancia de EntityManager representa un contexto de persistencia. Una aplicación de un contenedor puede obtener EntityManager mediante inyección en la aplicación o buscándola en el espacio de nombres de componente Java. Si la aplicación gestiona su persistencia, el EntityManager se obtiene desde EntityManagerFactory.

2.9.5 Objetos de entidad

Según Keith, J. (2011) es una clase Java simple que representa una fila de una tabla de base de datos en su forma más simple. Los objetos de entidades pueden ser clases concretas o clases abstractas. Mantienen estados mediante la utilización de propiedades o campos.

2.10 Lenguajes de desarrollo implementados

2.10.1 Java

Groussard, T. (2012) define Java como un lenguaje de propósito general capaz de acometer todo tipo de proyectos y ejecutarse en múltiples plataformas. Aquí aprenderás qué es Java y a programar en este lenguaje con diversos manuales.

Java es un lenguaje de programación de propósito general, uno de los más populares y con mayores aplicaciones del panorama actual. Existen diversos índices de lenguajes de programación y dependiendo el que tomemos como referencia puede considerarse el lenguaje más popular, o uno de los 3 más populares que existen en el mundo.

Fue creado inicialmente por la compañía Sun Microsystems que consiguió posicionar su lenguaje como uno de los más punteros y extendidos, debido sobre todo a su versatilidad y soporte prácticamente universal. Actualmente se encuentra en propiedad de Oracle, después que ésta adquiriera a Sun.

Java es multiplataforma, capaz de ejecutarse en la mayoría de los sistemas operativos y dispositivos, con una única base de código. Esto lo consigue gracias a una máquina virtual

que existe en cada sistema que es capaz de ejecutar Java y hacer de puente entre el lenguaje de programación y el dispositivo. Eso quiere decir que, si hacemos un programa en Java podrá funcionar en cualquier ordenador, dispositivo o cualquier tipo de máquina que soporte Java.

El hecho de ser multiplataforma es una ventaja significativa para los desarrolladores de software, pues anteriormente era necesario hacer un programa para cada sistema operativo, por ejemplo, Windows, Linux, MacOS, etc. Esto lo consigue porque se ha creado una Máquina virtual de Java para cada plataforma, que hace de puente entre el sistema operativo y el programa de Java y posibilita que este último se entienda perfectamente. Hoy hay muchos lenguajes multiplataforma, pero Java fue de los primeros en ofrecer esta posibilidad.

Con Java podemos hacer todo tipo de proyectos, desde aplicaciones web a servicios web basados en SOAP o REST, aplicaciones de escritorio de consola o interfaz gráfica. Además, Java es el lenguaje de programación que se usa para el desarrollo nativo para Android, lo que ha llevado a esta tecnología a un nivel de popularidad todavía mayor y con alta demanda profesional.

2.10.2 JSP: Java Server Pages

Como indica Morales, C. R. (2004). JSP es una tecnología que permite desarrollar sitios y aplicaciones web tradicionales, en las que el servidor ofrece el HTML directamente para que el navegador lo renderice al usuario.

En el ámbito de Java JSP sería la alternativa más sencilla para programar páginas web dinámicas del lado del servidor, con accesos a bases de datos, sistema de archivos, tratamiento de XML y con cualquier tipo de conexión de red entre sistemas, etc. JSP es muy práctico ya que permite programar con Java y devolver HTML para el consumo directo del navegador, sin embargo, podemos decir que la tecnología en si nunca tuvo demasiado tirón.

JSP sería un competidor de PHP, pero requiere el uso de un servidor compatible con Servlets de Java, una configuración que, si bien es soportada por un servidor popular como Apache, lo cierto es que no es nada habitual. Otro servidor conocido que soporta JSP es Tomcat. Sin embargo, este tipo de configuraciones en los servidores de Internet no están muy extendidas requieren disponer de un servidor propio para poder instalarlos, lo que no facilitó su penetración en el mercado.

2.10.3 XHTML

Equipo Vértice. (2009) indique que el lenguaje XHTML (Extensible Hypertext Markup Language), a grandes rasgos, es un lenguaje de marcado que te permite editar webs. Está basado en HTML 4.01, pero sus reglas son mucho más estrictas y versátiles que HTML. XHTML es una nueva formulación de HTML que utiliza el metalenguaje de marcado XML, lo cual es una ventaja clave en su usabilidad porque muchos formatos de datos están basados en XML y los dispositivos modernos requieren una versión estricta del lenguaje de marcado porque no pueden mostrar mal el código fuente.

XHTML ha sido diseñado para extender HTML y permitir la compatibilidad con nuevos formatos de datos como videos, imágenes o lenguajes de scripting. Dado que HTML siempre se basó en SGML y, por lo tanto, era relativamente complejo, se encontró que un lenguaje meta de marcado limitado, como XML, era adecuado para definir la estructura básica y las posibilidades sintácticas del XHTML. Las especificaciones de XHTML 1.0 fueron publicadas por el W3C en 2002. A finales de 2009 se interrumpió el trabajo sobre la versión 2.0 porque el Grupo de Trabajo de Tecnología de Aplicación de Hipertexto Web (WHATWG) estaba trabajando en HTML5 al mismo tiempo, y este último fue el que al final prevaleció.

2.10.3.1 características

Según Equipo Vértice. (2009) XHTML se utiliza para marcar contenido como texto, imágenes y enlaces en forma de hipervínculos para crear una cierta estructura que puede ser mostrada por los navegadores. Los documentos pueden ser estructurados con XHTML para hacerlos legibles para un analizador. El analizador interpreta los elementos de marcado especificados en las definiciones del lenguaje XHTML y reproduce el contenido de estos elementos de una manera específica.

En última instancia, el usuario ve lo que contienen estos elementos y no la forma en que se anotan o estructuran los elementos. Este último se llama código fuente XHTML y puede ser editado con diferentes editores de texto, por ejemplo, con un editor WYSIWYG. La notación separa el contenido y el diseño. El contenido está encerrado por los elementos. La forma en que se muestra este contenido está determinada por los elementos y sólo la muestra el navegador.

Antes de crear un documento XHTML, debe seleccionar una Definición de tipo de documento. Hay tres tipos, cada uno con ciertas ventajas y desventajas, transicional, estricto y frameset. Al mismo tiempo, la versión del lenguaje debe ser especificada, como XHTML 1.0 o 2.0.

El marco básico de un archivo XHTML debe ajustarse exactamente a las reglas, de lo contrario no es WFF (fórmula bien formada), que es un código fuente válido que cumple con los requisitos de lenguaje y definiciones de documentos y que puede ser leído por un navegador.

Las diferencias importantes entre XHTML y HTML son, por ejemplo, la declaración XML, la definición del tipo de documento, que está en minúsculas, la conformidad exacta con el marco básico y el hecho de que cada elemento abierto debe cerrarse de nuevo. Si una de estas reglas no se adhiere en XHTML, el navegador no puede mostrar el documento, lo que probablemente pueda hacer con HTML.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
      "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>An XHTML document</title>
</head>
<body>
< h1 >Headline of the first order< /h1 >
< p >A paragraph with any text content< /p >
< h2 >Secondary heading< /h2 >
< p >Another paragraph with any text content< /p >
</body>
</html>
```

Fig. 10 Ejemplo de un código de XHTML (Fuente: Enriques V. 2009)

2.11 MySQL

De acuerdo con Morteo y Bocalandro autores del libro “Un enfoque práctico de MySQL” (2004); MySQL es un lenguaje de dominio específico utilizado en programación, diseñado para administrar, y recuperar información de sistemas de gestión de bases de datos relacionales. Una de sus principales características es el manejo del álgebra y el cálculo relacional para efectuar consultas con el fin de recuperar, de forma sencilla, información de bases de datos, así como realizar cambios en ellas.

Originalmente basado en el álgebra relacional y en el cálculo relacional, MySQL consiste en un lenguaje de definición de datos, un lenguaje de manipulación de datos y un lenguaje de control de datos. El alcance de MySQL incluye la inserción de datos, consultas, actualizaciones y borrado, la creación y modificación de esquemas y el control de acceso a los datos.

Es un lenguaje declarativo de "alto nivel" o "de no procedimiento" que, gracias a su fuerte base teórica y su orientación al manejo de conjuntos de registros —y no a registros individuales— permite una alta productividad en codificación y la orientación a objetos. De esta forma, una sola sentencia puede equivaler a uno o más programas que se utilizarían en un lenguaje de bajo nivel orientado a registros. MySQL también tiene las siguientes características:

Lenguaje de definición de datos: El Lenguaje de Definición de datos (LDD) de SQL proporciona comandos para la definición de esquemas de relación, borrado de relaciones y modificaciones de los esquemas de relación.

Lenguaje interactivo de manipulación de datos: El lenguaje de manipulación de datos (LMD) de MySQL incluye lenguajes de consultas basado tanto en álgebra relacional como en cálculo relacional de tuplas.

Integridad: El LDD de MySQL incluye comandos para especificar las restricciones de integridad que deben cumplir los datos almacenados en la base de datos.

Definición de vistas: El LDD incluye comandos para definir las vistas.

Control de transacciones: MySQL tiene comandos para especificar el comienzo y el final de una transacción.

MySQL incorporado y dinámico: Esto quiere decir que se pueden incorporar instrucciones de MySQL en lenguajes de programación.

Autorización: El LDD incluye comandos para especificar los derechos de acceso a las relaciones y a las vistas.

Algunos de los tipos de datos básicos de MySQL son:

- Varchar Recibe cadena de palabras compuestas de letras, números y caracteres especiales.
- Int es el principal tipo de datos de valores enteros de SQL Server. Con números enteros con o sin signo
- Date: una fecha de calendario que contiene el año (de cuatro cifras), el mes y el día.
- Time: La hora del día en horas minutos segundos (el valor predeterminado es 0).

También el lenguaje de definición de datos es el que se encarga de la modificación de la estructura de los objetos de la base de datos. Incluye órdenes para modificar, borrar o definir las tablas en las que se almacenan los datos de la base de datos. Existen cuatro operaciones básicas:

- **CREATE**: permite crear objetos de datos, como nuevas bases de datos, tablas y vistas.
- **ALTER:** Modifica la estructura de una tabla u objeto. Se puede agregar o quitar campos a una tabla, modificar el tipo de un campo, agregar o quitar índices a una tabla, modificar un disparador de acción, etc.
- **DROP:** Elimina un objeto de la base de datos, puede ser una vista, una tabla o un índice.
- **TRUNCATE**: Su función es borrar el contenido completo de una tabla especificada en el comando. Solo elimina el contenido, no modifica ni elimina el objeto de la tabla.
- **SELECT:** Muestra información deseada cuando se establece el nombre de una tabla o un objeto que puede ser utilizado complementariamente.

2.12 FRAMEWORKS

2.12.1 JSF Java Server Faces

Jeronimo, S. U. (2012) indica que la tecnología Java Server Faces es un framework de interfaz de componentes de usuarios del lado del servidor para las aplicaciones web basadas en la tecnología Java. Los principales componentes de la tecnología JSF son:

- Una API para representar componentes de Interfaz de Usuario (UI) y gestionar su estado.
    - Manejar eventos, validar en el servidor y conversión de datos.
    - Definir la navegación de páginas.
    - Soporte de internacionalización y accesibilidad.
- Dos librerías de etiquetas JSP personalizadas para expresar componentes en una página JSP y enlazar los componentes a objetos del servidor.

El modelo de programación bien definido y las librerías de etiquetas facilitan la construcción y mantenimiento de las aplicaciones web con Interfaces de Usuario (UI) de servidor. Con un mínimo esfuerzo se podría:

- Poner componentes en una página mediante etiquetas de componentes.
- Enlazar eventos generados por componentes con código de la aplicación en el servidor.
- Relacionar componentes UI en una página con datos del servidor.
- Construir una UI con componentes reutilizables y extensibles.
- Salvar y restaurar el estado de la UI más allá de la vida de las peticiones.

2.12.2 Bootsfaces

Valenzuela Chancosa, M. L. (2017) indica que Bootsfaces es una implementación de JSF que incluye Bootstrap 3 y jQuery en un set de tags increíblemente fáciles de utilizar. Rediseñaremos nuestra aplicación de blog para que utilice Bootsfaces.

2.12.3 PrimeFaces

De igual manera Valenzuela Chancosa, M. L. (2017) menciona que PrimeFaces es una biblioteca de componentes para JavaServer Faces (JSF) de código abierto que cuenta con un conjunto de componentes enriquecidos que facilitan la creación de las aplicaciones web. Primefaces está bajo la licencia de Apache License V2. Una de las ventajas de utilizar Primefaces, es que permite la integración con otros componentes como por ejemplo RichFaces.

2.13 Sistemas de gestión de bases de datos (SGBD)

De acuerdo con C.J. Date (2012) la gestión de base de datos (SGBD) es una labor ya antigua pero que se ha vuelto imprescindible debido a la gran cantidad de información y datos que se maneja hoy en día. La adopción del concepto de base de datos se dio con los primeros sistemas electrónicos computacionales.

Ahora bien, un gestor de base de datos o sistema de gestión de base de datos (SGBD) hace referencia al software que manipula esta base de datos y es la interfaz de acceso de parte del usuario para poder trabajar con esta información en diferentes niveles: consulta, análisis, almacenamiento y modificación.

Para que un SGBD pueda operar adecuadamente es necesario que los datos almacenados tengan un hilo conductor de relación, estructura y organización para crear ecosistemas que faciliten el acceso y la gestión. Esto último es de gran importancia para permitir la óptima operación, independientemente tanto del software como de los datos.

También debe entenderse que el gestor de datos o SGBD será el mecanismo que brindará las herramientas adecuadas para controlar la base de datos por parte del usuario (herramientas como generación de informes o sistemas de búsqueda). Incluso sistemas más acabados permiten tener perspectivas de aplicación o definición de la información obtenida. Por tanto, los datos siempre se mantienen seguros, íntegros y consistentes (sin ningún tipo de modificación).

Los tipos de SGBD existentes se agrupan de acuerdo con la relación de criterios del modelo de los datos. En relación con si se es propietario o no y dependiendo de la cantidad de usuario o sitios, así los tipos de bases de datos son:

Relacionales. Modelo más empleado por empresas en el que, como el nombre indica, los datos mantienen cierta relación.

Jerárquico. Un modelo con una estructura de estilo de árbol. Sin embargo, este tipo no ofrece flexibilidad y hay redundancia de datos, aunque gestiona gran cantidad de datos.

De red. Modelo similar al jerárquico, pero su diferenciación se debe a una mayor complejidad. Así, los registros pueden mantener relación por distintas vías.

Multidimensionales. Este es un tipo similar al relacional, pero en lugar de mantener una relación bidimensional, entre columna y fila, se presenta un número indeterminado de columnas y filas para presentar una indeterminada cantidad de dimensiones y relaciones.

De objetos. Un modelo más reciente cuyo principal objetivo es intentar almacenar en la base de datos objetos completos, integrando además conceptos del paradigma. Así, este tipo puede almacenar voz, texto, imágenes y hasta gráficos.

Transaccionales. Modelo cuya principal finalidad es el correcto y rápido traslado de datos para envió y recepción. Y un claro ejemplo son las transacciones bancarias.

Documentales. Un modelo específico para indexación completa de texto cuyo objetivo es almacenar grandes cantidades de información (como acervos históricos).

Deductivas. Modelo que se diferencia del resto al obtener deducciones a partir de ciertas inferencias y cuya función depende de algunas reglas y hechos que se almacenan en la base de datos.

CAPÍTULO III

DESARROLLO

3.1 INTRODUCCIÓN

En este capítulo se presentará todo el desarrollo del proyecto, en el cual se presenta la situación actual en donde se describe por qué se crea y la utilidad final. Posteriormente el análisis, el plan a seguir para el desarrollo el cual incluye diagramas y justificación del desarrollo.

Con el análisis creado, se recopilan los requisitos del sistema y esto se dividirá en requerimientos funcionales y requerimientos no funcionales.

La situación actual es el estado en el que se encuentran con respecto a los sistemas de este tipo, paneles de control en el momento en el que se inicia su estudio.

Asumiendo en cuenta el objetivo del estudio de la situación actual, se realiza una valoración de la información existente acerca de los sistemas de información afectados. En función de dicha valoración, se especifica el nivel con el que se lleva a cabo el estudio.

3.2 Estudio Situación Actual

Actualmente no existen en el mercado un software que pueda abarcar las funciones requeridas por los dispositivos de reciclaje desarrollados en IntGen Technologies S.A de C.V , por lo cual será un sistema único que se encargue de cubrir las necesidades requeridas por los dispositivos ya mencionados. Es un producto encargado de gestionar los dispositivos de reciclaje por ello los requerimientos son propuestos para cumplir con dichas funciones.

La aplicación reside en un servidor Wildfly el cual se le introdujo las configuraciones necesarias para la conexión a la base de datos para el correcto funcionamiento e interacción del sistema. Y su interacción está basada en el modelo cliente – servidor donde nuestro servidor Wildfly funge como un cliente manteniendo la lógica de negocio y vista en el lado del usuario.

El sistema está siendo considerado en diversas fases, la cual en el desarrollo se encuentra la fase uno, donde abarca el inicio de sesión, creación de la base de datos y presentación de esta.

3.3 Análisis

3.3.1 Cronograma de Actividades

Las actividades presentadas en el cronograma en la Tabla 1 lleva consigo la investigación de las tecnologías necesarias para llevar a cabo el desarrollo, modelo y pruebas para el adecuado funcionamiento y que el desarrollo sea escalable, lo que permite la futura incorporación de módulo y funcionalidades.

Tabla 1. Cronograma de Actividades (Fuente: Propia Autoría, 2022)

ACT	**QUIEN**	**ESTADO**	Semana										0	1	2	3	4	5	6	7
			Horas	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	Mario - Jaziell	Completado																		
2	Mario - Jaziell	Completado																		
3	Mario - Jaziell	Completado																		
4	Mario - Jaziell	En proceso			0	1														
5	Mario - Jaziell	En proceso			0	1														
6	Mario - Jaziell	No Iniciado							2											
7	Mario - Jaziell	No Iniciado							2											
8	Mario - Jaziell	No Iniciado									2	0	0							
9	Mario - Jaziell	No Iniciado									2	2	0	2						

Tabla 1. (Continuación)Cronograma de Actividades (Fuente: Propia Autoría, 2022)

ACT	**QUIEN**	**ESTADO**	Semana										0	1	2	3	4	5	6	7
			Horas	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	Mario - Jaziell	No Iniciado													2					
11	Mario - Jaziell	No Iniciado													2		0			
12	Mario - Jaziell	No Iniciado															0			
13	Mario - Jaziell	No Iniciado															0			
14	Mario - Jaziell	No Iniciado																4	0	0

3.4 Arquitectura

Para la elaboración del sistema es necesario utilizar tecnologías que faciliten y contribuyan al buen funcionamiento del sistema como lo es Hibernate, spring Security y JPA. Para hacer uso de estas tecnologías es necesario hacer en base a un Maven proyecto el cual facilita la instalación de librerías en los proyectos de java, al crear el proyecto en java, las librerías se declaran en un archivo el cual lleva el nombre de poom.xml

```
Registrador/pom.xml
64
65          <dependency>
66              <groupId>org.springframework.security</groupId>
67              <artifactId>spring-security-web</artifactId>
68              <version>5.3.13.RELEASE</version>
69          </dependency>
70          <dependency>
71              <groupId>org.springframework.security</groupId>
72              <artifactId>spring-security-config</artifactId>
73              <version>5.3.13.RELEASE</version>
74          </dependency>
75
76          <dependency>
77              <groupId>org.springframework</groupId>
78              <artifactId>spring-orm</artifactId>
79              <version>5.3.13</version>
80          </dependency>
81
82
83          <!-- https://mvnrepository.com/artifact/org.springframework.security/spring-security-taglibs -->
84          <dependency>
85              <groupId>org.springframework.security</groupId>
86              <artifactId>spring-security-taglibs</artifactId>
87              <version>5.3.13.RELEASE</version>
88          </dependency>
89
90          <!-- https://mvnrepository.com/artifact/org.springframework.webflow/spring-faces -->
91          <dependency>
92              <groupId>org.springframework.webflow</groupId>
93              <artifactId>spring-faces</artifactId>
94              <version>2.5.0.RELEASE</version>
95          </dependency>
96          <dependency>
97              <groupId>org.reactivestreams</groupId>
98              <artifactId>reactive-streams</artifactId>
99              <version>1.0.1</version>
100         </dependency>
101         <!-- https://mvnrepository.com/artifact/org.slf4j/slf4j-reload4j -->
102         <dependency>
103             <groupId>org.slf4j</groupId>
104             <artifactId>slf4j-reload4j</artifactId>
```

Fig. 11 Importación de librerías (Fuente: Propia Autoría, 2022)

Una vez que las tecnologías están instaladas en el proyecto se configura la conexión del jpa con la base de datos creada en MySQL, por medio de un archivo percistence.xml en la cual se declaran los enlaces, contraseñas y usuarios de conexión. Como de menciono anteriormente jpa se basa en la creación de clases que contienen los mismos atributos que las tablas de las bases de datos por lo cual en el archivo *percistence.xml* se definen las clases de las tablas que serán desarrolladas como se muestra en la Fig 12,

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.2" xmlns="http://xmlns.jcp.org/xml/ns/persistence" xmlns:xsi="http://www.w3.org
    <persistence-unit name="reciclador" transaction-type="JTA">
    <provider>org.hibernate.jpa.HibernatePersistenceProvider</provider>
    <jta-data-source>jdbc/MySQLDS</jta-data-source>
    <class>com.integen.reciclador.entity.BiReciladorContenedor</class>
    <class>com.integen.reciclador.entity.TbCodigoError</class>
    <class>com.integen.reciclador.entity.TbConfiguracion</class>
    <class>com.integen.reciclador.entity.TbContenedor</class>
    <class>com.integen.reciclador.entity.TbCuenta</class>
    <class>com.integen.reciclador.entity.TbError</class>
    <class>com.integen.reciclador.entity.TbEstatus</class>
    <class>com.integen.reciclador.entity.TbMaterial</class>
    <class>com.integen.reciclador.entity.TbMovimiento</class>
    <class>com.integen.reciclador.entity.TbPerfil</class>
    <class>com.integen.reciclador.entity.TbReciclador</class>
    <class>com.integen.reciclador.entity.TbTipoMovimiento</class>
    <class>com.integen.reciclador.entity.TbTipoEstatus</class>
    <class>com.integen.reciclador.entity.TbTipoMovimiento</class>
    <class>com.integen.reciclador.entity.TbUsuario</class>
    <class>com.integen.reciclador.entity.TbTarjeta</class>
    <class>com.integen.reciclador.entity.TbDetalleVenta</class>
    <class>com.integen.reciclador.entity.TbProducto</class>
    <class>com.integen.reciclador.entity.TbVenta</class>
        <properties>
            <property name="hibernate.dialect" value="org.hibernate.dialect.MySQL8Dialect"></property>
            <!--  <property name="hibernate.show_sql" value="true"/>
            <property name="hibernate.format_sql" value="true"/>
            <property name="hibernate.use_sql_comments" value="true"/>-->


        </properties>
    </persistence-unit>
</persistence>
```

Fig. 12 Archivo Persistence (Fuente: Propia Autoría, 2022)

Para las cuestiones de seguridad se crean archivos .XML donde se define los parámetros que va a gestionar spring security para la seguridad como se muestra en la Fig 13.

```
<bean id="interceptorDeAutenticacion"
    class="com.intgen.reciclador.seguridad.InterceptorDeAutenticacion" />

<http auto-config="true"
    authentication-manager-ref="AuthManagerPP"
    xmlns="http://www.springframework.org/schema/security"
    use-expressions="true">

    <intercept-url pattern="/login.xhtml" access="permitAll" />
    <intercept-url pattern="/errores/**" access="permitAll" />
    <intercept-url pattern="/login" access="permitAll" />
    <intercept-url pattern="/javax.faces.resource/**"
        access="permitAll" />
    <intercept-url pattern="/app/**"
        access="isAuthenticated()" />
    <intercept-url pattern="/resources/**"
        access="permitAll" />


    <form-login login-page="/login.xhtml"
        default-target-url="/app/ma.mocard.xhtml"
        always-use-default-target="true"
        username-parameter="input_formLogin:username"
        password-parameter="input_formLogin:password"
        authentication-failure-url="/login.xhtml?error=login_failed"
        authentication-success-handler-ref="interceptorDeAutenticacion"
        login-processing-url="/login" />


    <sec:custom-filter ref="logoutFilter"
        after="LOGOUT_FILTER" />
    <csrf disabled="false" />

    <sec:session-management>
        <sec:concurrency-control max-sessions="1" />
```

Fig. 13 Configuración de Seguridad (Fuente: Propia Autoría, 2022)

3.5 Requisitos del sistema

A continuación, se detallarán todas las necesidades o condiciones que debe satisfacer el software. Dichos requisitos serán expresados por las necesidades de la Fase I, expresados con un lenguaje simple para su fácil compresión.

Estos requisitos son de necesidad esencial, solo se trata de la Fase I dado que va a incrementar el resto del proyecto, todos estos requisitos provienen de la necesidad en la industria detectadas con el análisis de datos y el crecimiento de la información.

A la hora de realizar la redacción de los requerimientos se ha evitado utilizar el lenguaje demasiado técnico. Con el tiempo de un lenguaje común, sencillo y claro se pretende que todas las partes implicadas en el proyecto.

Se hace distinción entre dos tipos de requisitos:

- **Funcionales**. Estos requisitos se encargan de especificar las funciones que realiza el sistema.
- **No Funcionales**. Estos requisitos especifican algo sobre el propio sistema, y cómo debe realizar sus funciones.

3.5.1 Requerimientos Funcionales

3.4.1.1 Iniciar Sesión en el Sistema

Los usuarios pueden iniciar sesión en base a un correo y una contraseña.

Tabla. 2 Tabla de Requerimiento Iniciar Sesión en el Sistema (Fuente: Propia Autoría, 2022)

Nombre: Iniciar Sesión en el Sistema	
PRIORIDAD: Alta	
NECESIDAD: Esencial	
CLARIDAD: Alta	**VERIFICABILIDAD:** Alta
ESTABILIDAD: Toda la vida del software	
DESCRIPCIÓN: Los usuarios pueden iniciar sesión dentro de la plataforma con su correo electrónico y contraseña	

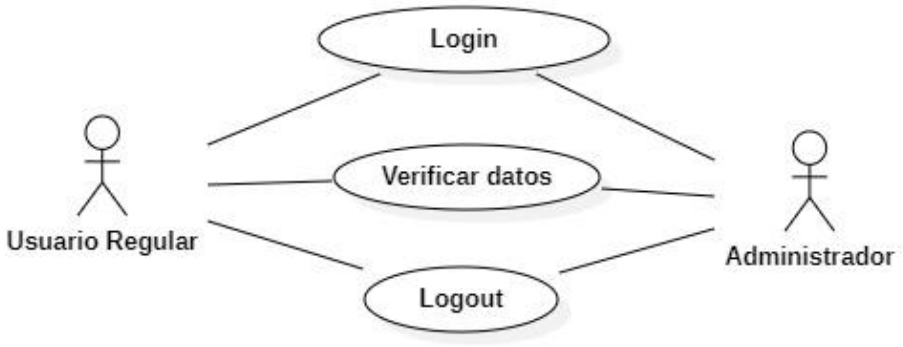


Fig. 14 Caso de Uso de Iniciar Sesión en el Sistema (Fuente: Propia Autoría, 2022)

Tabla. 3 Tabla Caso de Uso Iniciar Sesión en el Sistema (Fuente: Propia Autoría, 2022)

Nombre: Iniciar Sesión en el Sistema	
Caso de uso:	Iniciar Sesión en el Sistema
Actores:	Usuario Regular Administrador
Objetivo:	Iniciar Sesión dentro de la plataforma
Precondiciones:	El usuario debe registrar previamente a el Usuario
Postcondiciones:	Credenciales creadas
Escenario:	Introducir el correo y la contraseña enviada por correo electrónico para el inicio de sesión

3.4.1.2 Gestión de Usuarios para el Administrador (CRUD)

Un Administrador puede gestionar los usuarios registrados dentro del sistema (CRUD).

Tabla. 4 Tabla de Requerimiento Gestión de Usuarios para el Administrador (CRUD) (Fuente: Propia Autoría, 2022)

Nombre: Gestión de Usuarios para el Administrador (CRUD)	
PRIORIDAD: Alta	
NECESIDAD: Esencial	
CLARIDAD: Alta	**VERIFICABILIDAD:** Alta
ESTABILIDAD: Toda la vida del software	
DESCRIPCIÓN: Los Administradores son capaces de registrar, editar, consultar y eliminar los usuarios en el sistema.	

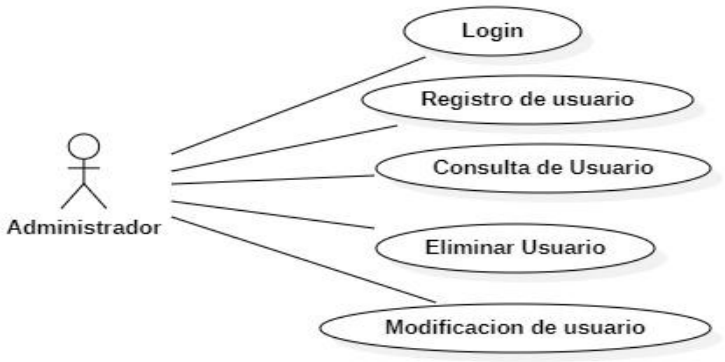


Fig. 15 Caso de Uso de Gestión de Usuarios para el Administrador (CRUD) (Fuente: Propia Autoría, 2022)

Tabla. 5 Tabla Caso de Uso Gestión de Usuarios para el Administrador (CRUD) (Fuente: Propia Autoría, 2022)

Nombre: Gestión de Usuarios para el Administrador (CRUD)	
Caso de uso:	Gestión de Usuarios para el Administrador (CRUD)
Actores:	Administrador
Objetivo:	Registrar, Editar, eliminar y consultar los usuarios dentro del sistema
Precondiciones:	El usuario debe ser un Administrador y haber iniciado sesión
Postcondiciones:	La gestión de usuarios
Escenario:	Para Registrar un usuario, en la página de usuarios, se debe llenar el formulario con los datos solicitados (Nombre, Apellido Paterno, Apellido Materno, correo Electrónico, matricula y perfil del usuario). Si los datos son correctos se le enviara al usuario un correo electrónico con su contraseña para poder iniciar Sesión. El Administrador podrá visualizar una tabla con la lista de usuarios registrados dentro del sistema con sus datos generales. Al hacer presionar en editar el usuario el formulario utilizado para el registro se llena con los datos del usuario para preparar la edición. E muestran las opciones de Editar y cancelar. Al presionar en eliminar un usuario se abre una ventana de confirmación para eliminar el usuario.

3.4.1.3 Gestión de Tarjetas para el Administrador (CRUD)

Un Administrador puede gestionar las tarjetas registradas dentro del sistema (CRUD)

Tabla. 6 Tabla de Requerimiento Gestión de Usuarios para el Administrador (CRUD) (Fuente: Propia Autoría, 2022)

Nombre: Gestión de Tarjetas para el Administrador (CRUD)	
PRIORIDAD: Alta	
NECESIDAD: Esencial	
CLARIDAD: Alta	**VERIFICABILIDAD:** Alta
ESTABILIDAD: Toda la vida del software	
DESCRIPCIÓN: Los Administradores son capaces de registrar, consultar y eliminar las tarjetas de los usuarios en el sistema.	

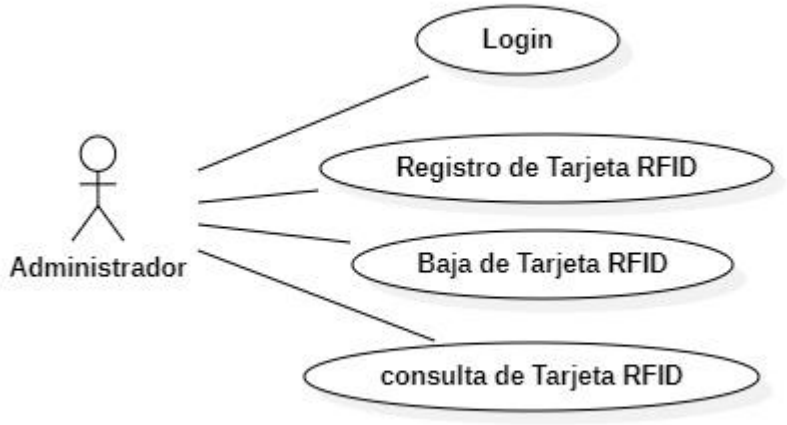


Fig. 16 Caso de Uso de Iniciar Sesión en el Sistema (Fuente: Propia Autoría, 2022)

Tabla. 7 Caso de Uso Gestión de Tarjetas para el Administrador (CRUD) (Fuente: Propia Autoría, 2022)

Nombre: Gestión de Tarjetas para el Administrador (CRUD)	
Caso de uso:	Gestión de Tarjetas para el Administrador (CRUD)
Actores:	Administrador
Objetivo:	Registrar, eliminar y consultar las tarjetas de los usuarios dentro del sistema
Precondiciones:	El usuario debe ser un Administrador y haber iniciado sesión
Postcondiciones:	La gestión de Tarjetas
Escenario:	Para acceder a las tarjetas del usuario, en la página de usuarios, el administrador tiene que presionar en editar al usuario y se abre la opción de tarjetas, al presionar la opción de abre un panel que contiene la lista de tarjetas con las que cuenta el usuario junto con un campo para introducir el numero para registrar una nueva tarjeta. Al presionar en eliminar una tarjeta se abre una ventana de confirmación para eliminar la tarjeta del usuario.

3.4.1.4Consulta y búsqueda de Movimientos para el administrador

Un administrador puede consultar los movimientos realizados en el sistema en base a un filtro de búsqueda.

Tabla. 8 Tabla de Requerimiento Consulta y búsqueda de Movimientos para el administrador (Fuente: Propia Autoría, 2022)

Nombre: Consulta y búsqueda de Movimientos para el administrador	
PRIORIDAD: Alta	
NECESIDAD: Esencial	
CLARIDAD: Alta	**VERIFICABILIDAD:** Alta
ESTABILIDAD: Toda la vida del software	
DESCRIPCIÓN: Los Administradores son capaces de consultar y buscar por medio de un filtro de búsqueda todos los movimientos realizados dentro del sistema	

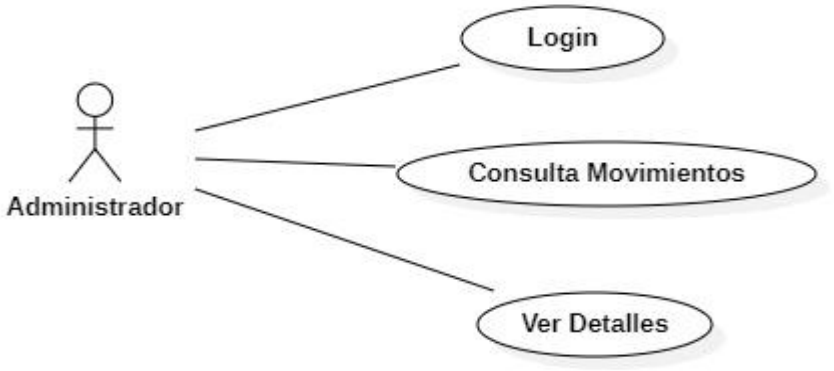


Fig. 17 Caso de Uso de Consulta y búsqueda de Movimientos para el administrador (Fuente: Propia Autoría, 2022)

Tabla. 9 Tabla Caso de Uso Consulta y búsqueda de Movimientos para el administrador (Fuente: Propia Autoría, 2022)

Nombre: Consulta y búsqueda de Movimientos para el administrador	
Caso de uso:	Consulta y búsqueda de Movimientos para el administrador
Actores:	Administrador
Objetivo:	Consultar y buscar todos los movimientos realizados dentro del sistema
Precondiciones:	El usuario debe ser un Administrador y haber iniciado sesión
Postcondiciones:	La consulta y búsqueda de movimientos
Escenario:	El Administrador, en la página de movimientos, puede encontrar un filtro de búsqueda en base a la fecha y a el usuario que realizo el movimiento para consultar todos los movimientos realizados dentro del sistema. Al consultar los

	movimientos tendrán la opción de ver los detalles más específicos del movimiento dichos detalles son el material ingresado y el total de puntos obtenidos.

3.4.1.5 Consulta y búsqueda de errores para el Administrador

Un administrador puede revisar los errores surgidos en el sistema en base a un filtro de búsqueda

Tabla. 10 Tabla de Requerimiento Consulta y búsqueda de errores para el administrador (Fuente: Propia Autoría, 2022)

Nombre: Consulta y búsqueda de errores para el administrador	
PRIORIDAD: Alta	
NECESIDAD: Esencial	
CLARIDAD: Alta	**VERIFICABILIDAD:** Alta
ESTABILIDAD: Toda la vida del software	
DESCRIPCIÓN: Los Administradores son capaces de consultar y buscar por medio de un filtro de búsqueda todos los movimientos realizados dentro del sistema	

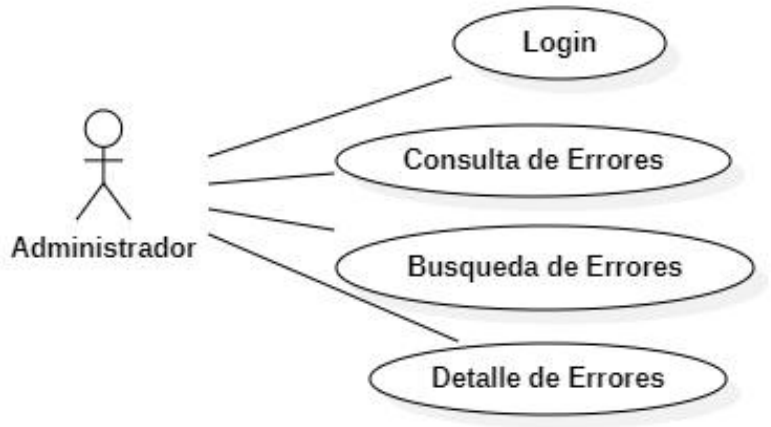


Fig. 18 Caso de Uso de Consulta y búsqueda de errores para el administrador (Fuente: Propia Autoría, 2022)

Tabla. 11 Tabla Caso de Uso Consulta y búsqueda de Errores para el administrador (Fuente:Propia Autoría, 2022)

Nombre: Consulta y búsqueda de Errores para el administrador	
Caso de uso:	Consulta y búsqueda de Errores para el administrador
Actores:	Administrador
Objetivo:	Consultar y buscar todos los errores realizados dentro del sistema dentro del sistema
Precondiciones:	El usuario debe ser un Administrador y haber iniciado sesión
Postcondiciones:	La consulta y búsqueda de errores
Escenario:	El Administrador, en la página de errores, puede encontrar una filtro de búsqueda en base a la fecha, el estatus, y la maquina en la que surgió el error para consultar todos los errores surgidos dentro del sistema. Al consultar los errores tendrán la opción de ver los detalles más específicos del error dichos detalles so el estatus el tipo de error y la fecha en la que se solucionó.

3.4.1.5 Cambio de estatus para los errores surgidos para el Administrador

Un Administrador puede indicar que un error ya fue solucionado para así cambiar el estatus de este.

Tabla. 12 Tabla de Requerimiento Cambio de estatus para los errores surgidos para el Administrador (Fuente: Propia Autoría, 2022)

Nombre: Cambio de estatus para los errores surgidos para el Administrador	
PRIORIDAD: Alta	
NECESIDAD: Esencial	
CLARIDAD: Alta	**VERIFICABILIDAD:** Alta
ESTABILIDAD: Toda la vida del software	
DESCRIPCIÓN: Los Administradores son capaces de consultar y buscar por medio de un filtro de búsqueda todos los movimientos realizados dentro del sistema	

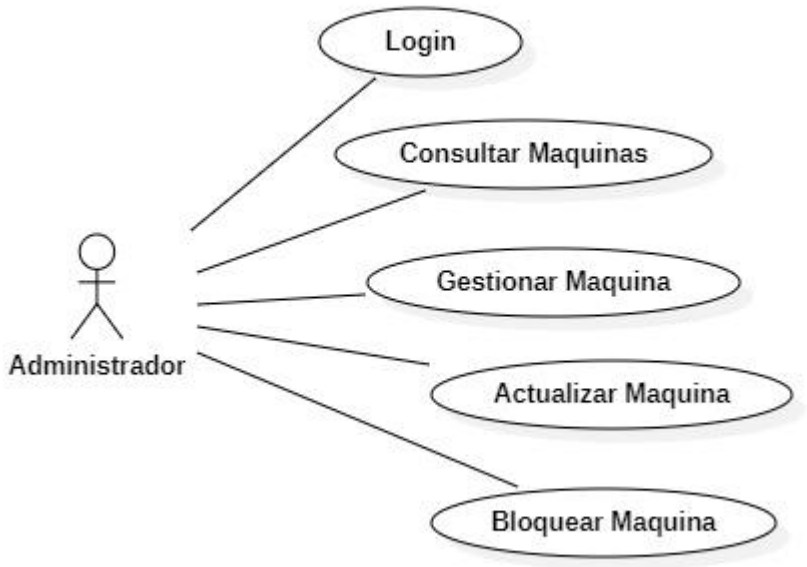


Fig. 19 Caso de Uso de Cambio de estatus para los errores surgidos para el Administrador (Fuente: Propia Autoría, 2022)

Tabla. 13. Tabla Caso de Uso Cambio de estatus para los errores surgidos para el Administrador (Fuente: Propia Autoría, 2022)

Nombre: Cambio de estatus para los errores surgidos para el Administrador	
Caso de uso:	Cambio de estatus para los errores surgidos para el Administrador
Actores:	Administrador
Objetivo:	Cambiar el estatus del error surgido dentro del dispositivo de reciclaje
Precondiciones:	El usuario debe ser un Administrador y haber iniciado sesión
Postcondiciones:	La consulta y búsqueda de errores
Escenario:	El Administrador, en la página de errores, al realizar una búsqueda, los errores tendrán la opción de cambiar el estatus de los errores activos a solucionado en dicho caso de que el error haya sido resuelto satisfactoriamente

3.4.1.6 Gestión de máquinas para los Administradores

Un administrador puede visualizar, editar y bloquear las máquinas registradas dentro del sistema.

Tabla. 14 Tabla de Requerimiento Gestión de máquinas para los Administradores (Fuente: Propia Autoría, 2022)

Nombre: Gestión de máquinas para los Administradores	
PRIORIDAD: Alta	
NECESIDAD: Esencial	
CLARIDAD: Alta	**VERIFICABILIDAD:** Alta
ESTABILIDAD: Toda la vida del software	
DESCRIPCIÓN: Los Administradores son capaces de registrar, consultar y eliminar las tarjetas de los usuarios en el sistema.	

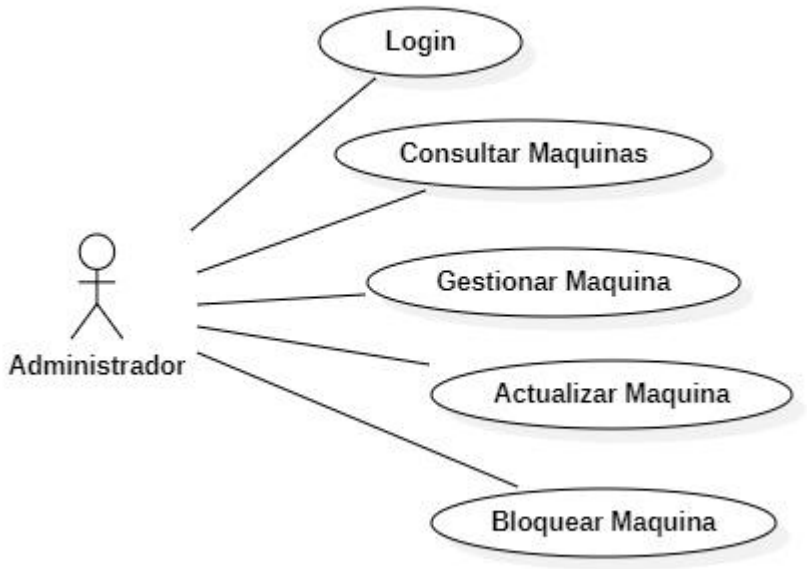


Fig. 20 Caso de Uso de Gestión de máquinas para los Administradores (Fuente: Propia Autoría, 2022)

Tabla. 15. Tabla Caso de Uso Gestión de máquinas para los Administradores (Fuente: Propia Autoría, 2022)

Nombre: Gestión de Tarjetas para el Administrador (CRUD)	
Caso de uso:	Gestión de máquinas para los Administradores
Actores:	Administrador
Objetivo:	Consultar y Editar las maquinas registradas dentro del sistema
Precondiciones:	El usuario debe ser un Administrador y haber iniciado sesión
Postcondiciones:	La gestión de Maquinas
Escenario:	El administrador, en la página de máquinas, podrá visualizar cada una de las maquinas registradas den el sistema junto con sus datos en específicos como lo es: estatus, fecha y ubicación. Cada maquina contiene dos opciones las cuales son editar y bloquear máquina, al presionar editar maquina se abrirá una ventana emergente en la cual vendrá un formulario con los datos cargados de la máquina para su edición. Si el administrador desea bloquear la maquina deberá presionar el botón de bloquear maquina y está abrirá una ventana emergente de confirmación.

3.4.1.6 Cambio de Contraseña para todos los usuarios

La plataforma va a contener un apartado donde los usuarios comunes y los administradores visualicen todos los datos de su perfil y estos mismos podrán realizar el cambio de su contraseña manualmente.

Tabla. 16 Tabla de Requerimiento Cambio de Contraseña para todos los usuarios (Fuente: Propia Autoría, 2022)

Nombre: Cambio de Contraseña para todos los usuarios	
PRIORIDAD: Alta	
NECESIDAD: Deseable	
CLARIDAD: Alta	**VERIFICABILIDAD:** Alta
ESTABILIDAD: Toda la vida del software	
DESCRIPCIÓN: Los Administradores y Usuarios Regulares son capaces de cambiar manualmente su contraseña.	

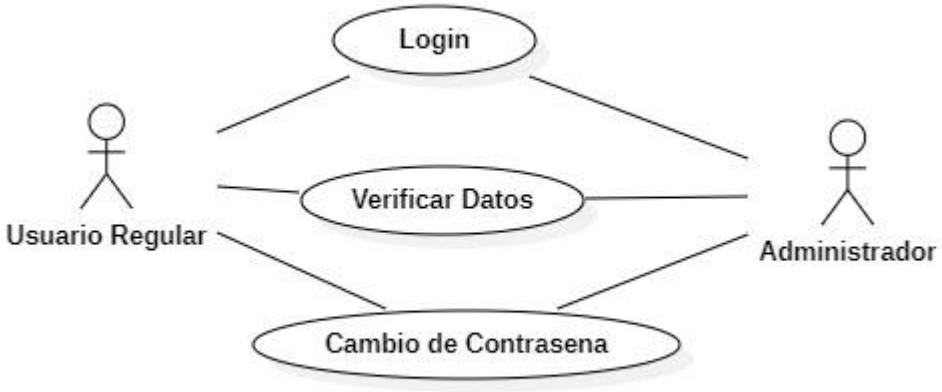


Fig. 21 Caso de Uso de Cambio de Contraseña para todos los usuarios (Fuente: Propia Autoría, 2022)

Tabla. 17 Tabla Caso de Uso Cambio de Contraseña para todos los usuarios (Fuente: Propia Autoría, 2022)

Nombre: Cambio de Contraseña para todos los usuarios	
Caso de uso:	Cambio de Contraseña para todos los usuarios
Actores:	Administrador y Usuario Regular
Objetivo:	El usuario regular y el administrador podrán modificar la contraseña manualmente.
Precondiciones:	El usuario debe ser un usuario registrado dentro del sistema y haber iniciado sesión
Postcondiciones:	Cambio de contraseña del usuario manualmente
Escenario:	El usuario, en la página de perfil, podrá visualizar los datos generales de cuenta y en la siguiente sección vendrá la opción de cambiar la contraseña. Para cambiar la contraseña el usuario deberá introducir su contraseña actual más la nueva contraseña y confirmación de la nueva contraseña, al llenar los datos correctamente de abrirá una ventana emergente de confirmación de cambio de contraseña

3.4.1.6 Consulta y búsqueda de los movimientos de cuenta propia para un usuario Regular

Un Usuario Regular puede consultar los movimientos que ha realizado en el sistema en base a un filtro de búsqueda.

Tabla. 18 Tabla de Requerimiento Consulta y búsqueda de Movimientos para el Usuario Regular (Fuente: Propia Autoría, 2022)

Nombre: Consulta y búsqueda de Movimientos para el Usuario Regular	
PRIORIDAD: Alta	
NECESIDAD: Esencial	
CLARIDAD: Alta	**VERIFICABILIDAD:** Alta
ESTABILIDAD: Toda la vida del software	
DESCRIPCIÓN: Los Administradores son capaces de consultar y buscar por medio de un filtro de búsqueda todos los movimientos realizados dentro del sistema	

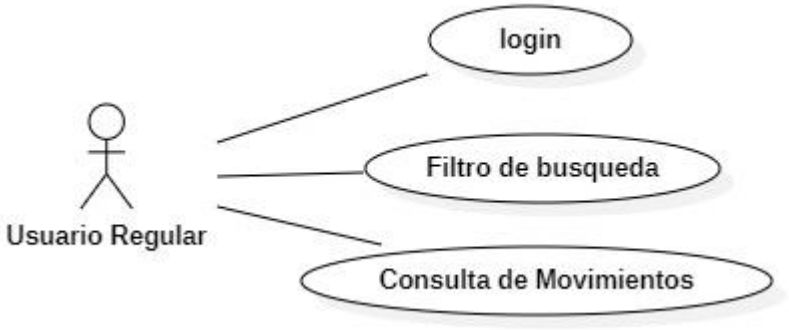


Fig. 22 Caso de Uso de Consulta y búsqueda de Movimientos para el Usuario Regular (Fuente: Propia Autoría, 2022)

Tabla. 19. Tabla Caso de Uso Consulta y búsqueda de Movimientos para el Usuario Regular (Fuente: Propia Autoría, 2022)

Nombre: Consulta y búsqueda de Movimientos para el Usuario Regular	
Caso de uso:	Consulta y búsqueda de Movimientos para el Usuario Regular
Actores:	Usuario Regular
Objetivo:	Consultar y buscar los movimientos realizados por el mismo usuario dentro del sistema
Precondiciones:	El usuario debe ser un Usuario Regular y haber iniciado sesión
Postcondiciones:	La consulta y búsqueda de movimientos
Escenario:	El Usuario Regular, en la página de movimientos, puede encontrar una filtro de búsqueda en base a la fecha en la que realizo el movimiento para consultar todos los movimientos que el mismo ha realizado dentro del sistema. Al consultar los movimientos tendrán la opción de ver los detalles más específicos del movimiento dichos detalles son el material ingresado y el total de puntos obtenidos.

CAPÍTULO IV

RESULTADOS

4.1 INTRODUCCIÓN

En este último capítulo se presentarán los resultados obtenidos con el desarrollo del capítulo pasado, la integración de todas las funcionalidades que se ocupó para llegar a este punto.

Los resultados presentados ya han sido validados y corregidos sea el caso de cada uno de sus problemas. También los resultados son presentados en un entorno de producción y ya no de desarrollo como se ha manejado en el capítulo pasado, de la misma forma ha sido probado en este entorno.

Los resultados que se presentan en esta sección muestran información y esta no se debe ser comprendida como verídica, puesto que son datos sin ningún sentido para proteger información real utilizada en el sistema. La plataforma ya ha sido probada con información sensible en la parte de pruebas, pero como se ha referido en todo el documento es información sensible y no puede ser mostrada.

4.2 Resultado de Interfaces

Las siguientes descripciones se describe las páginas que se van a ejecutar y que fueron desarrolladas por los requerimientos propuestos, como se ha mencionado estos requerimientos varían dependiendo a el tipo de usuario registrado del sistema por lo cual se mostrarán los resultados dependiendo a el tipo de usuario.

4.2.1 Inicio de Sesión en la Plataforma

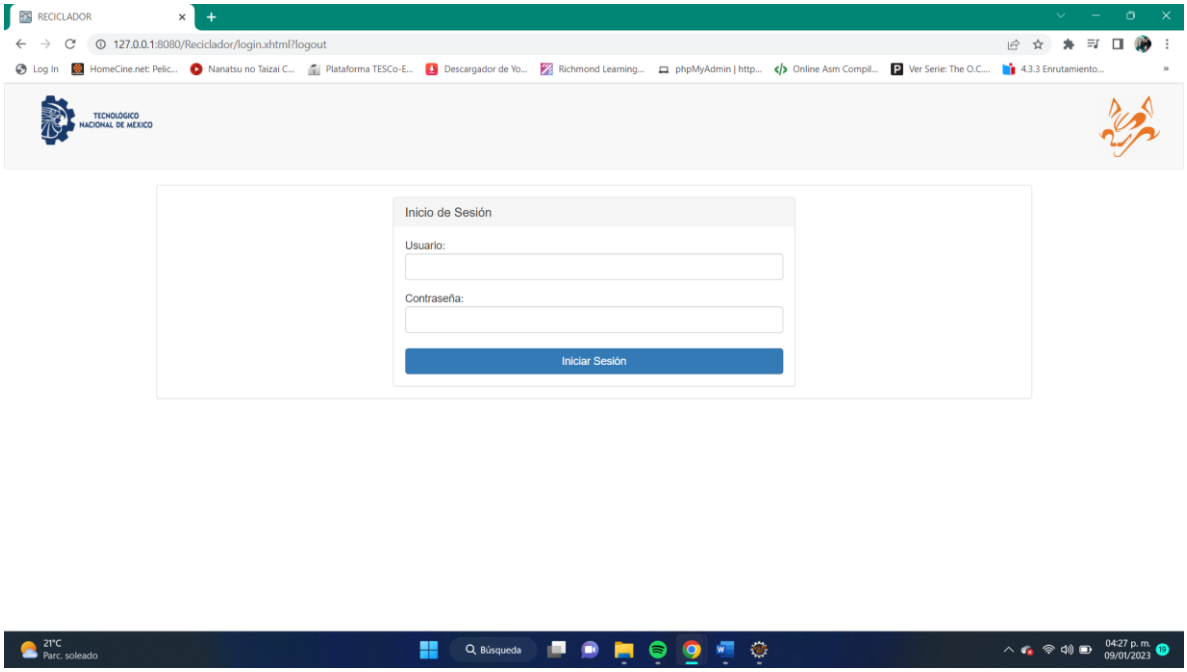


Fig. 23 Pagina de Iniciar Sesión (Fuente: Propia Autoría, 2022)

En la página principal se muestra de manera concreta en la Fig. 19, el uso de la plataforma mostrando el flujo de Iniciar sesión e iniciar en el cual se muestra un formulario donde el usuario debe introducir su correo de usuario y contraseña para acceder al sistema ahora, al introducir los datos correctos va a dirigir al usuario a la página de inicio del sistema como se muestra en la Fig. 21.

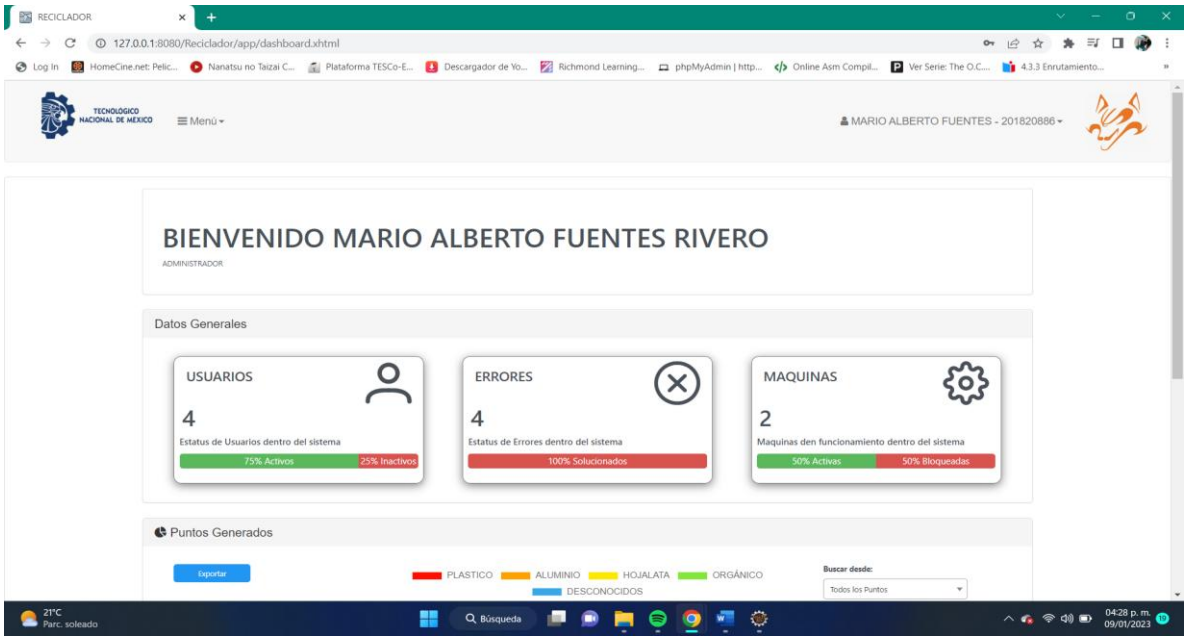


Fig. 24 Página de Inicio Administrador (Fuente: Propia Autoría, 2022)

4.2.2 Cambio de Contraseña en la página de Perfil

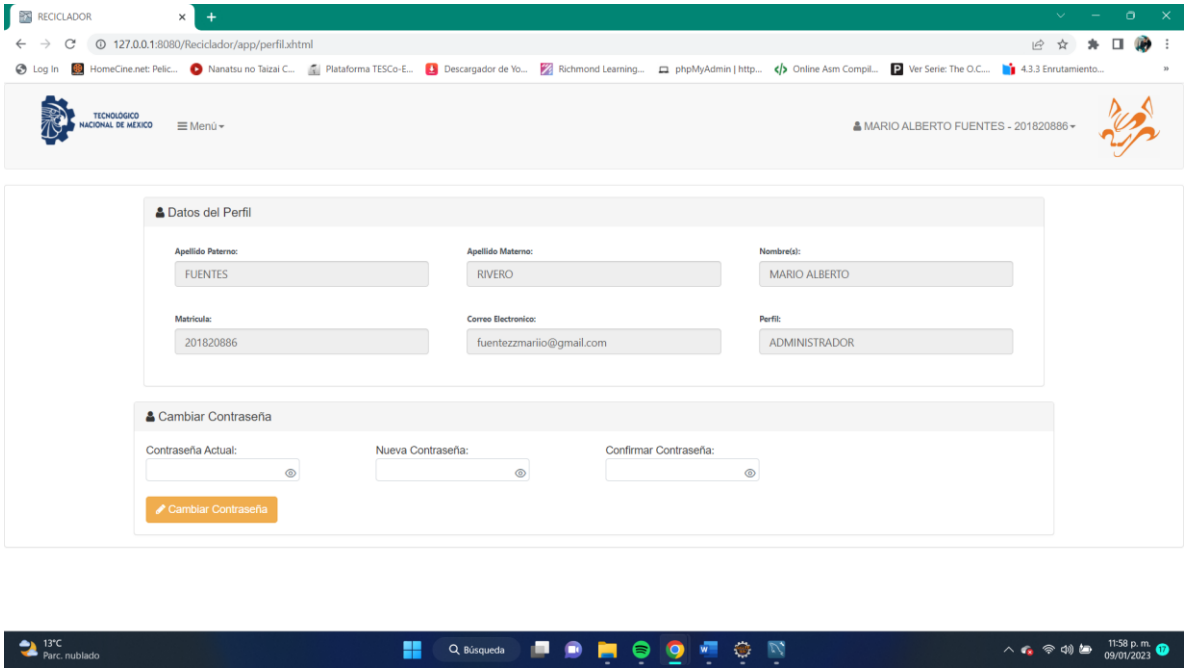


Fig. 25 Pagina de Perfil (Fuente: Propia Autoría, 2022)

La página de Perfil cuenta con una sección donde se encuentran los datos del perfil del usuario, tanto sus datos personales como su tipo de usuario en el sistema. De igual manera contine una sección en la que el usuario tiene como opción cambiar su contraseña ingresando su contraseña actual, la nueva contraseña y la confirmación de la contraseña, al ingresar los datos correctos se abre una ventana emergente de confirmación. Cabe recalcar que en esta página puede acceder tanto los usuarios regulares como los Administradores.

Fig. 26 Ventana emergente de confirmación de cambio de contraseña (Fuente: Propia Autoría, 2022)

4.2.3 Usuario Administrador

4.2.3.1 Página de Inicio del Administrador

Al ingresar, el diseño manejado se basa en un menú desplegable en una barra superior en donde se muestral las diferentes opciones en las cuales el administrador puede acceder

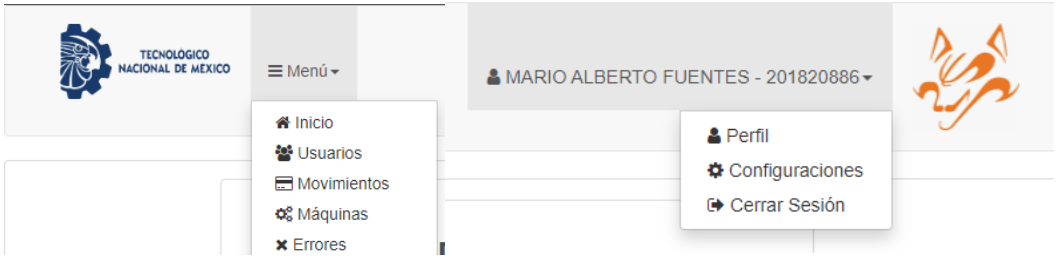


Fig. 27 Menú para el Administrador (Fuente: Propia Autoría, 2022)

La página de inicio contine información general del sistema divida en 3 secciones. La primera sección muestra el nombre completo junto el tipo del perfil del usuario que ingreso a la plataforma.

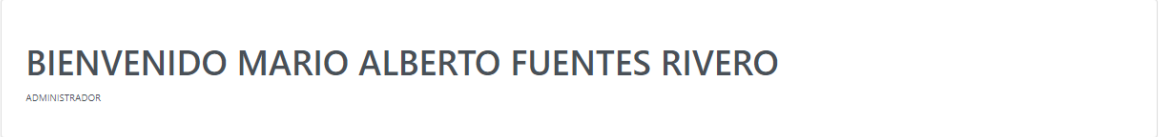


Fig. 28 Sección de Datos del Usuario (Fuente: Propia Autoría, 2022)

La segunda Sección muestra 3 cuadros que muestra la siguiente información:

- El primer cuadro muestra el total de usuarios registrados dentro del sistema con una barra inferior que muestra el porcentaje de usuarios inactivos y activos del total de la suma total.
- El segundo cuadro muestra el total de errores acumulados que ha detectado el sistema de igual manera cuenta con una barra inferior que muestra el porcentaje de errores solucionados y activos de la suma total.
- El tercer cuadro muestra el total de máquinas registradas dentro del sistema, de igual manera cuenta con una barra en la parte inferior que muestra el porcentaje de máquinas activas y bloqueadas de la suma total.

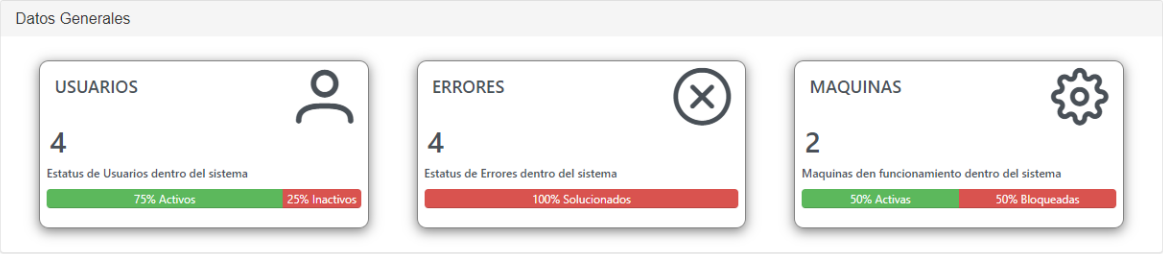


Fig. 29 Sección de datos generales del Sistema (Fuente: Propia Autoría, 2022)

La tercera y última sección contiene un esquema que grafica los puntos totales generados dentro del sistema dividido entre los diferentes tipos materiales que aceptan los dispositivos de reciclaje. Por otra parte, contiene la opción para que el administrador grafique los datos en un tiempo específico.

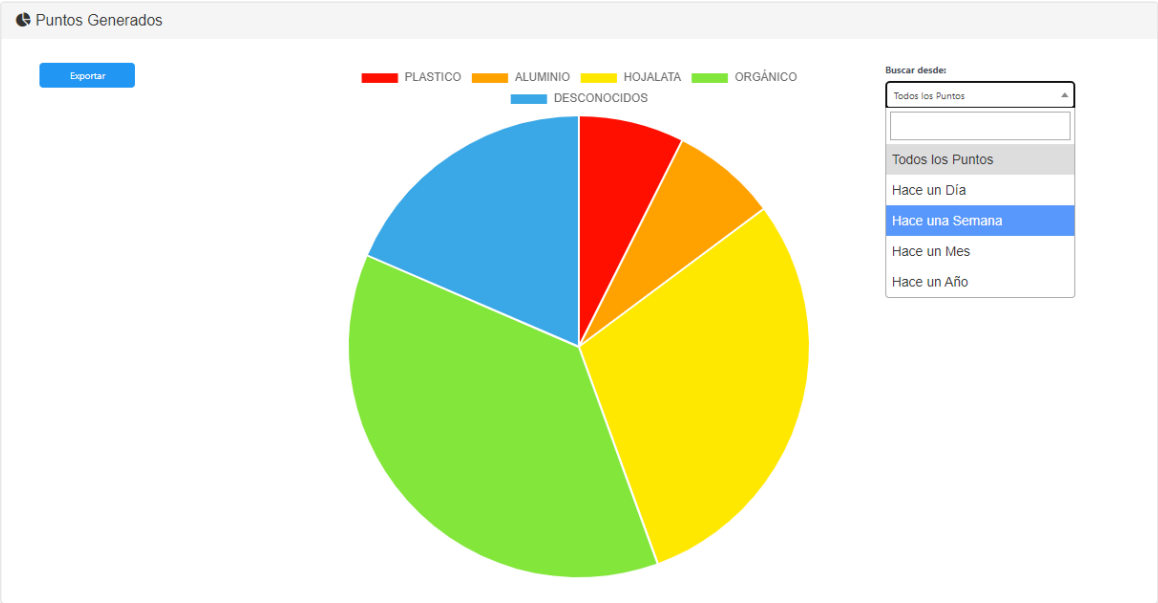


Fig. 30 Sección de Grafica de puntos generados (Fuente: Propia Autoría, 2022)

4.2.3.2 Gestión de usuarios para el Administrador

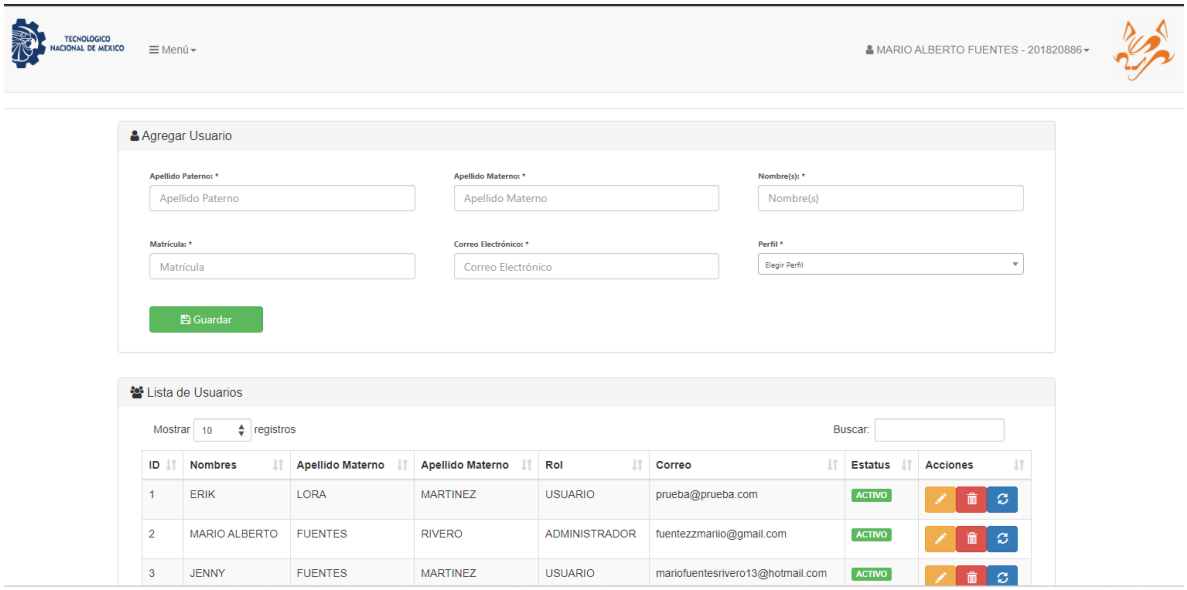


Fig. 31 Pagina de Gestión de Usuarios para el Administrador (Fuente: Propia Autoría, 2022)

En la página de usuarios, el administrador realiza la gestión de usuarios donde se visualiza un formulario con los datos requeridos para que se pueda registrar un usuario dentro del sistema. Cabe recalcar que al registrar un usuario se le va a hacer llegar un correo electrónico a la dirección registrada con una contraseña generada automáticamente para que el usuario pueda iniciar sesión.

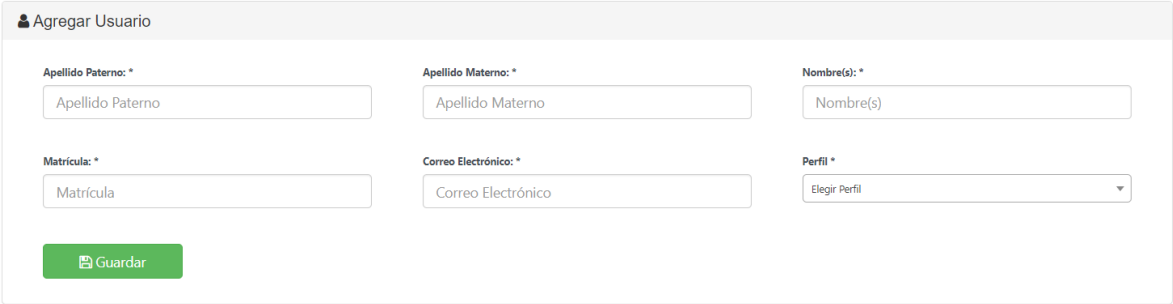


Fig. 32 Formulario para registrar un nuevo Usuario (Fuente: Propia Autoría, 2022)

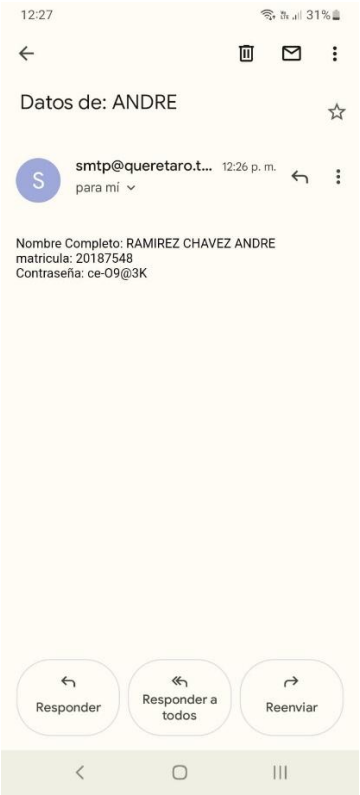
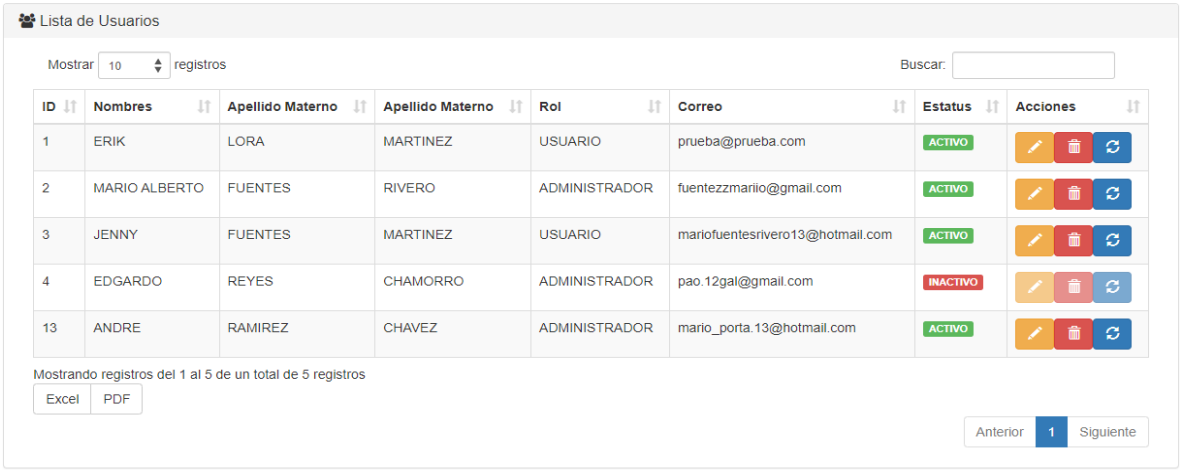


Fig. 33 Ejemplo de Correo con contraseña para el registro de un usuario nuevo (Fuente: Propia Autoría, 2022)

Por otra parte, la página cuenta con una tabla que contiene todos los datos de los usuarios registrados dentro del sistema. Cada usuario contiene las opciones de *Eliminar*, *Editar* y en dicho caso que sea requerido *Restaurar Contraseña* en donde se le va a hacer llegar una nueva contraseña a el usuario por medio de correo electrónico.



ID	Nombres	Apellido Materno	Apellido Materno	Rol	Correo	Estatus	Acciones
1	ERIK	LORA	MARTINEZ	USUARIO	prueba@prueba.com	ACTIVO	
2	MARIO ALBERTO	FUENTES	RIVERO	ADMINISTRADOR	fuentezzmariio@gmail.com	ACTIVO	
3	JENNY	FUENTES	MARTINEZ	USUARIO	mariofuentesrivero13@hotmail.com	ACTIVO	
4	EDGARDO	REYES	CHAMORRO	ADMINISTRADOR	pao.12gal@gmail.com	INACTIVO	
13	ANDRE	RAMIREZ	CHAVEZ	ADMINISTRADOR	mario_porta.13@hotmail.com	ACTIVO	

Fig. 34 Tabla de lista de Usuario registrados (Fuente: Propia Autoría, 2022)

Cabe recalcar que antes de realizar cualquier acción se va a abrir una ventana emergente de confirmación para realizar dicha acción como se muestra en la fig.

Fig. 35 Ventana de Confirmación para Eliminar un Usuario (Fuente: Propia Autoría, 2022)

Fig. 36 Ventana de Restaurar la contraseña de un Usuario (Fuente: Propia Autoría, 2022)

4.2.3.3 Gestión de tarjetas de los usuarios para el Administrador

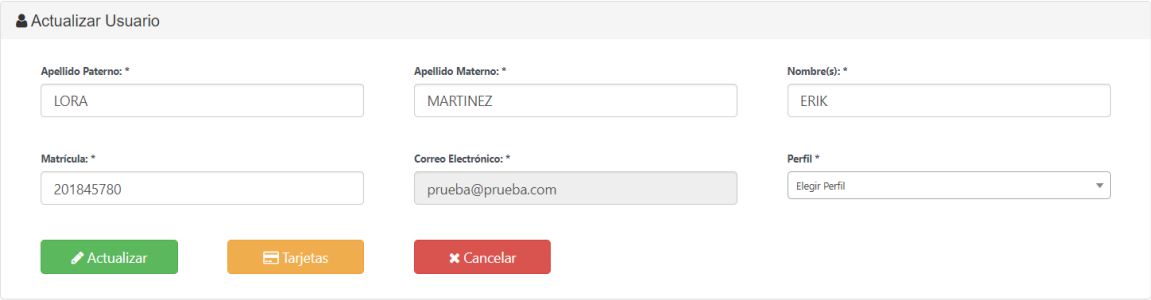


Fig. 37 Pagina de Edición de los Usuarios para el Administrador (Fuente: Propia Autoría, 2022)

Cuando el administrador trate de editar un usuario se mostrará la opción de Tarjetas la cual abre un apartado con un listado de las tarjetas con las que cuenta el usuario. De igual manera contiene la opción de agregar una nueva tarjeta, cabe recalcar que si el usuario cuenta con una tarjeta activa no se va a poder agregar una nueva tarjeta

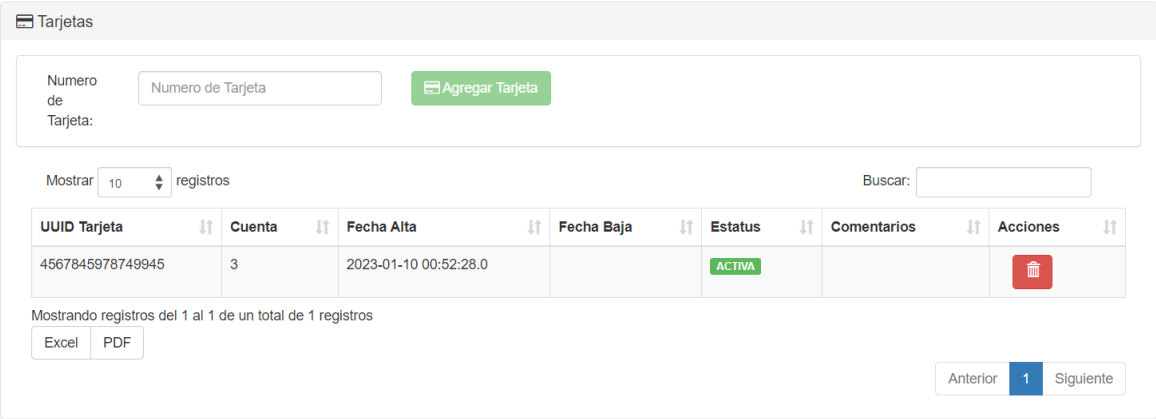


Fig. 38 Tabla de Tarjetas de los Usuarios (Fuente: Propia Autoría, 2022)

4.2.3.4 Consulta y filtro de búsqueda de los movimientos para el Administrador

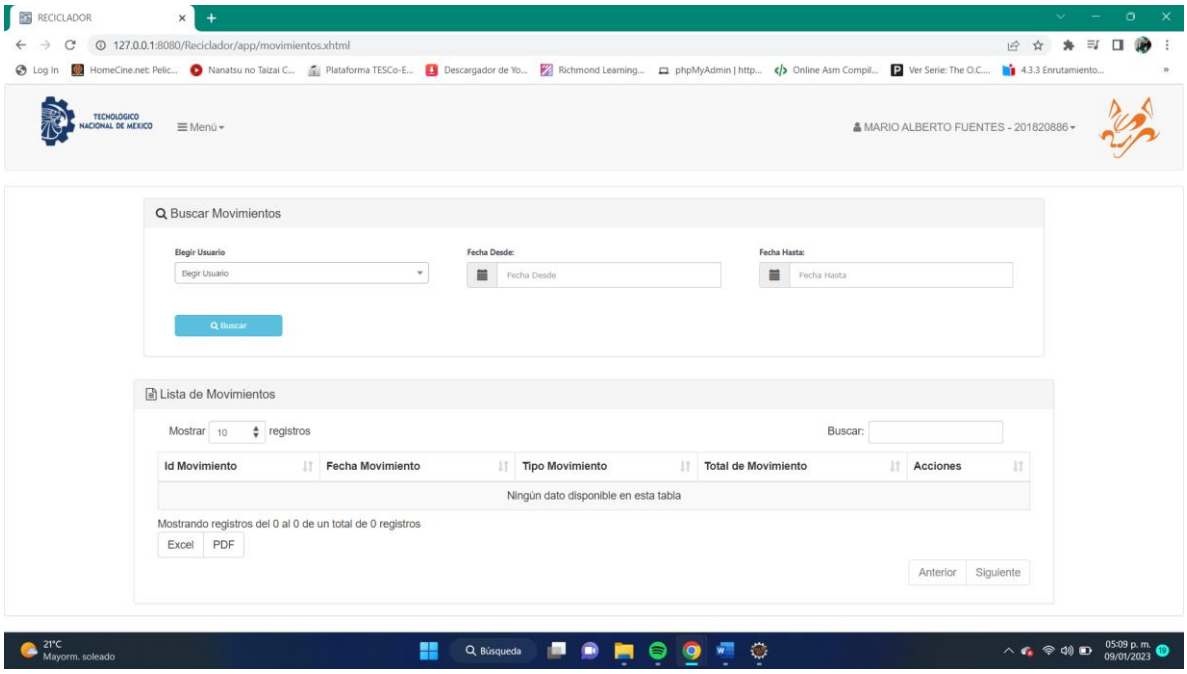


Fig. 39 Pagina de Consulta y filtro de búsqueda de movimientos para el Administrador (Fuente: Propia Autoría, 2022)

La página de movimientos cuenta con un filtro de búsqueda para la consulta de todos los movimientos en base a la fecha del movimiento y al usuario que realizo dicho movimiento.

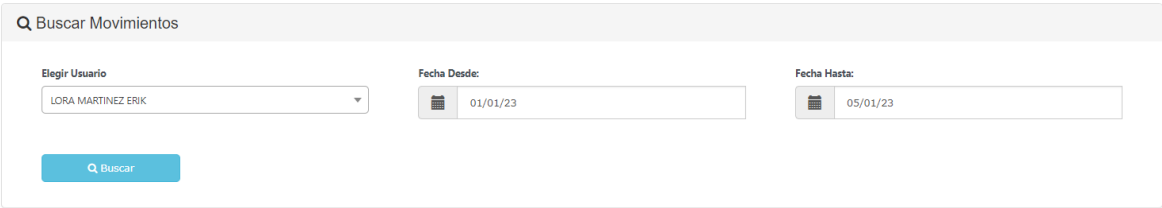


Fig. 40 Filtro de búsqueda para la consulta de movimientos para el Administrador (Fuente: Propia Autoría, 2022)

La siguiente sección cuenta con una tabla de la lista de movimientos con los datos generales de cada uno de los movimientos junto con un botón que abre una ventana que especifica los detalles de cada uno de los movimientos.

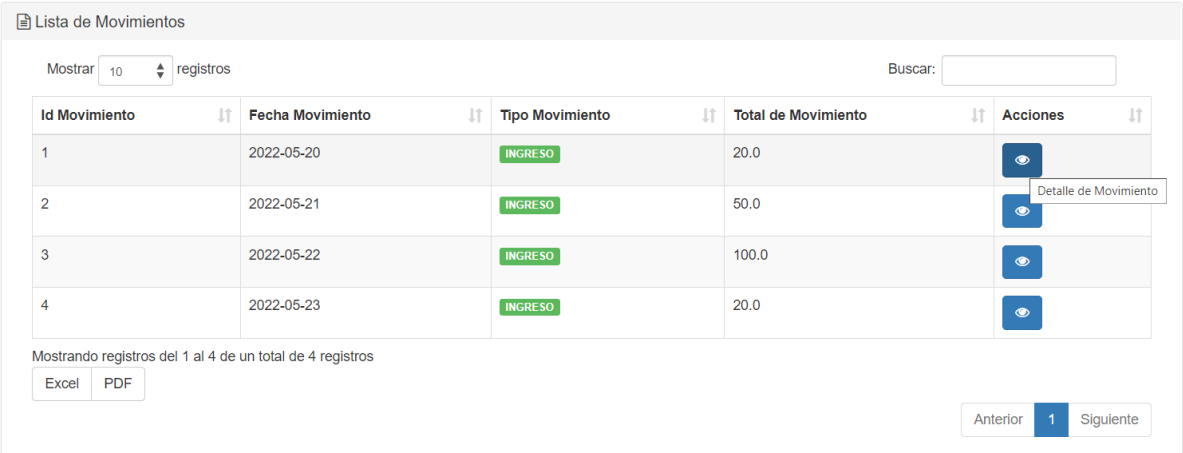


Fig. 41 Tabla de lista de Movimientos (Fuente: Propia Autoría, 2022)

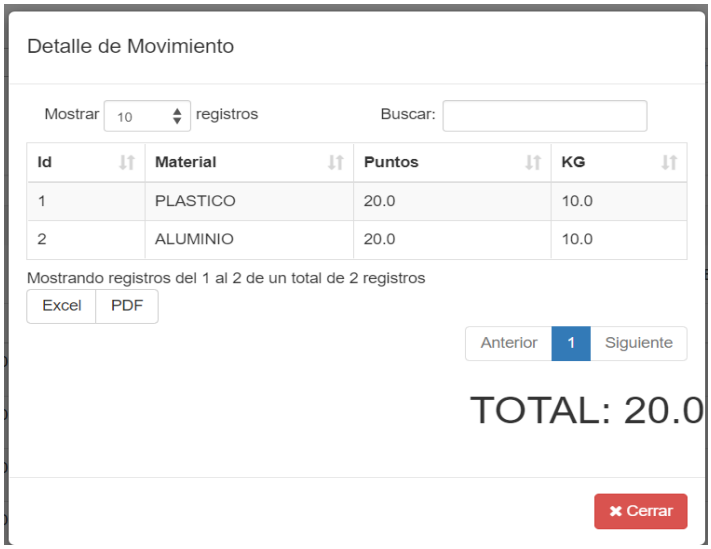


Fig. 42 Ventana Emergente con el detalle del movimiento (Fuente: Propia Autoría, 2022)

4.2.3.5 Consulta y filtro de búsqueda de los errores para el Administrador

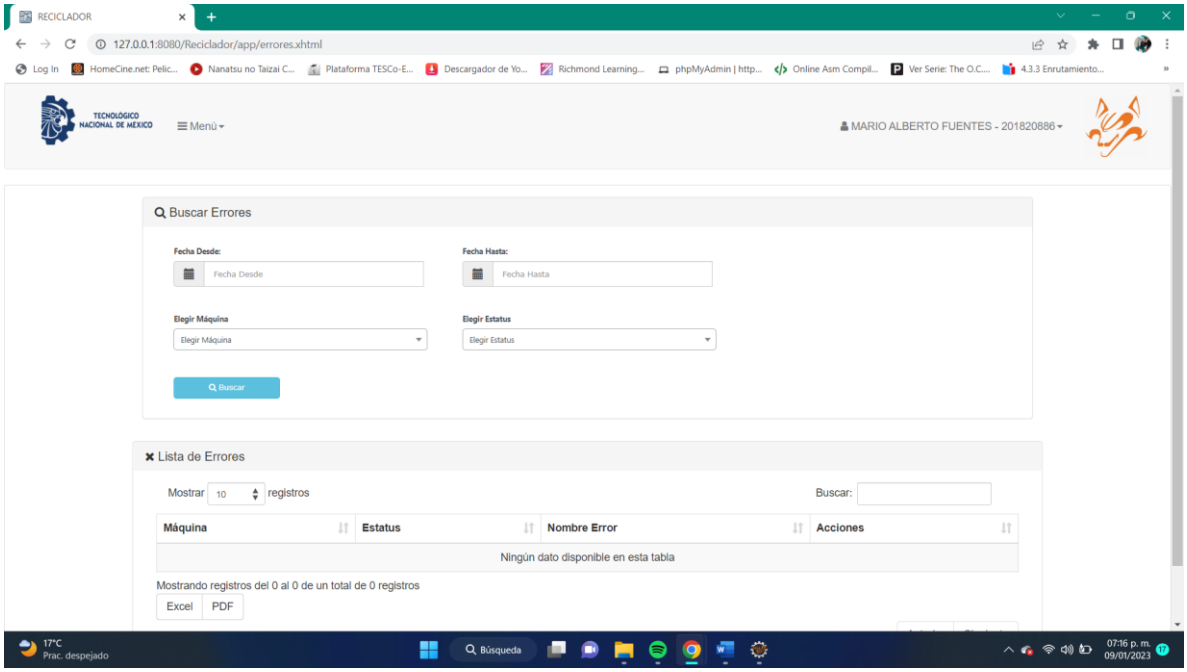


Fig. 43 Pagina de Consulta y filtro de búsqueda de errores para el Administrador (Fuente: Propia Autoría, 2022)

La página de movimientos cuenta con un filtro de búsqueda para la consulta de todos los errores surgidos en el sistema en base a la fecha del error, a la maquina en la que surgió y el estatus del error.

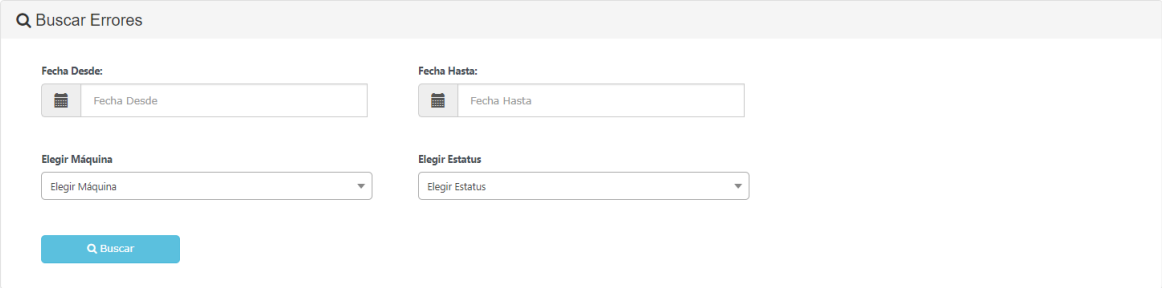


Fig. 44 Filtro de búsqueda para la consulta de errores para el Administrador (Fuente: Propia Autoría, 2022)

La siguiente sección cuenta con una tabla de la lista de errores con los datos generales de cada uno de los errores. Cada error cuenta con las opciones de ver detalle del error, la cual abre una ventana con los detalles del error y la opción de cambiar el estatus del error a solucionado.

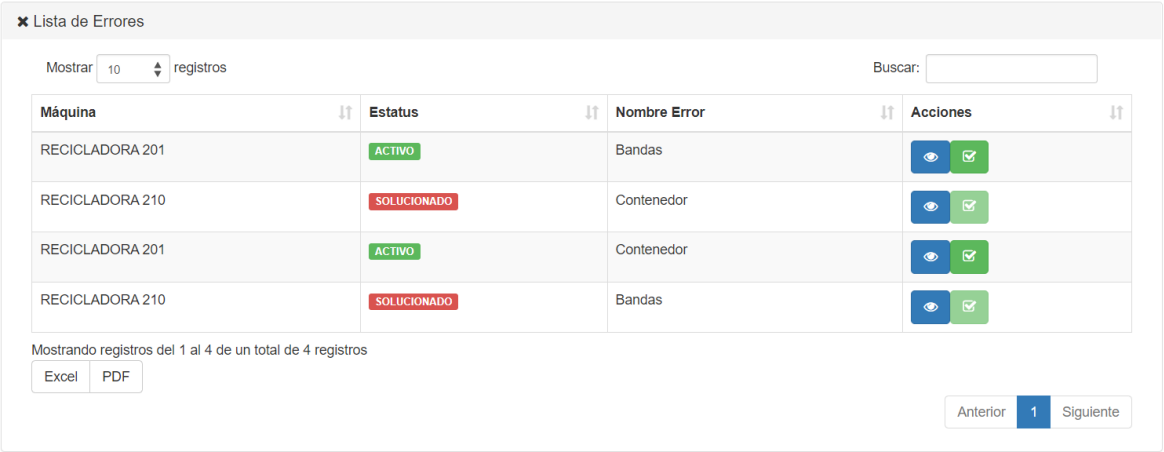


Fig. 45 Tabla de lista de Errores para el Administrador (Fuente: Propia Autoría, 2022)

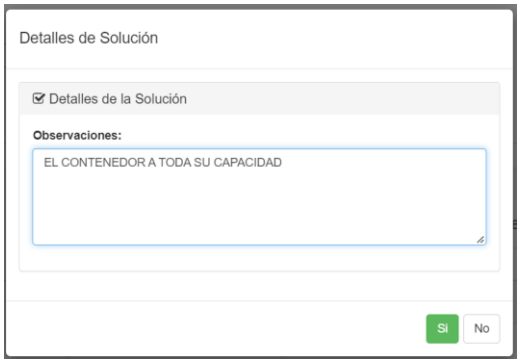


Fig. 46 Ventana Emergente para cambiar el estatus del error (Fuente: Propia Autoría, 2022)

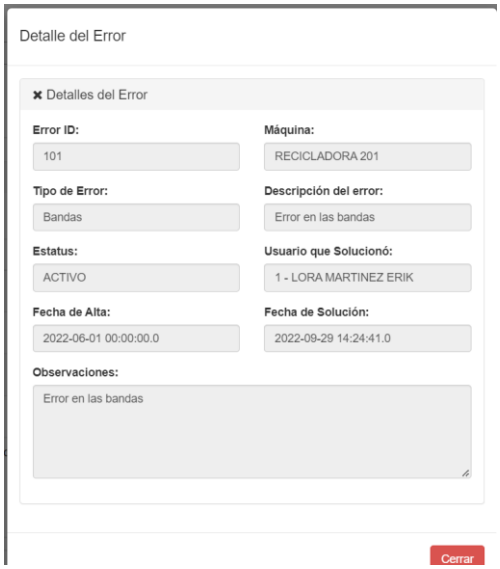


Fig. 47 Ventana Emergente con el detalle del error (Fuente: Propia Autoría, 2022)

4.2.4 Usuario Regular

4.2.4.1 Página de Inicio de un Usuario Regular

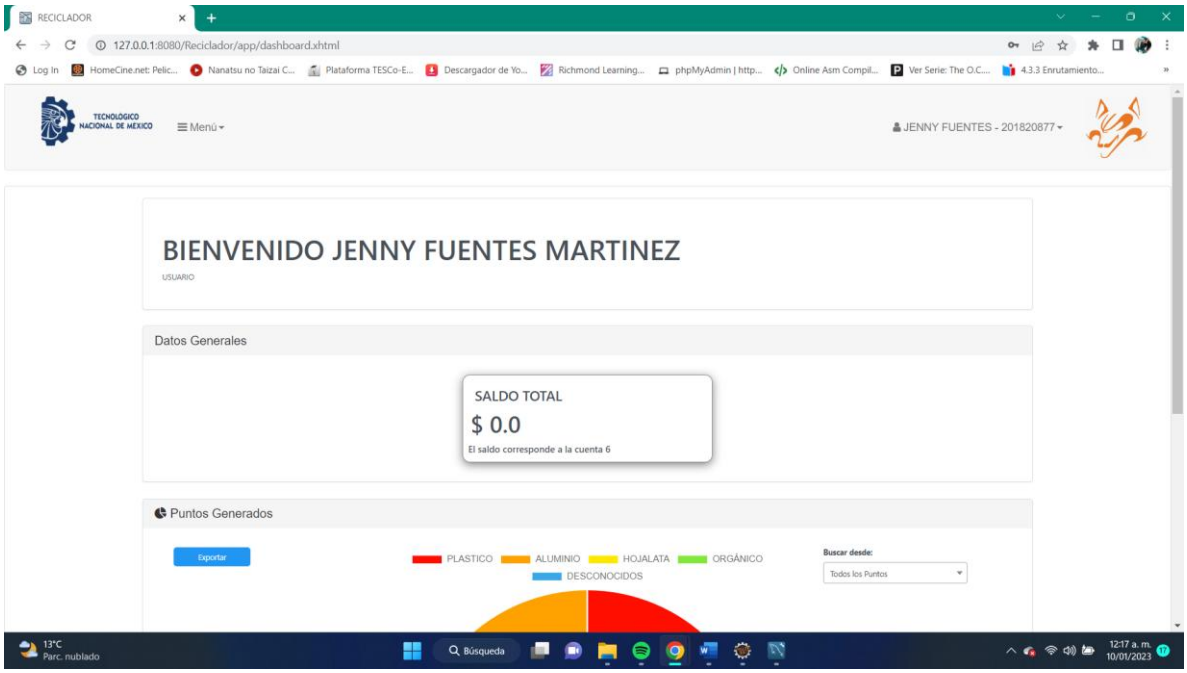


Fig. 48 Página de inicio de un usuario regular (Fuente: Propia Autoría, 2022)

Al ingresar, el diseño manejado se basa en un menú desplegable en una barra superior en donde se muestral las diferentes opciones en las cuales un usuario regular puede acceder

Fig. 49 Menú para un Usuario Regular (Fuente: Propia Autoría, 2022)

La página de inicio contine información general del usuario divida en 3 secciones. La primera sección muestra el nombre completo junto el tipo del perfil del usuario que ingreso a la plataforma.

Fig. 50 Sección de Datos del Usuario (Fuente: Propia Autoría, 2022)

La segunda Sección muestra un cuadro con el saldo total con el que cuenta el usuario en base a los puntos que ha generado dentro del sistema.

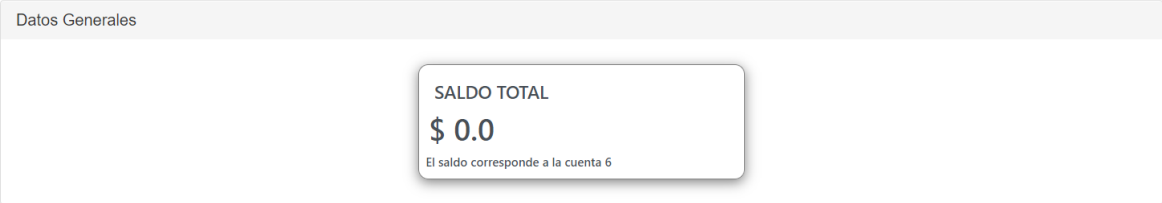


Fig. 51 Sección de datos de la cuenta del usuario (Fuente: Propia Autoría, 2022)

La tercera y última sección contiene un esquema que grafica los puntos totales generados por el usuario dentro del sistema dividido entre los diferentes tipos materiales que aceptan los dispositivos de reciclaje. Por otra parte, contiene la opción para que el Usuario regular grafique los datos en un tiempo específico.

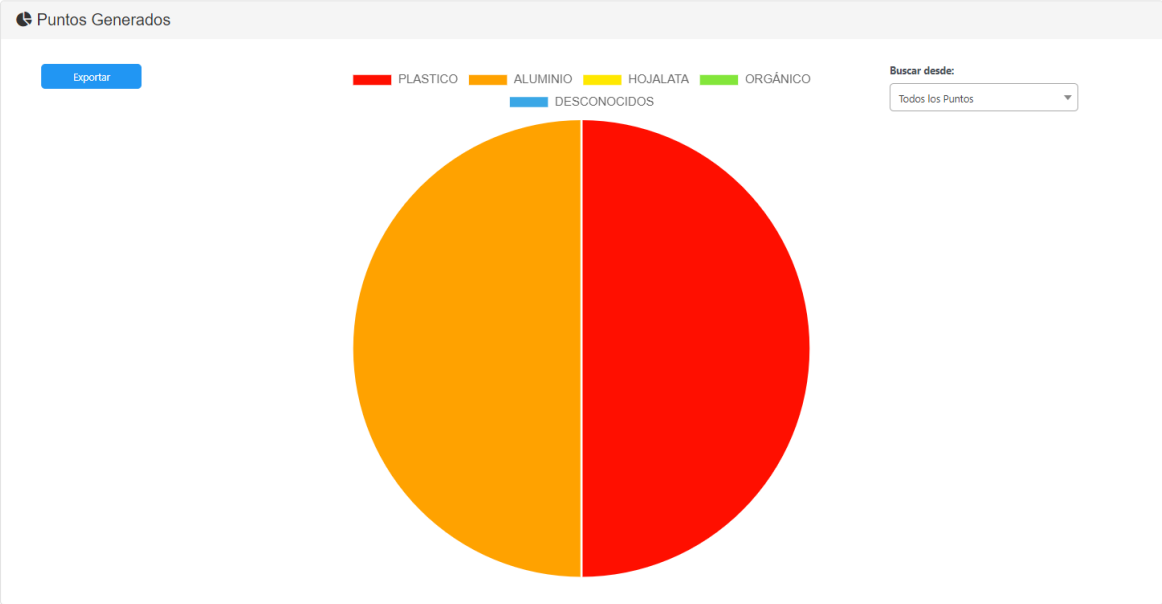


Fig. 52 Sección de Grafica de puntos generados por el usuario (Fuente: Propia Autoría, 2022)

4.2.3.3 Consulta y filtro de búsqueda de los movimientos del usuario Regular

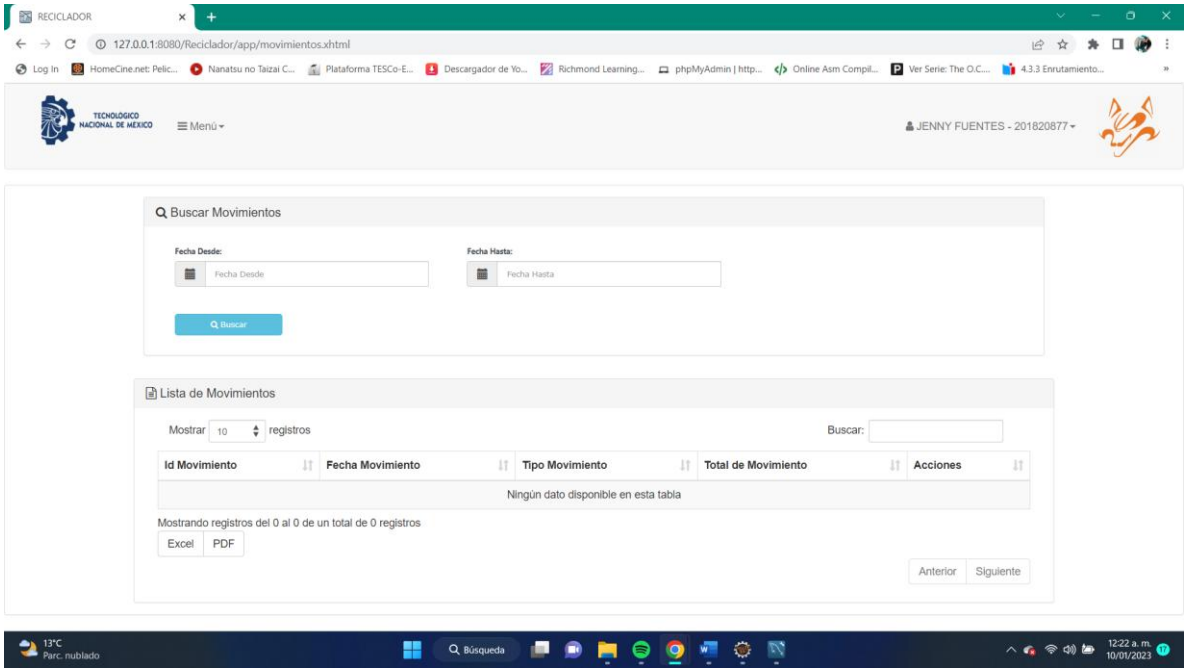


Fig. 53 Pagina de Consulta y filtro de búsqueda de movimientos para el Usuario Regular (Fuente: Propia Autoría, 2022)

La página de movimientos cuenta con un filtro de búsqueda para la consulta de todos los movimientos en base a la fecha del movimiento en la que el usuario realizo dicho movimiento.

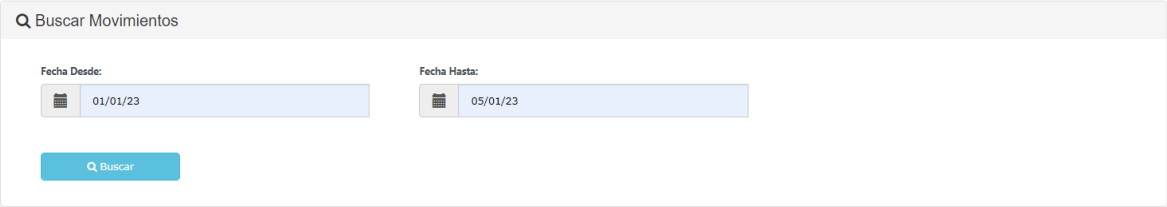


Fig. 54 Filtro de búsqueda para la consulta de movimientos para un Usuario Regular (Fuente: Propia Autoría, 2022)

La siguiente sección cuenta con una tabla de la lista de movimientos con los datos generales de cada uno de los movimientos junto con un botón que abre una ventana que especifica los detalles de cada uno de los movimientos.

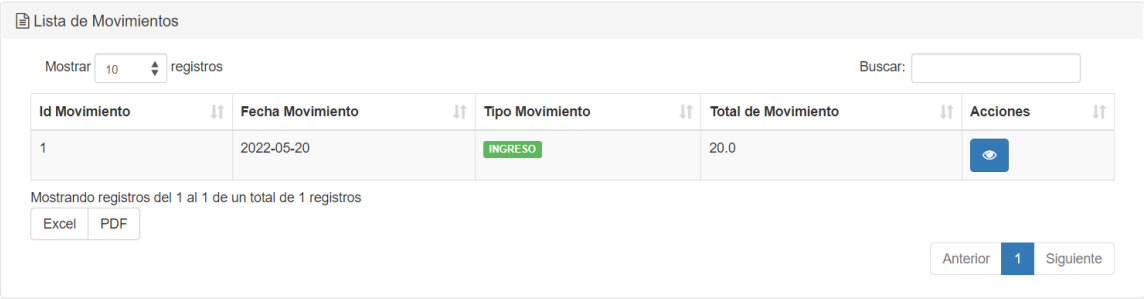


Fig. 55 Tabla de lista de Movimientos de un Usuario Regular (Fuente: Propia Autoría, 2022)

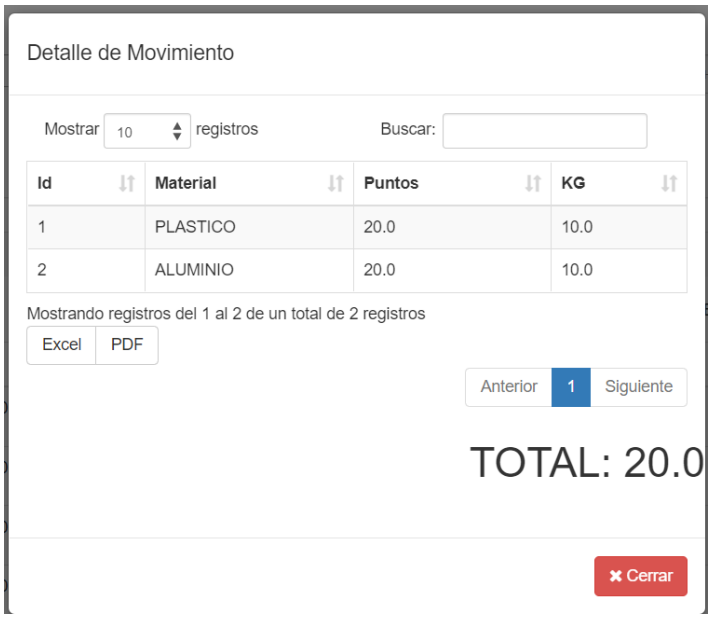


Fig. 56 Ventana Emergente con el detalle del movimiento (Fuente: Propia Autoría, 2022)

CONCLUSIONES

El desarrollo del presente proyecto se han desarrollado y aprendido habilidades importantes para la creación e implementación del Sistema Integral de Dispositivos de Reciclaje, dentro de la empresa INTGEN TECHNOLOGIES DE MEXICO S.A de C.V.

El sistema se desarrolló con éxito por medio de Java, Python y MySQL por lo que el sistema fue construido con distintas tecnologías, para hacer principalmente el cuerpo interno donde se recaba la información, se puede hacer uso de ella y consultar datos de manera constante.

Cabe recalcar que el objetivo general expuesto al inicio del del proyecto, se cumplió satisfactoriamente debido a que se desarrolló un sistema capaz de gestionar a los usuarios registrados en el Sistema Integral de Dispositivos de Reciclaje. El sistema de igual manera es capaz de gestionar los movimientos y materiales d los dispositivos de reciclaje, todo esto lo puede realizar los usuarios administradores mientras que los usuarios básicos, solo serán capaces de ver un histórico de los movimientos elaborados por el mismo.

Entre otras funciones dentro sistema web, el usuario y el administrador, dependiendo el caso, pueden de manera general cambiar contraseñas de manera muy sencilla, en casos más avanzados como con los administradores, tienen forma de ver todos los movimientos, errores y detalles que puedan tener las maquinas en un solo sitio.

Con base en los resultados, donde se automatizo la gestión de los usuarios, máquinas y movimientos de los dispositivos de reciclaje se deduce que se ha logrado la satisfacción del cliente debido a que se cumplieron satisfactoriamente los objetivos generales y específicos planteados,

COMPETENCIAS DESARROLLADAS

En el tiempo de instancia en la empresa INTGEN TECHNOLOGIES DE MEXICO S.A. de C.V, en donde se desarrolló el proyecto nombrado "Módulo de Detección de Materiales del Sistema Integral de Dispositivos de Reciclaje con JAVA PHYTON y MYSQL para el Tecnológico de Querétaro" se utilizaron y desarrollaron diferentes competencias las cuales se presentan a continuación:

- Interpersonales:
  - Capacidad de identificar, analizar y resolver problemas ante una situación inesperada.
  - Adaptabilidad al cambio constante del entorno.
  - Organización para la gestión de tareas y actividades.
  - Aplicación de conocimientos obtenidos de las siguientes materias
    - **Fundamentos de Programación:** Aplica algoritmos y lenguajes de programación para diseñar e implementar soluciones a problemáticas del entorno.
    - **Programación Orientada a Objetos:** Aplica la programación orientada a objetos para resolver problemas reales y de ingeniería
    - **Administración de Bases de Datos:** Instala, configura y administra un gestor de base de datos para el manejo de la información de una organización, optimizando la infraestructura computacional existente.
    - **Programación Web:** Desarrolla aplicaciones web dinámicas del lado cliente y del servidor, considerando la conectividad a orígenes de datos, la interconectividad entre aplicaciones y cómputo en la nube.
    - **Programación Lógica y Funcional:** Conoce los principios lógicos y funcionales de la programación para aplicarlos en la resolución de problemas.
    - **Ingeniería de Software:** Desarrolla soluciones de software, considerando la metodología y herramientas para la elaboración de un proyecto aplicativo en diferentes escenarios.

- Personales
    - Comunicación y trabajo en equipo donde se propusieron y aportaron ideas para el desarrollo del proyecto.
    - Argumentación critica en una situación inesperada.
    - Responsabilidad de la entrega de avances parciales del desarrollo del proyecto.
    - Comprender de manera critica, los requerimientos del proyecto, así como también las necesidades del cliente.
    - Tener criterio y tener la capacidad de resolver problemas a presión mediante la toma de decisiones.

FUENTES DE INFORMACIÓN

- Abduldaem, A. and Gravell, A. (2019), "Principles for the Design and Development of Dashboards: Literature Review", Proceedings of INTCESS 2019 - Equipo Vértice. (2009). Diseño básico de páginas web en XHTML. Editorial Vértice.
- Adrián, Yirda. (Última edición:30 de marzo del 2021). Definición de Autenticación. Recuperado de: https://conceptodefinicion.de/autenticacion/. Consultado el 28 de abril del 2021
- Groussard, T. (2012). JAVA 7: Los fundamentos del lenguaje Java. Ediciones Eni.
- Brown, T. B. (2015). CSS Master: Organized, Fast Efficient - CSS Done Right! Sitepoint.
- Chacon, S., & Straub, B. (2014). Pro Git (English Edition) (2.a ed.). Apress.
- Date, C. (2012). Database design and relational theory. Van Duuren Media.
- Valenzuela Chancosa, M. L. (2017). Implementación de un sistema para la automatización de procesos de recaudación en la junta central de regadío acequia Rosauco Zapallo Pogyo, mediante los Frameworks Bootsfaces y Primefaces en la plataforma JEE (Bachelor's thesis)
- Francis, M. (2020, June 26). The importance of data analysis. Resagratia. https://bit.ly/3dXoshG
- IOSR Journal Of Humanities And Social Science, Bhise, R. B., Thorat, S. S., & Supekar, A. K. (2017, February). "Importance of data mining in higher education system (Vol. 6). https://bit.ly/2Q22CBr

- Krasner, Glenn & Pope, Stephen. (1988). A Description of the Model-ViewController User Interface Paradigm in the Smalltalk80 System. Journal of Object-oriented Programming - JOOP. 1.
- Martínez, C. S. (2017, 26 octubre). Framework, ¿Qué es y por qué debería o no usarlo? Loopeando. https://www.loopeando.com/que-es-un-framework/
- Morteo, Bocalandro, M. B., & Francisco Nicolas, F. N. (2004). Un enfoque practico de MySQL (1.a ed., Vol. 1). Ediciones Cooperativas.
- Nazir, S., Khan, S., Khan, H. U., Ali, S., Garcia-Magarino, I., Atan, R. B., & Nawaz, M. (2020). A Comprehensive Analysis of Healthcare Big Data Management, Analytics and Scientific Programming. IEEE Access, 8, 95714–95733. https://doi.org/10.1109/access.2020.2995572
- Keith, M., Schincariol, M., & Keith, J. (2011). Pro JPA 2: Mastering the JavaTM Persistence API. Apress.
- Bauer, C., & King, G. (2005). Hibernate in action (Vol. 1). Greenwich CT: Manning.
- Reenskaug, T. (2003). The Model-View-Controller (MVC) Its Past and Present. Santiago Cely, C. (2006, 24 junio). Autenticación de usuarios. oas.org. https://bit.ly/2Su1p6X
- Luján-Mora, S. (2002). Programación de aplicaciones web: historia, principios básicos y clientes web. Editorial Club Universitario. Ramos Martín, A., & Ramos Martín, M. J. (2014). Aplicaciones Web 2. Ediciones Paraninfo, SA.
- Ferris, C., & Farrell, J. (2003). What are web services?. Communications of the ACM, 46(6), 31.
- Mularien, P. (2010). Spring Security 3 (p. 18). Birmingham,, England: Packt Publishing.
- Pech-May, F., Gomez-Rodriguez, M. A., Luis, A., & Lara-Jeronimo, S. U. (2012). Desarrollo de Aplicaciones web con JPA, EJB, JSF y PrimeFaces. Instituto Tecnológico Superior de los Ríos, Tabasco, México.