



INSTITUTO TECNOLÓGICO SUPERIOR DE TEZIUTLÁN

Tesis



“Mejora de mantenimiento correctivo mediante notificaciones con señalética para la empresa Confetex S.A. de C.V”

PRESENTA:

JUAN DIEGO PARRA MÉNDEZ

CON NÚMERO DE CONTROL
17TE0215

PARA OBTENER EL TÍTULO EN EL PROGRAMA DE:
INGENIERA EN SISTEMAS COMPUTACIONALES

CLAVE DEL PROGRAMA ACADÉMICO
ISIC-2010-224

DIRECTOR(A) DE TESIS:
D.I.E CARREÓN ROMERO MARÍA EUGENIA

“La Juventud de hoy, Tecnología del Mañana”

TEZIUTLÁN, PUEBLA, SEPTIEMBRE 2022

PRELIMINARES

Agradecimientos

A MIS PADRES

Quienes con su amor, paciencia y esfuerzo me han permitido cumplir otra de mis metas

A MIS HERMANOS

Por la paciencia y apoyo que me han brindado durante todo este proceso

A MI PAREJA

La persona que me apoyo en cada momento y me inspiro a ser cada día mejor.

A MIS MAESTROS Y ASESORA

Por la paciencia, apoyo, y sobre todo por confiar en mí.

Resumen

La manera en la que una industria es capaz de reaccionar a los cambios cuándo sucede un problema es de suma importancia, para mantener toda su maquinaria y equipos funcionando a su máxima capacidad.

Este proyecto tiene como objetivo mejorar el proceso en el que se realiza el mantenimiento correctivo de las líneas de producción dentro de CONFETEX, S.A. de C.V. Mediante una aplicación Web, desde la que puedan acceder todos los usuarios dentro de la planta, supervisores y mecánicos, para llevar un control de las solicitudes de reparación de maquinaria, y el historial de reparaciones que ocurrieron dentro de la planta.

Índice general

1.1	Introducción	5
1.2	Objetivo general	7
1.2.1	Objetivos particulares.....	7
1.3	Justificación.....	7
1.4	Propuesta de solución	9
1.5	Viabilidad	10
1.5.1	Viabilidad económica.....	10
1.5.2	Viabilidad operacional	11
2.1	Definición de mantenimiento correctivo.....	8
2.2	Estándares de mantenimiento correctivo.	9
2.2.1	Mantenimiento correctivo planificado.	9
2.2.2	El mantenimiento correctivo no planificado.....	9
2.3	Técnicas y procesos de mantenimiento correctivo	9
2.4	Beneficios que ofrece el mantenimiento correctivo.....	10
2.5	Desventajas del mantenimiento correctivo	10
2.6	Aplicaciones web	11
2.7	UML.....	12
2.7.1	Diagrama de secuencia UML.....	12
2.8	Diseño	13
2.8.1	Wireframes.....	13
2.8.2	Prototipo	13
2.8.3	Balsamiq	14
2.9	Modelo entidad de relación.....	14

2.10 Herramientas y lenguajes usados.....	15
2.10.1 HTML	15
2.10.2 CSS.....	15
2.10.3 JavaScript.....	16
2.10.4 PHP.....	17
2.10.5 SQL.....	18
2.10.6 Apache.....	19
2.10.7 Git	20
2.10.8 Visual Studio Code	21
2.11 Metodología ágil	21
2.11.1 Scrum	21
2.11.2 Fases	22
3.1 Etapa de requerimiento y análisis.	23
3.1.1 Usuarios.....	23
3.1.1.1 Administrador.	23
3.1.1.2 Supervisor.	23
3.1.1.3 Mecánico.....	23
3.2 Requisitos específicos.....	23
3.3 Casos de uso.....	25
3.2 Etapa de diseño.....	37
3.2.1 Diagramas de secuencia.....	37
3.2.2 Prototipo del proyecto.....	39
3.6 Recolección de información	46
3.7.1 Inicio	49

3.7.2 Planificación y estimación	49
3.7.3 Implementación.....	49
4.1 Presentación del software, descripción de interfaces	50
4.1.1 Pantalla de inicio.....	51
4.1.2 Bitácoras	51
4.1.2 Usuarios.....	52
4.1.2 Solicitudes.....	53
4.1.3 Reparaciones.....	53
4.1.4 Máquinas	54
4.1.5 Líneas de producción	54
4.2 Desarrollo de la metodología de investigación	55
5.1 Conclusiones del proyecto, recomendaciones y experiencia profesional adquirida	57
5.2 Conclusiones relativas a los objetivos específicos	57
5.3 Conclusiones relativas al objetivo general.....	57
5.4 Limitaciones del modelo planteado	57
5.5 Recomendaciones	58
6.1 Competencias desarrolladas y/o aplicadas	59
6.2 Competencias genéricas.....	59
6.3 Competencias específicas.....	59
7.1 Referencias	60
8.1 Diagrama de base de datos	64
8.2 Anexo de hoja de liberación	65
8.3 Carta de liberación.....	66

8.4 Índice de figuras.....	67
----------------------------	----

CAPÍTULO I

GENERALIDADES DEL PROYECTO

1.1 Introducción

Según un estudio realizado, con relación al total de la producción nacional de fabricación de prendas de vestir, el 8.5 es generado por empresas manufactureras del estado de Puebla. A saber, dentro de Puebla existen 141 municipios, y según Hernández (2020) de los cuales, tres de ellos generan el 68% de la producción estatal total, convirtiéndose así en un referente de esta actividad económica, y cuyos valores se enuncian a continuación:

- Teziutlán 29.2%
- Puebla 23.5%
- Tehuacán 15.3%

Como se ha evidenciado, la industria manufacturera representa una de las principales actividades económicas del estado, existiendo una afluencia significativa de empresas que pertenecen a este gremio, por ello, es necesario y emergente que dichas empresas cuenten con las capacidades y herramientas necesarias para implementar estrategias de gestión en el mantenimiento correctivo de su maquinaria y equipo, que a la par, le permita aumentar los niveles de productividad de las maquinarias con las que cuenta la organización, con el objetivo de ofrecer productos que cumplan cabalmente con las fechas de entregas estipuladas por sus clientes.

El proceso de mantenimiento en cualquier tipo de empresas es necesario para la correcta funcionalidad de todas las partes que la conforman, el mantenimiento correctivo tiene la función de corregir todas aquellas fallas y problemas que se presentan en la maquinaria e instalaciones en general, con el fin de regresar a su estado original de funcionalidad, es por ello por lo que surgen protocolos de mantenimiento y normas a seguir para lograrlo de manera eficaz, sin embargo, en lo mejor de los casos es necesario que se implemente la mejora de procesos productivos que sugieren el mudar todos los formatos y medidas antiguas a un nuevo, el cual beneficie para obtener mejoras en cuanto a productividad y mejorar

la parte visual de la interpretación de los datos e información recopilada y presentada.

Un sistema de gestor, con toda la información generada por estos procesos, es capaz de mejorar la eficiencia y productividad, así como la posibilidad de identificar de manera más sencilla las principales fallas y/o problemas presentados en el transcurso de las tareas realizadas por los mecánicos en el proceso de mantenimiento, tanto preventivo como en el correctivo, y finalmente, es idóneo para facilitar la gestión en las solicitudes de refacciones de la maquinaria y contabilizarlas de la misma manera.

1.2 Objetivo general

Mejorar el proceso de mantenimiento correctivo de la maquinaria que conforman las líneas de producción de la empresa CONFETEX, S.A. de C.V. mediante una aplicación web que genere notificaciones al personal de mantenimiento al identificarse un fallo mecánico.

1.2.1 Objetivos particulares

1. Identificar el proceso necesario para llevar a cabo una solicitud de mantenimiento creada por el supervisor de la línea y la reparación realizada por el mecánico.
2. Desarrollar un flujo dentro de la aplicación Web que conduzca a reducir tiempos y los costos asociados al mantenimiento correctivo.
3. Identificar las actividades más frecuentes que se realizan dentro del mantenimiento correctivo.
4. Almacenar un historial de fallas dentro del sistema, así como las solicitudes de mantenimiento realizadas.

1.3 Justificación

Confecciones Textiles de Teziutlán (CONFETEX S.A. de C.V) fué creado el 5 de marzo del año 1996 con iniciativa del Ing. Alejandro Garza Anaya. Se ubica en la Av. San Rafael No. 65, sección 11, en la ciudad de Teziutlán, Puebla. El giro de la empresa es la elaboración de jeans de diferentes modelos y marcas.

La empresa inició operaciones con 25 trabajadores y 20 máquinas de coser, el mobiliario se encontraba ubicado en una bodega de aproximadamente 11 x 30 mts. La capacidad de producción al inicio era de tan sólo 800 prendas semanales.

En el inicio de vida de la compañía, se implementó un control para el mantenimiento de la maquinaria y el equipo, el cual consistía en un plan de mantenimiento correctivo plasmado en papel, sin embargo, dicho sistema implicaba una fuerte

inversión de tiempo y generaba conflictos para acumular historiales durante largos periodos de tiempo.

Empero, debido al crecimiento exponencial que ha experimentado la empresa, hoy por hoy, es necesario e inmediato implementar nuevos sistemas, o bien, sistemas modificados para el control del mantenimiento que le permitan adaptarse a las nuevas demandas y fallas a las que se exponen sus equipos, de un sistema que optimice el tiempo y mejore el control y el flujo de historiales de la maquinaria.

La modificación al actual sistema de control que se emplea, mediante una aplicación web, puede contribuir a la disminución del tiempo de respuesta para la reparación de una máquina dañada por parte del personal de mantenimiento. Para lograr este objetivo, se requiere de un registro de las solicitudes creadas y atendidas, a fin de generar un historial de información que evidencie dos importantes factores; el tiempo que invierte el personal de mantenimiento en reparar una máquina y las reparaciones que resulten ser las más críticas para una línea de producción (aquellas que requieren mayor tiempo para ser atendidas).

El historial de fallas y reparaciones permitirá: 1) anticipar la adquisición de las piezas que con más asiduidad requieren reemplazo, 2) identificar las máquinas que con más frecuencia presentan fallas y su tipo de falla, 3) categorizar marcas y modelos de maquinaria que presentan mayor periodicidad en fallas o daños, 4) Jerarquizar la maquinaria que requirió mayor atención, entre otras.

1.4 Propuesta de solución

Desarrollar una aplicación web que se ejecute dentro de las instalaciones de CONFETEX. La aplicación está diseñada para ser empleada por tres importantes departamentos de la organización y en los cuales recae la toma de decisiones para el mantenimiento del equipo, a saber, personal de mantenimiento (mecánicos), encargados de línea (supervisores) y administrador, cuya descripción se enuncia a continuación:

El administrador tendrá mayor control y gestión de la aplicación, mediante ésta, podrá crear o añadir nuevas líneas de producción, agregar, cambiar, o quitar mecánicos, agregar, cambiar o quitar supervisores, agregar bitácoras y visualizar el estatus del sistema, así como descargar los reportes de las solicitudes por la periodicidad que mejor se ajuste a sus necesidades (día, semana o mes).

El administrador del sistema tendrá acceso al estado de las solicitudes en CONFETEX, dentro de la sección de bitácoras podrá observar un código de colores que indica el estado de cada bitácora, el cual se expresa a continuación:

- Rojo: Solicitud nueva sin respuesta.
- Amarillo: Solicitud nueva que está siendo atendida.
- Verde: No hay solicitudes

El formato del reporte que la aplicación puede generar incluye los siguientes datos:

1. Línea de producción.
2. Fecha del periodo a la que pertenece.
3. Prioridad que asignó el supervisor a la solicitud que generó.
4. Nombre de la operación que generó el mecánico.
5. Número o identificador de la máquina que se atendió.
6. El módulo al que pertenece la máquina.
7. El código del problema.
8. La hora en la que el supervisor hizo la solicitud.

9. La hora a la que llego el mecánico.
10. La hora a la que la máquina quedo reparada.
11. El tipo de reparación que el mecánico realizo.
12. El nombre del mecánico que reparo la máquina.

Por su parte, los supervisores pueden crear solicitudes de mantenimiento para alguna de las bitácoras que previamente fueron creadas por el administrador, cuando un supervisor crea una solicitud de mantenimiento, inserta los datos correspondientes y la aplicación hace la notificación al mecánico. Cabe destacar que el supervisor es el responsable de colocar la hora de llegada del mecánico después de que generó la solicitud.

Mientras tanto, los mecánicos pueden ver las solicitudes que les fueron realizadas en la línea de producción a la que fueron asignados, es decir, de la que son responsables. Cuando llega una solicitud de mantenimiento nueva les aparece en su inicio y son los responsables de llenar sus datos.

1.5 Viabilidad

Un software de administración de mantenimiento centra, utiliza y analiza la información acerca de las labores de mantenimiento en una compañía y permite organizar y manejar con mucha efectividad. Ayudando a priorizar dichas labores y garantizar que los puntos del proceso se encuentren donde corresponden.

Manejar el sistema es fácil, es fácil de entender y fácil de generar los datos. Se conoce y se domina las herramientas necesarias para el desarrollo.

1.5.1 Viabilidad económica.

Las tecnologías necesarias para este proyecto son completamente de código abierto, eso quiere decir que son gratis. Para el desarrollo de la app se cuenta con los instrumentos requeridos: Como son las herramientas de hardware de las que el

sistema depende es fácil de conseguir y es barato. No sé necesita una PC con gran cantidad de recursos.

1.5.2 Viabilidad operacional

La empresa CONFETEX cuenta con una laptop con un procesador Intel i3 y sistema operativo Windows 10. La aplicación puede ser ejecutada sin problemas por una sola persona.

CAPÍTULO II

MARCO TEÓRICO

2.1 Definición de mantenimiento correctivo.

Para realizar el mantenimiento correctivo en una empresa es necesario que intervenga un mecánico especializado, que labore cuando ocurra un desperfecto, ya sea un cambio de alguna refacción, o una pieza de la maquinaria (Textilínca, s. f.). El mantenimiento correctivo ocurre inmediatamente para corregir los defectos o fallas técnicas que se presentan en un artefacto.

Al hacer un óptimo mantenimiento correctivo, se considera la limpieza y lubricación de la máquina, así como también la observación y anotaciones del mecánico. Lo más relevante a considerar son las opiniones del mecánico, debido a que determina e indica la compostura o tarea a realizar para que la máquina funcione correctamente.

“El mantenimiento correctivo es más costoso que el preventivo, implica que la empresa deba encontrar más rápido la refacción, y en algunos casos el personal técnico para la reparación. El mantenimiento correctivo puede costar un 200% más que el preventivo a causa del tiempo y cantidad de paradas y las averías”. (BYRON & CASTRILLÓN, 2019)

“Hay varios tipos de mantenimiento correctivo, puede ser no planeado, programado o de emergencia. El mantenimiento correctivo no planeado es cuando se corrige una falla que no se espera. El mantenimiento correctivo programado es cuando se conoce el origen de la falla y se diagnostica el problema, pero se espera a un momento óptimo para la reparación. El tercero es cuando se corrigen las fallas de inmediato”. (BYRON & CASTRILLÓN, 2019)

2.2 Estándares de mantenimiento correctivo.

Los estándares de mantenimiento evalúan el estado de los equipos, indican los métodos para medir el deterioro de los equipos para la inspección y la prueba, detener el progreso del deterioro mantenimiento diario de rutina y restablecer las condiciones de los equipos (Sagnier, 2021).

En caso de ocurrir algún fallo se debe reestablecer el sistema en su estado funcional, siendo este la meta de este tipo de mantenimiento. Existen dos tipos diferentes de este mantenimiento el planificado y el no planificado.

2.2.1 Mantenimiento correctivo planificado.

En el mantenimiento correctivo, se deja el hecho de que el artefacto falle, para posteriormente usar una propuesta de mantenimiento correctivo en la cual se decide si la máquina es reparada o es reemplazada.

“Una estrategia de mantenimiento correctivo planificado sólo debe utilizarse con activos que pueden ser fácilmente reemplazados. Este enfoque se utiliza cuando sólo se ha aplicado el mantenimiento correctivo”. (Sagnier, 2021)

2.2.2 El mantenimiento correctivo no planificado.

El mantenimiento de este tipo, en cierta manera, forma parte de la estrategia en el mantenimiento preventivo. Cuando ocurre un fallo, es en ese momento donde se identifican los problemas, antes de que se agraven, y consiste en resolverlos en el transcurso de dicha situación.

2.3 Técnicas y procesos de mantenimiento correctivo

La administración de las interacciones en la empresa puede mejorar con un mayor conocimiento en distintos niveles de mantenimiento correctivo.

Según Talva (2021), existen 5 niveles de mantenimiento.

- Las reparaciones sencillas son asociadas al nivel uno, siendo estos, solo elementos a los cuales es fácil acceder. Generalmente el encargado de producción lleva estas operaciones y las planifica en un calendario.
- El mantenimiento de segundo nivel incluye reparaciones que requieren procedimientos sencillos, y deben ser realizadas por un personal calificado, por lo tanto, requiere procedimientos detallados.
- El tercer nivel hace alusión a intervenciones requieran un mantenimiento complejo, y así mismo se necesitan técnicos calificados.
- Para el cuarto nivel, las intervenciones que requieran un mantenimiento complejo son realizadas y supervisadas por un técnico o dependiendo de la situación se requerirá un equipo especializado.
- El mantenimiento de quinto nivel incluye las intervenciones realizadas por un fabricante de equipamientos o una empresa autorizada por el fabricante

2.4 Beneficios que ofrece el mantenimiento correctivo.

(Aner, 2020) Identifica las ventajas del mantenimiento correctivo dentro de una empresa textil que se enumeran a continuación.

1. La primera ventaja sería que los costos son menores en el periodo de un corto plazo.
2. La segunda utilidad radica en requerir solo la mínima planificación.
3. Todo lo relacionado al proceso sería más fácil de controlar.
4. En ciertos casos se podría objetar este mantenimiento como la mejor solución.

2.5 Desventajas del mantenimiento correctivo

Según (Aner, 2020) el mantenimiento correctivo puede llegar a tener problemas secundarios o desventajas tales como: La puede ser la imprevisibilidad, otro problema sería el tiempo de inactividad, también puede ocurrir que el equipo no este maximizado y ciertos problemas ocasionen mayores costos a un largo plazo.

2.6 Aplicaciones web

Las aplicaciones Web se ejecutan en internet. Una aplicación Web procesa y almacena los datos o los archivos requeridos para trabajar dentro de la web. Para utilizar una aplicación web es necesario acceder a un servidor web a través de internet o intranet mediante un navegador (GCFglobal, s. f.).

Las aplicaciones web son populares debido a lo prácticas que pueden llegar a ser, no requieren de un dispositivo de gran capacidad de cómputo, no dependen de un sistema operativo, y a la vez para los desarrolladores, ya que tienen una gran facilidad para mantener y actualizar aplicaciones web.

Ventajas de las aplicaciones web

- No se requiere instalar debido a que se ingresan a ellas a mediante de un navegador.
- Se denomina multiplataforma a una aplicación Web y también se considera que es multidispositivo.
- No se requiere de gran capacidad de cómputo, puesto que la aplicación se soporta en el servidor dónde se aloja.
- Se puede acceder a una aplicación web a través de cualquier dispositivo móvil que tenga acceso a internet o Intranet.
- Es adaptable y fácil de usar.

Usos comunes:

- Permitir a los usuarios localizar de manera rápida y eficaz datos que se almacenan en un servidor web.
- Recoger, guardar, analizar y procesar datos suministrados por los usuarios.

2.7 UML

En el mundo de la ingeniería de software, es de gran importancia los módulos que describan el diseño y proceso de "algo". Siendo que este "algo" pueda estar en desarrollo, o estar planeado.

Los que diseñan el sistema que se está desarrollando o se desarrollará, deben de buscar y definir los requerimientos necesarios para terminar el producto, en los requerimientos necesarios se incluye el rendimiento y la funcionalidad del programa, así como también se debe considerar la contabilidad. Además, el modelo a menudo se divide en módulos, de los cuales cada uno de ellos describe un aspecto específico del producto o sistema en construcción (Lucidchart, s. f.-a).

UML es una herramienta para la especificación de sistemas desde la etapa de la idea hasta la etapa de finalización. Con UML se ha creado un lenguaje de modelado visual común, semántico y sintácticamente rico para la arquitectura, el diseño y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento de un requerimiento (Román Zamitiz, s. f.).

2.7.1 Diagrama de secuencia UML

En el diagrama de secuencia se explica cómo y en qué orden los objetos funcionan en conjunto, por ello se considera que es de tipo interacción. Se usan estos diagramas para comprender los requisitos de un sistema nuevo o documentar un proceso existente. (Lucidchart, s. f.-b).

2.7.1.2 Los beneficios de los diagramas de secuencia

Este tipo de diagrama es útil para las compañías. Los beneficios más comunes de los diagramas de secuencia, según Lucidchart (s. f.-b) son:

- Los diagramas de caso de uso son mostrados en detalle.

- Representan los procesos lógicos y complejos.
- Los procesos muestran la interacción entre los objetos de distintos componentes.
- Planifica y comprende de manera detallada la funcionalidad ya sea este en el presente o a largo plazo.

2.8 Diseño

Diseñar la interfaz de usuario, es el proceso de desarrollar, con el que interactuará un usuario cuando se comunica con un software.

2.8.1 Wireframes

Los Wireframes son el esqueleto o plano del diseño, es útil para mantener una comunicación clara entre diseñadores y programadores, sobre la construcción de la estructura de la aplicación que se esté desarrollando. Contienen una menor fidelidad visual, se muestran en escalas de grises, puesto que no se le da demasiada importancia al aspecto estético. Se utilizan escalas de grises con el propósito de que no haya un excesivo diseño visual o gráfico que pueda distraer la discusión de lo realmente importante en un Wireframe: el concepto, la estructura y los componentes básicos del diseño (Herranz, 2018).

En este se consideran los componentes que utilizara la aplicación final, sin definir a detalle la parte visual.

Los Wireframes están contruidos para no ser el diseño final, la baja fidelidad y los pocos colores hacen posible centrarse en la estructura por sobre los detalles. Los Wireframes están diseñados para transmitir todo lo que está en discusión, son rápidos de hacer, por lo tanto, hacen que se sea muy fácil dar retroalimentación.

2.8.2 Prototipo

Estos representan no de manera exacta el diseño del software, sin embargo, indican la forma en que el usuario interaccionara con la interfaz (UI). El usuario puede usar

y experimentar con el producto. A diferencia del Wireframe que define la estructura, el prototipo define sobre todo cómo se comporta el producto.

En esta etapa los usuarios son importantes, ellos evaluarán al prototipo y se realizarán las modificaciones necesarias para un óptimo rendimiento de interfaz. Al funcionar de manera simulada, con un aspecto visual más parecido al del producto final, son más adecuados para evaluar la experiencia que tienen los usuarios interactuando con el producto.

2.8.3 Balsamiq

Balsamiq es una herramienta para crear Wireframes, tiene un conjunto de funciones enfocadas a crear Wireframes de manera sencilla, intuitiva y rápida.

La aplicación permite que equipos de UI/UX transmitan rápidamente sus ideas a los diseñadores creativos sin la expectativa de decidir la marca visual del sitio. Si bien el conjunto de funciones es bastante limitado y la herramienta no se ha actualizado recientemente, sigue siendo el método más eficiente para elaborar Wireframes.

- Fácil de aprender y excelente para usuarios avanzados.
- Capacidad para descargar componentes prefabricados desde una biblioteca central o compartir componentes fácilmente dentro de un proyecto

2.9 Modelo entidad de relación.

Un diagrama ER, ilustra la relación de las entidades y objetos dentro de un software. Los diagramas entidad relación son usados para elaborar BD relacionales. Emplea un conjunto definido de símbolos, como rectángulos, diamantes, óvalos y líneas de conexión, para representar la interconexión de entidades, relaciones y sus atributos (Lucidchart, s. f.-b).

2.10 Herramientas y lenguajes usados

2.10.1 HTML

Lenguaje de Marcas de Hipertexto o HTML por sus siglas en inglés, es un elemento muy básico e importante para el etiquetado de datos, muestra la estructura a considerar para el contenido web.

El hipertexto es usado al referenciar otras piezas del texto, y el lenguaje de marcado sirve para que los servidores web identifiquen la estructura del documento.

Afirmando lo que dice B. (2021b) HTML no es un lenguaje de programación, debido a que este no crea funcionalidades dinámicas. Mas bien crea la estructura de las secciones, párrafos y links usando elementos, etiquetas y atributos.

Los casos de uso más comunes para HTML se encuentran en:

1. Desarrollo web: Los desarrolladores web usan código HTML para diseñar como un navegador web mostrara los elementos.
2. Navegación a través de internet: El usuario puede navegar fácilmente, también puede utilizar enlaces entre sitios web.
3. Documentación Web: HTML es muy útil para modificar y organizar los documentos de la misma manera que lo haría Microsoft Word.

2.10.2 CSS

Hojas de Estilo en Cascada o CSS por sus siglas en inglés, es usado para especificar la apariencia en los documentos HTML en un sitio web.

La organización Mozilla (2021a), describe que:

“CSS describe como debe ser renderizado el elemento estructurado en la pantalla o en otros medios. Usando CSS se puede controlar el color del texto, el estilo de las fuentes, el espacio entre párrafos, la manera en la que colocan

las columnas y una gran variedad de efectos, transiciones entre otras cosas a una página web.”

CSS es creado y mantenido a través de un grupo llamado el W3C Group. Este grupo sin ánimos de lucro elabora ciertos documentos denominados especificaciones. En el momento en que una especificación fue discutida y es oficialmente aprobada por los miembros de dicho grupo, esta especificación se vuelve una sugerencia.

2.10.2.1 Bootstrap (CSS framework)

Bootstrap es un framework del lado del cliente utilizado para desarrollar aplicaciones web y sitios responsivos, es decir, los sitios web están diseñados para ser funcionales y ser visualizados en computadores de escritorio, en las distintas tabletas y celulares, de forma simple y ordenada.

Proporciona herramientas para la interactividad del sitio, maneja distintos componentes como son el menú de navegación, los controles de la paginas, entre otros, con el objetivo de facilitar la comunicación con el usuario.

2.10.3 JavaScript

Es un lenguaje de programación que se encarga de ofrecer a los sitios web una mayor interactividad y mejor dinamismo, este un lenguaje ligero, también es interpretado, o es compilado “justo a tiempo”.

Mozilla lo define como:

“...un lenguaje de scripting para páginas web, y es usado en muchos entornos fuera del navegador, tal como Node.js, etc. JavaScript es un lenguaje de programación basada en prototipos, multiparadigma, de un solo hilo, dinámico, con soporte para programación orientada a objetos, imperativa y declarativa.”

2.10.4 PHP

PHP es un lenguaje de programación destinado a desarrollar aplicaciones para la web y crear páginas web, favoreciendo la conexión entre los servidores y la interfaz de usuario (de Souza, 2021).

Una de las razones por las que PHP es utilizado por muchos programadores es debido a que es de código abierto.

1. Este lenguaje no tiene delimitaciones al momento de usarlo que estén relacionadas a los derechos. El desarrollador puede utilizar PHP en cualquiera de sus proyectos y comercializarlos sin algún problema.
2. PHP es constantemente perfeccionado, porque su comunidad de desarrolladores es proactiva y está comprometida al mejoramiento del lenguaje.

2.10.4.1 Laravel (Framework)

PHP cuenta con distintos frameworks entre estos se encuentra Laravel que es muy fácil de manipular además de ser de código abierto. Arsys (2021) lo define como: "simple, muy potente y tiene una interfaz elegante y divertida de usar. Fué creado en 2011 y tiene una gran influencia de frameworks como Ruby on Rails, Sinatra y ASP.NET MVC".

"El objetivo de Laravel es el de ser un framework que permita el uso de una sintaxis refinada y expresiva para crear código de forma sencilla, evitando el «código espagueti» y permitiendo multitud de funcionalidades. Aprovecha todo lo bueno de otros frameworks y utiliza las características de las últimas versiones de PHP." (de Zúñiga, 2021b)

2.10.4.2 Blade

"Laravel usa Blade para la creación de vistas HTML basadas en plantillas. Esta librería permite realizar todo tipo de operaciones con datos, además de la sustitución de

secciones de las plantillas por otro contenido, herencia entre plantillas, definición de layouts o plantillas base, etc.” (*Plantillas mediante Blade - Laravel 5*, s. f.)

2.10.5 SQL

Lenguaje de consulta estructurado o SQL por sus siglas en inglés, es un lenguaje usado por los sistemas de gestión de BD relacionales. Entre algunos sistemas se puede mencionar a Oracle y MySQL.

Es utilizado para gestionar los datos, la información en las BD relacionales está organizada y estructurada en tablas. SQL permite controlar y manipular la información de manera fácil, las tablas de datos pueden ser consultadas, actualizadas, eliminadas e incluso modificarlas tanto en información como estructura y es conveniente relacionar varios registros incorporados en las tablas, considerando que tengan campos en común.

“A diferencia de las hojas de cálculo como por ejemplo Microsoft Excel, SQL está diseñado para compilar y administrar datos en volúmenes mayores. Si bien las hojas de cálculo se pueden volver difíciles con mucha información, las bases de datos pueden manejar millones, o incluso miles de millones de celdas de datos.” (Pickett, 2020)

2.10.5.1 MariaDB

Debido a la compra de MySQL por parte de Oracle, MySQL dejó de ser completamente Open Source, y se acerca a ser software comercial, existe una versión comunitaria que se puede usar de manera gratuita, pero el desarrollo de MySQL se basa en los intereses de Oracle.

Para resolver esta situación se desarrolló un sistema de administración de bases de datos relaciones (DBMS) denominado MariaDB, siendo este un reemplazo que es

compatible con tecnología de MySQL. Es desarrollado para ser de código abierto y es una derivación de MySQL por los programadores que realizaron la BD original.

“MariaDB se basa en SQL y admite el procesamiento de datos al estilo ACID, es decir: Atomicidad, consistencia, aislamiento y durabilidad garantizados para las transacciones. Entre otras características MariaDB admite objetos JSON, replicación de datos en paralelo y múltiples motores de almacenamiento incluidos InnoDB, MyRocks entre otros.” (Vaughan, 2018)

MariaDB está disponible a través de la licencia GPL y tiene librerías y clientes para todos los lenguajes de programación.

2.10.6 Apache

Existen diferentes tipos de servidor, unos se especializan en archivos, otros en BD, también están los servidores de correo, y los servidores web, entre otros. Todos ellos permiten el acceso a archivos de información que son almacenados en servidores físicos y usar dicha información para distintas finalidades.

“El trabajo de un servidor web es servir sitios web en internet, para ello, actúa como un intermediario entre el servidor y el cliente. Extrae el contenido del servidor en cada solicitud y lo envía de nuevo al usuario a través de la Web.” (B., 2021a)

Un servidor Web procesa las solicitudes de cada usuario de manera distinta, dichos servidores emplean y procesan distintos lenguajes de programación como, por mencionar a algunos serían Python, Java, y PHP, entre otros. Un servidor web convierte los archivos HTML estáticos y los sirven para que el usuario los pueda visualizar en un navegador.

Un software de código abierto y enfocado a servicios web gratuitos sería Apache HTTP Server, el cual otorga a los sitios web mostrar información en la web, por ello se le llama “servidor web”. Este es un servidor web antiguo, su primera versión fue lanzada en 1995, siendo hace más de 20 años.

2.10.7 Git

Es una herramienta utilizada para el control de versiones, y es la más usada en el mundo. Es un proyecto de código abierto. Linus Torvalds creó Git, y él también es famoso por ser el creador del Kernel del sistema operativo Linux. Cuando se realizan cambios en un archivo, Git hace un seguimiento de estos, para tener un registro de lo modificado y del mismo modo acceder a versiones posteriores u específicas si es requerido. Una característica de Git es el hecho de que permite la colaboración de varias personas en un mismo archivo o proyecto, facilitando que los cambios realizados por cada una de ellas, sobre el código fuente se fusionen en una sola fuente.

Git es un software que se ejecuta localmente, sus archivos y su historial se almacenan de manera local. También se puede usar servicios como GitHub o Bitbucket para almacenar una copia de los archivos, su historial y sus revisiones de código. Tener un lugar centralizado donde se pueda cargar y descargar los cambios de código facilita la colaboración con otros desarrolladores. Git es capaz de fusionar automáticamente los cambios, por lo que dos o más personas pueden trabajar en diferentes partes del mismo archivo y luego fusionar esos cambios sin perder su trabajo.

Está estructurado en ramas lo cual permite que sea potente, cada una de las ramas está destinada a contener proyectos divergentes que pertenecen a un proyecto principal, ya sea para realizar experimentos o encontrar nuevas funcionalidades.

Estas pueden enfocarse o contener una línea de progreso totalmente diferente al enfoque de la ramificación principal, la cual contiene el desarrollo central en la aplicación. En cualquier lugar o momento es posible fusionar las ramas con la rama principal, y todo esto lo gestiona Git de manera eficiente.

2.10.8 Visual Studio Code

Es un editor cuyo código fuente es gratuito y se distribuye bajo la licencia MIT creado por Microsoft. Está disponible para los distintos sistemas operativos como son Linux, MacOS y Windows. A pesar de que es un editor bastante ligero, viene con herramientas poderosas que lo hacen ser uno de los editores de texto más populares en los últimos años.

VS Code admite una gran amplia gama de lenguajes de programación desde Java, C++ y Python hasta CSS, Go y otros. VS Code permite agregar y crear nuevas extensiones como linters de código, depuradores y soporte de desarrollo web y en la nube.

Entre las funciones integradas de VS Code se encuentran: la capacidad para depurar el código, el control de versiones con Git Integrado, el resaltado de sintaxis, los fragmentos de código. Además, VS tiene soporte para IntelliSense, que es un término general para funciones de edición de código que incluye acciones para: finalización de código, información de parámetros, información rápida de las funciones y miembros de objetos y clases.

2.11 Metodología ágil

2.11.1 Scrum

Esta metodología ágil, es una de las más utilizadas dentro de la elaboración de distintos proyectos en el desarrollo de software, principalmente por su fácil adaptación a proyectos que están constantemente expuestos a cambios, además de ser adaptable, es flexible, rápida y efectiva.

Lo que caracteriza a esta metodología es que trabaja con sprints, los cuales son bloques de tiempo en los que el equipo trabaja en el desarrollo del proyecto. Esta metodología se basa en un desarrollo incremental, iterando los sprints de manera

que cada vez que se completa el ciclo este se va acercando más al resultado esperado, agregando y ajustando las características que sean necesarias para mejorar el proyecto.

2.11.2 Fases

La metodología consta de 5 fases para el desarrollo de proyectos.

Inicio: en esta primera fase se crea una visión del proyecto, determinando y analizando los requerimientos y las necesidades básicas que debe cumplir el proyecto.

Planificación y estimación: en esta etapa se estiman periodos de tiempo limitados para cumplir metas, estableciendo fechas e identificando tareas para cada miembro del equipo.

Implementación: una vez determinadas las tareas se lleva a cabo la ejecución de las actividades, además de la creación de entregables del sprint. En esta etapa se realizan reuniones diariamente con la finalidad de que los miembros del equipo tengan una actualización acerca del progreso del proyecto. También se realiza una priorización de actividades acumuladas.

Revisión y retrospectiva: se revisan los entregables y la funcionalidad del proyecto. Los integrantes del equipo hablan acerca de los aciertos que tuvieron dentro del desarrollo del proyecto, limitantes e impedimentos, así como también en qué pueden mejorar para el siguiente producto.

Lanzamiento: en esta última etapa de la metodología se entrega el producto final y los entregables al cliente.

CAPÍTULO III

Desarrollo y metodología.

3.1 Etapa de requerimiento y análisis.

3.1.1 Usuarios

3.1.1.1 Administrador.

Es el usuario con mayores responsabilidades dentro de la aplicación web, ya que se encarga de agregar y editar los usuarios (mecánicos y supervisores), así como también de editar y validar las líneas de producción

3.1.1.2 Supervisor.

Este usuario tiene como finalizar identificar las fallas dentro de las líneas de producción y crear solicitudes a tiempo para los mecánicos.

3.1.1.3 Mecánico

Este usuario se encarga de realizar las reparaciones, su única función es hacer las reparaciones para las solicitudes realizadas por los supervisores.

3.2 Requisitos específicos.

Número de requisito	RF001
Nombre de requisito	Inicio de sesión
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Todos los actores.
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RF002
Nombre de requisito	Gestión de usuarios
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Encargado de mantenimiento (administrador)
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RF003
Nombre de requisito	Gestionar bitácoras.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Encargado de mantenimiento (administrador)
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RF004
Nombre de requisito	Cambiar responsable de cada bitácora.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Encargado de mantenimiento (administrador)
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input checked="" type="checkbox"/> Baja/ Opcional

Número de requisito	RF005
Nombre de requisito	Crear solicitud de mantenimiento.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Supervisor.
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RF006
Nombre de requisito	Mostrar el historial de fallas.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Encargado de mantenimiento (administrador)
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RF007
Nombre de requisito	Mostrar y responder solicitudes de mantenimiento.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Mecánico.
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RF008
Nombre de requisito	Descargar reportes.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Encargado de mantenimiento (administrador)
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RF009
Nombre de requisito	La aplicación debe funcionar sin conexión a internet (LAN)
Tipo	<input type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Fuente del requisito	Encargado de mantenimiento (administrador)
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

3.3 Casos de uso

CU-01	Inicio de sesión.	
Descripción	El inicio de sesión es la parte fundamental del sistema que se encargará de autenticar a los usuarios y de identificar al usuario por medio de sus roles.	
Precondición	Para poder iniciar sesión, el usuario deberá estar dado de alta en la BD.	
	Paso	Acción
	1	El usuario debe acceder a la aplicación web mediante el URL que le fue proporcionado.
	2	El sistema solicita tanto el correo electrónico como la contraseña.
	3	El usuario ingresa tanto el correo electrónico como la contraseña.
	4	El usuario presiona el botón de inicio de sesión.
	5	El sistema valida las credenciales y deja acceder al usuario al sistema.
Postcondición	El usuario solo podrá acceder al módulo o módulos que le corresponden.	
Excepciones	Paso	Acción
	2	En caso de olvidar la contraseña se deberá contactar con el administrador.
	3	Si el correo o la contraseña son erróneos, la aplicación indicara un mensaje de error.
	5	Si el usuario ha sido eliminado de la base de datos el usuario no podrá iniciar sesión.
Rendimiento	La aplicación web deberá realizar la operación dependiendo de la velocidad de la conexión a internet, y las características del equipo de cómputo.	

Frecuencia	Cada vez que se requiera.
Urgencia	Inmediatamente.
Comentarios	Usuario se refiere a alguno de los roles descritos en la aplicación.
Diagrama UML	<pre> graph LR Usuario[Usuario] --- Iniciar[Iniciar sesion] Iniciar -.-> «include» Validar[Validar correo y contraseña] </pre>

CU-02	Gestión de usuarios. Agregar nuevo usuario.	
Descripción	Permite al administrador agregar nuevos supervisores y mecánicos al sistema.	
Precondición	El administrador debe de haber iniciado sesión.	
	Paso	Acción
	1	El administrador debe acceder al módulo de gestión de usuarios.
	2	El administrador presiona el botón para agregar usuario.
	3	El sistema le solicita datos para agregar un usuario.
	4	El sistema valida los datos y agrega el usuario nuevo al sistema.
Postcondición	El usuario solo podrá acceder al módulo o módulos que le corresponden.	
Excepciones	Paso	Acción
	3	El sistema no permite correos duplicados. En caso de ingresar un correo duplicado se le notificará al administrador.

Rendimiento	La aplicación debe de poder realizar la acción sin problemas y la velocidad dependerá de la velocidad a la conexión a internet del dispositivo con el que se acceda.
Frecuencia	Cada vez que se requiera.
Importancia	Vital.
Urgencia	Inmediatamente.
Comentarios	Para agregar un nuevo usuario se necesita colocar el nombre, una contraseña, un correo, y la línea de producción a la que pertenece.
Diagrama UML	<pre> graph LR Admin[Administrador] --- Crear((Crear usuario)) Crear -.-> «include» Rol((Seleccionar rol)) Crear -.-> «include» Datos((Validar datos)) </pre> <p>The diagram shows an actor named 'Administrador' connected to a use case 'Crear usuario'. This use case includes two other use cases: 'Seleccionar rol' and 'Validar datos'.</p>

CU-03	Gestión de usuarios. Modificar usuario.	
Descripción	Permite al administrador modificar los usuarios del sistema.	
Precondición	El administrador requiere de haber iniciado sesión con anterioridad. El usuario que se desea modificar debe estar registrado en la BD.	
	Paso	Acción
	1	El administrador requiere acceder al módulo de gestión de usuarios.
	2	El administrador busca el usuario y presiona el botón para editar usuario.
	3	El sistema le solicita los datos para editar el usuario.
	4	El administrador edita los datos necesarios, y posteriormente presionara el botón de actualizar.

	5	La aplicación valida los datos y actualiza el usuario.
Postcondición	El administrador no puede cambiar el rol de los usuarios una vez creados.	
Excepciones	Paso	Acción
	4	El sistema no permite correos duplicados.
Rendimiento	La aplicación debe de poder realizar la acción sin problemas.	
Frecuencia	Cada vez que se requiera.	
Importancia	Vital.	
Urgencia	Inmediatamente.	
Diagrama UML	<pre> graph LR Admin[Administrador] --- Mod[Modificar usuario] Mod -.-> «include» Valid[Validar datos] </pre>	

CU-04	Gestión de usuarios. Listar usuarios.	
Descripción	Permite al administrador listar todos los usuarios agregados al sistema.	
Precondición	El administrador requiere de iniciar sesión con anterioridad.	
	Paso	Acción
	1	El administrador accede al módulo de usuarios.
Postcondición		
Excepciones	Paso	Acción
	1	Si no hay usuarios disponibles se mostrara una tabla vacía.
Rendimiento	La aplicación web deberá realizar la acción descrita dependiendo de la velocidad a internet y las características del equipo de cómputo.	

Frecuencia	Cada vez que se requiera.
Importancia	Media.
Urgencia	Inmediatamente.
Comentarios	Se listan los usuarios de todas las líneas de producción que existen en el sistema.
Diagrama UML	<pre> graph LR Admin[Administrador] --- UC((Mostrar usuarios)) </pre>

CU-05	Gestión de usuarios. Eliminar usuario.	
Descripción	Permite al administrador eliminar usuarios del sistema.	
Precondición	El usuario deberá estar registrado en la BD del sistema.	
	Paso	Acción
	1	El administrador ingresa al módulo de usuarios.
	2	El administrador selecciona el usuario que requiere borrar y presiona el botón de borrar.
	3	La aplicación le muestra un mensaje de confirmación si realmente desea borrar el usuario.
Postcondición	El usuario no podrá acceder de nuevo a la aplicación.	
Excepciones	Paso	Acción
	3	En caso de seleccionar cancelar no se realizará ninguna acción.
Rendimiento	1 minuto estimado para la autenticación.	
Frecuencia	Cada vez que se requiera.	
Importancia	Vital.	
Urgencia	Inmediatamente.	

Comentarios	Es necesario destacar que el usuario no se borrará realmente de la base de datos, lo que sucederá es que no se mostrará nuevamente.
Diagrama UML	<pre> graph LR Admin[Administrador] --- UC1(Validar existencia en la BD) Admin --- UC2(Eliminar usuario) UC2 -.-> UC3(Confirmar eliminación del usuario) </pre> <p>The diagram shows an actor named 'Administrador' connected to two use cases: 'Validar existencia en la BD' and 'Eliminar usuario'. A dashed arrow points from 'Eliminar usuario' to 'Confirmar eliminación del usuario'.</p>

CU-06	Gestión de bitácoras. Mostrar bitácoras.	
Descripción	Permite al administrador y al supervisor mostrar las bitácoras existentes dentro del sistema.	
Precondición	Deberá haber bitácoras creadas por el administrador.	
	Paso	Acción
	1	El administrador o supervisor acceden al módulo de bitácoras.
Postcondición	Al supervisor solo se le mostrarán las bitácoras para que pueda hacer una solicitud de mantenimiento, y no tendrá la posibilidad de editarlas o eliminarlas.	
	Paso	Acción
	1	Si no hay bitácoras agregadas por el administrador se mostrara una tabla vacía.
Rendimiento	La aplicación web deberá realizar la acción descrita dependiendo de la velocidad de internet y las características del equipo de cómputo.	

Frecuencia	Cada vez que se requiera.
Importancia	Media
Urgencia	Inmediatamente.
Comentarios	No se mostrarán las bitácoras que previamente se eliminaron del sistema.
Diagrama UML	<pre> graph TD Admin[Administrador] --- UC((Mostrar bitácoras)) Supervisor[Supervisor] --- UC </pre> <p>The diagram is a Use Case Diagram on a grid background. It features two actor icons: one labeled 'Administrador' at the top left and one labeled 'Supervisor' at the bottom left. A central use case is represented by an oval labeled 'Mostrar bitácoras'. Two lines connect the actor icons to the use case oval, indicating that both roles can interact with this function.</p>

CU-07	Gestión de bitácoras. Agregar bitácoras.	
Descripción	Permite al administrador agregar nuevas bitácoras al sistema.	
Precondición	El administrador debe iniciar sesión dentro de la aplicación.	
	Paso	Acción
	1	El administrador ingresa al módulo de gestión de bitácoras.
	2	El administrador da clic al botón de agregar nueva bitácora.
	3	El administrador le da un nombre a la bitácora.
	4	Seleccionar mecánico responsable.

	5	Seleccionar la línea de producción a la cual va a pertenecer la bitácora.
Postcondición	Solo puede existir un mecánico responsable por cada bitácora.	
Excepciones	Paso	Acción
	2	No se podrá crear una bitácora nueva si no hay mecánicos disponibles.
Rendimiento	2 minutos esperados para la creación de la bitácora.	
Frecuencia	Cada vez que se requiera.	
Importancia	Vital.	
Urgencia	Inmediatamente.	
Comentarios	El nombre de la bitácora es completamente opcional.	
Diagrama UML	<pre> graph LR Admin[Administrador] --- Agregar((Agregar bitacora)) Agregar -.-> «include» Mecanico((Seleccionar mecanico)) Agregar -.-> «include» Linea((Seleccionar linea de produccion)) </pre> <p>The diagram shows an actor named 'Administrador' connected to a use case 'Agregar bitacora'. This use case includes two other use cases: 'Seleccionar mecanico' and 'Seleccionar linea de produccion'.</p>	

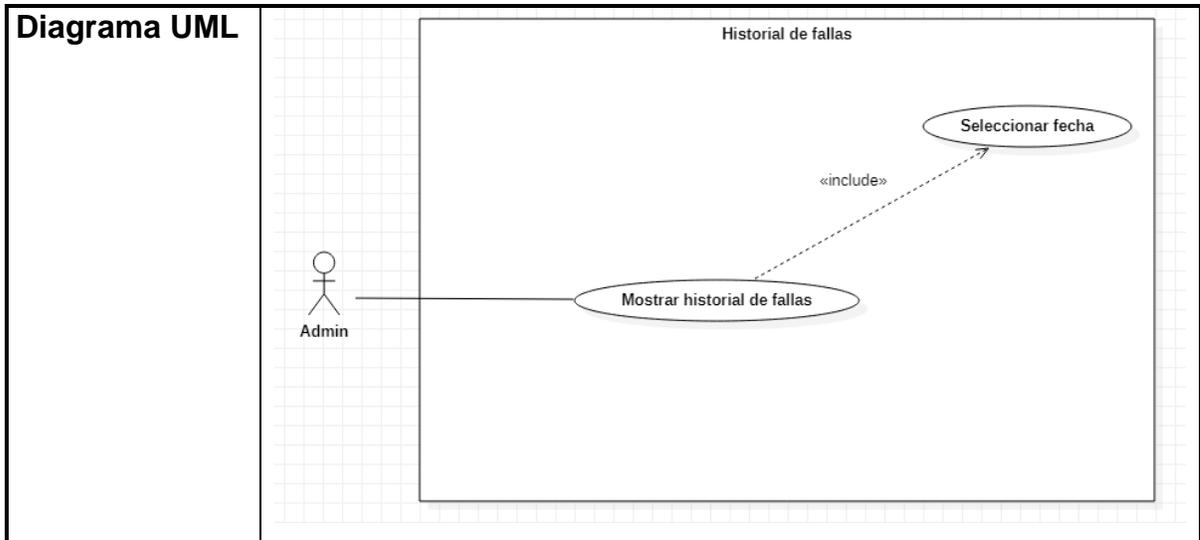
CU-08	Gestión de bitácoras. Cambiar mecánico responsable de la bitácora.	
Descripción	Permite al administrador cambiar el responsable de cada bitácora.	
Precondición	Deben existir mecánicos creados dentro del sistema. La bitácora debe de existir.	
	Paso	Acción
	1	El administrador accede al módulo de bitácoras.
	2	Selecciona la bitácora que se desea editar.
	3	Selecciona el nuevo mecánico responsable.

Postcondición	Solo puede haber un mecánico responsable por cada bitácora.	
Excepciones	Paso	Acción
	2	Al seleccionar un mecánico que es responsable de otra bitácora se quitará como responsable de esa bitácora.
Rendimiento	1 minutos esperados para la creación de la bitácora.	
Frecuencia	Cada vez que se requiera.	
Importancia	Vital.	
Urgencia	Inmediatamente.	
Comentarios		
Diagrama UML	<pre> graph LR Admin[Administrador] --- UC1((Cambiar responsable de la bitacora)) UC1 -.-> «include» UC2((Seleccionar mecanico.)) </pre>	

CU-09	Crear solicitud de mantenimiento.	
Descripción	Permite al supervisor crear una nueva solicitud de mantenimiento para alguna máquina específica.	
Precondición	<p>El supervisor debe conocer la bitácora en la que hará la solicitud. Debe existir la bitácora a la cual se desea crear la solicitud. Debe existir la máquina dentro de la base de datos. El supervisor debe conocer información esencial acerca de la máquina: Tipo, número o identificador, línea de producción.</p>	
	Paso	Acción
	1	El supervisor detecta una falla en la línea de producción.
	2	El supervisor accede al módulo de bitácoras en el sistema.
	3	El supervisor identifica la bitácora en la que creará una solicitud y da clic al botón de agregar solicitud.

	4	El supervisor selecciona la bitácora correspondiente e ingresa los datos correspondientes.
Postcondición	Se le mostrará una nueva solicitud al mecánico.	
Excepciones	Paso	Acción
Rendimiento	El proceso de agregar una nueva solicitud de mantenimiento será tan rápido como la velocidad de internet, y las capacidades de procesamiento físicas del dispositivo lo permita.	
Frecuencia	Cada vez que se requiera.	
Importancia	Vital.	
Urgencia	Inmediatamente.	
Comentarios	Datos en la solicitud: <ul style="list-style-type: none"> • Nombre de la bitácora • Identificador de la máquina, modelo, tipo. • Fecha y hora que se creó la solicitud • Supervisor que creo la solicitud (se agrega automáticamente). • Prioridad de reparación • Detalles adicionales 	
Diagrama UML	<pre> graph LR subgraph Sistema direction TB U1(Enviar solicitud al mecanico) U2(Almacenar en la base de datos) U3(Crear solicitud) U3 -.-> «include» U2 U3 -.-> «extend» U1 end S((Supervisor)) --- U3 </pre>	

CU-10	Mostrar el historial de fallas.	
Descripción	Permite al administrador descargar el historial de fallas que se han registrado dentro del sistema.	
Precondición	Se han registrado fallas dentro de la aplicación.	
	Paso	Acción
	1	El administrador ingresa al inicio.
	2	El administrador selecciona el rango de fechas en las que desea descargar el historial de fallas.
	3	El administrador selecciona la línea de producción.
Postcondición		
Excepciones	Paso	Acción
Rendimiento	La velocidad de generación del archivo donde se encuentra el historial de fallas dependerá completamente de las prestaciones donde se encuentre el servidor de la app.	
Frecuencia	Cada vez que se requiera.	
Importancia	Baja.	
Urgencia	Cuando se requiera.	
Comentarios		



CU-11	Mostrar y atender nuevas solicitudes de mantenimiento.	
Descripción	Permite al mecánico mostrar las solicitudes de mantenimiento creadas para la línea a la que pertenece y atenderlas.	
Precondición	El mecánico inicio sesión. Debe existir una solicitud de mantenimiento.	
	Paso	Acción
	1	El mecánico ingresa al apartado de solicitudes.
	2	El mecánico identifica la solicitud que atenderá.
	3	El mecánico da clic a la solicitud.
	4	Al terminar la reparación el mecánico rellena los campos correspondientes.
Postcondición		
Excepciones	Paso	Acción
Rendimiento	Tan rápido como el mecánico termine su reparación.	
Frecuencia	Cada vez que se requiera.	
Importancia	Vital.	

Urgencia	Inmediatamente.
Comentarios	El sistema tomará como la fecha de reparación la hora en la que el mecánico haya llenado los campos.
Diagrama UML	<pre> graph TD subgraph Solicitudes UC1(Actualizar hora de llegada) UC2(Mostrar las nuevas solicitudes) UC3(Marcar solicitud como reparada) UC1 -.-> «extend» UC2 end Supervisor((Supervisor)) --- UC1 Mecanico((Mecanico)) --- UC2 Mecanico --- UC3 </pre>

3.2 Etapa de diseño

3.2.1 Diagramas de secuencia.

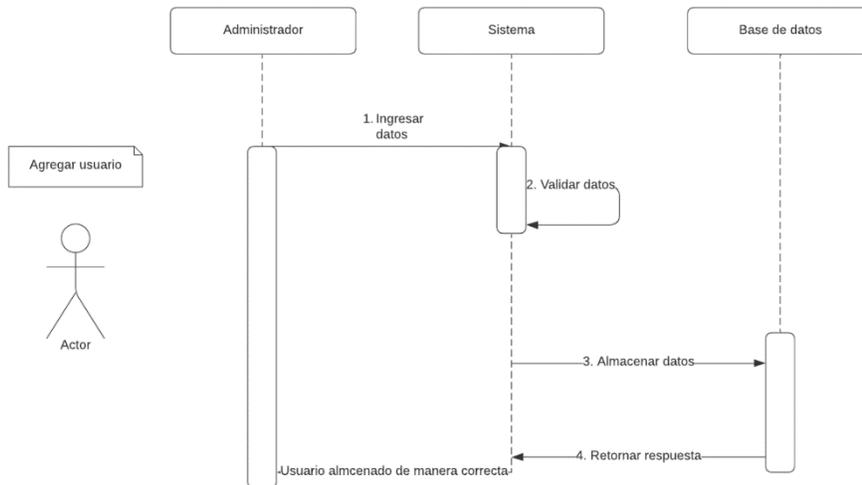


Ilustración 1 Diagrama de secuencia para agregar un nuevo usuario.

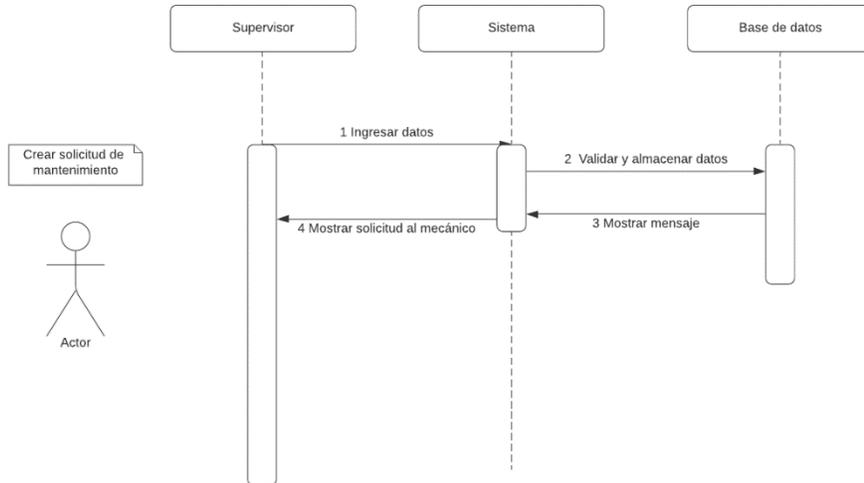


Ilustración 2 Diagrama de secuencia para crear solicitud de mantenimiento.

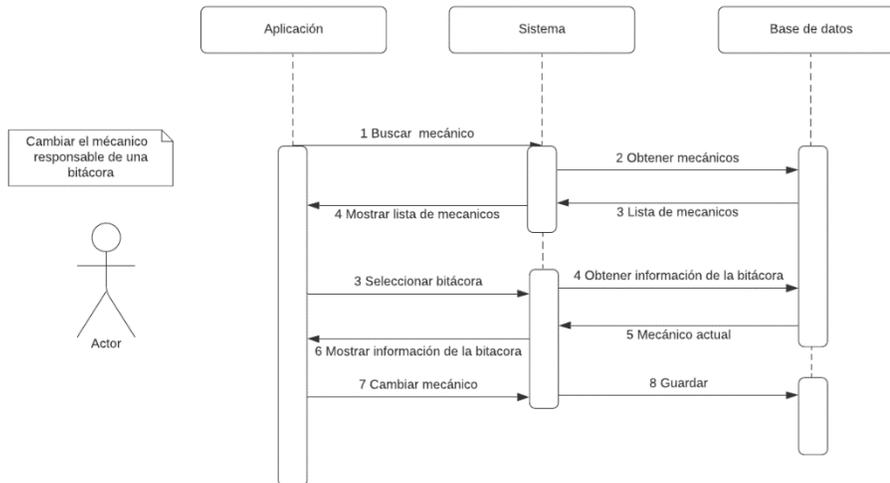


Ilustración 3 Diagrama de secuencia cambiar el mecánico responsable de una bitácora.

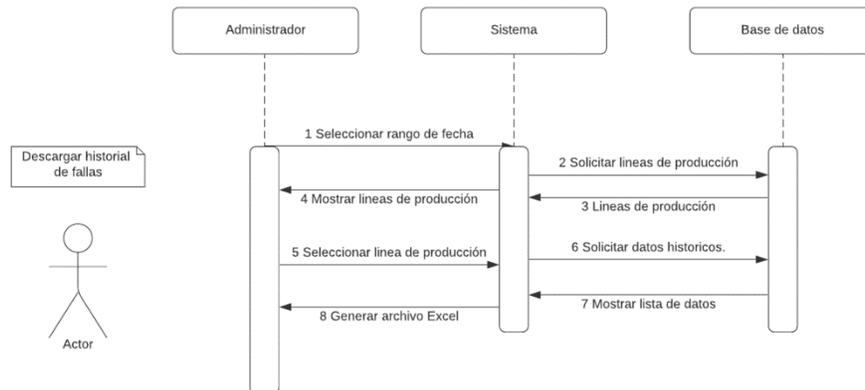


Ilustración 4 Diagrama de secuencia de descarga de historial de fallas.

3.2.2 Prototipo del proyecto.

3.2.2.1 Inicio.

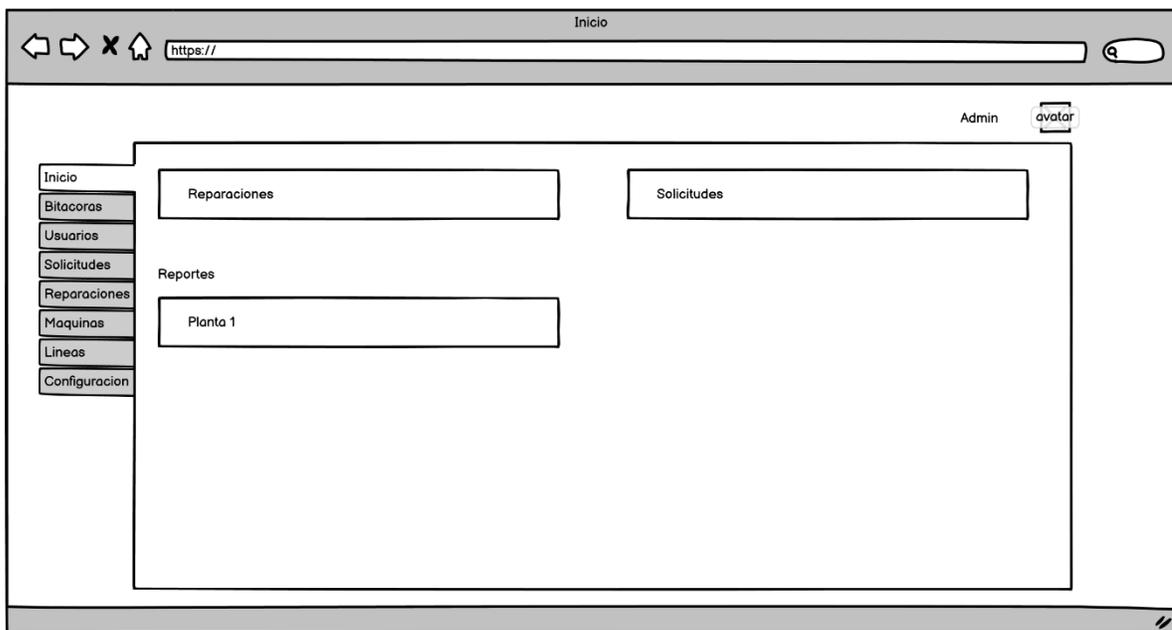


Ilustración 5 Wireframe de la vista del inicio de la aplicación.

3.2.2.2 Bitácoras

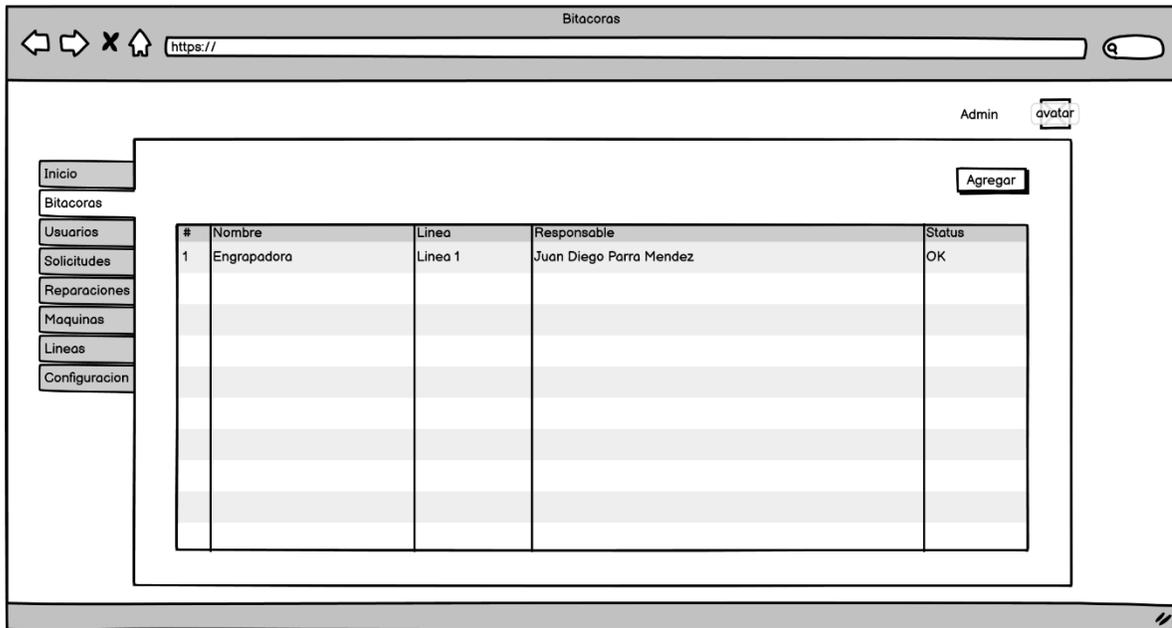


Ilustración 6 Wireframe de la vista de bitácoras.

3.2.2.3 Usuarios

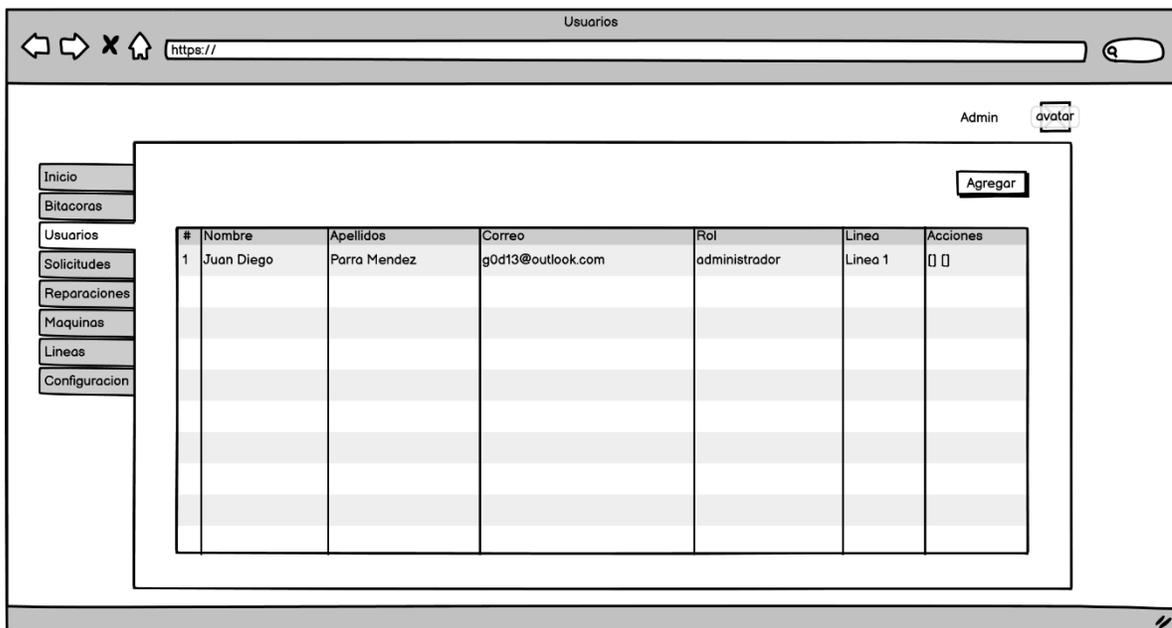


Ilustración 7 Wireframe de la vista de usuarios.

3.2.2.5 Solicitudes

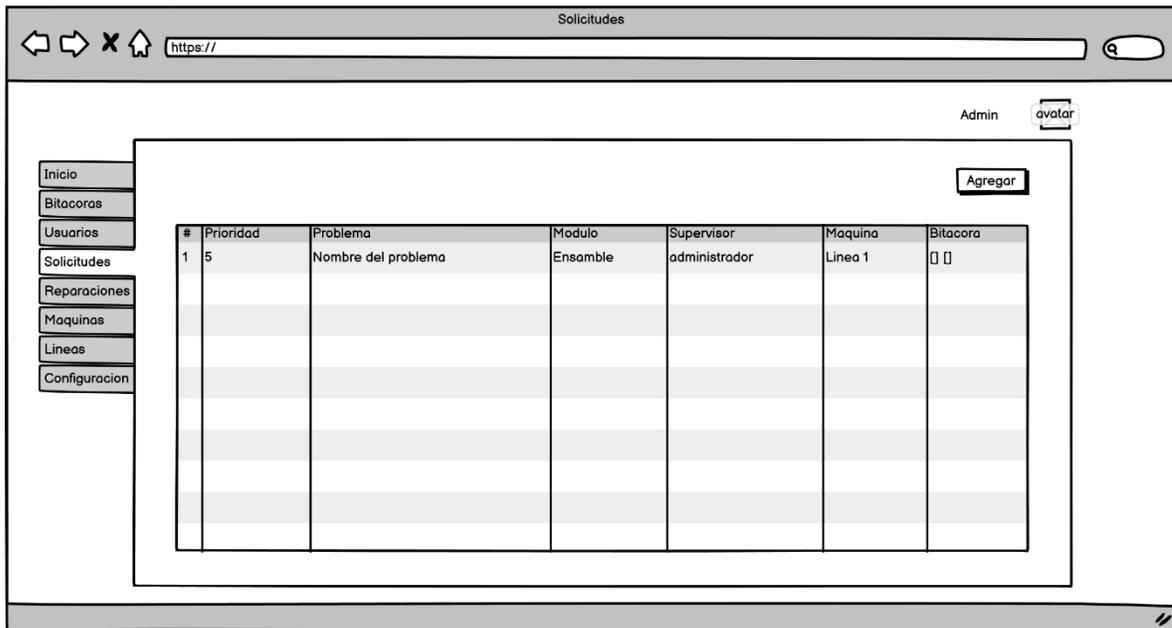


Ilustración 8 Wireframe de la vista de solicitudes.

3.2.2.6 Reparaciones

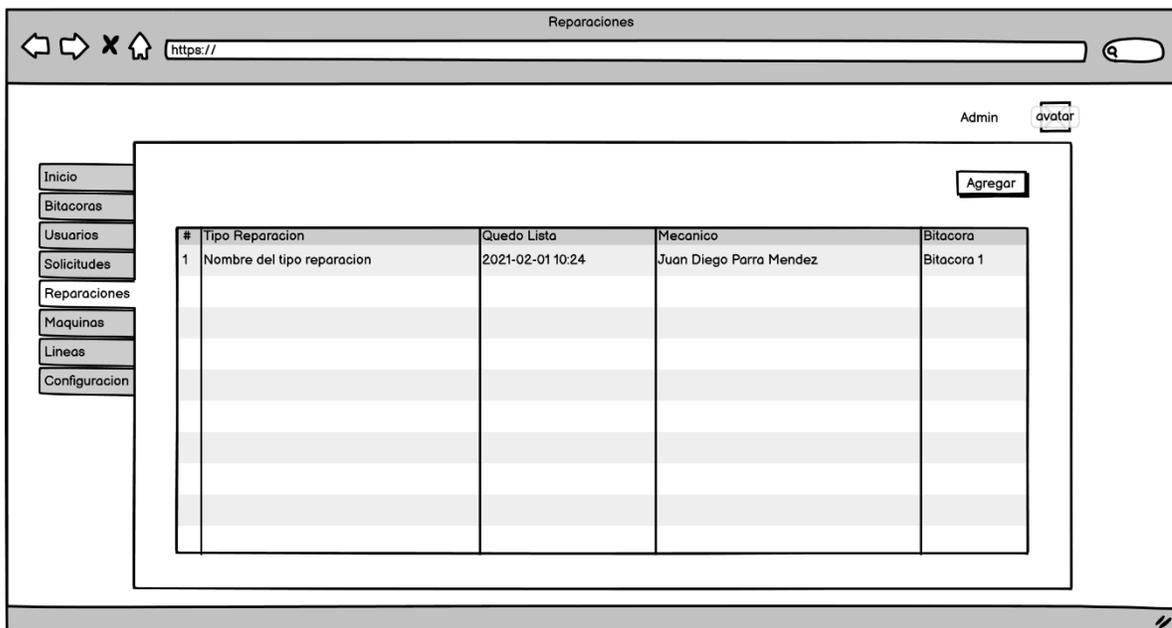


Ilustración 9 Wireframe de la vista de reparaciones.

3.2.2.7 Maquinas

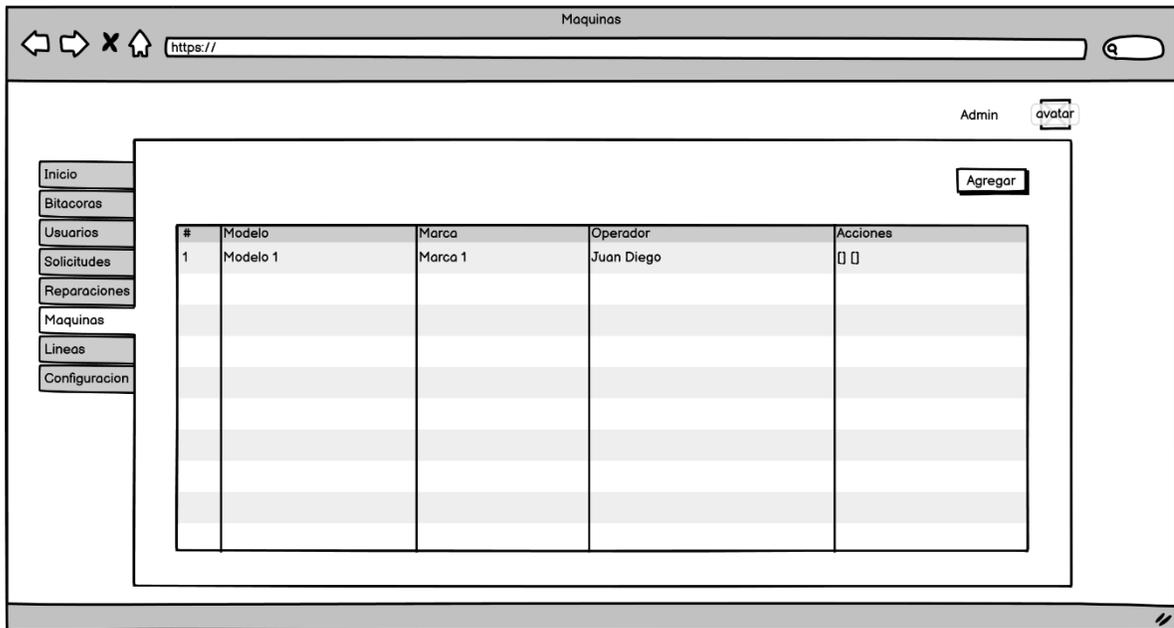


Ilustración 10 Wireframe de la vista de las máquinas.

3.2.2.8 Líneas

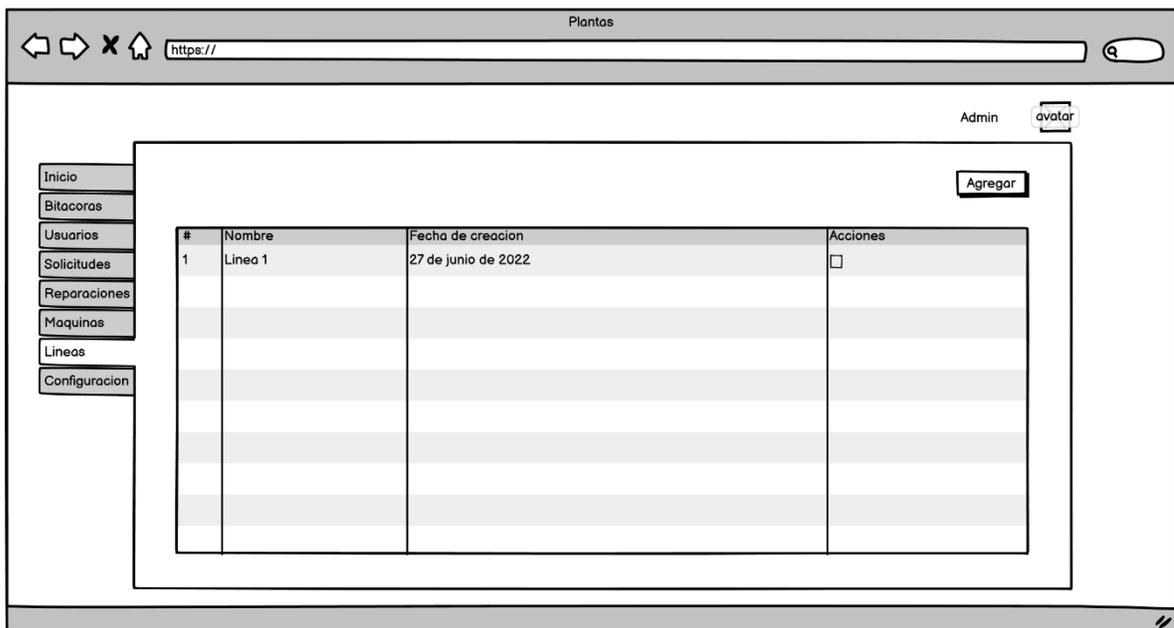


Ilustración 11 Wireframe de la vista de líneas de producción.

3.2.2.9 Configuraciones

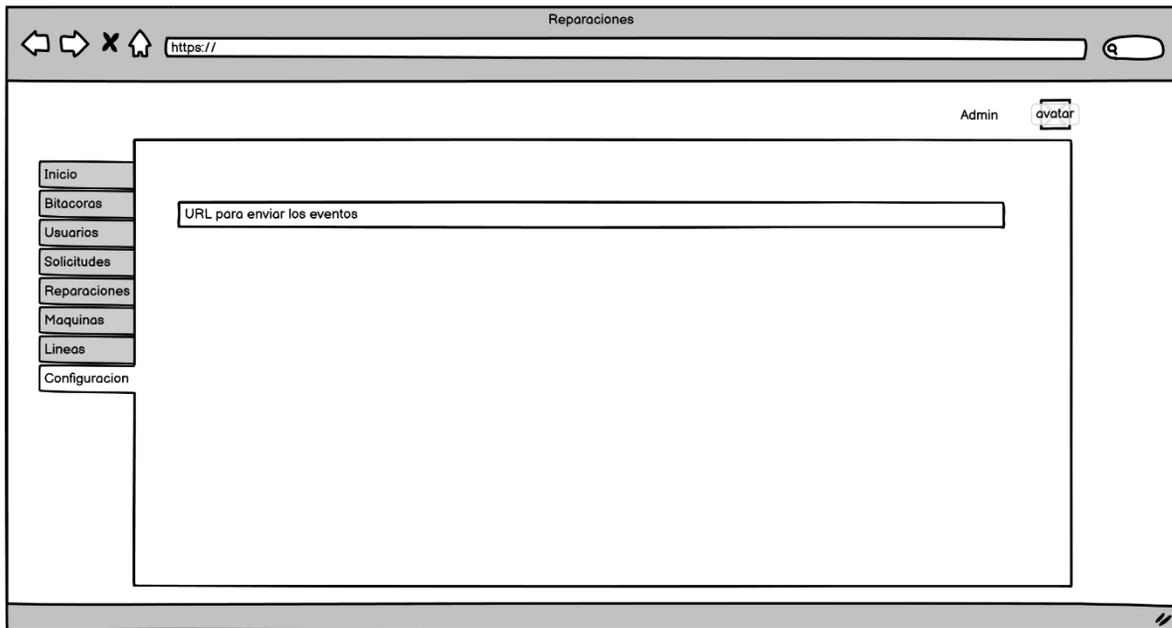


Ilustración 12 Wireframe de la configuración del sistema.

Metodología

En el presente capítulo, se explica a detalle el proceso que se siguió para el desarrollo del proyecto, así como también las diversas facetas de desarrollo.

El trabajo utiliza una investigación aplicada de tipo tecnológico, donde se buscará encontrar una solución enfocada a mejorar el control de mantenimiento correctivo en la empresa CONFETEX S. A de C. V. Este trabajo inicia con la búsqueda de información relacionada con el tema abordado, con la finalidad de otorgar una solución óptima, acorde a las solicitudes y necesidades de la empresa mediante el uso de tecnología de software innovador. Al final de este trabajo y con el objetivo de comprobar la eficiencia de la solución desarrollada, se presenta un análisis de resultados con lo cual se podrá verificar la hipótesis planteada en este estudio.

El establecimiento de las actividades se relaciona una con la otra, con la finalidad de llegar a un resultado conciso, dichas actividades descritas.

Alcances

Analizar de forma detallada las necesidades y requerimientos del área de mantenimiento, así identificar problemáticas significativas para la solución mediante la implementación de un software que apoye en el área de la empresa Confecciones Unión S. A. de C. V. Teziutlán, Puebla.

Identificación de necesidades y análisis de requisitos

Atraer beneficios con la funcionalidad del software y lograr la optimización del proceso de mantenimiento.

Analizar los resultados obtenidos y prevenir posibles problemas que se puedan encadenar para el rendimiento de la forma de trabajo.

Limitaciones

La solución propuesta estará acorde solamente a las necesidades actuales por el área de mantenimiento.

Debido a la cantidad de máquinas se tendrá que tomar para el análisis correspondiente pocas máquinas para obtener el estudio deseado.

La aplicación y diseño del mismo proyecto se podrá realizar durante el tiempo en el que se desarrolle el proyecto, con la información que se recabe.

Las decisiones de cambios, modificaciones e implementaciones quedan a cargo por parte de los mismos beneficiarios.

3.4 Hipótesis

La implementación de un software que genere notificaciones de mantenimiento correctivo mejorará el control de servicios de reparación de las máquinas ubicadas en las líneas de producción.

3.5 Selección de la muestra

Para realizar estudio debido al tipo de actividad de la empresa se tomará la línea de producción de una de las 3 líneas de producción de la empresa, conformada por 3 plantas, con un total de 50 máquinas.

En relación con la muestra se utilizará la fórmula para definir muestras de poblaciones finitas:

$$n = \frac{N \times Z_{\alpha}^2 \times p \times q}{d^2 \times (N - 1) + Z_{\alpha}^2 \times p \times q}$$

En esta fórmula se especifica que "N" sería el tamaño de la población, la variable "Z" hace referencia al nivel de confianza, la denotación "p" alude a la probabilidad de éxito o podría ser también la proporción esperada, la variable "q" indica la probabilidad de fracaso, y la notación "d" representa la precisión o podría ser el "Error máximo admisible en términos de proporción".

Después de lo cálculos y por la cantidad de población se observa que población= muestra.

3.6 Recolección de información

En esta primera fase se dio a la tarea de investigar información relevante en cuanto al proceso de mantenimiento correctivo que maneja la empresa CONFETEX S. A. de C. V.

En la recolección de datos se realizó un análisis de requisitos con la finalidad de identificar datos e información relevante del proceso; partiendo de la revisión de formatos utilizados para el registro y control de mantenimiento, las generalidades quedaron definidas de acuerdo con los escenarios en los cuales se aplica el proceso de mantenimiento.

Pasos que deberá realizar el personal del departamento de producción para la notificación de fallas de las máquinas de coser.

1. El operario de la máquina afectada se dirige con el encargado del módulo de producción y le notifica la falla de su máquina.
2. El encargado del módulo a partir de la notificación de falla de la máquina reporta al área de mantenimiento mediante el registro en la bitácora correspondiente a la planta; los datos que corresponden a la máquina reportada mediante los campos: número de planta, operación (selección a partir de un catálogo de operaciones), número de máquina (selección a partir de un catálogo de número económico de maquinaria).
3. En el caso de los campos que deberá solicitar el responsable del módulo del área de producción, cuando se trate del campo de planta se deberá elegir entre las 4 con las que cuenta la empresa, en el caso del campo operación se deberá seleccionar el tipo de operación del catálogo desplegado en la aplicación web, la cual indique la operación realizada por el operario en la máquina afectada, por último se deberá seleccionar en el campo de número de máquina correspondiente al número económico de la máquina a reparar,

dato localizado en la parte superior de la misma, finalmente deberá registrar la hora y fecha de la solicitud.

Pasos que deberá realizar el personal del departamento de mantenimiento para solventar notificación de fallas de las máquinas de coser.

1. En el momento de registrar la falla en la bitácora de la planta, se mostrará la nueva solicitud agregada al final de la lista; se indicará la máquina que presenta la falla, hora y fecha.
2. El mecánico revisa la bitácora y marca que la solicitud será atendida.
3. El mecánico se dirige al área donde se ubica la máquina descompuesta, y reporta su llegada.
4. El encargado del módulo registra la hora el formato de atención.
5. El mecánico inicia el diagnóstico de la falla de la máquina, en función al listado de códigos de problemas, e identifica el mismo.
6. El mecánico informa de forma inmediata al departamento de mantenimiento la falla que presenta la máquina y procede a la reparación.
7. El mecánico repara y registra la hora en que termina de repararla y anota observaciones adicionales en el formato correspondiente.

3.6.1 Análisis de formatos

Para el control y solicitud de mantenimiento correctivo existe un formato de solicitud y otro de registro de mantenimiento, donde se solicitan los datos de la máquina a la cual se va a evaluar, este formato se presenta en físico para su llenado individual.

RUTINA DIARIA DE MANTENIMIENTO CORRECTIVO **MANTENIMIENTO**

Planta: Duque Urc Fecha: 08-02-2014

No. Prioridad	No. Prioridad	Operacion	No. de Máquina	Modulo	Código del problema	Llamó al Mecánico	Llegó al Mecánico	Quedó Lista	Tipo de Reparación	Nombre del Mecánico
P 2	P 2	Ensamblado	SN	Ensamblar	06	8:23	8:30	8:50	Ajustar que sigue	
P 7	P 7	Ensamblar	460	Trazado	07	8:44	9:00	9:50	Pulir y ajustar guías y otros en obra	
P 7	P 7	Lustrado	475	Trazado	07	9:42	10:26	10:35	Pulir y ajustar guías	
P 1	P 1	Armar	85	Ensamblar	07	11:00	11:10	11:40	Ajustar que sigue	
P 2	P 2	Ensamblado	SN	Ensamblar	06	12:12	12:15	12:20	Ajustar tornillo de presión	
P 3	P 3	Armar	391	Ensamblar	07	1:00	1:02	1:10	Ajustar que sigue	
P 2	P 2	Problemas	902	Trazado	07	1:50	2:00	2:15	Ajustar guías y guías	

Ilustración 13. Formato de solicitud y registro de mantenimiento

CONFECCIONES DE TEZIUTLAN, S.A. DE C.V.
PROGRAMA ANUAL DE MANTENIMIENTO PREVENTIVO

MAQUINA: NO. MECANICO RESPONSABLE DE AREA:

ACTIVIDAD	DESARROLLO	FECHA	ESTADO	OBSERVACIONES
MANTENIMIENTO 1: PORTACORRERES	REVISAR COMPLETAMENTE LOS PORTACORRERES Y SI ESTAN DEBIDAMENTE OLEADOS.	13 ENE	✓	
MANTENIMIENTO 1: GUÍAS	REVISAR TODOS LOS TIPOS DE GUÍAS Y SI ESTAN DEBIDAMENTE OLEADAS.	13 FEB	✓	
MANTENIMIENTO 1: ENGRANAJES	EL ENGRANAJE DE LA IMPORTANTE VA QUE 1 CON BAJOS REVENTES ES POR ESO QUE D EL ENGRANAJE M.	13 FEB	✓	
MANTENIMIENTO 1: AGUJA	VERIFICAR QUE LA AGUJA SEA LA CORRECTA Y SI NO LO ES, CAMBIARLA POR LA CORRECTA.	13 FEB	✓	
MANTENIMIENTO 1: PLACA DE AGUJA	LA PLACA DE LA AGUJA DEBE DE ESTAR BIEN AJUSTADA Y SI NO LO ESTÁ, AJUSTARLA.	13 FEB	✓	
MANTENIMIENTO 1: DIENTES	LOS DIENTES DEBEN DE ESTAR BIEN AJUSTADOS Y SI NO LO ESTÁN, AJUSTARLOS.	13 FEB	✓	
MANTENIMIENTO 1: PRENSETAS	LOS PRENSETAS DEBEN DE ESTAR BIEN AJUSTADOS Y SI NO LO ESTÁN, AJUSTARLOS.	13 FEB	✓	
MANTENIMIENTO 1: GARRIF	EL GARRIF DEBE DE ESTAR BIEN AJUSTADO Y SI NO LO ESTÁ, AJUSTARLO.	13 FEB	✓	
MANTENIMIENTO 1: CONDICIONES DE LOS CARBETES	SON MUY IMPORTANTES EN LA MAQUINA, DEBE DE VERIFICAR EL ESTADO DE LOS CARBETES Y SI NO ESTAN BIEN, CAMBIARLOS.	13 FEB	✓	
MANTENIMIENTO 1: GUARDIA AGUJA	DEBE DE VERIFICAR QUE LA GUARDIA AGUJA ESTE BIEN AJUSTADA Y SI NO LO ESTÁ, AJUSTARLA.	13 FEB	✓	
MANTENIMIENTO 2: CAMBIO DE ACEITE	EL ACEITE DEBE DE ESTAR BIEN AJUSTADO Y SI NO LO ESTÁ, CAMBIARLO.	13 FEB	✓	
MANTENIMIENTO 3: LUBRICACION Y TANGUOS	DEBE DE VERIFICAR LA LUBRICACION Y TANGUOS Y SI NO ESTAN BIEN, CAMBIARLOS.	13 FEB	✓	
MANTENIMIENTO 3: CORREAS	DEBE DE VERIFICAR EL ESTADO DE LAS CORREAS Y SI NO ESTAN BIEN, CAMBIARLAS.	13 FEB	✓	
MANTENIMIENTO 4: MOTORES DE ELECTRICIDAD	DEBE DE VERIFICAR EL ESTADO DE LOS MOTORES DE ELECTRICIDAD Y SI NO ESTAN BIEN, CAMBIARLOS.	13 FEB	✓	

Ilustración 14 Programa anual de mantenimiento preventivo

3.7 Aplicación de metodología SCRUM

3.7.1 Inicio

Aplicando las fases de la metodología SCRUM, en esta primera fase se determinaron los requerimientos de la aplicación, los roles de los usuarios, los módulos de la aplicación, se identificaron las necesidades básicas que la aplicación tenía que cumplir, entre otros aspectos.

3.7.2 Planificación y estimación

En esta etapa se establecieron metas determinadas para poder cumplir con las tareas y actividades decretadas por el asesor del proyecto, en ésta fase se realizó el prototipo de la aplicación, determinando la distribución de los módulos, las funcionalidades y características con base en los requerimientos específicos del proyecto.

3.7.3 Implementación

En esta etapa se llevó a cabo la ejecución de las actividades planteadas en la fase anterior.

CAPÍTULO IV

Resultados

Con la finalización del software de aplicación y su posterior uso en ámbito real dentro del área de producción y de mantenimiento se obtuvo lo siguiente:

4.1 Presentación del software, descripción de interfaces

Para verificar la correcta funcionalidad del software fue necesario implementar la aplicación, posteriormente se realizó una exhaustiva validación de la correcta salida de las diferentes actividades agregadas al software mediante un profundo análisis de requisitos. Después de la implementación y pruebas del sistema, se pronosticó obtener una buena optimización de tiempos de atención entre cada solicitud, así como un buen control del registro de mantenimientos; información que posteriormente será utilizada para generar reportes. Para poder verificar las funcionalidades del sistema fue necesario el registro de usuarios en el sistema bajo los diferentes roles que intervienen en el proceso de mantenimiento.

A continuación, se muestran las interfaces de la aplicación web terminada.

4.1.1 Pantalla de inicio

Aquí se muestra la cantidad de solicitudes que se realizaron el día actual, y el número de reparaciones que se lograron completar.

Esta sección también permite descargar los reportes de las solicitudes realizadas. Para ello es necesario seleccionar el rango en el que se desea descargar la información y la línea de producción. El sistema generara automáticamente un Excel que se descargara en el equipo.

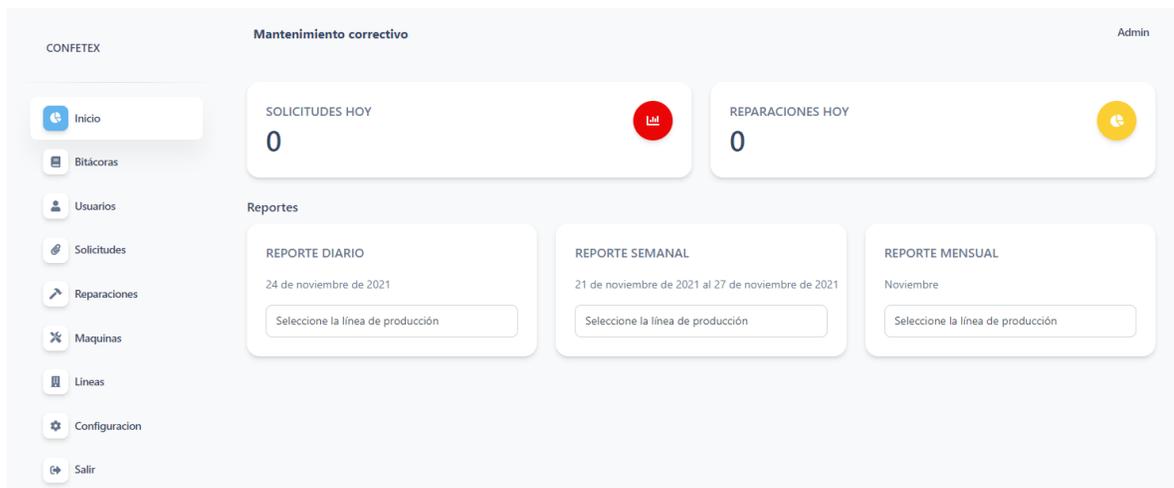


Ilustración 15 Ventana principal

4.1.2 Bitácoras

En la pantalla de bitácoras se muestran las bitácoras creadas por el administrador, en caso de iniciar sesión con otro usuario con el rol de supervisor, solo se le mostrará el botón de crear solicitud.

La columna "status" es donde se muestra que estado tiene esa solicitud: Si hay solicitudes por atender, si se han atendido todas, o si está en camino el mecánico para su reparación.

El administrador podrá, en esta sección, crear nuevas bitácoras o editar las que ya existen, así como cambiar el mecánico responsable de la bitácora.

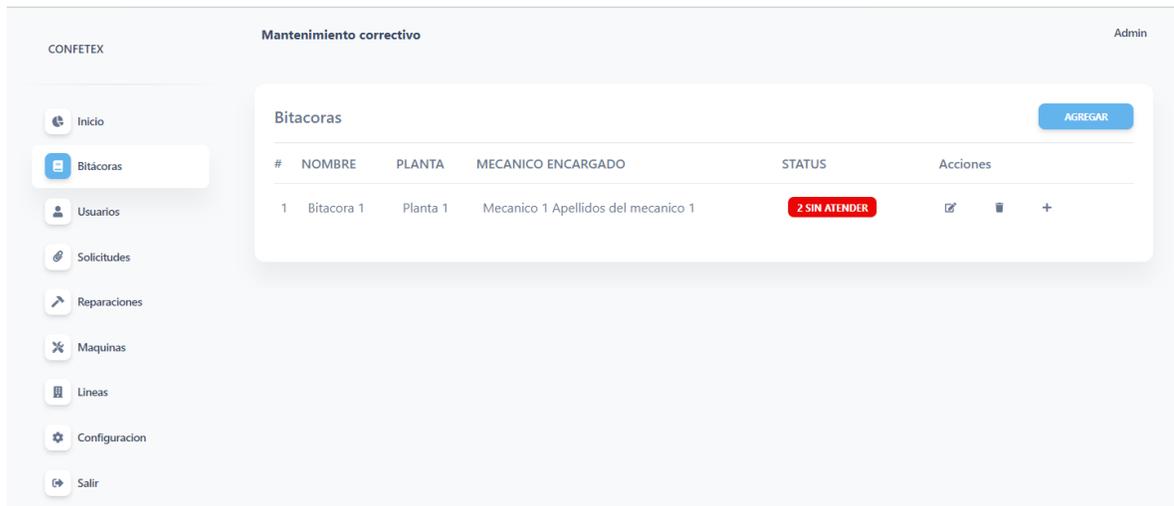


Ilustración 16 Vista de bitácoras creadas en el sistema.

4.1.2 Usuarios.

Aquí se muestran los usuarios agregados al sistema por el administrador, cabe destacar que en esta vista sólo se puede acceder por el administrador.

Esta sección permite al administrador ver los usuarios agregados, editarlos, o eliminarlos del sistema. Es necesario destacar que cuando un usuario se elimina del sistema, solo se eliminara de esta vista y no en la BD. Todo ello para mantener los datos consistentes y tener un mejor historial a través del tiempo.

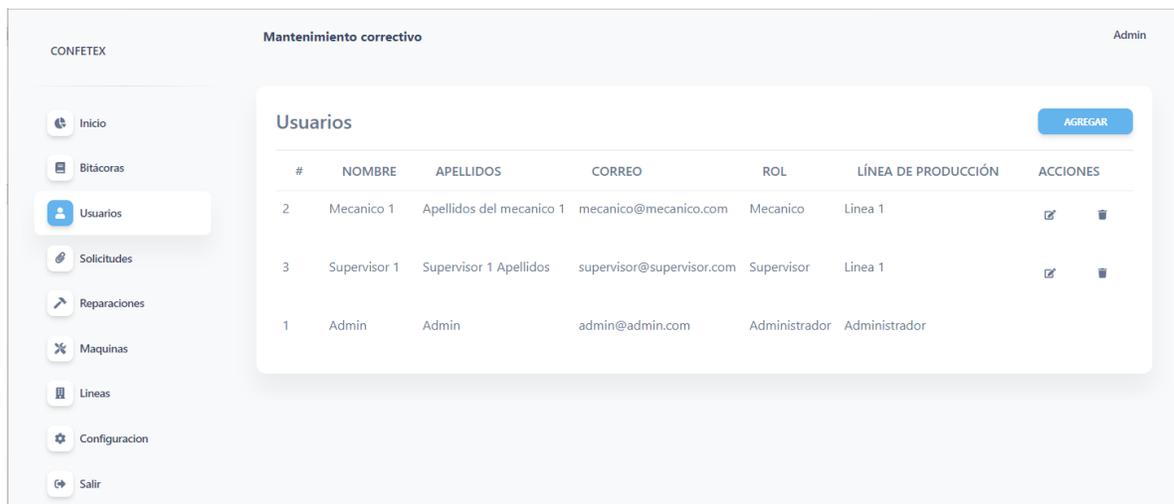
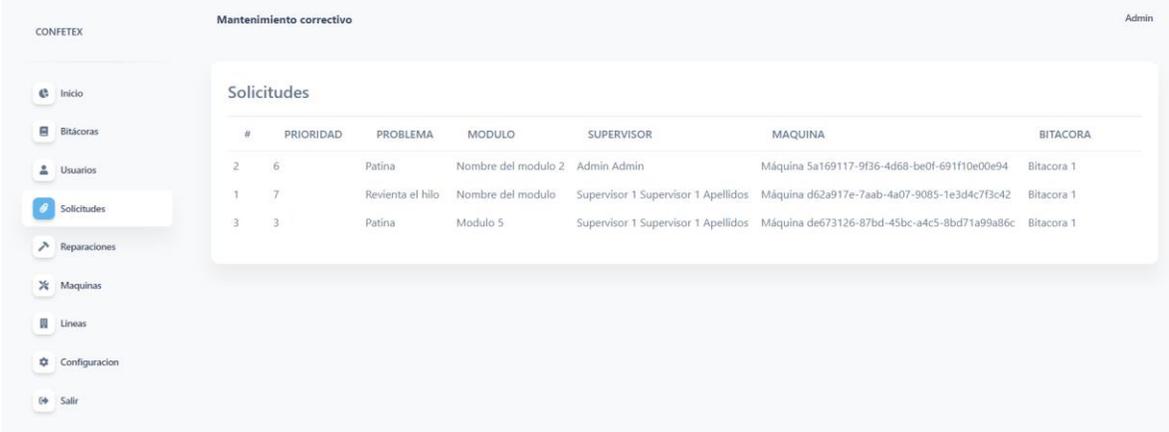


Ilustración 17 Vista de la lista de usuarios agregados al sistema.

4.1.2 Solicitudes

En esta sección se mostrarán todas las solicitudes realizadas por el supervisor en el sistema, en esta sección es donde el supervisor se encarga de marcar la hora de llegada del mecánico, para contabilizar el tiempo que tardo en resolverla.

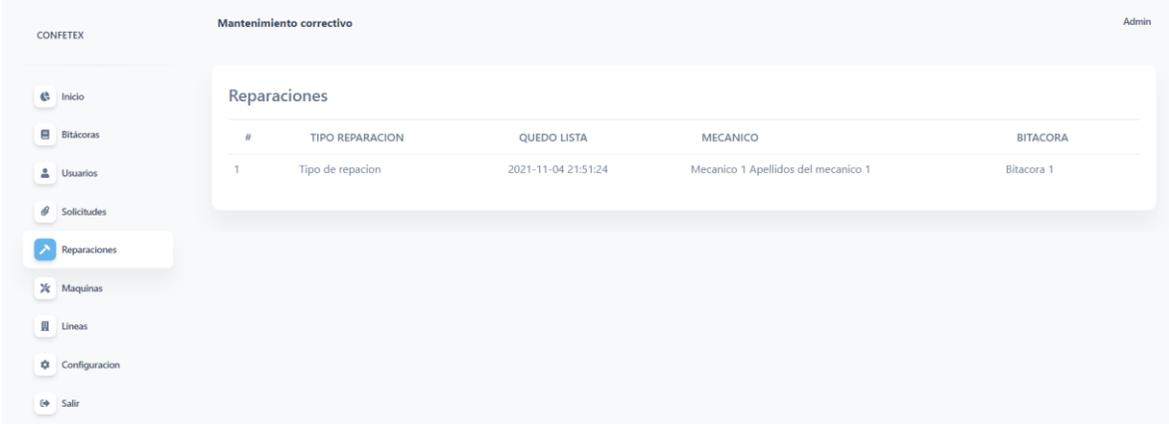


#	PRIORIDAD	PROBLEMA	MODULO	SUPERVISOR	MAQUINA	BITACORA
2	6	Patina	Nombre del modulo 2	Admin Admin	Máquina 5a169117-9f36-4d68-be0f-691f10e00e94	Bitacora 1
1	7	Revienta el hilo	Nombre del modulo	Supervisor 1 Supervisor 1 Apellidos	Máquina d62a917e-7aab-4a07-9085-1e3d4c7f3c42	Bitacora 1
3	3	Patina	Modulo 5	Supervisor 1 Supervisor 1 Apellidos	Máquina de673126-87bd-45bc-a4c5-8bd71a99a86c	Bitacora 1

Ilustración 18 Vista de solicitudes almacenadas en el sistema.

4.1.3 Reparaciones

Aquí se muestran las reparaciones realizadas por los mecánicos. Esta sección es puramente informativa y sirve para ver el historial a través del tiempo. Las fechas almacenadas en esta sección son guardadas automáticamente.



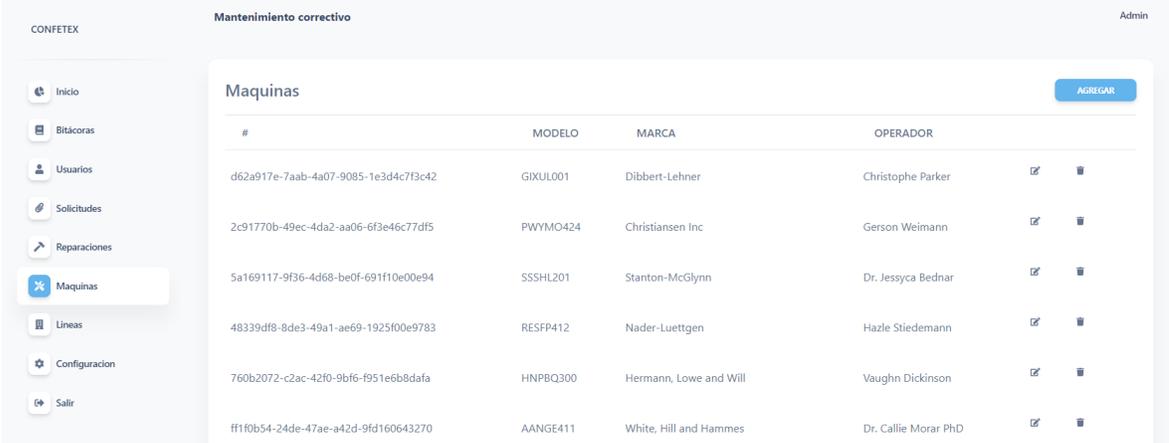
#	TIPO REPARACION	QUEDO LISTA	MECANICO	BITACORA
1	Tipo de repacion	2021-11-04 21:51:24	Mecanico 1 Apellidos del mecanico 1	Bitacora 1

Ilustración 19 Vista de reparaciones almacenadas en el sistema.

4.1.4 Máquinas

En ésta sección se muestran las máquinas agregadas por el administrador, el agregar las máquinas permite llevar un registro de fallos que ha tenido esa máquina en un periodo de tiempo específico.

También permite llevar a cabo de manera más precisa el proceso de creación de una nueva solicitud de mantenimiento, y la identificación de la máquina.



The screenshot shows a web application interface for 'Mantenimiento correctivo'. On the left is a sidebar with navigation options: Inicio, Bitácoras, Usuarios, Solicitudes, Reparaciones, Máquinas (highlighted), Líneas, Configuración, and Salir. The main content area is titled 'Máquinas' and contains a table with columns for ID, Model, Brand, Operator, and actions. A blue 'AGREGAR' button is in the top right of the table area. The user 'Admin' is logged in.

#	MODELO	MARCA	OPERADOR		
d62a917e-7aab-4a07-9085-1e3d4c7f3c42	GIXUL001	Dibbert-Lehner	Christophe Parker	✍	🗑
2c91770b-49ec-4da2-aa06-6f3e46c77df5	PWYMO424	Christiansen Inc	Gerson Weimann	✍	🗑
5a169117-9f36-4d68-be0f-691f10e00e94	SSSHL201	Stanton-McGlynn	Dr. Jessyca Bednar	✍	🗑
48339df8-8de3-49a1-ae69-1925f00e9783	RESFP412	Nader-Luettgen	Hazle Stiedemann	✍	🗑
760b2072-c2ac-42f0-9bf6-f951e6b8dafa	HNPBQ300	Hermann, Lowe and Will	Vaughn Dickinson	✍	🗑
ff1f0b54-24de-47ae-a42d-9fd160643270	AANGE411	White, Hill and Hammes	Dr. Callie Morar PhD	✍	🗑

Ilustración 20 Vista de máquinas almacenadas en el sistema.

4.1.5 Líneas de producción

Esta sección permite tener el control de las líneas de producción dentro del sistema, permite enlazar nuevas bitácoras y usuarios a una línea de producción específica para mantener los datos con más control y más separados.

#	NOMBRE	FECHA CREACION
1	Línea de producción 1	29 de octubre de 2021
2	Línea de producción 2	29 de octubre de 2021
3	Línea de producción 3	29 de octubre de 2021
4	Línea de producción 4	29 de octubre de 2021

Ilustración 21 Tabla con las líneas de producción almacenadas en el sistema.

4.1.6 Pantalla de configuración

La pantalla de configuración permite colocar la URL a donde se estarán emitiendo los eventos:

- Cuando se crea una nueva solicitud
- Cuando un mecánico está en camino.
- Cuando una bitácora se encuentra en perfecto estado.

Ilustración 22 Configuración

4.2 Desarrollo de la metodología de investigación

- 1) Implementación del software en la sección de mantenimiento en la compañía Confección CONFETEX S. A. de C. V.
- 2) Ejecutar la aplicación durante 2 semanas en las 50 máquinas de la línea 1.
- 3) Observación de funcionalidad del software.
- 4) Análisis de reportes de control de mantenimiento generados por la aplicación.
- 5) Verificación de resultados e hipótesis.

Con respecto a la mejora de los tiempos del proceso de mantenimiento correctivo, se puso a prueba el sistema en un lapso de 10 días hábiles, obtenido una serie de datos de los cuales se presentan las tablas de los tiempos promedios generados de las actividades siguientes: registro de la solicitud ejecutada por el supervisor de la línea, tiempo de revisión de las solicitudes que realiza el mecánico, tiempo del llenado del formato de registro de atención realizado por el mecánico, tiempo de elaboración de reportes, realizado por el personal de apoyo del área de mantenimiento de la compañía. Dicha información se muestra en la "Tabla 1".

Tabla 1. Tiempos de registro con el uso del sistema de información.

Días	Tiempo registro solicitud	Tiempo Revisión Solicitudes de Reparación	Tiempo Registro Atención	Tiempo Elaboración Reportes
1	3	1	1	1
2	3	1	1	1
3	3	1	0.5	1
4	2.5	1	0.5	1
5	3	1	1	1
6	3	1	1	1
7	2.5	1	0.5	1
8	2.5	1	0.5	1
9	2.5	1	0.5	1
10	2.5	1	0.5	1
PROMEDIOS	2.7	1	0.7	1

Para poder comparar la eficiencia del sistema se utilizaron los registros realizados en la línea 1, quince días antes de realizar la implementación del sistema, tomando para ello un total de 10 días en periodos de lunes a viernes.

Tabla 2. Tiempos de registro antes del uso del sistema de información.

Dias	Tiempo registro solicitud	Tiempo Revisión Solicitudes de Reparación	Tiempo Registro Atención	Tiempo Elaboración Reportes
1	5	2	2	360
2	6	2	2	350
3	5	1.5	1.7	345
4	5	2	1.8	360
5	6	2	2	366
6	6	1.5	2	370
7	5	1.5	2.5	349
8	5	1.2	2	367
9	5.5	1	1.8	389
10	5	2	2.5	375
PROMEDIOS	5.4	1.6	2	363.4

Comparando ambas tablas podemos observar que el tiempo de registro de solicitud, la revisión de solicitudes de reparación, el registro de atención y el tiempo de la elaboración de reportes que se muestra en la tabla 2 disminuyó favorablemente una vez que se implementó la aplicación, mostrando los resultados que se encuentran en la "Tabla 1", concluyendo así que, el sistema de mejora ayudó a la optimización y ahorro de tiempo en las actividades requeridas.

CAPÍTULO V

CONCLUSIONES

5.1 Conclusiones del proyecto, recomendaciones y experiencia profesional adquirida

El desarrollo de la aplicación Web permitió comprender la importancia de un sistema que administre las reparaciones de manera correcta, con un orden específico. Para que gracias a la información el personal responsable de la aplicación pueda tomar decisiones basadas en datos, además permite entender la importancia que tienen los datos para las empresas.

5.2 Conclusiones relativas a los objetivos específicos

El proceso necesario para realizar una solicitud de mantenimiento fue reducido al máximo, evitando que haya errores humanos al olvidar colocar datos, o colocándolos mal. También se asegura que haya consistencia en los datos almacenados para llevar de manera más precisa el historial de las fallas y reparaciones.

5.3 Conclusiones relativas al objetivo general

Con la aplicación Web en marcha, CONFETEX puede mejorar enormemente la velocidad de respuesta de alguna máquina, cuando sucede una falla, de esta manera puede reducir el costo que viene al tener una máquina fuera de producción. Puede llevar un historial de manera fácil y sencilla sobre las máquinas que han fallado dentro del sistema, y puede comprar y/o cambiar alguna máquina cuando se considere pertinente.

5.4 Limitaciones del modelo planteado

El modelo planteado solo se encarga del control de notificaciones generadas para mantenimiento correctivo, dejando de lado otros tipos de mantenimiento como el preventivo o el predictivo.

La aplicación Web no puede gestionar la cantidad de refacciones con la que se cuenta dentro de las instalaciones de CONFETEX, tampoco es capaz de crear recordatorios para compras nuevas de refacciones.

5.5 Recomendaciones

Con los datos obtenidos por el sistema y después tiempo de uso continuo de la aplicación, es posible aplicar operaciones sobre los datos para realizar mantenimiento predictivo con las máquinas registradas dentro del sistema, gracias a estas operaciones nos podemos hacer preguntas:

- ¿Cuál será la vida media de una máquina en una línea de producción concreta y con x cantidad de operadores?
- ¿De qué manera influye las fallas con la rotación de operadores?

CAPÍTULO VI
COMPETENCIAS
DESARROLLADAS

6.1 Competencias desarrolladas y/o aplicadas

6.2 Competencias genéricas

1. Flexibilidad y capacidad para adaptarme al cambio.
2. Habilidad para aprender nuevas tecnologías.
3. Motivación de aprender y seguir formándome.
4. Tolerancia a la frustración
5. Capacidad para tomar decisiones.
6. Mejorar mi capacidad para resolver problemas.

6.3 Competencias específicas

1. Análisis y diseño en el manejo de base de datos relacionales
2. Capacidad de análisis para discernir y crear los casos de uso.
3. Manejo del framework de Laravel con PHP.
4. Uso de CSS y frameworks como Bootstrap.
5. Uso de control de versiones con Git.

CAPITULO VII

FUENTES DE INFORMACIÓN

7.1 Referencias

- Aner. (2020, 29 septiembre). ¿Qué es el mantenimiento correctivo? Recuperado 9 de octubre de 2021, de <https://www.aner.com/blog/mantenimiento-correctivo.html>
- Arsys. (2020, 22 septiembre). Cuando, para qué y por qué utilizar MariaDB. Blog de arsys.es. Recuperado 21 de octubre de 2021, de <https://www.arsys.es/blog/programacion/mariadb/>
- Author, G. (2021, 12 febrero). Bootstrap: guía para principiantes de qué es, por qué y cómo usarlo. Rock Content - ES. Recuperado 21 de octubre de 2021, de <https://rockcontent.com/es/blog/bootstrap/>
- B., G. (2021a, abril 29). ¿Qué es Apache? Descripción completa del servidor web Apache. Tutoriales Hostinger. Recuperado 21 de octubre de 2021, de <https://www.hostinger.mx/tutoriales/que-es-apache/>
- B., G. (2021b, agosto 9). ¿Qué es HTML? Explicación de los fundamentos del Lenguaje de marcado de hipertexto. Tutoriales Hostinger. Recuperado 18 de noviembre de 2021, de <https://www.hostinger.es/tutoriales/que-es-html>
- BYRON, R., & CASTRILLÓN, D. (2019). DISEÑO DEL PLAN DE MANTENIMIENTO PREVENTIVO BASADO EN LA FILOSOFÍA RCM. http://bibliotecadigital.udea.edu.co/bitstream/10495/13758/1/GallegoRicardo_2019_MantenimientoEquiposConfecciones.pdf
- de Souza, I. (2021, 12 febrero). Descubre qué es el lenguaje de programación PHP y en qué situaciones se hace útil. Rock Content - ES. Recuperado 18 de noviembre de 2021, de <https://rockcontent.com/es/blog/php/>

de Zúñiga, F. G. (2021a, junio 15). ¿Qué es Laravel? Blog de arsys.es. Recuperado 21 de octubre de 2021, de <https://www.arsys.es/blog/programacion/que-es-laravel/>

GCFglobal. (s. f.). Informática Básica: ¿Qué son las aplicaciones web? GCFGlobal.org. Recuperado 18 de noviembre de 2021, de <https://edu.gcfglobal.org/es/informatica-basica/que-son-las-aplicaciones-web/1/>

Hernández, M. (2020, 14 enero). Maquila poblana eleva producción anual. El Economista. <https://www.eleconomista.com.mx/estados/Maquila-poblana-eleva-produccion-anual-20200114-0020.html>

Herranz, S. (2018, 13 marzo). Wireframe, Mockup y Prototipos: en busca de sus diferencias. Medium. Recuperado 25 de octubre de 2021, de <https://medium.com/rocket-studio-ux/wireframe-mockup-y-prototipos-en-busca-de-sus-diferencias-23a03bcdb69>

Lucidchart. (s. f.-a). Qué es el lenguaje unificado de modelado (UML). Recuperado 21 de octubre de 2021, de <https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml>

Lucidchart. (s. f.-b). Qué es un diagrama entidad-relación. Recuperado 18 de noviembre de 2021, de <https://www.lucidchart.com/pages/es/que-es-un-diagrama-entidad-relacion>

Lucidchart. (s. f.-c). Tutorial de diagrama de secuencia UML. Recuperado 18 de noviembre de 2021, de <https://www.lucidchart.com/pages/es/diagrama-de-secuencia>

Mozilla. (2021a, octubre 20). CSS | MDN. Developer Mozilla. Recuperado 21 de octubre de 2021, de <https://developer.mozilla.org/es/docs/Web/CSS>

- Mozilla. (2021b, octubre 20). HTML: Lenguaje de etiquetas de hipertexto | MDN. [Https://Developer.Mozilla.Org](https://Developer.Mozilla.Org).
<https://developer.mozilla.org/es/docs/Web/HTML>
- Mozilla. (2021c, octubre 20). JavaScript | MDN. Developer Mozilla. Recuperado 21 de octubre de 2021, de <https://developer.mozilla.org/es/docs/Web/JavaScript>
- Mozilla. (2021d, octubre 20). SQL - Glosario | MDN. Developer Mozilla. Recuperado 21 de octubre de 2021, de <https://developer.mozilla.org/es/docs/Glossary/SQL>
- Php.net. (s. f.). PHP: ¿Qué es PHP? - Manual. php. Recuperado 21 de octubre de 2021, de <https://www.php.net/manual/es/intro-what-is.php>
- Pickett, P. (2020, 21 Julio). What Is SQL? The Balance Careers. Recuperado 18 de noviembre de 2021, de <https://www.thebalancecareers.com/what-is-sql-and-uses-2071909>
- Plantillas mediante Blade - Laravel 5. (s. f.). gitbook. Recuperado 21 de octubre de 2021, de https://ajgallego.gitbook.io/laravel-5/capitulo_1/capitulo_1_plantillas
- StarAgile. (2020, 23 octubre). Overview of Scrum Phases | StarAgile. <https://staragile.com/blog/scrum-phases/>
- Román Zamitiz, C. A. (s. f.). ¿Qué es UML? UNAM. Recuperado 18 de noviembre de 2021, de <http://profesores.fi-b.unam.mx/carlos/aydoo/uml.html>
- Rubio, J. C. (2020, 30 julio). Qué es GIT y para qué sirve. OpenWebinars.net. Recuperado 21 de octubre de 2021, de <https://openwebinars.net/blog/que-es-git-y-para-que-sirve/>

Sagnier, C. (2021, 27 agosto). ¿Cuales son los diferentes tipos de mantenimiento? Mobility Work. Recuperado 17 de noviembre de 2021, de <https://mobility-work.com/es/blog/tipos-de-mantenimiento-estrategia>

Talva, M. (2021, 2 noviembre). Los 5 niveles de mantenimiento. Mobility Work. Recuperado 18 de noviembre de 2021, de <https://mobility-work.com/es/blog/niveles-mantenimiento-correctivo-preventivo/>

Textilnca. (s. f.). Cómo debe ser el mantenimiento correctivo de su máquina de confección. Recuperado 9 de octubre de 2021, de <https://textilnca.com/como-debe-ser-el-mantenimiento-correctivo-de-su-maquina-de-confeccion/>

Vaughan, J. (2018, 11 enero). MariaDB. SearchDataManagement. Recuperado 18 de noviembre de 2021, de <https://searchdatamanagement.techtarget.com/definition/MariaDB>

CAPÍTULO VIII

Anexos

8.1 Diagrama de base de datos

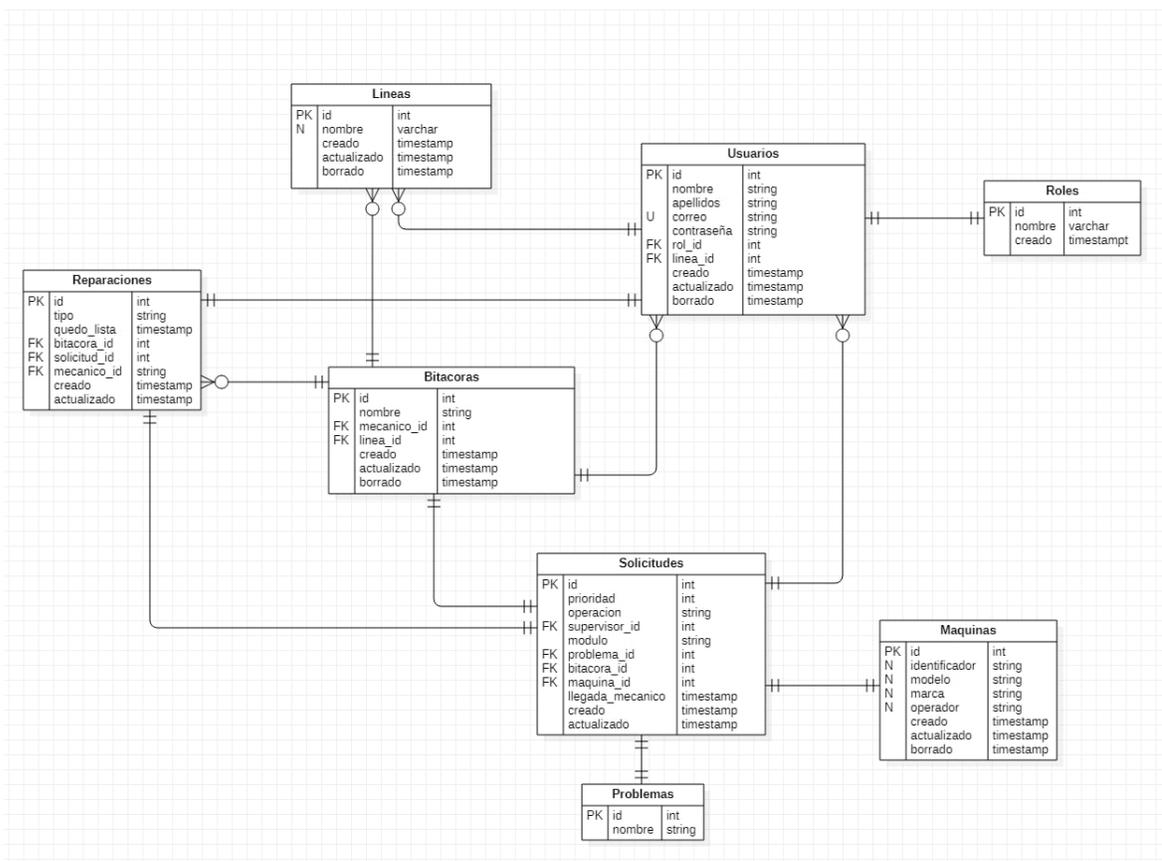


Ilustración 23 Diagrama de la base de datos

8.2 Dictamen de liberación

Asunto: Asignación de Asesor(a), Comisión Revisora y Entrega de Trabajo Profesional y Dictamen	
Teziutlán, Puebla, 28 de julio de 2022	
Asesor(a):	CARREON ROMERO MARIA EUGENIA
Integrante de Comisión Revisora:	GUTIERREZ MARCOS ROMAN
Integrante de Comisión Revisora:	SERRANO HERRERA RICARDO ALBERTO
Presentes	

Por este medio me permito informar que ha sido asignado como asesor(a) y comisión revisora del trabajo profesional que se convertirá en Tesis de:

Alumno (a):	PARRA MENDEZ JUAN DIEGO		
	<small>Apellido paterno/materno/nombre (s)</small>		
Número de Control:	17TE0215	Licenciatura o Posgrado:	INGENIERIA EN SISTEMAS COMPUTACIONALES
Plan:	2010	Correo Electrónico:	17TE0215@teziutlan.tecn.mx

Cuyo tema es: MEJORA DE MANTENIMIENTO CORRECTIVO MEDIANTE NOTIFICACIONES CON SEÑALÉTICA PARA LA EMPRESA CONFETEX S.A. DE C.V

25 palabras (máximo)

Se ha enviado a su correo institucional el trabajo profesional o de grado, por lo cual la comisión revisora tendrá 5 días hábiles para realizar las observaciones al alumnado, el(la) interesado(a) tendrá igualmente 5 días para corregir y las enviará al correo electrónico institucional de la comisión revisora, agradezco de antemano su valioso apoyo en esta importante actividad para la formación profesional de licenciatura o de grado de nuestro alumnado egresado.

Dictamen de Comisión Revisora y Aprobación para Grabación

Siendo el día: **30 de septiembre de 2022** se reunieron los miembros de la comisión para revisar el trabajo asignado y una vez analizado se decidió liberarlo y aprobarlo para su grabación y programación de examen profesional.

 CARREON ROMERO MARIA EUGENIA <small>Nombre y Firma del(la) Asesor(a)</small>	 GUTIERREZ MARCOS ROMAN <small>Nombre y Firma del integrante de la Comisión Revisora</small>
 SERRANO HERRERA RICARDO ALBERTO <small>Nombre y Firma del integrante de la Comisión Revisora</small>	 SÁNCHEZ PÉREZ MYRIAM <small>Subdirección Académica</small>

ccp. Expediente
Rosa G

Folio: R07/05/2021 F-SAC-18

Fracción I y II s/n Aire Libre Teziutlán, Puebla, C.P. 73960 Tels. 231 311 4000 / 4001 / 4002 / 4003
e-mail: itsteziutlan@hotmail.com | tecnm.mx | www.teziutlan.tecnm.mx

PRESENCIA DE LA REVOLUCIÓN MEXICANA

Ilustración 24 Dictamen de liberación

8.3 Carta de autorización

Tecnológico Nacional de México Instituto Tecnológico Superior de Teziutlán		
CARTA DE AUTORIZACIÓN DEL(LA) AUTOR(A) PARA LA CONSULTA Y PUBLICACIÓN ELECTRÓNICA DEL TRABAJO DE INVESTIGACIÓN		
El que suscribe:		
JUAN DIEGO	PARRA	MENDEZ
Con Número de Control	17TE0215	
Pertenece al Programa Educativo	INGENIERÍA EN SISTEMAS COMPUTACIONALES	
Por este conducto me permito informar que he dado mi autorización para la consulta y publicación electrónica del trabajo de investigación en los repositorios académicos.		
Registrado con el producto:	TESIS	
Cuyo Tema es:		
MEJORA DE MANTENIMIENTO CORRECTIVO MEDIANTE NOTIFICACIONES CON SEÑALÉTICA PARA LA EMPRESA CONFETEX S.A. DE C.V		
Correspondiente al periodo:		
AGOSTO-DICIEMBRE 2022		
Y cuyo(a) director(a) de tesis es:		
D.I.E. CARREÓN ROMERO MARÍA EUGENIA		
ATENTAMENTE		
		
JUAN DIEGO PARRA MENDEZ Nombre y firma		
Fecha de emisión: 08/SEPTIEMBRE/2022		
c.c.p. Subdirección Académica		

Ilustración 25 Carta de autorización

8.4 Índice de figuras

Ilustración 1 Diagrama de secuencia para agregar un nuevo usuario.	37
Ilustración 2 Diagrama de secuencia para crear solicitud de mantenimiento.	38
Ilustración 3 Diagrama de secuencia cambiar el mecánico responsable de una bitácora.	38
Ilustración 4 Diagrama de secuencia de descarga de historial de fallas.	39
Ilustración 5 Wireframe de la vista del inicio de la aplicación.	39
Ilustración 6 Wireframe de la vista de bitácoras.	40
Ilustración 7 Wireframe de la vista de usuarios.	40
Ilustración 8 Wireframe de la vista de solicitudes.	41
Ilustración 9 Wireframe de la vista de reparaciones.	41
Ilustración 10 Wireframe de la vista de las máquinas.	42
Ilustración 11 Wireframe de la vista de líneas de producción.	42
Ilustración 12 Wireframe de la configuración del sistema.	43
Ilustración 13. Formato de solicitud y registro de mantenimiento	48
Ilustración 14 Programa anual de mantenimiento preventivo	48
Ilustración 15 Ventana principal	51
Ilustración 16 Vista de bitácoras creadas en el sistema.	52
Ilustración 17 Vista de la lista de usuarios agregados al sistema.	52
Ilustración 18 Vista de solicitudes almacenadas en el sistema.	53
Ilustración 19 Vista de reparaciones almacenadas en el sistema.	53
Ilustración 20 Vista de máquinas almacenadas en el sistema.	54
Ilustración 21 Tabla con las líneas de producción almacenadas en el sistema.	55
Ilustración 22 Configuración.	55
Ilustración 23 Diagrama de la base de datos	64
Ilustración 24 Dictamen de liberación	65
Ilustración 25 Carta de autorización.	66