Proyecto APT

“Gestión Fuego”

Integrantes: Alan Gatica

Javier Ramirez

Carrera: Ingenieria en Informatica

Asignatura: Capstone 300D

Docente: [MARILUZ ALEJANDRA RODRIGUEZ DONOSO](mailto:mar.rodriguezd@profesor.duoc.cl)

**ÍNDICE**

[**Abstract**](#_9mjvo51txagp) **3**

[**Descripción del proyecto**](#_njvsxrqy4uj9) **3**

[**Análisis de Competencias**](#_26jmz2l02y2e) **3**

[**Monetización de la Solución**](#_sn1wlg7vi9zh) **3**

[**Competencias del Perfil de Egreso**](#_v3ts1km7jpow) **4**

[**Objetivos Generales**](#_dy71fnerx8er) **4**

[**Objetivos Específicos**](#_f5rkryhnkh4e) **4**

[**Metodología**](#_eiyxc1dkd66g) **4**

[Levantamiento de requerimientos:](#_ehytw215445y) 5

[Definición de Épicas e Historias de Usuario:](#_5es6wgskbrat) 6

[Sprint 5: semana 8 a 10](#_dqqcruh6yupv) 8

[Sprint 6: semana 10 a 12](#_qhtqwdh46zjn) 8

[**Stack tecnológico**](#_oiqclvgg7uug) **11**

[**Tabla de actividades**](#_1om9h14df113) **11**

[**Carta gantt**](#_xpmo8ad76qdc) **12**

[**Evidencia y justificación**](#_3w3aj16t1h5w) **14**

[**Conclusión**](#_cz3rg3g874ew) **15**

[**Reflexiones individuales**](#_4bf6w6ddvhmw) **16**

[**Bibliografía**](#_byg9ndfz1pmg) **16**

# Abstract

El proyecto tiene como objetivo desarrollar un sistema web que permita al Cuerpo de Bomberos, gestionar el mantenimiento de sus vehículos de emergencia en tiempo real. El sistema registrará el mantenimiento preventivo y correctivo, controlará el inventario de repuestos y gestionará la mano de obra utilizada, optimizando así la disponibilidad operativa.

The project aims to develop a web-based system that allows the Talcahuano Fire Department to manage the maintenance of its emergency vehicles in real-time. The system will record preventive and corrective maintenance, control spare parts inventory, and manage the labor used, thus optimizing the operational availability.

# Descripción del proyecto

El objetivo del proyecto es desarrollar una solución tecnológica web que permita al cuerpo de bomberos de Talcahuano acceder en tiempo real a información sobre el estado de sus vehículos de emergencia. Esto se logrará mediante el diseño y construcción de una web que registre y monitoree el mantenimiento preventivo y correctivo de los vehículos, así como el control de insumos y mano de obra utilizada.

# Análisis de Competencias

### **Fleet Complete**

* **Punto Fuerte:** Ofrece un sistema completo de gestión de flotas con monitoreo en tiempo real y una amplia gama de funcionalidades, incluyendo la gestión de mantenimiento.
* **Ventaja:** Su capacidad de integración con sistemas GPS y otros dispositivos telemáticos lo hace ideal para flotas grandes y dispersas.

### **2. Firehouse Software**

* **Punto Fuerte:** Está diseñado específicamente para departamentos de bomberos, con módulos para la gestión de vehículos, inventario y reportes de mantenimiento.
* **Ventaja:** Su enfoque específico en las necesidades de los cuerpos de bomberos permite una implementación más directa y relevante para las operaciones de emergencia.

### **3. Emergency Reporting**

* **Punto Fuerte:** Ofrece una plataforma completa para la gestión de recursos de emergencia, incluyendo la gestión de flotas y vehículos.
* **Ventaja:** Integración directa con otros sistemas de gestión de emergencias, facilitando la coordinación y el reporte en tiempo real.

# Monetización de la Solución

Nuestra propuesta de monetización incluye un sistema de suscripción escalonado. Los clientes pueden optar por un plan mensual o anual, que incluye soporte técnico y actualizaciones. Este modelo garantiza un flujo constante de ingresos y permite a la compañía de bomberos obtener mejoras continuas en el sistema a un costo predecible.

### **Modelo de Suscripción**

#### **a. Plan Mensual**

costo: 20 mil

**Características:**

* + **Acceso Completo:** Incluye todas las funcionalidades del sistema (gestión de mantenimiento, alertas, inventario, reportes).
  + **Soporte Básico:** Asistencia técnica vía email o chat durante horarios laborales.
  + **Actualizaciones Regulares:** Acceso a las actualizaciones menores del software.

#### **b. Plan Anual**

costo: 180 mil

**Características:**

* + **Descuento:** Ahorro significativo en comparación con el pago mensual.
  + **Acceso Completo:** Mismas funcionalidades que el plan mensual.
  + **Soporte Prioritario:** Soporte técnico con prioridad, con posibilidad de asistencia remota.
  + **Actualizaciones y Mejoras:** Acceso a todas las actualizaciones, incluyendo actualizaciones mayores (nuevas funcionalidades).
  + **Capacitación Anual:** Una sesión de capacitación en línea para el equipo.

### **2. Estrategia de Monetización**

**Periodo de Prueba Gratis:**

* + **Duración:** 14 días.
  + **Propósito:** Permitir que los usuarios prueben todas las funcionalidades del sistema sin compromiso.
  + **Conversión:** Al finalizar el periodo de prueba, se ofrece un descuento para el primer mes en el plan mensual o para el plan anual si deciden suscribirse.

**Upselling y Cross-selling:**

* + **Características Adicionales:** Ofrecer módulos adicionales o servicios complementarios (como reportes avanzados, integración con otros sistemas) por un costo adicional.
  + **Consultoría y Personalización:** Ofrecer servicios de consultoría o personalización del sistema para adaptarse a las necesidades específicas de la Compañía de Bomberos, a cambio de una tarifa única.

**Renovación Automática:**

* + **Facilidad:** Las suscripciones se renuevan automáticamente al final del ciclo (mensual o anual), con notificaciones previas para que el usuario esté informado.

**Descuentos para Organizaciones:**

* + **Ofertas Grupales:** Descuentos para organizaciones que deseen suscribir a múltiples unidades o estaciones de bomberos dentro de la misma región.

### **3. Beneficios de la Suscripción**

**Flexibilidad:** Los usuarios pueden elegir entre una opción mensual más económica o una suscripción anual con beneficios adicionales.

**Ingresos Recurrentes:** Genera ingresos constantes y predecibles a lo largo del tiempo, facilitando la planificación financiera.

**Retención de Clientes:** El modelo de suscripción, combinado con soporte prioritario y actualizaciones regulares, aumenta la satisfacción y la retención del cliente.

# Competencias del Perfil de Egreso

1. **Desarrollar soluciones de software:** El proyecto implica la construcción de una aplicación web que automatice el mantenimiento de la flota de vehículos.
2. **Construir modelos de datos escalables:** Se utilizará una base de datos relacional que soporta un creciente volumen de datos sobre los vehículos y su mantenimiento.
3. **Realizar pruebas de certificación:** Se implementarán pruebas de integración y seguridad para garantizar la funcionalidad del sistema.

# Objetivos Generales

Desarrollar una web para el control y seguimiento de la mantención de vehículos de emergencia de la Compañía

de Bomberos de Talcahuano.

# Objetivos Específicos

1. Diseñar una interfaz intuitiva para el registro y monitoreo del estado de los vehículos de emergencia.
2. Implementar un sistema de alertas automatizadas para el mantenimiento preventivo y correctivo.
3. Desarrollar un módulo para la gestión de inventarios de repuestos y control de insumos.
4. Validar el sistema mediante pruebas funcionales y de rendimiento.
5. Documentar el sistema y capacitar a los usuarios finales.

# Metodología

Se utilizará una **metodología ágil** (Scrum), que incluye las siguientes etapas:

1. Levantamiento de Requerimientos: Mapeo de actores y análisis del desafío.
2. Definición de Épicas e Historias de Usuario: Creación del product backlog priorizado.
3. Desarrollo de Sprints: Ciclos iterativos que incluyen planificación, desarrollo, revisión y retrospectiva.
4. Validación y Verificación: Pruebas de integración y certificación de la aplicación.
5. Entrega y Cierre: Presentación final y recopilación de lecciones aprendidas.

”Gestión Fuego” (Sistema de Control y Seguimiento de mantención de vehículos de emergencia de la

compañía de bomberos de Talcahuano)

#### Levantamiento de requerimientos:

los actores identificados en el proyecto son:

Bomberos:

* Rol: Usuarios finales que dependen de la disponibilidad de los vehículos de emergencia.
* Interacción con el sistema: Indirecta; se aseguran de que los vehículos estén listos para emergencias.

Mecánicos:

* Rol: Responsables del mantenimiento preventivo y correctivo de los vehículos.
* Interacción con el sistema: Acceden y actualizan el historial de mantenimiento y reciben alertas sobre tareas pendientes.

Jefes de Flota o Jefes de Operaciones:

* Rol: Supervisores de la operatividad y disponibilidad de los vehículos.
* Interacción con el sistema: Revisión de informes, planificación de mantenciones y toma de decisiones sobre asignación de vehículos.

Personal Administrativo:

* Rol: Encargados de la gestión de inventarios y compras de repuestos.
* Interacción con el sistema: Monitoreo de inventarios y coordinación con proveedores.

Directivos de la Compañía de Bomberos:

* Rol: Responsables de la toma de decisiones estratégicas y la asignación de recursos.
* Interacción con el sistema: Reciben informes sobre el estado de la flota y toman decisiones de largo plazo.

Análisis de desafío:

* Problema Identificado: La falta de un sistema centralizado para el seguimiento y control de la mantención de los vehículos de emergencia, lo que genera ineficiencias en la gestión de los mismos.
* Objetivo del Levantamiento: Identificar todos los requerimientos funcionales y no funcionales, así como las expectativas de cada actor involucrado en el proceso de mantenimiento de los vehículos.

#### Definición de Épicas e Historias de Usuario:

La Compañía de Bomberos enfrenta el desafío de garantizar que su flota de vehículos de emergencia esté siempre en condiciones operativas óptimas. Actualmente, la falta de un sistema centralizado para gestionar el mantenimiento de estos vehículos conlleva varios riesgos, incluyendo tiempos de inactividad inesperados y la posibilidad de fallas mecánicas durante emergencias críticas.

Causas Potenciales del Problema:

* Dependencia de Procesos Manuales:
  + El mantenimiento de los vehículos se gestiona principalmente de forma manual, lo que puede llevar a errores en el registro de datos, omisiones en el seguimiento de tareas pendientes, y dificultad para acceder al historial de mantenimiento.
* Falta de Alertas Automatizadas:
  + No existen mecanismos automatizados que avisen cuando un vehículo necesita mantenimiento preventivo o cuando alguna reparación programada está pendiente, lo que aumenta el riesgo de fallas no detectadas.
* Dificultad en la Coordinación Entre Actores:
  + La comunicación y coordinación entre mecánicos, jefes de flota, y personal administrativo es complicada y puede ser ineficiente, afectando la planificación y ejecución del mantenimiento.

de agosto hasta noviembre, considerando un ciclo de trabajo de 2 semanas por Sprint, lo que da un total de aproximadamente 6 Sprints.

Sprint 1: Semana 1- Semana 4

Planificación:

* Definir las funcionalidades principales que deben estar implementadas en la primera versión del sistema.
* Priorizar tareas relacionadas con la creación de la base de datos, interfaces básicas, y la estructura general del proyecto.
* Asignación de tareas entre los dos programadores.

Desarrollo:

* Configuración del entorno de desarrollo.
* Implementación de la estructura de la base de datos.
* Desarrollo de la interfaz de usuario básica (login, dashboard inicial).
* Integración inicial de backend y frontend.

Revisión:

* Verificar la correcta configuración del entorno y la base de datos.
* Realizar pruebas iniciales de la interfaz y la integración.

Retrospectiva:

* Evaluar el proceso de configuración y desarrollo inicial.
* Identificar problemas o bloqueos y proponer mejoras.

Sprint 2: semana 4 a semana 6

Planificación:

* Focalizarse en el desarrollo de funcionalidades clave como la gestión de vehículos y usuarios.
* Definir criterios de aceptación más detallados para las funcionalidades a desarrollar.

Desarrollo:

* Implementación de CRUD (Create, Read, Update, Delete) para la gestión de vehículos y usuarios.
* Desarrollar las vistas correspondientes y la lógica de negocio necesaria.

Revisión:

* Pruebas funcionales del módulo de gestión de vehículos y usuarios.
* Verificar la usabilidad de las interfaces desarrolladas.

Retrospectiva:

* Analizar el progreso realizado y la efectividad de la gestión de tareas.
* Ajustes en la planificación de tareas para los próximos Sprints.

Sprint 3: semana 6 a semana 8

Planificación:

* Priorizar la implementación de alertas automatizadas y funcionalidades relacionadas con el mantenimiento de vehículos.
* Planificar la integración con el módulo de inventario.

Desarrollo:

* Desarrollo del sistema de alertas para mantenimiento preventivo y correctivo.
* Implementación del módulo de inventario de repuestos.

Revisión:

* Probar el funcionamiento de las alertas automatizadas.
* Validar la lógica del inventario y su interacción con el sistema.

Retrospectiva:

* Evaluar la efectividad de las alertas y el módulo de inventario.
* Identificar mejoras en la coordinación del equipo y la distribución de tareas.

### Sprint 5: semana 8 a 10

Planificación:

* Preparar el sistema para las pruebas finales de validación y verificación.
* Planificar la creación de documentación y guías de uso.

Desarrollo:

* Realizar ajustes finales en el sistema basado en las pruebas de integración.
* Crear documentación de usuario y guías técnicas.

Revisión:

* Verificar que el sistema esté listo para las pruebas finales.
* Revisión de la documentación y guías de uso.

Retrospectiva:

* Evaluar la preparación del sistema para la validación final.
* Discutir cualquier reto o problema que pueda surgir durante la validación.

### Sprint 6: semana 10 a 12

Planificación:

* Planificar la entrega final del proyecto y la presentación.
* Revisión de los objetivos generales del proyecto para asegurar que todo esté cubierto.

Desarrollo:

* Realizar las pruebas finales de validación y verificación.
* Preparar la presentación final del proyecto.

Revisión:

* Validar y verificar que el sistema cumpla con todos los requisitos planteados.
* Preparar y revisar la presentación final.

Retrospectiva:

* Reflexionar sobre todo el proceso de desarrollo.
* Documentar las lecciones aprendidas y sugerencias para futuros proyectos.

Validación y Verificación

Objetivo: Asegurarse de que los diferentes módulos de la aplicación funcionan correctamente cuando se combinan.

Actividades:

* Definición de Casos de Prueba de Integración:
  + Casos de Prueba de Módulo: Verifica que cada módulo (gestión de vehículos, usuarios, inventario, alertas) interactúe correctamente con los demás.
  + Flujos de Trabajo Complejos: Pruebas que involucren múltiples módulos, como la creación de un vehículo, asignación de un usuario responsable, y activación de alertas de mantenimiento.
  + Escenarios de Error: Verificación de cómo el sistema maneja errores cuando los módulos interactúan (por ejemplo, manejo de datos incorrectos o falta de respuesta de un módulo).

Ejecutar las Pruebas de Integración:

* Entorno de Pruebas: Asegúrate de que el entorno de pruebas esté configurado de manera similar al entorno de producción.
* Automatización de pruebas: Si es posible, implementar pruebas automatizadas para los casos de integración. Herramientas como Selenium para pruebas de interfaz de usuario, o Postman para pruebas de API, pueden ser útiles.
* Ejecución de Casos de Prueba: Realiza las pruebas de integración siguiendo los casos definidos, registrando cualquier fallo o comportamiento inesperado.

Documentación y Corrección de Errores:

* Registro de Resultados: Documenta los resultados de las pruebas, especificando qué casos pasaron, fallaron o requieren revisión.
* Corrección de Errores: Si se identifican fallos durante las pruebas de integración, prioriza la corrección de los mismos y reitera las pruebas hasta que se logre una integración exitosa.

Certificación de la Aplicación

Actividades:

* Revisión de Requisitos y Criterios de Aceptación:
  + Checklist de Requisitos: Verifica que todos los requisitos del proyecto hayan sido implementados y que cumplan con los criterios de aceptación establecidos al inicio del proyecto.
  + Cumplimiento Normativo: Asegúrate de que la aplicación cumple con cualquier normativa relevante, especialmente en términos de seguridad y privacidad de los datos.
* Pruebas de Rendimiento y Escalabilidad:
  + Pruebas de Carga: Simula múltiples usuarios interactuando con el sistema para asegurar que la aplicación puede manejar la carga esperada sin degradar el rendimiento.
  + Pruebas de Resistencia: Ejecuta la aplicación durante un período prolongado bajo carga para identificar problemas de rendimiento o errores que puedan surgir con el tiempo.
* Pruebas de Seguridad:
  + Análisis de Vulnerabilidades: Usa herramientas como OWASP ZAP para identificar posibles vulnerabilidades de seguridad en la aplicación.
  + Pruebas de penetración: Simula ataques en la aplicación para comprobar su resistencia a posibles amenazas.
* Revisión Final y Documentación:
  + Revisión Completa del Código: Realiza una revisión del código final para asegurarte de que cumple con los estándares de calidad y seguridad.
  + Documentación Completa: Asegúrate de que toda la documentación técnica y de usuario esté completa, incluyendo guías de instalación, uso y mantenimiento.
* Certificación y Aprobación Final:
  + Revisión por Stakeholders: Presenta la aplicación y los resultados de las pruebas a los stakeholders o al equipo responsable para obtener su aprobación.
  + Aprobación y Firma: Una vez que todos los aspectos han sido verificados y validados, se puede proceder a la firma de la certificación que indica que la aplicación está lista para su implementación y uso.

Entrega y cierre

durante el proceso pudimos percatarnos de varias cosas relacionadas con el ámbito de trabajo de los bomberos, también pudimos dar un primer vistazo al orden de cómo ejecutaremos cada procedimiento de manera e iremos puliendo los detalles por el camino,

Sistema de Control y Seguimiento de Mantención de Vehículos de Emergencia

Compañía de Bomberos de Talcahuano

Desarrollar un sistema para gestionar el mantenimiento de vehículos de emergencia para mejorar la eficiencia de una compañía de bomberos por falta de control de mantenimiento.

tenemos el objetivo de mejorar la seguridad las soluciones que propusimos fueron:

Sistema: Gestión integral del mantenimiento de vehículos de emergencia.

Funcionalidades: Registro de vehículos, alertas de mantenimiento, inventario de repuestos.

Beneficios: Mayor control, reducción de riesgos operativos.

Usamos la metodología scrum siguiendo las siguientes etapas: Registro de vehículos, alertas de mantenimiento, inventario de repuestos, para un Mayor control y reducción de riesgos operativos.

las herramientas que usamos son:

los resultados que esperamos obtener son un Sistema implementado y validado, esto con un impacto que Mejora en la gestión del mantenimiento y mejor capacidad de respuesta.

Las lecciones que aprendimos

Mejora en la gestión del mantenimiento de los equipos usados, puesto que si no se lleva registro podría desembocar en daños graves para uno mismo o alguien más, por lo que el registro constante de ciertos cambios necesita ser llevado rigurosamente

# Stack tecnológico

* Lenguaje de código a ocupar: python
* Programas de gestión: drive, trelo, github, datamodeler, herramientas Microsoft office, bizagi
* Programación: visual studio code, sql server

# Tabla de actividades

|  |
| --- |
| **Plan de Trabajo Proyecto APT** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Competencia o unidades de competencias | Nombre de Actividades/Tareas | Descripción Actividades/Tareas | Recursos | Duración de la actividad | Responsable | Observaciones |
| Gestión, Programación y Base de Datos | Definición del proyecto | Establecer los requisitos, objetivos y alcance del proyecto. | Internet, herramientas microsoft Office | 4 semanas (Semana 1-4) | Alan Gatica | Dificultades para definir exactamente el alcance del proyecto debido a posibles cambios en las necesidades del cliente. |
| gestion | programar, gestion | estructuración de proyecto, organización de tareas y programador subordinado | internet, sql, windows 10 en adelante, pc con grafica si es posible y u ryzen 5 | 11 semanas (Semana 5-15) | Javier Ramirez | Problemas con los plazos debido a problemas externos, como dificultades técnicas o disponibilidad de recursos. |
| Gestión | Presentación del proyecto | Preparar y realizar la presentación final del proyecto. | Documentos de presentación, herramientas de presentación | 3 semanas (Semana 16-18) | Alan Gatica  Javier Ramirez | Coordinación para la presentación final y preparación de la documentación |

# Carta gantt

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Actividad** | **Fase 1** | | | | **Fase 2** | | | | | | | | | | | | **Fase 3** | | |
| **S 1** | **S 2** | **S 3** | **S 4** | **S 5** | **S 6** | **S 7** | **S 8** | **S 9** | **S 10** | **S 11** | **S 12** | **S 13** | **S 14** | **S 15** | **S 16** | | **S 17** | **S 18** |
| **Reunión Inicial con Stakeholders** | **x** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |
| **Investigación sobre Tecnologías** | **x** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |
| **Recopilación de Requisitos** | **x** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |
| **Redacción del Documento de Especificaciones** |  | **x** |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |
| **Diseño Preliminar de la Arquitectura** |  | **x** |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |
| **Revisión de Requisitos con Stakeholders** |  | **x** |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |
| **Planificación del Proyecto** |  |  | **x** |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |
| **Preparación de Herramientas y Entornos** |  |  | **x** |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |
| **Revisión Final de la Fase 1** |  |  |  | **x** |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |
| **Diseño y Configuración de la Base de Datos** |  |  |  |  | **x** |  |  |  |  |  |  |  |  |  |  |  | |  |  |
| **Creación de Prototipos de la Interfaz de Usuario** |  |  |  |  | **x** |  |  |  |  |  |  |  |  |  |  |  | |  |  |
| **Desarrollo de la Interfaz de Usuario** |  |  |  |  |  | **x** |  |  |  |  |  |  |  |  |  |  | |  |  |
| **Desarrollo del Módulo de Gestión de Vehículos** |  |  |  |  |  | **x** |  |  |  |  |  |  |  |  |  |  | |  |  |
| **Desarrollo del Módulo de Mantenimiento** |  |  |  |  |  |  | **x** |  |  |  |  |  |  |  |  |  | |  |  |
| **Integración de Módulos** |  |  |  |  |  |  | **x** |  |  |  |  |  |  |  |  |  | |  |  |
| **Implementación de Funcionalidades Adicionales** |  |  |  |  |  |  | **x** |  |  |  |  |  |  |  |  |  | |  |  |
| **Desarrollo del Módulo de Inventario** |  |  |  |  |  |  |  | **x** |  |  |  |  |  |  |  |  | |  |  |
| **Pruebas Unitarias de Módulos** |  |  |  |  |  |  |  |  | **x** |  |  |  |  |  |  |  | |  |  |
| **Pruebas de Integración** |  |  |  |  |  |  |  |  | **x** |  |  |  |  |  |  |  | |  |  |
| **Corrección de Errores** |  |  |  |  |  |  |  |  |  | **x** |  |  |  |  |  |  | |  |  |
| **Desarrollo de Reportes e Informes** |  |  |  |  |  |  |  |  |  | **x** |  |  |  |  |  |  | |  |  |
| **Validación de Funcionalidades con Usuarios** |  |  |  |  |  |  |  |  |  |  | **x** |  |  |  |  |  | |  |  |
| **Ajustes Finales y Preparación de Documentación** |  |  |  |  |  |  |  |  |  |  | **x** |  |  |  |  |  | |  |  |
| **Preparación para Pruebas Finales** |  |  |  |  |  |  |  |  |  |  |  | **x** |  |  |  |  | |  |  |
| **Redacción del Informe Final** |  |  |  |  |  |  |  |  |  |  |  |  | **x** |  |  |  | |  |  |
| **Creación del Manual de Usuario y Documentación** |  |  |  |  |  |  |  |  |  |  |  |  |  | **x** |  |  | |  |  |
| **Preparación de la Presentación Final** |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **x** |  | |  |  |
| **Ensayo de la Presentación** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **x** | |  |  |
| **Presentación Final** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | **x** |  |
| **Evaluación y Cierre del Proyecto** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | **x** |

# Evidencia y justificación

|  |  |  |  |
| --- | --- | --- | --- |
| Tipo de evidencia  (avance o final) | Nombre de la evidencia | Descripción | Justificación |
| AVANCE | Documento de Especificación de Requerimientos | Describe las evidencias acordadas con tu docente, siempre teniendo en mente que estas deben dar cuenta del desarrollo de tu Proyecto APT. Documento que detalla los requerimientos funcionales y no funcionales levantados durante la primera fase del proyecto. | Justifica cómo se realizó un levantamiento completo de requerimientos que guiarán el desarrollo posterior del sistema |
| Avance | Prototipo de la Interfaz Gráfica | Prototipo o mockup de la interfaz gráfica del sistema, mostrando las pantallas principales (login, dashboard, gestión de vehículos). | Muestra la planificación del diseño y usabilidad de la aplicación antes del desarrollo final, asegurando que cumple con los requisitos de los usuarios. |
| Final | Código Fuente del Sistema | Código fuente del sistema desarrollado en Python y Vue.js, incluyendo el backend, frontend, y base de datos. | Evidencia técnica que prueba el desarrollo funcional del sistema, cumpliendo con los requerimientos planteados al inicio. |
| Final | Informe de Pruebas y Validación | Informe que documenta las pruebas de integración, funcionales y de usuario realizadas para verificar y validar el sistema. | Justifica la calidad del sistema desarrollado, asegurando que cumple con los estándares y requisitos, y que funciona correctamente en diferentes escenarios. |
| Final | Manual de Usuario y Documentación Técnica | Documentación que incluye un manual para usuarios finales y guías técnicas para desarrolladores y mantenedores del sistema. | Facilita la implementación y el uso del sistema por parte de los usuarios finales y garantiza que futuros desarrolladores puedan mantener y actualizar el sistema. |

# Requerimientos:

## Requerimientos NO Funcionales:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| [R-N°] | [Nombre del Requerimiento] | Tipo Requerimiento  [Funcional, No Funcional] | Actores Relacionados | [Descripción corta del requerimiento] |
| R-9 | Buen rendimiento | NO Funcional | Sistema | El sistema debe tener un buen rendimiento y  ser capaz de manejar múltiples usuarios  simultáneamente sin demoras notables |
| R-10 | Seguridad de Datos | NO Funcional | Sistema | Los datos deben estar protegidos mediante cifrado  y solo accesibles por usuarios autorizados |
| R-11 | Interfaz de Usuario Intuitiva | NO Funcional | Usuario | La interfaz debe ser fácil de usar y comprensible  para todos los usuarios, incluso aquellos  sin conocimientos técnicos |
| R-12 | Disponibilidad del Sistema | NO Funcional | Sistema | El sistema debe estar disponible 24/7,  con un tiempo de inactividad mínimo |
| R-13 | Escalabilidad | NO Funcional | Sistema | El sistema debe ser escalable para soportar un  mayor número de usuarios y vehículos sin afectar  el rendimiento |

## Requerimientos Funcionales:

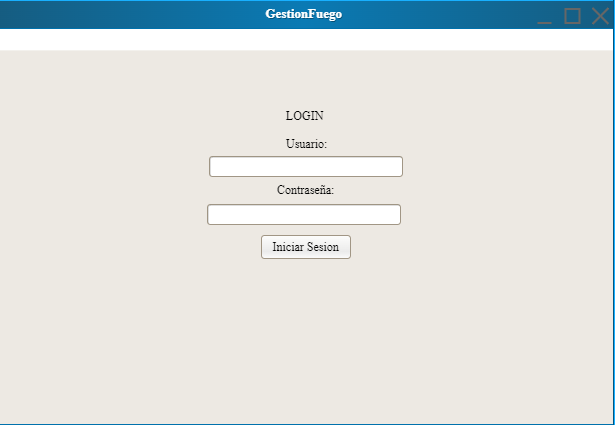
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| [R-N°] | [Nombre del Requerimiento] | Tipo Requerimiento  [Funcional, No Funcional] | Actores Relacionados | [Descripción corta del requerimiento] |
| R-1 | Registro de Vehículos | Funcional | Administrador | Permitir al administrador registrar nuevos vehículos en el sistema con información detallada, como matrícula, marca, modelo, año de fabricación, y tipo de vehículo. |
| R-2 | Asignación de Mantenimiento | Funcional | Administrador | El sistema debe permitir asignar tareas de  mantenimiento a técnicos específicos  para cada vehículo. |
| R-3 | Notificaciones de Mantenimiento | Funcional | Tecnico | Enviar notificaciones automáticas a los  técnicos cuando se asignen nuevas tareas  de mantenimiento o se acerquen las fechas  de vencimiento de tareas. |
| R-4 | Seguimiento del Mantenimiento | Funcional | Administrador,  Técnico | Proveer una interfaz para que los técnicos  actualicen el estado de las tareas de  mantenimiento y que los administradores  puedan seguir el progreso. |
| R-5 | Historial de Mantenimiento | Funcional | Administrador, Usuario | Mantener un registro completo del historial de  mantenimiento de cada vehículo, accesible  para el administrador y los usuarios autorizados |
| R-6 | Gestión de Insumos | Funcional | Técnico | Permitir a los técnicos registrar el uso de insumos  durante las tareas de mantenimiento y alertar al  administrador cuando los niveles de insumos  estén bajos. |
| R-7 | Generación de Reportes | Funcional | Administrador | El sistema debe generar reportes de mantenimiento  completado y pendientes, filtrados por vehículo,  fecha o técnico |
| R-8 | Compartir Horarios | Funcional | Usuario | Permitir que los usuarios compartan sus horarios de  disponibilidad para la planificación de tareas. |

**Evidencias**:

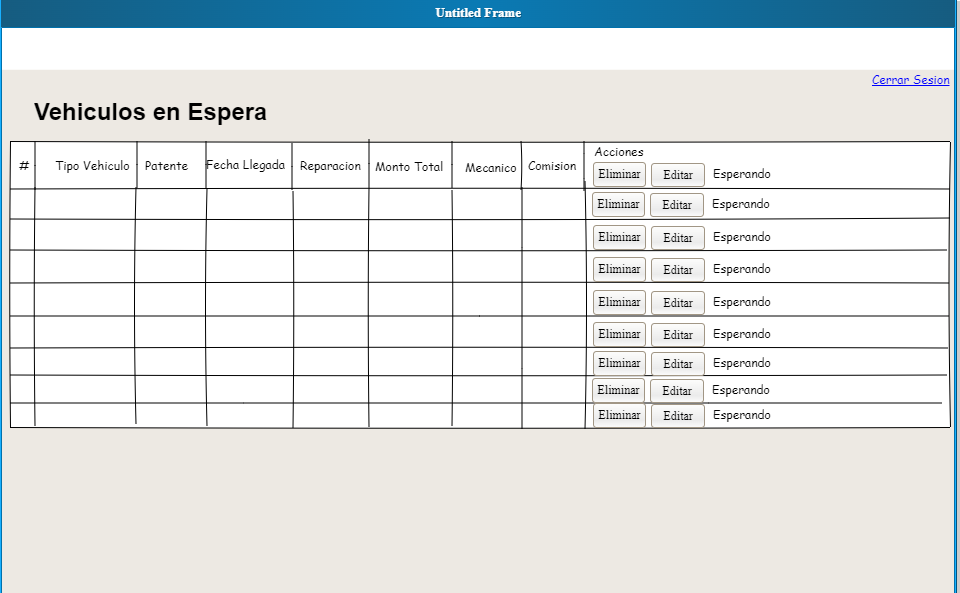
* Documento de especificación de requerimientos levantados.

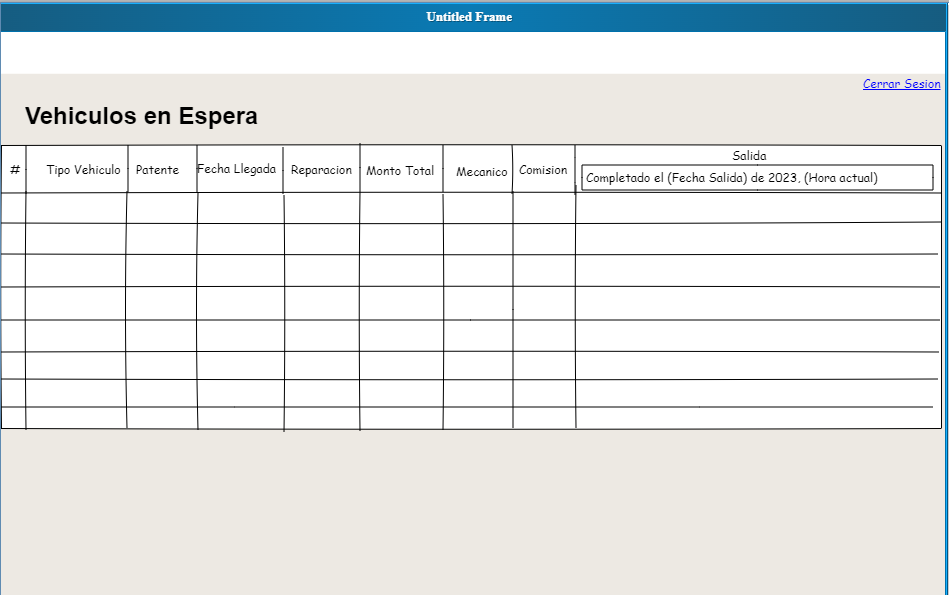
(link a git) https://github.com/JavierR216/GestionFuego

* Mockups de las interfaces principales









* Primeras implementaciones en el sistema

https://docs.google.com/presentation/d/1CF4B0WoDrX6WclCGLrDne4QFbE5\_tnEN6guR7aWfepk/edit#slide=id.g3083bd8d04c\_0\_10

* Revisión de los sprints y sus respectivas tareas concluidas.

**Reflexión sobre cambios**:

* Durante el desarrollo del proyecto, nos dimos cuenta de que algunas de las funcionalidades planeadas originalmente no ofrecían el nivel de automatización necesario para asegurar una gestión óptima de los vehículos. Por ejemplo, inicialmente, las alertas de mantenimiento se configuraron para que fueran manuales, pero tras revisar los procesos, decidimos integrar un sistema de alertas automatizadas que se generara en función de parámetros críticos como el kilometraje o el tiempo desde el último mantenimiento. Además, observamos que era necesario optimizar la interfaz de usuario para facilitar la interacción de los técnicos con el sistema, simplificando la actualización del estado de los vehículos. Estos cambios no solo mejoraron la usabilidad del sistema, sino que también redujeron la posibilidad de errores humanos en el proceso de gestión del mantenimiento.

**Stack Tecnológico**

1. **Lenguaje de Programación**:
   * **Python**: Seleccionamos Python por su robustez en el desarrollo backend y su extensa comunidad, lo que facilita encontrar soluciones y librerías. Además, es compatible con frameworks como Flask y Django.
   * **Vue.js**: Usamos este framework JavaScript por su flexibilidad y facilidad para construir interfaces reactivas y modulares, lo que facilita el desarrollo del frontend.

1. **Bases de Datos**:
   * **SQL Server**: Optamos por SQL Server debido a su capacidad de manejar grandes volúmenes de datos y su compatibilidad con el ecosistema Microsoft, que es ampliamente usado en entornos empresariales.

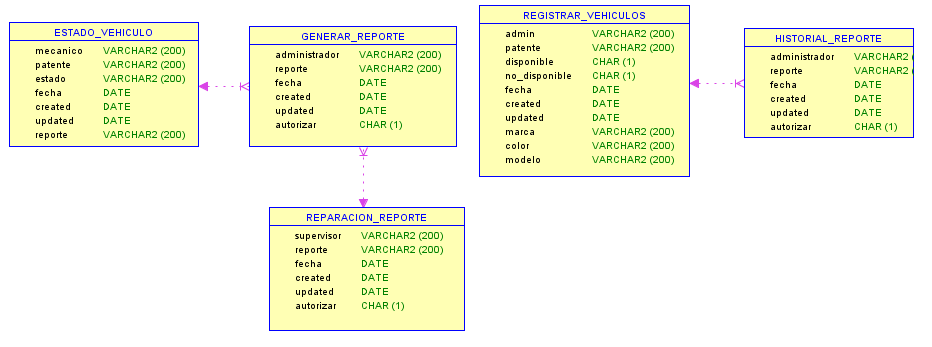
1. **Plataformas y Herramientas**:
   * **Trello**: Para la gestión de tareas y organización de los sprints.
   * **Github**: Para el control de versiones y colaboración entre los desarrolladores.
   * **Microsoft Azure** (opcional, si aplica): Para desplegar la aplicación y asegurar alta disponibilidad.

**Diagrama de arquitectura de la solución**

**Diagrama

Descripción generada automáticamente**

**Base de Datos**

****

**historias de usuario**

**https://docs.google.com/spreadsheets/d/1F-x76Z3OXu40wDOPj8WjhDff4CzAwqbq/edit?gid=335653030#gid=335653030**

**Casos de prueba**

**Interfaz de usuario gráfica, Texto

Descripción generada automáticamente**

**Metodología**

1. **Scrum**: Elegimos Scrum por su capacidad de iterar y adaptarse a cambios rápidos, lo cual fue esencial dado que tuvimos que ajustar ciertas funcionalidades durante el proceso.
2. **Herramientas**: Usamos Trello para gestionar las tareas del equipo, con un backlog donde se priorizaron las historias de usuario

# Conclusión

El sistema desarrollado tiene el potencial de optimizar la gestión del mantenimiento de vehículos de emergencia, reduciendo los tiempos de inactividad y mejorando la respuesta operativa del Cuerpo de Bomberos de Talcahuano. A lo largo del proyecto, hemos aplicado conocimientos técnicos y habilidades adquiridas en nuestra carrera, demostrando la relevancia de las competencias del perfil de egreso en la solución de problemas reales.

The developed system has the potential to optimize the management of emergency vehicle maintenance, reducing downtime and improving the operational response of the Talcahuano Fire Department. Throughout the project, we applied technical knowledge and skills acquired in our studies, demonstrating the relevance of the graduate profile competencies in solving real-world problems.

# Reflexiones individuales

**Javier Ramirez**

A la hora de la planificación del proyecto puede notar un gran énfasis en el uso de los contenidos durante la estadía en duoc, si bien no todos y tampoco explorados a fondo

During the project planning, a strong emphasis can be noticed on the use of the contents covered during the stay at Duoc, although not all of them and not explored in depth.

**Alan Gatica**

Durante el desarrollo de este proyecto, pude aplicar muchos de los conceptos aprendidos en mi carrera, como la programación y gestión de proyectos. Me di cuenta de la importancia de planificar cada paso y trabajar en equipo para lograr un sistema funcional que beneficie a la comunidad.

During the development of this project, I was able to apply many of the concepts learned in my studies, such as programming and project management. I realized the importance of planning each step and working as a team to achieve a functional system that benefits the community.

# Bibliografía

* [Fleet Complete Website](https://www.fleetcomplete.com)
* [Firehouse Software Website](https://www.firehousesoftware.com)
* [Emergency Reporting Website](https://www.emergencyreporting.com)