



UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

DEPARTAMENTO DE ELECTROTECNIA E INFORMÁTICA INGENIERÍA EN INFORMÁTICA

Gestión de proyecto para sistema de monitoreo de cuadrillas en terreno - Geocuadrilla

Alumno: Sebastián Andrés Parra Castillo

Correo: sebastian.parrac@usm.cl

Profesor Guía: Ricardo Cahe Cabach

Conceptos clave: Gestión de proyectos, Metodologías ágiles, Scrum, Georreferenciación, Monitoreo en tiempo real

Resumen: El presente proyecto, aborda la problemática de la gestión y monitoreo de cuadrillas de trabajo en terreno dentro del sector de transmisión de energía eléctrica, específicamente en el contexto de Chilquinta Transmisión, debido a que la coordinación de personal en terreno implica altos costos operativos y dificultades para obtener información en tiempo real. Como solución, se desarrolló el sistema de GeoCuadrilla, que es una solución tecnológica compuesta por un panel web y una aplicación móvil que permite georreferenciar las cuadrillas en tiempo real, visualizando el estado operativo y facilitando la toma de decisiones. El proyecto fue gestionado mediante la metodología Scrum, organizando trabajo en sprints iterativos que permitieron validar avances, incorporar feedback y adaptarse a cambios. La aplicación de este enfoque permitió entregar una solución funcional, escalable y alineada con las necesidades del patrocinador.



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☒ Memoria o trabajo de título ☐ Tesis de Postgrado

Título del trabajo: Gestión de proyecto para sistema de monitoreo de cuadrillas en terreno - Geocudrilla

Nombre del candidato(a): Sebastián Andrés Parra Castillo

Carrera / Grado: Ingeniería en Informática

Campus: Viña del mar **Departamento:** Informática

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Ricardo Cahe Cabach, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

☒ El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por (**marcar una opción**):

☐ 6 meses ☐ 12 meses ☐ 2 años ☐ 3 años ☐ 5 años ☐ 10 años

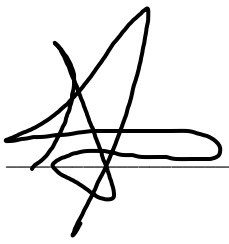
Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 03-09-2026 **Firma:** 

Estudiante o Candidato(a):

Fecha: 03/09/2026 **Firma:** 

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

Índice

1. Introducción	3
1.1 Planteamiento del problema	3
1.2 Objetivo general	3
1.3 Objetivos Específicos	3
1.4 Historias de Usuario	3
2. Marco Teórico	4
2.1 Gestión de proyectos.	4
2.2 Enfoques tradicionales vs ágiles.	5
2.3 Descripción de metodologías ágiles.	5
2.4 Beneficios de la gestión ágil en proyectos.	7
2.5 Requerimientos de Software	7
2.6 Sistemas de Georreferenciación	8
2.7 Gestión de la comunicación	8
2.8 Tecnologías Web y Móviles	8
3. Metodología y Plan de trabajo	8
3.1 Enfoque para la implementación de metodologías ágiles en el proyecto.	8
3.2 Herramientas de gestión empleadas.	9
4. Planificación de Sprints	9
4.1 Definición de Sprints y ciclos de desarrollo.	9
4.2 Estimación de tareas y asignación de recursos	9
4.3 Funcionalidades aplicadas	10
4.4 Desarrollo real y porcentajes de cumplimiento en diagrama Gantt	12
5. Ejecución de reuniones ágiles	12
5.1 Reuniones Semanales	12
5.2 Reuniones de Planificación (Sprint Planning)	12
5.3 Reuniones de revisión (Sprint Review)	13
5.4 Retrospectivas	13
5.5 Importancia de la comunicación y colaboración en el equipo.	13
6. Gestión de Cambios y Mejoras Continuas	14

6.1 Adaptación a cambios en los requerimientos	14
6.2 Implementación de Mejoras basadas en Feedback	14
6.3 Tabla de Gestión de Cambios	14
6.4 Tabla de Gestión de Riesgos	15
7. Evaluación del Desempeño Ágil	16
7.1 Métricas utilizadas para medir la eficiencia y efectividad.	16
7.2 Análisis de resultados obtenidos	22
8. Desafíos y Soluciones	23
8.1 Obstáculos Identificados.	23
8.2 Estrategias para solucionar los problemas	24
9. Estimación de costos.	24
10. Conclusiones	25
Bibliografía	26

1. Introducción

1.1 Planteamiento del problema

Chilquinta Transmisión, empresa dedicada a la transmisión de energía eléctrica con altos estándares de confiabilidad, sustentabilidad y calidad de servicio, planteó una problemática la cual tiene que ver con el cómo ellos gestionan sus cuadrillas de trabajo en terreno a lo largo de toda la red de transmisión, ya que el costo logístico de movilizar la fuerza de trabajo en terreno es muy alto, y a su vez, son los encargados de responder ante emergencias que pongan en peligro la continuidad de servicios de transmisión y distribución eléctrica.

Con esto en mente, ellos nos pidieron la solución que está a continuación.

1.2 Objetivo general

Diseñar e implementar una solución tecnológica para la gestión y monitoreo de las cuadrillas de trabajo de Chilquinta Transmisión, compuesta por un panel web y una aplicación móvil que permitan georreferenciar a los trabajadores que están en terreno en tiempo real, visualizando su ubicación en un mapa interactivo, facilitando la coordinación y el seguimiento de sus operaciones en terreno.

1.3 Objetivos Específicos

- i. Definir, documentar y priorizar los requerimientos funcionales y no funcionales del sistema GeoCuadrilla, asegurando claridad, trazabilidad y alineación con las necesidades del patrocinador y los usuarios finales.
- ii. Organizar el equipo de trabajo y fomentar la comunicación efectiva mediante prácticas y herramientas ágiles, promoviendo la responsabilidad compartida, retroalimentación continua y autonomía del equipo.
- iii. Establecer procesos de comunicación y retroalimentación con el patrocinador y actores involucrados, asegurando así la alineación de expectativas, satisfacción del cliente y adaptación eficiente ante cambios de requerimientos.
- iv. Medir el desempeño del proyecto mediante el cumplimiento de cronograma, esfuerzo, satisfacción del usuario, evaluando el impacto de la gestión ágil en los resultados.
- v. Generar y mantener actualizada la documentación del proyecto, incluyendo actas, backlog, reportes de avance y entregables funcionales, garantizando trazabilidad y transparencia a lo largo del proceso.

1.4 Historias de Usuario

Para contextualizar acerca del desarrollo e implementación de la solución tecnológica requerida por Chilquinta Transmisión, se han elaborado las siguientes historias de usuario:

- **ÉPICA 1: Monitoreo en tiempo real de cuadrillas:**
 - o **HU1 – Visualización de cuadrillas en el mapa:**

Como usuario del panel web, quiero ver en un mapa interactivo la ubicación en tiempo real de todas las cuadrillas, para poder monitorear su posición, desplazamiento y disponibilidad de forma eficiente.

Criterios de Aceptación:

- Dado que estoy en el panel web, y en la vista general del mapa, entonces debo ver todas las cuadrillas con su posición actual en el mapa.
 - Cuando una cuadrilla se mueve, el mapa debe actualizar su ubicación en el menor tiempo posible.
 - Se deben mostrar datos básicos, tales como nombre de la cuadrilla, estado y su última actualización.
- **HU2 – Visualización de detalles de una cuadrilla:**
Como usuario del panel web, quiero hacer clic en una cuadrilla del mapa, para poder ver su información detallada (ID cuadrilla, patente, técnico asociado, trabajo en ejecución).
Criterios de aceptación:
 - Al hacer clic en un marcador, debe aparecer sobre la cuadrilla la información requerida.
- **ÉPICA 2: Aplicación móvil para los trabajadores en terreno**
 - **HU3 – Compartir ubicación al iniciar jornada:**
Como trabajador en terreno, quiero iniciar mi jornada desde la aplicación móvil, para comenzar a compartir mi ubicación con la torre de control.
Criterios de aceptación:
 - La aplicación debe pedir permiso para compartir ubicación.
 - Una vez iniciada la jornada, tiene que enviarse la ubicación actual cada X segundos.
 - La aplicación debe mostrar un aviso: “Compartiendo ubicación”.
 - **HU4 – Finalizar jornada:**
Como trabajador, quiero finalizar mi jornada y en el envío de ubicación, para evitar difundir datos fuera de horario laboral.
Criterios de aceptación:
 - Debe existir el botón de finalizar jornada
 - Al finalizar, debe dejar de enviar coordenadas y registrar la hora de cierre

2. Marco Teórico

2.1 Gestión de proyectos.

La gestión de proyectos se define como la aplicación de conocimientos, habilidades, herramientas y técnicas para poder planificar, ejecutar y controlar actividades que permitan cumplir con los objetivos específicos de un proyecto dentro de los plazos, costos y calidad previamente definidos (*Project Management Institute, 2021*).

Tradicionalmente, esta disciplina se ha basado en enfoques predictivos o en cascada, donde se realiza una planificación exhaustiva al inicio y se ejecutan las tareas siguiendo una secuencia lineal. Sin embargo, el aumento de la complejidad de los entornos organizacionales y la necesidad de responder con rapidez a cambios y requerimientos del mercado ha impulsado la adopción de enfoques más flexibles y adaptativos, como lo son las metodologías ágiles.

2.2 Enfoques tradicionales vs ágiles.

Los métodos tradicionales, se caracterizan por definir de manera detallada el alcance, los tiempos, y el presupuesto desde el inicio del proyecto, minimizando así la variación durante la ejecución (*Instituto Europeo de Posgrado, 2022*). Por el contrario, las metodologías ágiles surgen como un conjunto de principios que priorizan la entrega temprana de valor, la adaptación constante a cambios y la colaboración continua con el cliente o usuario final (*The Agile Manifesto, 2001*)

El manifiesto ágil, publicado en 2001, establece cuatro valores fundamentales:

1. Individuos e interacciones sobre procesos y herramientas.
2. Software funcionando sobre documentación extensiva.
3. Colaboración con el cliente sobre negociación contractual.
4. Responder ante el cambio sobre seguir un plan.

Estos principios buscan hacer frente a entornos dinámicos y proyectos donde los requisitos pueden ir evolucionando a lo largo del tiempo.

2.3 Descripción de metodologías ágiles.

I. Scrum

Scrum es un marco de trabajo ágil orientado a equipos de trabajo que buscan entregar productos de manera incremental, mediante ciclos de trabajos cortos llamados sprints (*Schwaber & Sutherland, 2020*). En scrum, se definen tres roles principales:

Roles:

- Scrum Master: Facilita el proceso y elimina obstáculos.
- Product Owner: Define y prioriza el backlog del producto.
- Equipo de Desarrollo: Grupo de trabajadores que ejecuta el trabajo técnico necesario.

Eventos Clave:

- Sprint Planning: Se planifican las tareas a realizar en un período de tiempo determinado.
- Daily Scrum: Reunión diaria para coordinar actividades.
- Sprint Review: Presentación de lo completado.
- Sprint Retrospective: Reflexión sobre el proceso para buscar una mejora.

II. Kanban

Kanban se enfoca en visualizar el flujo de trabajo a través de un tablero, permitiendo gestionar y optimizar el flujo de tareas mediante límites de trabajo en progreso (*Work in Progress – WIP*). Su fortaleza más clara se encuentra en la flexibilidad y mejora continua que establece esta metodología.

Elementos Clave:

- Tablero Kanban: Visualización de tareas por estado (por ejemplo, "Por Hacer", "En Progreso", "Hecho").
- Tarjetas: Representan tareas individuales.

- Límites WIP: Restricciones sobre cuántas tareas pueden estar en progreso al mismo tiempo.

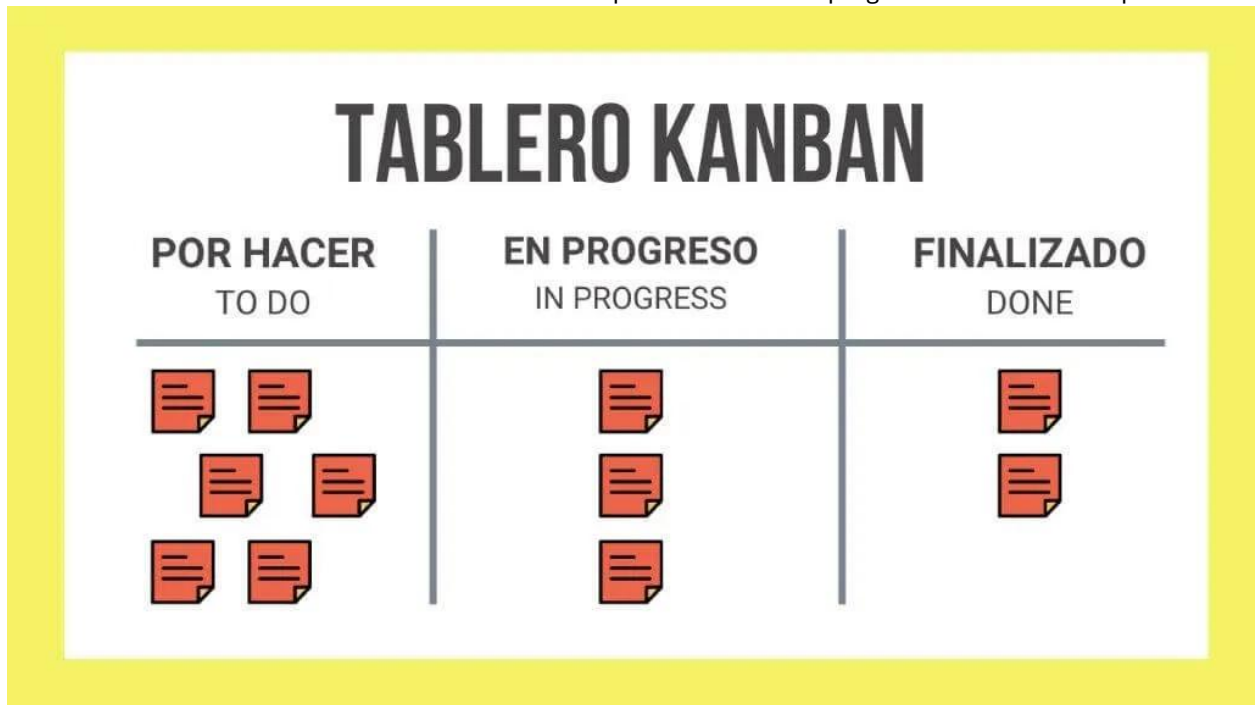


Ilustración 1: Ejemplo de Tablero Kanban

III. *Extreme Programming (XP)*

XP es una metodología ágil que busca mejorar la calidad del software mediante prácticas como la programación en pareja (*pair programming*), desarrollo guiado por pruebas (*Test-Driven Development – TDD*) y la integración continua, lo cual fomenta la comunicación y la simplicidad del código.

Prácticas Clave:

- Programación en Parejas: Dos desarrolladores trabajan juntos en una misma tarea.
- Desarrollo Guiado por Pruebas (TDD): Se escriben pruebas antes de desarrollar el código.
- Integración Continua: El código se integra y se prueba frecuentemente.
- Refactorización: Mejora continua del código existente.

IV. *Lean Software Development*

Basado en los principios lean manufacturing, esta metodología busca eliminar desperdicios, entregar de la forma más rápida posible, fomentar el aprendizaje y empoderar a los equipos para tomar decisiones

Principios Clave:

- Eliminar desperdicios: Cualquier actividad que no agrega valor se debe eliminar.
- Amplificar el aprendizaje: Fomentar la experimentación y el aprendizaje continuo.
- Decidir tarde: Mantener la flexibilidad y la capacidad de respuesta hasta el último momento.
- Entregar lo más rápido posible: Acelerar la entrega de valor al cliente.

2.4 Beneficios de la gestión ágil en proyectos.

La aplicación de metodologías ágiles en la gestión de proyectos presenta varios beneficios:

- **Adaptabilidad y flexibilidad** frente a los cambios de requisitos o condiciones del mercado
- **Mejora Continua** del proceso y del producto mediante retroalimentación continua y retrospectivas.
- **Mayor enfoque en el valor entregado al cliente** a través de entregas parciales y funcionales
- **Identificación temprana de riesgos y problemas**, gracias a la inspección constante y las iteraciones cortas.
- **Colaboración multidisciplinaria** y comunicación más fluida entre los involucrados.

Valores del Manifiesto Ágil

- **Individuos e interacciones sobre procesos y herramientas:** Se prioriza la comunicación y colaboración entre los miembros del equipo en lugar de seguir estrictamente procesos o depender de herramientas.
- **Software funcionando sobre documentación extensiva:** El enfoque está en entregar software que funcione, en lugar de crear una documentación exhaustiva que no contribuya directamente al valor del producto.
- **Colaboración con el cliente sobre negociación de contratos:** Se fomenta una relación colaborativa con el cliente, buscando su participación y ajustando los requisitos según sus necesidades cambiantes.
- **Respuesta ante el cambio sobre seguir un plan:** Se valora la capacidad de adaptarse a cambios en los requisitos, incluso en etapas avanzadas del desarrollo, en lugar de seguir ciegamente un plan predeterminado.

2.5 Requerimientos de Software

Los requerimientos representan las necesidades del usuario o del negocio que un sistema debe satisfacer.

Se clasifican en:

- **Funcionales:** describen comportamientos específicos del sistema.
- **No funcionales:** describen cualidades o restricciones, tales como seguridad, usabilidad, desempeño, disponibilidad.

GeoCuadrilla debe gestionar requerimientos funcionales complejos tales como:

- Geolocalización en tiempo real.
- Registro de jornada laboral.
- Visualización cartográfica

Y a su vez se gestionan requerimientos no funcionales como:

- Bajos requisitos mínimos.
- Rendimiento eficiente.
- Confidencialidad de datos.

Una adecuada gestión del alcance permite evitar desviaciones y maximizar valor.

2.6 Sistemas de Georreferenciación

La georreferenciación es el proceso de vincular elementos digitales con su representación espacial mediante coordenadas geográficas.

Los sistemas de georreferenciación permiten:

- Ubicar recursos en tiempo real.
- Analizar movimientos y patrones.
- Optimizar rutas y tiempos de respuesta.

Dentro del proyecto de GeoCuadrilla, la georreferenciación permite visualizar la posición de trabajadores en terreno mediante un mapa interactivo, generando información valiosa para la toma de decisiones operacionales en el menor tiempo posible.

2.7 Gestión de la comunicación

La gestión de la comunicación asegura que el flujo de información dentro del proyecto sea claro, oportuno y colaborativo.

Los proyectos ágiles integran:

- Retroalimentación continua.
- Reuniones cortas y frecuentes.
- Herramientas de colaboración en línea.

En GeoCuadrilla, herramientas como ClickUp y Discord permiten registrar acuerdos, documentar avances y coordinar tareas de manera asincrónica.

2.8 Tecnologías Web y Móviles

Las tecnologías web permiten construir soluciones accesibles desde navegadores, mientras que las aplicaciones móviles permiten capturar información desde dispositivos portátiles en contextos de movilidad.

Su combinación facilita:

- Flujo bidireccional de datos.
- Comunicación en tiempo real.
- Integración entre operaciones de oficina y terreno

En el contexto de GeoCuadrilla, la aplicación móvil recolecta datos en terreno, mientras que el panel web visualiza estos datos para el análisis y la toma de decisiones.

3. Metodología y Plan de trabajo

3.1 Enfoque para la implementación de metodologías ágiles en el proyecto.

En este proyecto se usará la metodología ágil de Scrum, ya que ofrece alta flexibilidad para la implementación de cambios, y la entrega constante de avances para poder medir la satisfacción del cliente y la efectividad del proceso.

3.2 Herramientas de gestión empleadas.

Las herramientas utilizadas para el monitoreo y comunicación con el equipo son ClickUp y Discord.

Se decidió el uso de ClickUp inicialmente para poder empezar a organizar las reuniones semanales del grupo de trabajo, estableciendo las tareas iniciales de cada participante, y monitoreando el progreso de éstas.

Discord es una aplicación de comunicación remota donde se pueden crear espacios para distintas finalidades, en el caso del proyecto GeoCuadrilla, Discord es usado como nuestra fuente de comunicación en las reuniones, respaldo de información con respecto a actas escritas, cambios en el código, feedback del patrocinador, resumen de reuniones y monitoreo del equipo de desarrollo.

Ambas aplicaciones fueron seleccionadas frente a alternativas como Trello, Microsoft Teams, Google Meet, entre otras, ya que todos los usuarios del grupo contaban con experiencia de uso en ambas aplicaciones, logrando así una adaptabilidad más natural al flujo de trabajo del proyecto y logrando así una mayor eficiencia desde el primer momento.

4. Planificación de Sprints

4.1 Definición de Sprints y ciclos de desarrollo.

El desarrollo del proyecto se estructuró mediante sprint, definidos como ciclos de trabajo iterativos y temporizados con una duración previamente establecida, cuyo propósito es entregar incrementos funcionales del producto para poder ser evaluados y validados por los stakeholders. En el contexto de GeoCuadrilla, las fechas y los contenidos de los sprint vienen designados en el ramo de Proyecto de Software 2, los cuales tienen los siguientes requerimientos:

- Entrega 1: Realizada el 29 de agosto del presente año, tenía como objetivo entregar un MVP funcional básico, una bitácora de feedback y plan de mejoras, roles y responsabilidad en el desarrollo, y una presentación en formato PDF incluida.
- Entrega 2: Realizada el 26 de septiembre del presente año, pedía el MVP actualizado, una nueva bitácora actualizada, registro de actividades y tareas realizadas por cada rol, y una nueva presentación.
- Entrega 3: Realizada el 24 de octubre, exigía una demo en video del MVP final, bitácora, registro de actividades, y una última presentación.

Por lo tanto, tomando en cuenta las fechas y las entregas efectuadas vía aula, los sprints fueron dispuestos de la siguiente manera:

- Sprint 1: 18 de agosto - 29 de agosto
- Sprint 2: 2 de septiembre – 16 de septiembre
- Sprint 3: 7 de octubre – 24 de octubre

4.2 Estimación de tareas y asignación de recursos

A lo largo de este proceso de ejecución del proyecto con nuestro patrocinador, se han establecido 4 funcionalidades que son primordiales para que el proyecto pueda cumplir su objetivo principal.

Bajo esta misma línea, y basándose en los distintos documentos de bitácoras y feedback creados a lo largo del semestre, se definirá un puntaje que haga alusión a la dificultad de implementar estas tareas (en una

medida del 1-10, donde 1 corresponde al menor grado de complejidad y 10 el mayor), y será mostrado de la siguiente manera: Nombre del requerimiento – 3 6 10

El criterio utilizado para la evaluación se basa en los siguientes aspectos:

- Dominio del lenguaje de programación: Esto se define en base a las capacidades del momento en que la tarea fue asignada hacia el programador, ya que un bajo o nulo dominio del lenguaje que se está usando puede llevar a un mayor tiempo de trabajo gastado en comprender la lógica atrás del lenguaje, buscar documentación al respecto e imaginar el flujo de trabajo de la forma que se le está pidiendo.
- Tiempo de realización: Esto se define gracias a los plazos que el cliente especificó a lo largo de las reuniones del semestre, las cuales normalmente eran acorde con los finales de sprint, pero muchas veces los tiempos eran difíciles de cumplir por factores externos al proyecto, considerando la carga académica, viajes, etc.

A continuación, se mencionarán las funcionalidades aplicadas a lo largo del proyecto.

4.3 Funcionalidades aplicadas

1. Creación de panel web con Google Maps API – 8

Involucra todo el proceso de carga de mapa a panel web, junto con la conexión de la API de Google Maps para poder usar varias de sus funcionalidades en la aplicación.

- Horas de trabajo: 350 horas
- Tareas:
 - o Crear vista donde debería ir el mapa
 - o Hacer la configuración previa para importar Google Maps API
 - o Cargar datos en el mapa
 - o Configurar punto de llegada de los datos para futura implementación de aplicación móvil
 - o Implementar archivos con puntos de interés señalados por nuestro patrocinador
 - o Filtrar puntos de interés por colores solicitados por el patrocinador
 - o Mostrar cuadrillas en el mapa
 - o Añadir una tabla tanto para los puntos de interés como para el estado de las cuadrillas

2. Creación de servidor privado para habilitar tráfico de datos entre aplicación móvil y panel web – 8

Incluye todo el proceso que se necesita internamente para que la aplicación móvil y el panel web puedan hablar entre sí en tiempo real.

- Horas de trabajo: 250 horas
- Tareas:
 - o Habilitar un entorno privado para el tráfico de datos.
 - o Brindar acceso a este entorno al encargado de la aplicación móvil.
 - o Brindar acceso a la encargada del panel web.
 - o Conectar el punto de ambas aplicaciones para permitir el envío de datos.

3. Creación de aplicación móvil – 7

Corresponde a todo el proceso de creación de la aplicación móvil, considerando botones para iniciar y finalizar jornada respectivamente, junto con el servicio de envío de ubicación en tiempo real.

- Horas de trabajo: 300 horas
- Tareas:
 - o Crear vista de inicio de sesión con Microsoft.
 - o Añadir botón de iniciar y finalizar jornada.
 - o Importar el servicio de RabbitMQ para compartir la ubicación en tiempo real.
 - o Configurar la solicitud de permisos para el primer uso de la aplicación

4. Integrar base de datos de nuestro patrocinador en el panel web – 9

Debido a que trabajamos activamente con nuestro patrocinador Chilquinta Transmisión, se nos pidió importar la base de datos que ellos tienen en el panel web, para poder correlacionar archivos y agilizar el proceso.

- Horas de trabajo: 350 horas
- Tareas:
 - o Emplear funcionalidad de poder exportar archivos del tipo solicitado
 - o Cargar cada campo del archivo en la aplicación web
 - o Poder ampliar el uso de la base de datos para que con los datos de la aplicación móvil se puedan hacer validaciones en tiempo real con el panel web.
 - o Mostrar parte de los datos de la tabla en el panel web con actualizaciones constantes

Todas estas funcionalidades previamente expuestas, pueden ser desglosadas en grupos más pequeños, pero si consideramos lo que es el punto principal de la aplicación, estos 4 aspectos son los principales al momento de desarrollar nuestra solución.

A continuación, se definirán los objetivos que hubo para cada sprint previamente mencionado.

Sprint 1	Sprint 2	Sprint 3
Igualar el prototipo generado en figma.	Hacer el envío de coordenadas de la aplicación móvil al servidor privado.	Tener tanto la aplicación móvil como el panel web funcionando en un servidor privado.
Integrar la API de Google Maps.	Terminar la integración de Google Maps API	Demostrar funcionamiento de la ubicación en tiempo real.
Mostrar las cuadrillas en terreno y sus distintos estados.	Integrar archivos kmz otorgados por el patrocinador	Añadir opción para agregar archivos.kmz de forma natural

Tabla 1: Objetivos por sprint

4.4 Desarrollo real y porcentajes de cumplimiento en diagrama Gantt

A continuación, se mostrará un diagrama Gantt focalizado en el desarrollo de la solución que correspondió a todo el segundo semestre de 2025, con los objetivos de cada sprint y el porcentaje de cumplimiento de cada uno de estos.

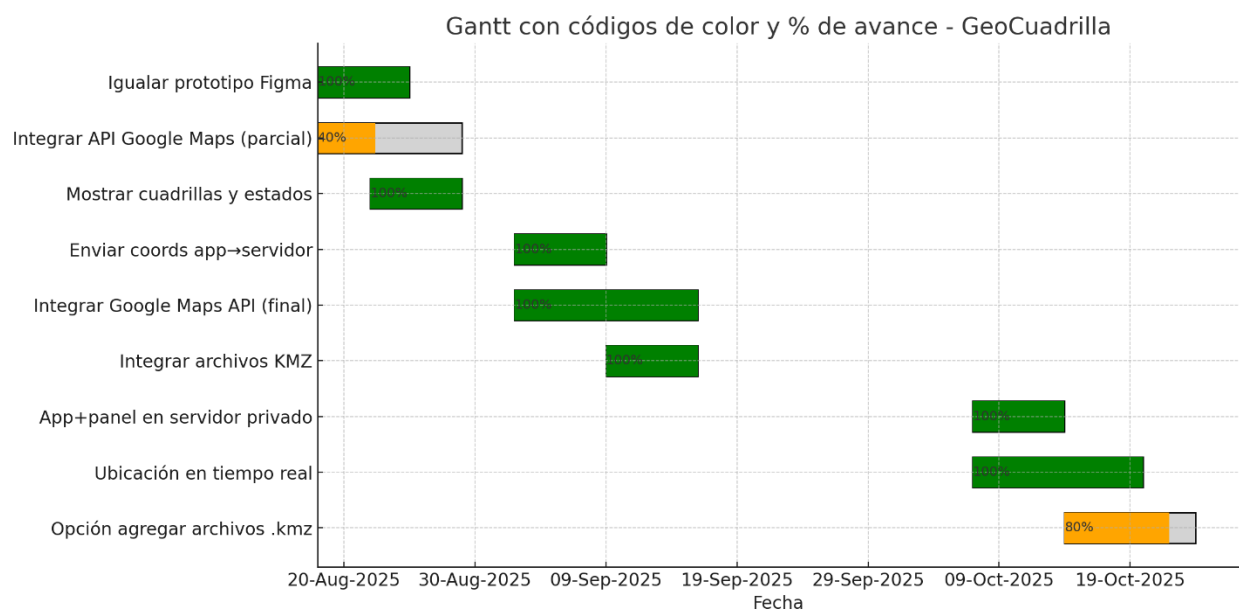


Ilustración 2 Diagrama Gantt con códigos de color y porcentaje de avance

5. Ejecución de reuniones ágiles

5.1 Reuniones Semanales

Durante el proyecto GeoCuadrilla, se realizaron 2 reuniones semanales desde inicio de semestre, que constaban de una llamada vía Discord (de entre 15-30 minutos), con el objetivo de mantener la visibilidad del progreso y detectar posibles impedimentos a la hora de añadir nuevas funcionalidades.

Inicialmente la idea era poder tener reuniones diarias como Daily scrum, pero por problemas de disponibilidad del grupo entero decidimos que lo mejor era dejar 2 días en la semana estipulados entre todos para poder tener estas reuniones de la forma más fructífera posible.

Aunque nuestro equipo era reducido, esta práctica permitió que tengamos una alineación constante, transparencia, y una rápida detección de problemáticas técnicas.

5.2 Reuniones de Planificación (Sprint Planning)

Antes del inicio de cada sprint, el equipo se encargó de definir:

- Funcionalidades prioritarias
- Esfuerzo estimado
- Criterios de aceptación

Estas reuniones influyeron directamente en la gestión del alcance y permitieron asignar tareas nuevas en función de habilidades específicas, lo que optimizó el uso de recursos a lo largo de todo el desarrollo del proyecto.

5.3 Reuniones de revisión (Sprint Review)

Al final de cada sprint se realizaron demostraciones funcionales al patrocinador y profesores encargados de la evaluación de los avances, quienes eran los encargados de entregarnos feedback inmediatos, solicitando mejoras o cambios. Este proceso fue muy importante para GeoCuadrilla, ya que el patrocinador fue claro desde un principio con sus necesidades, y los profesores aportaban ideas que se alineaban con el patrocinador, pero desde un punto de vista más amplio, lo que nos permitía entender mejor la escalabilidad de nuestro proyecto y asegurar la utilidad de nuestro proyecto desde etapas tempranas.

5.4 Retrospectivas

Las retrospectivas permitieron analizar cada sprint desde un punto de vista reflexivo acerca del rendimiento del equipo a nivel general, la calidad técnica del producto, y la calidad de nuestra comunicación a lo largo del proceso, gracias a esto, se lograron identificar mejoras progresivamente, tales como, la fijación de compromisos realizables con el patrocinador, mayor uso de documentación visual para las presentaciones de avances, entre otras.

5.5 Importancia de la comunicación y colaboración en el equipo.

En un proyecto de desarrollo de software gestionado bajo la metodología ágil Scrum, la comunicación y la colaboración juegan un papel importante, ya que permiten que el equipo funcione de una manera coordinada y adaptable frente a los cambios del entorno, los puntos más destacables de lo que refleja todo este proceso son los siguientes:

- Adaptación frente a cambios:
 - La interacción frecuente entre los integrantes del equipo facilita la respuesta de forma ágil a modificaciones en los requerimientos, lo que es un punto clave en proyectos dinámicos, donde las necesidades pueden variar a lo largo del desarrollo.
- Trabajo colaborativo:
 - La colaboración activa mejora las competencias individuales de todo el grupo, generando soluciones más creativas y eficientes. Un equipo con perfiles diversos puede abordar problemas desde múltiples enfoques, teniendo un resultado mucho más fructífero.
- Cultura de mejora constante:
 - Scrum incorpora instancias formales, como las retrospectivas, en las que el equipo se encarga de analizar su desempeño, identificando áreas de mejora y definiendo nuevas prácticas, promoviendo así el aprendizaje continuo y la optimización de procesos.
- Gestión de conflictos y clima laboral:
 - Una comunicación saludable contribuye a manejar desacuerdos de una manera constructiva, reduciendo las tensiones internas y favoreciendo un ambiente de trabajo productivo y respetuoso.
- Uso eficiente del tiempo:
 - Las reuniones periódicas ayudan a ordenar prioridades y coordinar esfuerzos, asegurando que el equipo logre avanzar de forma alineada y con foco en los objetivos.
- Transparencia y visibilidad del estado del proyecto

- Compartir información de manera abierta sobre avances, dificultades y decisiones favorece la transparencia, permitiendo así detectar posibles riesgos de manera temprana y tomar acciones correctivas oportunas.

6. Gestión de Cambios y Mejoras Continuas

6.1 Adaptación a cambios en los requerimientos

El proyecto adoptó un enfoque flexible basado en los principios ágiles, lo que facilita el responder rápidamente a cambios en las prioridades del cliente o en los requerimientos técnicos, se incluyen los siguientes aspectos:

- Priorización del backlog: Ante cualquier cambio brusco en los requerimientos, se acude a revisar el backlog del producto para ver cómo actuar.
- Gestión de Riesgos: Identificar posibles problemas técnicos y organizativos de manera temprana, logrando ajustar tanto tareas como recursos según la necesidad.
- Feedback: Añadir los comentarios de los stakeholders al final de cada sprint para poder ajustar el alcance final del proyecto.

6.2 Implementación de Mejoras basadas en Feedback

Se aplicaron mejoras continuas en:

- UX del panel web
- Calidad visual de marcadores
- Vistas de la aplicación móvil

Estas mejoras son directamente notables en las ilustraciones 3,4 y 5 respectivamente, donde las mejoras efectuadas con en base al feedback recibido por los stakeholders.

6.3 Tabla de Gestión de Cambios

Cambio solicitado	Motivo	Impacto Estimado	Acciones Por Realizar
Eliminar el estado de “En tránsito” en el panel web	Temas de comodidad, no es necesario tener un estado que no se le va a dar uso.	Medio: afecta tanto al backend como al frontend del panel web.	Identificar la función de los estados del vehículo y borrar todo lo referente a ese tercer estado.
Cambiar el color de las líneas de tensión de los archivos.kmz	Hay una normativa a nivel nacional con los colores que deben ser usados, y ellos nos mandan los colores necesarios.	Bajo: Afecta solo al frontend del panel web	Solo cambiar el color de las líneas al establecido por el patrocinador.
Añadir inicio de sesión con Microsoft para la aplicación móvil	Todos los trabajadores tienen un correo de empresa que funciona con Microsoft	Alto: Afecta tanto al back, como al front de la aplicación móvil	Investigar acerca de la aplicación de esta funcionalidad en una aplicación móvil, para posteriormente añadir esta funcionalidad a la

			aplicación y poder probar esto con los correos de la universidad.
Cambiar el logo de las cuadrillas en terreno	Quieren que sean camionetas, y en lo posible reflejar que es de la empresa.	Bajo: afecta directamente al front del panel web	Al ser una petición post feria de software, se tuvo que emplear un tiempo extra en poder resolver esto, pero solo involucra el cambio de logo y ponerle el color a la camioneta dependiendo de si se encuentra disponible o no.

Tabla 2 Gestión de cambios

6.4 Tabla de Gestión de Riesgos

Riesgo identificado		Impacto	Probabilidad	Acciones Preventivas	Acciones Correctivas
Falta de tiempo del equipo		Alto	Alta	Usar un calendario en conjunto para tener horarios en común y	Reducir el alcance, pedir extensión de plazos, reasignar tareas.

				establecer prioridades.	
Comunicación deficiente del equipo		Alto	Baja	Implementación de reuniones semanales y centralizar todo tipo de documentación en un solo punto.	Realizar unas sesiones de retroalimentación preentrega para resolver los problemas de forma rápida y eficiente. Tener una comunicación más frecuente con el patrocinador para evitar demoras y cambios sorpresa.
Integración fallida de archivos del patrocinador		Muy Alto	Media	Adaptar el entorno de la solución para que este sea compatible con los archivos entregados directamente por el patrocinador.	Evaluar la posibilidad de cambiar el formato del archivo entregado por el patrocinador a uno que tenga compatibilidad natural con el sistema.
Fallas técnicas durante la Feria de Software		Alto	Baja	Tener la solución web en nuestros computadores personales, y en nuestros celulares para la aplicación web	Dejar una copia de seguridad funcional en caso de una falla crítica
Entregar el prototipo sin terminar		Alto	Media	Ordenar las prioridades del equipo para un desarrollo eficiente, considerando también el tiempo estimado de finalización de cada iteración.	Pedir una extensión de plazo a nuestro patrocinador para hacer efectiva la entrega.

Tabla 3 Gestión de Riesgos

7. Evaluación del Desempeño Ágil

7.1 Métricas utilizadas para medir la eficiencia y efectividad.

Para poder medir la eficiencia y la efectividad a lo largo del desarrollo del proyecto, se usará el cumplimiento de objetivos de cada sprint como métrica para poder hacer una medición efectiva. Para esto, se hará uso de la tabla 1, que demuestra cada sprint con 3 objetivos específicos.

Para el Sprint 1, los objetivos son:

- Igualar el prototipo generado en figma
- Integrar la API de Google Maps

- Mostrar las cuadrillas en terreno y sus distintos estados.

A continuación, se presentarán las ilustraciones 3 y 4, que harán referencia al modelo realizado por figma y al entregado en el sprint 1 respectivamente, para hacer comparaciones.



Ilustración 3: Prototipo creado en figma

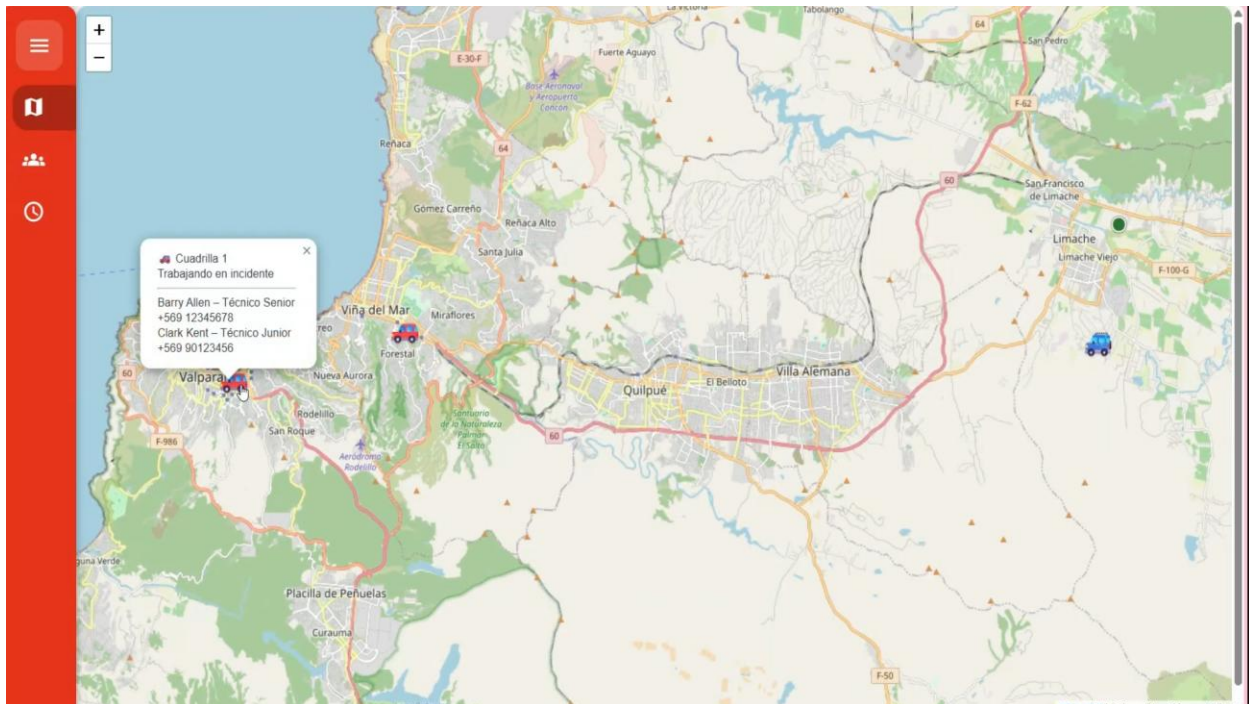


Ilustración 4: Prototipo entregado del Sprint 1

Para el Sprint 2, los objetivos son:

- Hacer el envío de coordenadas de la aplicación móvil a un servidor privado.
- Terminar la integración de Google Maps API
- Integrar Archivos.kmz otorgados por el patrocinador.

La ilustración 5, corresponde a un método de testeo para el envío de coordenadas hacia el servidor privado, donde el lado izquierdo es quien manda el mensaje y el lado derecho muestra el contenido enviado, logrando así confirmar que se reciben las coordenadas. Por otra parte, la ilustración 6 sirve para evidenciar algunos puntos de interés de nuestro patrocinador reflejados en la aplicación, con mejoras en la visualización de menús y en la muestra de funcionalidades.

```

+ test_scripts git(master) * python3 test-rabbitmq-message.py
Publishing location updates to RabbitMQ...
/Users/robertoromero/RiderProjects/Geocudrilla-2/test_scripts/test-rabbitmq-message.py:20:
DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal in
a future version. Use timezone-aware objects to represent datetimes in UTC: datetime.datetime
me.now(datetime.UTC).
"Timestamp": datetime.datetime.utcnow().isoformat() + "Z",
[1] Published: CREW-001 at (40.7128, -74.006)
[2] Published: CREW-002 at (35.0522, -118.2437)
[3] Published: CREW-003 at (41.8781, -87.6298)
[4] Published: CREW-001 at (40.7589, -73.9851)
[5] Published: CREW-004 at (29.7604, -95.3698)

Successfully published 5 location updates!
Check the API logs to see if they're being processed correctly.
+ test_scripts git(master) *

} Listening for updates for up to 30 seconds...
Update: {
  "crewId": "CREW-001",
  "timestamp": "2025-09-25T23:10:53.2621Z",
  "latitude": 40.7128,
  "longitude": -74.006,
  "isOnline": true,
  "calculatedAt": "2025-09-25T23:10:53.279891Z"
}
Update: {
  "crewId": "CREW-002",
  "timestamp": "2025-09-25T23:10:53.262946Z",
  "latitude": 35.0522,
  "longitude": -118.2437,
  "isOnline": true,
  "calculatedAt": "2025-09-25T23:10:53.292779Z"
}
Update: {
  "crewId": "CREW-003",
  "timestamp": "2025-09-25T23:10:53.263039Z",
  "latitude": 41.8781,
  "longitude": -87.6298,
  "isOnline": true,
  "calculatedAt": "2025-09-25T23:10:53.298851Z"
}
Update: {
  "crewId": "CREW-001",
  "timestamp": "2025-09-25T23:10:53.263113Z",
  "latitude": 40.7589,
  "longitude": -73.9851,
  "isOnline": true,
  "calculatedAt": "2025-09-25T23:10:53.312588Z"
}
Update: {
  "crewId": "CREW-004",
  "timestamp": "2025-09-25T23:10:53.263186Z",
  "latitude": 29.7604,
  "longitude": -95.3698,
  "isOnline": true,
  "calculatedAt": "2025-09-25T23:10:53.319868Z"
}
}

```

Ilustración 5: Funcionamiento de coordenadas hacia servidor privado

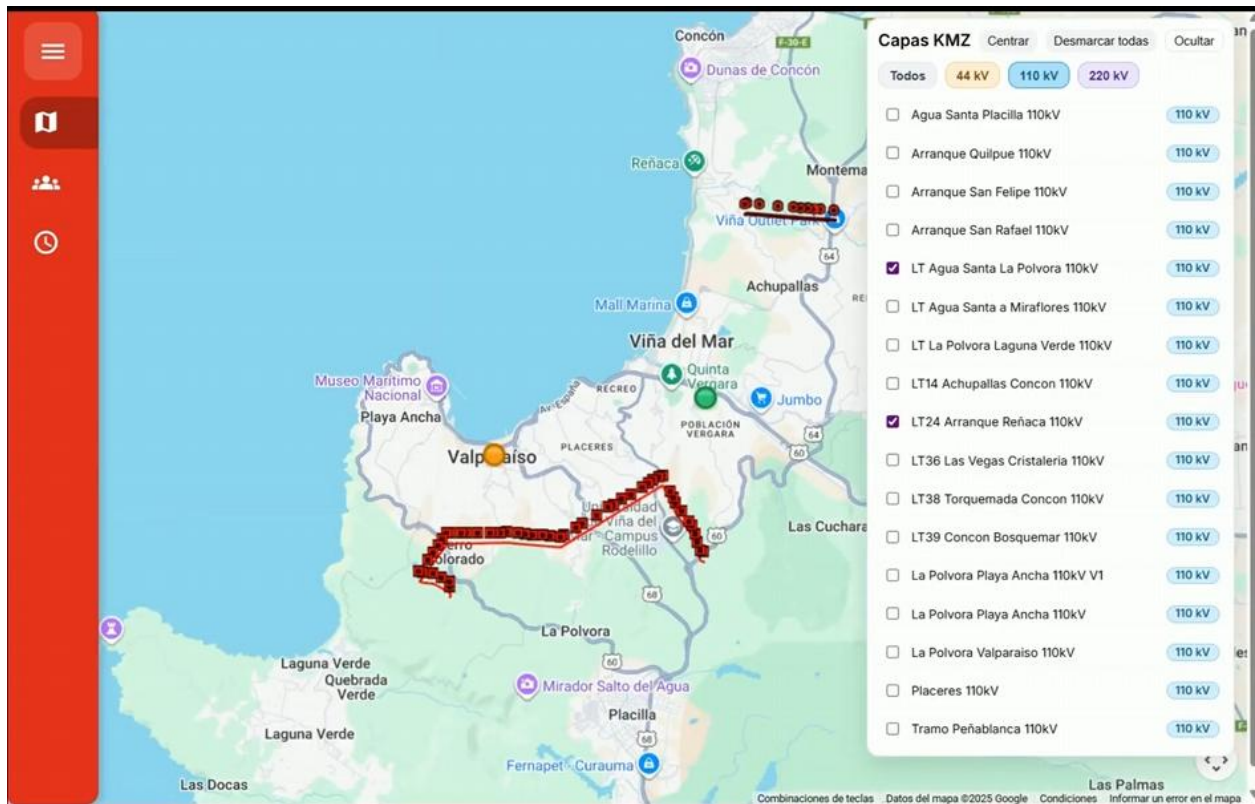


Ilustración 6: Funcionamiento de archivos.kmz y mejoras en la visualización de cuadrillas, con el uso de Google Maps API

Para el Sprint 3, se definieron los siguientes objetivos:

- Tener tanto la aplicación móvil como el panel web funcionando en un servidor privado.
- Demostrar funcionamiento de la ubicación en tiempo real.
- Añadir opción de agregar archivos.kmz de forma natural.

Para este caso, se usarán 2 imágenes de visualización de la aplicación para demostrar el cambio de ubicación de forma activa, si bien en el video de la entrega no aparece la opción de agregar los archivos.kmz, esto fue añadido previo a la feria de software, y mostrado posteriormente al patrocinador quién se mostró contento con la implementación.

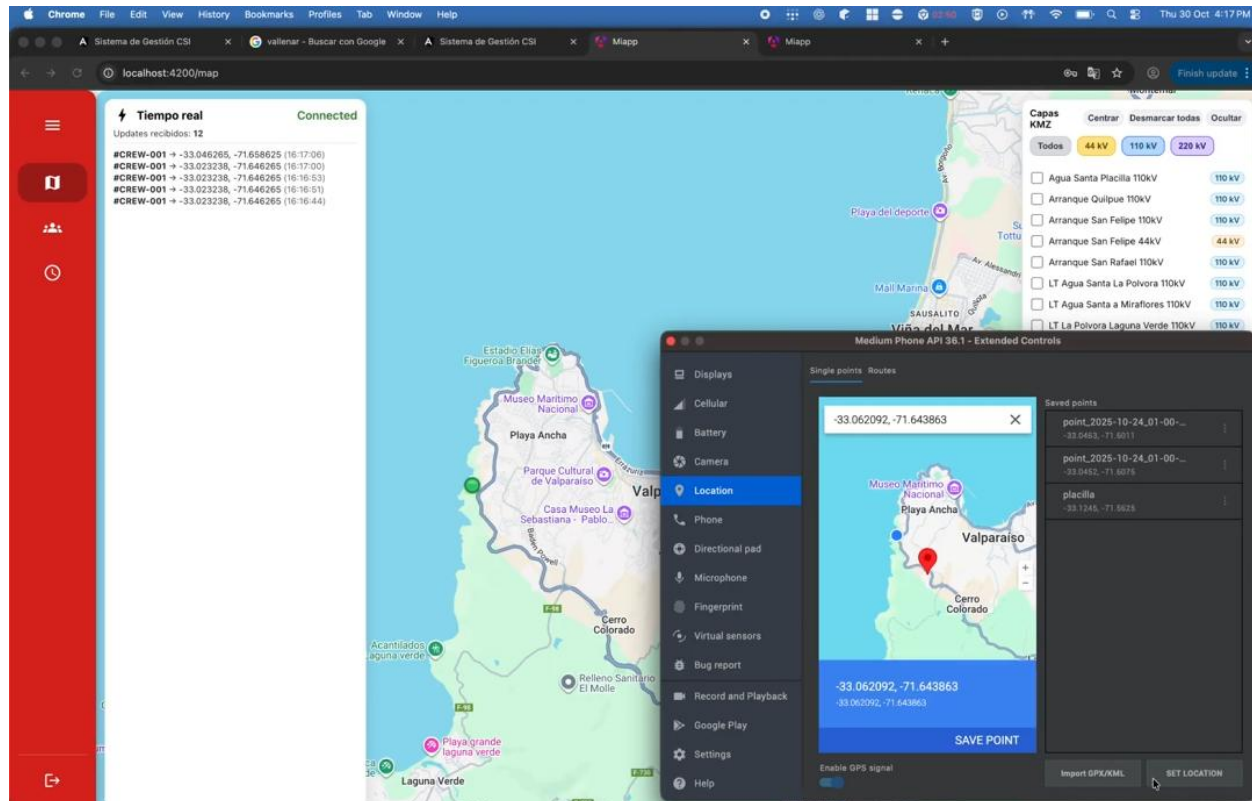


Ilustración 7: Demostración de funcionamiento de ubicación en tiempo real

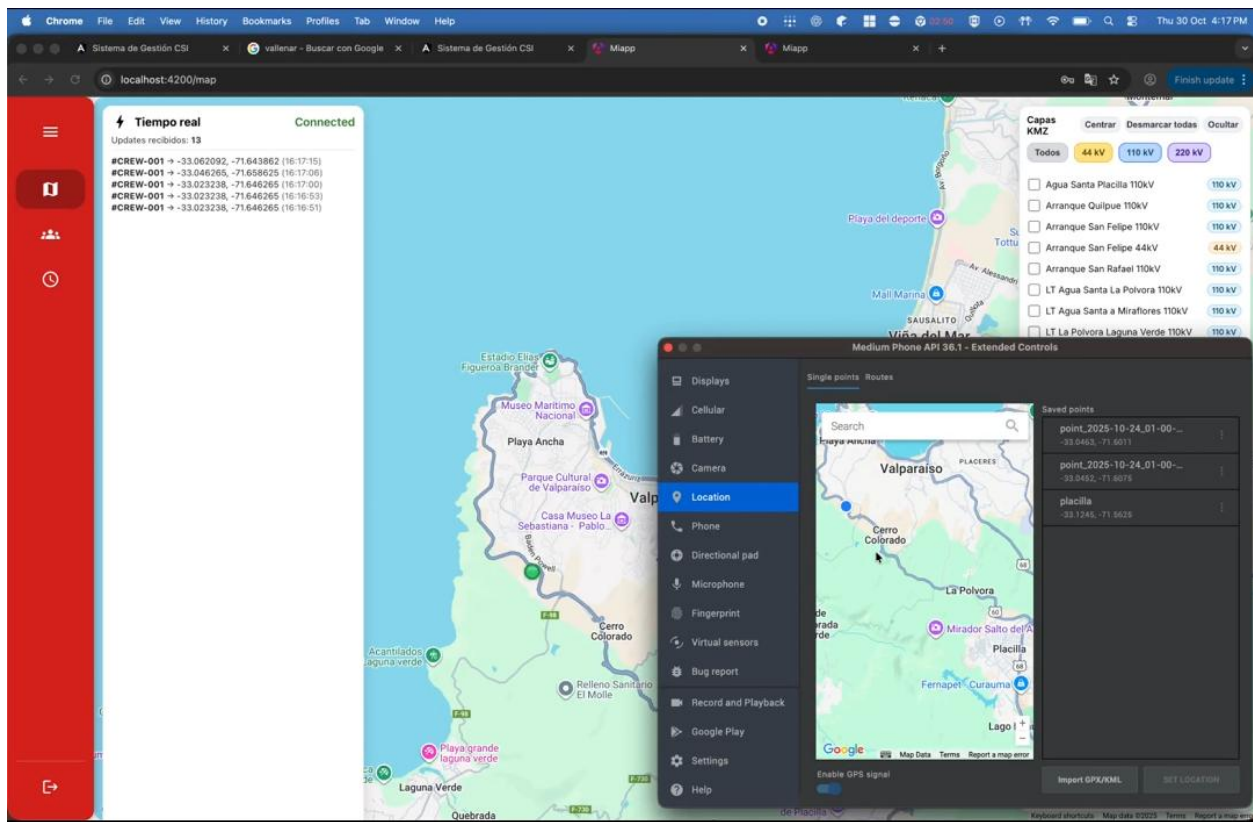


Ilustración 8: Confirmación de funcionamiento con el cambio de lugar.

Para evaluar el desempeño general de todo el proceso de creación de la solución, usaremos las métricas ágiles de Velocity y Lead Time para contextualizar bien el proceso de desarrollo del producto. Para el caso de Velocity, en vez de usar Historias de Usuario se hará uso de una medición basada en los objetivos completados por sprint.

En el caso de Velocity, la tabla quedaría de la siguiente manera:

Sprint	Objetivos planificados	Objetivos completados	Velocity (%)
Sprint 1	3	2	66,7 %
Sprint 2	3	3	100 %
Sprint 3	3	2	66,7 %

Tabla 4: Métrica Ágil Velocity

La velocidad del equipo fue estable a lo largo del desarrollo del proyecto, donde tuvo su punto más alto en el Sprint 2 logrando un 100% de los objetivos establecidos, mientras que en el Sprint 1 y 3 se vió un 66,7% de cumplimiento, esto principalmente se debe a la curva de aprendizaje para poder integrar Google Maps API y a cambios solicitados por nuestro patrocinador fuera del alcance inicialmente planificado, más que a

problemas de organización del equipo como tal. Estos resultados hacen ver una mejora progresiva en la capacidad de estimación y planificación durante el desarrollo del proyecto.

Para el caso de la métrica Lead Time se tomarán las fechas de cada sprint estipulado, junto con las funcionalidades críticas de la aplicación, para evidenciar cómo se dio el proceso y si realmente hubo buenos resultados.

Por lo tanto, la tabla de Lead time quedaría de la siguiente forma:

Funcionalidad	Sprint	Estado
Panel Web	Sprint 1	Completada
Google Maps API	Sprint 2	Completada
Servidor privado	Sprint 2	Completada
Aplicación móvil	Sprint 3	Completada
Carga natural de KMZ	Sprint 3	Postergada

Tabla 5: Métrica Ágil Lead Time

Como se puede evidenciar en la tabla, las funcionalidades críticas fueron entregadas dentro del sprint planificado, lo cual fue la mayor prioridad de todo el proceso de desarrollo del producto, mientras que la carga natural de los archivos KMZ fue una implementación hecha posterior al Sprint 3, siendo un requerimiento del patrocinador después de la evaluación de la asignatura.

7.2 Análisis de resultados obtenidos

- Cumplimiento de objetivos:
 - Sprint 1: Se cumplieron 2 de 3 objetivos (66%)
 - Sprint 2: Se cumplieron 3 de 3 objetivos (100%)
 - Sprint 3 Se cumplieron 2 de 3 objetivos (66%)
- Incumplimiento de plazos:
 - Integración inicial de Google Maps API:
 - Por problemas de tiempo y de conocimiento para poder usar esta API, se decidió usar OpenStreetMap como solución parcial.
 - Añadir opción de agregar archivos.kmz de forma natural:
 - Esto no se logró cumplir por cuestiones de tiempo, ya que el patrocinador pidió esta funcionalidad posterior a la entrega final del sprint 3, por lo que se hizo esta mejora fuera del plazo estipulado.
- Satisfacción del cliente:

Las distintas reuniones presenciales efectuadas a lo largo del semestre con el equipo de Chilquinta Transmisión y el buen recibimiento del público como del jurado de la feria de software,

evidenció que los objetivos del proyecto fueron cumplidos y satisfacen las necesidades tanto del cliente como de nuestro público objetivo.

La satisfacción del patrocinador fue respaldada mediante la validación realizada en las reuniones de revisión de Sprint, y por la difusión oficial efectuada por Chilquinta en su cuenta institucional de Instagram, donde se presentó de manera pública el sistema GeoCuadrilla y su funcionamiento. Esta publicación constituye evidencia del interés de la organización en la solución desarrollada.



Ilustración 9: Publicación Geocuadrilla x Chilquinta

8. Desafíos y Soluciones

8.1 Obstáculos Identificados.

Los principales problemas que aparecieron a lo largo del proyecto fueron los siguientes:

- Problemas de organización: Los integrantes de distintas áreas no tenían claridad de cómo demostrar los cambios hechos al proyecto, lo que entorpecía el progreso al inicio del desarrollo.
- Falta de tiempo: Todos los integrantes del grupo tienen responsabilidades académicas adicionales. Al ser alumnos de la Universidad Técnica Federico Santa María, es imposible dedicarnos sólo al proyecto, forzando a cada integrante del grupo a organizarse con respecto a los distintos tiempos de obligaciones variadas, para usar parte del tiempo libre en el desarrollo del proyecto.

8.2 Estrategias para solucionar los problemas

Para los problemas planteados anteriormente, se plantearon las siguientes soluciones:

- Estructurar la plataforma de reuniones: Usando Discord, se dividió el grupo en roles, logrando así que cada uno pueda ver lo que tiene que hacer puntualmente, y con todos los respaldos necesarios para hacer efectivo el trabajo.
- Gestión del tiempo: Se uso la página de <https://www.when2meet.com/> para poder revisar los horarios que todos los integrantes del grupo tenían libres, y en base a eso agendar todas las reuniones necesarias a lo largo del proyecto.

9. Estimación de costos.

La estimación de costos refleja el trabajo del desarrollo de software, donde la mano de obra especializada constituye el principal componente del presupuesto. Los costos asociados a infraestructura, licencias y servicios complementarios representan una proporción menor pero necesaria para el correcto funcionamiento del sistema durante su desarrollo.

Categoría	Elemento	Descripción	Costo Unitario (CLP)	Cantidad de horas	Costo total (CLP)
Desarrollo de software	Panel web con Google Maps API	Desarrollo y pruebas de panel web + Integración API	12.000	350	4.200.000
Desarrollo de software	Aplicación Móvil	Aplicación móvil + servicios de autenticación y ubicación	12.000	300	3.600.000
Desarrollo de software	Integración base de datos	Integración de datos del patrocinador con el panel web	12.000	350	4.200.000
Licencias de software	Servicios Cloud/APIs	Costos proyectados mínimos al momento de usar el producto	200.000	1	200.000
Hosting	Servidor para pruebas y producción	Hosting durante 6 meses	15.000	6	90.000

Otros costos	Electricidad y conectividad	Costo aproximado 6 meses de trabajo	50.000	6	300.000
Total					12.590.000

Tabla 6 Estimación de costos

El costo de mano de obra total sería de 12.000.000 CLP, esto sin considerar el pago de un servidor privado, ya que, dependiendo del cliente, es un gasto que ya se está efectuando, en nuestro caso, el servidor de prueba que utilizamos correspondía a un computador personal de los integrantes, así que no hubo gastos adicionales.

10. Conclusiones

El desarrollo del proyecto GeoCuadrilla permitió abordar una problemática relevante para la operación de cuadrillas en terreno dentro del sector de transmisión de energía, ofreciendo una solución capaz de georreferenciar el personal en terreno, monitorear su actividad en tiempo real y facilitar la toma de decisiones operacionales.

El enfoque metodológico basado en Scrum fue determinante para la organización del trabajo y el proceso iterativo de la solución. La planificación mediante sprints, la definición de objetivos parciales y la entrega continua de incrementos funcionales permitieron validar tempranamente avances, corregir desviaciones y mantener una visión clara del producto final. Esta forma de trabajo facilitó la adaptación ante cambios en requerimientos, especialmente aquellos asociados al patrocinador, reflejando la flexibilidad necesaria en proyectos de software donde las necesidades evolucionan de manera constante. (Objetivo específico 1 ,3 y 5)

Desde el punto de vista técnico, el proyecto requirió el desarrollo de un panel web con la integración de mapas, un servidor privado para comunicación entre módulos web y móvil, una aplicación móvil para la captura de ubicación y la integración de una base de datos del patrocinador. Estas tareas representaron desafíos relacionados con la complejidad tecnológica, funcionamiento entre sistemas y rendimiento en tiempo real. Sin embargo, la descomposición del trabajo en funcionalidades manejables, el uso de herramientas colaborativas y la validación progresiva permitieron alcanzar resultados satisfactorios.

En términos de la gestión de este proyecto, la comunicación y la colaboración del equipo resultaron fundamentales para el desempeño a lo largo del proyecto. Las reuniones periódicas, la documentación ligera, el feedback constante y la claridad en roles contribuyeron a sostener un flujo de trabajo eficiente, reducir las dudas y favorecer un ambiente de aprendizaje. Las retrospectivas, por su parte, fomentaron la mejora continua e incentivaron la revisión de procesos y prácticas críticas. (Objetivo específico 2)

El análisis de costos evidenció que el desarrollo del sistema implica una inversión mayoritariamente asociada a mano de obra especializada, lo cual es consistente con proyectos de software de este tipo. Asimismo, los costos operaciones y de infraestructura fueron estimados de manera proyectiva, evidenciando que, en un escenario de despliegue real, serían necesarios recursos adicionales para asegurar estabilidad, escalabilidad y seguridad del sistema.

A nivel de resultados, el proyecto logró materializar una plataforma funcional que integra datos en tiempo real, visualización geográfica y comunicación entre módulos, alcanzando los principales objetivos planteados. No obstante, la solución actual se posiciona como un prototipo robusto y aceptable, con el potencial para evolucionar hacia un producto comercial o institucional. (Objetivo específico 4)

Para eso, sería importante tomar en cuenta las siguientes recomendaciones:

- Implementar mecanismos avanzados de seguridad y privacidad: Dado que el sistema manipula información sensible entre ubicación y trabajo en terreno, es recomendable integrar protocolos de cifrado de datos, autenticación multifactor, control de acceso y políticas de retención de datos conforme a normativas legales y estándares industriales.
- Integrar algoritmos de optimización y análisis predictivo: El sistema podría beneficiarse de modelos que sugieran rutas óptimas, prioricen tareas, o anticipen eventos críticos, mediante técnicas de machine learning, agregando valor estratégico al monitoreo en terreno.
- Realizar estudios comparativos y validaciones en terreno: Para determinar el impacto real de GeoCuadrilla, sería valioso realizar pilotos controlados, mediciones cuantitativas y comparaciones antes/después, evaluando mejoras en tiempos de respuesta, costos de operación y satisfacción de usuarios.

Esto busca proyectar el desarrollo de GeoCuadrilla más allá del prototipo funcional desarrollado, orientándolo hacia un producto seguro, escalable y alineado a los desafíos tecnológicos y operacionales del sector energético. La implementación de estas mejoras permitiría no solo optimizar el desempeño técnico del sistema, sino también potenciar su valor estratégico y su capacidad de impacto en entornos reales.

Finalmente, la experiencia obtenida a través del desarrollo de GeoCuadrilla permitió comprender la importancia de aplicar metodologías ágiles en proyectos tecnológicos complejos, destacando su capacidad para mejorar la gestión del tiempo, fomentar el aprendizaje continuo y aumentar la satisfacción del cliente. A su vez, demostró que la integración de herramientas tecnológicas, buenas prácticas de desarrollo, y una gestión eficiente del equipo son factores clave para el éxito en el diseño de soluciones innovadoras en contextos organizacionales dinámicos.

Bibliografía

- Agile Alliance. (2001). Manifiesto for Agile Software Development. <https://agilemanifesto.org/>
- Beck, K., & Andres, C. (2004). Extreme Programming Explained: Embrace Change (2.a ed.). Addison-Wesley.
- Highsmith, J. (2009). Agile Project Management: Creating Innovative Products. Addison-Wesley Professional.
- Instituto Europeo de Posgrado. (2022). Gestión de proyectos: Enfoques tradicionales y ágiles. <https://iep.edu.es/>
- Kohn, B. (2010). Kanban: Successful Evolutionary Change for Your Technology Business. Blue Hole Press.
- Project Management Institute. (2021). A Guide to the Project Management Body of Knowledge (PMBOK® Guide) (7.a ed.). PMI.
- Schwaber, K., & Sutherland, J. (2020). The Scrum Guide: The Definitive Guide to Scrum. Scrum.org.
- Google Developers. (2024). Maps JavaScript API Documentation. <https://developers.google.com/maps/>
- OpenStreetMap Foundation. (2024). OpenStreetMap Documentation. <https://www.openstreetmap.org/>
- RabbitMQ Team. (2024). RabbitMQ Documentation. <https://www.rabbitmq.com/documentation.html>