



UNIVERSIDAD DE AYSÉN
DEPARTAMENTO DE CIENCIAS NATURALES Y TECNOLOGÍA
INGENIERÍA CIVIL INFORMÁTICA

Informe de Avance 2

Taller Avanzado de Desarrollo de Software

9 de mayo de 2025

1. Datos Generales

- **Nombre del Proyecto:** Gestión Y Organización para la Unidad de Logística y Servicios Generales.
- **Integrantes del Equipo:** Francisco Marió Chiguay
- **Nombre del Socio Comunitario:** Rene Peña Ulloa
- **Fecha de Entrega:** 9 de mayo de 2025
- **Repositorio del Proyecto:** <https://github.com/Pixelardo/Logistica---Uaysen>

2. Selección y Justificación de la Metodología

Para el desarrollo de este proyecto se ha optado por la metodología ágil **Scrum**, debido a su enfoque iterativo e incremental, el cual se adapta eficazmente a las condiciones de un entorno académico y colaborativo con un socio comunitario. Scrum permite dividir el proyecto en *sprints*, lo que facilita la entrega continua de valor, así como la validación frecuente por parte de los usuarios involucrados.

Considerando que el equipo de desarrollo está conformado por estudiantes con diversas responsabilidades académicas, esta metodología permite organizar el trabajo de manera flexible y efectiva. Scrum promueve la comunicación constante mediante reuniones breves y enfocadas (como las *Reuniones Periódicas*, revisiones de avance y retrospectivas), lo que resulta ideal en contextos donde el tiempo y los recursos pueden ser limitados.

Además, a partir de experiencias anteriores en proyectos similares en el desarrollo de plataformas web haciendo el uso de **Scrum** ha demostrado ser una herramienta útil para mejorar la gestión del tiempo, la coordinación del equipo y la calidad del producto final. Esta metodología fomenta la autonomía, la autoorganización y la responsabilidad compartida, valores fundamentales para un trabajo eficiente en equipo.

2.1 Aplicación de Scrum en el Proyecto

La adaptación de Scrum se realizará de forma progresiva, comenzando con una estructura más simple de reuniones y roles, para posteriormente ir profundizando en la formalización de los sprints y las actividades de

seguimiento. La flexibilidad y la continua evaluación permitirán que el proceso se ajuste según los resultados y la evolución del proyecto.

- **Duración de los sprints:** Cada sprint tendrá una duración de una semana, permitiendo avances rápidos y revisiones frecuentes.
- **Reuniones Periódicas** Se realizarán reuniones con semanas de anticipación, con el fin de alinear al equipo, resolver bloqueos y mantener el flujo de trabajo constante con los requerimientos del Socio Comunitario.
- **Roles definidos:** Se asignará un *Scrum Master* (El alumno encargado de desarrollar el proyecto), encargado de facilitar el proceso y eliminar obstáculos, y un *Product Owner*, que en este caso es el socio comunitario y validará los avances.
- **Backlog del producto:** Se mantendrá un listado de tareas priorizadas, construido y refinado en conjunto con el Product Owner, para guiar el desarrollo según las necesidades del proyecto.
- **Revisión y retrospectiva:** Al término de cada sprint, se realizará una revisión del entregable y una retrospectiva interna para identificar aprendizajes y oportunidades de mejora en el proceso.

3. Captura y Especificación de Requerimientos

3.1 Requerimientos Funcionales

- RF01: El sistema debe permitir registrar y modificar rutas de transporte.
- RF02: El sistema debe permitir registrar horarios de salida y llegada.
- RF03: El sistema debe permitir la asignación de vehículos a rutas específicas.
- RF04: El sistema debe mostrar en tiempo real el estado de las unidades (activo/inactivo, Ocupado/Reservado/Disponibile, etc).
- RF05: El sistema debe generar reportes mensuales de uso de transporte.
- RF06: El sistema debe permitir al usuario consultar rutas disponibles.
- RF07: El sistema debe permitir al administrador agregar y editar vehículos.
- RF08: El sistema debe permitir al usuario modificar y/o cancelar una solicitud.

3.2 Requerimientos No Funcionales

- RNF01: El sistema debe estar disponible al menos el 99 % del tiempo.
- RNF02: El tiempo de respuesta ante consultas no debe superar los 2 segundos.
- RNF03: El diseño de la interfaz debe de ser **responsive**.
- RNF04: Los datos deben ser respaldados automáticamente cada 24 horas.
- RNF05: El sistema debe cumplir con prácticas de seguridad recomendadas por la **Política general de seguridad de la información de la Universidad de Aysén (368)**.¹
- RNF06: El sistema debe ofrecer una experiencia intuitiva y accesible para el usuario.

¹Este documento establece directrices y principios para proteger los activos de información institucionales, definiendo responsabilidades, ámbitos de aplicación, objetivos y un marco normativo para asegurar la confidencialidad, integridad y disponibilidad de la información.

4. Diagrama

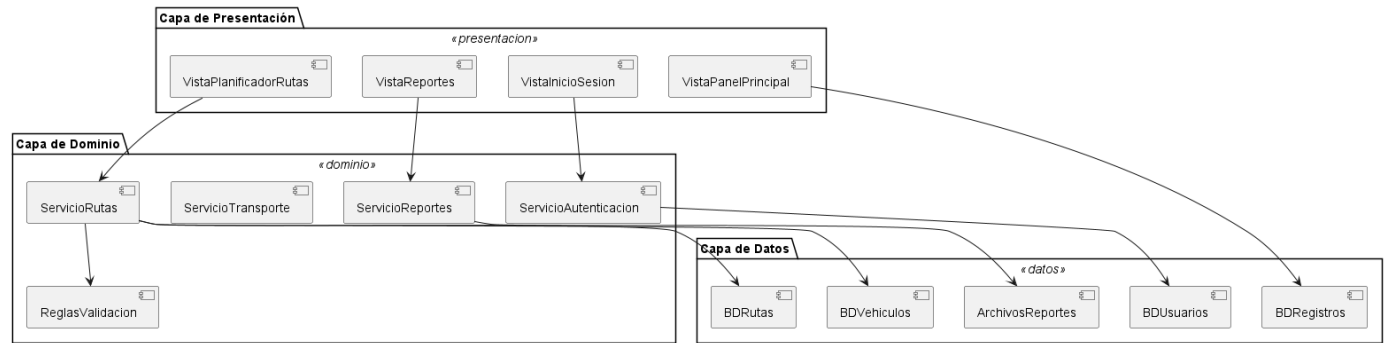


Figura 1: Diagrama de Arquitectura de Sistemas

4.1 Descripción Técnica del Diagrama de Arquitectura

El sistema propuesto se organiza en tres capas principales, siguiendo un enfoque de **arquitectura en capas**, lo que permite estructurar el desarrollo de forma ordenada, mantenible y escalable.

1. Capa de Presentación

Esta capa representa la interfaz con el usuario, es decir, todo lo que el usuario ve y con lo que interactúa directamente. Su objetivo es facilitar una experiencia de uso clara y funcional. Cada una de estas vistas se comunica con los servicios de la siguiente capa para enviar solicitudes y recibir respuestas.

- **VistaInicioSesion:** Permite a los usuarios ingresar al sistema mediante credenciales.
- **VistaPanelPrincipal:** Funciona como un menú central para acceder a las funcionalidades.
- **VistaPlanificadorRutas:** Interfaz para visualizar, crear o modificar rutas de transporte.
- **VistaReportes:** Espacio destinado a la consulta y generación de reportes.

2. Capa de Dominio

Aquí reside la **lógica del negocio**, es decir, las reglas que definen cómo debe funcionar el sistema. Esta capa se encarga de procesar los datos recibidos desde la presentación y preparar la información antes de almacenarla o recuperarla. Finalmente actúa como puente entre la presentación y los datos, garantizando que todo fluya correctamente y según lo esperado.

- **ServicioAutenticacion:** Valida credenciales y gestiona el acceso de los usuarios.
- **ServicioRutas:** Administra todo lo relacionado con la planificación y edición de rutas.
- **ServicioTransporte:** Gestiona los vehículos, su estado y disponibilidad.
- **ServicioReportes:** Genera informes según los datos almacenados.
- **ReglasValidacion:** Aplica controles para garantizar que los datos cumplan con los criterios del sistema.

3. Capa de Datos

En esta última capa se almacenan todos los datos del sistema. Cada componente aquí representa una base de datos o repositorio específico. La comunicación con esta capa está restringida a los servicios de la capa de dominio, lo que refuerza la seguridad y el orden estructural del sistema.

- **BDUsuarios:** Guarda la información de los usuarios del sistema.
- **BDRutas:** Contiene los datos de las rutas creadas o planificadas.
- **BDVehiculos:** Almacena detalles de los vehículos disponibles.
- **BDRregistros:** Historial de acciones y eventos del sistema.
- **ArchivosReportes:** Documentos generados a partir de reportes solicitados.

5. Análisis de Riesgos

5.1 Identificación de Riesgos

Cambio en los requerimientos por parte del socio comunitario: En muchos proyectos reales, los usuarios finales ajustan sus expectativas o descubren nuevas necesidades durante el desarrollo. Dado que el sistema se basa en las rutas de transporte de una comunidad, podrían surgir nuevas prioridades, modificaciones o solicitudes adicionales que alteren el alcance original. Este riesgo es especialmente probable cuando no se han documentado formalmente todos los requerimientos desde el inicio.

Baja disponibilidad del equipo de desarrollo: Los integrantes del equipo son estudiantes con múltiples asignaturas y responsabilidades personales. Esto puede llevar a periodos de baja productividad o retrasos en entregas si no se gestiona bien la carga de trabajo. Por ello, es fundamental planificar las tareas considerando la disponibilidad real del equipo.

Falta de capacitación de usuarios finales: Existe el riesgo de que los usuarios del sistema, como choferes o encargados del transporte, no cuenten con la capacitación adecuada para operar o interpretar correctamente la solución implementada. Esto puede limitar el impacto positivo del proyecto o requerir tiempo adicional para realizar sesiones de entrenamiento.

Fallas o daños en los medios de transporte: Si bien no es directamente responsabilidad del software, es importante considerar que la implementación del sistema depende de la disponibilidad funcional de los medios de transporte. Fallas mecánicas o deterioro del parque vehicular pueden afectar la planificación y seguimiento que realiza la herramienta, impactando su utilidad en la práctica.

Fallos técnicos o problemas con la infraestructura: Errores de conexión a la base de datos, incompatibilidades de software, mal funcionamiento del servidor web o pérdida de información pueden retrasar significativamente el desarrollo. Este riesgo también contempla posibles problemas derivados del uso de herramientas como XAMPP, que si no se configuran correctamente, pueden fallar en producción o prueba.

5.2 Matriz de Evaluación de Riesgos

Riesgo	Probabilidad	Impacto	Descripción
Cambio de requerimientos	Alta	Alta	Cambios inesperados desde el socio comunitario que pueden generar retrabajo y afectar los plazos establecidos.
Baja disponibilidad del equipo	Media	Alta	Otras asignaturas y responsabilidades académicas que pueden limitar la dedicación al proyecto.
Falta de capacitación de usuarios finales	Media	Media	Los encargados de operar el sistema pueden no tener la formación necesaria, lo que afectaría su uso correcto y la efectividad del sistema.
Fallas o daños en los medios de transporte	Baja	Alta	El deterioro o mal estado de los vehículos puede impedir que el sistema planifique o registre rutas de forma efectiva, reduciendo su utilidad.
Fallos técnicos	Media	Media	Problemas en la conexión a la base de datos o mal funcionamiento del sistema pueden dificultar su uso o afectar la disponibilidad de la información.

Cuadro 1: Matriz de Evaluación de Riesgos

5.3 Estrategias de Mitigación

- **Reuniones periódicas con el socio comunitario:** Se establecerá una comunicación con el Socio Comunitario para agendar de reuniones de seguimiento para validar los requerimientos, detectar cambios oportunamente y asegurar una comunicación clara y continua.
- **Asignación de tareas según la disponibilidad del equipo:** Las actividades se distribuirán considerando la carga académica y el tiempo real de los integrantes, utilizando herramientas colaborativas como **GitHub** (En este caso el repositorio creado para el desarrollo del proyecto) para mantener la organización y el seguimiento del trabajo.
- **Verificación cruzada de información:** Se emplearán múltiples fuentes para validar datos críticos, como rutas de transporte y horarios, asegurando así la consistencia y confiabilidad del sistema.
- **Pruebas frecuentes y control de versiones:** Se realizarán pruebas continuas durante los sprints para detectar errores de forma temprana, junto con el uso de control de versiones (GitHub) para asegurar trazabilidad y respaldo del código.
- **Capacitación básica para los usuarios finales:** Se elaborará una guía de uso del sistema y se organizarán sesiones de capacitación (presenciales o virtuales) para garantizar que los encargados puedan utilizar correctamente la herramienta.
- **Evaluación técnica de los medios de transporte:** Se recopilará información sobre el estado de los vehículos y se considerarán alternativas en caso de fallas (por ejemplo, ajustes automáticos en la planificación si se detecta inoperatividad en alguna ruta).
- **Flexibilidad en la planificación:** El cronograma del proyecto incluye márgenes para adaptarse a imprevistos y se actualizará de forma iterativa en función del progreso real y las validaciones con los usuarios.

6. Cronograma Actualizado

Actividad	Inicio	Término
Captura de Requerimientos	01-04-2025	10-04-2025
Validación con Usuario	11-04-2025	15-04-2025
Análisis de Riesgos	16-04-2025	18-04-2025
Actualización del Cronograma	19-04-2025	20-04-2025
Desarrollo Sprint 1	21-04-2025	28-04-2025
Desarrollo Sprint 2	29-04-2025	03-05-2025
Desarrollo Sprint 3 y 4	04-05-2025	07-05-2025

Cuadro 2: Cronograma de Diseño y Validación

7. Evidencias y Anexos

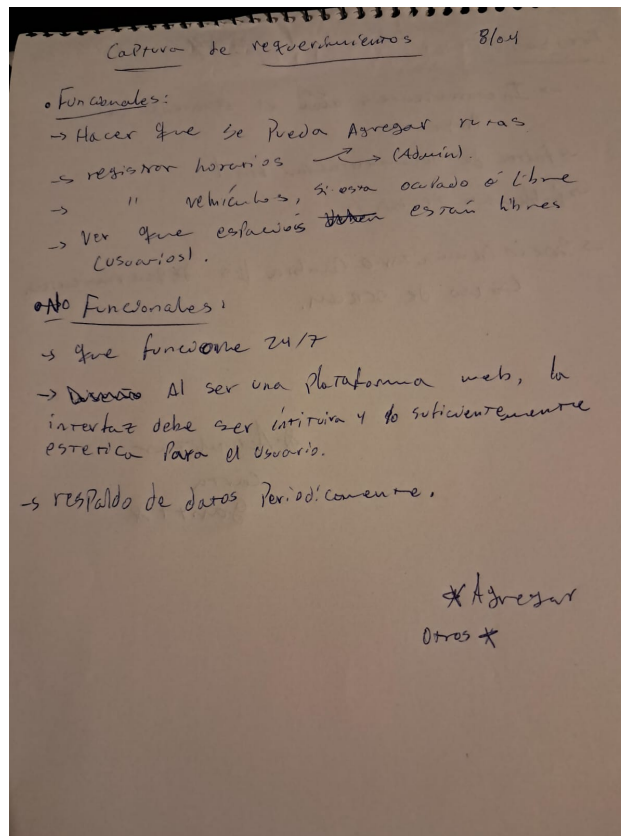


Figura 2: Borrador de captura de requerimientos funcionales y no funcionales

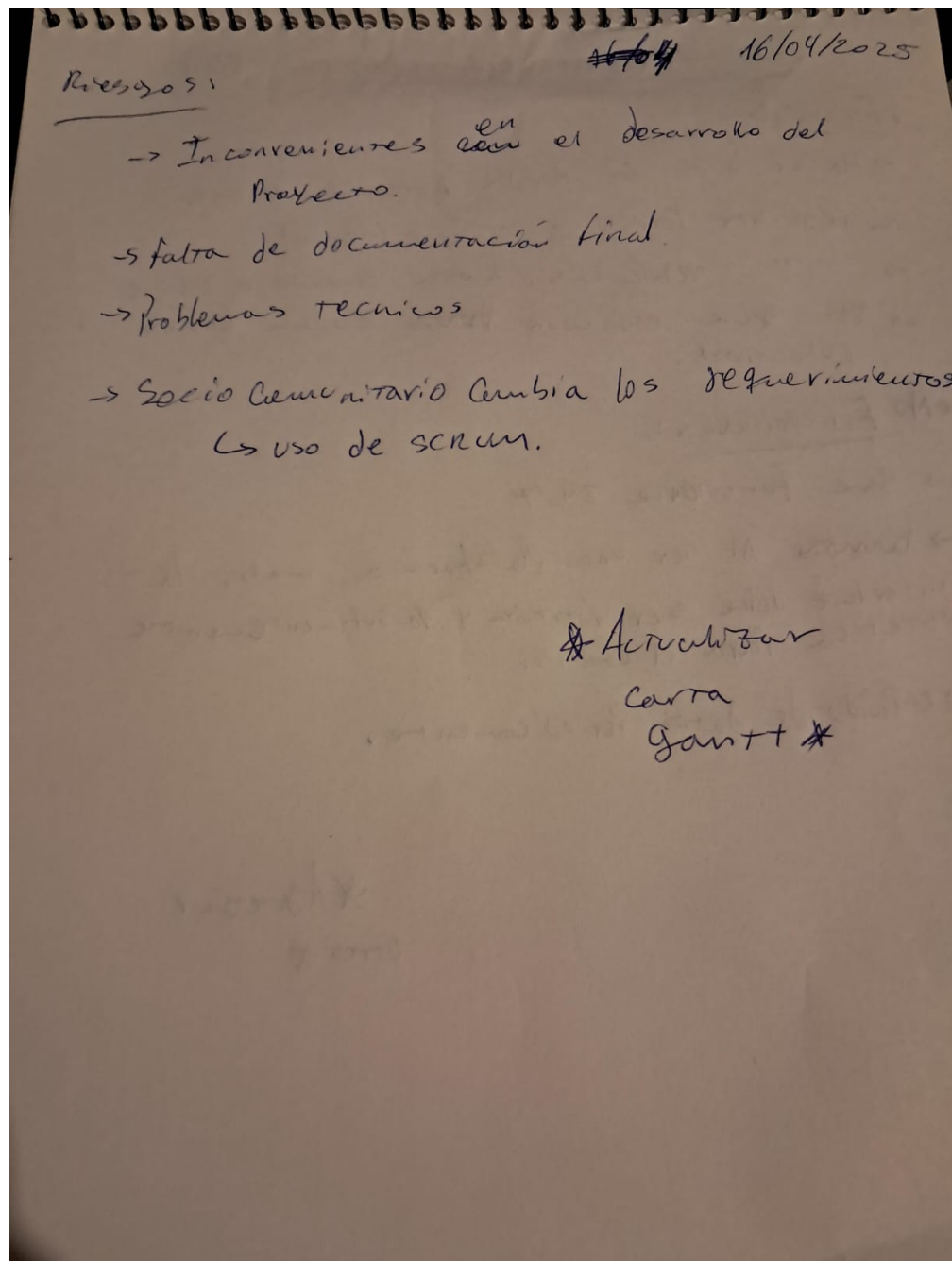


Figura 3: Captura de identificación inicial de riesgos

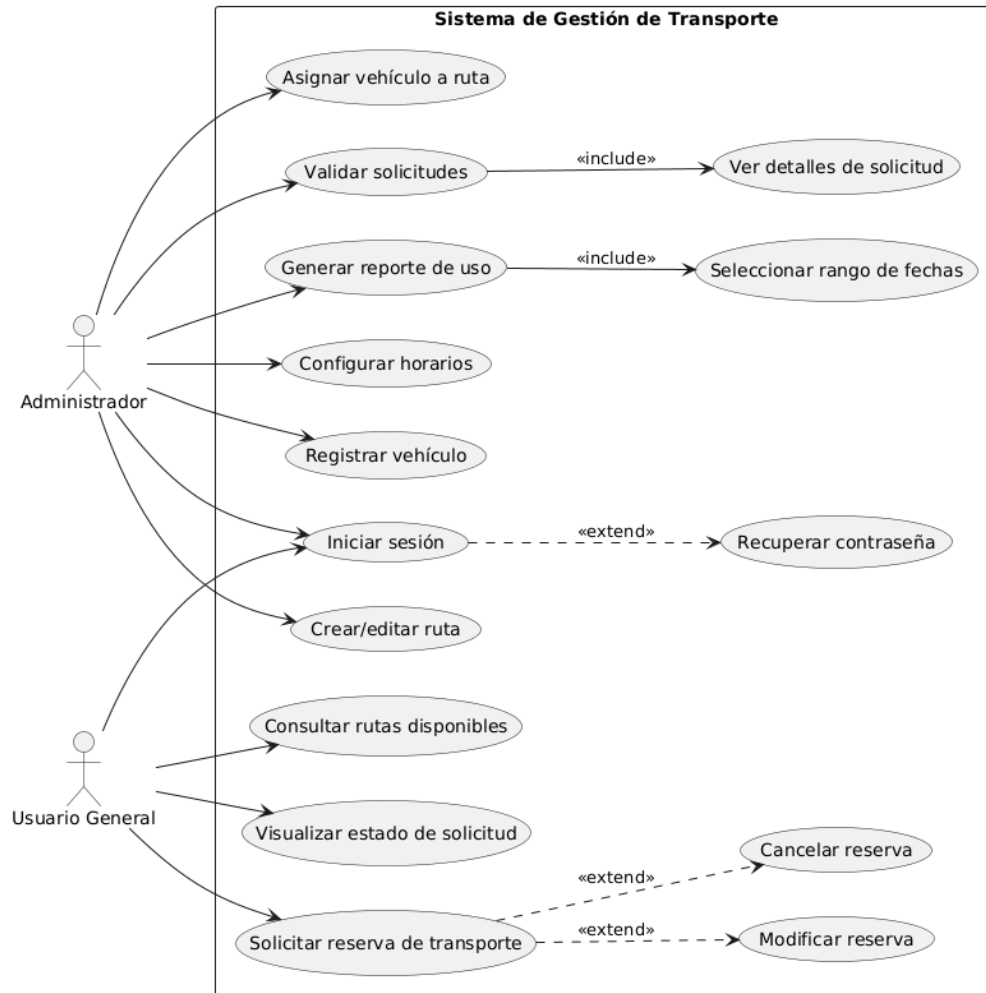


Figura 4: Diagrama de Casos de Uso propuesto

7.1 Referencias Bibliográficas

- Blog, R. (2018, abril 20). *Requerimientos Funcionales y No Funcionales, ejemplos y tips*. Medium. Recuperado de: <https://medium.com/@requeridosblog/requerimientos-funcionales-y-no-funcionales-ejemplos-y>
- Diagrama de casos de uso: Definición. (s/f). *Pmoinformatica.com*. Recuperado el 8 de mayo de 2025, de: <https://www.pmoinformatica.com/2021/02/diagrama-de-casos-de-uso-definicion-elementos-ejemplos-como.html>
- Núñez-Vivanco, G., Valdés-Jiménez, A., Besoain, F., & Reyes-Parada, M. (2016). *Geomfinder: a multi-feature identifier of similar three-dimensional protein patterns: a ligand-independent approach*. Journal of Cheminformatics, 8(1), 19. <https://doi.org/10.1186/s13321-016-0131-9>
- Universidad de Aysén. *Política general de seguridad de la información*. Resolución Exenta N.º 368, 30 de diciembre de 2024.