



# Recorrido Fluvial Valdivia – Tracker

**Integrantes:** Jorge González  
Leonardo Rodríguez  
Yoan Salom  
Jose Manuel Godoy

**Profesor:** Luis Veas Castillo

**Asignatura:** INFO 288

21 de abril, 2024 Valdivia

## Introducción

La eficiencia y accesibilidad en el transporte público son aspectos fundamentales para la movilidad urbana sostenible. En el contexto específico de Valdivia, la ciudad enfrenta desafíos únicos debido a la presencia de taxis fluviales, cuya operación está intrínsecamente ligada a las condiciones climáticas y a la complejidad de las rutas fluviales. La falta de información oportuna sobre los horarios y ubicaciones de estos taxis puede generar tiempos de espera prolongados y frustración para los usuarios.

Con el objetivo de abordar esta problemática, nuestro proyecto se centra en la creación de un sistema que permita a los usuarios conocer el recorrido de los taxis fluviales y su ubicación en tiempo real. Más allá de simplemente proporcionar datos de ubicación, nuestro enfoque busca brindar una experiencia completa que empodere a los usuarios con información detallada sobre las rutas, tiempos estimados de llegada y otros aspectos relevantes del servicio de taxis fluviales en Valdivia.

Este proyecto se enmarca en el ámbito de los sistemas distribuidos, una disciplina informática que se centra en la coordinación y gestión de recursos de manera descentralizada para lograr un objetivo común. Los sistemas distribuidos ofrecen ventajas significativas en entornos como el nuestro, donde múltiples fuentes de datos deben ser integradas y actualizadas en tiempo real. Al distribuir la carga de trabajo y descentralizar el procesamiento de información, podemos mejorar la escalabilidad, la disponibilidad y la tolerancia a fallos del sistema.

En este informe, presentaremos una serie de interrogantes diseñadas para obtener información crucial en la fase inicial del diseño del sistema distribuido. Estas preguntas estarán dirigidas a identificar requisitos clave y guiarán nuestras decisiones en cuanto a la arquitectura y tecnología a utilizar. Además, proporcionaremos un análisis de las respuestas del equipo de desarrollo, que servirán para respaldar nuestras elecciones de diseño y modelos propuestos.

El proyecto estará alojado en github, en el repositorio:

[https://github.com/info288-17-202401/INFO288-Recorrido\\_Fluvial](https://github.com/info288-17-202401/INFO288-Recorrido_Fluvial)

## Preguntas para el cliente

1. ¿Cuál es el objetivo principal de implementar este sistema de seguimiento de taxis fluviales?  
**R:** Permitirle a los usuarios de nuestro servicio de taxis conocer las posiciones de nuestros barcos en tiempo real, las rutas que estos siguen, los horarios y por último evitar las largas esperas, especialmente en invierno en el puerto.
2. ¿Cuáles son las principales funcionalidades que espera del sistema?  
**R:** El ver la posición de los barcos en un mapa de la ciudad y poder visualizar las rutas.
3. ¿Quiénes serán los usuarios finales de este sistema y cuáles son sus roles?  
**R:** Por un lado tenemos a nuestros clientes, que serían los usuarios del sistema, y por otro nuestros conductores, que podrían indicar la ruta que están siguiendo.
4. ¿Qué tipo de información específica desea mostrarle a los posibles usuarios sobre los taxis?  
**R:** Posición, ruta y conductor
5. ¿Hay algún requisito específico en términos de la precisión de la ubicación de los taxis y la frecuencia de actualización?  
**R:** Se espera que la precisión sea lo más exacta posible, además en frecuencia se espera una actualización instantánea.
6. ¿El sistema necesitará integrarse con otras plataformas o sistemas existentes?  
**R:** Esta sería la primera plataforma con la que contamos.
7. ¿Cuáles son las expectativas en cuanto al rendimiento del sistema, especialmente en términos de tiempos de respuesta y escalabilidad?  
**R:** Nos gustaría que los usuarios tengan disponibilidad del sistema la mayor parte del tiempo. En cuanto a tiempos de respuesta dado que nuestro objetivo es entregar un información en tiempo real sobre los taxis, se espera que este software tenga un retraso mínimo. Finalmente se busca un software que nos permita adaptarnos a cambios en demanda y usuarios concurrentes que hagan uso de nuestro sistema.
8. ¿Se espera que los usuarios de su sistema tengan cuentas?  
**R:** No, únicamente se esperaría cuenta para los conductores.
9. ¿Hay algún requisito legal o regulatorio que debemos tener en cuenta al diseñar este sistema?  
**R:** Actualmente en Chile se tienen regulaciones sobre la protección de datos y privacidad, por lo que es importante garantizar una recopilación, almacenamiento y procesamiento de datos adecuado y seguro.

10. ¿Cuál es el presupuesto disponible y el plazo de tiempo para completar la implementación del sistema?

**R:** No tenemos un presupuesto ajustado pero tampoco muy grande y en tiempos se esperaría una implementación dentro de los próximos 4 meses

### Requisitos Funcionales

- Rastreo en tiempo real de la posición de los barcos.
- Almacenamiento de la información de ubicación de los barcos.
- Integración con un mapa de la ciudad para visualizar la posición de los barcos.
- Interfaz de usuario para que los usuarios puedan acceder y ver la ubicación de los barcos.
- Cálculo de velocidad de los barcos para estimar tiempos de llegada a los puertos.

### Requisitos No Funcionales

- Alta disponibilidad y escalabilidad para manejar múltiples usuarios y barcos.
- Tiempo de respuesta rápido para mostrar la ubicación en tiempo real.
- Seguridad para proteger la información de los usuarios y la integridad de los datos.

### Preguntas al Equipo

1. ¿Qué tipo de herramientas utilizarán en su propuesta: software libre, de pago, desarrolladas por usted o híbridas?

**R:** Se buscará el utilizar software libre junto a desarrollo propio, con el fin de abaratar costos e igualmente poseer un mayor control sobre las capacidades de la aplicación, buscando entregar un producto acorde a los requisitos obtenidos. En un inicio pensamos en **Docker** para el uso de imágenes y crear un sistema distribuido, **Nginx** para balancear carga, **MongoDB** como base de datos y **OpenStreetMap**, para el uso de mapas.

2. ¿Qué Bibliotecas y Frameworks utilizará en su implementación?

**R:** Como se definió en la presentación inicial se piensa utilizar Python junto a Fast API para implementar el backend. Igualmente se buscaría el uso de GeoPY y GPS Broadcaster para la ubicación de los taxis. Además se considerará utilizar flask para levantar al app web.

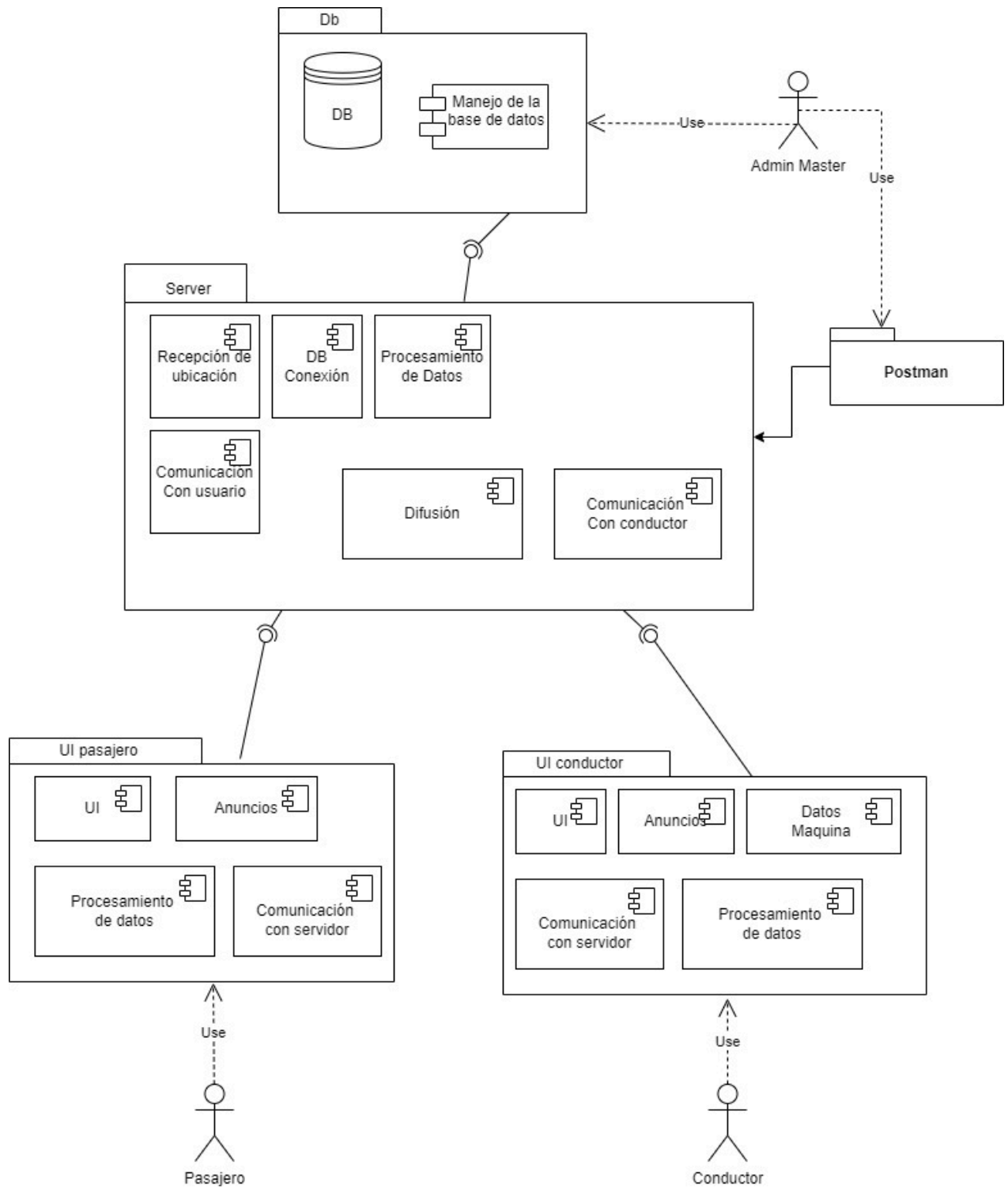
3. ¿Contempla un Frontend? ¿Que utilizara para su implementación?

**R:** En el caso del Frontend se buscaría utilizar JavaScript con reactJS y utilizando openStreet maps para visualizar un mapa de la ciudad.

4. Considere que la organización no cuenta con infinitos recursos, por ende, espera aprovechar sus recursos de la mejor manera posible.

**R:** Actualmente el proyecto no presenta tecnologías que requieran de un pago por su uso, se usarán tecnologías open source, trabajo de estudiantes y hardware propio. En un futuro se espera ejecutar en hardware propietario.

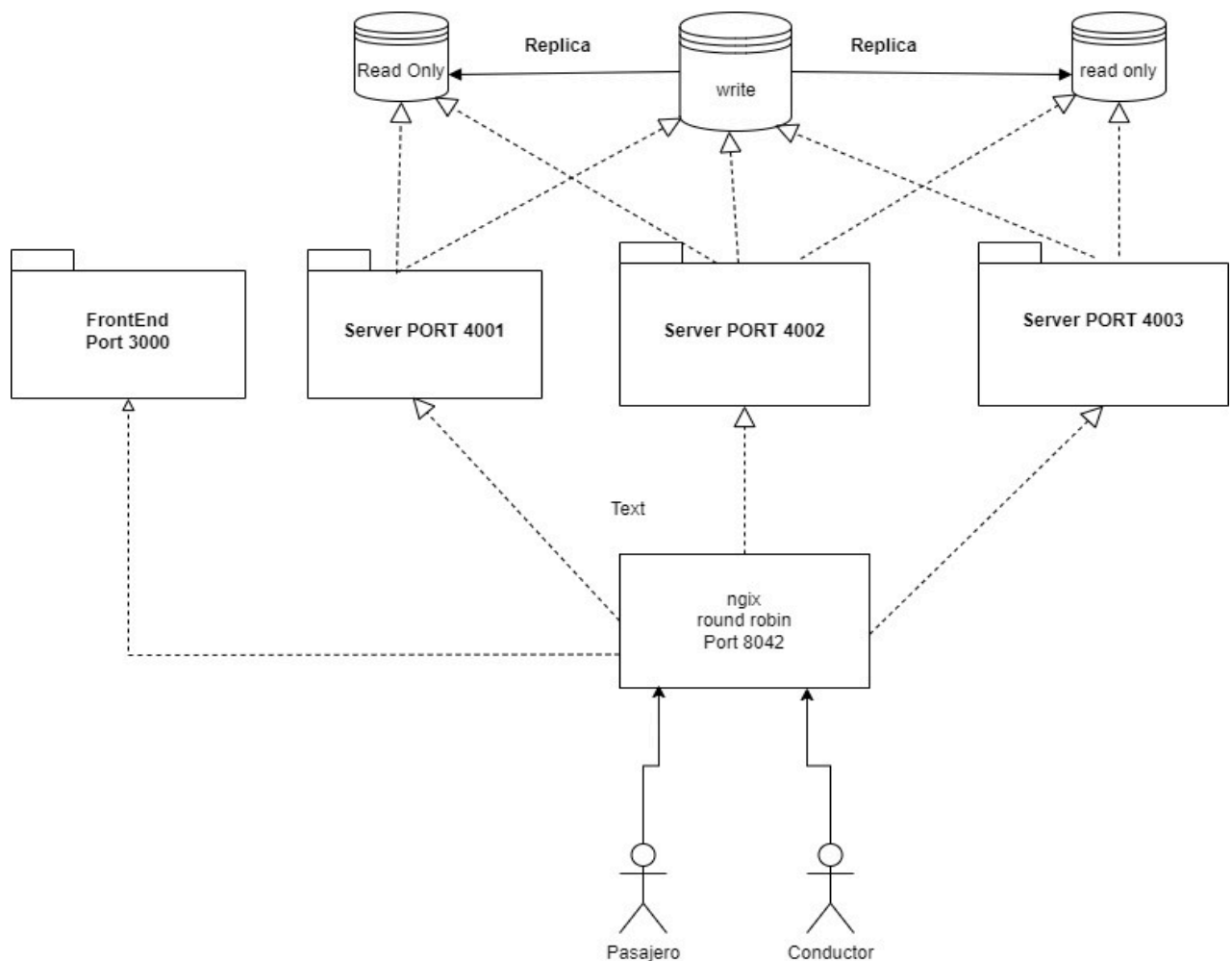
## Diagrama de componentes (Arquitectura)



## Lista de los paquetes con sus componentes (Python, ReactJS, FastAPI)

- **Paquete “Db”:** Componente que incluirá la base de datos dentro de él, así como también funcionalidad que se encargará de comunicar con el servidor web.
  - **Componente de Base de datos:** Base de datos no relacional (MongoDB)
  - **Componente manejo de db:** Encargado de realizar operaciones en la db.
- **Paquete server (Python):**
  - **Recepción de ubicación:** Componente que recibe los datos de ubicación de barcos y pasajeros.
  - **Conexión db:** Encargado de conectar y hacer peticiones a la db.
  - **Procesamiento de datos:** encargado de realizar cálculos y más.
  - **Comunicación con el usuario:** Contiene la lógica para atender peticiones de los usuarios (recibir y responder).
  - **Difusión:** Componente para función de anuncios (no ads si).
  - **Comunicación con el conductor:** Componente que gestiona la interacción con el conductor.
- **Paquete UI Pasajero (ReactJS):**
  - **UI:** Interfaz gráfica visible desde el navegador
  - **Anuncios:** Tablero que mostrará anuncios importantes
  - **Procesamiento de datos:** Componente que se encargará localmente de procesar data necesaria
  - **Comunicación con el servidor:** Componente que comunicará al dispositivo con el servidor
- **Paquete UI Conductor(ReactJS):**
  - **UI:** Interfaz gráfica visible desde el navegador
  - **Anuncios:** Tablero que mostrará anuncios importantes
  - **Procesamiento de datos:** Componente que se encargará localmente de procesar data necesaria
  - **Comunicación con el servidor:** Componente que comunicará al dispositivo con el servidor.
  - **Datos máquina:** Componente que verá datos extra de la máquina e identificación.

## Infraestructura de despliegue:



## Modelo fundamental:

El sistema distribuido que se construirá para el proyecto de recorrido fluvial Valdivia estará constituido por una serie de componentes, los cuales operarán de forma separada y se comunicarán mediante mensajes en la red, ya sea local o por medio del internet. El formato se definirá más adelante de forma más concreta, puesto que se necesita ir descubriendo las necesidades específicas de la implementación. Por otro lado, las tecnologías que usaremos usarán su propio formato de mensajes. En contraparte las consultas a la base de datos tendrán un formato estándar de NOSQL. Por la naturaleza de tiempo real del proyecto el sistema distribuido deberá ser síncrono, lo cual nos permitirá dar el servicio prometido a los clientes de la mejor manera posible.

Como nuestro sistema no forma parte de alguna infraestructura crítica que necesite estar operativa la mayor parte del tiempo, por ahora se plantearán políticas de recuperación o

grandes medidas de seguridad, pero se construirá de forma que permita implementar medidas de este tipo en un futuro, aunque medidas que nos faciliten el desarrollo como los logs si se implementarán.

### **Modelo Base de Datos:**

En el siguiente enlace se encuentra el modelo propuesto para la base de datos, diseñado específicamente para MongoDB, una base de datos NoSQL. Este modelo consta de 5 conjuntos distintos, cada uno con sus respectivas variables y descripciones. Estas variables se han definido para cumplir funciones específicas dentro del contexto de la base de datos.

**Enlace a Excel:** [📄 Variables BSD informe - 01](#)

### **Modelo Físico:**

- **Dispositivos de rastreo(Móvil con sistema android):** Se utilizará un smartphone para rastrear los barcos mediante su GPS y el envío de consultas a los servidores.
- **Servidor (Sistemas operativos ubuntu o Windows):** En este se almacenará frontend, réplicas de base de datos y réplicas del software a desarrollar, creadas mediante el uso de imágenes de Docker. Este sistema correrá sobre un servidor propietario en su despliegue, mientras se encuentre en desarrollo y estado de pruebas se utilizarán los PCs de los desarrolladores.
- **Dispositivos cliente (Móvil con sistema android y PC mediante navegador web):** Dispositivos con los cuales se visualizará el frontend del software.

### **Definición de tecnologías**

1. **React:** React es una biblioteca de JavaScript de código abierto utilizada para construir interfaces de usuario interactivas y reutilizables. Se centra en la creación de componentes independientes que gestionan su propio estado, lo que facilita la construcción de aplicaciones web escalables y mantenibles.
2. **GeoPy:** Es una librería utilizada para la geocodificación de direcciones, permitiendo calcular el tiempo esperado de un punto a otro.
3. **GPSTransmitter:** Esta librería utiliza el GPS del dispositivo para obtener la ubicación y la transmite a través de una conexión de socket.
4. **Django:** Es un framework web de alto nivel que puede ayudar a construir una aplicación web completa para visualizar y rastrear el recorrido fluvial. Con Django, puedes crear fácilmente vistas, modelos y administrar la lógica de la aplicación.
5. **Flask:** Es un microframework web ligero y flexible que se puede utilizar para construir una API RESTful o una aplicación web simple para el rastreo del recorrido fluvial. Flask es ideal para aplicaciones pequeñas y proyectos a corto plazo.
6. **Pyramid:** Es un framework web maduro y robusto que se centra en la modularidad y la escalabilidad. Puede ser una buena opción si estás buscando un framework más empresarial y escalable para el rastreo del recorrido fluvial.



7. **FastAPI:** Es un framework moderno y de alto rendimiento para construir APIs RESTful. Si planeas construir una API para transmitir y procesar los datos del recorrido fluvial, FastAPI puede ser una excelente opción.
8. **Dash:** Es un framework de visualización de datos de código abierto que se basa en Flask y React.js. Puede ser útil para crear paneles y visualizaciones interactivas para el rastreo del recorrido fluvial.
9. **Folium:** Folium es una librería que facilita la visualización de datos en mapas interactivos basados en Leaflet.js. Puede ser muy útil para visualizar el recorrido fluvial en un mapa.

"Dichas herramientas pueden estar sujetas a cambios o reemplazos según el proyecto lo requiera."