



UCSC

**UNIVERSIDAD CATÓLICA DE LA SANTÍSIMA CONCEPCIÓN
FACULTAD DE INGENIERÍA
INGENIERÍA CIVIL INFORMÁTICA**

**Desarrollo y modernización de módulos de visualización para monitoreo
ambiental**

R9 Ingeniería

Diego Alberto Araneda Hidalgo

Informe de Práctica Tutelada para optar al título de

INGENIERO CIVIL INFORMÁTICO

Supervisor: Marco Chandía

Profesor tutor: Pedro Gómez

Concepción, 08 de enero del 2026



UCSC

**UNIVERSIDAD CATÓLICA DE LA SANTÍSIMA CONCEPCIÓN
FACULTAD DE INGENIERÍA
INGENIERÍA CIVIL INFORMÁTICA**

Dedicatoria

El presente informe de práctica profesional está dedicado primeramente a Dios, por haberme concedido la oportunidad de ingresar a la universidad y la fortaleza necesaria para culminar esta etapa. A mi familia, quienes, a través de su apoyo constante, motivación y confianza, hicieron posible este logro, manteniéndose a mi lado aun cuando las dificultades, tanto externas como internas, parecían insuperables. Asimismo, dedico este trabajo a mis amigos y compañeros de carrera, por el respaldo brindado durante el proceso académico y por acompañarme a través de las diversas circunstancias que conlleva la vida universitaria, incentivando mi desarrollo personal y profesional.

Agradecimientos

En primer lugar, agradezco a la empresa R9 Ingeniería SpA por la oportunidad de realizar la práctica profesional en sus instalaciones, así como por la confianza depositada en mí para formar parte del desarrollo y modernización de sus plataformas. Extiendo mis agradecimientos al equipo de Desarrollo de Software y a la unidad técnica de la empresa, por el apoyo brindado, la disposición para el trabajo colaborativo y el aporte sustancial al aprendizaje profesional obtenido durante el desarrollo de las actividades. Finalmente, agradezco a mis profesores de la Universidad Católica de la Santísima Concepción, por ser un pilar fundamental en mi vida universitaria, entregando las herramientas académicas necesarias para el cumplimiento de los objetivos propuestos y la finalización exitosa de esta etapa formativa.



Resumen ejecutivo

Este informe detalla el desarrollo realizado durante la práctica profesional en R9 Ingeniería SpA, enfocada en la modernización del portal generador. El trabajo más importante se centró en implementar la vista de datos en tiempo real (Realtime), un módulo clave para que el cliente pudiera visualizar variables ambientales de forma inmediata y tomar decisiones operativas.

La vista Realtime se diseñó para ser flexible y escalable. Se construyó con HTML y funciones JavaScript, pero su lógica de renderizado está completamente separada del código mediante un archivo de configuración JSON. Este archivo define qué elementos se muestran y cómo funcionan, permitiendo que la vista sea adaptable a diferentes clientes sin modificar el código de la vista. Esto requirió un análisis detallado previo para modelar una estructura capaz de soportar múltiples componentes configurables: listados de alarmas, tarjetas de parámetros, tarjetas de viento con rotación dinámica, y gráficos complejos implementados con Highcharts.js.

El mapa de estaciones se integró de forma eficiente como una vista externa mediante un iframe, cuya URL se inyecta dinámicamente desde el JSON. Así, el trabajo se enfocó en su correcta disposición dentro de la interfaz, delegando su lógica interna a la vista nativa. Además, se implementó un sistema de caché para que las preferencias del usuario se guarden automáticamente, mejorando la experiencia de uso.

Uno de los desafíos principales fue que la API de Airviro solo estaba disponible mediante VPN en producción. Para probar la vista en el entorno local, se desarrolló un proxy en PHP que simuló las peticiones a la API real, garantizando que al desplegar en producción todo funcionara sin errores. Estas decisiones técnicas y una gestión de los tiempos permitieron cumplir con la prioridad de entrega exigida para la presentación ante el cliente, resultando en un producto robusto, estable y más adaptable que la versión anterior.



Executive summary

This report details the development carried out during the professional internship at R9 Ingeniería SpA, focused on modernizing the generator portal. The most important work centered on implementing the real-time data view (Realtime), a key module that allowed the client to immediately visualize environmental variables and make operational decisions.

The Realtime view was designed to be flexible and scalable. It was built with HTML and JavaScript functions, but its rendering logic is completely separated from the code using a JSON configuration file. This file defines which elements are displayed and how they work, making the view adaptable to different clients without modifying the view code. This required a detailed preliminary analysis to model a structure capable of supporting multiple configurable components: alarm lists, parameter cards, wind cards with dynamic rotation, and complex graphics implemented with Highcharts.js.

The station map was efficiently integrated as an external view using an iframe, whose URL is dynamically injected from the JSON. Thus, the work focused on its correct layout within the interface, delegating its internal logic to the native view. In addition, a cache system was implemented so that user preferences are automatically saved, improving the user experience.

One of the main challenges was that the Airviro API was only accessible via VPN in the production environment. To test the view locally, a PHP proxy was developed to simulate requests to the actual API, ensuring error-free deployment to production. These technical decisions combined with effective time management enabled us to meet the tight delivery deadline demanded for the client presentation, resulting in a robust, stable, and more adaptable product than the previous version.



Índice de Tablas

Tabla 2.1 : Resumen de tecnologías utilizadas y su aplicación en el proyecto 13

Índice de Figuras

Figura 2.1: Organigrama general de R9 Ingeniería 12

Figura 3.1: Gráfico de tendencias implementado con Highcharts 15

Figura 3.2 Arquitectura de solución al problema CORS mediante proxy en entorno local 17

Figura 3.3: Estructura genérica del archivo de configuración (JSON) 18

Figura 4.1: Vista consolidada del módulo Realtime en entorno de desarrollo (datos ficticios). 23



Contenido

Dedicatoria..... 2

Agradecimientos..... 2

Resumen ejecutivo..... 3

Executive summary..... 4

Índice de Tablas 5

Índice de Figuras..... 5

Capítulo 1: Introducción..... 8

 1.1 Contexto..... 8

 1.2 El Problema 8

 1.3 Objetivos..... 8

 1.4 Metodología..... 9

 1.5 Contribución a la Organización..... 10

Capítulo 2: Antecedentes generales de la organización 11

 2.1 Situación Actual Empresa 11

 2.2 Plataformas tecnológicas de apoyo 13

Capítulo 3: Descripción Actividades realizadas..... 14

 3.1 Actividad 1: Desarrollo de visualizaciones avanzadas con Highcharts.js..... 14

 3.2 Actividad 2: Integración y Optimización de la API de Airviro 16

 3.3 Actividad 3: Implementación de Mapas Interactivos con Leaflet..... 19

 3.4 Actividad 4: Desarrollo de Componentes Frontend y Backend 20

 3.5 Actividad 5: Documentación Técnica..... 21

Capítulo 4: Resultados 22

 4.1 Resultados Obtenidos 22

 4.2 Aprendizajes logrados 24

 4.3 Desafíos enfrentados..... 24

 4.4 Balance: Planificado vs Realizado 25

Capítulo 5: Reflexión..... 26

 5.1 Habilidades personales 26

 5.2 Habilidades interpersonales..... 26

 5.3 Habilidades de Concebir, Diseñar, Implementar y Operar (CDIO) 27



UCSC

**UNIVERSIDAD CATÓLICA DE LA SANTÍSIMA CONCEPCIÓN
FACULTAD DE INGENIERÍA
INGENIERÍA CIVIL INFORMÁTICA**

Capítulo 6: Conclusión.....	28
6.1 Objetivos planteados	28
6.2 Trabajo futuro	29



Capítulo 1: Introducción

1.1 Contexto

El presente informe muestra las actividades y resultados que se obtuvieron durante la Práctica Profesional Tutelada realizada en la empresa R9 Ingeniería, ubicada en la ciudad de Concepción. Esta empresa se especializa en brindar soluciones tecnológicas para la industria, con un enfoque en el monitoreo ambiental, desarrollo de software, gestión y obtención de datos. La práctica se llevó a cabo en el área de Desarrollo de Software, entre el 11 de agosto y el 26 de diciembre del año 2025, bajo una modalidad de trabajo primeramente full remota, luego fue solamente presencial y en su etapa final fue híbrida.

1.2 El Problema

Mi inserción al equipo de trabajo respondió a la necesidad de poder modernizar la plataforma, desarrollar nuevos módulos de visualización de datos e integrar sistemas de monitoreo. En este contexto, el trabajo se centró en aplicar conocimientos de ingeniería de software para la optimización de procesos que apoyan la toma de decisiones estratégicas dentro de los servicios que la empresa ofrece a sus clientes.

1.3 Objetivos

Para poder guiar el desarrollo de la práctica se tuvieron que establecer los siguientes objetivos general y específicos:

Objetivo General:

- Desarrollar e implementar diferentes módulos para el análisis, monitoreo y visualización, para la toma de decisiones que permitan al usuario final, el poder visualizar sus datos ambientales finales, de esta manera, aseguran la escalabilidad y seguridad de la información.



Objetivos Específicos:

- Analizar los requerimientos funcionales y no funcionales del proyecto en conjunto con el equipo de desarrollo.
- Diseñar la arquitectura de los módulos utilizando tecnologías modernas tales como Laravel (PHP) para el backend, MySQL para la base de datos y Tailwind CSS para la interfaz de usuario, también una librería de Javascript como Highcharts para la implementación de gráficos robustos.
- Implementar el código fuente siguiendo buenas prácticas y estándares de programación para asegurar la mantenibilidad.
- Realizar pruebas unitarias y de integración para validar el correcto funcionamiento de los módulos entregados.

1.4 Metodología

La metodología de trabajo utilizada se basó en el marco de trabajo ágil SCRUM, participando en reuniones de planificación, seguimiento o daily reviews y revisión de avances iterativos a través de la plataforma Slack, Zoom y Clickup. Esto permitió una adaptación flexible a los cambios en los requerimientos y una comunicación constante con el supervisor y el equipo de proyecto. Asimismo, se utilizaron herramientas de gestión y versión de código tales como Github y Git para mantener el orden y trazabilidad del desarrollo.



1.5 Contribución a la Organización

Lo más importante que aporté a R9 Ingeniería SpA fue entregar funcionando los módulos de visualización del "Portal Generador" (Monitoring, Dashboards, Whitepages y Realtime), actualizados con una interfaz estandarizada y adaptable que mejora bastante la experiencia de los usuarios. Pero más allá de que funcione, lo más valioso técnicamente fue diseñar una arquitectura escalable usando archivos JSON para la configuración. Esto permite separar la lógica del código de los datos de cada cliente, de manera que la empresa pueda agregar nuevas estaciones y parámetros sin tener que tocar el código base, ahorrando tiempo en futuras implementaciones. Además, se dejó un código mejorado y bien documentado para el equipo, incluyendo el proxy en PHP que hice para poder trabajar localmente sin problemas de conexión con la API, y también se implementó mecanismos de caché y paginación para que el sistema maneje un desempeño estable cuando hay muchos datos.

La implementación lógica demostró ser robusta según la validación de la supervisión técnica, permitiendo que las últimas fases se dedicaran solo al pulido visual y detalles estéticos, dado que el rendimiento y la estabilidad de los datos funcionaron sin inconvenientes mayores



Capítulo 2: Antecedentes generales de la organización

2.1 Situación Actual Empresa

R9 Ingeniería SpA lleva más de 20 años operando en Chile, enfocándose en soluciones tecnológicas y gestión ambiental para la industria pesada. La empresa orienta su estrategia hacia la Industria 4.0, priorizando la digitalización de procesos importantes y el cumplimiento de normativas ambientales. Trabaja con clientes relevantes del sector minero y energético, como Codelco, ENAP y el Ministerio del Medio Ambiente.

La empresa tiene una estructura jerárquica que parte desde la Gerencia General, dividida en tres áreas principales: Comercial, Servicios y Proyectos. Esto permite organizar las operaciones en dos unidades de negocio: la Unidad Ambiental, que se encarga del monitoreo y manejo de datos; y la Unidad Tecnológica, dedicada al desarrollo de software y soporte técnico.

La práctica se llevó a cabo dentro de la Gerencia de Proyectos, dirigida por Marco Chandía. Me incorporé al equipo de Ingeniería de Proyectos, donde trabajé bajo la supervisión de Rodrigo Villegas, un desarrollador que guio mi trabajo. Colaboré al equipo de desarrollo Full Stack en tareas relacionadas con la modernización del portal de la empresa.

La *Figura 2.1* muestra el organigrama actual de R9 Ingeniería, donde se puede ver la jerarquía y dónde se ubica el equipo de ingeniería.

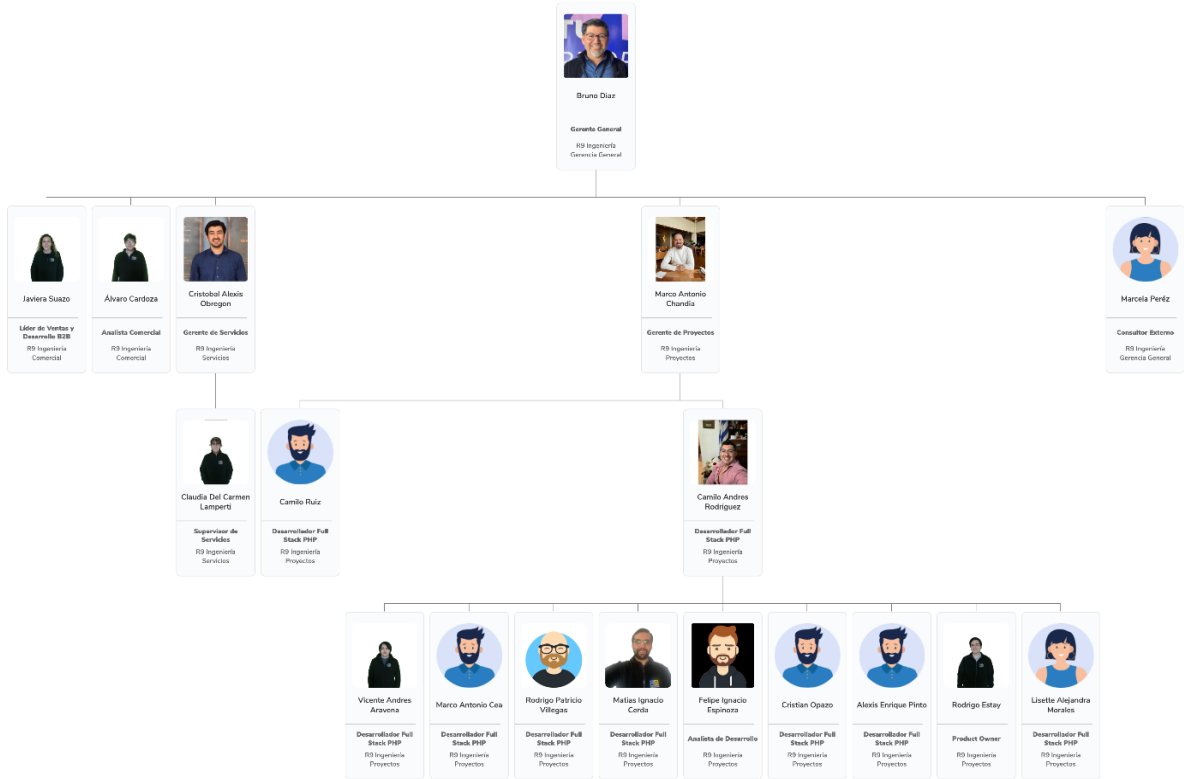


Figura 2.1: Organigrama general de R9 Ingeniería



2.2 Plataformas tecnológicas de apoyo

A continuación, en la *Tabla 2.1* Tabla 2.1 se detalla el conjunto de tecnologías y herramientas seleccionadas para el desarrollo del proyecto, describiendo su función y justificación de su elección.

Tecnología	Uso	Justificación
Laravel	Backend/API	Permite crear una arquitectura segura, para gestionar lógica del negocio y procesar solicitudes del servidor.
Tailwind CSS	Estilos UI	Framework utilizado para crear interfaces modernas, facilitando la visualización de datos en distintos dispositivos.
MySQL	Base de Datos	Base de datos relacional utilizada para almacenar y consultar eficientemente grandes volúmenes de información ambiental, asegurando la integridad de los datos
Git y GitHub	Control Versiones	Herramienta para el trabajo en conjunto y cambios continuos del código, asegurando el seguimiento de cambios durante el desarrollo del proyecto.

Tabla 2.1 Resumen de tecnologías utilizadas y su aplicación en el proyecto



Capítulo 3: Descripción Actividades realizadas

3.1 Actividad 1: Desarrollo de visualizaciones avanzadas con Highcharts.js

Periodos de Ejecución

- Fase Inicial: Semanas 8, 9 y 10
- Fase Avanzada: Semanas 18, 19, 20 y 21

El trabajo con Highcharts.js comenzó con el estudio de la tecnología y la implementación de la vista de Dashboards, integrando filtros por resolución temporal y variables. Posteriormente, en la fase avanzada (vista Realtime), se implementó un gráfico de tendencias. Tal como se detalla en la *Figura 3.1*, esta visualización incorpora un doble eje Y para correlacionar variables estándar con la dirección del viento, utilizando un formateador automático para convertir grados a puntos cardinales. Se desarrollaron modales de detalle que incluyen plotbands (franjas de rango crítico) y lógica dinámica para cambiar el color de las series según el estado (normal o crítico). Finalmente, se implementaron controles de zoom condicionales y se aseguró la persistencia de configuración de temas (claro/oscur).

Desafíos y Soluciones

- Sincronización y Representación: hubo dificultades iniciales con la sincronización de datos al cambiar resoluciones, lo que se solucionó ajustando la lógica de peticiones.
- Gráficos Complejos (Viento): la implementación del doble eje Y y la rotación de flechas presentó desafíos de redundancia visual, corregidos mediante mapeo automático de series y ajustes de cálculo de rotación.
- Rendimiento: la lentitud en gráficos con grandes volúmenes de datos (resolución minutal) se mitigó implementando estrategias de carga progresiva.



Estado Final

- Se logra un sistema de visualización robusto con gráficos de tendencias interactivos, con soporte para doble eje, alertas visuales dinámicas y modales detallados integrados completamente con la API.

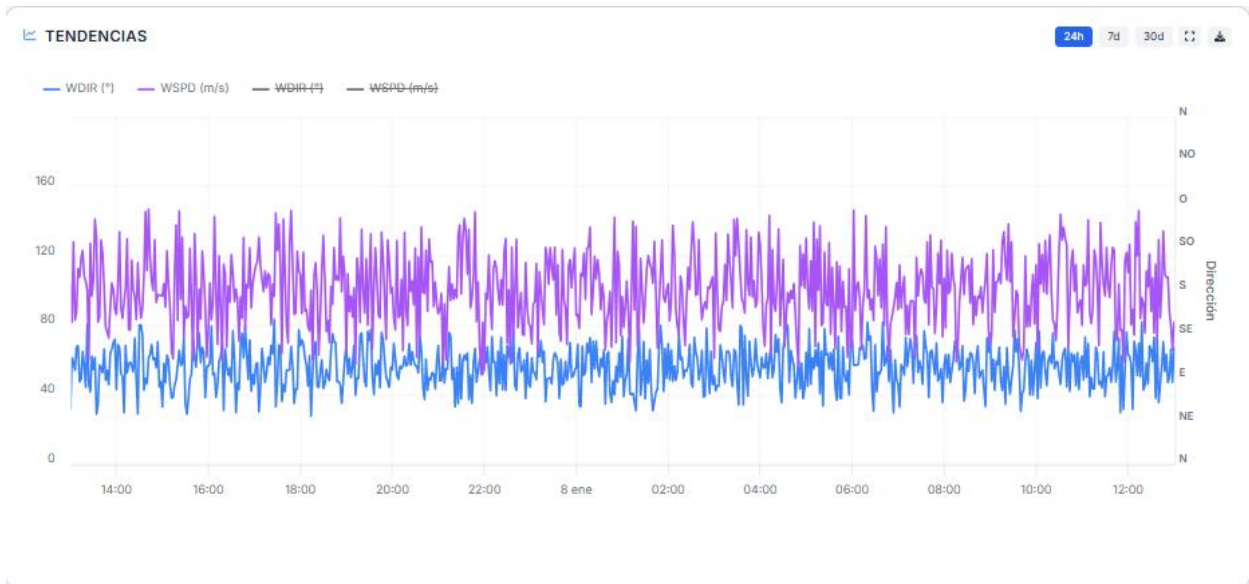


Figura 3.1: Gráfico de tendencias implementado con Highcharts



3.2 Actividad 2: Integración y Optimización de la API de Airviro

Periodos de Ejecución

- Implementación: Semanas 7, 8, 11, 12 y 13.
- Seguridad y Dinamismo: Semanas 16 y 17.

La integración con Airviro se centró en desarrollar la lógica del lado del cliente, backend para procesar datos ambientales. Se implementaron peticiones dinámicas mediante ApiRest para alimentar las vistas de Dashboards, Monitoreo, Whitepages y Realtime. Donde para esta última, se diseñó una arquitectura basada en archivos de configuración JSON, tal como se puede apreciar en la *Figura 3.3* permitiendo así que la interfaz se construya y actualice dinámicamente sin datos *hardcodeados*. Además, se optimizó el rendimiento mediante sistemas de paginación para grandes volúmenes de datos y caché temporal para evitar consultas redundantes.

Desafíos y Soluciones

- Restricciones de Desarrollo Local (CORS): para solucionar el bloqueo de peticiones a la API de Airviro por políticas CORS en el entorno local, se implementó un proxy temporal en PHP. Tal como se muestra en la *Figura 3.2*. Este componente sirvió como intermediario para simular las peticiones a la API sin necesidad de conectarse directamente desde el entorno de desarrollo, permitiendo validar exitosamente la lógica de consumo y visualización de datos en el Frontend.
- Manejo de Datos Nulos: algunos medidores retornaban valores nulos desde la API, lo que generaba fallos en la visualización. Para resolver esto, se agregaron validaciones en el código y se usaron medidores de prueba que aseguraran el correcto funcionamiento del sistema.

Estado Final

- Se entregó el módulo funcional con la lógica de consumo de datos operativa, archivos de configuración JSON estructurados y las vistas integradas, listas para su funcionamiento en el entorno de producción.

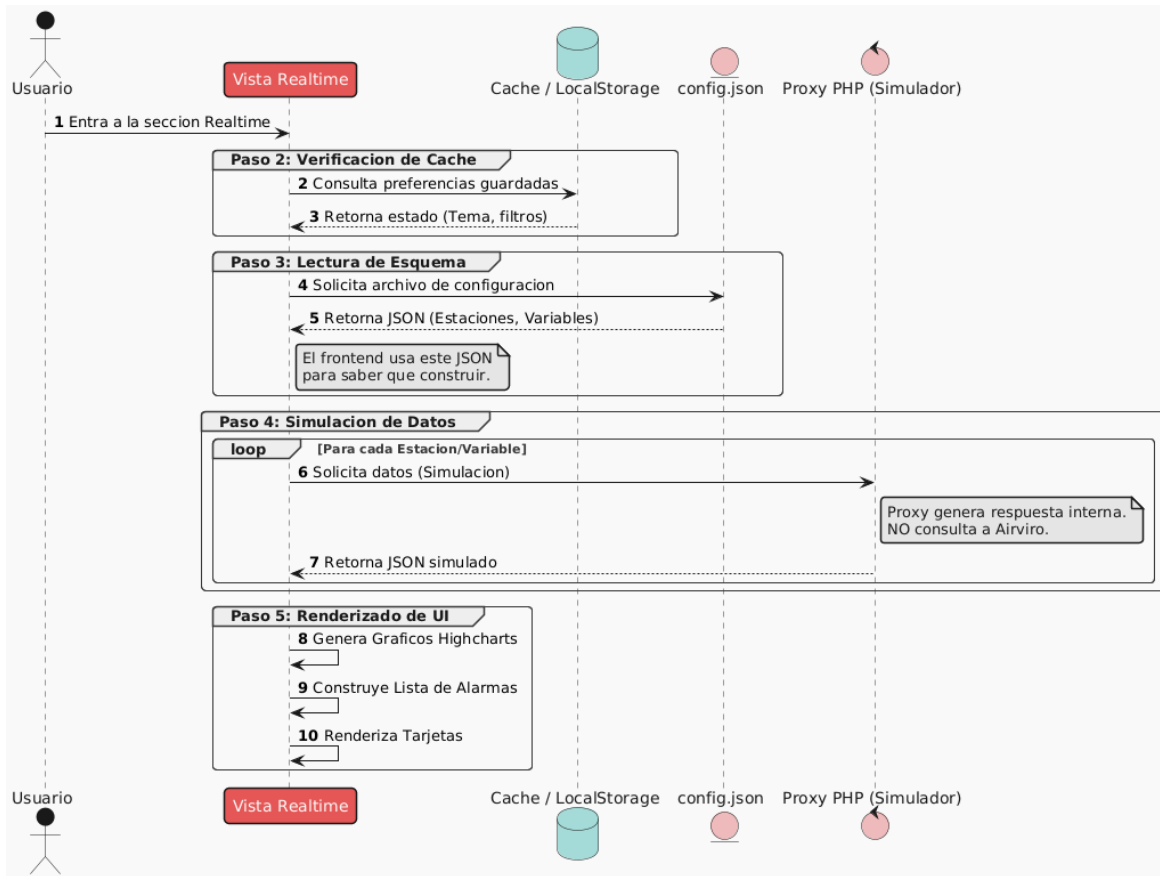


Figura 3.2 Arquitectura de solución al problema CORS mediante proxy en entorno local



```
{
  "config": {
    "name": "Realtime",
    "timezone": "<String: Zona horaria, ej. Etc/GMT+4>",
    "groupTitles": {
      "gp1": "<String: Nombre del grupo de visualización>",
      "....": "...."
    },
    "updateConfig": {
      "apiBaseUrl": "<URL: Endpoint de la API de Airviro>",
      "interval": "<Integer: Tiempo en ms, ej. 300000>",
      "apiToken": "<String: Token de autenticación oculto>"
      // ... otras configuraciones de conexión
    },
    "windConfig": {
      "directionParam": "DIRECCION",
      "speedParam": "VELOCIDAD"
    }
  },
  "stations": [
    {
      "key": "<String: ID único, ej. BTM>",
      "name": "<String: Nombre visible, ej. BOCATOMA>",
      "latlng": [ "<Float: Latitud>", "<Float: Longitud>" ],
      "parameters": {
        "<CODIGO_PARAMETRO>": {
          "serie": "<String: ID de la serie en base de datos>",
          "unitName": "<String: Unidad, ej. µg/m³>",
          "status": [
            {
              "name": "<String: Etiqueta del estado (Normal/Alerta)>",
              "from": "<Number: Valor mínimo del rango>",
              "to": "<Number: Valor máximo del rango>",
              "colorClass": "<String: Clase CSS para el color>"
            },
            { "....": "...." } // Más rangos de estado
          ]
        },
        // ... se repite para otros parámetros (PM2.5, SO2, etc.)
      }
    },
    { "....": "...." } // Resto de las estaciones configuradas
  ]
}
```

Figura 3.3: Estructura genérica del archivo de configuración (JSON)



3.3 Actividad 3: Implementación de Mapas Interactivos con Leaflet

Periodos de Ejecución

- Optimización y Ajustes: Semana 21

El trabajo específico reporta en esta actividad se centró en mejorar la experiencia de usuario y la visualización del mapa en la vista Realtime. Se implementó y ajustó la funcionalidad de pantalla completa, asegurando que los contenedores del mapa (basados en Leaflet/iframe) respondieran correctamente a la redimensión de la ventana. Se modificó la estructura de los contenedores estáticos para hacerlos dinámicos y evitar recortes visuales en distintas resoluciones.

Desafíos y Soluciones

- Visualización pantalla completa: el mapa tenía recortes y problemas de altura al activarse la pantalla completa. La solución implicó refactorizar los estilos del Tailwind CSS y modificar el contenedor padre para permitir la redimensión del iframe.

Estado Final

- El componente de mapa se integra correctamente con la vista de Realtime, soportando el modo de pantalla completa sin errores visuales ni de su escala.



3.4 Actividad 4: Desarrollo de Componentes Frontend y Backend

Periodo de Ejecución

- Transversal: Semanas 1 a 6 y Semanas 14 y 15.

Esta actividad abarca la construcción de la interfaz de usuario y la lógica de servidor. Se realizó la migración de estilos de Bootstrap a Tailwind CSS en el módulo de monitoreo. Se desarrollaron componentes modulares reutilizables, y se llevó a cabo una estandarización masiva UI, reemplazando elementos *selects* nativos por dropdowns personalizados con buscadores e íconos. En el backend, se optimizó la lógica de mantenimiento, eliminando redundancias y creando funciones modulares para la gestión de datos. También se implementaron mejoras de usabilidad como tablas con scroll horizontal y arrastre manual.

Desafíos y Soluciones

- Conflictos de Diseño y CSS: la integración de nuevos componentes generó superposiciones con el *footer* y errores visuales en la resolución de móviles, resueltos mediante ajustes de *z-index*, refactorización de *layouts* y pruebas de responsividad.
- Deuda Técnica: la existencia de código antiguo y funciones redundantes complicaba el mantenimiento. Se dedicaron semanas específicas a la limpieza, refactorización y modularización del código.

Estado Final

- En la *Figura 4.1* se puede ver cómo quedaron integrados estos componentes en la interfaz final, mostrando la homogeneidad visual y la capacidad de adaptación a distintos dispositivos que se logró.



3.5 Actividad 5: Documentación Técnica

Periodos de Ejecución

- Ejecución: Semana 4 y Semana 21.

La documentación se abordó en dos etapas. Inicialmente, se realizó la documentación y ordenamiento del código fuente siguiendo buenas prácticas para mejorar la legibilidad tras las primeras implementaciones. Hacia el final de la práctica, se procedió a analizar y sistematizar las funcionalidades complejas implementadas (modales, estados dinámicos, configuración JSON en Realtime y Dashboards), estableciendo la base técnica para el informe final y la transferencia de conocimiento.

Estado Final

- El código se encuentra comentado y ordenado, y se ha generado la base de información necesaria para la documentación técnica final del proyecto.



Capítulo 4: Resultados

4.1 Resultados Obtenidos

Durante el desarrollo de la práctica, se lograron implementar varias soluciones tecnológicas que dan cumplimiento directo a los objetivos específicos planteados en el *Capítulo 1*:

- **Diseño de Arquitectura y Tecnologías:** se logró desarrollar 4 vistas importantes (Monitoring, Dashboards, Whitepages y Realtime) que fueron migradas a Tailwind CSS y funcionan con Laravel, confirmando que la arquitectura planteada en el diseño era adecuada. Esto se puede observar en la *Figura 4.1*.
- **Visualización Avanzada:** se implementó de manera sólida Highcharts.js, permitiendo crear gráficos de doble eje y plotbands, lo que satisface la necesidad de contar con visualizaciones gráficas robustas.
- **Optimización y Buenas Prácticas:** se diseñaron estrategias de carga específicas para cada módulo. Dashboards y Whitepages incorporaron paginación para gestionar rangos dinámicos (ej. 30 días con aprox. 7.200 datos segmentados) sin bloquear la interfaz. Monitoring utilizó ventanas predefinidas según la resolución temporal, y Realtime optimizó la carga con ventanas de 24 horas (1.440 registros minutales), garantizando fluidez en el sistema completo.
- **Validación del Sistema:** el uso del proxy junto con las pruebas realizadas en ambiente local garantizó que el despliegue en producción se realizara sin inconvenientes ni errores.

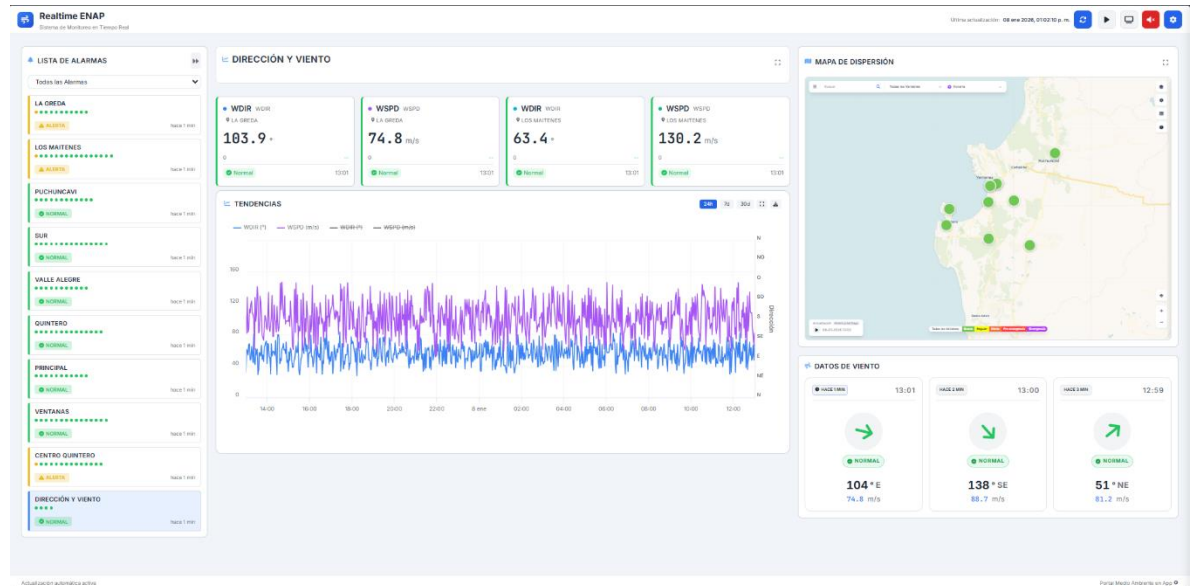


Figura 4.1: Vista consolidada del módulo Realtime en entorno de desarrollo (datos ficticios).



4.2 Aprendizajes logrados

- Dominio de frameworks: se profundizó en el uso de Highcharts.js para la visualización de los datos y leaflet para mapas interactivos, comprendiendo como utilizar el DOM para lograr renderizados complejos (como la rotación de flechas de viento).
- Integración Backend-Frontend: se consolidó el conocimiento en Laravel y PHP, específicamente en la creación de puentes de comunicación entre una API externa (Airviro) y una interfaz reactiva, aprendiendo a manejar tokens, cabeceras y códigos de respuesta HTTP.
- Gestión de control de versiones: se adquirió experiencia práctica en flujos de trabajo con git, aprendiendo a manejar ramas personales, resolver conflictos de merge complejos con la rama principal y mantener un historial de commits limpio.
- Resolución de problemas: se desarrolló la capacidad de diagnosticar errores de entorno (como fallo de MySQL en XAMPP o bloqueos CORS), implementando soluciones instrumentales (como el proxy) para mantener la continuidad del desarrollo.

4.3 Desafíos enfrentados

- Curva de aprendizaje en integración de API: uno de los mayores desafíos fue el bloqueo por políticas de CORS al trabajar en entorno local. Esto obligó a investigar y desarrollar una solución intermedia (proxy PHP) para continuar con el desarrollo de la vista sin depender de la configuración del servidor de producción.
- Manejo de deuda técnica y código heredado: al comenzar, se encontró código con estilos mezclados (Bootstrap/Tailwind) y datos escritos directamente en el código. El desafío fue reorganizar y limpiar estas vistas sin romper lo que ya funcionaba, migrando poco a poco hacia una arquitectura más flexible basada en archivos de configuración JSON.



- Conflictos de integración: la integración del trabajo propio con la rama principal generó conflictos de código que rompían el layout o la lógica. Superar este requirió análisis detallado del código de otros desarrolladores y comunicación con el equipo para restaurar la funcionalidad.
- Complejidad en la visualización de datos: implementar gráficos que combinaran variables escalares (temperaturas) con vectoriales (viento) en un mismo plano presentó retos matemáticos y visuales, como la correcta rotación de iconos y la sincronización de doble eje Y.

4.4 Balance: Planificado vs Realizado

Si bien la planificación inicial establecía una estructura lineal de actividades, la realidad del desarrollo de software exigió un enfoque iterativo y flexible.

- Adaptación al cambio: hubo cambios importantes, destacando la reimplementación total de la vista de monitoreo durante la semana 2. Una integración que falló comprometió la estabilidad del módulo y no fue posible hacer rollback, por lo que se decidió empezar de nuevo. Afortunadamente, el entendimiento técnico ya logrado sobre la lógica facilitó una reconstrucción más rápida y resultó en código mucho más limpio que el primer intento.
- Expansión del alcance: al principio se contemplaba la visualización estándar, pero el alcance creció hacia requerimientos más complejos como la exportación de CSV, la implementación de modales detallados y la configuración de entornos para nuevos clientes, los cuales fueron abordados exitosamente dentro del plazo de las 21 semanas.
- Conclusión balance: como conclusión del balance, se puede confirmar que los objetivos específicos fueron cumplidos. Se realizó el análisis de los requerimientos que fueron evolucionando, se diseñó una estructura modular utilizando JSON, se escribió código respetando estándares de optimización y se validó que todo funcionara correctamente mediante pruebas simuladas y el despliegue final.



Capítulo 5: Reflexión

5.1 Habilidades personales

Durante la práctica, la resolución de problemas y adaptabilidad fueron constantes. Un momento clave fue enfrentar un error crítico en la segunda semana que obligó a reiniciar la vista de monitoreo. Esta experiencia cambió mi forma de trabajar hacia un enfoque más ordenado y modular, revisando cada paso, lo que fortaleció mi tolerancia a la frustración y me enseñó a ver los errores como parte del aprendizaje. Mi aporte técnico fue importante al resolver problemas de infraestructura local, como los bloqueos CORS, diseñando un proxy en PHP que me permitió simular peticiones reales y continuar desarrollando la vista Realtime sin contratiempos. Finalmente, mi motivación y respeto por el trabajo ajeno se reflejaron en mantener el código limpio y bien estructurado, pensando en facilitar el trabajo de futuros desarrolladores, y en mi gestión de tiempos, ajustando mi horario y extendiendo jornadas cuando fue necesario para cumplir con los plazos.

5.2 Habilidades interpersonales

Pasar a la modalidad presencial fue muy importante para mí integración, porque me permitió conocer de cerca cómo funciona la empresa y la dinámica de respeto profesional entre las distintas áreas. Esto hizo más fácil mi adaptación al ritmo de trabajo y me hizo sentir más parte del equipo. Respecto al trabajo en equipo y la comunicación, siempre busqué hablar directamente y coordinar con mis compañeros. Por ejemplo, cuando había conflictos al juntar código, no tomaba decisiones solo: me comunicaba con los demás para entender la lógica de sus cambios antes de integrar todo. De esta forma mantenía el proyecto estable y las relaciones con el equipo en buenos términos. También aprendí a comunicarme de forma clara y a estar abierto a recibir comentarios. Tomé las correcciones de mis supervisores como una forma de aprender y mejorar el producto, y usé los canales de comunicación del equipo para proponer mejoras por mi cuenta, el cómo optimizar la estructura del JSON, buscando un balance entre aportar ideas nuevas y seguir las indicaciones del equipo.



5.3 Habilidades de Concebir, Diseñar, Implementar y Operar (CDIO)

La práctica me permitió aplicar lo aprendido en teoría al desarrollo real de software:

- **Concebir:** me di cuenta de que necesitaba crear una herramienta de apoyo. El proxy se hizo en PHP debido a que en mi entorno local tenía restricciones, entonces tuve que pensar en otras soluciones para poder avanzar.
- **Diseñar:** al estructurar el archivo JSON para el sistema de configuración, entendí la importancia de pensar en la escalabilidad y mantenibilidad. Diseñé pensando en que fuera sencillo el poder agregar clientes después sin modificar el código base.
- **Implementar:** trabajé con tecnologías nuevas para mí, investigando documentación oficial para crear gráficos con Highcharts.js y consumir APIs en Laravel. Este proceso de aprendizaje autónomo fortaleció mi capacidad de resolver problemas y me mostró la importancia de seguir buenas prácticas de programación.
- **Operar:** implementé paginación y caché pensando en el comportamiento del sistema con datos reales y múltiples usuarios. Esto me hizo consciente de la responsabilidad de crear sistemas estables y eficientes para producción.

Estos 4 meses de práctica en R9 Ingeniería fueron clave para esta transición del mundo académico y la industria real. Más allá de aprender sobre visualización de datos e integración de APIs, lo más importante fue desarrollar la capacidad de enfrentar problemas poco claros y convertirlos en soluciones que realmente funcionan. Esta experiencia fortaleció mi perfil profesional, dándome la confianza para trabajar con sistemas complejos y la madurez para integrarme en equipos de desarrollo, entendiendo que el valor de un ingeniero no está solo en el código que escribe, sino en qué tan eficientes y escalables son sus soluciones.



Capítulo 6: Conclusión

El portal avanzó desde interfaces simples a un sistema flexible y escalable. Lo más importante fue el implementar un archivo de configuración JSON para la vista Realtime, separando su lógica del código y permitiendo gestionarla sin modificar la programación base. Se superaron limitaciones técnicas con soluciones como el proxy PHP, y se aseguró el buen funcionamiento del sistema mediante paginación y caché, fundamentales para procesar eficientemente grandes volúmenes de datos.

6.1 Objetivos planteados

- **Analizar requerimientos:** este objetivo se logró de manera satisfactoria. La participación constante en las reuniones de planificación y la capacidad de adaptarse cuando no se tuvo acceso a la API (por tema de VPN) permitieron analizar las limitaciones técnicas y ajustar los requerimientos del proyecto trabajando en conjunto con el equipo de desarrollo.
- **Diseñar arquitectura:** se cumplió completamente. Se diseñó e implementó una arquitectura actualizada que combina Laravel en el backend con Tailwind CSS para la interfaz visual. Además, se incorporó una estructura de configuración dinámica mediante JSON y se agregaron visualizaciones avanzadas usando Highcharts.js.
- **Implementar buenas prácticas:** este objetivo se alcanzó entregando código modular, bien documentado y optimizado. La incorporación de paginación y caché para gestionar los datos de forma eficiente muestra el compromiso con la mantenibilidad y el buen rendimiento del sistema.
- **Realizar pruebas y validación:** se cumplió satisfactoriamente mediante pruebas funcionales en los módulos Monitoring, Dashboards y Whitepages. Para Realtime, que opera de forma independiente, se desarrolló un Proxy en PHP que permitió validar localmente su funcionamiento antes del despliegue en producción.



UCSC

**UNIVERSIDAD CATÓLICA DE LA SANTÍSIMA CONCEPCIÓN
FACULTAD DE INGENIERÍA
INGENIERÍA CIVIL INFORMÁTICA**

6.2 Trabajo futuro

Basado en los módulos trabajados, se podría continuar con estos desarrollos para un futuro:

- Automatización de alertas: dado que la vista Realtime ya muestra rangos críticos (plotbands), se propone desarrollar un servicio en el backend que monitoree estos valores y envíe notificaciones automáticas por correo cuando se excedan los umbrales establecidos por la normativa.
- Generación automática de informes: se sugiere ampliar la funcionalidad de exportación de datos para generar reportes ejecutivos en PDF de forma automática y programada, facilitando la entrega periódica de informes mensuales sobre los datos ambientales.