



CARONTE

Stack tecnológico

Devising a Project

Grupo 9

Borrego Angulo, Hugo

Chico Castellano, Álvaro

Duque Colete, Rafael

Galván Cancio, Daniel

García Carballo, Juan

García Escudero, Ángel

García Rivero, Andrés Francisco

Guillén Fernández, David

Herencia Solís, Lucas Manuel

Linares Barrera, Jaime

Muñoz Rodríguez, Jorge

Pérez Santiago, Alejandro

Rodríguez Reina, Javier

Solís Padilla, Isaac

Youssafi Benichikh, Karim

Índice

1. Introducción	3
2. Arquitectura	3
3. Herramientas de desarrollo	4
4. Justificación	4
5. Departamentos	5

1. Introducción

Este documento detalla las herramientas y frameworks que se emplearán en el backend, frontend, bases de datos, pruebas, despliegue y control de versiones del proyecto CARONTE.

El objetivo principal es establecer una arquitectura tecnológica robusta y eficiente, asegurando escalabilidad, mantenibilidad y facilidad de integración entre los distintos componentes del sistema. Se han seleccionado tecnologías modernas y ampliamente utilizadas en la industria para optimizar el desarrollo y la calidad del software.

2. Arquitectura

- **Backend:**
 - Spring Boot - Framework principal del backend
 - Spring Web - Para manejar solicitudes HTTP y crear API REST
 - Spring Data JPA - Para interactuar con BBDD mediante HIBERNATE
 - Swagger - Para documentar la API
 - Lombok - Reducir código repetitivo en Java
 - Maven - Para gestionar dependencias y compilación
 - JWT - Autenticación
- **Frontend:**
 - React - Interfaz de usuario
 - React Router - Navegación entre páginas
 - Fetch API - Llamadas HTTP al backend
- **Base de datos:**
 - MySQL.
- **Tests:**
 - JUnit y Mockito: Pruebas unitarias y de integración
 - Jest: Pruebas en React
- **Despliegue:**
 - Backend y Frontend (.jar): Railway (Plan Hobby: 5€/mes)
- **Dominio y email:**
 - **Dominio:** IONOS (caronte.site)
 - **Email:** zoho (admin@caronte.site)

3. Herramientas de desarrollo

El equipo utiliza diversas herramientas para optimizar el proceso de desarrollo y colaboración:

- **IDE y Editores.** Visual Studio Code e IntelliJ IDEA para desarrollo en frontend y backend
- **Control de versiones.** GitHub para la gestión del código fuente y colaboración en equipo
- **Gestión de dependencias.** Maven para backend y npm/yarn para frontend
- **Análisis de código.** Sonarqube
- **CI/CD.** GitHub Actions para integración y despliegue continuo
- **Documentación.** Swagger para API y Google Docs/Spreadsheets para documentación interna

4. Justificación

Se ha seleccionado Spring Boot para el backend y React para el frontend de la aplicación, puesto que es una tecnología con la que todos los miembros del equipo han trabajado anteriormente. Además, React posibilita una interfaz web reactiva adaptada a escritorio y móvil. Junto a estas tecnologías, se suman las herramientas de testing JUnit, Mockito y Jest.

Dado que se trata de una mediana aplicación, se ha seleccionado como base de datos MySQL, ya que este software permite un soporte adecuado y menos costoso en recursos que otras alternativas como PostgreSQL.

Para el análisis de código se utilizará Sonarqube, ya que es una tecnología conocida por todo el equipo de trabajo.

Para el hosting de CARONTE se ha elegido el plan Hobby de Railway, que permite el alojamiento para el backend, frontend y base de datos de la aplicación en sus distintas versiones. Proporciona 8Gb de RAM y 8 vCPUs que estimamos suficientes para el correcto funcionamiento del servicio. Adicionalmente, se ha contratado por un año el dominio caronte.site, usado para el despliegue de la aplicación principal y las cuentas de correo electrónico corporativas.

Como sistema de gestión de versiones se usará Git. Se alojará el código en repositorios pertenecientes a una organización CARONTE de GitHub. También se utilizará la tecnología GitHub Actions para tareas de automatización CI/CD.

Se utilizará la suite ofimática de Google para los documentos internos, y swagger para la documentación de la API.

5. Departamentos

Departamento Backend	
Jorge (Coordinador)	Ángel
Alejandro	Lucas
Javier	David

Departamento Frontend	
Juan (Coordinador)	Andrés
Jaime	Hugo
Daniel	Karim
Rafael	Álvaro

Departamento GitHub y CI/CD	
Ángel (Coordinador)	Daniel

Departamento Frontend	
Daniel (Coordinador)	Álvaro
Karim	

Departamento Hosting	
Alejandro (Coordinador)	Jorge

Departamento Documentación	
Álvaro (Coordinador)	David
Dani	Ángel
Karim	Alejandro

Departamento Marketing	
Daniel (Coordinador)	Álvaro
Karim	

"Stack tecnológico". La utilización de la IA en este documento mejora la redacción del mismo. Todo el material generado con herramientas de inteligencia artificial ha sido revisado cuidadosamente por el equipo CARONTE.