

**“ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO”**

**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**

**CARRERA DE SOFTWARE**

**Aplicaciones Informáticas II**

**Paralelo 8vo “A”**

Víctor Ochoa – 7198

**TEMA**

**Tecnologías**



## DEFINICIÓN DE TECNOLOGÍAS PARA EL PROYECTO

### 1. Lenguajes de Programación

Lenguaje	Ventajas	Desventajas	Recomendación
Python	Excelente para aprendizaje automático, extensa comunidad, bibliotecas puntuales como TensorFlow y Pandas	Rendimiento bajo en comparación con lenguajes compilados	Uso para Microservicios de Procesamiento, Clustering y Predicción. Ya tiene los servicios para ser utilizados.
JavaScript/TypeScript	Ideal para Front-End y Back-End con Node.js, amplio soporte para frameworks de visualización como React	Necesidad de configuración adicional para manejar tareas computacionalmente intensivas	Uso para Front-End y Microservicios con Node.js

#### Selección:

- Python para los microservicios de procesamiento de datos históricos y predicción.
- TypeScript con React en Front-End y Node.js en Back-End.

### 2. IDE's

IDE	Ventajas	Desventajas	Recomendación
VS Code	Ligero, extensiones para múltiples lenguajes, soporte para debugging y control de versiones.	Requiere configuración para proyectos grandes	Para Front-End y Back-End
PyCharm	Soporte avanzado para Python, depuración eficiente.	Pesado en recursos	Para microservicios en Python
IntelliJ IDEA	Soporte integral para Java, escalabilidad para proyectos grandes.	Requiere licencia completa para características avanzadas	Opcional si se usa Java

### Selección:

- VS Code como IDE principal.
- PyCharm para microservicios en Python.

### 3. Frameworks

Framework	Uso	Ventajas	Desventajas	Recomendación
React.js	Front-End	Excelente rendimiento, compatible con mapas interactivos (Leaflet, Mapbox)	Complejidad inicial para desarrolladores nuevos	Seleccionar debido a su poca complejidad y disponibilidad de mapas.
Node.js	Back-End	Excelente para trabajar en sistemas modernos Para proyectos con necesidad de tiempo real	No tan fácil integrar con Python	No tan bueno para el proyecto, pero a tener en cuenta
Django	Back-End	Más robusto que algunos Proyectos con gran complejidad	Poco flexible Más complejo que otros	La complejidad lo descarta debido al tiempo
Flask / FastAPI	Back-End (Python)	Ligero, fácil de integrar con servicios de ML	Menos robusto para aplicaciones grandes en comparación con Django	FastAPI por su mejor rendimiento Flask por su integración con python
Spring Boot	Microservicios (Java)	Ideal para aplicaciones escalables	Requiere curva de aprendizaje mayor	Opcional
RabbitMQ	Bus de Mensajes	Fácil de implementar, escalable	Requiere configuración adicional para alta disponibilidad	Seleccionado

### Selección:

- React.js para Front-End.
- Flask para microservicios de Python.

- RabbitMQ como Bus de Mensajes.

#### 4. Sistemas de Gestión de Bases de Datos

SGBD	Ventajas	Desventajas	Recomendación
PostgreSQL	Soporte para datos geoespaciales (PostGIS), robusto, escalable	Configuración inicial compleja	Para datos de accidentes, predicciones y usuarios
MongoDB	NoSQL, fácil de escalar horizontalmente	Menor eficiencia en transacciones complejas	Opcional para configuraciones de usuario

#### Selección:

- PostgreSQL con extensiones PostGIS para datos históricos y predicciones.
- MongoDB para configuraciones de usuario.