

DOCUMENTO TÉCNICO

Página 1 de 1

UNACEM -**PUCE**

Documentación Técnica

Versión: 1.0 Final

HERRAMIENTAS Y ENTORNO DE EJECUCIÓN

Hackathon PUCE - UNACEM | Equipo GEOINNOVA

PROYECTO

Dashboard Analítico + Modelo
Predictivo de Incidentes

EVENTO

Hackathon PUCE Quito (UNACEM)

EQUIPO

GEOINNOVA

RESPONSABLE TÉCNICO

Johnatan Guacho - Ricardo Carrion

ROL PRINCIPAL

Desarrollo Python (Modelo) + Desarrollo
Web (Dashboard)

ÁREAS COMPLEMENTARIAS

Preparación de datos - Equipo
GEOINNOVA

1. OBJETIVO DEL ANEXO TÉCNICO

El presente documento describe las herramientas, dependencias de software y consideraciones técnicas necesarias para la instalación, ejecución y replicación del trabajo desarrollado durante el Hackathon PUCE Quito en conjunto con UNACEM. La solución incluye un módulo de análisis de datos, un modelo predictivo entrenado y una plataforma web de visualización (dashboard), integrados bajo un enfoque modular.

2. ENTORNO DE DESARROLLO UTILIZADO

COMPONENTE	HERRAMIENTA / TECNOLOGÍA	VERSIÓN	USO
Lenguaje Principal	Python	3.10.8	Entrenamiento y ejecución del modelo

COMPONENTE	HERRAMIENTA / TECNOLOGÍA	VERSIÓN	USO
Editor de Código	Visual Studio Code	Última estable	Desarrollo de scripts y backend
Herramienta Estadística	RStudio	Última estable	Análisis exploratorio y validación
Control de Versiones	Git / GitHub	https://github.com/JohnatanGuacho/geoinnova-unacem	Versionamiento del código y entregables
Frontend Web	HTML5 / CSS3 / JavaScript	ES6+	Dashboard de visualización y UI
Visualización	Chart.js	N/A	Gráficos interactivos y KPI

3. LIBRERÍAS Y DEPENDENCIAS DE PYTHON

LIBRERÍA	FUNCIÓN TÉCNICA	USO
pandas	Procesamiento, transformación y estructuración de datos tabulares	ETL / Features
numpy	Operaciones matemáticas y manipulación de arreglos numéricos	Soporte
scikit-learn	Preprocesamiento, entrenamiento, validación y evaluación de modelos ML	Modelado
joblib	Persistencia de modelos y almacenamiento de objetos serializados	Exportación
matplotlib	Visualización exploratoria de tendencias y validación interna	Ánálisis

LIBRERÍA	FUNCIÓN TÉCNICA	USO
fastapi	Implementación de API REST para consumo del modelo predictivo	Backend
uvicorn	Servidor ASGI para despliegue local y ejecución del servicio FastAPI	Ejecución

4. ARQUITECTURA MODULAR DEL PROYECTO

MÓDULO	DESCRIPCIÓN	TECNOLOGÍAS
Preparación de Datos	Normalización, limpieza y estructuración de datos. Generación de variables finales para entrenamiento y análisis estadístico (sin exposición de datos sensibles).	R / Python
Entrenamiento del Modelo	Implementación de scripts para entrenamiento supervisado, evaluación y exportación del modelo. Se genera un modelo persistente listo para ser consumido mediante API.	Python
Servicio de Predicción (API)	Servicio REST desarrollado para recibir variables de entrada y retornar predicciones estructuradas para el dashboard.	FastAPI
Dashboard Web	Interfaz web interactiva que presenta métricas, visualizaciones y consumo de resultados del modelo predictivo a través de endpoints.	HTML / CSS / JS

5. ESPECIFICACIONES PARA REPLICACIÓN EN ENTORNO UNACEM

Para ejecutar la solución en instalaciones de UNACEM se recomienda un entorno controlado con las siguientes características mínimas:

- **Sistema Operativo:** Windows 10/11 o Linux (Ubuntu 20.04+ recomendado)
- **Procesador:** Intel i5 / Ryzen 5 o superior
- **Memoria RAM:** 8 GB mínimo (16 GB recomendado)
- **Almacenamiento:** 5 GB libres mínimo
- **Python:** 3.10.8 instalado y configurado en PATH
- **Acceso a red local:** requerido únicamente para consumo interno de API (localhost o red interna)

6. CREACIÓN DE ENTORNO CONTROLADO (RECOMENDACIÓN)

Página 1 de 1

PASO	DESCRIPCIÓN
1. Instalación de Python	Instalar Python 3.10.8 desde repositorio oficial y validar versión con el comando: python --version
2. Creación de entorno virtual	Crear un entorno aislado para dependencias del proyecto: python -m venv venv Activación en Windows: venv\Scripts\activate Activación en Linux: source venv/bin/activate
3. Instalación de dependencias	Instalar librerías necesarias desde un archivo requirements.txt: pip install -r requirements.txt
4. Ejecución del API	Levantar el servicio de predicción (FastAPI) con Unicorn: uvicorn main:app --host 0.0.0.0 --port 8000
5. Acceso al Dashboard	Ejecutar el dashboard abriendo el archivo principal HTML en un navegador moderno o mediante un servidor local (Live Server en VS Code o servidor HTTP).
6. Validación	Confirmar que el dashboard consume correctamente los endpoints del API y que la respuesta del modelo se interpreta de manera adecuada para la visualización.

7. CONSIDERACIONES DE SEGURIDAD Y CONFIDENCIALIDAD

La solución fue desarrollada bajo un enfoque de confidencialidad de información. No se incluyen datos sensibles en el repositorio, ni se exponen registros individuales en la interfaz. El sistema trabaja únicamente con variables agregadas y estructuras de datos previamente anonimizadas, respetando las restricciones establecidas por UNACEM.