

DOCUMENTO TÉCNICO

# PLATAFORMA WEB, DASHBOARD E INTEGRACIÓN

Hackathon PUCE - UNACEM | Equipo GEOINNOVA

<b>NOMBRE DEL PROYECTO</b> Dashboard Analítico + Integración Predictiva	<b>EVENTO</b> Hackathon PUCE - UNACEM
<b>TIPO DE SOLUCIÓN</b> Plataforma Web Analítica (BI) + Modelo Predictivo Integrado	<b>ENFOQUE</b> Modular, Escalable y Orientado a Decisión Ejecutiva
<b>EQUIPO</b> GEOINNOVA	<b>RESPONSABLE TÉCNICO</b> Johnatan Guacho Ricardo Carrion
<b>CONFIDENCIALIDAD</b> Datos Restringidos / Uso Interno UNACEM	<b>ESTADO DEL ENTREGABLE</b> Final / Prototipo Funcional

## 1. INTRODUCCIÓN

El presente documento describe de manera técnica y estructurada el desarrollo de la **plataforma web** implementada durante la Hackathon PUCE - UNACEM, enfocada en la visualización de indicadores clave (KPIs), tendencias temporales y análisis comparativo de incidentes reportados.

La solución fue construida con un enfoque de **inteligencia de negocios (BI)** y analítica de datos, permitiendo transformar información compleja en gráficos y reportes comprensibles para usuarios técnicos y ejecutivos.

Adicionalmente, se integró un **modelo predictivo** entrenado previamente, el cual permite estimar escenarios futuros mediante patrones históricos, complementando el dashboard con capacidades de análisis prospectivo.

**Restricción de datos:** Por políticas de confidencialidad establecidas por UNACEM, este documento no expone datos reales, valores numéricos exactos, registros originales ni identificadores específicos. Se describe el funcionamiento en términos de variables, estructura y flujo de procesamiento.

2. OBJETIVO GENERAL DEL DASHBOARD

La plataforma web fue desarrollada con el objetivo de brindar a UNACEM una herramienta interactiva que permita analizar incidentes reportados desde un enfoque estratégico, identificando patrones, riesgos y áreas críticas mediante visualizaciones.

2.1 OBJETIVOS ESPECÍFICOS

- Presentar indicadores clave (KPIs) consolidados de reportes e incidentes.
- Facilitar el análisis temporal mediante filtros por año y trimestre.
- Identificar tendencias de ocurrencia por tipo de evento, categoría y sitio.
- Implementar comparaciones entre periodos para medir evolución y comportamiento.
- Integrar predicciones automáticas mediante un modelo entrenado.
- Generar una interfaz clara, moderna y ejecutiva para toma de decisiones.
- Garantizar que la solución sea escalable y adaptable a producción.

3. ARQUITECTURA GENERAL DE LA SOLUCIÓN

La solución fue diseñada bajo un enfoque modular, separando componentes de forma que cada parte sea independiente, reutilizable y fácilmente actualizable. Esto permite a UNACEM modificar o ampliar el sistema en el futuro sin necesidad de reconstruir todo el proyecto.

3.1 COMPONENTES PRINCIPALES

COMPONENTE	FUNCIÓN	TECNOLOGÍAS		DESCRIPCIÓN TÉCNICA
Frontend Web	Visualización del dashboard	HTML JS	CSS	Interfaz gráfica responsiva con filtros, KPIs y gráficas interactivas para análisis ejecutivo.
Motor de Visualización	Gráficos dinámicos	CHART.JS		Renderizado de gráficos de barras, líneas, doughnut y comparativos para indicadores clave.
Resultados Analíticos	Datos procesados	JSON	CSV	Datos estructurados (agregados y anonimizados) generados mediante scripts analíticos y exportados al dashboard.

COMPONENTE	FUNCIÓN	TECNOLOGÍAS	DESCRIPCIÓN TÉCNICA
Página 1 de 1			
API Predictiva	Ejecutar el modelo	PYTHON   REST	Servicio backend que carga el modelo entrenado y expone un endpoint para predicciones bajo demanda.
Modelo Entrenado	Predicción de escenarios	ML JOBLIB/PICKLE	Modelo entrenado con datos históricos procesados y exportado en formato serializado para inferencia.

**Diseño modular:** La separación por capas (visualización, datos y predicción) permite mantener independencia técnica, facilitando que UNACEM pueda integrar una base de datos real, actualizar el modelo o desplegar el sistema en servidores internos.

## 4. TECNOLOGÍAS Y HERRAMIENTAS DE DESARROLLO

La plataforma web fue implementada utilizando tecnologías estándar, priorizando portabilidad, compatibilidad y facilidad de despliegue en entornos corporativos. Se evitó el uso de frameworks pesados para asegurar que el sistema pueda ejecutarse sin dependencias complejas.

### 4.1 STACK PRINCIPAL

- **HTML5:** estructura del dashboard y secciones de visualización.
- **CSS3:** estilos, diseño responsivo y presentación profesional.
- **JavaScript:** lógica del frontend, filtros, carga de datos y consumo de API.
- **Chart.js:** generación de gráficos interactivos.
- **Python:** integración con el modelo predictivo y ejecución de predicciones.
- **API REST:** canal de comunicación entre frontend y backend.
- **Git/GitHub:** versionamiento y control del proyecto.

### 4.2 PRINCIPIOS APLICADOS EN EL DESARROLLO

- **Compatibilidad:** el dashboard puede abrirse en navegadores modernos sin instalaciones adicionales.
- **Portabilidad:** el proyecto puede ejecutarse localmente o en un servidor interno.
- **Escalabilidad:** estructura lista para conectar con una base de datos real.
- **Seguridad:** no se expone información sensible en frontend.
- **Separación de responsabilidades:** predicción independiente de visualización.

## 5. DISEÑO FUNCIONAL DEL DASHBOARD

El dashboard fue construido con un enfoque orientado a usuarios ejecutivos y técnicos, buscando ofrecer una visión clara del comportamiento de incidentes, su distribución y tendencias temporales.

5.1 ELEMENTOS PRINCIPALES DE LA INTERFAZ

ELEMENTO	DESCRIPCIÓN	FINALIDAD
KPIs / Indicadores	Tarjetas con métricas consolidadas y resumen general.	Entrega rápida de información clave para decisión ejecutiva.
Gráficos comparativos	Barras, doughnut y gráficos de distribución por variables.	Identificar factores críticos por tipo, sitio o categoría.
Series temporales	Gráficos de línea y tendencias por trimestre.	Detectar incremento o reducción de incidentes por periodo.
Filtros dinámicos	Selección de año, periodo o categoría analítica.	Permitir análisis segmentado y comparativo por escenario.
Semáforo de desempeño	Clasificación cualitativa: Óptimo / Adecuado / Mejorable.	Interpretación rápida para identificar áreas críticas.
Módulo de predicción	Interfaz que consulta el modelo predictivo y muestra resultados.	Proyección de escenarios futuros como herramienta preventiva.

**Enfoque estratégico:** el dashboard fue diseñado para convertir información compleja en resultados interpretables, priorizando claridad visual, lectura ejecutiva y reducción de carga técnica para el usuario final.

6. FLUJO DE CARGA Y PROCESAMIENTO DE DATOS

Para garantizar seguridad y rendimiento, el dashboard consume información procesada y estructurada, evitando el uso de datos crudos directamente en la interfaz. Los datos fueron previamente transformados a formatos optimizados para visualización (por ejemplo, JSON y CSV agregados).

6.1 ESTRATEGIA DE ESTRUCTURACIÓN DE DATOS

- Transformación de datos en estructuras agregadas por periodo (ejemplo: año-trimestre).
- Consolidación por variables categóricas (tipo, sitio, categoría, género, etc.).

- Exportación de tablas analíticas a formato JSON para consumo rápido desde JavaScript.
- Reducción de registros individuales para evitar exposición de información sensible.

**Optimización:** El uso de JSON permite que el dashboard cargue información de manera rápida y renderice gráficas sin retrasos, incluso en entornos con hardware limitado.

## 7. INTEGRACIÓN CON EL MODELO PREDICTIVO

Uno de los elementos principales del sistema es la integración con un modelo predictivo entrenado previamente, el cual permite generar estimaciones futuras sobre incidentes bajo un enfoque preventivo.

### 7.1 ARQUITECTURA DE INTEGRACIÓN

La comunicación entre el dashboard y el modelo se implementó mediante una API REST, donde el frontend realiza solicitudes HTTP enviando variables de entrada, y recibe como respuesta predicciones estructuradas.

### 7.2 FLUJO DE PREDICCIÓN

- El usuario define parámetros dentro del dashboard (periodo, categoría, sitio, tipo, etc.).
- El frontend genera un JSON estructurado con dichas variables.
- Se envía una solicitud al endpoint del backend.
- El backend carga el modelo entrenado y ejecuta inferencia.
- El backend devuelve una respuesta con predicción y nivel de riesgo.
- El dashboard presenta el resultado con gráficos, KPIs o semáforos.

### 7.3 EJEMPLO DE COMUNICACIÓN API (ESTRUCTURAL)

```
POST /predict HTTP/1.1 Host: localhost:8000 Content-Type: application/json { "periodo": "YYYY-T#", "categoria_evento": "CATEGORIA", "tipo_evento": "TIPO", "sitio": "SITIO", "tipo_empleado": "EMPLEADO|CONTRATISTA" } RESPONSE: { "prediccion": "VALOR_ESTIMADO", "nivel_riesgo": "BAJO|MEDIO|ALTO", "mensaje": "INTERPRETACION_GENERAL" }
```

**Desacoplamiento:** El dashboard no contiene lógica interna del modelo; únicamente consume el resultado. Esto permite reemplazar el modelo en el futuro sin reescribir el frontend.

## 8. INTERPRETACIÓN Y PRESENTACIÓN DE RESULTADOS

El dashboard fue construido para que los resultados analíticos sean interpretables por usuarios no técnicos. Por ello, se implementaron mecanismos visuales y semánticos como semáforos, rankings y KPIs.

### 8.1 TIPOS DE VISUALIZACIÓN IMPLEMENTADOS

- **Gráficos de barras:** comparaciones entre categorías o tipos.
- **Gráficos doughnut:** distribución porcentual de eventos.
- **Gráficos de línea:** tendencias temporales por trimestre.
- **Ranking Top-N:** identificación de factores con mayor recurrencia.
- **Semáforo de riesgo:** clasificación interpretativa para decisión rápida.

8.2 CLASIFICACIÓN CUALITATIVA (SEMÁFORO)

CLASIFICACIÓN	INTERPRETACIÓN	ACCIÓN RECOMENDADA
ÓPTIMO	Comportamiento controlado y bajo nivel de recurrencia.	Continuar monitoreo y mantener estrategias preventivas actuales.
ADECUADO	Se detectan patrones moderados o incrementos leves.	Revisión preventiva, refuerzo de controles y seguimiento periódico.
MEJORABLE	Existe recurrencia significativa o tendencia creciente.	Acciones correctivas inmediatas y análisis detallado del área o causa.

9. SEGURIDAD Y CONFIDENCIALIDAD DE LA PLATAFORMA

Debido a que la información proporcionada por UNACEM se considera sensible y de uso interno, se establecieron prácticas para evitar exposición de datos y garantizar que el dashboard cumpla con criterios de confidencialidad.

9.1 PRINCIPALES MEDIDAS APLICADAS

- Uso de datos agregados en lugar de registros individuales.
- No publicación de identificadores personales o campos sensibles.
- Separación del dataset real y el dashboard público.
- Modelo predictivo utilizado únicamente para inferencia, sin exposición de entrenamiento.
- Arquitectura orientada a despliegue interno en instalaciones UNACEM.

**Importante:** Se recomienda que la solución final sea desplegada únicamente en un entorno privado (red interna UNACEM o servidor corporativo), evitando exposición en servidores públicos sin autenticación.

10. REQUISITOS TÉCNICOS PARA EJECUCIÓN EN INSTALACIONES UNACEM

Con el objetivo de permitir que el equipo técnico de UNACEM pueda emular y ejecutar la solución desarrollada, se detallan los requisitos mínimos y recomendaciones para despliegue local o corporativo.

10.1 REQUISITOS DE HARDWARE RECOMENDADOS

Página 1 de 1

RECURSO	MÍNIMO RECOMENDADO	ÓPTIMO RECOMENDADO
CPU	Intel i3 / Ryzen 3 (2 núcleos)	Intel i5 / Ryzen 5 (4 núcleos o más)
RAM	8 GB	16 GB
Almacenamiento	5 GB libres	20 GB libres (para logs y datasets adicionales)
Sistema Operativo	Windows 10/11 o Linux	Linux Server / Windows Server
Red	LAN interna estable	Servidor interno accesible desde red corporativa

10.2 REQUISITOS DE SOFTWARE

- **Python:** versión 3.10.x (recomendado 3.10.8 para compatibilidad exacta).
- **Dependencias:** instalación mediante requirements.txt.
- **Navegador:** Chrome / Edge actualizado.
- **Servidor local opcional:** (para servir frontend) python -m http.server.

10.3 CONFIGURACIÓN DE ENTORNO CONTROLADO (RECOMENDADO)

Para garantizar replicabilidad, se recomienda crear un entorno virtual de Python, aislando dependencias del sistema operativo. Esto permite a UNACEM ejecutar el modelo sin conflictos con otras instalaciones de Python.

```
# CREACIÓN DE ENTORNO VIRTUAL python -m venv venv # ACTIVACIÓN EN WINDOWS
venv\Scripts\activate # ACTIVACIÓN EN LINUX / MAC source venv/bin/activate # INSTALACIÓN
DE DEPENDENCIAS pip install -r requirements.txt # EJECUCIÓN DE LA API (EJEMPLO) python
app.py
```

10.4 EJECUCIÓN DEL DASHBOARD

El dashboard puede ejecutarse de dos maneras:

- **Modo local:** abrir el archivo index.html directamente desde el navegador.
- **Modo servidor interno:** alojar el dashboard en un servidor HTTP dentro de la red corporativa.

```
# OPCIÓN SERVIDOR LOCAL SIMPLE cd dashboard python -m http.server 5500 # Luego acceder
en navegador: http://localhost:5500
```

**Recomendación UNACEM:** En un entorno corporativo real, se recomienda desplegar la API del modelo en un servidor interno, y el dashboard como un portal web accesible desde la intranet institucional.

Página 1 de 1

## 11. ESCALABILIDAD Y PROPUESTA DE IMPLEMENTACIÓN CORPORATIVA

La solución fue concebida como un prototipo funcional, sin embargo su diseño modular permite una transición hacia un entorno productivo real dentro de UNACEM. Esto significa que el sistema puede evolucionar de un dashboard local a una plataforma corporativa de análisis.

### 11.1 ESCENARIOS DE ESCALABILIDAD

- Conexión directa a base de datos (SQL Server / PostgreSQL / Oracle) para actualización automática.
- Integración con sistemas internos de reportes y gestión de incidentes.
- Automatización de pipeline de datos (ETL) para procesamiento periódico.
- Despliegue en servidor interno con control de acceso (usuarios y roles).
- Integración con autenticación corporativa (Active Directory / LDAP).

### 11.2 MEJORAS FUTURAS SUGERIDAS

MEJORA	DESCRIPCIÓN	IMPACTO	PRIORIDAD
Autenticación y roles	Implementar login y permisos para restringir acceso por áreas y perfiles.	Alto	ALTA
Base de datos corporativa	Reemplazar JSON por conexión directa a BD para datos en tiempo real.	Alto	ALTA
Automatización ETL	Crear pipeline automático de limpieza y carga de datos semanal/mensual.	Medio	MEDIA
Logs y auditoría	Registrar accesos y consultas para trazabilidad y control interno.	Medio	MEDIA
Explicabilidad del modelo	Incluir interpretabilidad (SHAP/LIME) para justificar predicciones.	Alto	MEDIA



## 12. RESUMEN EJECUTIVO TÉCNICO

Página 1 de 1

**3****MÓDULOS  
PRINCIPALES****BI****ENFOQUE  
ANALÍTICO****REST****INTEGRACIÓN  
MODELO****MOD****ARQUITECTURA  
MODULAR**

**Resultado:** El sistema implementado permite visualizar información de forma ejecutiva mediante dashboards interactivos, y complementa el análisis con predicciones automatizadas, siendo una herramienta útil para prevención, monitoreo y toma de decisiones.

## 13. CONCLUSIONES

La plataforma web desarrollada en la Hackathon PUCE - UNACEM cumple con los objetivos propuestos, ofreciendo una solución analítica clara, modular y orientada a usuarios corporativos.

El dashboard permite transformar datos en indicadores comprensibles mediante visualizaciones dinámicas, mientras que la integración con el modelo predictivo añade una capa de valor estratégico al permitir proyecciones futuras.

Finalmente, debido a la estructura modular del sistema, UNACEM puede replicar y desplegar el proyecto en un entorno controlado, así como escalarlo hacia una plataforma corporativa con autenticación, base de datos y automatización de procesamiento de información.