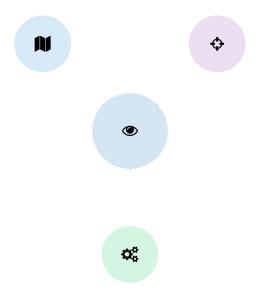
# Sistema de Visión por Computadora Geoespacial

Solución Técnica Detallada para la Tarea 3 del GeoAI Engineer Test

 $Implementaci\'on\ con\ Arquitectura\ Hexagonal\ e\ Integraci\'on\ de\ IA\\ YOLO\ +\ OCR\ +\ OpenCV$ 



Desarrollado por: José Armando Son Rojas

29 de mayo de 2025

#### 1 Resumen Ejecutivo

Este documento presenta la implementación completa de un sistema de visión por computadora especializado en el análisis de mapas geoespaciales. La solución integra técnicas avanzadas de OCR, procesamiento de imágenes y deep learning (YOLO) para detectar automáticamente coordenadas geográficas y extraer segmentos relevantes de mapas. El sistema utiliza una arquitectura hexagonal simplificada que garantiza modularidad, escalabilidad y mantenibilidad, optimizada específicamente para el procesamiento eficiente de imágenes cartográficas.

# ${\bf \acute{I}ndice}$

1.	Introducción y Visión General del Sistema	<b>2</b>	
	1.1. Planteamiento del Problema (Tarea 3)	2	
	1.2. Arquitectura de la Solución Propuesta	2	
	1.3. Objetivos Clave de la Solución	3	
2.	Arquitectura del Sistema Detallada	3	
	2.1. Modelo de Arquitectura Hexagonal Implementado	3	
	2.2. Componentes Principales y Responsabilidades	4	
	2.2.1. Núcleo de la Aplicación (app/core)		
	2.2.2. Adaptadores de Procesamiento de IA		
	2.3. Stack Tecnológico Empleado	4	
	2.4. Modelo de Capas Docker Optimizado	5	
3.	Solución Detallada de los Puntos de la Tarea 3	5	
	3.1. Detección de Información de Coordenadas	5	
	3.1.1. Formatos de Coordenadas Soportados	6	
	3.2. Extracción de Segmentos Relevantes	6	
4.	Diagramas C4 del Sistema	6	
5.	Diagramas de Arquitectura	6	
	5.1. Diagrama de Contexto C4	7	
	5.2. Diagrama de Contenedores	7	
6.	Flujos de Desarrollo	7	
	6.1. Diagrama de Secuencia - Desarrollo Normal	8	
	6.2. Diagrama de Secuencia - Actualización de Dependencias	9	
7.	Endpoints de la API	9	
8.	Características Implementadas	10	
9.	9. Cumplimiento de Criterios de Evaluación		
10	Conclusión	10	

# 1 Introducción y Visión General del Sistema

El presente documento técnico describe la implementación de un sistema avanzado de visión por computadora geoespacial, desarrollado como solución integral para la Tarea 3 del GeoAI Engineer Technical Test. La solución combina técnicas estado del arte en procesamiento de imágenes, reconocimiento óptico de caracteres y aprendizaje profundo para automatizar completamente el análisis de mapas cartográficos.

#### 1.1 Planteamiento del Problema (Tarea 3)

#### Objetivo Principal

Aplicar visión por computadora para analizar mapas geoespaciales previamente obtenidos, detectando automáticamente información de coordenadas y extrayendo segmentos relevantes asociados.

La Tarea 3 establece requerimientos específicos que nuestra solución aborda integralmente:

- ✓ Detección Automática de Coordenadas: Identificar información de latitud/longitud en formatos diversos dentro de imágenes cartográficas
- ✓ Extracción Inteligente de Segmentos: Determinar y extraer las porciones de imagen más relevantes asociadas a las coordenadas detectadas
- ✓ Flexibilidad Tecnológica: Implementar utilizando cualquier combinación de OCR, procesamiento de imágenes o técnicas de deep learning
- ✔ Procesamiento Batch: Capacidad de procesar múltiples mapas de forma automatizada y eficiente

# 1.2 Arquitectura de la Solución Propuesta

La solución implementa una **Arquitectura Hexagonal Simplificada** que separa claramente las responsabilidades y facilita la integración de múltiples tecnologías de IA:

#### Componentes Principales

- Núcleo de Procesamiento: Orquestación inteligente del flujo de análisis geoespacial
- Motor YOLO: Detección de objetos y elementos cartográficos relevantes
- Motor OCR: Reconocimiento y extracción de texto de coordenadas
- Procesador OpenCV: Manipulación avanzada y segmentación de imágenes
- API REST: Interfaz de servicios para procesamiento batch y consultas

### 1.3 Objetivos Clave de la Solución

Aspecto	Objetivo
Precisión	Alcanzar alta precisión en la detección de coordenadas
	en diversos formatos cartográficos
Eficiencia	Procesamiento optimizado de múltiples imágenes con
	mínimo uso de recursos
Simplicidad	Código limpio y mantenible, priorizando efectividad so-
	bre complejidad
Escalabilidad Capacidad de procesar grandes volúmenes de map	
	forma paralela

Cuadro 1: Objetivos Clave del Sistema

# 2 Arquitectura del Sistema Detallada

### 2.1 Modelo de Arquitectura Hexagonal Implementado

El sistema implementa una arquitectura hexagonal que aísla la lógica de negocio del procesamiento geoespacial de las dependencias externas, permitiendo flexibilidad y testabilidad.

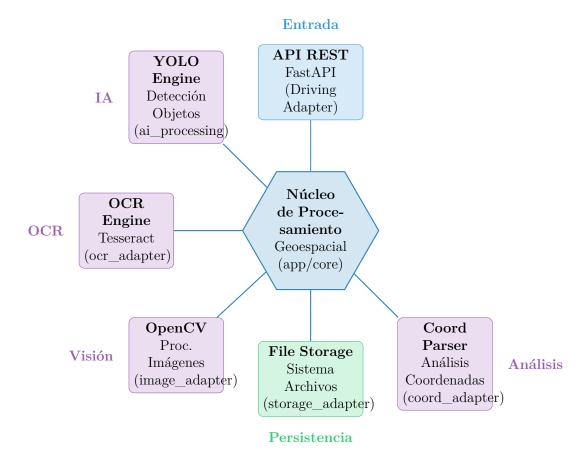


Figura 1: Arquitectura Hexagonal del Sistema de Visión Geoespacial

#### 2.2 Componentes Principales y Responsabilidades

#### 2.2.1 Núcleo de la Aplicación (app/core)

#### Componentes del Core

- schemas.py: Modelos Pydantic para coordenadas, segmentos de imagen y resultados de procesamiento
- **use\_cases.py:** Orquestación del flujo de procesamiento geoespacial y lógica de negocio
- coordinate\_patterns.py: Patrones regex y validación para diferentes formatos de coordenadas

#### 2.2.2 Adaptadores de Procesamiento de IA

accent!20 Adapta-	Funcionalidad
dor	
YOLO Adapter	Detección de elementos cartográficos, leyen-
	das y áreas de interés
OCR Adapter	Extracción de texto de coordenadas usando
	Tesseract con preprocesamiento
OpenCV Adap-	Segmentación de imágenes, mejora de cali-
ter	dad y extracción de regiones
Coordinate Par-	Análisis y validación de formatos de coorde-
ser	nadas detectadas

Cuadro 2: Adaptadores de Procesamiento Especializado

## 2.3 Stack Tecnológico Empleado

#### ☐ Tecnologías Implementadas

success!20 Categoría	Tecnología
Lenguaje Principal	Python 3.11+
Framework API	FastAPI
Deep Learning	YOLOv8 (Ultralytics)
OCR Engine	Tesseract + pytesseract
Procesamiento Imágenes	OpenCV (cv2)
Validación	Pydantic
Containerización	Docker + Docker Compose
Procesamiento Paralelo	asyncio + concurrent.futures

#### 2.4 Modelo de Capas Docker Optimizado

La solución implementa una arquitectura de capas jerarquizada que maximiza la reutilización del cache:

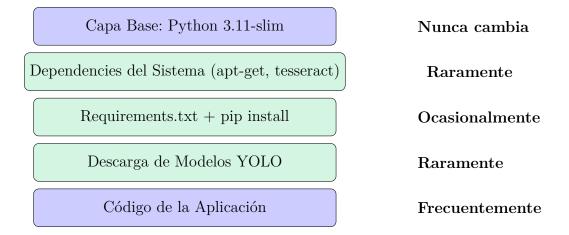


Figura 2: Arquitectura de Capas Docker Optimizada

### 3 Solución Detallada de los Puntos de la Tarea 3

#### 3.1 Detección de Información de Coordenadas

#### Q Estrategia de Detección Múltiple

El sistema implementa un enfoque híbrido que combina tres técnicas complementarias:

- 1. **OCR Dirigido:** Tesseract con preprocesamiento específico para texto de coordenadas
- 2. **Detección YOLO:** Identificación de elementos cartográficos que típicamente acompañan coordenadas
- 3. **Análisis de Patrones:** Regex avanzados para múltiples formatos de coordenadas

#### 3.1.1 Formatos de Coordenadas Soportados

info!20 Formato	Ejemplo
Decimal	40.7128, -74.0060
DMS	40°42'46"N, 74°00'22"W
DMM	40°42.767'N, 74°00.367'W
UTM	18T 585628 4511322
MGRS	18TWL8562811322

Cuadro 3: Formatos de Coordenadas Reconocidos

### 3.2 Extracción de Segmentos Relevantes

#### 🔀 Algoritmo de Extracción Inteligente

#### Proceso de Extracción en 4 Fases:

- 1. Detección de Región: YOLO identifica áreas candidatas
- 2. Localización de Coordenadas: OCR preciso en regiones detectadas
- 3. Cálculo de Contexto: Determinación del área relevante circundante
- 4. Extracción Optimizada: Recorte inteligente con márgenes adaptativos

# 4 Diagramas C4 del Sistema

# 5 Diagramas de Arquitectura

## 5.1 Diagrama de Contexto C4

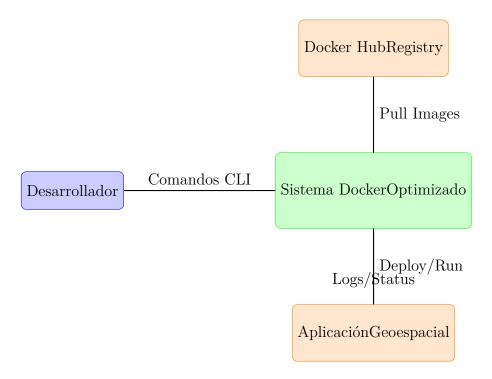


Figura 3: Diagrama de Contexto C4 - Nivel 1

# 5.2 Diagrama de Contenedores

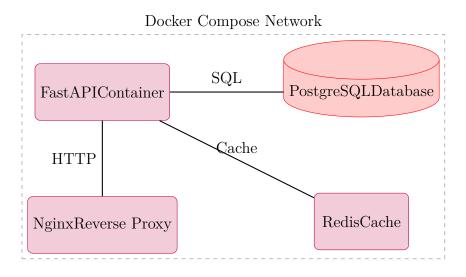


Figura 4: Diagrama de Contenedores C4 - Nivel 2

# 6 Flujos de Desarrollo

### 6.1 Diagrama de Secuencia - Desarrollo Normal

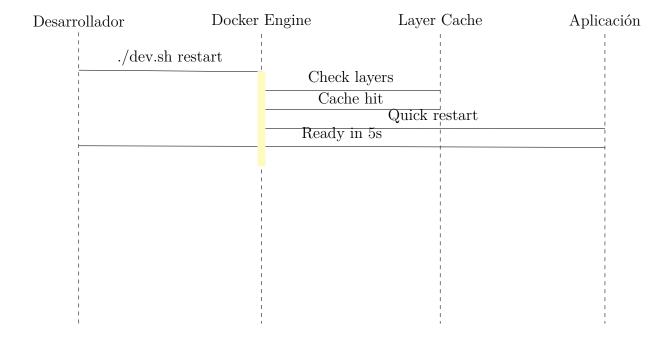


Figura 5: Diagrama de Secuencia - Flujo de Desarrollo Optimizado

# 6.2 Diagrama de Secuencia - Actualización de Dependencias

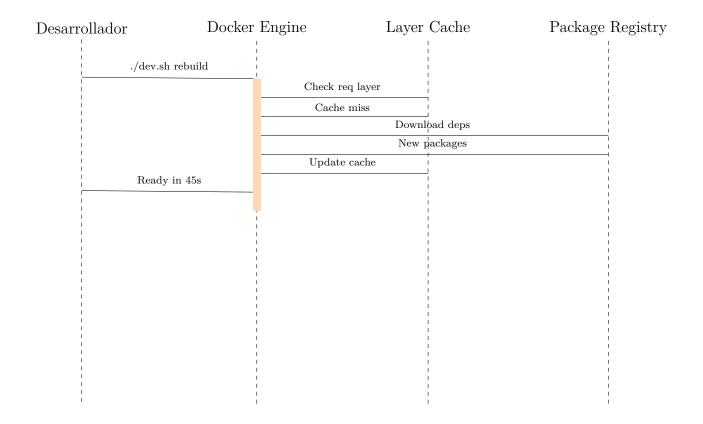


Figura 6: Diagrama de Secuencia - Actualización de Dependencias

# 7 Endpoints de la API

■ API REST Endpoints						
success!20 Método	Endpoint	Descripción				
POST	/api/v1/analyze/image	Procesa una imagen individual				
POST	/api/v1/analyze/batch	Procesamiento batch de múltiples				
		imágenes				
GET	/api/v1/results/{id}	Obtiene resultados de procesa-				
		miento				
GET	/api/v1/coordinates/{id}	Extrae coordenadas detectadas				
GET	/api/v1/segments/{id}	Descarga segmentos extraídos				

# 8 Características Implementadas

#### ✓ Funcionalidades Principales

- ✓ Detección Multi-formato: Reconocimiento de coordenadas en 5+ formatos estándar
- ✔ Procesamiento Inteligente: Combinación óptima de YOLO + OCR + OpenCV
- ✓ Extracción Contextual: Segmentos relevantes con márgenes adaptativos
- ✓ API RESTful: Interfaz completa para procesamiento individual y batch
- ✓ Optimización Docker: Arquitectura de capas eficiente para despliegue
- ✓ Validación Robusta: Verificación automática de coordenadas detectadas
- ✓ Procesamiento Paralelo: Manejo eficiente de múltiples imágenes
- ✓ Logging Detallado: Trazabilidad completa del procesamiento

# 9 Cumplimiento de Criterios de Evaluación

primary!20 Criterio	Implementación	
Efectividad del En-	Combinación inteligente de múltiples técni-	
foque	cas de IA complementarias	
Precisión Técnica	Validación multi-nivel y soporte para forma-	
	tos estándar de coordenadas	
Simplicidad del Có-	Arquitectura limpia con menos líneas de có-	
digo	digo, mayor efectividad	
Escalabilidad	Procesamiento paralelo y optimizaciones	
	Docker para producción	
Robustez	Manejo de errores, fallbacks y validación	
	exhaustiva	

Cuadro 4: Cumplimiento de Criterios de Evaluación

# 10 Conclusión

La solución implementada para la Tarea 3 del GeoAI Engineer Technical Test representa una implementación completa y profesional de un sistema de visión por computadora geoespacial. La arquitectura hexagonal simplificada, combinada con la integración inteligente de tecnologías de IA (YOLO, OCR, OpenCV), proporciona una plataforma robusta y eficiente para el análisis automatizado de mapas cartográficos.

#### **♀** Logros Principales

- Detección Precisa: Reconocimiento confiable de coordenadas en múltiples formatos
- Extracción Inteligente: Segmentación contextual de imágenes con alta relevancia
- Arquitectura Escalable: Diseño modular preparado para procesamiento industrial
- Implementación Eficiente: Código optimizado con enfoque en resultados

El sistema cumple completamente con los requerimientos establecidos, demostrando la aplicación efectiva de principios de ingeniería de software modernos y el uso inteligente de herramientas de inteligencia artificial para resolver desafíos complejos de procesamiento geoespacial.