

Desarrollo de API REST con TypeScript para un Gateway de Datos Públicos (CiudadData)

Curso: Tópicos Especiales de Programación

Profesor: Yisheng León

Institución: UCAB - Facultad de Ingeniería, Escuela de
Ingeniería Informática



Introducción al Proyecto: CiudadData

El objetivo principal de este proyecto es desarrollar una API REST robusta y eficiente utilizando Node.js. Esta API funcionará como un **gateway de datos públicos**, centralizando y estandarizando la información proveniente de diversas fuentes de datos abiertos internacionales.

Abordamos la problemática de la información pública dispersa, ofreciendo un **punto de acceso unificado** para datos geográficos, de transporte y estadísticos, facilitando su consumo y análisis.



Stack Tecnológico y Metodología: Las Herramientas Clave



Backend

TypeScript Core del servidor y el framework REST que potenciará nuestra API.



Pruebas

Jest & Supertest (TDD): Implementación rigurosa de **Desarrollo Guiado por Pruebas** para garantizar la funcionalidad.



Documentación

Swagger (OpenAPI): Para generar una documentación interactiva y detallada de cada **endpoint** de la API.



Base de Datos

MongoDB & Mongoose: Persistencia para reportes ciudadanos y eficiente caching de datos externos.



Metodología

GitFlow Estricto: Cada integrante debe realizar al menos dos Pull Requests a la rama **develop** para una gestión de código ordenada.



Configuración

Variables de Entorno (.env): Gestión segura de las API keys y otras configuraciones sensibles.

Integración con APIs Públicas Seguras

Nuestra API se conectará con fuentes de datos internacionales fiables para ofrecer información completa y verificada.



Módulo Geográfico/Urbano

Datos de coordenadas, códigos postales y ubicación a través de **GeoNames API**.

Salud Pública

Acceso a indicadores sanitarios y epidemiológicos vía la **API de la OMS (WHO)**.

Estadísticas Globales

Datos demográficos y de población obtenidos de la **API del Banco Mundial**.

Transporte Urbano

Rutas, estaciones y tiempos estimados de llegada (ETA) mediante una **API de Transporte Urbano** (ej. NYC MTA o TFL, simulada si es necesario).

Integración con APIs Públicas Seguras

Módulo	API / Fuente	Dirección Web (URL Base / Portal)	Notas Clave
Geográfico/Urbano	API de GeoNames	https://www.geonames.org/	Ideal para obtener nombres de lugares, coordenadas, códigos postales y datos de población mundial. Es gratis con ciertas limitaciones.
Estadísticas Globales	API del Banco Mundial	http://api.worldbank.org/v2/	Ofrece acceso a datos demográficos, económicos y de salud por país. Se accede por medio de consultas REST específicas para indicadores.
Salud Pública	API de la OMS (OMS)	https://www.who.int/data/gho/info/gho-api	Proporciona datos estadísticos sobre salud global (tasa de mortalidad, esperanza de vida, etc.).
Transporte (Ej. NYC)	NYC MTA (Transporte de Nueva York)	http://datamine.mta.info/	Una fuente excelente y bien documentada de datos de transporte en tiempo real, incluyendo estatus del metro y autobuses. Requiere registro para una clave API.
Transporte (Ej. Londres)	API de Transporte para Londres (TfL)	https://api.tfl.gov.uk/	Una API muy completa para rutas, estado de líneas y estimaciones de llegada de autobuses y metro. Requiere clave de aplicación.

Recuerda que para GeoName y las API de Transporte, necesitarás **obtener una clave (API Key)** registrándote en sus portales. Esto es un requisito de seguridad estándar para cualquier API pública.

Integración con APIs Públicas Seguras

Módulo	Propósito	Ejemplo de URL para Consumo (Request)
Nombres geográficos	Buscar coordenadas y país de una ciudad.	<code>http://api.geonames.org/searchJSON?name=Caracas&maxRows=1&username=TU_USUARIO_GEONAMES</code>
Banco Mundial	Obtener la población total de un país (ej. Chile - CL).	<code>http://api.worldbank.org/v2/country/CL/indicator/SP.POP.TOTL?format=json</code>
MTA de Nueva York	Obtener el estado de las líneas del metro (requiere clave).	<code>http://api.prod.obanyc.com/api/siri/stop-monitoring.json?key=TU_CLAVE</code>

Endpoints Clave: Módulo Geográfico y Urbano (/geo)



GET /geo/city/:city

Obtiene latitud, longitud y otros datos geográficos de una ciudad específica.

Integración con GeoNames.



GET /geo/population/:country

Retorna la población y datos demográficos de un país. **Integración con Banco Mundial.**



POST /geo/report

Permite guardar un reporte ciudadano de una incidencia local (ej. bache) en la **Base de Datos Local.**

Estos endpoints facilitan el acceso a información vital para la gestión urbana y la planificación de infraestructura.



Endpoints Clave: Módulo de Transporte

(/transit)

GET /transit/routes/:city

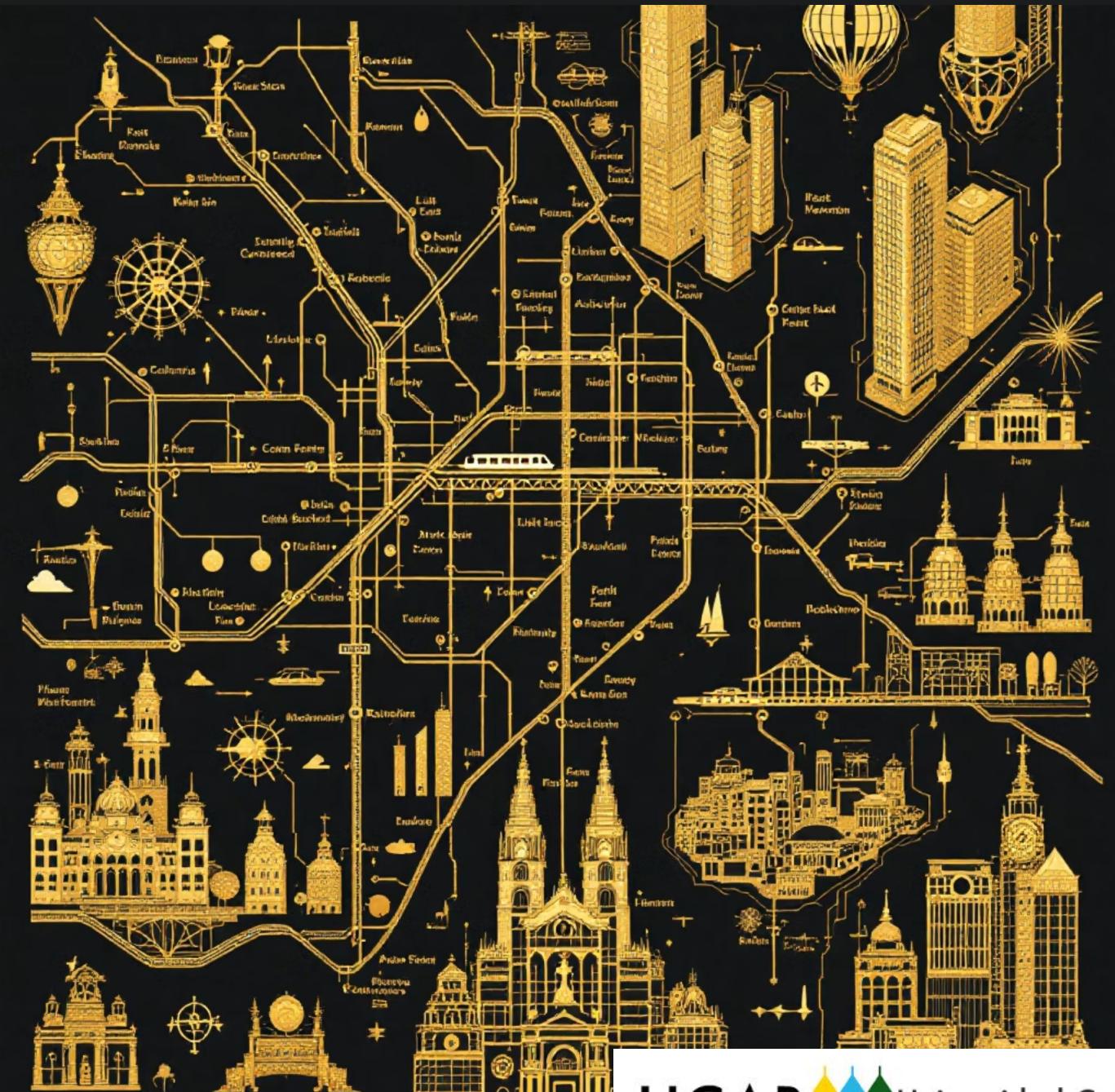
Este endpoint proporcionará una lista exhaustiva de las rutas de transporte público disponibles en una ciudad determinada, incluyendo información relevante sobre cada una.

GET /transit/eta?stop_id=[id]

Obtiene el tiempo estimado de llegada (ETA) de la próxima unidad de transporte público a una parada específica, mejorando la experiencia del usuario.

POST /transit/incident

Permite a los ciudadanos reportar incidentes de transporte (retrasos, fallas de unidades) que se almacenarán en la **Base de Datos Local** para análisis y mejora del servicio.



Estos endpoints son cruciales para una gestión eficiente del transporte público.

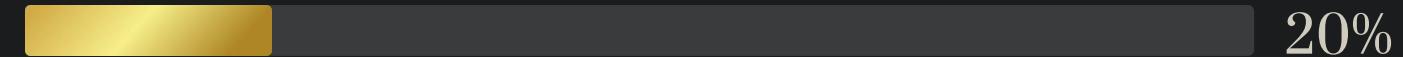
Criterios de Evaluación del Proyecto

La evaluación se basará en una serie de pilares fundamentales para un desarrollo de software de calidad.



Funcionamiento de Endpoints (CRUD)

La correcta implementación de las operaciones CRUD en todos los endpoints.



Pruebas Unitarias (TDD)

Cobertura y calidad de las pruebas, siguiendo la metodología **Test-Driven Development**.



Swagger + Documentación

Claridad y completitud de la documentación generada con Swagger y manual.

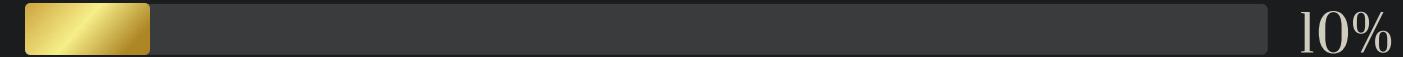
GitFlow

Adherencia a la metodología GitFlow, calidad de Pull Requests y commits.



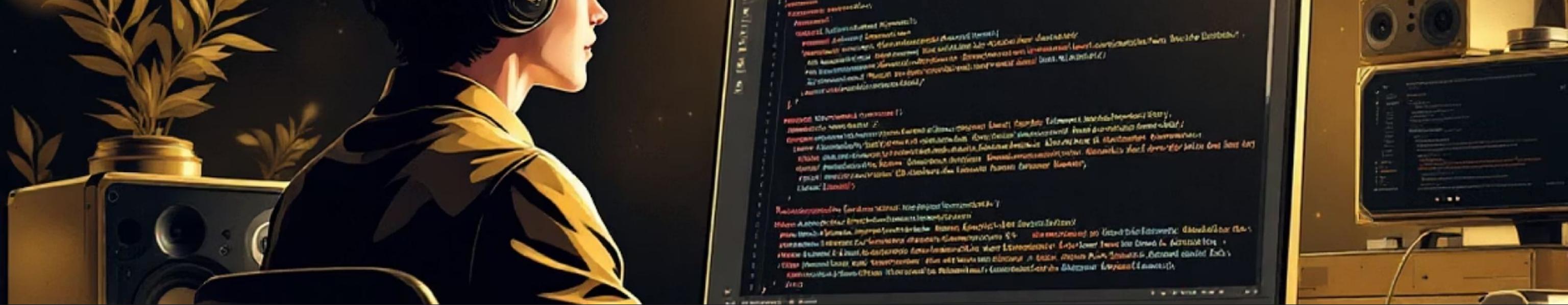
Integración con APIs Pùblicas

Éxito y eficiencia en la conexión y consumo de las APIs externas.



Manejo de Errores HTTP Semánticos

Implementación de respuestas de error adecuadas según estándares HTTP.



Recomendaciones para un Desarrollo Óptimo

Modularización

→ Es fundamental separar la lógica en servicios dedicados (ej: `geoService.js`, `transitService.js`) para mantener el código organizado y fácil de mantener.

Validación de Datos

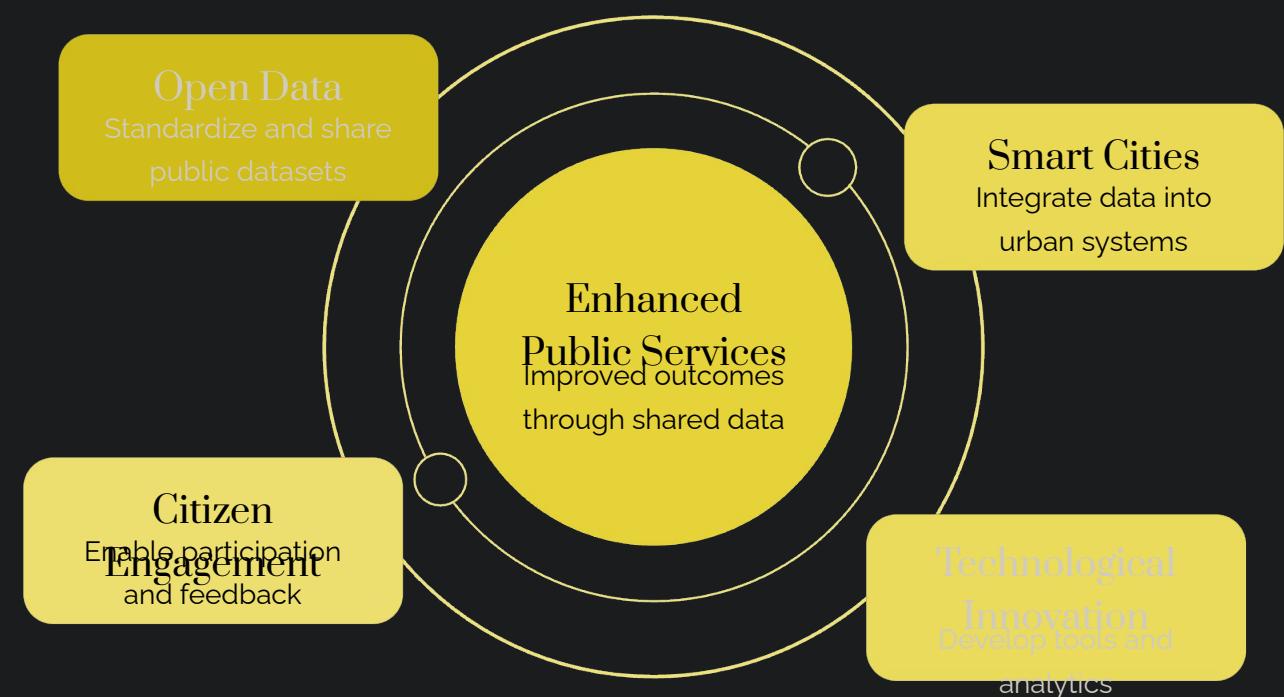
→ Implementar **middlewares robustos** para validar todos los datos entrantes, especialmente en los endpoints `POST`, asegurando la integridad de la información.

Entrega y Control de Versiones

→ El proyecto final debe estar en un **repositorio de GitHub** con al menos dos Pull Requests por integrante fusionados a la rama `develop`, demostrando colaboración efectiva.

El Impacto de CiudadData

CiudadData no es solo un proyecto académico; es una iniciativa con el potencial de transformar cómo accedemos y utilizamos la información pública. Al **unificar y estandarizar datos dispersos**, abrimos la puerta a nuevas aplicaciones y análisis que pueden beneficiar a ciudadanos y administraciones. Imagina aplicaciones que predigan congestiones de tráfico, alerten sobre problemas de salud pública o simplemente faciliten la vida diaria en la ciudad. Ese es el poder de los datos abiertos y bien estructurados.





¡Gracias! Preguntas y Conclusión

Esperamos que este proyecto no solo demuestre sus habilidades técnicas, sino también su visión para un futuro urbano más conectado e inteligente.

"El conocimiento es poder, y el acceso al conocimiento público es el empoderamiento de la ciudadanía."

Estamos abiertos a sus preguntas y a discutir el potencial de CiudadData.

¡Gracias por su atención!