

PRUEBA TÉCNICA DESARROLLADOR BACK-END

1. Consideraciones

- Asume que tu código se irá a producción y que pasará por code review.
- Sé claro y preciso en cómo debe desplegarse tu proyecto.
- Diviértete creando.

2. Requisitos mínimos

- Entrega de código en repositorio (compartir link en GitHub, Bitbucket o similar)
- Instrucciones de despliegue del proyecto y todas las consideraciones necesarias.

3. Instrucciones

1. Intenta cumplir la mayor cantidad de puntos posibles de la prueba, no son obligatorios ni se tienen que cumplir en ese orden, pero entre más resueles, podremos conocer mejor tus habilidades.
2. Evita copiar código de internet, nos daremos cuenta.
3. Procura que tu código sea claro y legible, será evaluado.
4. Si lo consideras necesario, documenta las funciones, métodos o bloques de código que sientas que son importantes.
5. Desarrolla la prueba con Python, Java, Go, PHP, eres libre de implementar el framework que más te guste.

4. Problema

Gustavo quiere iniciar un negocio de envíos por paquetería y le gustaría contar con la información de Estados, Municipios, Colonias ligados a SEPOMEX.

1. Obten una copia de la información expuesta en el sitio de [SEPOMEX](#)
2. Modela las tablas para la información de SEPOMEX.

- Separa la información en, por lo menos, 3 tablas: Estado, Municipio, Colonia
 - Relaciona tus tablas por medio de sus identificadores diseñados por SEPOMEX
3. Desarrolla una técnica de carga de los datos a partir de la información de SEPOMEX que pueda ejecutarse a demanda (seed script).
 4. Crea un API para obtener los datos del registro de cada recurso modelado.
 - El intercambio de datos se debe dar por medio de la estructura JSON.
 5. Permita a tu API por lo menos:
 - Búsqueda de colonias por CP
 - Búsqueda de colonias, municipios, estados por nombre
 - Adición de nuevos registros a los recursos.
 6. Agrega el manejo de errores.
 7. Agrega una capa de seguridad a tu API (Basic Auth, Token, APIkey, etc.)
 8. Aplica uno o varios patrones de diseño
 9. Muestra tu proyecto live en alguna plataforma en la nube.
 10. Empaqueta tu aplicación en un contenedor de Docker
 11. Optimiza la performance de la aplicación mediante el uso de técnicas de caching
 12. Implementa pruebas automatizadas para asegurar la calidad del código.
 13. Diseña un sitio web, full responsive para interactuar con tu API.