

Prueba técnica - Backend developer con Python

Información general para la prueba

- El aspirante debe tener conocimientos sólidos en Python. - Se recomienda utilizar frameworks que faciliten el desarrollo del producto a entregar (Flask, Django, etc).
- Durante la prueba es libre de utilizar cualquier herramienta que considere necesaria para la realización de los ejercicios y con la cual usted se sienta más cómodo.
- No existen malas formas de generar código en la aplicación, ya que cada desarrollador piensa distinto, más aún si se tendrán en cuenta buenas prácticas en desarrollo a la hora de la evaluación general de la prueba.
- El API que debe utilizar debe estar relacionado con Google, se adiciona documentación: [API](#)

Prueba de desarrollo BackEnd

Se requiere hacer un sistema de información para registrar los datos de los votantes de los distintos municipios de Colombia.

Allí se debe ingresar los datos básicos del votante y asignarle a éste, en que puesto de votación estará asignado.

Quien ingresa los datos debe ser un usuario del sistema definido como el Rol Líder.

El sistema debe tener un usuario administrador que podrá ver toda la información ingresada por los usuarios líderes.

Los usuarios con rol Líder, únicamente pueden ver la información que ellos mismos registran (el sistema no permitirá que un líder pueda acceder a la información de otro líder).

Debe quedar registrado cuando se registra un votante, y la información del usuario que lo registra (Log).

Para crear un usuario líder, se debe pedir los datos básicos de identificación mínimos requeridos (ciudad, dirección) y una foto, para su perfil.

Al momento de registrar un usuario (líderes y mesas de votación) en la plataforma es necesario obtener las coordenadas por medio del servicio de Georreferenciación de Google

Usar base de datos relacional y/o no relacional, usted puede elegir qué datos almacena en una o la otra.

Maneje validaciones en los campos de entrada del formulario de votantes. (Usted decide cuales).

Tenga en cuenta que cada mesa de votación, está asociada a un Municipio y este a un Departamento.

La siguiente es una estructura en la cual puede basarse, si usted quiere trabajarla o bien hacer la suya propia.

	Datos del votante			Lider		capitanes	
	Nombres			id_lider		id_capitan	
	Apellidos			nombres		nombres	
	Direccion			apellidos		apellidos	
	Telefono			Celular		Celular	
	cédula			capitan_id			
	lider_id						
	barrio_id			Lider_Resp_barrios			Capitan_Comuna
	puetovotacion_id			id_liderresponsable			id_capitancomuna
	mesa			lider_id			comuna_id
				capitancomuna_id			capitan_id
				barrio_id			
Puestos de votacion		Barrios				Comuna	
id_puesto		id_barrio				id_comuna	
nombre		nombre				Nombre	
dirección						municipio_id	
municipio_id		comuna_id					
						Municipio	
						id_municipio	
						nombre	

Requerimientos:

Este sistema debe ser una API Rest usando JSON como método de transporte de los datos.

Para obtener coordenadas, debe utilizarse el API de Georreferenciación de Google o similares.

Tener un servicio de Login (usuario y contraseña), que permita obtener un Token para consumir el resto de servicios.

El sistema deberá validar la existencia del Token en los servicios, para permitir acceder a la información.

Cada servicio creado, debe tener un CRUD básico para gestionar la información.

Exponer servicios que permitan realizar la gestión de Departamentos, Municipios, y mesas de votación.

Exponer un servicio que permita obtener la cantidad total de votantes inscritos por líder.

Exponer un servicio que permita obtener la cantidad total de votantes, en el sistema.

Exponer un servicio que permita obtener la cantidad total de votantes inscritos por municipio.

Exponer un servicio que permita obtener la cantidad total de votantes inscritos por mesa de votación.

Notas:

La fecha máxima de entrega es: 28 de Mayo de 2024, 5:00 .P.M

1. El tiempo comprometido y su cumplimiento hacen parte de la prueba.
2. Documentar en un archivo README las capas de la solución describiendo las responsabilidades de los métodos usados, en dónde se almacenan los datos, también describir cómo realizar la instalación y configuración de la base de datos, y qué Frameworks de Python fueron utilizados (si se utilizó alguno).
3. La solución se debe alojar en un repositorio privado, como Gitlab, Github ó Bitbucket, con acceso a la persona que revisa esta prueba.
4. El repositorio donde se aloja la solución, debe contener el código fuente, Modelo Entidad Relación de la solución (en caso de que use un modelo relacional), Script de la base de datos y cualquier otra información relevante.
5. Se debe entregar la documentación de cada servicio solicitado, puede entregarse como colección de los programas postman e insomnia.