

# Test de habilidades generales

## Área de desarrollo

### **Ingeniero Backend**





## Información general para la prueba

- 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
- Durante toda la prueba puede tener acceso a internet y consultar lo que considere pertinente.
- Es posible hacer preguntas a las personas encargadas de la prueba, En el caso de presentar la prueba de forma virtual es posible realizar las preguntas por whatsapp siempre y cuando las preguntas no sean orientadas al desarrollo de la misma, por ejemplo:
  - **Pregunta aceptable:** ¿Como puedo obtener una llave para usar el API de geocoding de Google Maps?
  - **Pregunta inaceptable:** ¿Como puedo crear un método que me permita aceptar solicitudes GET en Python?
- 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.

## Prueba de desarrollo backend

La prueba de conocimientos consiste en varios ejercicios que se calificarán de manera individual los cuales están orientados a medir las habilidades del aspirante y evaluar las prácticas de codificación y desarrollo que este posea. así como la facilidad para explicar y comunicar los diferentes procesos y código que el aspirante genere en caso de pasar a la fase siguiente del proceso.

### proyecto 1

Desarrollar un sistema sencillo de creación de usuarios que permita la creación, edición, activación, desactivación y eliminación de los mismos (Crud). para usar los métodos de dicho sistema el usuario final debe realizar un proceso de autenticación al sistema utilizando un token fijo para un servicio de autenticación el cual debe retornar los datos del usuario y un token de sesión que se deberá usar para los diferentes métodos que componen el crud según el estándar OAUTH 2.0

Una vez se construya este servicio se debe implementar lo siguiente:

Se debe crear una tabla en base de datos llamada usuarios la cual posea la siguiente información

- |             |              |
|-------------|--------------|
| ○ id        | ○ ciudad     |
| ○ nombre    | ○ longitud   |
| ○ apellido  | ○ latitud    |
| ○ dirección | ○ estado_geo |
| ○ tipo      | ○ cargo      |

Sobre la tabla usuarios elabore los siguientes servicios web en Python:

| Método HTTP | Endpoint | Descripción  |
|-------------|----------|--|
| POST        | /crear   | Método que permite crear usuarios almacenando nombre, apellido, dirección y ciudad, el ID debe ser único. En la misma tabla se van a almacenar usuarios del sistema de dos tipos, compradores y vendedores, para diferenciarlos se debe utilizar la columna tipo. Para el caso en el que los usuarios sean de tipo vendedor pueden existir dos cargos (asesor, cajero) los usuarios tipo vendedor no requieren ser georreferenciados por lo que la dirección no se debe solicitar para estos usuarios. para el caso de usuarios de tipo comprador la dirección debe ser obligatoria pero la información de cargo debe ser nula |

|        |                    |  |
|--------|--------------------|--|
| GET    | /lista             | Método que retorna el listado de todos los usuarios con sus diferentes atributos.  |
| GET    | /usuario           | Método que recibe el ID del usuario y retorna el usuario con sus atributos.  |
| DELETE | /eliminar          | Método que reciba en la url un identificador y elimine el usuario que tenga ese identificador.   |
| GET    | /geocodificar_base | Método que recorre todos los compradores almacenados, busca los que no tienen asignada longitud y latitud, utilice la dirección y ciudad para obtener ambos valores por medio del servicio de geocoder de Google y los actualiza, las direcciones que no son geocodificadas por Google se guardan con valor de 0 para ambos valores. |

**Observación:** Para conocer la api remitirse al siguiente documento

<https://developers.google.com/maps/documentation/geocoding/intro?hl=es-41>

**API KEY Para la prueba:**

**AlzaSyD0txU5n7xIPZ3zkEsEjW09yVHmmEVjU4o**

### Recomendaciones

- Ya que se van a crear usuarios de varios tipos se recomienda utilizar un patrón de diseño adecuado para la construcción del sistema
- aunque la prueba específica un solo endpoint para la creación considere si es necesario una estructura diferente de base de datos para almacenar los datos
- considere a nivel de la base de datos colocar las restricciones necesarias que garanticen las condiciones solicitadas.
- Recuerde que el uso de las apis puede ser costoso por lo que se debe limitar su uso y optimizar de la mejor manera el uso de las mismas (En este caso el api de geocoding)

## proyecto 2

En la página de datos abiertos colombia se debe conseguir el conjunto de datos correspondiente al Registro del núcleo básico de las víctimas de minas antipersonal (MAP) dicho conjunto de datos se encuentra de manera pública en el portal.

[https://storage.googleapis.com/media-help-ots-geomarketing-odoo/Situacion\\_Victimas\\_Minas\\_Antipersonal\\_en\\_Colombia\\_20240103.csv](https://storage.googleapis.com/media-help-ots-geomarketing-odoo/Situacion_Victimas_Minas_Antipersonal_en_Colombia_20240103.csv)

depto x victimas
SE HA CREADO UNA COMUNIDAD
Inclusión Social y Reconciliación
Mapa

Registro del núcleo básico de las víctimas de minas antipersonal (MAP), artefactos explosivos improvisados (AEI) con características de minas antipersonal y municiones sin explotar (MUSE) que
Más

Etiquetas No se seleccionaron etiquetas

Se actualizó  
15 de diciembre de 2022  
Vistas  
178

Una vez se descargue la información la misma debe ser limpiada ya que contiene caracteres especiales (tildes, acentos, ñ) que generan problemas al momento de realizar el tratamiento de los datos

|            |       |       |
|------------|-------|-------|
| TALAIGUA N | 13780 | Rural |
| CARURÁŠ    | 97161 | Rural |
| TEORAMA    | 54800 | Rural |
| TEORAMA    | 54800 | Rural |
| TEORAMA    | 54800 | Rural |

Para ello debe desarrollar un script en python que haga la limpieza y de ser necesario organice las columnas de forma correcta validando algunas entradas lógicas de datos (Por ejemplo Género (Hombre, Mujer)). por favor enviar bien sea el script en formato .py o algunos pantallazos del método seguido para realizar la limpieza de datos.

Se solicita crear un script que entregue información resumida en consola y escriba un archivo de texto con los resultados sobre las víctimas analizando el archivo de entrada en formato csv para encontrar la siguiente información

- total víctimas por año
- total víctimas por mes (Independiente del año)
- total víctimas por departamento
- total víctimas por actividad
- total de víctimas por género
- total de víctimas por estado

(Recuerde que la información debe entregarse de manera limpia)



### proyecto 3 Sistema de Gestión de Tareas

Se te pide que construyas un sistema simple de gestión de tareas. Debes implementar una API RESTful para realizar operaciones básicas en tareas. Cada tarea tiene un título, una descripción y un estado (pendiente, en progreso, completada).

Requisitos:

#### Modelo de Datos:

Debes definir un modelo de datos para representar las tareas. Utiliza el lenguaje/framework python/Django.

#### Operaciones Básicas:

Implementa las siguientes operaciones CRUD (Crear, Leer, Actualizar, Eliminar) para las tareas.

Los usuarios deben poder:

- Crear una nueva tarea.
- Obtener la lista de todas las tareas.
- Obtener detalles de una tarea específica.
- Actualizar el estado de una tarea (pendiente, en progreso, completada).
- Eliminar una tarea.

#### Validaciones:

Implementa validaciones para garantizar que los campos obligatorios (título, descripción) estén presentes y que el estado sea uno de los valores permitidos (pendiente, en progreso, completada).

#### Documentación:

Proporciona documentación sobre cómo usar la API. Puede ser un documento simple o comentarios en el código.

#### Pruebas Unitarias:

Escribe pruebas unitarias para al menos una parte importante del sistema.

**Entregar los tres proyectos de desarrollo en una cuenta pública a nombre propio asociada a git (GITLAB - GITHUB)**