

# Exámen Técnico

Instancia evaluativa

Evaluación técnica para postulantes a desarrollador

**DG Ciudadanía Digital**

Secretaría de Innovación y Transformación Digital



# Índice

Introducción.....	3
Aclaraciones previas.....	3
Backend + Postman.....	4
Frontend.....	4
GIT.....	5
Base de datos - SQL.....	5

# Introducción

---

Este documento consta de un examen para la evaluación técnica de los postulantes a desarrollador para el equipo de Tecnología de la DGCIUD.

## Aclaraciones previas

---

Antes de comenzar, tener en cuenta las siguientes aclaraciones sobre el examen técnico:

- No es obligatorio hacer completos todos los ejercicios
- Se puede usar cualquier herramienta que necesites, pero deberás trabajar por tu cuenta sin soporte externo salvo el nuestro
- Se pueden hacer todas las preguntas que se necesiten
- Para los objetivos de desarrollo se puede utilizar la tecnología que desees dentro de las siguientes opciones (y frameworks populares relacionados)
  - Python 3.9.16 | 3.11.4
  - Java, OpenJDK 11.0.9 | 17.0.7
  - NodeJS 16.19.1 | 18.14.1
  - PHP 8.0.13 | 8.1.20
  - .NET 6.0.119 | latest

El examen está organizado en 4 secciones: Backend + Postman, Frontend, GIT y SQL. Como aclaramos previamente, es ideal pero no necesario completar el 100% de la evaluación de todos los puntos ya que apunta a entender tu conocimiento y experiencia actual

# Backend + Postman

---

Desarrollo backend y pruebas con la herramienta Postman:

- a) Desarrollar un servicio RESTful utilizando el framework que desees y que permita realizar operaciones CRUD sobre una entidad "Producto".

Producto			
id_producto	int	PK	
nombre	varchar		
descripcion	varchar		
precio	decimal		
cantidad	int		

- b) Crear una colección en Postman que incluya diferentes solicitudes HTTP para interactuar con el servicio RESTful desarrollado en el ejercicio a).

## Frontend

---

Se necesita implementar una página que al recibir por parámetro una calle/dirección, devuelva un listado de calles sugeridas. Se debe mostrar sólo el nombre de la calle y dirección.

Para lograr esto se deberá consumir el siguiente servicio:

<http://servicios.usig.buenosaires.gob.ar/normalizar/>

### **Ejemplo de consulta al servicio:**

- **Request:** localhost/testphp?texto=corrientes 100
- **Response:** devuelve la lista
  - "direccion": "CORRIENTES AV. 100, CABA", "nombre\_calle": "CORRIENTES AV.", "direccion": "Corrientes 100, Almirante Brown", "nombre\_calle": "Corrientes", "direccion": "Corrientes 100, Escobar", "nombre\_calle": "Corrientes", "direccion": "Cabo Corrientes 100, Esteban Echeverría", "nombre\_calle": "Cabo Corrientes", [...]

Simulando que contamos con la **branch Master** y la **branch DEV**, donde la última está adelantada dado que tiene nuevas funcionalidades.

¿Qué pasos seguiría para resolver los siguientes requerimientos? Las soluciones pueden ser en palabras o con comandos GIT.

- a) Se quiere revertir la última funcionalidad pusheada en branch DEV
- b) Se quiere hacer un deploy en producción de la versión que llamaremos v1.2.0

## Base de datos - SQL

---

Ampliando el modelo de datos mencionado en el punto 1), se proponen las siguientes entidades nuevas:

Venta			
id_venta	int	PK	
fecha	datetime		
id_cliente	int	FK	
monto_total	decimal		
estado	varchar		

Cliente			
id_cliente	int	PK	
nombre	varchar		
email	varchar		
direccion	varchar		

DetalleVenta			
id_detalle	int	PK	
id_venta	int	FK	
id_producto	int	FK	
cantidad	int		
precio_unitario	decimal		

Se requiere resolver las siguientes consultas SQL:

- a) Productos que tengan nombre pero no descripción
- b) Productos vendidos en las últimas 24 horas
- c) Recuento de todos los estados de venta
- d) Ventas en las que el cliente con email X gastó una suma mayor a Y