

# Prueba Técnica: Desarrollador(a) Full Stack

## Descripción del Proyecto

El objetivo de este proyecto es desarrollar un aplicativo web utilizando Node.js, TypeScript, Angular que consuma una API externa de datos de COVID-19 , almacene la información en una base de datos MongoDB y presente la información de resultado en páginas web.

## Requisitos Funcionales

1. Consumir la API de datos de COVID-19 de la siguiente URL:  
<https://www.datos.gov.co/resource/prrv-jnta.json>.
2. Almacenar los datos de departamentos y cantidad de dosis aplicadas en la base de datos MongoDB.
3. Implementar un endpoint tipo GET que permita almacenar los datos obtenidos de COVID-19 en mongodb.
4. Implementar un endpoint tipo GET que permita obtener el listado de todos los departamentos y sus dosis aplicadas.
5. El endpoint debe aceptar un parámetro opcional "departamento" para filtrar los datos por departamento.
6. La estructura de respuesta del endpoint debe ser consistente en todos los casos.
7. Generar pantallas web (Angular y framework de CSS ) de consulta para los datos almacenados en la base de datos por medio de api rest. (Preferiblemente con framework de CSS - tailwind)

## Arquitectura por Capas - Microservicios

El proyecto seguirá una arquitectura por capas para garantizar la separación de responsabilidades y la escalabilidad del código:

1. Controlador: Se encarga de manejar las solicitudes HTTP y las respuestas.
2. Dominio: Contiene la lógica de negocio de la aplicación.
3. Servicio: Proporciona funcionalidades específicas para conexión con servicios externos.

4. Repositorio: Gestiona la interacción con la base de datos.

## Endpoints

1. GET /vacunas

Descripción: Obtiene el listado de todos los departamentos y sus dosis aplicadas.

Parámetros de consulta:

`departamento` (opcional): Nombre del departamento para filtrar los datos.

Respuesta:

```
json Copy code

[
  {
    "departamento": "Antioquia",
    "dosis_aplicadas": 12345
  },
  {
    "departamento": "Bogotá",
    "dosis_aplicadas": 23456
  },
  ...
]
```

## Pruebas Unitarias

Se utilizará Jest para escribir pruebas unitarias que validan el funcionamiento correcto de los endpoints y/o aplicación web.

## Tecnologías Utilizadas

- Express.js: Framework web para Node.js.
- TypeScript: Superset de JavaScript que agrega tipado estático.
- Mongoose: Librería de modelado de objetos para MongoDB.
- Axios: Cliente HTTP para realizar peticiones a la API externa.
- Jest: Framework de pruebas para JavaScript.
- MongoDB: Base de datos NoSQL.
- Angular: Framework basado en componentes para crear aplicaciones web

## Instrucciones de Ejecución Backend

1. Instalar las dependencias utilizando el comando `npm install`.
2. Configurar las variables de entorno necesarias, como la URL de conexión a MongoDB.
3. Ejecutar el proyecto utilizando el comando `npm start`.
4. README con documentación de ejecución del proyecto

## Instrucciones de Ejecución Frontend

5. Instalar las dependencias utilizando el comando `npm install`.
6. Configurar las variables de entorno necesarias, como la URL de conexión a microservicios.
7. Ejecutar el proyecto utilizando el comando `npm run dev`.
8. README con documentación de ejecución del proyecto

## Entregables

1. Comprimido con código fuente del proyecto.