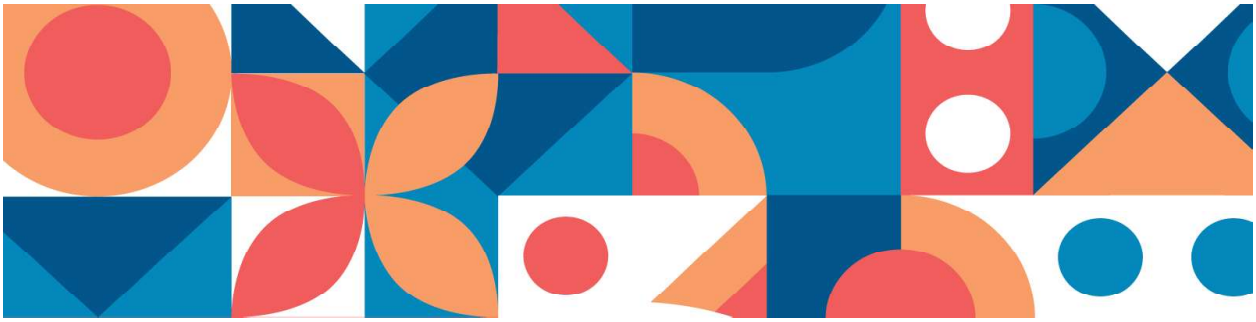


Descripción de la Prueba Técnica

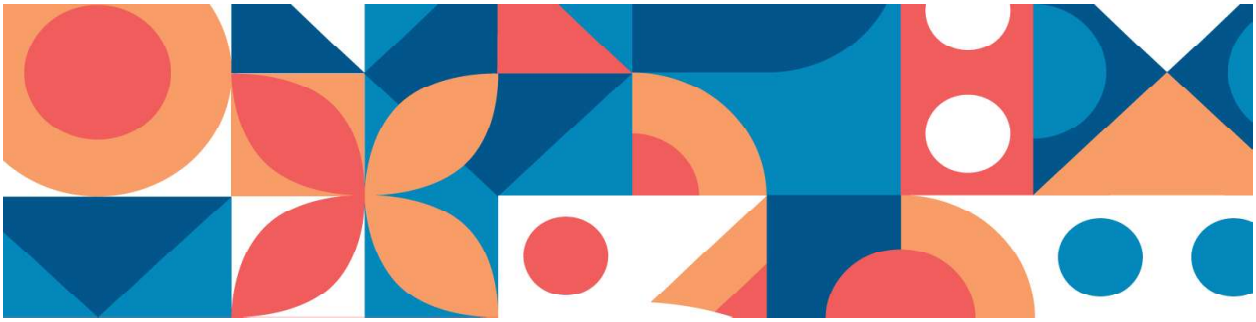
Para esta prueba, te recomendamos que puedas generar un repositorio en Git, subirlo al público y luego brindarnos el link para que podamos descargarlo.

Objetivo:

El objetivo de esta prueba técnica es evaluar las habilidades del candidato en el desarrollo backend en NestJS (o Node.js en su defecto). Se espera que el candidato demuestre conocimientos en el diseño de API REST, manipulación de datos y manejo de errores.

Instrucciones:

1. Configuración del Proyecto:
    1. Crea un nuevo proyecto en NestJs
    2. Configura un archivo de configuración de TypeScript (`tsconfig.json`).
    3. Utiliza npm o yarn para gestionar las dependencias del proyecto.
    4. Utiliza una base de datos Mongo

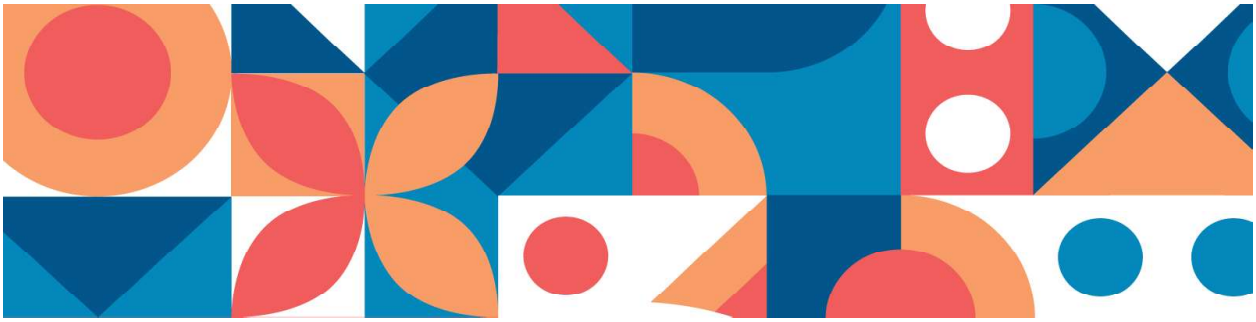


2. API REST de Usuarios:

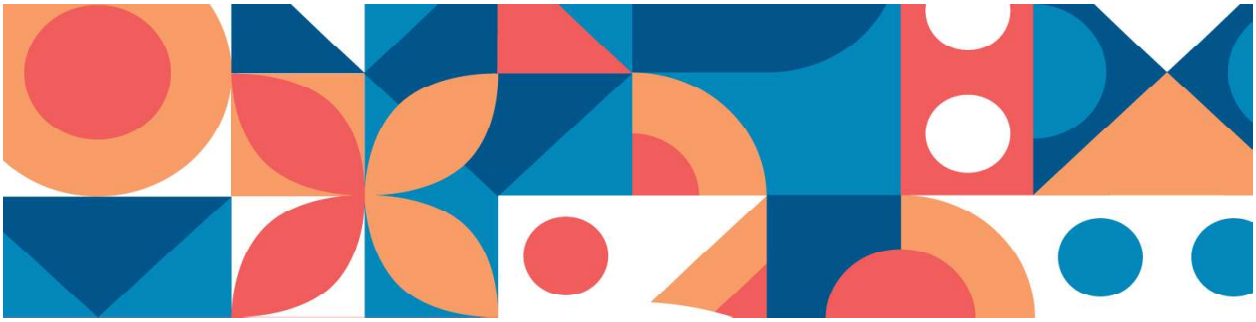
    1. Diseña una API REST para gestionar usuarios con las siguientes operaciones:

        1. Crear un nuevo usuario y sus datos de perfil.
        2. Obtener la lista de usuarios con su perfil
        3. Obtener los detalles de un usuario por su ID.
        4. Actualizar la información de un usuario por su ID.
        5. Eliminar un usuario por su ID.
        6. Filtros por texto.
        7. Agregado de paginado y ordenamiento

    2. Módulo de autenticación



3. Modelo de Usuario y Perfil:
    1. Crea un modelo de usuario y uno de perfil
4. Validaciones:
    1. Implementa validaciones para asegurarte de que los campos requeridos estén presentes al crear y actualizar un usuario.
5. Manejo de Errores:
    1. Implementa un manejo adecuado de errores para todas las operaciones.
    2. Proporciona respuestas JSON significativas en caso de errores.
6. Documentación:
    1. Documenta la API utilizando comentarios en el código o una herramienta como Swagger.
    2. Proporciona ejemplos de uso para cada endpoint.
7. Pruebas Unitarias:
    1. Escribe pruebas unitarias utilizando Jest o cualquier otra herramienta de prueba de tu elección.



Entrega:

1. Código Fuente:
    1. Repositorio Público: Se requiere un enlace al repositorio público que contenga todo el código necesario para el proyecto. El repositorio debe estar correctamente organizado, incluyendo una estructura clara de carpetas y archivos que reflejan la arquitectura del software.
    2. Dockerfile: Se espera la entrega de un Dockerfile que contenga todas las configuraciones necesarias para construir y ejecutar el servicio en un contenedor Docker de manera eficiente.
       El Dockerfile debe incluir todos los pasos requeridos para la instalación de dependencias, la configuración del entorno y la puesta en marcha del servicio.
2. Documentación:
    1. Incluye cualquier documentación necesaria para comprender y ejecutar el proyecto.
3. Pruebas Unitarias:
    1. Si has realizado pruebas unitarias, incluye los scripts y resultados de las pruebas.

Evaluación:

Se evaluará la calidad del código, la organización del proyecto, la implementación correcta de la lógica de negocio, el manejo de errores, la documentación y la cobertura de las pruebas unitarias.
Recuerda proporcionar comentarios en el código para explicar decisiones importantes que hayas tomado durante el desarrollo. ¡Buena suerte!