

Reto Técnico – Full Stack

Base de Datos

Crear una base de datos con las siguientes tablas:

1. Usuario
2. Producto
3. Categoría

Requisitos:

- Usar base de datos relacional
- Respetar estándares de diseño y modelamiento de base de datos
- Usar buenas prácticas a nivel de tablas, campos y tipo de datos
- Los campos de las tablas son de libre elección.

Back End

Crear las APIs para las tablas:

1. EndPoint de Inicio de Sesión (JWT)
2. EndPoints para Crear, Obtener, Editar y Eliminar un Producto.
3. EndPoints para Crear, Obtener, Editar y Eliminar un Categoría
4. EndPoint de listar Productos.

Requisitos:

- Usar Node Js, Prisma (ORM), Nest Js (Framework).
- Usar buenas prácticas a nivel del proyecto, código, nomenclatura, etc.
- Agregar validaciones a nivel de campos como limites caracteres y tipo de datos.
- Los endpoints de productos y categoría solo se podrán consumir con un token valido.
- El endpoint de crear producto no debe de permitir registro duplicados por nombres.
- El endpoint de listar productos debe contemplar paginación.
- Implementar el manejo control de errores.

Front End

Crear las siguientes interfaces:

1. Formulario que permita Crear, Obtener, Editar y Eliminar una Categoría
2. Formulario que permita Crear, Obtener, Editar y Eliminar un Producto.
3. Grid o Tabla de datos con información de Productos (lista de productos).

Requisitos:

- Usar React Js, Next Js (Framework).
- Usar buenas prácticas a nivel del proyecto, código, nomenclatura, etc.
- Consumir las APIs de Producto en el Formulario.
- Consumir las APIs de Categoría en el Formulario.
- El grid o la tabla de productos debe contemplar paginación.
- Agregar validaciones de datos en los formularios.

Opcional: Siéntase con la libertad de agregar mejoras, ya que serán tomado en cuenta.

¡Puntos Extras! Usar los servicios de AWS para la publicación de APIs y Front End.