Forma

Descripción generada automáticamente con confianza baja

PRUEBA TECNICA FULL STACK DEVELOPER

Tabla de contenido

[INTRODUCCION 3](#_Toc116424315)

[CONSIDERACIONES 3](#_Toc116424316)

[ACTIVIDAD 4](#_Toc116424317)

[Base de datos 4](#_Toc116424318)

[Backend 4](#_Toc116424319)

[Frontend 5](#_Toc116424320)

[Respaldo y despliegue 5](#_Toc116424321)

[Entregables 5](#_Toc116424322)

# INTRODUCCION

Apreciado candidato, agradecemos su interés y participación en el proceso de selección del cargo en mención. Para IDATA es un honor contar con talentos como el suyo participando de nuestros procesos.

Lea con atención cada uno de los siguientes puntos.

La presente prueba de conocimientos técnicos deberá ser desarrollada utilizando los siguientes componentes.

* El backend deberá ser desarrollado con el lenguaje de programación C# (.NET Core) o Python (Django o Flask).
* El frontend deberá ser desarrollado con el lenguaje de programación Javascript o Typescript.
* Deberá elegir entre Angular y React como framework para el desarrollo frontend.
* Puede utilizar bibliotecas de estilos a su elección (Ej. Bootstrap, SASS, Bulma, Tailwind, Material UI, etc).
* Tenga en cuenta que para desarrollar la prueba de conocimientos deberá tener experiencia en HTML, CSS, SQL y servicios REST. (Obligatorio)

# CONSIDERACIONES

Estos son otros aspectos que debe tener en cuenta durante el desarrollo de la prueba:

* Para el desarrollo del **frontend** no puede hacer uso de plantillas descargadas de internet.
* Si elige el lenguaje de programación Python para desarrollar el backend, se calificará la prueba con puntos extras.
* Si automatiza el despliegue del backend por medio de flujos de CI/CD en Azure DevOps, se calificará la prueba con puntos extra.
* Si elige desarrollar el frontend con el lenguaje de programación Typescript, se calificará con puntos extra.
* Si aplica patrones como Redux o Bloc en el desarrollo web, se calificará con puntos extra.
* Si protege el backend por medio de openid, oauth2 o JWT, se calificará con puntos extra.
* Los endpoints de backend deben seguir el estándar de toda API Rest.
* Se valorará el uso de buenas prácticas de programación como patrones de diseño y estructura del código fuente.

# ACTIVIDAD

La actividad consiste en desarrollar un sitio web para gestión de productos, deberá tener en cuenta los siguientes aspectos:

## Base de datos

* Debe definir y generar el MER de la base de datos con la siguiente información:

Se tiene una empresa que vende productos a sus clientes, se necesita conocer los datos personales de los clientes (documento, tipo de documento, nombre, apellido, dirección, teléfono, fecha de nacimiento y url de imagen de perfil) cada producto tiene nombre, categoría, marca, valor unitario, unidades en stock y descripción. Un cliente puede comprar varios productos y un mismo producto puede ser comprado por varios clientes, los productos son suministrados por proveedores. Se debe tener en cuenta que un producto solo puede ser suministrado por un proveedor y que un proveedor puede suministrar diferentes productos, de cada proveedor se desea almacenar el nit, nombre, ciudad, correo y numero de contacto.

* Se debe contar con tablas adicionales para el manejo de usuarios y roles.

## Backend

* Debe generar un endpoint que permita la autenticación de los usuarios almacenados en la base de datos.
* Debe generar un endpoint que permita obtener la lista de los usuarios con su respectivo rol.
* Debe generar un endpoint que permita obtener un único usuario.
* Debe generar los endpoints necesarios que permitan las operaciones CRUD de la entidad de clientes.
* Debe generar los endpoints necesarios que permitan las operaciones CRUD de la entidad de productos.
* Debe generar un endpoint que permita obtener el histórico transaccional de compras con la siguiente información:
  + Nombre y documento del cliente.
  + Nombre, cantidad, proveedor y valor del producto.
  + Fecha de compra

**Nota:** Para la prueba no se requiere generar un endpoint que permita generar compras, pero si se debe ingresar la información de al menos 3 transacciones de manera manual en la base de datos, esto con el fin de poder probar el correcto funcionamiento del anterior requerimiento.

## Frontend

* Se debe desarrollar toda la interfaz de autenticación y su funcionalidad, no es necesario generar un modulo de registro, la información de usuarios y roles se puede ingresar de manera manual en la base de datos.
* Una vez el usuario inicie sesión, se deberá poder ver su nombre y rol dentro de la aplicación.
* Debe generar un módulo que permita listar todos los clientes y que además permita hacer todas las operaciones CRUD sobre los clientes.
* Debe generar un módulo que permita listar todos los productos y que además permita hacer todas las operaciones CRUD sobre los productos.
* Se deben generar alertas y notificaciones que permitan al usuario conocer el estado de cada acción que ejecuta dentro del sitio web.

## Respaldo y despliegue

* Todo el código debe estar versionado utilizando git, además de que se debe trabajar con las siguientes 2 ramas:
  + develop: rama donde se debe trabajar todo el desarrollo.
  + master: rama donde debe estar vacía.
* Todo el código debe ser subido a repos de Azure DevOps.
* Debe aprovisionar 2 instancias de Azure Web App y deberá desplegar el código tanto de backend como de frontend.

## Entregables

* Url del repo donde está todo el código del frontend.
* Url del repo donde está todo el código del backend.
* MER de la base de datos.
* Script SQL que permite generar la base de datos y los datos necesarios para utilizar el aplicativo.
* Matriz de roles y usuarios con sus respectivas credenciales.