

Objetivo

Determinar si el candidato está en condiciones de interactuar con una base de datos a través de una aplicación web, implementando cliente y servidor.

Este ejercicio, no tiene ningún fin comercial ni estratégico, sólo queremos medir tus habilidades.

Ejercicio

Construir una tienda online que despliegue productos agrupados por la categoría a la que pertenecen, generando por separado backend (API REST) y frontend (Aplicación que la consume) y utilizando la base de datos que se disponibiliza para su desarrollo.

Además, hay que agregar un buscador, **el cual tiene que estar implementado a nivel de servidor**, mediante una Api Rest **cuyo lenguaje y framework puede ser de libre elección**. Es decir, los datos de productos deben llegar filtrados al cliente.

Opcionalmente, puedes implementar filtros por atributo, ordenar productos y paginación.

La aplicación de cliente tiene que estar desarrollada con **vanilla javascript (javascript puro), sin ningún framework**, si puedes usar librerías o componentes específicos, tales como; bootstrap, material, jquery, entre otros.

Finalmente, disponibilizar la aplicación y el repositorio con el código en un hosting como a modo de ejemplo, puede ser; Heroku, Netlify, Vercel u otros. También puedes levantar el sitio en una cuenta AWS.

Acceso a la base de datos de uso obligatorio

Base de datos

- Motor: MySQL
- Host: itnovai-test.czny55quvxhk.us-east-1.rds.amazonaws.com
- Usuario: itnovai_test
- Contraseña: itnovai_test
- Nombre base de datos: itnovai_test

Tablas

product	
id	<u>Identificador único del producto (int)</u>
name	<u>Nombre del producto (varchar)</u>
url_image	<u>URL de la imagen asociada al producto (varchar)</u>
price	<u>Precio de venta del producto (float)</u>
discount	<u>Porcentaje de descuento del producto (int)</u>
category	<u>Identificador de la categoría (int)</u>

category	
id	<u>Identificador único de la categoría (int)</u>
name	<u>Nombre de la categoría (varchar)</u>

CONOCE LA PAUTA DE EVALUACIÓN

<u>Categoría</u>	<u>Acción que evaluaremos</u>	<u>Ponderación</u>
Requisito obligatorio	Se debe utilizar control de versiones	Obligatorio
Requisito obligatorio	La aplicación debe tener deploy	Obligatorio
Programación	El código debe ser limpio, ordenado y seguir buenas prácticas	40%
Programación	La aplicación debe ser eficiente y controlar errores	20%
Documentación	Documentar la aplicación*	40%
Funcionalidad	Que se entregue buena usabilidad y experiencia al usuario.	40%
TOTAL		100%

- ¿Qué es documentar? implica explicar ejercicio, Api Rest, Fronted y Readme (puede ser en documento aparte o en el repositorio).

¡MUCHO ÉXITO!