

Prueba de desarrollo GEEK Cosmetics

Objetivo: desarrollar un sistema que permita la simulación de la venta de artículos, la generación de un pequeño informe y el almacenamiento en base de datos, empleando Javascript, AJAX, HTML5, CSS3 (Se recomienda implementar **ReactJS** para el frontend y **ExpresJS** para el backend), objetos JSON y una **base de datos SQL** (MySQL O PostgreSQL).

El sistema debe ofrecer una página de inicio que indica el nombre del desarrollador del proyecto y le ofrece al usuario dos opciones:

1. **Realizar compra:** dirige al usuario a una página que permite el ingreso de una nueva compra y se compone de las siguientes secciones:
 - I. Orden: es un formulario con los siguientes campos:
 - a. **Número de orden:** campo no editable y auto numérico.
 - b. **Nombre:** campo que le permite al usuario indicar quien realiza el pedido.
 - c. **Fecha:** campo no editable que indica la fecha actual con la que se realiza el pedido en formato año/mes/día hora/minuto.
 - d. **Artículo:** lista desplegable con las descripciones de los productos.
 - e. **Cantidad:** campo numérico que indica las unidades del producto que se desea ordenar.
 - f. **Subtotal:** muestra el valor total del ítem a insertar al detalle de la orden, es decir, cantidad por precio unitario del producto.
 - g. **Agregar:** este botón ingresa los datos de este formulario al detalle de la orden y refresca el formulario.
 - II. Detalle de la orden: es una cuadrícula que muestra los ítems que le usuario desea ordenar y a su vez permite la eliminación de estos, sigue el siguiente ejemplo:

Artículo	Cantidad	Subtotal	Borrar ítem
Articulo1	2	\$270.000	✕
Articulo2	4	\$560.000	✕
Articulo3	8	\$64.000	✕
Subtotal:		\$894.000	
Total IVA:		\$169.860	
Total:		\$1'063.860	

- III. Botón finalizar compra: este botón almacena el número de orden, el subtotal, total del IVA y el total de la orden y dirige al usuario al inicio.

Para los procesos de calculo y almacenamiento temporal se deben emplear arrays. Una vez se seleccione finalizar compra se hace el almacenamiento en **base de datos**.

2. **Total compras:** es una página web que muestra una cuadrícula con el numero de orden, el subtotal, total del IVA y el total de todas las compras realizadas que se

encuentran almacenadas en la base de datos, la parte visual se realiza con Javascript y para la extracción de datos se debe generar un JSON mediante NodeJS que envíe al frontend los datos necesarios para este informe.

Restricciones:

- En el detalle de la orden no puede haber artículos repetidos.
- No se permite la venta de más unidades que las especificadas en las existencias de cada producto.
- Se deben realizar validaciones de integridad de datos, por ejemplo: los campos numéricos no admiten el ingreso de valores negativos, los nombres no deben tener números.
- A la hora de leer el JSON, la información de los productos información debe ser almacenada en base de datos a la hora de cargar el proyecto.

Recursos: los datos necesarios para trabajar deben ser solicitados por método GET consumiendo desde el archivo JSON que se adjunta: **info_prueba.json**, esta solicitud devuelve un objeto JSON con la descripción de cada producto, su precio unitario y su existencia actual.

Observaciones:

- El diseño de las páginas queda a libertad del desarrollador, es un punto a tener en cuenta en la calificación.
- Para la entrega se debe subir a un repositorio GIT el proyecto junto con el archivo SQL de la base de datos.
- Tener en cuenta la siguiente sesión del gitbook – CheckPointsGeeks para presentar pruebas técnicas:
 - <https://john-florez.gitbook.io/checkpointgeeks/tip-geek-pruebas-tecnicas>
- Puntos para valorar en la prueba:
 - Planeación de actividades o tareas a realizar.
 - Implementar estrategias de ramificación haciendo uso de Git y GitHub.
 - Arquitectura Limpia o por capas (Capa de presentación, lógica, datos).
 - Despliegue de la solución (Deseable GCP).
 - Tener en cuenta manejo de excepciones y errores.
 - Documentación de la API con SWAGGER.
 - Pruebas Unitarias.
 - Implementación de microservicios.