



Evidencia: Aplicación de React con Axios, filtros y estadísticas

Objetivo

Evaluar la capacidad del estudiante para:

- Obtener datos de una API utilizando Axios.
- Aplicar filtros dinámicos usando `useState` y `useEffect`.
- Mostrar estadísticas basadas en los datos obtenidos.
- Crear y utilizar componentes reutilizables en React.
- Implementar renderizado condicional y manejo de listas.

Actividades

A partir del código base proporcionado, realizar las siguientes modificaciones y agregados:

1. Agregar estadísticas adicionales

Incorporar al menos **tres estadísticas nuevas** a la sección de estadísticas. Algunas sugerencias posibles:

- El producto más caro (mostrar el nombre y el precio).
- El producto más barato (mostrar el nombre y el precio).
- La cantidad de productos cuyo título contiene más de 20 caracteres.
- El precio total de todos los productos filtrados.
- El promedio de descuento (`discountPercentage`) de los productos filtrados.

Puedes proponer otras estadísticas, siempre que se basen en los productos filtrados y se calculen dinámicamente.

2. Crear dos componentes adicionales

Dividir el código en al menos **dos nuevos componentes funcionales**:

- **ProductList**: se encargará de mostrar la lista de productos.
- **StatsPanel**: se encargará de mostrar todas las estadísticas.

Ambos componentes deben recibir las props necesarias para funcionar correctamente. No deben contener lógica de estado interno.

3. Validar búsqueda vacía

Cuando el campo de búsqueda esté vacío, mostrar todos los productos sin errores. Asegurarse de que el código no falla ni en las estadísticas ni en el renderizado de la lista.

4. Diseño básico

Aplicar estilos básicos con CSS o Tailwind para mejorar la visualización. Ejemplos:

- Separar la sección de estadísticas con un fondo distinto.
- Mostrar los productos en tarjetas.
- Agregar una animación leve cuando se actualiza la lista o las estadísticas.

Entregables

- Código fuente completo del proyecto en una carpeta o repositorio.
- Breve explicación escrita (puede ser en un archivo README.md) de:
 - Qué estadísticas se agregaron.
 - Cómo se dividió el código en componentes.

Criterios de evaluación

Criterio	Puntaje
Axios se utiliza correctamente	2 pts

Filtro dinámico funciona correctamente	2 pts
Se agregaron al menos 3 estadísticas nuevas	3 pts
Uso correcto de props y componentes	3 pts
Código organizado y legible	2 pts
(Bonificación) Estilos o animaciones útiles	+1 pt

Total: 12 puntos posibles