

<https://github.com/rsmediabce/mercadokuka/>

Campos incluidos y su propósito:

JSON-SERVER en Mockapi.io

Ejemplo del arreglo recibido

```
{
  "id": "1",
  "name": "iPhone 13 Pro",
  "description": "El iPhone 13 Pro cuenta con un sistema de cámaras Pro más avanzado, pantalla Super Retina XDR con ProMotion, chip A15 Bionic y gran autonomía.",
  "price": 999.99,
  "category": "Smartphones",
  "brand": "Apple",
  "images": [
    "img/products/iphone13-1.webp",
    "img/products/iphone13-2.webp",
    "img/products/iphone13-3.webp"
  ],
  "thumbnail": "img/products/iphone13-thumb.png",
  "stock": 45,
  "specifications": {
    "color": "Grafito",
    "storage": "128GB",
    "screen": "6.1 pulgadas",
    "camera": "Sistema triple cámara 12MP",
    "battery": "Hasta 22 horas de reproducción de video",
    "os": "iOS 15"
  },
  "condition": "Nuevo"
},
```

Para listado de productos:

- `id` - Identificador único
- `name` - Nombre del producto
- `price` - Precio actual
- `category` - Categoría para filtrar
- `brand` - Marca del producto
- `thumbnail` - Imagen pequeña para la lista
- `images` - Array de imágenes cargadas el producto
- `stock` - Disponibilidad
- `condition` - Estado del producto, nuevo, usado, reacondicionado

Para detalle del producto:

- `description` - Descripción completa
- `images` - Array de imágenes en alta calidad
- `specifications` - Array con distintas especificaciones del producto

Se implementaron funcionalidades como:

- Filtro por precio máximo, mínimo y rango
- Listado con filtros por categoría
- Listado con filtros por marca
- Listado con filtros por condición
- Ordenamiento por precio
- El listado de productos tiene agregado un paginador completamente funcional que se actualiza si se actualizan los criterios de búsqueda o filtrado de los productos.
- Detalle completo del producto
- Si no hay productos en el carrito, el botón ver carrito siempre se encontrará deshabilitado.
- En el carrito se pueden agregar o quitar unidades, eliminar productos o directamente vaciar el carrito y el precio total se va actualizando dinámicamente ante cada operación
- Al realizar el pago se puede seleccionar un domicilio
- Esta pensado el pago con distintos medios de pago y se simula un proceso de pago con un `timeout()` pero no esta validado si se carga o no una tarjeta de crédito correctamente

Ordenamiento y filtros

Para el ordenamiento por precio y rangos se trabaja con los datos en el array local (no se vuelven a pedir con un `fetch`)

Para el ordenamiento y filtro por categoría, marca y condición los datos se solicitan con un `fetch` al servidor remoto.

Para no complicarla demasiado, los filtros son únicos, si filtro por marca, no puedo filtrar por categoría o condición, y lo mismo por precio.