

1. Objetivo general del frontend

Crear una **tienda online de ropa y accesorios**:

- Visualmente similar a los ejemplos que has mandado.
- Construida con **React + Vite** y **fetch** hacia nuestro backend.
- Con experiencia clara para el alumno:
 - Ver **productos destacados** en un carrusel.
 - Navegar por **categorías** (menú lateral + tarjetas grandes).
 - Gestionar sesión: **login, registro, mis pedidos**.
 - Ver detalle de producto y detalle de pedido.

2. Paleta de colores (rosa / lila / verde)

Inspirada en e-commerce “fashion”, pero manteniendo accesibilidad:

-  **Color principal (acciones / botones)**
 - Rosa fuerte: #F0557A
-  **Color secundario (detalles, hover)**
 - Lila suave: #C77DFF
-  **Color de acento (etiquetas, badges de oferta)**
 - Verde menta: #4CC9A6
-  **Fondo principal**
 - Gris muy claro: #F9F9FB
-  **Fondo tarjetas**

- Blanco: #FFFFFF
-  **Texto principal**
 - Gris oscuro: #222222
-  **Texto secundario**
 - Gris medio: #666666
-  **Estados de error / alerta**
 - Rojo suave: #E63946
 - Amarillo aviso: #FEEA9A

Notas:

- Contraste mínimo AA para texto (botones rosa sobre blanco, texto oscuro sobre fondos claros).
 - En hover, aclarar el fondo y mantener el texto oscuro.
-

3. Tipografía y estilo general

- **Tipografía principal:**
 - Poppins o Nunito (Google Fonts), ligera y moderna.
- **Jerarquía:**
 - Títulos (H1/H2): 24–32px, semibold.
 - Subtítulos (H3/H4): 18–20px.
 - Texto normal: 14–16px.
 - Pequeño/etiquetas: 12–13px.

Estilo general:

- Tarjetas con bordes redondeados (`border-radius: 16px`) y sombras suaves.
 - Mucho aire: padding generoso, secciones separadas visualmente.
 - Iconos sencillos (carrito, usuario, categorías).

4. Layout global (Layout.jsx)

Estructura base de la app:

- **Responsive:**
 - En escritorio: sidebar de categorías visible a la izquierda.
 - En móvil/tablet: categorías en un menú horizontal o desplegable.

5. Secciones principales

5.1. Barra de navegación (Navbar.jsx)

Contendrá:

- Logo / nombre tienda (texto o simple logotipo).
- Menú:
 - **Inicio**
 - **Catálogo**
 - **Mis pedidos** (solo si logado)
- A la derecha:
 - Icono usuario con:
 - “Iniciar sesión” / “Registrarse” si NO logado
 - Nombre del usuario + “Salir” si logado
 - (Más adelante) icono carrito.

Comportamiento:

- Fijo arriba (`position: sticky; top: 0`) para que siempre se vea.
 - Cambia ligeramente el fondo al hacer scroll (sombra sutil).
-

5.2. HomePage.jsx

Bloques:

1. **Hero principal con carrusel:**
 - Carrusel de **productos destacados** (endpoint `/api/productos/destacados`).
 - Imagen grande en un lado, texto en el otro:

- Título (“Nuevas colecciones”, “Rebajas”, etc.)
- CTA: botón “Ver colección” / “Ver catálogo”.
- Slides automáticos con navegación manual (puntitos o flechas).

2. Bloque de categorías destacadas:

- Tarjetas grandes por categoría: “Mujer”, “Hombre”, “Niños”, “Accesorios...”
- Cada tarjeta:
 - Imagen representativa (de la tabla `categorias` o fija en frontend).
 - Nombre de categoría.
 - CTA: “Ver productos”.
- Al hacer click → navega a `/catalogo?categoria_id=X`.

3. Carrusel / grid de promociones:

- Productos con `destacado=1` repartidos en tarjetas horizontales.
 - Etiquetas de descuento / “Novedad”.
-

5.3. CatalogoPage.jsx

Estructura:

- En escritorio:
 - **Sidebar categorías** (`FiltroCategorias.jsx`) a la izquierda.
 - **Grid de productos** a la derecha (`ProductoCard.jsx`).

Sidebar:

- Lista de categorías con:
 - Nombre.
 - MiniiImagen/icono.

- Contador de productos (opcional).
- Opción “Todas las categorías”.
- En móvil: se convierte en dropdown o menú superior tipo tabs.

Contenido principal:

- Barra superior con:
 - Texto “Mostrando X productos”.
 - Select de orden: “Precio ↑”, “Precio ↓”, “Nombre A-Z”.
- Grid responsive:
 - 4 columnas en escritorio, 2 en tablet, 1 en móvil.
- Cada tarjeta ProductoCard:
 - Imagen principal (`imagen_1` o `imagen_url`).
 - Nombre.
 - Precio.
 - Badge si `destacado=1`.
 - Link “Ver detalle”.

5.4. Detalle de producto (PedidoDetallePage.jsx o futura ProductDetailPage.jsx)

Aunque el nombre de página en tu listado es “PedidoDetallePage.jsx”, para frontend de tienda conviene también una **página de detalle de producto** (la podemos llamar luego `ProductoDetallePage.jsx`).

Contenido:

- Imagen grande del producto + miniaturas (`imagen_1/_2/_3`).
- Nombre, precio, stock.

- Descripción.
 - Botón “Añadir a mi pedido / carrito” (si está logado).
 - Productos relacionados (misma categoría).
-

5.5. LoginPage.jsx y RegisterPage.jsx

- Formulario centrado en pantalla, en una tarjeta blanca con sombras suaves.
 - Botones con color rosa principal.
 - Mensajes de error claramente visibles.
 - Enlace de cambio:
 - “¿No tienes cuenta? Regístrate”
 - “¿Ya tienes cuenta? Inicia sesión”.
-

5.6. MisPedidosPage.jsx y PedidoDetalle.jsx

- **MisPedidosPage:**
 - Listado de pedidos (PedidoResumen.jsx):
 - Nº pedido.
 - Fecha.
 - Estado (badge con color según estado: carrito, pendiente, pagado, enviado).
 - Total.
 - Click → abre detalle.
- **PedidoDetallePage.jsx:**
 - Cabecera con datos del pedido.

- Debajo, lista de líneas (PedidoDetalle.jsx):

- Imagen producto.
 - Nombre.
 - Cantidad.
 - Precio unitario.
 - Subtotal.
-

6. Componentes clave (resumen)

Ya teníamos definidos, con algunos añadidos:

- **Layout.jsx** → estructura general (Navbar, contenido, Footer).
 - **Navbar.jsx** → navegación principal + login/logout.
 - **Footer.jsx** → información básica, links, derechos.
 - **ProductoCard.jsx** → tarjeta de producto.
 - **FiltroCategorias.jsx** → sidebar o barra de categorías.
 - **PedidoResumen.jsx** → tarjeta para mis pedidos.
 - **PedidoDetalle.jsx** → línea de pedido.
 - **ProtectedRoute.jsx** → protege rutas “Mis pedidos”, etc.
 - **HeroCarousel.jsx** (nuevo) → carrusel de destacados.
 - **CategoryGrid.jsx** (nuevo) → tarjetas de categorías en Home.
-

7. Estados de usuario

1. **No autenticado:**

- Puede navegar por Home, Catálogo, ver productos.
- No puede ver “Mis pedidos”.
- Botones de “Añadir a mi pedido” muestran mensaje tipo “Inicia sesión para continuar”.

2. Cliente autenticado:

- Puede:
 - Ver Mis pedidos.
 - Crear pedido desde frontend (más adelante).
 - Ver detalle pedido.

3. (Opcional futuro) Admin:

- Vería enlaces extra (gestión de productos, etc.) – no lo montamos ahora.
-

8. Responsive y accesibilidad

- Puntos de corte recomendados:
 - `>= 1200px` → escritorio amplio.
 - `768–1199px` → tablet.
 - `< 768px` → móvil.
- Sidebar de categorías:
 - Desaparece a la izquierda en móvil y pasa a un menú superior con scroll horizontal o selector.
- Accesibilidad:
 - Botones con `:focus` claro.
 - Colores con contraste suficiente.
 - Textos alternativos en imágenes de producto.

- Formularios con `label` asociadas.
-

9. Integración con el backend (recordatorio rápido)

Front llamará a:

- GET `/api/productos` + filtros (`categoria_id`, `nombre`, paginación).
- GET `/api/productos/destacados`.
- GET `/api/productos/:id`.
- POST `/api/auth/register`, POST `/api/auth/login`.
- GET `/api/pedidos/mios`.
- GET `/api/pedidos/:id`.
- POST `/api/pedidos` (crear pedido).
- POST `/api/pedidos/:id/productos` (añadir productos).

Todo centralizado en `src/services/api.js`.

10. Plan de implementación (paso a paso)

1. **Maquetar Layout + Navbar + Footer** con la paleta definida.
2. Crear `HomePage.jsx`:
 - Hero “dummy” + secciones vacías.
 - Integrar luego `HeroCarousel` con `/productos/destacados`.
3. Crear `CatalogoPage.jsx` con:
 - FiltroCategorias (dummy) + grid de ProductoCard.

- Integrar llamada a `getProductos()` y filtro por `categoria_id`.
4. Crear **LoginPage** y **RegisterPage** conectados a `api.js` y usando el token.
 5. Crear **MisPedidosPage** + **PedidoDetallePage** consumiendo las rutas ya probadas en ApiTest.
 6. Pulir estilos, responsive y pequeños detalles (hover, estados de carga, etc.).