

# 👁 Proyecto E-commerce de Óptica - Guía Completa

## 📋 Índice de Fases

1. Stack Tecnológico Recomendado
  2. Arquitectura del Sistema
  3. Fase 1: Estructura Base
  4. Fase 2: Sistema de Autenticación
  5. Fase 3: Catálogo de Productos
  6. Fase 4: Carrito y Checkout
  7. Fase 5: Panel de Administración
  8. Fase 6: Panel de Cliente
- 

## 🔧 Stack Tecnológico Recomendado {#stack}

### Frontend

- **Framework:** Next.js 14+ con TypeScript
- **Estilado:** Tailwind CSS
- **Gestión de Estado:** Zustand o Redux Toolkit
- **UI Components:** shadcn/ui
- **Responsive:** Mobile-first approach (funciona en desktop y móvil automáticamente)

### Backend

- **API:** Next.js API Routes o tRPC

- **Base de datos:** PostgreSQL con Prisma ORM
- **Autenticación:** NextAuth.js
- **Almacenamiento:** Cloudinary o AWS S3 (imágenes de productos)

## Justificación

- Next.js ofrece SSR/SSG para SEO optimizado (crucial en e-commerce)
- TypeScript garantiza código mantenible
- Prisma facilita el modelado de dominio
- Responsive nativo sin código duplicado

---

## Arquitectura del Sistema {#arquitectura}

```
├── src/
│   ├── app/           # App Router (Next.js 14)
│   │   ├── (auth)/    # Grupo de rutas autenticadas
│   │   ├── (admin)/   # Panel administrativo
│   │   ├── (shop)/    # Tienda pública
│   │   └── api/       # API Routes
│   ├── components/    # Componentes reutilizables
│   ├── lib/           # Utilidades y configuraciones
│   ├── models/        # Modelos de dominio (TypeScript)
│   └── stores/        # Estado global (Zustand)
```

---

## FASE 1: Estructura Base y Diseño {#fase1}

### Prompt para IA:

Actúa como desarrollador senior especializado en e-commerce con Next.js 14 y TypeScript.

#### CONTEXTO:

Estoy construyendo un e-commerce de óptica (gafas, monturas, lentes, accesorios).

#### REFERENCIAS DE DISEÑO:

1. Funcionalidad similar a: <https://www.opticabogota.com/>
2. Estética inspirada en: plantilla minimalista Wix con paleta neutra

#### TAREA:

Crea la estructura base del proyecto con:

1. Layout principal responsive con:
  - Header sticky con logo, búsqueda, carrito y perfil
  - Footer con información de contacto y enlaces
  - Navigation bar con categorías
2. Página de inicio con:
  - Hero section con CTA principal
  - Grid de categorías (Monturas, Lentes, Accesorios)
  - Sección de productos destacados
  - Sección "Sobre Nosotros" resumida

#### ESPECIFICACIONES TÉCNICAS:

- Next.js 14 con App Router
- TypeScript strict mode
- Tailwind CSS con paleta:
  - \* Primarios: Neutros (grises #F5F5F5, #E5E5E5, #333333)
  - \* Acento: Azul moderno (#3B82F6) o verde menta (#10B981)

- Diseño mobile-first
- Componentes modulares en /components

#### PLACEHOLDERS REALISTAS:

- Logo: "OptiVision" o "Clarity Optics"
- Hero text: "La visión perfecta está a un clic de distancia"
- Categorías: Monturas de Sol | Monturas Oftálmicas | Lentes de Contacto | Accesorios
- Productos ejemplo: "Ray-Ban Aviator Clásico - \$89.990", "Oakley Deportivo Polarizado - \$124.990"

#### ENTREGABLES:

1. Código completo del layout
2. Página de inicio funcional
3. Sistema de navegación responsive

---

## FASE 2: Sistema de Autenticación {#fase2}

### Prompt para IA:

Desarrollador senior en sistemas de autenticación para e-commerce con NextAuth.js.

#### CONTEXTO:

Necesito implementar registro, login y gestión de sesiones para mi e-commerce de óptica.

#### REQUERIMIENTOS FUNCIONALES:

1. Registro de usuarios con:
  - Nombre completo
  - Email
  - Contraseña (mínimo 8 caracteres)
  - Teléfono (opcional)
  - Aceptación de términos

## 2. Login con:

- Email y contraseña
- Opción "Recordarme"
- Recuperación de contraseña

## 3. Roles de usuario:

- CLIENTE: puede comprar y ver su historial
- ADMIN: acceso al panel administrativo

## 4. Protección de rutas:

- /admin/\* solo para ADMIN
- /cuenta/\* solo para usuarios autenticados

## STACK TÉCNICO:

- NextAuth.js v5
- Prisma con PostgreSQL
- bcrypt para hashing
- Middleware de Next.js para protección de rutas

## MODELO DE DOMINIO (Prisma Schema):

```
``prisma
model User {
  id      String  @id @default(cuid())
  email    String  @unique
  name     String
  password String
  phone    String?
  role     Role    @default(CLIENTE)
  emailVerified DateTime?
  createdAt DateTime @default(now())
  orders   Order[]
}
```

```
enum Role {  
  CLIENTE  
  ADMIN  
}  
````
```

ENTREGABLES:

1. Configuración completa de NextAuth.js
2. Páginas de registro y login con validación
3. Middleware de protección de rutas
4. Componente de perfil de usuario

---

## FASE 3: Catálogo de Productos {#fase3}

### Prompt para IA:

Experto en arquitectura de e-commerce con enfoque en catálogos de productos.

CONTEXTO:

Catálogo de productos ópticos con características específicas.

MODELO DE DOMINIO (TypeScript):

```
````typescript  
// Categorías principales  
enum Categoria {  
  MONTURAS_SOL = 'monturas_sol',  
  MONTURAS_OFTALMICA = 'monturas_ofthalmica',  
  LENTES_CONTACTO = 'lentes_contacto',  
  ACCESORIOS = 'accesorios'  
}
```

```
// Producto base
interface Producto {
  id: string
  nombre: string
  marca: string
  categoria: Categoria
  precio: number
  descuento?: number
  imagenes: string[]
  descripcion: string
  características: Caracteristicas
  stock: number
  activo: boolean
}

// Características específicas
interface Caracteristicas {
  // Para monturas
  material?: 'acetato' | 'metal' | 'titanio' | 'plastico'
  color?: string
  forma?: 'rectangular' | 'redonda' | 'aviador' | 'cat-eye'
  genero?: 'hombre' | 'mujer' | 'unisex'

  // Para lentes
  graduacion?: string
  tipoLente?: 'monofocal' | 'bifocal' | 'progresivo'
  proteccionUV?: boolean
}
'''
```

#### FUNCIONALIDADES:

##### 1. Página de catálogo con:

- Filtros por categoría, marca, precio, características

- Ordenamiento (precio, popularidad, novedad)
- Paginación
- Vista grid responsive (4 cols desktop, 2 móvil)

2. Página de detalle de producto:

- Galería de imágenes con zoom
- Información completa del producto
- Selector de cantidad
- Botón "Agregar al carrito"
- Productos relacionados

3. Búsqueda inteligente:

- Autocompletado
- Búsqueda por nombre, marca, características

PLACEHOLDERS REALISTAS:

- "Ray-Ban Aviator Clásico RB3025" - \$89.990
- "Oakley Flak 2.0 XL Polarizado" - \$124.990
- "Lentes de Contacto Acuvue Oasys (6 und)" - \$42.990
- "Estuche Rígido Premium" - \$12.990

ENTREGABLES:

1. Schema de Prisma completo
2. API endpoints para productos
3. Página de catálogo funcional
4. Página de detalle de producto
5. Sistema de filtros y búsqueda

---

## FASE 4: Carrito y Checkout {#fase4}

### Prompt para IA:



Especialista en flujos de checkout y pasarelas de pago para e-commerce.

#### CONTEXTO:

Implementar carrito de compras persistente y proceso de checkout completo.

#### REQUERIMIENTOS:

##### 1. CARRITO DE COMPRAS:

- Persistencia en localStorage (no logueado) y BD (logueado)
- Agregar/eliminar/actualizar cantidad
- Cálculo automático de subtotal, descuentos, envío
- Contador visible en header
- Componente lateral deslizable

##### 2. PROCESO DE CHECKOUT (3 pasos):

###### PASO 1 - Información de envío:

- Nombre completo
- Dirección completa
- Ciudad/Departamento
- Teléfono de contacto
- Instrucciones adicionales (opcional)

###### PASO 2 - Método de pago:

- Tarjeta de crédito/débito (integración con Wompi o PayU)
- PSE
- Efectivo contra entrega

###### PASO 3 - Confirmación:

- Resumen completo del pedido
- Términos y condiciones
- Botón "Confirmar compra"

### 3. POST-COMPRA:

- Email de confirmación
- Página de "Pedido exitoso"
- Número de orden único
- Información de seguimiento

### MODELO DE DOMINIO:

``typescript

interface CarritoItem {

  productId: string

  cantidad: number

  precioUnitario: number

  subtotal: number

}

interface Orden {

  id: string

  numero: string // OP-2024-0001

  userId: string

  items: CarritoItem[]

  subtotal: number

  descuento: number

  costoEnvio: number

  total: number

  direccionEnvio: DireccionEnvio

  metodoPago: MetodoPago

  estadoPago: 'pendiente' | 'pagado' | 'fallido'

  estadoEnvio: 'procesando' | 'enviado' | 'entregado'

  createdAt: Date

}

``

#### PLACEHOLDERS:

- Dirección ejemplo: "Calle 123 #45-67, Apto 801, Bogotá"
- Costo de envío: \$8.000 (gratis sobre \$150.000)
- Tiempo de entrega: 3-5 días hábiles

#### ENTREGABLES:

1. Componente de carrito completo
2. Flujo de checkout paso a paso
3. Integración con pasarela de pago (mock o real)
4. Sistema de gestión de órdenes
5. Emails transaccionales

---

## FASE 5: Panel de Administración {#fase5}

### Prompt para IA:

Arquitecto de sistemas backend especializado en paneles administrativos.

#### CONTEXTO:

Panel completo para gestión del e-commerce (solo accesible por ADMIN).

#### MÓDULOS REQUERIDOS:

##### 1. DASHBOARD:

- Ventas del día/semana/mes (gráficos)
- Órdenes pendientes
- Productos con bajo stock
- Nuevos clientes

##### 2. GESTIÓN DE PRODUCTOS:

- Tabla con todos los productos

- CRUD completo (crear, leer, actualizar, eliminar)
- Carga múltiple de imágenes
- Gestión de stock
- Activar/desactivar productos
- Importación masiva (CSV)

### 3. GESTIÓN DE ÓRDENES:

- Lista de todas las órdenes
- Filtros por estado, fecha, cliente
- Actualizar estado de envío
- Imprimir guía de envío
- Historial completo

### 4. GESTIÓN DE CLIENTES:

- Lista de clientes registrados
- Historial de compras por cliente
- Exportar base de datos

### 5. GESTIÓN DE INVENTARIO:

- Control de stock por producto
- Alertas de stock bajo
- Reportes de rotación

### 6. CONFIGURACIÓN:

- Costos de envío
- Métodos de pago activos
- Información de la empresa
- Usuarios administradores

### CONSIDERACIONES TÉCNICAS:

- Usar React Server Components para mejor rendimiento
- Tablas paginadas (50 registros por página)
- Validación exhaustiva en servidor

- Logs de auditoría para cambios críticos

#### DISEÑO UI:

- Sidebar con navegación
- Tema claro/oscuro
- Tablas con sorting y búsqueda
- Modales para formularios
- Notificaciones toast para feedback

#### ENTREGABLES:

1. Layout del panel administrativo
2. Dashboard con métricas
3. CRUD completo de productos
4. Sistema de gestión de órdenes
5. Panel de configuración

---

## FASE 6: Panel de Cliente {#fase6}

### Prompt para IA:

Desarrollador UX especializado en áreas de cliente para e-commerce.

#### CONTEXTO:

Área privada donde los clientes pueden gestionar su cuenta y ver su historial.

#### SECCIONES DEL PANEL:

##### 1. MI CUENTA:

- Editar información personal
- Cambiar contraseña
- Preferencias de comunicación

## 2. MIS PEDIDOS:

- Historial completo de compras
- Estado de cada pedido
- Seguimiento de envío
- Opción de recompra rápida
- Descargar factura

## 3. DIRECCIONES GUARDADAS:

- Agregar/editar/eliminar direcciones
- Marcar dirección predeterminada

## 4. LISTA DE DESEOS:

- Productos guardados
- Notificación si baja de precio
- Agregar al carrito desde aquí

## 5. HISTORIAL DE GRADUACIÓN (específico óptica):

- Guardar prescripciones médicas
- Adjuntar fórmula en PDF
- Usar en próximas compras

## DISEÑO:

- Navegación lateral o tabs
- Cards para visualización de información
- Responsive con menú hamburguesa en móvil
- Paleta consistente con el resto del sitio

## EJEMPLO DE PEDIDO:

Pedido #OP-2024-1234

Fecha: 15 de enero, 2024

Total: \$234.980

Estado: En camino 📦

Artículos:

- Ray-Ban Aviator Clásico x1 - \$89.990
- Estuche Premium x1 - \$12.990
- Lentes Transitions x1 - \$132.000

Dirección de envío:

Carrera 7 #123-45, Bogotá

Estimado de entrega: 18 de enero, 2024

ENTREGABLES:

1. Layout del panel de cliente
2. Página de perfil editable
3. Historial de pedidos interactivo
4. Sistema de direcciones
5. Lista de deseos funcional

---

## Orden de Implementación Sugerido

### Semana 1-2: Fundamentos

- ☐ Fase 1: Estructura base y diseño

### Semana 3-4: Autenticación y Productos

- ☐ Fase 2: Sistema de autenticación

☐ Fase 3: Catálogo de productos (50%)

### **Semana 5-6: Catálogo y Carrito**

☐ Fase 3: Catálogo de productos (100%)

☐ Fase 4: Carrito y checkout

### **Semana 7-8: Paneles**

☐ Fase 5: Panel de administración

☐ Fase 6: Panel de cliente

### **Semana 9-10: Refinamiento**

☐ Testing

☐ Optimización de rendimiento

☐ SEO

☐ Deploy

---

### **Consideraciones Mobile-First**

Cada componente debe construirse pensando primero en móvil:

CSS



```
/* Approach recomendado en Tailwind */
```

```
/* Mobile (default) */
```

```
<div className="grid grid-cols-1 gap-4">
```

```
/* Tablet */
```

```
<div className="grid grid-cols-1 md:grid-cols-2 gap-4">
```

```
/* Desktop */
```

```
<div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-4 gap-4">
```

---

## **Tips Finales**

1. **Usa los prompts secuencialmente:** No intentes todo a la vez
  2. **Itera en cada fase:** Prueba, ajusta, mejora
  3. **Mantén el diseño consistente:** Usa un sistema de design tokens
  4. **Prioriza la performance:** Optimiza imágenes, usa lazy loading
  5. **Testing continuo:** En cada fase, prueba en mobile y desktop
- 

## **Recursos Adicionales**

- [Documentación Next.js 14](#)
- [Tailwind CSS](#)
- [Prisma ORM](#)
- [shadcn/ui Components](#)

---

### ¿Listo para comenzar?

Copia el prompt de la **Fase 1** y pégalo en tu IA favorita para empezar a construir tu e-commerce de óptica. Cada fase está diseñada para ser completada en 3-7 días de trabajo.