

# Gioielli 2025 - Documentación del Proyecto

---

## Descripción del Proyecto

**Gioielli 2025** es una aplicación web de e-commerce desarrollada en React que permite a los usuarios navegar por un catálogo de productos, gestionar un carrito de compras y realizar operaciones CRUD sobre los productos. La aplicación está diseñada con un enfoque moderno, responsivo y optimizada para una experiencia de usuario excepcional.

## Objetivos del Proyecto

- Crear una plataforma de e-commerce completa y funcional
  - Implementar un sistema de autenticación de usuarios
  - Proporcionar una interfaz intuitiva para la gestión de productos
  - Garantizar la responsividad en todos los dispositivos
  - Optimizar el rendimiento y la accesibilidad
- 

## ★ Características Principales

### Gestión del Carrito

- Agregar productos al carrito desde el catálogo
- Modificar cantidades de productos
- Eliminar productos individuales
- Vaciar el carrito completo
- Persistencia del carrito en localStorage

### Autenticación de Usuarios

- Sistema de login simulado
- Protección de rutas privadas
- Estado de autenticación global
- Redirección automática para usuarios no autenticados

### CRUD de Productos

- Crear nuevos productos con validación
- Editar productos existentes
- Eliminar productos con confirmación

- Visualización del catálogo completo

## **Búsqueda y Filtros**

- Búsqueda en tiempo real por nombre
- Filtros por categoría
- Paginación de resultados
- Navegación fluida entre páginas

## **Diseño Responsivo**

- Adaptación a móviles, tablets y escritorio
- Interfaz moderna con Bootstrap
- Componentes estilizados personalizados
- Iconografía con React Icons

---

## **Tecnologías Utilizadas**

### **Core**

- **React 19.1.0** - Framework principal
- **React Router DOM 7.7.0** - Navegación y rutas
- **React Scripts 5.0.1** - Herramientas de desarrollo

### **Estado y Contexto**

- **Context API** - Gestión del estado global
- **useState & useEffect** - Hooks de React

### **Estilos y UI**

- **Bootstrap 5.3.7** - Framework CSS
- **Styled Components 6.1.19** - CSS-in-JS
- **React Icons 5.5.0** - Iconografía
- **React Slick 0.30.3** - Carruseles

### **Funcionalidades Adicionales**

- **Axios 1.10.0** - Cliente HTTP
- **React Toastify 11.0.5** - Notificaciones
- **React Helmet 6.1.0** - SEO y metadatos
- **React Modal 3.16.3** - Ventanas modales

## Testing

- **@testing-library/react 16.3.0** - Pruebas de componentes
  - **@testing-library/jest-dom 6.6.3** - Matchers adicionales
  - **@testing-library/user-event 13.5.0** - Simulación de eventos
- 

## Instalación y Configuración

### Prerequisitos

- Node.js (versión 14 o superior)
  - npm o yarn
  - Git
- 

## Funcionalidades Implementadas

### Requerimiento #1: Gestión del Carrito y Autenticación

#### CarritoContext

- **Estado global:** Productos, cantidades, total
- **Funciones:** addToCart, removeFromCart, clearCart, updateQuantity
- **Persistencia:** localStorage para mantener el carrito entre sesiones

#### AuthContext

- **Estado:** isAuthenticated, user, loading
- **Funciones:** login, logout, checkAuth
- **Credenciales:** Usuario: user, Contraseña: password

#### Rutas Protegidas

### Requerimiento #2: CRUD de Productos

#### Formulario de Productos

- **Validaciones implementadas:**
  - Nombre obligatorio (mínimo 3 caracteres)
  - Precio mayor a 0
  - Descripción mínima de 10 caracteres

- Categoría seleccionada

## **Operaciones CRUD**

- **CREATE:** POST a MockAPI con validación
- **READ:** GET con paginación y filtros
- **UPDATE:** PUT con confirmación
- **DELETE:** DELETE con modal de confirmación

## **Manejo de Errores**

- Estados de carga durante operaciones
- Mensajes de error específicos
- Retry automático en fallos de red
- Toasts para feedback al usuario

## **Requerimiento #3: Optimización de Diseño**

### **Sistema de Grillas Bootstrap**

#### **Styled Components**

- Componentes reutilizables
- Temas dinámicos
- Props-based styling
- Animaciones CSS

#### **Accesibilidad**

- Etiquetas ARIA
- Contraste adecuado
- Navegación por teclado
- Texto alternativo en imágenes

## **Requerimiento #4: Búsqueda y Paginación**

### **Barra de Búsqueda**

- Búsqueda en tiempo real (debounced)
- Filtros por nombre y categoría
- Resultados destacados
- Historial de búsquedas

#### **Paginación**

- Navegación por páginas

- Información de resultados
- Tamaño de página configurable
- URLs amigables

## Requerimiento #5: Preparación para Despliegue

### Optimizaciones

- Code splitting
- Lazy loading de componentes
- Compresión de imágenes
- Minificación de assets

### Compatibilidad

- ✓ Chrome (últimas 2 versiones)
  - ✓ Firefox (últimas 2 versiones)
  - ✓ Safari (últimas 2 versiones)
  - ✓ Edge (últimas 2 versiones)
- 

## Guía de Uso

### Para Usuarios Finales

1. **Navegación**
  - Explorar el catálogo desde la página principal
  - Usar la barra de búsqueda para encontrar productos específicos
  - Filtrar por categorías
2. **Gestión del Carrito**
  - Agregar productos haciendo clic en "Agregar al Carrito"
  - Modificar cantidades desde el ícono del carrito
  - Proceder al checkout (funcionalidad simulada)
3. **Autenticación**
  - Iniciar sesión con: Usuario: `user`, Contraseña: `password`
  - Acceder a funcionalidades premium tras autenticarse

### Para Administradores

#### Gestión de Productos

- Acceder al panel de administración
- Crear nuevos productos con el formulario
- Editar productos existentes

- Eliminar productos (con confirmación)

---

## API y Servicios

MockAPI }

---

## ☐ Componentes Principales

ProductCard

SearchBar

Pagination

---

## Estilos y Diseño

Tema de Colores

Componentes Estilizados

Responsividad

- **Mobile First:** Diseño optimizado para dispositivos móviles
- **Breakpoints:** Tablet (768px), Desktop (992px), Large (1200px)
- **Flexbox/Grid:** Layouts modernos y flexibles
- **Touch-friendly:** Botones y enlaces apropiados para touch

---

## ☐ Pruebas y Optimización

Estrategia de Testing

Performance

- **Code Splitting:** Carga lazy de componentes
- **Memoización:** React.memo para componentes pesados

- **Imágenes optimizadas:** WebP con fallback
- **Bundle analysis:** Webpack Bundle Analyzer

## Métricas de Rendimiento

- **First Contentful Paint:** < 1.5s
  - **Largest Contentful Paint:** < 2.5s
  - **Cumulative Layout Shift:** < 0.1
  - **Time to Interactive:** < 3.5s
- 

## Despliegue

### Build de Producción

```
npm run build
```

### Opciones de Despliegue

#### Netlify

1. Conectar repositorio de GitHub
2. Configurar comando de build: `npm run build`
3. Directorio de salida: `build`
4. Deploy automático en push

#### Vercel

1. Importar proyecto desde GitHub
2. Configuración automática detectada
3. Deploy en cada commit

### Servidor Propio

```
# Usar servidor estático
npm install -g serve
serve -s build -l 3000
```

### Variables de Entorno

```
# Producción
REACT_APP_API_URL=https://api.production.com
REACT_APP_ENVIRONMENT=production

# Desarrollo
REACT_APP_API_URL=https://api.development.com
```

## Mantenimiento

### Actualizaciones

- **Dependencias:** Revisar mensualmente
- **Seguridad:** Auditorías con `npm audit`
- **Performance:** Monitoreo continuo
- **Compatibilidad:** Testing en navegadores

### Monitoreo

- **Error tracking:** Sentry o similar
- **Analytics:** Google Analytics
- **Performance:** Lighthouse CI
- **Uptime:** Monitoreo de disponibilidad

### Backup

- **Código:** Repositorio Git
- **Configuración:** Variables de entorno documentadas
- **Base de datos:** Backups automáticos de MockAPI

---

## Credenciales de Prueba

- **Usuario:** `user`
- **Contraseña:** `password`

## Documentación Adicional

- Consultar comentarios en el código
- Revisar tests para ejemplos de uso
- Documentación de dependencias en sus respectivos sitios

## Resolución de Problemas

1. **Verificar dependencias:** `npm install`
2. **Limpiar cache:** `npm start -- --reset-cache`
3. **Revisar logs:** Consola del navegador
4. **Validar API:** Verificar endpoints de MockAPI



---

## Notas de Versión

### **v1.0.0 (Actual)**

- ✓ Implementación completa de todos los requerimientos
- ✓ Gestión de carrito y autenticación
- ✓ CRUD completo de productos
- ✓ Diseño responsivo y optimizado
- ✓ Búsqueda y paginación funcionales
- ✓ Preparado para despliegue en producción