

SETUP_LOCAL.md - Guía de Implementación Local

Proyecto: Sistema ERP B2B Móvil

Versión: v1.0

Actualizado: 24 de agosto de 2025

Esta guía te permitirá configurar y ejecutar localmente el Sistema ERP B2B Móvil completo, incluyendo el Backend-For-Frontend (BFF) y la aplicación móvil React Native/Expo.

1. Pre-requisitos

1.1 Software Requerido

Esenciales:

- **Node.js v18.0+** - [Descargar aquí](https://nodejs.org/) (https://nodejs.org/)
- **npm v8.0+** o **yarn v1.22+** - Incluido con Node.js
- **PostgreSQL v13+** - [Descargar aquí](https://www.postgresql.org/download/) (https://www.postgresql.org/download/)
- **Git** - Para clonar repositorios

Para App Móvil:

- **Expo CLI** - Se instalará automáticamente
- **Expo Go App** en tu móvil:
 - iOS: [App Store](https://apps.apple.com/app/expo-go/id982107779) - [Expo Go](https://apps.apple.com/app/expo-go/id982107779) (https://apps.apple.com/app/expo-go/id982107779)
 - Android: [Google Play](https://play.google.com/store/apps/details?id=host.exp.exponent) - [Expo Go](https://play.google.com/store/apps/details?id=host.exp.exponent) (https://play.google.com/store/apps/details?id=host.exp.exponent)

Opcional (para simuladores):

- **Android Studio** - Para emulador Android
- **Xcode** - Para simulador iOS (solo macOS)

1.2 Verificar Instalación

```
# Verificar versiones
node --version # Debe ser v18.0+
npm --version  # Debe ser v8.0+
psql --version # Debe ser v13.0+
```

2. Configuración del Backend (BFF)

2.1 Navegar a la carpeta del backend

```
cd exp-system/app
```

2.2 Instalar dependencias

```
# Usando npm
npm install

# O usando yarn (recomendado)
yarn install
```

2.3 Configurar Base de Datos PostgreSQL

Crear base de datos:

```
-- Conectar a PostgreSQL como superusuario
psql -U postgres

-- Crear usuario y base de datos
CREATE USER erp_user WITH PASSWORD 'erp_password';
CREATE DATABASE erp_b2b_mobile OWNER erp_user;
GRANT ALL PRIVILEGES ON DATABASE erp_b2b_mobile TO erp_user;
\q
```

2.4 Crear archivo de variables de entorno

```
# Crear archivo .env en la raíz de erp-system/app/
touch .env
```

Contenido del archivo .env :

```
# Database
DATABASE_URL="postgresql://erp_user:erp_password@localhost:5432/erp_b2b_mobile"

# JWT Secrets
JWT_SECRET="tu-jwt-secret-super-seguro-aqui"
NEXTAUTH_SECRET="tu-nextauth-secret-super-seguro-aqui"

# Firebase Cloud Messaging (Opcional - para notificaciones push)
FIREBASE_PROJECT_ID="tu-proyecto-firebase"
FIREBASE_PRIVATE_KEY="-----BEGIN PRIVATE KEY-----\ntu-private-key-aqui\n-----END\nPRIVATE KEY-----\n"
FIREBASE_CLIENT_EMAIL="firebase-adminsdk@tu-proyecto.iam.gserviceaccount.com"

# Almacenamiento (Opcional)
STORAGE_BUCKET_URL="https://tu-bucket.s3.amazonaws.com"
STORAGE_ACCESS_KEY="tu-access-key"
STORAGE_SECRET_KEY="tu-secret-key"
```

2.5 Generar cliente Prisma y ejecutar migraciones

```
# Generar cliente de Prisma
npx prisma generate

# Crear y aplicar migraciones (primera vez)
npx prisma db push

# Alternativamente, si usas migraciones:
# npx prisma migrate dev --name init
```

2.6 Poblar base de datos con datos de prueba

```
# Ejecutar script de seed
npx prisma db seed
```

Salida esperada:

```
🌱 Iniciando seed de la base de datos...
👤 Creando usuario admin...
👉 Creando tipos de cliente...
📁 Creando categorías...
📏 Creando unidades de medida...
🏢 Creando sucursales...
📦 Creando almacenes...
📦 Creando productos...
👤 Creando vendedores...
👤 Creando clientes demo...
💰 Creando precios...
📊 Creando inventario...

✅ Seed completado exitosamente!

🔑 Credenciales de prueba:
- Admin: john@doe.com / johndoe123
- Cliente 1: alexander.godoy@example.com / demo123
- Cliente 2: maria.gonzalez@example.com / demo123
```

2.7 Iniciar el servidor de desarrollo

```
# Usando npm
npm run dev

# O usando yarn
yarn dev
```

Verificar que funciona:

- Abrir navegador en: `http://localhost:3000`
- Deberías ver el dashboard del Sistema ERP B2B Móvil
- Verificar estadísticas: 2 clientes, 3 productos, 0 pedidos, 1 almacén

3. Configuración de la App Móvil

3.1 Navegar a la carpeta de la app móvil

```
cd ../erp-mobile  
# 0 desde la raíz: cd erp-mobile
```

3.2 Instalar dependencias

```
npm install
```

3.3 Configurar URL del API

Archivo: `src/services/api.ts`

```
// Línea ~18: Verificar que la URL apunte a tu BFF local  
const BASE_URL = 'http://localhost:3000'; // Para web y iOS simulator  
// const BASE_URL = 'http://10.0.2.2:3000'; // Para Android emulator
```

Para diferentes entornos:

- **Web/iOS Simulator:** `http://localhost:3000`
- **Android Emulator:** `http://10.0.2.2:3000`
- **Dispositivo móvil real:** `http://TU-IP-LOCAL:3000` (ej: `http://192.168.1.100:3000`)

3.4 Verificar configuración de Expo

Archivo: `app.config.js` (ya configurado)

```
export default {  
  expo: {  
    name: "ERP B2B Mobile",  
    slug: "exp-mobile",  
    version: "1.0.0",  
    // ... resto de configuración  
  }  
};
```

3.5 Iniciar la aplicación móvil

```
# Iniciar Expo  
npx expo start  
  
# 0 alternativamente  
npm run web # Solo para web  
npm run ios # Solo para iOS (requiere macOS)  
npm run android # Solo para Android
```

Opciones de ejecución:

1. **En navegador web:** Presiona `w` en la terminal → Se abre `http://localhost:8081`
2. **En móvil real:** Escanea el código QR con la app Expo Go
3. **En simulador:** Presiona `i` (iOS) o `a` (Android) en la terminal

4. Pasos de Verificación

4.1 Verificar Backend (BFF)

Prueba 1 - Página principal:

```
curl http://localhost:3000
# Debe devolver HTML del dashboard
```

Prueba 2 - Endpoint de pre-login:

```
curl -X POST http://localhost:3000/api/auth/mobile/pre-login \
-H "Content-Type: application/json" \
-d '{"email": "alexander.godoy@example.com"}'

# Respuesta esperada:
# {"success":true,"data":{"name":"ALEXANDER GODOY CASTELLON","address":"Av. Comercial
789, Ciudad","phone":"70123456","nit":""}}
```

Prueba 3 - Endpoints protegidos (deben devolver 401):

```
curl http://localhost:3000/api/products
# Respuesta esperada: 401 Unauthorized
```

4.2 Verificar App Móvil

Prueba 1 - Acceso web:

1. Abrir `http://localhost:8081` en navegador
2. Debe mostrar la pantalla de pre-login
3. No debe haber errores en consola del navegador (F12)

Prueba 2 - Flujo completo de login:

1. **Email:** `alexander.godoy@example.com`
2. **Presionar:** "Continuar"
3. **Debe mostrar:** Datos del cliente y selectores
4. **Seleccionar:** "Sucursal Principal"
5. **Seleccionar:** "Almacén Central"
6. **Contraseña:** `demo123`
7. **Presionar:** "Iniciar Sesión"
8. **Debe mostrar:** Pantalla principal con productos

Prueba 3 - Funcionalidades core:

1. **Catálogo:** Ver 3 productos (Fanta, Coca-Cola, Vital)
2. **Carrito:** Agregar productos y ver contador en tab
3. **Navegación:** Probar todos los tabs (Inicio, Favoritos, Pedidos, Mi carrito)
4. **Pedidos:** Ver historial (vacío inicialmente)

4.3 Verificar Integración BFF ↔ App Móvil

Prueba de integración completa:

1. **Login exitoso** en app móvil
2. **Ver productos** en pantalla principal
3. **Agregar productos al carrito**

4. **Crear pedido** desde el carrito
5. **Verificar en BFF** que el pedido se creó:

```
bash
# En otra terminal, verificar base de datos
psql -U erp_user -d erp_b2b_mobile
SELECT * FROM "Order";
```

5. Solución de Problemas Comunes

5.1 Error de conexión a base de datos

```
# Verificar que PostgreSQL esté corriendo
sudo systemctl status postgresql # Linux
brew services list | grep postgresql # macOS

# Verificar conexión
psql -U erp_user -d erp_b2b_mobile -h localhost
```

5.2 Error “Cannot find module @prisma/client”

```
cd erp-system/app
npx prisma generate
```

5.3 App móvil no conecta con BFF

1. **Verificar URL** en `src/services/api.ts`
2. **Para dispositivo real, usar IP local:**

```
bash
# Encontrar tu IP local
ifconfig | grep inet # macOS/Linux
ipconfig | findstr IPv4 # Windows
```

3. **Verificar firewall/antivirus**

5.4 Expo no inicia

```
# Limpiar cache
npx expo start --clear

# Reinstalar dependencias
rm -rf node_modules package-lock.json
npm install
```

6. Comandos de Desarrollo Útiles

6.1 Backend (BFF)

```
# Ver logs de base de datos
npx prisma studio # Interfaz web para ver datos

# Reset completo de BD
npx prisma db push --force-reset
npx prisma db seed

# Generar nueva migración
npx prisma migrate dev --name mi_nueva_migracion
```

6.2 App Móvil

```
# Limpiar cache de Metro
npx expo start --clear

# Generar build para testing
npx expo build:android # Android APK
npx expo build:ios     # iOS (requiere Apple Developer)

# Ejecutar en diferentes plataformas
npx expo start --web    # Solo web
npx expo start --ios    # Solo iOS
npx expo start --android # Solo Android
```

7. Estructura de Archivos Importantes

7.1 Backend (erp-system/app/)

```
├── app/
│   ├── api/
│   │   ├── auth/mobile/ # Endpoints API
│   │   ├── products/    # Auth móvil
│   │   ├── orders/      # Catálogo
│   │   └── favorites/    # Pedidos
│   └── page.tsx          # Favoritos
├── prisma/
│   └── schema.prisma     # Dashboard principal
├── scripts/
│   └── seed.ts           # Schema de BD
└── .env                  # Datos de prueba
                        # Variables de entorno
```

7.2 App Móvil (erp-mobile/)

src/	
components/	# Componentes UI
screens/	# Pantallas principales
services/	# Servicios API
store/	# Estado global (Zustand)
types/	# Tipos TypeScript
constants/	# Constantes y tema
assets/	# Iconos e imágenes
app.config.js	# Configuración Expo

8. Credenciales de Prueba

8.1 Para testing de la aplicación:

```
{
  "admin": {
    "email": "john@doe.com",
    "password": "johndoe123"
  },
  "cliente_principal": {
    "email": "alexander.godoy@example.com",
    "password": "demo123",
    "sucursal": "Sucursal Principal",
    "almacen": "Almacén Central"
  },
  "cliente_secundario": {
    "email": "maria.gonzalez@example.com",
    "password": "demo123"
  }
}
```

8.2 Productos disponibles:

- **SKU 2519:** Fanta Naranja Vidrio Retornable 190 MI - Bs. 31.20
- **SKU 2518:** Coca-Cola Original 500 MI - Bs. 27.90
- **SKU 2520:** Vital Sin Gas 2 Lt - Bs. 91.98

🎉 ¡Configuración Completa!

Si has seguido todos los pasos correctamente, ahora deberías tener:

- ✅ Backend ejecutándose en `http://localhost:3000`
- ✅ App móvil ejecutándose en `http://localhost:8081`
- ✅ Base de datos poblada con datos de prueba
- ✅ Integración completa funcionando

Para soporte adicional o problemas específicos, revisar los logs de ambos servicios y verificar que todas las dependencias estén correctamente instaladas.