





# Guía de Importación a DeepAgent - Sistema ERP v4.0

Esta guía te ayudará a importar y continuar el desarrollo del Sistema ERP Completo en una nueva cuenta de DeepAgent.

## Resumen del Proyecto

**Sistema ERP Completo v4.0** es una solución integral desarrollada en 4 fases:

-  **FASE 1:** Gestión básica (clientes, productos, ventas, pagarés)
-  **FASE 2:** Crédito y garantías (notas, reestructuras, garantías)
-  **FASE 3:** Analytics y reportes (dashboard, reportes, configuración)
-  **FASE 4:** Automatización avanzada (compras, BI, auditoría, facturación electrónica)

## Estructura Completa de Archivos

### Archivos Base del Proyecto

```
sistema_erp_completo/
├── app/
│   ├── package.json          # Dependencias y scripts
│   ├── next.config.js       # Configuración Next.js
│   ├── tailwind.config.ts   # Configuración Tailwind
│   ├── tsconfig.json        # Configuración TypeScript
│   ├── .env.example         # Variables de entorno ejemplo
│   ├── .env                 # Variables de entorno (configurar)
│   └── app/
│       ├── layout.tsx       # Layout principal con PWA
│       ├── page.tsx         # Dashboard principal
│       ├── globals.css      # Estilos globales
│       └── ... (módulos detallados abajo)
```

## Módulos Principales (Páginas)

### FASE 1 - Gestión Básica

- `app/clientes/page.tsx` - Gestión de clientes
- `app/productos/page.tsx` - Catálogo de productos
- `app/pedidos/page.tsx` - Sistema de pedidos
- `app/ventas/page.tsx` - Gestión de ventas
- `app/pagares/page.tsx` - Control de pagarés

### FASE 2 - Crédito y Garantías

- `app/notas-cargo/page.tsx` - Notas de cargo
- `app/notas-credito/page.tsx` - Notas de crédito
- `app/reestructuras/page.tsx` - Reestructuras de crédito
- `app/garantias/page.tsx` - Sistema de garantías

## FASE 3 - Analytics y Reportes

- `app/reportes/page.tsx` - Sistema de reportes
- `app/configuracion/page.tsx` - Configuraciones del sistema



## FASE 4 - Módulos Avanzados

- `app/compras/page.tsx` - **Módulo de compras y proveedores**
- `app/automatizacion/page.tsx` - **Sistema de automatización**
- `app/auditoria/page.tsx` - **Auditoría y seguridad**
- `app/facturacion-electronica/page.tsx` - **Facturación electrónica (CFDI)**
- `app/business-intelligence/page.tsx` - **Business Intelligence y BI**



## APIs REST Completas

### APIs Base (FASES 1-3)

```

app/api/
├── auth/                                # Autenticación NextAuth
├── clientes/route.ts                    # CRUD clientes
├── productos/route.ts                  # CRUD productos
├── pedidos/route.ts                     # Gestión de pedidos
├── ventas/route.ts                      # Gestión de ventas
├── pagares/route.ts                    # Control de pagarés
├── notas-cargo/route.ts                 # Notas de cargo
├── notas-credito/route.ts               # Notas de crédito
├── reestructuras/route.ts               # Reestructuras
├── garantias/route.ts                   # Garantías
├── reportes/                            # Sistema de reportes
│   ├── ventas/route.ts
│   ├── cobranza/route.ts
│   └── inventario/route.ts
├── dashboard/analytics/route.ts         # Métricas dashboard
├── configuracion/route.ts               # Configuraciones
├── integraciones/                       # Integraciones externas
│   ├── webhooks/route.ts
│   └── sync/route.ts

```
















### APIs FASE 4 (Nuevas)

```

app/api/
├── compras/                            #  APIs de Compras
│   ├── proveedores/route.ts           # Gestión de proveedores
│   ├── ordenes/route.ts               # Órdenes de compra
│   └── recepciones/route.ts            # Recepción de mercancía
├── automatizacion/                     #  APIs de Automatización
│   ├── workflows/route.ts              # Workflows automáticos
│   ├── tasks/route.ts                  # Tareas programadas
│   └── notifications/route.ts           # Reglas de notificación
├── auditoria/                          #  APIs de Auditoría
│   ├── logs/route.ts                  # Logs de actividad
│   ├── security/route.ts               # Eventos de seguridad
│   └── changes/route.ts                 # Control de cambios
├── sistema/                           #  APIs de Sistema
│   ├── backup/route.ts                 # Sistema de backups
│   └── sincronizacion/route.ts          # Sincronización externa

```

## Componentes y Librerías

app/components/	
 ui/	# Componentes Shadcn/UI (40+ componentes)
 navigation/	
 sidebar.tsx	#  Navegación actualizada v4.0
 forms/	# Formularios especializados
 providers/	# Providers de contexto
 pwa-install.tsx	# Instalación PWA
app/lib/	
 auth.ts	# Configuración NextAuth
 prisma.ts	# Cliente Prisma
 utils.ts	# Utilidades generales
 types.ts	# Tipos TypeScript
 offline-storage.ts	# Almacenamiento offline
 aws-config.ts	# Configuración AWS S3

## Proceso de Importación Paso a Paso

### 1. Preparación del Entorno

#### Crear nuevo proyecto en DeepAgent

1. Inicia una nueva conversación en DeepAgent
2. Solicita: "Crear un nuevo proyecto Next.js 14 con TypeScript, Tailwind y Shadcn/UI"
3. Espera a que se configure la estructura base

#### Verificar dependencias requeridas

```
{
  "dependencies": {
    "next": "14.2.28",
    "react": "18.2.0",
    "typescript": "5.2.2",
    "@prisma/client": "6.7.0",
    "next-auth": "4.24.11",
    "recharts": "2.15.3",
    "@radix-ui/react-*": "latest",
    "lucide-react": "0.446.0",
    "tailwindcss": "3.3.3"
  }
}
```

### 2. Importación por Fases

#### Método Recomendado: Importación Gradual

##### PASO 1: Configuración Base

Copia estos archivos primero:

-  package.json (verificar dependencias)
-  next.config.js
-  tailwind.config.ts
-  tsconfig.json
-  app/layout.tsx
-  app/globals.css
-  .env.example

## PASO 2: Componentes Base

Copia la estructura de componentes:

- └─ app/components/ui/ (todos los archivos)
- └─ app/components/providers/
- └─ app/lib/ (todos los archivos)

## PASO 3: FASE 1 - Módulos Básicos

- └─ app/clientes/page.tsx + /api/clientes/route.ts
- └─ app/productos/page.tsx + /api/productos/route.ts
- └─ app/pedidos/page.tsx + /api/pedidos/route.ts
- └─ app/ventas/page.tsx + /api/ventas/route.ts
- └─ app/pagares/page.tsx + /api/pagares/route.ts

## PASO 4: FASE 2 - Crédito y Garantías

- └─ app/notas-cargo/page.tsx + /api/notas-cargo/route.ts
- └─ app/notas-credito/page.tsx + /api/notas-credito/route.ts
- └─ app/reestructuras/page.tsx + /api/reestructuras/route.ts
- └─ app/garantias/page.tsx + /api/garantias/route.ts

## PASO 5: FASE 3 - Analytics

- └─ app/reportes/page.tsx + /api/reportes/
- └─ app/configuracion/page.tsx + /api/configuracion/route.ts
- └─ app/api/dashboard/analytics/route.ts
- └─ app/api/integraciones/

## PASO 6: FASE 4 - Módulos Avanzados

- └─ app/compras/page.tsx + /api/compras/
- └─ app/automatizacion/page.tsx + /api/automatizacion/
- └─ app/auditoria/page.tsx + /api/auditoria/
- └─ app/facturacion-electronica/page.tsx
- └─ app/business-intelligence/page.tsx + /api/business-intelligence/
- └─ /api/sistema/ (backup y sincronización)

## PASO 7: Navegación Actualizada

- └─ app/components/navigation/sidebar.tsx (versión v4.0)

### 3. Configuración de Variables de Entorno

```
# Variables básicas (ya existentes)
DATABASE_URL="postgresql://..."
NEXTAUTH_URL="http://localhost:3000"
NEXTAUTH_SECRET="..."

#  Nuevas variables FASE 4
# Facturación Electrónica
PAC_URL="https://api.tu-pac.com"
PAC_USER="tu-usuario-pac"
PAC_PASSWORD="tu-password-pac"
CSD_CERTIFICATE_PATH="/path/to/certificate.cer"
CSD_PRIVATE_KEY_PATH="/path/to/private.key"
CSD_PASSWORD="password-del-certificado"

# OpenPay (configurado)
OPENPAY_MERCHANT_ID="[DISPONIBLE]"
OPENPAY_PRIVATE_KEY="[DISPONIBLE]"
OPENPAY_PUBLIC_KEY="[DISPONIBLE]"

# Cloud Storage
CLOUD_STORAGE_URL="https://storage.cloud.com"
CLOUD_STORAGE_ACCESS_KEY="tu-access-key"
CLOUD_STORAGE_SECRET_KEY="tu-secret-key"

# Business Intelligence
BI_API_KEY="tu-api-key-bi"
ML_MODEL_ENDPOINT="https://api.ml-service.com"
```

### 4. Verificación y Testing









Después de cada fase, verificar:

```
# Instalar dependencias
yarn install

# Verificar compilación
yarn build

# Verificar funcionamiento
yarn dev
```

#### Checklist de verificación:

- [ ]  Navegación funciona entre módulos
- [ ]  APIs responden correctamente (datos simulados)
- [ ]  Componentes UI se renderizan sin errores
- [ ]  No hay errores de TypeScript
- [ ]  Estilos se aplican correctamente
- [ ]  Nuevos módulos FASE 4 son accesibles
- [ ]  Business Intelligence muestra gráficos
- [ ]  Sistema de auditoría registra eventos

## Comandos de Utilidad para DeepAgent

### Solicitudes Útiles para Continuar el Desarrollo

#### Para debugging:

"Revisar errores de compilación en el sistema ERP y corregir problemas de TypeScript"

#### Para nuevas funcionalidades:

"Agregar validación de formularios en el módulo de [MODULO] usando Zod y react-hook-form"

#### Para integraciones:

"Implementar integración real con OpenPay para procesar pagos en el módulo de pagarés"

#### Para base de datos:

"Crear el esquema de Prisma completo para las tablas de [MODULO] con relaciones"

#### Para testing:

"Crear tests unitarios para las APIs del módulo [MODULO] usando Jest"

### Estado Actual de Integraciones

#### Configurado y Listo

- **OpenPay:** Credenciales disponibles en variables de entorno
- **NextAuth:** Sistema de autenticación funcionando
- **Prisma:** ORM configurado (esquema pendiente de completar)
- **Shadcn/UI:** Componentes instalados y funcionando
- **Tailwind:** Estilos aplicados correctamente

#### En Desarrollo (Requiere Configuración)

- **PAC para CFDI:** Requiere credenciales reales de proveedor
- **Base de Datos:** Esquema completo pendiente
- **Cloud Storage:** Para backups automáticos
- **APIs ML:** Para Business Intelligence predictivo

#### Pendientes (Próximas Fases)

- Tests automatizados (Jest/Cypress)
- Migraciones de base de datos
- Deployment en producción
- Documentación API con OpenAPI
- App móvil (React Native)

## Documentación de Referencia

---

### Archivos de Documentación Incluidos

- `README_UPDATED.md` - Documentación completa v4.0
- `CHANGELOG.md` - Historial de cambios detallado
- `API_REFERENCE.md` - Referencias de APIs
- `DATABASE_SCHEMA.md` - Esquema de base de datos
- `DEPLOYMENT_GUIDE.md` - Guía de despliegue

### Recursos Adicionales

- [Next.js 14 Documentation](https://nextjs.org/docs) (<https://nextjs.org/docs>)
- [Prisma Documentation](https://www.prisma.io/docs) (<https://www.prisma.io/docs>)
- [Shadcn/UI Components](https://ui.shadcn.com) (<https://ui.shadcn.com>)
- [Tailwind CSS](https://tailwindcss.com/docs) (<https://tailwindcss.com/docs>)

## Notas Importantes

---

### Aspectos Críticos a Considerar

1. **Seguridad:** Las APIs actuales usan datos simulados. Implementar validación real antes de producción.
2. **Performance:** Los módulos de BI pueden ser pesados. Considerar lazy loading.
3. **Escalabilidad:** El sistema está diseñado para crecer. Mantener la estructura modular.
4. **Backup:** Implementar backups reales antes de usar datos importantes.
5. **Compliance:** El módulo de facturación electrónica requiere certificados SAT válidos.

### Limitaciones Actuales

- Datos simulados en todas las APIs
- Esquema de base de datos incompleto
- Integraciones externas no probadas en producción
- Tests automatizados pendientes
- Documentación API incompleta

### Próximos Pasos Recomendados

1. Completar esquema de base de datos con Prisma
  2. Implementar validaciones reales en APIs
  3. Configurar integraciones externas
  4. Crear tests automatizados
  5. Preparar para deployment en producción
-

## Soporte

---

Si tienes problemas durante la importación:

1. **Revisa los logs** de compilación para errores específicos
2. **Verifica dependencias** - todas deben estar en package.json
3. **Confirma estructura** - todos los archivos en las ubicaciones correctas
4. **Consulta documentación** - README\_UPDATED.md tiene información detallada

¡Buena suerte con tu proyecto ERP! 🚀

---

💡 **Tip:** Importa el proyecto gradualmente, fase por fase, para identificar problemas temprano y mantener la estabilidad del sistema.