

Guía de Importación a DeepAgent

Importar este Proyecto ERP en Otra Cuenta DeepAgent

Esta guía te ayudará a importar exitosamente el **Sistema ERP Completo v4.0** en otra cuenta de DeepAgent para continuar el desarrollo o realizar modificaciones.

Objetivo

Transferir completamente el proyecto ERP de una cuenta DeepAgent a otra, manteniendo toda la funcionalidad, estructura y configuraciones.

Preparación Previa

Requisitos

- Cuenta DeepAgent activa (destino)
- Acceso a los archivos del proyecto actual
- Conocimiento básico de Next.js y TypeScript

Archivos Críticos para la Importación

- Todo el directorio `/app` (aplicación Next.js)
- `package.json` y `yarn.lock`
- `.env.example` (plantilla de variables)
- `prisma/schema.prisma` (esquema de base de datos)
- Documentación del proyecto

Proceso de Importación

Paso 1: Crear Nueva Conversación DeepAgent

1. Inicia sesión en tu cuenta DeepAgent de destino
2. Crear una nueva conversación
3. Especificar que quieres **“importar un proyecto Next.js existente”**

Paso 2: Subir Archivos del Proyecto

```
# Comprimir el proyecto completo (excepto node_modules)
tar -czf sistema_erp_completo.tar.gz \
  --exclude=node_modules \
  --exclude=.next \
  --exclude=.build \
  sistema_erp_completo/
```

Paso 3: Instrucciones para DeepAgent

Comparte este mensaje con el nuevo DeepAgent:

Necesito importar un Sistema ERP completo desarrollado en Next.js 14 con TypeScript.

ESTRUCTURA DEL PROYECTO:

- Frontend: Next.js 14 + TypeScript + Tailwind CSS
- Backend: API Routes de Next.js + Prisma ORM
- Base de datos: PostgreSQL
- UI: Radix UI + shadcn/ui
- Autenticación: NextAuth.js

CARACTERÍSTICAS PRINCIPALES:

- ✓ 65+ páginas y componentes implementados
- ✓ Sistema completo de ventas y cobranza
- ✓ Gestión de clientes e inventario
- ✓ Dashboard ejecutivo con métricas
- ✓ Módulo de automatización
- ✓ Facturación electrónica (CFDI)
- ✓ Business Intelligence
- ✓ Sistema de auditoría completo

NEXT STEPS:

1. Crear la estructura base del proyecto Next.js
2. Configurar las dependencias necesarias
3. Implementar la base de datos con Prisma
4. Importar todos los componentes y páginas
5. Configurar las variables de entorno
6. Verificar que compile y funcione correctamente

¿Puedes ayudarme a configurar este proyecto?

Paso 4: Configuración de Base de Datos

Después de importar, será necesario:

1. Inicializar PostgreSQL

```
# En DeepAgent, solicitar:
"Necesito configurar PostgreSQL para este proyecto ERP"
```

1. Configurar Prisma

```
yarn prisma generate
yarn prisma db push
yarn prisma db seed
```

1. Variables de entorno mínimas

```
DATABASE_URL="postgresql://..."
NEXTAUTH_URL="http://localhost:3000"
NEXTAUTH_SECRET="generar_nuevo_secret"
```

Dependencias Principales

El proyecto utiliza estas dependencias críticas:

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

Configuración Post-Importación

1. Verificar Instalación

```
yarn install
yarn build
yarn dev
```

2. Configurar Autenticación

- Generar nuevo `NEXTAUTH_SECRET`
- Configurar proveedores de auth si es necesario

3. Configurar Integraciones (Opcional)

- OpenPay para pagos
- APIs de SMS/WhatsApp
- Servicios de facturación electrónica

4. Personalizar Configuración

- Logo de la empresa
- Colores corporativos
- Configuraciones específicas

Personalización Inmediata

Después de la importación, puedes personalizar:

Variables de Empresa

```
// En /lib/config.ts
export const EMPRESA_CONFIG = {
  nombre: "Tu Empresa",
  logo: "/logo-empresa.png",
  colorPrimario: "#3B82F6",
  colorSecundario: "#10B981"
}
```

Configuración Regional

```
// En /lib/config.ts
export const REGIONAL_CONFIG = {
  moneda: "MXN",
  timezone: "America/Mexico_City",
  iva: 16
}
```

Consideraciones Importantes

Seguridad

- **NUNCA** importar archivos `.env` con credenciales reales
- Generar nuevos secrets y API keys
- Revisar configuraciones de producción

Base de Datos







- Crear nueva instancia de PostgreSQL
- **NO** importar datos de producción sin sanitizar
- Usar datos de prueba inicialmente

APIs Externas

- Registrar nuevas cuentas en servicios externos
- Obtener nuevas API keys
- Configurar webhooks si es necesario

Testing Post-Importación

Verificaciones Básicas

1.  El proyecto compila sin errores
2.  La aplicación inicia correctamente
3.  Las páginas principales cargan
4.  La base de datos se conecta
5.  La autenticación funciona
6.  Los componentes se renderizan

Testing Funcional

```
# Ejecutar en DeepAgent
yarn test
yarn build
yarn start
```

Solución de Problemas Comunes

Error de Dependencias

```
# Limpiar e instalar
rm -rf node_modules yarn.lock
yarn install
```

Error de Base de Datos







```
# Regenerar Prisma
yarn prisma generate
yarn prisma db push
```

Error de Build

```
# Verificar TypeScript
yarn tsc --noEmit
```

Resultado Esperado

Después de seguir esta guía:

-  Sistema ERP 100% funcional
-  Todas las páginas operativas
-  Base de datos configurada
-  Autenticación activa
-  Dashboard con métricas
-  Módulos principales funcionando

Próximos Pasos

Una vez importado exitosamente:

1. **Personalizar** para tu empresa
2. **Configurar** integraciones necesarias
3. **Migrar** datos si es necesario
4. **Capacitar** usuarios finales
5. **Implementar** en producción

¡Éxito en la importación!

Si encuentras problemas, consulta `PROYECTO_STATUS.md` para obtener detalles técnicos adicionales.