

Push Exitoso - VertexERP

Fecha: 24 de Octubre, 2025

Hora: 19:02 UTC





Repositorio: <https://github.com/qhosting/vertexerp>

Resumen del Push

Commits Subidos:





515e9da - Gestión de dependencias con yarn.lock
debdeb2 - VertexERP v4.0 - Push exitoso GitHub

Archivos Actualizados:

1.  **app/package.json** - Dependencias actualizadas y corregidas
 2.  **app/yarn.lock** - Lockfile completo con 12,300+ líneas
 3.  **DEPENDENCIAS_LOCK.md** - Documentación completa de dependencias
 4.  **DEPENDENCIAS_LOCK.pdf** - Versión PDF de la documentación
-

Garantía de Versiones Exactas

El proyecto ahora incluye `yarn.lock` en el repositorio, lo que garantiza:

-  **Mismo comportamiento** en desarrollo y producción
 -  **Sin sorpresas** al deployar
 -  **Builds reproducibles** en cualquier entorno
 -  **Instalación más rápida** gracias al cache de Yarn
-



Dependencias Principales Fijadas

Categoría	Paquetes Clave	Versión
Framework	Next.js	14.2.28
Database	Prisma	6.7.0
Auth	NextAuth	4.24.11
UI	Radix UI	1.x - 2.x
Forms	React Hook Form	7.53.0
Charts	Recharts	2.15.3
Dev Tools	ESLint	8.57.0
Dev Tools	TypeScript	5.2.2



Próximos Pasos

1. Verificar en GitHub

Visita: <https://github.com/qhosting/vertexerp>

Verifica que todos los archivos estén presentes:

- ☒ app/yarn.lock (debe aparecer en el listado)
- ☒ DEPENDENCIAS_LOCK.md
- ☒ Dockerfile, docker-compose.yml, start.sh
- ☒ EASYPANEL-COMPLETE-GUIDE.md

2. Deploy en Easypanel

Ahora que el código está en GitHub, puedes deployar en Easypanel:

```
# En Easypanel, el sistema automáticamente:  
1. Clonará el repositorio  
2. Ejecutará: yarn install --immutable  
3. Generará Prisma Client  
4. Hará el build de Next.js  
5. Iniciará la aplicación
```

3. Configurar Variables de Entorno

En Easypanel, configura estas variables:

```

DATABASE_URL=postgresql://user:password@host:5432/dbname
NEXTAUTH_URL=https://tu-dominio.com
NEXTAUTH_SECRET=tu-secret-aqui
OPENPAY_API_KEY=tu-api-key
OPENPAY_MERCHANT_ID=tu-merchant-id
# ... otras variables según necesites

```

4. Ejecutar Migraciones


```

# En el primer deploy, ejecuta:
yarn prisma migrate deploy



```

Estado del Repositorio

Rama Principal:





- **Branch:** main
- **Commits totales:** 5
- **Último commit:** 515e9da
- **Estado:** Sincronizado con GitHub 

Estructura del Proyecto:




vertexerp/		
├── app/		# Aplicación Next.js
│ ├── package.json		# Dependencias (versionado)
│ ├── yarn.lock		# Lockfile (versionado) 
│ ├── prisma/		# Schema de base de datos
│ └── ...		
├── Dockerfile		# Imagen Docker optimizada
├── docker-compose.yml		# Orquestación de servicios
├── start.sh		# Script de inicialización
├── EASYPANEL-COMPLETE-GUIDE.md		# Guía de deployment
├── DEPENDENCIAS_LOCK.md		# Documentación de deps 
└── ...		

Ventajas del Nuevo Setup

Antes (Sin yarn.lock):

-  Versiones inconsistentes
-  "Works on my machine"
-  Bugs en producción
-  Builds lentos

Ahora (Con yarn.lock):

-  Versiones exactas garantizadas
-  Mismo comportamiento en todos lados
-  Instalación predecible

-  Builds más rápidos con cache

Comandos Útiles

Desarrollo Local:

```
# Instalar dependencias
cd app && yarn install --immutable

# Generar Prisma Client
yarn prisma generate

# Modo desarrollo
yarn dev

# Build de producción
yarn build
```

Verificar Integridad:

```
# Verificar que lockfile esté sincronizado
yarn install --immutable

# Si falla, actualizar lockfile
yarn install
```

Docker:

```
# Build
docker build -t vertexerp:latest .

# Run
docker-compose up -d
```

Soporte

Enlaces Útiles:

- **Repositorio:** <https://github.com/qhosting/vertexerp>
- **Documentación:** Ver archivos .md en el repositorio
- **Guía de Deploy:** EASYPANEL-COMPLETE-GUIDE.md
- **Dependencias:** DEPENDENCIAS_LOCK.md

Archivos de Referencia:

- `README.md` - Información general del proyecto
- `INSTALL.md` - Guía de instalación
- `QUICK_START.md` - Inicio rápido
- `DATABASE_SCHEMA_COMPLETE.md` - Schema completo de DB

- `CHANGELOG_v4.md` - Cambios en v4.0

🌟 Logros de Esta Sesión

1. ✅ Sistema renombrado a **VertexERP**
2. ✅ Configuración completa de Docker y Easypanel
3. ✅ **yarn.lock generado y versionado**
4. ✅ Dependencias corregidas y documentadas
5. ✅ Health check endpoint implementado
6. ✅ Service Worker corregido
7. ✅ Todo pusheado a GitHub

🎉 ¡Proyecto Listo para Producción!

Tu proyecto VertexERP está ahora completamente listo para deployment:

- ✅ Código en GitHub
- ✅ Dependencias fijadas
- ✅ Docker configurado
- ✅ Documentación completa
- ✅ Health checks implementados

Siguiente paso: Deploy en Easypanel siguiendo la guía `EASYPANEL-COMPLETE-GUIDE.md`

VertexERP v4.0.0 - Sistema ERP Completo

© 2025 - Todos los derechos reservados