💻 Comandos Útiles - Deployment y Gestión

📋 Índice

1. Git y GitHub
2. Prisma y Base de Datos
3. Vercel CLI
4. Next.js
5. Verificación y Testing
6. Troubleshooting

---

1. Git y GitHub

Inicializar y Subir a GitHub

```
# Navegar al proyecto
cd /home/ubuntu/homming_vidaro/nextjs_space

# Inicializar Git
git init

# Agregar todos los archivos
git add .

# Commit inicial
git commit -m "Preparación para deployment en Vercel"

# Cambiar a branch main
git branch -M main

# Agregar remote de GitHub
git remote add origin https://github.com/dvillagrab/inmova-platform.git

# Push inicial
git push -u origin main
```

Verificar Estado

```
# Ver estado actual
git status

# Ver archivos trackeados
git ls-files

# Ver remotes configurados
git remote -v

# Ver historial de commits
git log --oneline
```

Commits y Push

```
# Agregar archivos específicos
git add archivo.ts

# Agregar todo
git add .

# Commit
git commit -m "Mensaje del commit"

# Push a main
git push origin main

# Push forzado (usar con cuidado)
git push -f origin main
```

Branches

```
# Crear nueva branch
git checkout -b feature/nueva-funcionalidad

# Cambiar de branch
git checkout main

# Ver todas las branches
git branch -a

# Eliminar branch local
git branch -d feature/vieja

# Eliminar branch remota
git push origin --delete feature/vieja
```

Deshacer Cambios

```
# Deshacer cambios locales (no commiteados)
git checkout -- archivo.ts

# Deshacer último commit (mantener cambios)
git reset --soft HEAD~1

# Deshacer último commit (descartar cambios)
git reset --hard HEAD~1

# Remover archivo de Git pero mantenerlo local
git rm --cached archivo.ts
```

---

2. Prisma y Base de Datos

Generar Cliente Prisma

```
cd /home/ubuntu/homming_vidaro/nextjs_space

# Generar cliente
yarn prisma generate

# Generar con output específico
yarn prisma generate --schema=./prisma/schema.prisma
```

Migraciones

```
# Crear nueva migración (desarrollo)
yarn prisma migrate dev --name nombre_migracion

# Aplicar migraciones (producción)
yarn prisma migrate deploy

# Ver estado de migraciones
yarn prisma migrate status

# Resetear BD (CUIDADO: Borra datos)
yarn prisma migrate reset
```

Prisma Studio (UI)

```
# Abrir Prisma Studio
yarn prisma studio

# Se abre en http://localhost:5555
```

Database Push (Sin migraciones)

```
# Sincronizar schema directo a BD
yarn prisma db push

# Útil para prototipos, NO usar en producción
```

Seed (Poblar datos)

```
# Ejecutar seed
yarn prisma db seed

# O directamente
yarn db:seed
node scripts/seed.ts
```

Introspect (Generar schema desde BD existente)

```
# Introspect BD
yarn prisma db pull
```

Validar Schema

```
# Validar schema.prisma
yarn prisma validate

# Formatear schema.prisma
yarn prisma format
```

Conectar a BD de Producción

```
# Crear .env.production
echo "DATABASE_URL=postgresql://user:pass@host:5432/db" > .env.production

# Usar .env.production
export $(cat .env.production | xargs)
yarn prisma migrate deploy
```

---

3. Vercel CLI

Instalar Vercel CLI

```
# Instalar globalmente
npm install -g vercel

# O con yarn
yarn global add vercel
```

Login y Setup

```
# Login en Vercel
vercel login

# Link proyecto local con Vercel
cd /home/ubuntu/homming_vidaro/nextjs_space
vercel link
```

Deployment

```
# Deploy a preview
vercel

# Deploy a producción
vercel --prod

# Deploy con build log completo
vercel --prod --debug
```

Variables de Entorno

```
# Listar env vars
vercel env ls

# Agregar env var
vercel env add DATABASE_URL production

# Remover env var
vercel env rm DATABASE_URL production

# Descargar env vars
vercel env pull .env.production
```

Logs

```
# Ver logs en tiempo real
vercel logs tu-proyecto --follow

# Ver logs de un deployment específico
vercel logs [deployment-url]

# Ver logs con errores solamente
vercel logs --filter=error
```

Dominios

```
# Listar dominios
vercel domains ls

# Agregar dominio
vercel domains add inmova.app

# Remover dominio
vercel domains rm inmova.app
```

Proyectos

```
# Listar proyectos
vercel projects ls

# Ver info del proyecto
vercel project inmova-platform
```

Otros

```
# Ver información del usuario
vercel whoami

# Ver lista de deployments
vercel ls

# Rollback a deployment anterior
vercel rollback [deployment-url]

# Eliminar deployment
vercel rm [deployment-url]
```

---

4. Next.js

Desarrollo

```
cd /home/ubuntu/homming_vidaro/nextjs_space

# Iniciar dev server
yarn dev

# Iniciar en puerto específico
yarn dev -p 3001

# Iniciar con turbopack (más rápido)
yarn dev --turbo
```

Build

```
# Build para producción
yarn build

# Build con análisis de bundle
ANALYZE=true yarn build

# Build con output standalone
NEXT_OUTPUT_MODE=standalone yarn build
```

Start (Producción)

```
# Iniciar servidor de producción
yarn start

# Iniciar en puerto específico
yarn start -p 3000
```

Lint

```
# Ejecutar ESLint
yarn lint

# Lint con fix automático
yarn lint --fix
```

Info

```
# Ver info de Next.js
npx next info
```

---

5. Verificación y Testing

TypeScript

```
# Verificar tipos
yarn tsc --noEmit

# Verificar archivo específico
yarn tsc archivo.ts --noEmit
```

Tests

```
# Ejecutar tests
yarn test

# Tests con coverage
yarn test:ci

# Tests e2e
yarn test:e2e

# Tests en modo watch
yarn test --watch
```

Build y Verificación Local

```
# Build completo
yarn build

# Si build tiene éxito, iniciar
yarn start

# Abrir en navegador
open http://localhost:3000
```

Verificar Env Vars

```
# Ver variables de entorno
env | grep DATABASE

# Verificar archivo .env
cat .env | grep DATABASE_URL
```

---

6. Troubleshooting

Limpiar Caché

```
cd /home/ubuntu/homming_vidaro/nextjs_space

# Limpiar .next
rm -rf .next

# Limpiar node_modules
rm -rf node_modules
yarn install

# Limpiar yarn cache
yarn cache clean

# Limpiar todo y reinstalar
rm -rf .next node_modules yarn.lock
yarn install
```

Regenerar Prisma Client

```
# Eliminar cliente generado
rm -rf node_modules/.prisma
rm -rf node_modules/@prisma/client

# Regenerar
yarn prisma generate
```

Verificar Conexión a BD

```
# Usar psql
psql "postgresql://user:pass@host:5432/db"

# O con docker
docker run -it --rm postgres psql "postgresql://user:pass@host:5432/db"

# Verificar con node
node -e "const { PrismaClient } = require('@prisma/client'); const prisma = new Pris-
maClient(); prisma.\$connect().then(() => console.log('Conectado')).catch(e => con-
sole.error(e));"
```

Ver Logs de Errores

```
# Ver logs de desarrollo
cat .next/trace

# Ver logs de Next.js
tail -f .next/server/pages-manifest.json

# Ver errores en tiempo real
yarn dev 2>&1 | tee dev.log
```

Verificar Puerto en Uso

```
# Ver qué está usando puerto 3000
lsof -i :3000

# Matar proceso en puerto 3000
kill -9 $(lsof -t -i:3000)
```

Verificar Permisos

```
# Ver permisos de archivos
ls -la

# Cambiar permisos de scripts
chmod +x scripts/*.sh

# Cambiar owner
sudo chown -R $USER:$USER .
```

Verificar Dependencias

```
# Verificar dependencias rotas
yarn check

# Verificar versiones
yarn list --depth=0

# Actualizar dependencias
yarn upgrade-interactive
```

🚀 Scripts de Deployment

Script Completo de Pre-Deployment

```
#!/bin/bash

cd /home/ubuntu/homming_vidaro/nextjs_space

echo "🧼 Limpiando..."
rm -rf .next

echo "📦 Instalando dependencias..."
yarn install

echo "🔧 Generando Prisma client..."
yarn prisma generate

echo "🔍 Verificando TypeScript..."
yarn tsc --noEmit

echo "🏛️ Building..."
yarn build

echo "✅ Todo listo para deployment!"
```

Script de Migración en Producción

```
#!/bin/bash

cd /home/ubuntu/homming_vidaro/nextjs_space

echo "DATABASE_URL de producción:"
read -s DATABASE_URL

export DATABASE_URL

echo "📦 Generando cliente Prisma..."
yarn prisma generate

echo "🚀 Ejecutando migraciones..."
yarn prisma migrate deploy

echo "✅ Migraciones completadas!"
```

Script de Verificación Post-Deployment

```
#!/bin/bash

URL="https://tu-proyecto.vercel.app"

echo "🔍 Verificando health..."
curl -I $URL

echo ""
echo "🔍 Verificando API..."
curl $URL/api/health

echo ""
echo "🔍 Verificando login..."
curl -X POST $URL/api/auth/signin/credentials \
  -H "Content-Type: application/json" \
  -d '{"email":"admin@inmova.com","password":"admin123"}'

echo ""
echo "✅ Verificación completada!"
```

📚 Comandos de Referencia Rápida

```
# Setup inicial
./deploy-setup.sh

# Commit y push
git add . && git commit -m "Update" && git push

# Build local
yarn build && yarn start

# Migraciones
yarn prisma migrate deploy

# Deploy a Vercel
vercel --prod

# Ver logs
vercel logs --follow

# Verificar que todo funciona
curl https://tu-proyecto.vercel.app
```

🔗 Enlaces Útiles

- **Documentación:**
- Next.js: https://nextjs.org/docs
- Prisma: https://www.prisma.io/docs
- Vercel: https://vercel.com/docs
- Git: https://git-scm.com/doc

- **Troubleshooting:**

- Next.js Errors: https://nextjs.org/docs/messages
- Prisma Errors: https://www.prisma.io/docs/reference/api-reference/error-reference
- Vercel Status: https://vercel-status.com

Comandos Útiles para INMOVA Platform - Enero 2026