

Comandos Rápidos - Thunderrol

Referencia rápida de comandos para el día a día con Thunderrol.

Comandos Principales

```
# Iniciar la aplicación completa
make start
# o
./start.sh

# Ver estado de servicios
make status
# o
./status.sh

# Detener servicios
make stop

# Ejecutar tests
make test
# o
./test.sh

# Resetear base de datos
make reset-db
# o
./reset_db.sh
```

Desarrollo

Backend (FastAPI)

```
# Solo backend
make dev-backend

# Manualmente
cd backend
source venv/bin/activate
uvicorn app.main:app --reload --port 8000

# Crear migración
make create-migration DESC="agregar tabla productos"

# Aplicar migraciones
make migrate

# Poblar base de datos
make seed-db
```

Frontend (Next.js)

```
# Solo frontend
make dev-frontend

# Manualmente
cd frontend/app
yarn dev
# o
npm run dev

# Build de producción
make build-frontend

# Prisma Studio (admin DB)
make prisma-studio
```

Base de Datos

```
# Conectar a PostgreSQL
psql -h localhost -U thunderrol -d thunderrol

# Ver usuarios creados
psql -h localhost -U thunderrol -d thunderrol -c "SELECT email, role FROM users;"

# Backup de base de datos
pg_dump -h localhost -U thunderrol thunderrol > backup_$(date +%Y%m%d).sql

# Restaurar backup
psql -h localhost -U thunderrol thunderrol < backup_20240821.sql
```

Testing y Debugging

```
# Tests del backend
cd backend
source venv/bin/activate
pytest tests/ -v

# Tests específicos
pytest tests/test_auth.py -v
pytest tests/test_units.py::test_create_unit -v

# Linting
make lint

# Formatear código
make format

# Ver logs en tiempo real
tail -f backend/logs/app.log # si existen archivos de log
```

Instalación y Mantenimiento

```
# Instalar todas las dependencias
make install-deps

# Solo backend
make install-backend

# Solo frontend
make install-frontend

# Limpiar caches
make clean

# Limpieza completa (elimina node_modules, venv)
make clean-all

# Reconstruir todo
make rebuild
```

Debugging

Ver procesos corriendo

```
# Ver procesos de Thunderrol
ps aux | grep -E "(uvicorn|next-server)"

# Ver puertos ocupados
netstat -tlnp | grep -E "(3000|8000|5432)"
lsof -i :3000
lsof -i :8000
```

Logs importantes

```
# Logs de PostgreSQL (Linux)
sudo tail -f /var/log/postgresql/postgresql-*.log

# Ver logs de servicios systemd
sudo journalctl -u postgresql -f
```

Solución de problemas comunes

```
# PostgreSQL no conecta
sudo systemctl restart postgresql
sudo systemctl status postgresql

# Puerto ocupado
sudo lsof -t -i:8000 | xargs kill -9 # Backend
sudo lsof -t -i:3000 | xargs kill -9 # Frontend

# Permisos de archivos
sudo chown -R $USER:$USER /home/ubuntu/thunderrol/

# Resetear migraciones completamente
cd backend
source venv/bin/activate
alembic downgrade base
alembic upgrade head
```

URLs y Accesos

- **Frontend:** `http://localhost:3000`
- **Backend:** `http://localhost:8000`
- **API Docs:** `http://localhost:8000/docs`
- **Redoc:** `http://localhost:8000/redoc`
- **Prisma Studio:** `make prisma-studio` (puerto dinámico)

Credenciales de Desarrollo

```
admin@thunderrol.com / admin123 (ADMIN)
inventario@thunderrol.com / inv123 (MANAGER)
taller@thunderrol.com / taller123 (OPERATOR)
ventas@thunderrol.com / ventas123 (VIEWER)
```

Estructura de Archivos Importantes

```
thunderrol/
├── start.sh           # Iniciar todo
├── status.sh          # Ver estado
├── test.sh            # Ejecutar tests
├── reset_db.sh        # Resetear DB
├── Makefile           # Comandos make
├── GUIA_INSTALACION_MANUAL.md # Guía completa
├──
├── backend/
│   ├── .env           # Variables de entorno
│   ├── requirements.txt # Dependencias Python
│   ├── alembic.ini     # Config migraciones
│   └── app/main.py     # Punto de entrada
├──
└── frontend/app/
    ├── .env.local      # Variables de entorno
    ├── package.json    # Dependencias Node
    └── prisma/schema.prisma # Schema DB
```

Variables de Entorno Importantes

Backend (.env)

```
DATABASE_URL=postgresql://thunderrol:thunderrol123@localhost:5432/thunderrol
SECRET_KEY=your-secret-key-here-thunderrol-2024
ALGORITHM=HS256
ACCESS_TOKEN_EXPIRE_MINUTES=30
ENVIRONMENT=development
```

Frontend (.env.local)

```
NEXTAUTH_URL=http://localhost:3000
NEXTAUTH_SECRET=super-secret-nextauth-key-change-in-production
NEXT_PUBLIC_API_URL=http://localhost:8000
DATABASE_URL=postgresql://thunderrol:thunderrol123@localhost:5432/thunderrol
```

Flujo de Desarrollo Típico

1. Iniciar desarrollo:

```
bash
make status # Verificar estado
make start  # Iniciar servicios
```

2. Hacer cambios en el código:

- Backend: Los cambios se auto-recargan (-reload)
- Frontend: Los cambios se auto-recargan (hot reload)

3. Probar cambios:

```
bash
make test  # Ejecutar tests
make lint  # Verificar código
```

4. Cambios en DB:

```
bash
make create-migration DESC="descripción"
make migrate
```

5. Terminar sesión:

```
bash
make stop    # Detener servicios
```

 **Tip:** Usa `make help` para ver todos los comandos disponibles en cualquier momento.