

Instrucciones de Configuración - Sistema de Testing INMOVA

Guía Paso a Paso para el Equipo de Desarrollo

Índice

1. [Requisitos Previos](#)
 2. [Instalación de Dependencias](#)
 3. [Configuración de Base de Datos](#)
 4. [Configuración de Scripts](#)
 5. [Configuración de CI/CD](#)
 6. [Validación del Setup](#)
 7. [Troubleshooting](#)
-

Requisitos Previos

Software Necesario

- [x] **Node.js** 22.x o superior
- [x] **Yarn** 1.22+ (ya instalado)
- [x] **PostgreSQL** 15+ (para tests de integración)
- [] **Java** 17+ (para JMeter, solo si se ejecuta localmente)
- [] **Docker** (opcional, para OWASP ZAP local)

Verificar Versiones

```
node --version    # Debería mostrar v22.x.x
yarn --version    # Debería mostrar 1.22.x
psql --version    # Debería mostrar 15.x
java --version    # Debería mostrar 17.x (opcional)
```

Instalación de Dependencias

Paso 1: Instalar Dependencias de Proyecto

```
cd /home/ubuntu/homming_vidaro/nextjs_space
yarn install
```

Paso 2: Instalar Dependencias de Testing (si no están)

```
# Jest (ya instalado)
yarn add -D jest @types/jest ts-jest jest-environment-jsdom

# Playwright (ya instalado)
yarn add -D @playwright/test
npx playwright install --with-deps chromium

# Pally (accesibilidad)
yarn add -D pally

# node-fetch (para jest.setup.js)
yarn add -D node-fetch@2

# wait-on (para esperar servidor)
yarn add -D wait-on
```

Paso 3: Verificar Instalación

```
# Verificar que Jest funciona
yarn jest --version

# Verificar que Playwright funciona
npx playwright --version

# Verificar Pally
node -e "require('pally'); console.log('Pally OK');"
```

Configuración de Base de Datos

Paso 1: Crear Base de Datos de Pruebas

```
# Conectar a PostgreSQL
psql -U postgres

# Crear base de datos de pruebas
CREATE DATABASE innova_test;
CREATE USER test WITH PASSWORD 'test';
GRANT ALL PRIVILEGES ON DATABASE innova_test TO test;
\q
```

Paso 2: Configurar Variables de Entorno

Crear archivo `.env.test` en `nextjs_space/` :

```
cat > .env.test << 'EOF'
# Database
DATABASE_URL="postgresql://test:test@localhost:5432/inmova_test"

# NextAuth
NEXTAUTH_SECRET="test-secret-key-for-testing-only"
NEXTAUTH_URL="http://localhost:3000"

# Node Environment
NODE_ENV="test"
EOF
```

Paso 3: Ejecutar Migraciones

```
# Aplicar schema a la BD de pruebas
DATABASE_URL="postgresql://test:test@localhost:5432/inmova_test" npx prisma migrate de
ploy

# Generar cliente Prisma
npx prisma generate

# Seed (opcional)
DATABASE_URL="postgresql://test:test@localhost:5432/inmova_test" npx prisma db seed
```



Configuración de Scripts

Paso 1: Actualizar package.json

Añadir estos scripts a la sección "scripts" :

```
{
  "scripts": {
    // ... scripts existentes ...

    // NUEVOS SCRIPTS DE TESTING
    "test:integration": "jest --ci --testPathPatterns=__tests__/integration --maxWork-
ers=2",
    "test:visual": "playwright test --grep @visual",
    "test:ally": "node scripts/run-pally.js",
    "test:security": "echo 'Security tests run in CI/CD with OWASP ZAP'",
    "test:load": "jmeter -n -t jmeter/load-test.jmx -l results.jtl -e -o jmeter-re-
port/",
    "test:all": "yarn test:ci && yarn test:integration && yarn test:e2e",
    "coverage:view": "open coverage/lcov-report/index.html"
  }
}
```

Paso 2: Verificar Scripts

```
# Verificar que los scripts están disponibles
yarn run | grep test:
```

Debería mostrar:

```
- test:ci
- test:integration
- test:e2e
- test:visual
- test:ally
- test:load
- test:all
```



Configuración de CI/CD

Paso 1: Verificar Workflow

El workflow ya está creado en:

```
.github/workflows/quality-assurance.yml
```

Paso 2: Configurar Secrets en GitHub (si es necesario)

1. Ir a **Settings > Secrets and variables > Actions**
2. Añadir estos secrets (si aplica):
 - DATABASE_URL (opcional, usa PostgreSQL service)
 - Otros secrets específicos del proyecto

Paso 3: Activar Workflow

```
# Hacer commit de los cambios
git add .
git commit -m "feat: Add comprehensive testing system"
git push origin main

# El workflow se ejecutará automáticamente
```

Paso 4: Ver Ejecución

1. Ir a **Actions** tab en GitHub
2. Ver workflow **Quality Assurance - INMOVA**
3. Ver logs y reportes



Validación del Setup

Test 1: Tests Unitarios

```
cd /home/ubuntu/homming_vidaro/nextjs_space
yarn test:ci
```

Resultado Esperado:

```
Test Suites: 3 passed, 3 total
Tests:      78 passed, 2 failed (expected), 80 total
Coverage:   85%+ lines, 90%+ functions
```

Test 2: Tests de Integración

```
# Asegurarse de que PostgreSQL está corriendo
psql -U test -d inmovi_test -c "SELECT 1;"

# Ejecutar tests
DATABASE_URL="postgresql://test:test@localhost:5432/inmovi_test" yarn test:integration
```

Resultado Esperado:

```
Test Suites: 2 passed, 2 total
Tests:      15 passed, 15 total
```

Test 3: Tests E2E

```
# Iniciar servidor de desarrollo
yarn dev &

# Esperar a que el servidor esté listo
npx wait-on http://localhost:3000 --timeout 60000

# Ejecutar tests E2E
yarn test:e2e

# Detener servidor
kill %1
```

Resultado Esperado:

```
Passing: 10/10
Failing: 0/10
Duration: ~2 minutes
```

Test 4: Tests de Accesibilidad

```
# Iniciar servidor
yarn dev &
npx wait-on http://localhost:3000 --timeout 60000

# Ejecutar Pally
yarn test:ally

# Ver reporte
open pally-report/accessibility-report.html

# Detener servidor
kill %1
```

Resultado Esperado:

✓ Tests de accesibilidad PASADOS exitosamente.
 Total de errores críticos: 0
 Total de advertencias: <10

Test 5: Visual Regression

```
# Iniciar servidor
yarn dev &
npx wait-on http://localhost:3000 --timeout 60000

# Primera ejecución (crear baseline)
yarn test:visual --update-snapshots

# Segunda ejecución (verificar)
yarn test:visual

# Detener servidor
kill %1
```

Resultado Esperado:

All snapshots match!



Troubleshooting

Problema 1: “Cannot find module ‘@/lib/db’”

Causa: Path mapping no configurado

Solución:

```
// En tsconfig.json
{
  "compilerOptions": {
    "paths": {
      "@/*": ["./*"]
    }
  }
}
```

Problema 2: “Port 3000 already in use”

Causa: Servidor ya corriendo

Solución:

```
# Matar proceso en puerto 3000
lsof -ti:3000 | xargs kill -9

# O usar otro puerto
PORT=3001 yarn dev
```

Problema 3: “Database connection failed”

Causa: PostgreSQL no está corriendo o credenciales incorrectas

Solución:

```
# Verificar que PostgreSQL está corriendo
sudo systemctl status postgresql

# Si no está corriendo
sudo systemctl start postgresql

# Verificar conexión
psql -U test -d innova_test -c "SELECT 1;"
```

Problema 4: “Playwright browsers not installed”

Causa: Navegadores de Playwright no instalados

Solución:

```
npx playwright install --with-deps chromium
```

Problema 5: “Pally timeout”

Causa: Servidor no está corriendo o muy lento

Solución:

```
# Verificar servidor
curl http://localhost:3000

# Aumentar timeout en scripts/run-pally.js
// timeout: 120000 (2 minutos)
```

Guía Rápida para Nuevos Desarrolladores

Setup Inicial (5 minutos)

```
# 1. Clonar repo (si aún no lo tienes)
git clone <repo-url>
cd homming_vidaro/nextjs_space

# 2. Instalar dependencias
yarn install

# 3. Configurar BD de pruebas
psql -U postgres -c "CREATE DATABASE inmovi_test;"
psql -U postgres -c "CREATE USER test WITH PASSWORD 'test';"
psql -U postgres -c "GRANT ALL PRIVILEGES ON DATABASE inmovi_test TO test;"

# 4. Aplicar migraciones
DATABASE_URL="postgresql://test:test@localhost:5432/inmovi_test" npx prisma migrate de
ploy

# 5. Instalar navegadores Playwright
npx playwright install --with-deps chromium

# 6. Ejecutar tests
yarn test:ci
```

Workflow Diario

```
# Antes de hacer commit
yarn test:ci      # Tests unitarios rápidos
yarn lint         # Lint
yarn build        # Build (si es necesario)

# Antes de hacer PR
yarn test:all     # Todos los tests
yarn test:allly   # Accesibilidad
```



Checklist de Validación

- [] Node.js 22+ instalado
- [] Yarn instalado
- [] PostgreSQL configurado
- [] Base de datos inmovi_test creada
- [] Dependencias instaladas (yarn install)
- [] Playwright navegadores instalados
- [] Tests unitarios pasan (yarn test:ci)
- [] Tests de integración pasan (yarn test:integration)
- [] Tests E2E pasan (yarn test:e2e)
- [] Tests de accesibilidad pasan (yarn test:allly)
- [] Workflow de CI/CD configurado en GitHub
- [] Primer workflow ejecutado exitosamente

Soporte

Documentación:

- [TESTING_COMPLETE_GUIDE.md](#) (./TESTING_COMPLETE_GUIDE.md) - Guía completa
- [TESTING_REPORT.md](#) (./TESTING_REPORT.md) - Reporte detallado
- [tests/README.md](#) (./__tests__/README.md) - Guía rápida

Contacto:

- Slack: `#inmova-qa`
 - Email: `qa@inmova.app`
-

ÚLTIMA ACTUALIZACIÓN: Diciembre 2024

VERSIÓN: 2.0