

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
CENTRO UNIVERSITARIO DE OCCIDENTE
DIVISIÓN DE CIENCIAS DE LA INGENIERÍA
MANEJO E IMPLEMENTACIÓN DE ARCHIVOS



MANUAL DE TECNICO

DIEGO ALESSANDRO GONZALEZ BRAVO / 202230393

1. Descripción general

El proyecto se compone de:

- Frontend: Aplicación Angular, desplegada en Netlify.
- Backend: API REST desarrollada con Spring Boot (Java 21), ejecutada localmente y expuesta mediante Ngrok.
- Base de datos: PostgreSQL ejecutándose localmente.

Este entorno permite probar el sistema completo sin necesidad de un servidor remoto dedicado.

2. Requisitos previos

Software necesario

Componente	Versión mínima	Descripción
------------	----------------	-------------

- | | | |
|---------------|-----|--|
| • Java | 21 | Para ejecutar el backend Spring Boot |
| • Node.js | 20+ | Para ejecutar Angular y sus dependencias |
| • Angular CLI | 17+ | Compilación y despliegue del frontend |
| • Maven 3.9+ | | Compilación del backend |
| • PostgreSQL | 15+ | Base de datos del sistema |
| • Ngrok | 3.x | Túnel HTTPS para exponer el backend local |
| • Git | — | Control de versiones y despliegue en Netlify |

3. Despliegue del Backend con Ngrok

3.1. Compilar y ejecutar el backend

```
cd backend
```

```
mvn clean package
```

```
java -jar target/backend-0.0.1-SNAPSHOT.jar
```

Por defecto, el backend corre en: <http://localhost:8080>

3.2. Configurar Ngrok

1. Inicia sesión en Ngrok

```
ngrok config add-authtoken TU_TOKEN_AQUI
```

2. Exponer el backend local

```
ngrok http --domain=tu-dominio.ngrok-free.dev 8080
```

Ejemplo:

```
ngrok http --domain=luana-counsellable-strictly.ngrok-free.dev 8080
```

Verificar la conexión

Ngrok mostrará una URL HTTPS del tipo:

`https://luana-counsellable-strictly.ngrok-free.dev -> http://localhost:8080`

Este será el dominio que el frontend usará como apiUrl.

3.3. Configuración CORS (Backend)

Asegúrate de tener en tu clase de configuración de seguridad o @Bean:

@Bean

```
public CorsConfigurationSource corsConfigurationSource() {
    CorsConfiguration cfg = new CorsConfiguration();
    cfg.setAllowedOrigins(List.of(
        "http://localhost:4200",
        "https://proyecto2-arch.netlify.app",
        "https://*.ngrok-free.dev"
    ));
    cfg.setAllowedMethods(List.of("GET", "POST", "PUT", "DELETE", "PATCH",
"OPTIONS"));
    cfg.setAllowedHeaders(List.of("*"));
    cfg.setAllowCredentials(true);
    UrlBasedCorsConfigurationSource source = new UrlBasedCorsConfigurationSource();
    source.registerCorsConfiguration("/**", cfg);
    return source;
}
```

4. Despliegue del Frontend en Netlify

4.1. Construir la aplicación Angular

```
cd frontend
ng build --configuration=production
```

Esto genera una carpeta /dist lista para desplegar.

4.2. Subir a Netlify

Opción 1: Desde la interfaz web

Ir a <https://app.netlify.com>

Crear nuevo sitio → Deploy manually

Arrastrar el contenido de /dist/<nombre-proyecto>

Netlify generará una URL temporal:

`https://proyecto2-arch.netlify.app`

Opción 2: Desde GitHub

Conectar tu repositorio.

Configurar build command:

`ng build --configuration=production`

Public directory:

`dist/<nombre-proyecto>`

4.3. Configurar el archivo `environment.ts`

```
export const environment = {  
  production: true,  
  apiUrl: 'https://luana-counsellable-strictly.ngrok-free.dev/api'  
};
```

Y opcionalmente el `proxy.conf.json` para desarrollo local:

```
{  
  "/api": {  
    "target": "http://localhost:8080",  
    "secure": false,  
    "changeOrigin": true  
  }  
}
```

4.4. Verificación final

Abre tu sitio en Netlify.

Inspecciona las peticiones en la consola del navegador:

Todas deben apuntar a `https://<tu-ngrok>.ngrok-free.dev/api/...`

Y recibir respuestas JSON, no HTML (ERR_NGROK_6024 indica cabeceras incorrectas o dominio no permitido).

5. Principales Clases Utilizadas Backend (Spring)

Clase	Descripción
<code>Usuario</code>	Entidad principal que representa los usuarios del sistema.
<code>Producto</code>	Entidad con la información de los productos publicados.
<code>Pedido</code> <code>DetallePedido</code>	/ Control de las compras y sus detalles.
<code>Notificacion</code>	Manejo de notificaciones automáticas (correo y sistema).
<code>AuthController</code>	Controlador REST para autenticación y login.
<code>JwtUtil</code>	Genera y valida tokens JWT.
<code>NotificacionService</code>	Envía correos y registra notificaciones en BD.
<code>ReportesController</code>	Devuelve reportes de top clientes, vendedores, productos, etc.

Frontend (Angular)

Archivo	Función
<code>auth.interceptor.ts</code>	Inyecta el token JWT y omite el aviso de Ngrok.
<code>auth.service.ts</code>	Maneja la sesión, login y almacenamiento del token.
<code>environment.ts</code>	Configura la URL base de la API (local o producción).
<code>reportes.service.ts</code>	Realiza las peticiones a los endpoints <code>/api/reportes</code> .
<code>login.component.ts</code>	Controla la autenticación y redirección según rol.
<code>app.config.ts</code>	Configura rutas y proveedores globales.

Errores comunes

Error	Causa	Solución
ERR_NGROK_6024	Ngrok intercepta el tráfico sin el header de bypass	Agregar <code>'ngrok-skip-browser-warning': 'true'</code> en el interceptor.
401 Unauthorized	Token inválido o vencido	Revisar expiración del JWT y limpiar <code>localStorage</code> .
403 Forbidden (CORS)	Dominio no autorizado	Agregar dominio Ngrok/Netlify en CORS backend.
No carga datos	URL <code>apiUrl</code> mal configurada o proxy incorrecto	Revisar <code>environment.ts</code> y logs del navegador.