# Migración de Autenticación con Usuario y Contraseña a OAuth2 en Gmail SMTP

## 1. Introducción

En la implementación original del envío de correos en nuestra aplicación, se utilizaba la autenticación basada en usuario y contraseña para conectarse al servidor SMTP de Gmail. Sin embargo, este método presenta serias limitaciones y problemas de seguridad. Por esta razón, se ha realizado una migración al protocolo OAuth2, que ofrece una autenticación más segura y estandarizada.

# 2. Problemas con la autenticación tradicional (Usuario y Contraseña)

Antes de la migración, la aplicación utilizaba credenciales estáticas almacenadas en el código o en archivos de configuración para autenticarse con Gmail. Esto presentaba los siguientes inconvenientes:

* **Menor seguridad**: Las credenciales estáticas son vulnerables a filtraciones si no se manejan correctamente.
* **Restricciones de Google**: Gmail ya no permite el acceso a aplicaciones que usen contraseñas sin autenticación moderna.
* **Problemas de escalabilidad**: No es fácil cambiar contraseñas o gestionar permisos sin afectar toda la aplicación.
* **Dificultad para auditorías**: No hay un control granular sobre quién accede al correo.

## 3. Beneficios de la Autenticación con OAuth2

OAuth2 soluciona los problemas anteriores y ofrece numerosas ventajas:

* **Mayor seguridad**: No se almacenan ni transmiten contraseñas, solo tokens de acceso temporales.
* **Cumplimiento con las políticas de Google**: OAuth2 es el método recomendado por Google para acceder a sus servicios de manera programática.
* **Gestín de permisos**: Se pueden revocar tokens sin afectar la cuenta principal.
* **Automatización**: Se pueden generar nuevos tokens de acceso automáticamente sin interacción manual.

## 4. Proceso de Migración a OAuth2

Para realizar la migración, se implementaron los siguientes cambios:

1. **Registro de la aplicación en Google Cloud Console**: Se crearon credenciales de OAuth2.
2. **Generación de un Refresh Token**: Se configuró la aplicación para obtener y renovar tokens de acceso dinámicamente.
3. **Implementación del flujo OAuth2**:
   * Se reemplazó el uso de usuario y contraseña por autenticación con tokens de acceso.
   * Se usó XOAUTH2 como mecanismo de autenticación en el servidor SMTP.
4. **Automatización de la renovación de tokens**: Se desarrolló un servicio en Spring Boot para actualizar el access\_token antes de que expire.

## 5. Impacto y Resultados

Tras la migración a OAuth2, la aplicación presenta mejoras significativas en:

* **Seguridad**: Se eliminó la exposición de contraseñas.
* **Fiabilidad**: La autenticación ahora está alineada con las mejores prácticas de Google.
* **Escalabilidad**: Es más fácil gestionar permisos y accesos sin afectar la infraestructura existente.

**6. Conclusión**

El cambio de autenticación con usuario y contraseña a OAuth2 no solo mejora la seguridad, sino que también garantiza el cumplimiento con las políticas de Google y facilita la administración de accesos. Con este cambio, la aplicación está mejor preparada para el futuro y alineada con las mejores prácticas de seguridad en la industria.