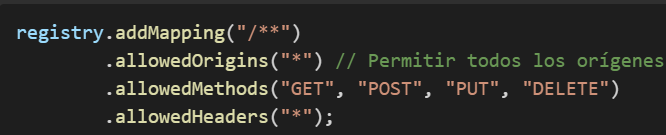
# **Problema de Seguridad con allowOrigins("\*") en CORS y su Solución**

## Introducción

El uso de allowedOrigins("\*") en la configuración de CORS dentro de una aplicación Spring Boot puede generar problemas de seguridad, especialmente cuando se manejan datos sensibles o autenticación basada en tokens. Este documento explica los riesgos asociados y propone una solución segura para entornos de desarrollo y producción.

## **Problema con allowOrigins("\*")**

Cuando se configura CORS de la siguiente manera en Spring Boot:



### Riesgos de seguridad:

* **Exposición a ataques CSRF y XSS**: Cualquier sitio web puede enviar solicitudes a la API sin restricciones.
* **Fuga de datos**: Un atacante puede hacer que el navegador de un usuario envíe peticiones sin su consentimiento.
* **Problemas con autenticación**: Si la API usa autenticación basada en cookies o tokens, otros sitios podrían hacer uso indebido de esas credenciales.
* **No permite allowCredentials(true)**: Si allowedOrigins("\*") está presente, |Spring Boot no permite el uso de allowCredentials(true), lo que impide la autenticación segura.

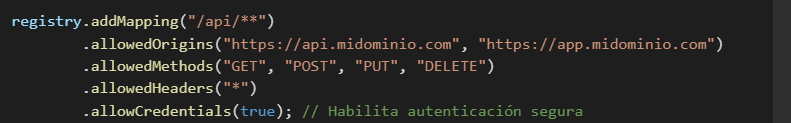
## **Solución Segura para Entornos de Desarrollo**

Si el frontend Angular se ejecuta en http://localhost:4200 y el backend en http://localhost, se debe permitir explícitamente estos orígenes:



## **Solución Segura para Producción**

Cuando la API está en un dominio público, la configuración de CORS debe ser aún más estricta:



## **Conclusión**

Permitir allowedOrigins("\*") en una aplicación Spring Boot representa un gran riesgo de seguridad. La mejor práctica es definir explícitamente los orígenes permitidos, tanto en entornos de desarrollo como en producción. De esta forma, se protege la API contra ataques y se asegura un acceso controlado a los recursos.