

# AdaGestor | Para desarrolladores

En esta sección se describe toda la información acerca de los repositorios, compilación, despliegue y dependencias de servicios del proyecto.

## Repositorios:

Los repositorios del proyecto se encuentran en el GitLab en esta ruta: [AdaGestor en GitLab](#)

- El proyecto del front end (ng-adagestor) esta desarrollado con Angular 18.2.0, y se encuentra alojado en este repositorio: [Proyecto Front](#)
- El proyecto del back end (spring-adagestor) esta desarrollado con Spring Boot 3.3.4, y se encuentra alojado en este repositorio: [Proyecto Back](#)

## Compilación

Para la compilación inicial de los proyectos se utilizo NodeJS v22.4.1 y Java 21.0.4 (Coretto).

- Para compilar el proyecto de Angular se debe ejecutar el comando “ng build”, que por defecto realiza una compilación en modo para producción. Este modo utiliza el archivo **environment.prod.ts** de variables de entorno. Una vez finalice el build, se puede encontrar el artefacto en la ruta “**dist/ng-adagestor/browser/**”.
- Para compilar el proyecto de Spring se deben ejecutar las tareas de gradle “**clean build**”, para limpiar el directorio del proyecto y construir el artefacto **jar**. Una vez finalice el build, el artefacto puede encontrarse bajo el directorio “**build/libs/spring-adagestor-0.0.1-SNAPSHOT.jar**”.

## Despliegue

A continuación se describen los pasos para realizar un despliegue del aplicativo.

1. Asegurarse de tener actualizados los proyectos del front-end y back-end desde el repositorio de git.
2. Compilar los artefactos de cada proyecto.
3. Construir las respectivas imágenes de Docker con los **Dockerfile** de cada proyecto, ejecutando en cada uno “**docker build -t fabricaada/<nombre-imagen>:<version> .**”. Luego subir las imágenes al repositorio de docker hub de fabrica con “**docker push fabricaada/<nombre-imagen>:<versión>**”.
4. Conectarse al servidor que contenga el ambiente de Docker, y copiar a este el archivo “**docker-compose.yml**” que se encuentra en el proyecto del back.
5. En el archivo compose se encuentran definidas las **imágenes**, los **parámetros** y las **variables de entorno** para cada servicio. Se deben actualizar las imágenes de cada servicio a las versiones recién subidas al docker hub, y asegurarse de que los demás parámetros estén actualizados.
6. Ejecutar “**docker compose up**” para que se haga el despliegue de las imágenes configuradas.
7. Validar el estado de los contenedores creados, para asegurarse que no haya habido errores al

iniciarse.

## Dependencias

El proyecto depende de los siguientes servicios internos:

### SicofConfig

Este servicio brinda la funcionalidad de login, registro y modificación de usuarios. Para poder desarrollar AdaGestor en el tiempo estimado inicialmente, fue necesario hacer un fork de SicofConfig para agregar y modificar algunas funcionalidades faltantes requeridas por el proyecto. Este fork se llama **"sicoconfig-ag"**, y también se encuentra alojado en el repositorio de [GitLab de Adagestor](#). Respecto a SicofConfig, se hicieron modificaciones en UserController, UserService y ConsultarTerceroExistenteService.

Ademas, Sicoconfig depende del **servicio de terceros** del ecosistema para hacer las consultas de terceros, lo cual impone unos requisitos adicionales:

Deben estar parametrizados algunos registros en ciertas tablas del esquema SICOCONFIG en la base de datos a la que este conectado sicoconfig-ag (inicialmente "conexionmul"):

- APPLICATION: Debe haber un registro que identifique a AdaGestor, con un **APP\_ID** único.
- CONNECT\_DB: Debe haber un registro para AdaGestor con un CONNECT\_CODE único, en el que se parametrize la **conexion** a la base de datos de la que se van a consultar los **terceros**, se defina un contexto en **CONNECT\_CONTEXT**, y se relacione el **COMPANY\_ID** de la empresa.
- SUBSIDIARY: Debe haber un registro para AdaGestor que relacione el **COMPANY\_ID** de la empresa.
- APPLICATION\_COMPANY: Debe haber un registro que relacione el **APP\_ID**, **SUBSIDIARY\_ID** y **CONNECT\_DB\_ID** de los registros anteriores.

### AdaEmailService

Este servicio brinda la funcionalidad para enviar las notificaciones por correo electrónico.

- El servicio se consume por medio de peticiones HTTP **POST**.
- El cuerpo de la petición debe contener las propiedades: **subject**, **from**, **toEmail**, y **content**, que son los parámetros del correo.
- Además se debe incluir en el header **"token"** un token valido para el servicio, el cual se encuentra parametrizado en las propiedades de spring-adagestor.

From:  
<http://wiki.adacsc.co/> - Wiki

Permanent link:  
<http://wiki.adacsc.co/doku.php?id=ada:howto:sicoferp:factory:internos:adagestor:fordevs&rev=1734106764>

Last update: 2024/12/13 16:19

