

1. Instalación

1.1. Requerimientos de software

Para la instalación local se requiere tener instalada previamente la versión más actualizada de [npm](#) (Node Package Manager). Además se necesita [Docker](#) para poder levantar una base de datos local.

1.2. Proceso de instalación (local)

Para ejecutar localmente al servidor se deben instalar todas las dependencias que se encuentran definidas en los archivos de package.json (uno en el *root* del proyecto y otro en *root/appclient-src*, que corresponden a las dependencias del backoffice y del frontoffice respectivamente). Para esto se debe ejecutar el comando `npm install` desde la terminal estando en el directorio *root* y luego en *root/appclient-src*.

2. Configuración

El servidor se crea con un usuario de negocios de tipo admin con las siguientes credenciales:

- Usuario: superusercito
- Contraseña: pass

A su vez, se crea también un servidor con los siguientes datos para autenticación:

- Nombre: superservercito
- Token: superservercito-token

Este token nunca cambia al actualizarse.

3. Forma de uso (local)

Para levantar al servidor localmente primero se debe ejecutar el comando `bash docker-start.sh` estando en el directorio *root*.

```
2017-12-19 16:32:10.645 UTC [1] LOG: listening on IPv4 address "0.0.0.0", port 5432
2017-12-19 16:32:10.645 UTC [1] LOG: listening on IPv6 address ":::", port 5432
2017-12-19 16:32:10.648 UTC [1] LOG: listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
2017-12-19 16:32:10.657 UTC [70] LOG: database system was shut down at 2017-12-19 16:32:10 UTC
2017-12-19 16:32:10.661 UTC [1] LOG: database system is ready to accept connections
```

Figura 1: Mensaje de inicio correcto de docker-start

La dirección dónde se encontrará corriendo el servidor luego de esto es: [localhost:5000](#).

Cuando finalice la ejecución del mismo, sin cerrar esa terminal, se debe abrir una segunda en la misma locación y ejecutar `npm run-script start-local`. Para finalizar la ejecución se debe primero ejecutar `ctrl+c` en ambas terminales y el comando `bash docker-stop.sh` estando en el directorio root.

```
AC2017-12-19 16:35:08.607 UTC [1] LOG:  received fast shutdown request
2017-12-19 16:35:08.610 UTC [1] LOG:  aborting any active transactions
2017-12-19 16:35:08.612 UTC [1] LOG:  worker process: logical replication launcher (PID 76) exited with exit cod
e 1
2017-12-19 16:35:08.613 UTC [71] LOG:  shutting down
2017-12-19 16:35:08.623 UTC [1] LOG:  database system is shut down
aczarnitzki-mbp:SharedServer aczarnitzki$ bash docker-stop.sh
postgres-database
postgres-database
```

Figura 2: Mensaje de finalización correcta de docker-stop

3.1. Logs y modo debug

Automáticamente el servidor genera tres archivos de tipo log, *info.log*, *warn.log* y *error.log*, en la carpeta root. Los mismos contienen la información de:

Nivel	Condiciones
Error	Condición de falla catastrófica, el sistema no puede funcionar Condición que haga que la aplicación no pueda ejecutar una funcionalidad.
Warn	Cualquier condición anómala que afecte el funcionamiento del sistema, pero no impida la funcionalidad básica.
Info	Cualquier acción correspondiente a un caso de uso iniciada por el usuario o el sistema. Información que permita trazar el historial de las entidades.

Si se quiere levantar al mismo en modo debug se puede ejecutar el comando `npm run-script debug` luego de levantar la base de datos localmente como se indica en la sección 3.

```
> sharedserver@1.0.0 debug /Users/aczarnitzki/Documents/SharedServer
> DEBUG=ServerApi env URI='postgres://postgres:password@localhost:6000/postgres' node ./src/index.js

ServerApi Iniciando 'Shared server' +0ms
ServerApi Escuchando puerto 5000 +209ms
```

Figura 3: Modo debug

3.2. Documentación

Para generar la documentación del servidor se puede ejecutar el comando `npm run-script create-doc` estando en el directorio root.