

## Manual- Frontend

---

**Nombre:** Brandon David Quintero

**Código:** 230212019

---

### Desarrollo Frontend con Angular en IntelliJ IDEA

#### 1. Instalación de Angular CLI

- Asegurarse de tener Node.js instalado.
- Abrir la terminal y ejecutar el siguiente comando para instalar Angular CLI globalmente:
- `npm install -g @angular/cli`

#### 2. Creación de un nuevo proyecto Angular

- Abrir IntelliJ IDEA y seleccionar "File" > "New" > "Project...".
- Elegir "Angular CLI" en la lista de tecnologías y seguir los pasos del asistente para crear un nuevo proyecto Angular.

#### 3. Desarrollo de Microservicios

- Cada microservicio debe tener su propio directorio en la estructura del proyecto.
- Para cada microservicio, se debe crear un nuevo módulo y componentes según sea necesario.

#### 4. Integración con Jira para la Gestión de Tareas

- Utilizar Jira para crear un proyecto y definir las historias y tareas relacionadas con cada microservicio.

- Asignar las tareas a los miembros del equipo y establecer los tiempos estimados y reales.

## 5. Inicio de Sprint en Jira

- Crear un sprint en Jira para la implementación de la funcionalidad del CRU y el login.
- Asignar las tareas a realizar durante el sprint a los miembros del equipo.

## Desarrollo con GitHub

### 6. Creación de Repositorio en GitHub

- Crear un repositorio en GitHub para el proyecto.
- Configurar el repositorio con una estructura de carpetas que refleje la arquitectura de microservicios.
- Antes de comenzar a trabajar, clonar el repositorio a la máquina local con el siguiente comando:  
  
`git clone <URL_del_repositorio>`

### 7. Uso de Ramas y Features

- Cada microservicio debe tener su propia rama (feature branch).
- Los desarrolladores deben trabajar en sus ramas respectivas y fusionarlas con la rama principal (por ejemplo, **main** o **develop**) cuando la funcionalidad esté completa.

### 8. Trabajo con Git (Fetch, Pull, Commit, Push)

- Antes de comenzar a trabajar en una nueva tarea, realizar git pull para obtener las últimas actualizaciones del repositorio.

- Realizar commits frecuentes y utilizar git push para subir los cambios al repositorio remoto.

## 9. Uso de Bootstrap y SweetAlert2 en Angular

- Instalar Bootstrap utilizando el siguiente comando en la terminal:

`npm install Bootstrap`

- Importar Bootstrap en los archivos CSS del proyecto según sea necesario.
- Para SweetAlert2, instalarla biblioteca y el paquete para Angular con los siguientes comandos:

`npm install sweetalert2 @sweetalert2/ngx-sweetalert2`

Importar SweetAlert2 en los componentes y servicios donde sea necesario.

## Despliegue

1. Instalamos en Docker en nuestro Windows o Mac e iniciamos sección y creamos un archivo DockerFile en nuestro proyecto IntelliJ-idea que vamos a desplegar
2. Instalamos el Docker en el AWS con los siguientes comando:

# Add Docker's official GPG key:

- `sudo apt-get update`
- `sudo apt-get install ca-certificates curl gnupg`
- `sudo install -m 0755 -d /etc/apt/keyrings`
- `curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg`
- `sudo chmod a+r /etc/apt/keyrings/docker.gpg`

# Add the repository to Apt sources:

- `echo \`  
`"deb [arch="$(dpkg --print-architecture)" signed-`  
`by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \`  
`"$(. /etc/os-release && echo "$VERSION_CODENAME")" stable" | \`  
`sudo tee /etc/apt/sources.list.d/docker.list > /dev/null`
- `sudo apt-get update`

3. En la terminal del IntelliJ-idea utilizamos los siguientes comandos:

- `docker login`: para iniciar sesion
- `docker build -t tu-usuario/tu-repositorio:tag .` : Construye la imagen docker del proyecto
- `docker push tu-usuario/tu-repositorio:tag` : Publica la imagen en tu repositorio

4. En nuestra AWS creamos una carpeta para nuestro Docker y dentro de ella creamos un archivo llamado “compose.yaml”.

5. Dentro de ese archivo configuramos las direcciones y las conexiones de nuestro proyecto y guardamos

6. Escribimos el comando “`sudo docker compose up`” para levantar el servicio

7. Si queremos bajar el servicio lo hacemos con “`sudo compose down`”