

# IMPLEMENTACIÓN

PROYECTO DE DESARROLLO DE APLICACIONES MULTIPLATAFORMA - 2020

**Daniel Santano Fernández**

Descripción de los diferentes paquetes y clases de cada aplicación, explicación textual de cada uno de ellos, el proyecto consta de tres aplicaciones:

- **ApiRest NodeJs:**
  - **controlles:** paquete donde se encuentran los controladores de peticiones del apirest
    - **error:** controlador de errores global del api rest
    - **location:** controlador de peticiones sobre localizaciones
    - **services:** controlador de peticiones sobre servicios
    - **users:** controlador de peticiones sobre usuarios
  - **middleware:** paquete donde se encuentran los middelwares del apirest que sirven para poner un filtro a las peticiones
    - **auth:** filtro de autenticación
    - **has\_role\_admin:** filtro para comprobar que el rol del usuario es ADMIN
    - **has\_type\_user\_offering\_acommodation:** filtro para comprobar que el usuario es del tipo que ofrece alojamiento
    - **has\_type\_user\_offering\_service:** filtro para comprobar que el usuario es del tipo que ofrece servicios
  - **models:** paquete donde se encuentran los modelos de mongoose de la apirest
    - **location:** clase modelo de localización
    - **services:** clase modelo de servicio
    - **user:** clase modelo de usuario
  - **routes:** ficheros de rutas del apirest
    - **location:** rutas de localizaciones
    - **services:** rutas de servicios
    - **users:** rutas de servicios
- **Aplicación Android(app/src/main/java/com/dsantano/proyectodam):**
  - **common**
    - **Constans:** archivo con constantes de la app
    - **MyApp:** clase para obtener el contexto de la app

- SharedPreferencesManager: gestor de shared preferences
- data
  - repository: repositorios de los diferentes modelos de datos
  - viewmodel: viewModels de las diferentes ui
- datepicker
  - DialogFragment, Listener y DateTransformation para la gestión de fechas en nuestra app
- listeners
  - Diferentes listener de listados para el traspaso de datos a detalles (el id)
- models
  - locations: diferentes modelos necesarios para el tratamiento de localizaciones
  - auth: diferentes modelos necesarios para el tratamiento de el registro y login
  - services: diferentes modelos necesarios para el tratamiento de los servicios
  - users: diferentes modelos necesarios para el tratamiento de los usuarios
- retrofit
  - service: interfaz de servicio que usa la app
  - servicegeneratos: generador de servicio para registro y login y generador para peticiones autenticadas con JWT
- ui
  - auth: interfaz de usuario para login y registro
  - locations: diferentes interfaces de usuario para el tratamiento de las localizaciones
  - services: diferentes interfaces de usuario para el tratamiento de los servicios
  - users: diferentes interfaces de usuario para el tratamiento de los usuarios
  - MainActivity: activity que carga la navegación de la app y al hacer login
- Aplicación Angular(src/app):
  - core
    - Se tratan menús de la aplicación
  - dashboard
    - Se trata el contenido de la aplicación mayormente la parte donde se implementan el CRUD de los servicios

- models
  - Dto y Interfaces de los modelos necesarios para las acciones CRUD de la app
- services
  - Servicios de la app tanto google para el login, como las operaciones CRUD de los modelos
- session
  - Parte de login en la app y tratamiento de errores