APLICACIÓN AMAP 2.1

PRINCIPIO DE RESPONSABILIDAD ÚNICA	
Los servicios y componentes están codificados en distintos ficheros	
Cada fichero tiene un máximo de 400 líneas	
El máximo de líneas por función es de 75 líneas	
NOMENCLATURA	
GENERALIDADES - Nombrado de ficheros	
Se utilizan los nombres de manera homogenea	
Se emplea el patrón ***funcionalidad.tipo.ts*** en el nombrado de ficheros	
Se emplean guiones **"-"** para separar palabras en **nombres descriptivos**	
Se emplean puntos **"."** para separar el nombre descriptivo del **tipo**	
Se emplea el tipo "**service**" para identificar servicios	
Se emplea el tipo "**component**" para identificar componentes	
Se emplea el tipo "**pipe**" para identificar pipes	
Se emplea el tipo "**module**" para identificar módulos	
Se emplea el tipo "**directive**" para identificar directivas	
Los nombres de las clases y otros símbolos están en formato camel case	
Los nombres de las clases y otros símbolos coinciden con los nombres de los ficheros que lo contienen	
Se añade a los símbolos el sufijo en formato camel case con su tipo correspondiente	
BOOTSTRAPPING	
La lógica de arranque de la plataforma está contenida en el fichero **main.ts**	
El manejo de errores está incluido en la lógica de arranque	
No hay lógica de aplicación en el fichero main.ts	
SELECTOR DE COMPONENTES	
Se usa dashed-case o kebab-case para los nombres de los selectores de los componentes	
En caso de posible ambigüedad se utiliza un prefijo para cada selector	
SELECTOR DE DIRECTIVAS	
Se usa lower camel case para los selectores de directivas	
En caso de posible ambigüedad se utiliza un prefijo para cada selector	
PIPES	
Los nombres de los pipes indican correctamente su función	
CONVENCIONES DE CÓDIGO	
CLASES	
Se usa formato upper camel case para el nombrado de clases	
CONSTANTES	
Si el valor de una variable no cambia se define como const	
Se usa el formato lower came case para su definición	
INTERFACES	
Se usa el formato upper camel case para el nombrado de interfaces	
No se emplea el prefijo I para su nombrado	
PROPIEDADES Y MÉTODOS	
Se usa el formato lower came case para su definición	

Se usa el guión bajo "_" como prefijo para definir propiedades o métodos privados	
ESTRUCTURA DE LA APLICACIÓN Y NG-MODULES	
Todo el código fuente cuelga de la carpeta src	
Los ficheros de un mismo componente están agrupados en una única carpeta	
Se han creado carpetas por funcionalidad	
El módulo raíz está codificado en el fichero src/app/app.module.ts	
Se ha creado un NgModule para cada funcionalidad	
Cada módulo funcional está ubicado en una carpeta con su nombre	
La funcionalidad compartida se ubica en el fichero app/shared/shared.module.ts	