

EVENTS HALL

Manual del desarrollador

Contenido

Introducción	3
Guía para la generación de ejecutables	3
Instalación de Nodejs	3
Windows o Mac.....	3
Distribución Linux con gestor de paquetes	3
Instalación de angular-cli	3
Instalación de las dependencias.....	3
Generación de ejecutables.....	3
Tecnologías usadas en la aplicación.....	3
Nodejs	3
Electron	4
Angular	4
Otras tecnologías usadas	4
Arquitectura del sistema	5
AppModule.....	6
PresentationsModule.....	6
TwitterModule	7
TrafficModule.....	8
WeatherModule.....	8
VideoPlayerModule.....	9
Jerarquía de vistas.....	10

Introducción

Este manual tiene como objetivo facilitar a un desarrollador el manejo del código y la distribución de la aplicación. En este documento se describen las tecnologías usadas para la construcción de la aplicación. También se incluye una pequeña guía para construir un ejecutable o instalador (sin requerir conocimientos previos de las tecnologías usadas).

Guía para la generación de ejecutables

En los siguientes apartados se guiará en la instalación de los programas necesarios para generar los ejecutables a partir del código fuente.

Instalación de Nodejs

Windows o Mac

Para generar ejecutables es necesario instalar Node.js. Puede ser descargado desde la página oficial (<https://nodejs.org/es/>).

Distribución Linux con gestor de paquetes

Usando el gestor de paquetes de su distribución, como apt-get en Debian/Ubuntu, tendrá que instalar el paquete “nodejs”: “sudo apt-get install nodejs”.

También es necesario npm, el gestor de paquetes de Nodejs: “sudo apt-get install npm”.

Instalación de angular-cli

Angular Cli es la herramienta de línea de comandos que permite transformar el código fuente de la SPA (Single Page Application) hecha con Angular en un código que entienda el navegador web (Embebido por Electron en nuestro caso).

Instalación de las dependencias

Es necesario descargar las dependencias del proyecto antes de generar el ejecutable. Para ello hemos de situarnos con una terminal en la carpeta raíz del proyecto (contiene un archivo llamado package.json). Ejecutando el comando “npm install” se descargarán todas las dependencias en una carpeta llamada “node_modules”

Generación de ejecutables

Llegados a este punto es posible generar ejecutables utilizando únicamente un comando. De nuevo en la carpeta raíz del proyecto ejecutaremos uno de los siguientes comandos, dependiendo del sistema operativo:

- Windows: “npm run electron:windows”
- Linux: “npm run electron:linux”
- Mac: “npm run electron:mac”

Una vez ejecutado el comando que corresponda se habrán generado los ejecutables en la subcarpeta “app-builds”.

Tecnologías usadas en la aplicación

Nodejs

La aplicación es un paquete de Nodejs, gestionado con Npm. Esto permite una fácil gestión de dependencias y versionado.

Electron

La aplicación usa como base Electron. Electron es una librería que permite ejecutar aplicaciones web o Javascript en cualquier sistema operativo. Básicamente es una especie de navegador que muestra una página web como si de una aplicación nativa tratase. En la memoria del TFG se encuentra más información al respecto, así como en la página web de electron (<https://electronjs.org>).

Angular

Para construir la página web que muestra Electron se ha usado Angular. Angular es un framework que permite crear páginas web modulares. Su unidad básica es un componente, que contiene el código HTML, TypeScript y estilos que irán asociados a una vista. Cuenta con una excelente y amplia documentación. Entorno a este framework gira una gran comunidad. Su página web es <https://angular.io/>.

La aplicación utiliza la arquitectura recomendada por Angular. Esta se detalla en el último apartado de este documento.

Otras tecnologías usadas

Para complementar las anteriores se usan varios paquetes:

- Angular-Material, que incluye muchos componentes UI listos para usar con Angular. Gira entorno a “Material design”, creado por Google, que define unas normas de diseño para una experiencia de usuario óptima. <https://material.angular.io/>
- Electron-builder, que permite generar ejecutables e instaladores para distribuir la aplicación. <https://www.electron.build/>

Arquitectura del sistema

La arquitectura del sistema está inspirada por la propuesta de Angular. Se trata de una arquitectura modular, en la que en cada módulo se agrupan ciertos elementos relacionados con una característica concreta. Los elementos dentro de estos módulos son componentes o servicios. Cada componente es una pequeña implementación del patrón Modelo Vista Controlador (MVC).

La aplicación está centrada en la visualización de información. Así, los módulos estarán caracterizados por el tipo de información que representan. Existen 5 módulos, además del módulo app-raíz que engloba a los demás. Son los siguientes:

- **Presentations:** Contiene toda lógica y componentes asociado a la gestión y visualización de presentaciones. Usa el resto de los componentes para mostrar los submódulos (o paneles) que contiene.
- **Twitter:** Componentes y servicios para mostrar información de Twitter.
- **Traffic:** Componentes para mostrar información del tráfico en una zona.
- **Weather:** Componentes y servicios para mostrar la previsión del tiempo en una ubicación.
- **Video-player:** Componentes para la reproducción de vídeos.

Además de estos módulos, se ha hecho uso de muchos otros. En Angular las librerías externas de código también se agrupan en módulos, de los cuales se importan los componentes y servicios que se necesite. Las dependencias con módulos externos no se muestran en los diagramas ya que, dada la elevada reutilización de código, resulta muy complicado incluir todos los módulos externos utilizados en los diagramas y además dificultarían enormemente la lectura de los mismos.

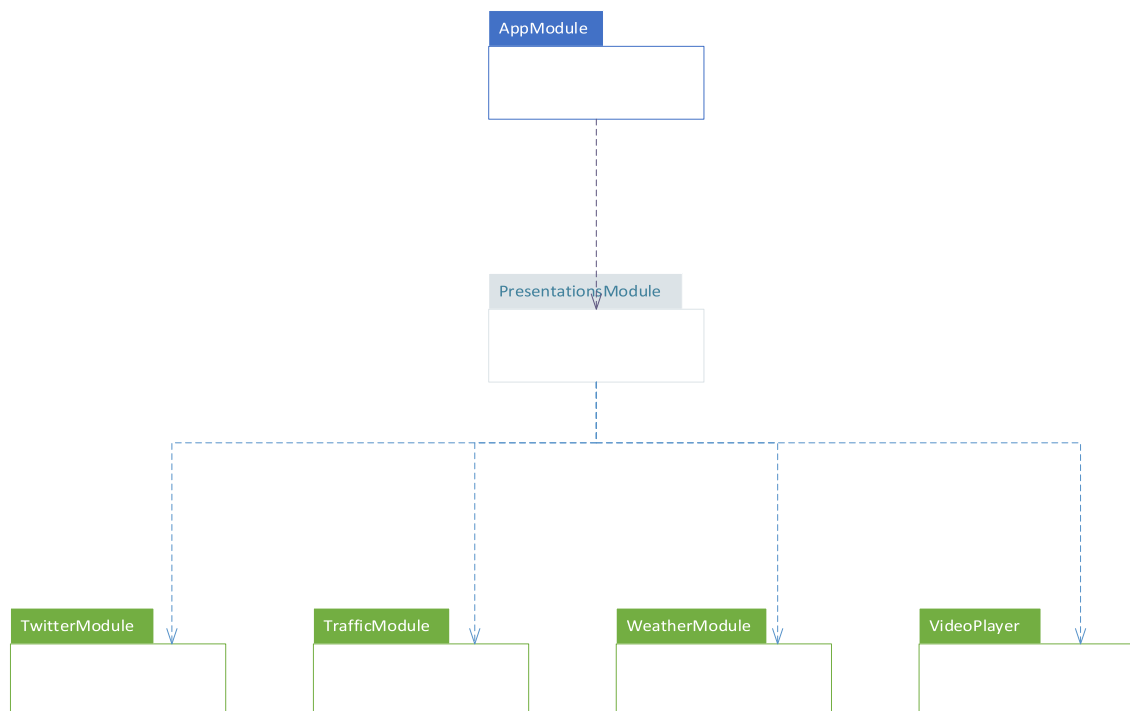


Ilustración 1 Diagrama de arquitectura

La Ilustración 1 expresa la arquitectura modular del sistema. Las dependencias entre los módulos vienen determinadas únicamente por las dependencias entre los componentes (vistas). Así, se puede establecer una jerarquía de vistas que muestra estas dependencias. Se mostrará más adelante, pero antes se detallará el contenido de cada uno de estos módulos.

AppModule

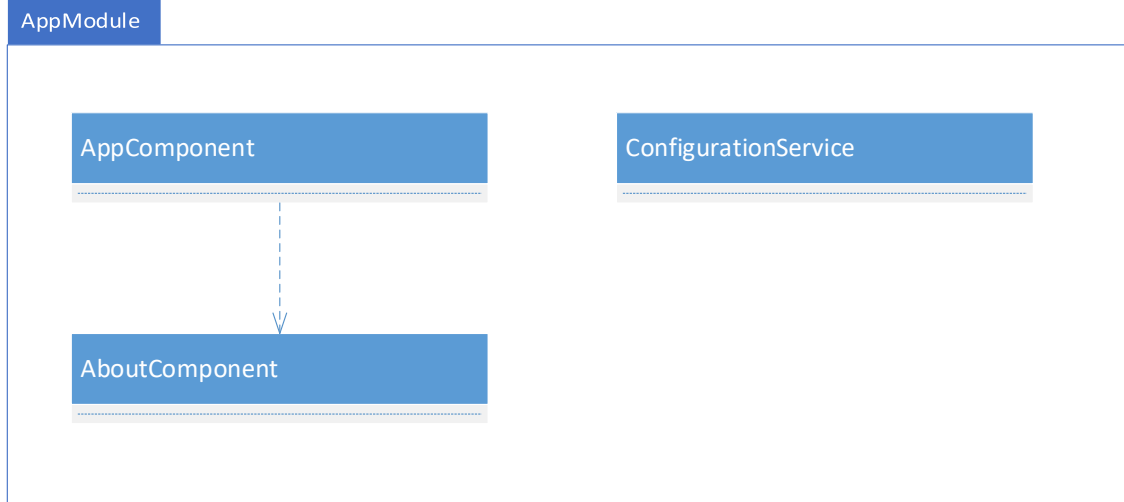


Ilustración 2 Módulo App

La Ilustración 2 muestra módulo App, o AppModule, cuenta únicamente con dos componentes. AppComponent es el componente raíz y es el inyectado por angular en el *index* de la página web. Utiliza componentes y servicios de AppModule y de PresentationsModule.

PresentationsModule

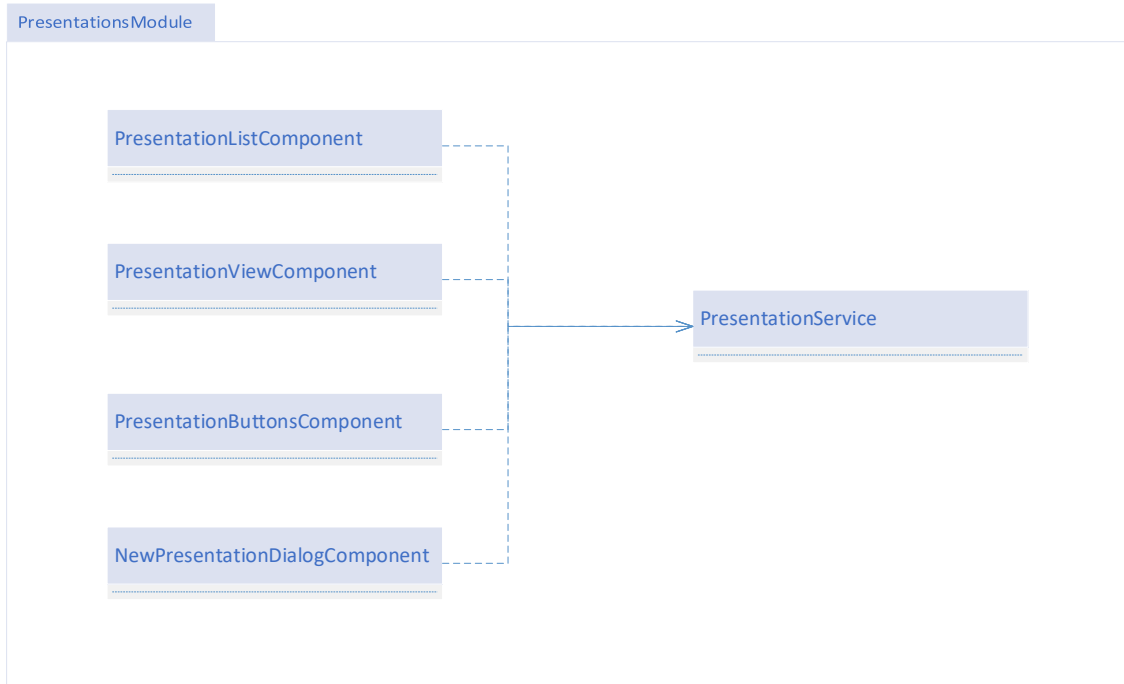


Ilustración 3 Módulo Presentations

La Ilustración 3 muestra el contenido del módulo *Presentations*.

Los componentes del módulo *Presentations* utiliza algunos componentes de los módulos destinados a cada subpanel (Twitter, Traffic, Weather y VideoPlayer). Todos estos módulos presentan una estructura similar, exponiendo un componente para mostrar la información configurada por otro componente que hace de formulario de configuración.

TwitterModule

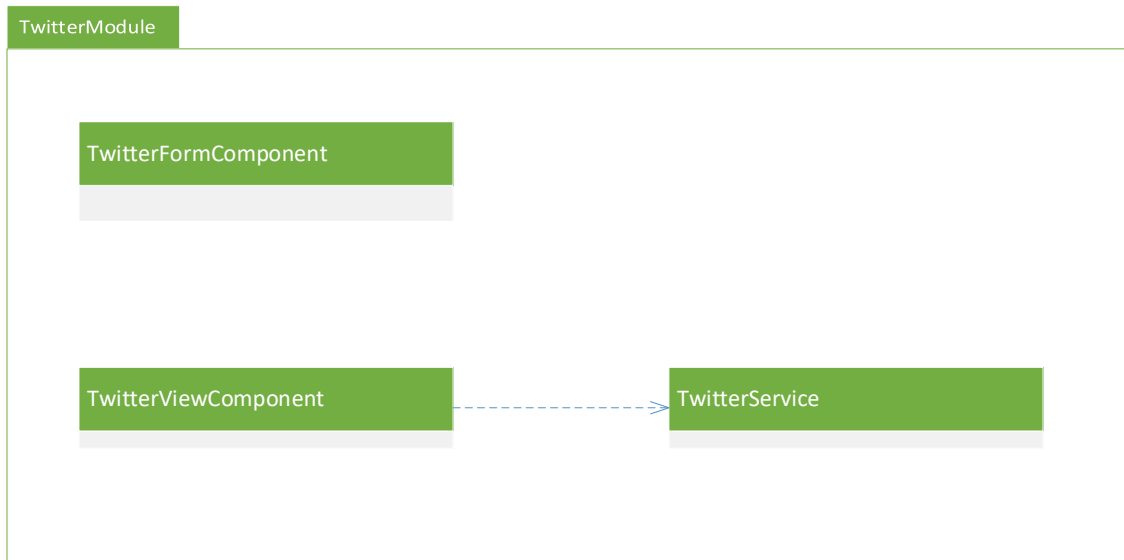


Ilustración 4 Módulo de Twitter

En este módulo, presentado en la Ilustración 4, el componente formulario (**TwitterFormComponent**) modifica un objeto **Module** (ver **¡Error! No se encuentra el origen de la referencia.**) actualizando la configuración según la entrada del usuario. En el resto de módulos de subpanel el funcionamiento es similar.

El componente **TwitterView** obtiene del servicio (**TwitterService**) unos *twits* que posteriormente muestra. El servicio se comunica directamente con la API de Twitter para obtener la información.

TrafficModule

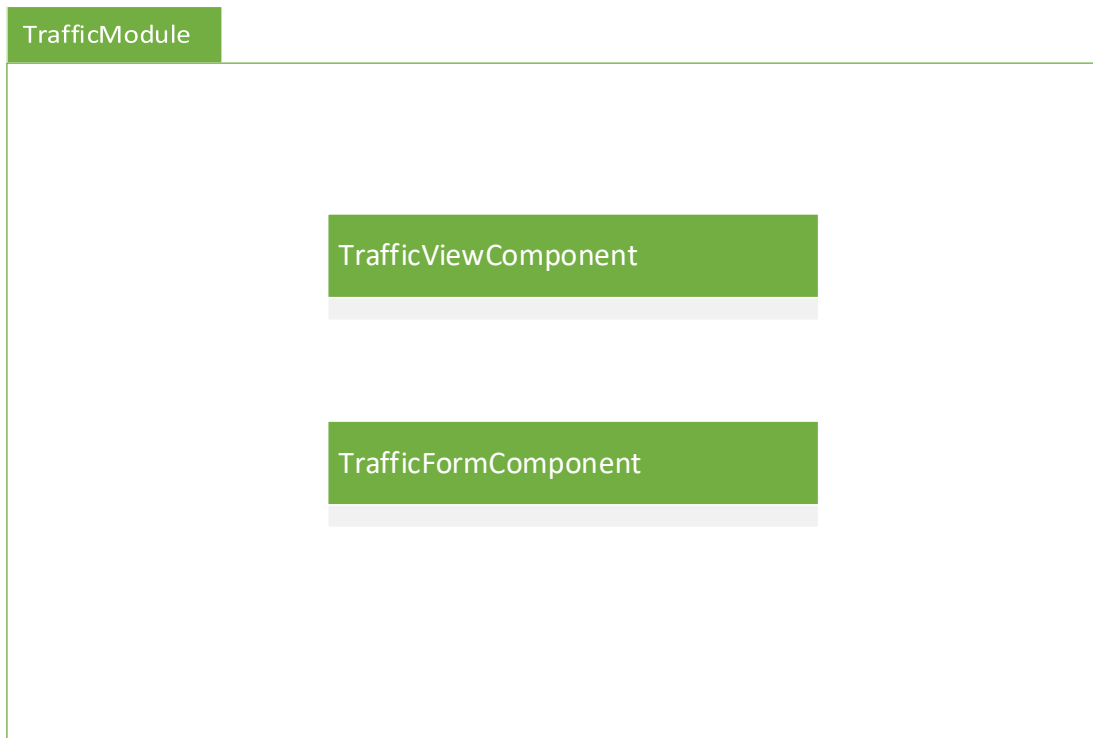


Ilustración 5 Módulo de Tráfico

El módulo de tráfico (visto en la Ilustración 5) solo consta de dos componentes, gracias al uso de componentes externos para dibujar el mapa. La librería usada es “angular-google-maps”.

WeatherModule

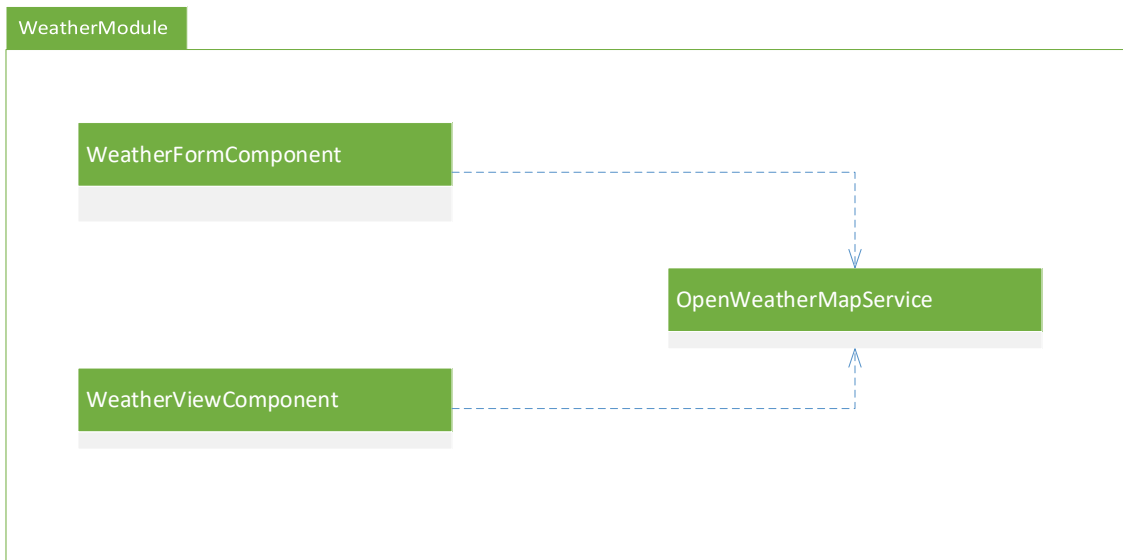


Ilustración 6 Módulo de previsión del tiempo

La Ilustración 6 muestra el módulo de previsión del tiempo. Cuenta con una estructura análoga a los anteriores. El servicio *OpenWeatherMapService* se sirve de la API de <http://openweathermap.org> para obtener la previsión del tiempo.

VideoPlayerModule

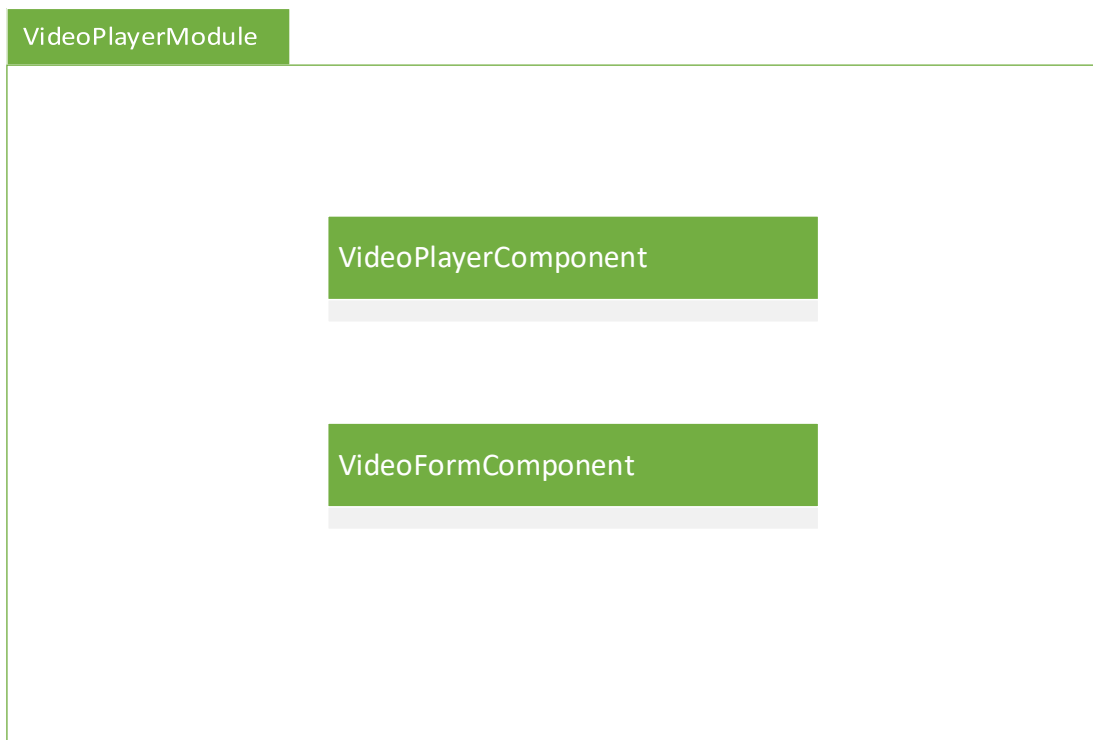


Ilustración 7 Módulo del reproductor de vídeos

La Ilustración 7 muestra el módulo del reproductor de vídeos. De nuevo solo cuenta con dos componentes. Para reproducir los videos se ayuda de componentes de la librería “videogular”, la cual actúa de interfaz con el elemento HTML5Video para reproducir los vídeos.

Jerarquía de vistas

Dado que Angular está centrada en el concepto de componente, y este agrupa el código relacionado con una vista, una forma más apropiada de expresar la arquitectura de la aplicación es con una jerarquía de vistas.

Visualmente, un componente se encarga de una porción de pantalla. El componente raíz sería el encargado de toda la pantalla. Sin embargo, normalmente es posible dividir una interfaz en pedazos con una función concreta. Estos pedazos de vista son extraídos a nuevos componentes, incluyendo su controlador. El componente raíz delega en diversos subcomponentes la responsabilidad sobre determinados pedazos de pantalla.

En la Ilustración 8 se muestra la jerarquía de vistas del sistema.

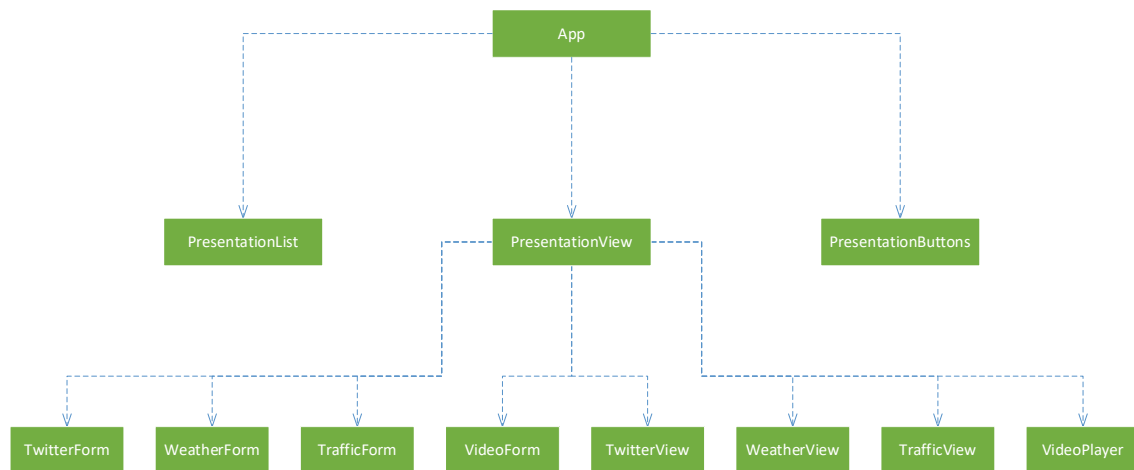


Ilustración 8 Jerarquía de vistas

Cabe destacar que, aunque las vistas de la jerarquía se corresponden con componentes de la aplicación, no aparecen en ella todos los componentes que se utilizan.