Ejercicio 1

La API utilizada es la API de la NASA. Concretamente se usan las imágenes que EPIC (Earth Polychromatic Imaging Camera) toma de la tierra, ofrecidas mediante la API llamada por el mismo nombre.

La descripción de las APIs de la NASA puede consultarse en https://api.nasa.gov/. En cuanto a la API utilizada (la API EPIC), la aplicación utiliza dos puntos de entrada:

1) *https://api.nasa.gov/EPIC/api/natural/all?api_key=<api-key>* Ofrece un JSON con una relación de fechas en las que existen imágenes.

   Así pues, en la página principal se sitúa el componente “dates”, que pide una relación de fechas gracias a este primer punto de entrada y las muestra como lista, usando el método getDates() de un servicio llamado “nasapic.service”. Todo esto se realiza de forma muy similar a lo descrito en la teoría de la PEC5.

2) *https://api.nasa.gov/EPIC/api/natural/date/<YYYY-MM-DD>?api_key=<api-key>* Ofrece un JSON con una relación de imágenes perteneciente a la fecha indicada.

   Cuando una fecha es pulsada, se solicita, a través de este segundo punto de entrada, la información de las imágenes que pertenecen a la fecha pulsada, la cual se recoge, a través del routing “/images/:date”. Todo esto también es similar a lo descrito en la teoría de la PEC5.

   Esta llamada devuelve, no uno, sino una lista de datos de imágenes (al contrario que la teoría). Además, a cada objeto del array devuelto por el servicio se le añade un campo que contiene la url de cada imagen, para que sea más sencillo, en el html de este componente, mostrar las imágenes y datos de cada imagen. Aquí se ha tenido que incorporar un *ngFor, por ser un listado, en vez de una sola imagen como en la teoría.

Ejercicio 2

src/main.server.ts	Creado	Un nuevo punto de entrada, con el modo de producción habilitado por defecto.
src/app/app.server.module.ts	Creado	Un nuevo módulo que contiene al módulo de la aplicación completa, y el módulo SerrverModule.
tsconfig.server.json	Creado	Un fichero de configuración del servidor, al estilo de tsconfig.json, donde se establece el JS al que se transpilará el código Typescript de nuestro proyecto, el punto de entrada, etcétera.
server.ts	Creado	La configuración del servidor Express, sobre qué puertos escuchará el servidor, y qué acciones tomará al recibir una petición.

package.json	Modif.	Añade scripts NPM y dependencias, como es lógico.
angular.json	Modif.	Cambia el directorio de compilación (“/dist/>proyecto>” -> “/dist/proyecto/browser”) y añade el fichero recién creado “tsconfig.server.json” a la configuración del transpilador “ts”.
src/main.ts	Modif.	Encapsula el código con un evento que se dispara cuando el contenido ha sido cargado.
src/app/app.module.ts	Modif.	Modifica el módulo Browser módule con una transición
src/app/app-routing.module.ts	Modif.	Configura RouterModule para que bootstrap sea bloqueado hasta la navegación inicial.

Ejercicio 3

Se ha utilizado el método manual descrito en la teoría. El repositorio y el apartado de Pages está en las siguientes urls:

https://github.com/techurbana/nasa-epic

https://techurbana.github.io/nasa-epic/images/2021-05-31