

ENUNCIADO PEC 5

PWA, Angular Universal, Deploy

Desarrollo front-end avanzado

**Máster Universitario en Desarrollo de
sitios y aplicaciones web**

UOC

Contenido

- Introducción
- Formato y fecha de entrega
- Enunciado
- Puntuación

Introducción

En esta práctica vamos a implementar una aplicación libre que consuma una api pública y posteriormente la transformaremos a **PWA**.

Finalmente, desplegaremos dicha aplicación en **Github**.

Formato y fecha de entrega

Se entregará todo el proyecto comprimido en formato **.zip** sin incluir los directorios “**node_modules**” y “**.angular**” junto con un documento de texto respondiendo a las dos/tres tareas escritas que se piden. La fecha de entrega se mostrará en el aula de la asignatura.

Enunciado

Ejercicio 1 – Implementar una PWA

En este primer ejercicio deberemos implementar una aplicación completa libre y deberá transformarse a **PWA**.

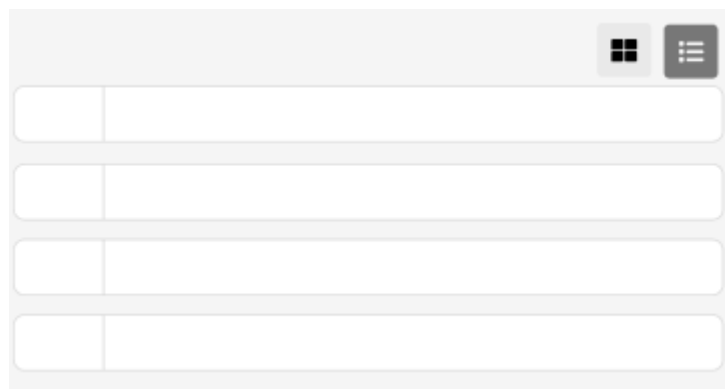
Podemos seguir los pasos de los materiales de la teoría del tema 5. La idea sería buscar una api pública que podamos consultar, que devuelva un listado de alguna entidad, e implementar una aplicación lista-detalle tipo hemos estudiado en la teoría (con la api de imágenes) y una vez hecha esta implementación pasarla a **PWA**.

Deberemos configurar el **manifest** y todo lo necesario del **service worker** para que pueda instalarse y funcionar sin internet.

Deberéis pensar en qué estrategias de caché seguir en función de los recursos que tengáis.

La aplicación deberá tener al menos:

- Component-list:
 - Una **home** donde aparezca el listado de la entidad devuelta por la api escogida utilizando una **tabla** o un listado de **cards** de **Angular Material**
 - Haremos la siguiente implementación:



En la parte superior derecha, implementaremos dos botones para intercambiar la vista “**cards**” de la vista “**tabla**”.

- Mientras esperamos el listado mostraremos el **spinner** de **Angular Material**

- La aparición del listado la haremos mediante alguna animación (elección libre)
- Al hacer clic a un elemento de la lista navegaremos a la página de detalle
- Component-detail:
 - Página donde mostraremos los detalles de la entidad.
 - Esta pantalla va a depender de cada api utilizada. Sería importante que la api devuelva algún tipo de imagen para poder comprobar cómo se cachea.
 - En la parte superior mostraremos solo un campo de la entidad, el nombre o el título (una propiedad que sea representativa de la entidad).
 - A la derecha mostraremos el botón de “**back**” para poder volver a la **home**
 - Debajo mostraremos un botón “**show all details**”, al pulsar el botón:
 - Se mostrarán el resto de las propiedades de la entidad
 - Para mostrar el resto de las propiedades deberemos utilizar al menos uno de los siguientes elementos de **Angular Material**:
 - **Tree**
 - **Tabs**
 - **Expansion panel**
 - **Progress bar**
 - **Slider**
 - *Si vemos que en la respuesta de la api nos falta información para poder utilizar alguno de los componentes anteriores nos podemos inventar algunas propiedades “añadiéndoles” en el array de respuesta “manualmente”.*

Implementación de componentes base

Tendremos que implementar al menos los siguientes componentes base:

- Componente **card** para reaprovechar en el listado de **cards** de la **home**. Al hacer clic en la *card* iremos al detalle de la entidad.
- Componente **grid** para reaprovechar en el listado de la **home** para cuando pulsemos el botón “modo tabla”. Al hacer clic a la fila iremos al detalle de la entidad.

Podéis implementar más componentes base en función de vuestros requisitos.

Una vez implementada, deberemos adjuntar un pequeño documento con algunos comentarios básicos de la información que nos muestra el informe generado por la

herramienta **LightHouse** de **Google Chrome** y las decisiones de diseño más relevantes en cuanto a configuración del **service worker**.

Ejercicio 2 – Desplegar una aplicación Angular en GitHub

Una vez tenemos una aplicación **Angular** finalizada, debemos subirla a un servidor remoto en modo producción. Para subir una aplicación **Angular** en modo producción disponemos de varias opciones:

- Si estuviéramos en un entorno real, deberíamos copiar todos los ficheros generados a un servidor web. Solamente tenemos que crear un **build** en modo producción de nuestra aplicación utilizando el siguiente comando:

- **ng build --outputHashing=all**

- el **flag outputHashing=all** es para forzar al navegador a que limpie la caché y se quede con la última versión del código, esto es súper útil ya que, si nosotros tenemos una **app** con muchos usuarios y vamos desplegando nuevas versiones, no podemos decirles a lo usuarios, oye! Limpia la caché para tener la última versión. Con este **flag**, forzamos a que limpie caché y que el navegador trabaje con la última versión del código.

Nota: Ten en cuenta que pueden aparecer problemas que en la fase de desarrollo no existían al hacer el **ng build**.

A continuación, debemos copiar el directorio **dist/** a nuestro servidor web (**Apache**, **nginx** o el que hayamos seleccionado para tal fin).

- Ahora que estamos estudiando, podríamos desplegar nuestros proyectos en páginas **GitHub**. Una aplicación **Angular** está compuesta por ficheros estáticos **HTML**, **JavaScript** y **CSS** que pueden ser servidos por cualquier servidor web. Es por ello por lo que se pueden desplegar aplicaciones **Angular** transpiladas en **GitHub**, puesto que el contenido que se genera es estático. Por lo tanto, en este ejercicio se pide que despleguemos vuestra aplicación en GitHub, para ello, podéis hacerlo de dos maneras:

- Siguiendo este tutorial podéis desplegar a GitHub de la manera “tradicional”:

Haciendo deploy de una app en angular a GitHub Pages - DEV Community

- Otra alternativa sería utilizar **Netlify**, aquí un tutorial:

Deploying Angular app with Netlify in 3 steps - DEV Community

Cuando tengáis la aplicación funcionando en **GitHub**, mirad de acceder des de vuestros teléfonos y validad que funciona como **PWA**, que os pide de instalar y funciona correctamente.

Añadid un documento de texto con la **url** resultante para poder revisar la aplicación desplegada.

Puntuación

A continuación, mostramos cuánto puntúan cada uno de los apartados de la práctica para obtener la nota final de la misma.

- Ejercicio 1[**8 puntos**]
- Ejercicio 2[**2 puntos**]