



# Desarrollo Web en Entorno Cliente

## Examen Convocatoria Ordinaria:

### PROCESO DE DESARROLLO

A partir de los archivos index.html y styles.css proporcionados, implementar la siguiente aplicación web en Angular:

1. Actividad 1:

Tarea: Crear las vistas: View1, View2, View3 y el componente header, desde el cual se podrá navegar a las tres vistas creadas. *(0.5 puntos)*.

2. Actividad 2.

Tarea: Crear los componentes Carousel y Pokemon instanciarlos en la vista View1. En la vista View1 implementa un formulario que permita al usuario introducir el nombre de un pokémon. Cuando el usuario pulse sobre el botón Enviar, se borrará la información del formulario y se realizará una petición GET a la API REST pública '[PokeAPI](#)' para obtener cuatro fotos de los sprites del Pokémon indicado. Para ello deberás implementar un servicio, las interfaces necesarias para gestionar la respuesta a la petición GET e instanciar el componente Pokemon en la vista View1. Empleando @Input, @Output y EventEmitter implementa la lógica necesaria para que se muestren en el componente Pokemon los datos siguientes: nombre del Pokemon y una de las fotos que puedes encontrar en los Sprites, obtenidos de la petición GET.

Al pulsar sobre el nombre del pokémon del componente Pokemon, se activará el componente Carousel -una pantalla modal-. Para este componente y empleando @Input, @Output y EventEmitter deberás implementar la lógica necesaria para que, a partir de los botones ‘anterior’ y ‘siguiente’ se puedan pasar las cuatro imágenes obtenidas de los Sprites de la petición GET del pokémon seleccionado - se deberá poder pasar de la primera foto a la última y viceversa-. Si el usuario hace click sobre la X, el componente Carousel se ocultará. Si el usuario hace click sobre el botón con la inscripción ‘volver’, se volverá a mostrar el formulario inicial. Si el usuario efectúa una nueva búsqueda, el resultado se acumulará, pudiéndose mostrar los componentes Pokemon con la información de todos los Pokémon buscados. (3,5 puntos).

3. Actividad 3.

Tarea: Crear el componente Ricks e instáncialo en la vista View2. Implementa la lógica necesaria para que al cargarse la vista se muestre mediante el componente Ricks los datos –imágenes- de los cinco primeros personajes con nombre Rick y los cinco primeros personajes con nombre Morty de la serie ‘Rick and Morty’, obtenidos mediante una petición GET a la API REST pública [The Rick & Morty API](#). Para ello deberás implementar las funciones pertinentes en el servicio creado para la actividad anterior y las interfaces necesarias para gestionar la respuesta a la petición GET. Empleando @Input, @Output y EventEmitter implementa la lógica necesaria para que se muestren inicialmente en el componente Ricks los datos del primer personaje con nombre Rick y del primer personaje con nombre Morty. Si el usuario hace click sobre la imagen de cualquiera de los dos personajes, se ocultará y se mostrarán las imágenes de los cinco personajes en miniatura -es decir, si se hace click sobre la foto del personaje con el nombre Rick, se oculatarán las fotos de los dos personajes y se mostrarán las cinco imágenes de los cinco personajes con nombre Rick, en miniatura-. Si el usuario hace ‘click’ sobre cualquiera de las cinco imágenes en miniatura, se ocultarán todas ellas y se mostrará la imagen seleccionada agrandada. Si vuelve a hacer click, se mostrará el menú inicial. (3 puntos).

4. Actividad 4.

Tarea: Crear el componente Card e instáncialo en la vista View3. En la vista View3 implementa un formulario que permita al usuario introducir el nombre de un personaje de la saga de películas Star Wars y realizar una búsqueda de este personaje mediante una petición GET a la API REST pública '[SWAPI - Star Wars API](#)' para obtener el nombre, la altura y el peso así como el título y el opening crawl de todas las películas en las que aparece dicho personaje. Para ello deberás implementar las funciones pertinentes en el servicio creado para las actividades anteriores y las interfaces necesarias para gestionar la respuesta. Cuando el usuario introduzca un término de búsqueda en el formulario y pulse sobre el botón Enviar, se borrará la información del formulario.

Implementa la lógica necesaria para mostrar los datos del personaje introducido, empleando para ello el componente Card y @Input. Si se pulsa sobre el botón 'nueva búsqueda', se volverá al formulario inicial, reiniciado. Si el usuario realiza nuevas búsquedas, se mostrarán con el componente Card los resultados de las búsquedas acumuladas.

*(3 puntos).*

## **EVALUACIÓN**

---

Esta prueba supondrá el 50% de la nota de la convocatoria ordinaria.

## **OBSERVACIONES**

---

Se entregará en Florida Oberta un único archivo comprimido con la carpeta src del proyecto Angular implementado. Los archivos a entregar deberán ceñirse a lo que se pide explícitamente en cada enunciado de la actividad. Las funcionalidades no solicitadas o el uso de librerías de terceros supondrán una penalización de la nota.

**La copia o plagio de contenido de compañeros, la utilización de herramientas de IA o la utilización del móvil o auriculares durante la realización del examen conllevará suspender automáticamente la convocatoria ordinaria de la asignatura.**