

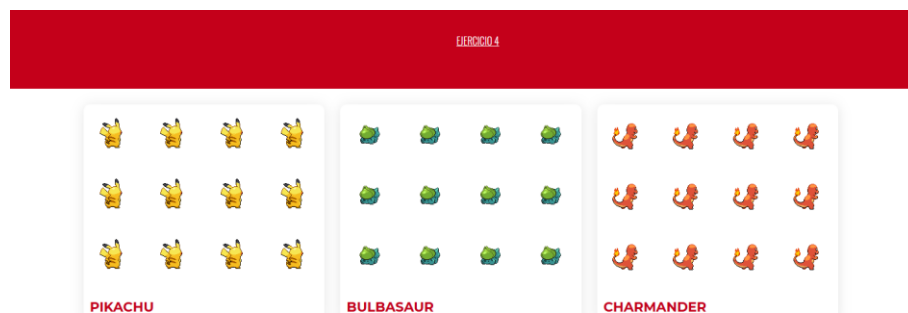
# Desarrollo Web en Entorno Cliente

## 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: *Ejercicio2*, *Ejercicio3*, *Ejercicio4* 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 *Pokemon* y *Photos* e instanciarlos en la vista *Ejercicio2*. Implementa la lógica necesaria para que al cargarse la vista se muestre mediante el componente *Pokemon* los datos de los pokémon 'Pikachu', 'Bulbasaur' y 'Charmander', obtenidos mediante una petición GET a la API REST pública 'PokeAPI'. 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 *Ejercicio2*. 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 esconderá el componente Pokemon y se mostrará el componente Photos. Para este componente y empleando @Input, @Output y EventEmitter deberás implementar la lógica necesaria para que se muestren inicialmente cuatro imágenes en miniatura obtenidas de los Sprites de la petición GET del pokémon seleccionado – pikachu, bulbasaur o charmander-. Si el usuario hace click sobre cualquiera de las cuatro imágenes, se ocultarán las cuatro imágenes y se mostrará únicamente la imagen seleccionada agrandada. Si vuelve a pulsar sobre la imagen agrandada, se volverán a mostrar las cuatro imágenes. Si el usuario hace click sobre el botón con la inscripción ‘volver’, se volverá al estado inicial, con los tres componentes Pokémon mostrando los datos indicados para los pokémon ‘Pikachu’, ‘Bulbasaur’ y ‘Charmander’. (3,5 puntos).

EJERCICIO 4

ATRÁS



EJERCICIO 4

ATRÁS



3. Actividad 3.

Tarea: Crear el componente Carousel e instanciarlo en la vista Ejercicio3. Implementa la lógica necesaria para que al cargarse la vista se muestre mediante el componente Carousel los datos de los personajes de la serie 'Rick and Morty' – nombre e imagen-, 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 en el componente Carousel los datos de los personajes de la serie, el componente debe permitir que cuando el usuario haga click sobre los botones anterior y siguiente del componente, se pueda navegar por los 20 personajes en que divide cada página la paginación de la API –se debe de poder pasar del primer personaje al último y viceversa-. Además, en la vista Ejercicio3 habrá otros dos botones anterior y siguiente, que permitirán navegar por las paginas de la paginación –también se deberá permitir pasar de la primera página a la última y viceversa-.

EJERCICIO 4

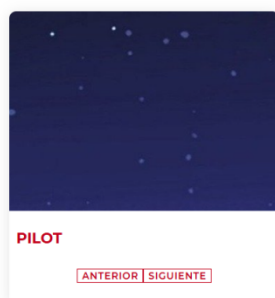
PERSONAJES | EPISODIOS



ANTERIORES | SIGUIENTES

Finalmente, la vista ofrecerá la posibilidad de elegir entre mostrar los personajes o los capítulos de la serie. En el caso de que el usuario seleccione los capítulos, se mostrará el nombre del capítulo y una imagen por defecto. (3,5 puntos).

PERSONAJES | EPISODIOS



ANTERIORES | SIGUIENTES

4. Actividad 4.

Tarea: Crear el componente Text e instancialo en la vista Ejercicio4. En la vista Ejercicio4 implementa un formulario que permita al usuario introducir una palabra en inglés y realizar una búsqueda de esta palabra mediante una petición GET a la API REST pública '[Free Dictionary API](#)' para obtener sinónimos, antónimos y significados de la palabra introducida. 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 mostrarán los botones 'significados', 'sinónimos' y 'antónimos'.

Implementa los eventos necesarios para que cuando se pulse sobre el botón significados, se muestren todos los significados de la palabra introducida, empleando para ello el componente Text y @Input. Si se pulsa sobre el botón 'sinónimos' se mostrarán todos los sinónimos de la palabra introducida. Si se pulsa sobre el botón 'antónimos', se ocultarán los botones y se mostrarán todos los antónimos de la palabra introducida. Debajo de los resultados correspondientes para cada opción –significados, sinónimos o antónimos-, habrá dos botones: 'volver', que permitirá volver al menú anterior –significados, sinónimos, antónimos- y 'nueva búsqueda', que permitirá volver al formulario inicial, reiniciado.

(2,5 puntos).



<input type="text" value="nombre"/>	<input type="button" value="Enviar"/>
-------------------------------------	---------------------------------------



SIGNIFICADOS	SINÓNIMOS	ANTÓNIMOS
--------------	-----------	-----------

### EJERCICIO 3

The state of being free, of not being imprisoned or enslaved.

The lack of a specific constraint, or of constraints in general; a state of being free, unconstrained.

Frankness; openness; unreservedness.

Improper familiarity; violation of the rules of decorum.

[VOLVER](#) [NUEVA BÚSQUEDA](#)

### EJERCICIO 3

exemption

liberty

license

[VOLVER](#) [NUEVA BÚSQUEDA](#)

### EJERCICIO 3

bondage

constraint

imprisonment

slavery

[VOLVER](#) [NUEVA BÚSQUEDA](#)

## 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.**