

Ejercicio de evaluación - Sprint 3

Antes de empezar, tenéis que [crear un nuevo repositorio desde GitHub Classroom](#). Una vez creado, lo clonamos en nuestro ordenador y en la carpeta creada empezaremos a trabajar en el ejercicio.

El ejercicio consiste en desarrollar una página web con un listado de personajes de *Harry Potter*, que podemos filtrar por el nombre del personaje. Vamos a usar React para realizarlo.

Harry Potter Characters

Character	House
Harry Potter	Gryffindor
Cho Chang	Ravenclaw
Rubeus Hagrid	Gryffindor
Kingsley Shacklebolt	SIN CASA

Vamos de definir los distintas partes del ejercicio:

1. Listado de personajes

En primer lugar, vamos a realizar una web con el listado de personajes de Harry Potter. Para eso, vamos a utilizar el servicio de <http://hp-api.herokuapp.com/> que nos devuelve información sobre 25 personajes de la saga. Sobre cada uno, vamos a pintar al menos:

- nombre
- foto
- casa

Para esta primera parte del ejercicio es suficiente pintar la información sin maquetar.

2. Filtrado de personajes

Ahora que ya tenemos el listado de personajes en pantalla, la segunda parte consiste en poder buscarlos por nombre. Para eso, añadimos un [input](#) a la interfaz, de forma que al ir escribiendo un nombre queden en la interfaz solo los personajes cuyo nombre contiene las letras escritas. En el pantallazo de arriba, al escribir 'ha' aparecen personajes cuyo nombre completo contiene esas letras en ese orden.

NOTA: en principio no es necesario tener en cuenta si las letras están en mayúscula/minúscula para la búsqueda, pero si queréis añadir esta mejora pues genial.

3 Componentes del listado de personajes

El listado debe tener los siguientes componentes como mínimo:

- Componente para los filtros
- Componente para el listado
- Componente para la tarjeta de cada personaje del listado

Como en el ejemplo:



4. Detalle de personajes

Vamos a implementar una nueva funcionalidad: al hacer clic sobre la tarjeta de un personaje, su información aparecerá a pantalla completa. Para hacer esto usaremos rutas y React router. En la pantalla de detalle aparecerá además de nombre, foto y casa, el año de nacimiento, el patronus y si está vivo o muerto al final de la saga.



Ya que esta API no nos devuelve resultados con un identificador único por objeto tendremos que añadirle uno nosotros 😊

PISTA: ¿Quizás podríamos usar el operador spread para conseguirlo? ¿Yay or nay?

5. BONUS: Mejoras visuales

Para terminar, podéis realizar algunas mejoras visuales del ejercicio. Por ejemplo:

- mostrar la casa y si un personaje está muerto con un ícono
- usar algún sistema de grid para pintar el listado de personajes
- que funcione bien el responsive en dispositivos pequeños

Entrega

La entrega del ejercicio se realizará en el mismo repositorio que has creado al comienzo del ejercicio. Hemos pautado 12 horas de dedicación al ejercicio, por lo que el límite de entrega es

- turno de mañana: próxima sesión a las 15:00h
- turno de tarde: próxima sesión a las 22:00h

Normas

Este ejercicio está pensado para que lo realices de forma individual en clase, pero podrás consultar tus dudas con la profesora y tus compañeras si lo consideras necesario. Aún facilitando la comunicación entre compañeras, durante la prueba está prohibido copiar código de otra persona o acceder a su portátil. Confiamos en tu responsabilidad. La evaluación es una buena oportunidad para conocer cómo estás progresando, saber qué temas debes reforzar durante las siguientes semanas y cuáles dominas. Te recomendamos que te sientas cómoda con el ejercicio que entregues y no envíes cosas copiadas que no entiendas. Si detectamos que has entregado código copiado de una compañera, no evaluaremos tu ejercicio y pasarás directamente a la re-evaluación del sprint. Tu objetivo no debería ser pasar la evaluación sino convertirte en programadora, y esto debes tenerlo claro en todo momento. Una vez entregado el ejercicio realizarás una revisión del mismo con la profesora (30 minutos), que te pedirá que expliques las decisiones tomadas para realizarlo y te propondrá realizar cambios *in situ* sobre tu solución. Al final, tendrás un feedback sobre aspectos a destacar y a mejorar en tu ejercicio, y sabrás qué objetivos de aprendizaje has superado de los listados a continuación.

Criterios de evaluación

Vamos a listar los criterios de evaluación de este ejercicio. Si superas más del 80% de los criterios, estás aprendiendo al ritmo que hemos pautado para poder afrontar el proyecto final.

React básico

- Crea componentes con sintaxis correcta
- Crea una estructura adecuada de componentes
- Usa las props para pasar datos a componentes hijos
- Sabe pintar listados de datos
- Usa el estado para almacenar información de la interfaz
- El componente principal maneja el estado de la aplicación
- Usa eventos en React para atender a interacciones del usuario

- Usa lifting para pasar info de hijos a padres
- Usa métodos del ciclo de vida como componentDidMount para las peticiones AJAX
- Escribe un código sólido, sin errores en la consola
- Usa propTypes para evitar errores de tipado

React router

- Crea rutas navegables dentro de una aplicación

Otros criterios a tener en cuenta

- Usar inglés para nombres de variables, funciones, clases, mensajes de commit, nombres de ficheros
- El repositorio de GitHub debe tener README y un enlace a la web en GitHub Pages accesible desde la página principal
- El perfil de GitHub debe tener datos de contacto, organización de Adalab visible, repositorios destacados

¡Mucha suerte, muggles!