

Módulo 2: Ejercicio de evaluación final bis

- Antes de empezar, hay que crear un nuevo repositorio desde GitHub Classroom usando [este enlace](#). Una vez creado, hay que clonar en nuestro ordenador y en la carpeta creada empezaremos a trabajar en el ejercicio.
- A continuación hay que descargar e incluir en el proyecto el [starter kit de Adalab](#).

Enunciado

El ejercicio consiste en desarrollar una aplicación web con el juego de cartas de buscar las parejas.

El juego arranca la una opción marcada (por ejemplo 4) y el correspondiente número de cartas boca abajo sobre la mesa.

Si la usuaria elije otra opción (6 u 8) y pulsa en **Comenzar** hay que mostrar el número correspondiente de cartas sobre la mesa, también boca abajo.

El juego de las parejas

☒ 4 ☐ 6 ☐ 8



Cuando la usuaria pulse en una carta hay que darle la vuelta a la carta mostrando la parte frontal de la carta.

El juego de las parejas

☒ 4 ☐ 6 ☐ 8



Cuando la usuaria pulse una segunda carta, si es la pareja de la primera carta hay que dejar las dos cartas boca arriba hasta el final del juego. Si por el contrario no son la misma carta hay que esperar un segundo y poner las dos cartas boca abajo.

Cuando la usuaria haya encontrado todas las parejas el juego ha acabado.

Fases del ejercicio

Arranque de la página:

- Leer desde los radio buttons la opción seleccionada, por ejemplo un **4**.
- Pedir la información de las cartas a un API.
- La dirección del API es <https://beta.adalab.es/ejercicios-extra/api/pokemon-cards/:NUMERO.json>. Hay que sustituir **:NUMERO** por la opción seleccionada, por ejemplo <https://beta.adalab.es/ejercicios-extra/api/pokemon-cards/4.json>.
- Cuando el API haya contestado tenemos que pintar en pantalla el número de cartas que nos haya respondido el API.

Escuchar el botón Comenzar:

Cuando la usuaria pulse en el botón **Comenzar** hay que:

- Guardar en el local storage la opción seleccionada (4, 6 u 8).
- La próxima vez que se acceda a la página habrá que comprobar si esta está opción en el local storage y marcar el radio button correspondiente.
- A continuación hay que volver a ejecutar los pasos de la fase **Arranque de la página**, es decir, reiniciar el juego.

Voltear las cartas:

Cuando la usuaria pulse en cada una de las cartas hay que:

- Mostrar la carta pulsada boca arriba.
- Si la usuaria pulsa en una carta que ya está boca arriba no hay que ponerla boca abajo.

Bonus

Las siguientes fases no son obligatorias para la evaluación. Os recomendamos desarrollar las fases anteriores hasta que funcionen perfectamente. A continuación limpiar y optimizar el código lo más posible. Solo después os animamos a continuar con los bonus.

Barajar las cartas

Al recuperar las cartas del API hay que cambiar su orden aleatoriamente antes de mostrarlas en pantalla. Hay muchas formas de hacerlo. Os aconsejamos buscar en Google cómo barajar un array de datos en JavaScript.

Implementar el juego

- En los datos del API hay un campo que nos indica qué cartas son parejas.
- Al pulsar en una primera carta hay que mostrarla boca arriba.
- Al pulsar en una segunda carta también hay que mostrarla boca arriba.
- Después de 1 segundo hay que comprobar si las dos cartas pulsadas son la misma.
 - Si son pareja hay que dejar las cartas boca arriba para siempre.
 - Si no lo son hay que poner las dos cartas boca abajo.

Esta parte del ejercicio es en sí un reto, así que no os desesperéis si no lo conseguís en los primeros intentos



Hay muchas formas de implementar esta parte del juego y hay que contar con muchos factores, como por ejemplo si la usuaria pulsa muchas cartas rápidamente o si hay dos o más cartas boca arriba. En estos casos lo mejor es pintar un buen diagrama de flujos a prueba de bombas y después empezar a programar.

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. El límite de entrega es 24 horas antes de la evaluación presencial.

La evaluación solo se considerará terminada cuando:

- Esté publicada en GitHub Pages y esté funcionando, para lo cual tendréis que subir el código, también a la carpeta `docs/` del repositorio.
- El enlace a GitHub Pages esté en la página principal del repositorio, en la parte superior, al lado de la descripción.

Normas

Este ejercicio está pensado para que lo realices de forma individual, pero podrás consultar tus dudas con la profesora y tus compañeras si lo consideras necesario. Ellas no te darán directamente la solución de tu duda, pero sí pistas para poder solucionarla. Aún facilitando la comunicación entre compañeras, durante la prueba no debes copiar código de otra persona ni 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.

Una vez entregado el ejercicio realizarás una revisión del mismo con la profesora (25 minutos), que se asemejará a una entrevista técnica: te pedirá que expliques las decisiones tomadas para realizarlo y te propondrá realizar cambios in situ sobre tu solución.

Es una oportunidad para practicar la dinámica de una entrevista técnica donde te van a proponer cambios sobre tu código que no conoces a priori. Si evitas que otras compañeras te den pistas sobre la dinámica de feedback, podrás aprovecharlo como una práctica y pasar los nervios con la profesora en lugar de en tu primera entrevista de trabajo.

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 los conocimientos del siguiente módulo.

JavaScript básico

- Crear código JavaScript con sintaxis correcta, bien estructurado e indentado*.
- Usar constantes/variables para almacenar información y re-asignar valores*.
- Usar condicionales para ejecutar acciones distintas en función de una condición.
- Saber trabajar con listados de datos (arrays)*.
- Usar funciones para estructurar el código.
- Saber modificar la información del DOM para añadir contenido dinámico*.
- Saber escuchar eventos del DOM y actuar en consecuencia*.

AJAX y APIs

- Crear peticiones con fetch y promesas*.
- Saber trabajar correctamente con la respuesta del servidor*.
- Gestionar información en formato JSON.
- Usar el localStorage para guardar información en el navegador.

Issues

- Haber resuelto las issues de la evaluación intermedia.

Otros criterios a tener en cuenta

- Usar inglés para nombres de variables, funciones, clases, mensajes de commit, nombres de ficheros.

¡Al turrón!