Ejercicio de evaluación final (bis) - Módulo 2

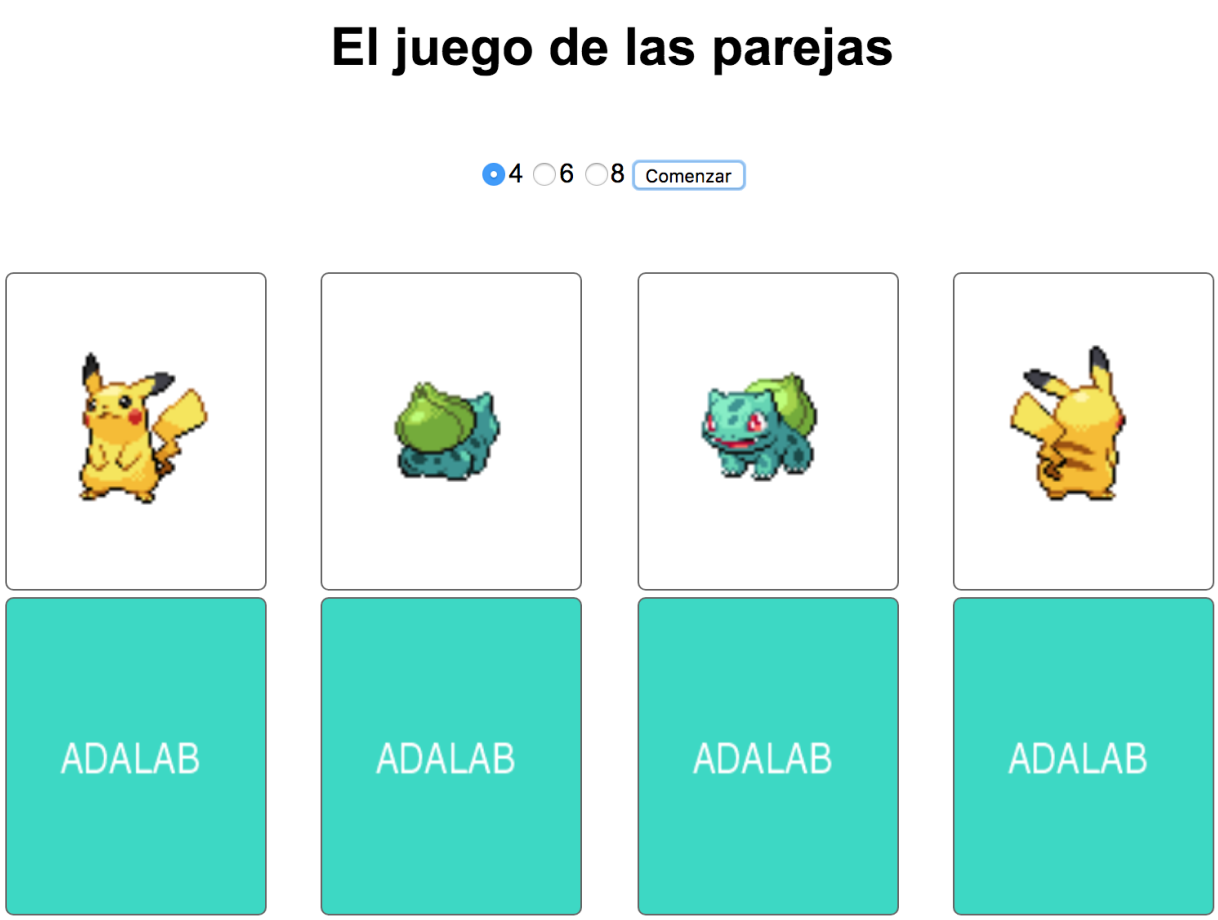
El ejercicio consiste en desarrollar una aplicación web que simula un juego de buscar las parejas. El objetivo del ejercicio es desarrollar la interfaz del juego, **no implementar el juego en sí** que quedará como un BONUS.

Antes de empezar, tenéis que crear un nuevo repositorio desde GitHub Classroom usando este enlace. Una vez creado, lo clonamos en nuestro ordenador y en la carpeta creada empezaremos a trabajar en el ejercicio.

Vamos de definir los distintos hitos del ejercicio:

1. Maquetación

En primer lugar vamos a realizar una maquetación básica del juego.



La aplicación consta de dos partes

1. Un formulario para elegir el tamaño de la partida y un botón
2. Un listado de cartas, cada elemento del listado tiene 2 imágenes que representan las dos caras de una carta (cuando una está visible la otra está oculta).

En la imagen de arriba tienes un ejemplo de un listado de 4 cartas, cada una tiene una imagen de la parte frontal con un Pokemon y de la parte trasera con la palabra ADALAB.

La imagen para la parte frontal la obtendremos de un servidor y para la trasera usaremos https://via.placeholder.com/160x195/30d9c4/ffffff/?text=ADALAB.

Para realizar la maquetación básica del ejercicio usaremos Sass y la base de gulp del Adalab Web Starter Kit.

2. Inicio de la partida

Al hacer clic sobre el botón de 'Comenzar', nuestra aplicación debe recoger el valor del tamaño de la partida y conectarse a un API que devuelve un listado de cartas. La URL del API es `https://raw.githubusercontent.com/Adalab/cards-data/master/:NUMERO.json`, donde *:NUMERO* puede tomar el valor de 4, 6 u 8. Por ejemplo, para pedir 6 cartas usariamos https://raw.githubusercontent.com/Adalab/cards-data/master/6.json

Por cada carta obtendremos, entre otros datos, la URL de la imagen a mostrar. Estos datos los usaremos para pintar tantas cartas como datos nos lleguen del servidor. Recuerda que cada carta tiene 2 caras y por defecto la frontal debe estar oculta.

3. Almacenamiento local

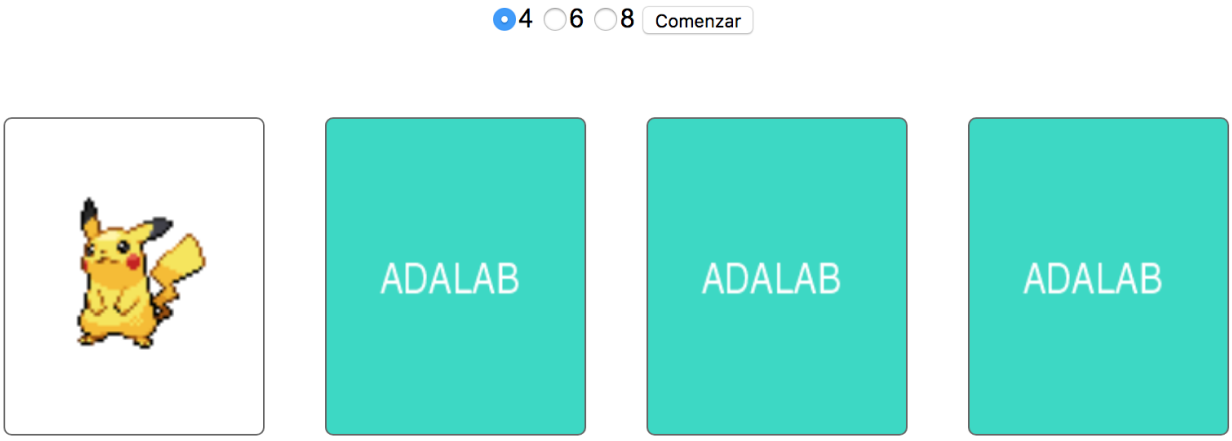
Vamos a guardar el número que la usuaria elija en localStorage, de forma que al recargar la página aparezca seleccionado el número que se eligió la última vez.

4. Interacción

Una vez mostramos el listado de cartas vamos a hacer que sea interactivo.

Al hacer clic sobre una carta vamos a mostrar su parte frontal y a ocultar su parte trasera. Al volver a hacer clic haremos la operación contraria, y volveremos a ver su parte trasera y ocultar la frontal.

El juego de las parejas



En la imagen anterior podemos ver algunas cartas boca arriba y otras boca abajo.

5. BONUS: Implementar el juego

Una vez terminada la parte obligatoria, os animamos a intentar implementar el juego de las parejas en una rama del repositorio. Por cada carta, tenemos información en el JSON de cuáles son pareja.

- Cuando se hace clic en una primera carta esta se da la vuelta y nos muestra su pokemon (como hasta ahora).
- Al hacer clic en una segunda carta esta se da la vuelta y: si es la pareja de la primera las dos se quedan boca arriba (como hasta ahora), si no es la pareja de la primera las dos deben mantenerse visibles durante un periodo corto de tiempo (para que la usuaria vea los pokemon) y ponerse boca abajo.

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

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, pero debido a que lo entregamos en el periodo de vacaciones debería estar entregado para el día 7 de mayo de 2019 a las 5pm. Por supuesto os animamos a que intentéis la parte de bonus y completéis el juego de las parejas.

Como orientación podéis hacer una rama para el ejercicio y una de bonus para el juego completo.

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 módulo. 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 no superas al menos el 80% de estos criterios o no has superado algún criterio clave (marcados con *) te pediremos que realices una re-evaluación con el fin de que termines el curso mejor preparada y enfrentes tu primera experiencia profesional con más seguridad. En caso contrario, 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
- El repositorio de GitHub debe tener README y un enlace a la web en GitHub Pages accesible desde la página principal

¡Al turrón!