Módulo 2: Ejercicio de evaluación final

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.

Enunciado

El ejercicio consiste en desarrollar una tienda online de productos, que nos permite añadir/quitar los productos al carrito de la compra y guardarlos en local storage.

El ejercicio también tiene una parte de maquetación con HTML y Sass, os recomendamos dedicar esfuerzo a la maquetación una vez terminada la parte de JavaScript, ya que los criterios de evaluación están relacionados con esta última.

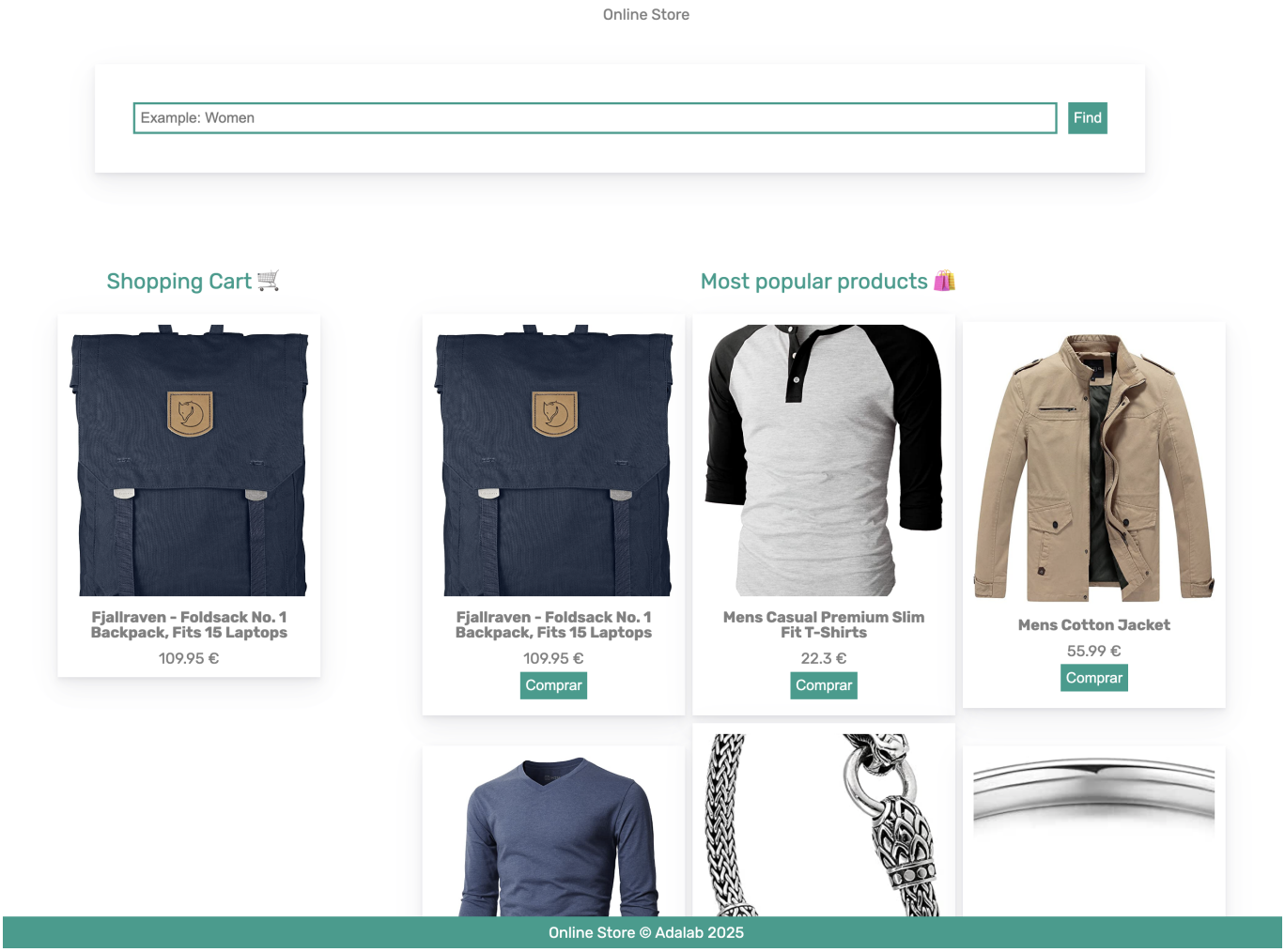
Vamos de definir los distintos hitos del ejercicio:

1. Estructura básica

En primer lugar hay que realizar una estructura básica sobre este modelo. No hay que preocuparse por las medidas, colores ni tipografía hasta un hito posterior.

La tienda online de productos consta de dos partes:

1. Un campo de texto y un botón para buscar productos por su título.
2. Un listado de resultados de búsqueda donde aparece la foto del producto, el nombre, precio y un botón para añadir el producto al carrito de la compra.



2. Búsqueda

- Al levantar la aplicación debe conectarse a un API para obtener los datos de los productos y mostrarlas en el listado de resultados.:
  - **Opción 1:** https://fakestoreapi.com/products. Esta APi https://fakestoreapi.com/docs devuelve un listado de productos. Nos puede banear la IP si hacemos muchas peticiones, pero si lo hace, se puede usar la opción 2.
  - **Opción 2:** https://raw.githubusercontent.com/Adalab/resources/master/apis/products.json. Esta opción debe ser usada si la opción 1 no funciona.
  - Os recomendamos echar un vistazo al JSON que devuelve una petición de búsqueda para ver qué datos son los que necesitamos.
- Por cada producto contenido en el resultado hay que pintar una tarjeta donde mostramos una imagen del producto y el precio.
- En el caso de que algún producto no tenga imagen (es decir, que la propiedad `image` sea `undefined`), mostrar una imagen de relleno. Podemos crear una imagen de relleno con el servicio de https://placehold.co/ donde en la propia URL indicamos el tamaño: https://placehold.co/600x400.
- Para pintar la información en la página se puede elegir entre hacerlo usando `innerHTML` o manipulando de forma avanzada el DOM.
- Al hacer clic sobre el botón de **Buscar**, la aplicación debe filtrar los productos por su nombre.

3. Carrito de la compra

Una vez aparecen los resultados de búsqueda, la usuaria puede añadir productos al carrito de la compra. Para ello, al hacer clic sobre el botón de Comprar debe pasar lo siguiente:

- El color de fondo y el de fuente cambian y cambia el texto del botón Comprar por Eliminar, indicando que es un producto añadido al carrito.
- Hay que mostrar un listado en la parte izquierda de la pantalla, debajo del formulario de búsqueda, con los productos añadidos. Os recomendamos crear un variable o constante de tipo array en JavaScript para almacenar los productos del carrito.
- Los productos del carrito de la compra deben mantenerse a la izquierda de la página aunque la usuaria realice búsquedas.

4. Almacenamiento local

Hay que almacenar el carrito de la compra en el localStorage. De esta forma, al recargar la página el carrito de la compra se debe mostrarse.

5. BONUS:

Si te queda tiempo, puedes añadir las siguientes funcionalidades.

1. **Borrar elementos del carrito**: Como bonus, os proponemos la opción de borrar productos del carrito de la compra. Al hacer clic sobre el icono de una 'x' al lado de cada producto, hay que borrar el producto clicado de la lista y del localStorage.
2. **Añadir/quitar del carrito**: Para terminar de rematar nuestra app de compras, nos gustaría poder añadir/quitar del carrito del compra al hacer clic sobre un producto del lado de la derecha. Y que, si realizamos una nueva búsqueda y sale una producto que ya está añadido, aparezca ya resaltada en los resultados de búsqueda (con colores de fondo y texto intercambiados y el botón "Añadir" o "Quitar" según corresponda).
3. **Borrar todos los productos**: Y ya sería fantástico si al final de la lista de la compra hay un botón para borrar todos los productos a la vez.
4. **Actualizar la cantidad de productos en el carrito de la compra.** Al hacer clic sobre el icono de un '+' o un '-' al lado de cada producto, hay que aumentar o disminuir la cantidad del producto clicado en el carrito y en el localStorage.
5. **Afinar la maquetación**: Una vez terminada la parte de interacción, podemos centrarnos en la parte de maquetación donde tenéis libertad para decidir los estilos. En cualquier caso os dejamos una propuesta gráfica.

Entrega

Solo debéis hacer commits y merges en la rama master de vuestro repositorio hasta la fecha límite.

La evaluación solo se considerará terminada cuando:

- Esté publicada en GitHub Pages y funcionando.
- El enlace a GitHub Pages esté en la página página principal del repositorio, en la parte superior, en la sección "About" al lado de la descripción.

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. 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.

Si detectamos que has entregado código que no es tuyo, no entiendes y no lo puedes defender, 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 (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 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.

General

- Usar una estructura adecuada de ficheros y carpetas para un proyecto web, y enlazar bien los distintos ficheros*.
- Tener el código perfectamente indentado*.

JavaScript básico

- Crear código JavaScript con sintaxis correcta y bien estructurado*.
- 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*.

Peticiones a 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.

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 explicando muy brevemente cómo arrancar el proyecto.

¡Al turrón! (o a la rica tortilla de patatas)