



JS Shopping Cart

Noviembre 18, 2019

Resumen

En este proyecto la compañía necesita desarrollar un carrito de la compra usando exclusivamente Javascript. Además, es necesario que se guarde el contenido del carrito en el navegador del cliente, por lo que es indispensable usar localStorage para llevar a cabo la implementación.

Contenidos

Requisitos del Proyecto

Tareas

Incidencias

Lecciones aprendidas

Calendario del Proyecto

Cronograma del Proyecto

Uso de Git

Medidas de Control de Calidad

Métricas de Calidad

Riesgos

Herramientas usadas

Requisitos del Proyecto

1. Debes de usar GIT. Es importante que las indicaciones y los commits sean lo suficientemente explícitos y concretos como para poder entender los cambios sin la necesidad de requerir de información adicional en la medida de lo posible.
2. No puedes hacer uso de Frameworks o librerías de terceros para la parte de Javascript.
3. Documentar todos tus algoritmos.
4. Usar programación orientada a objetos.
5. Trabajar de forma adecuada con los precios en Euros € y Dólares \$ (puedes colocar el valor de conversión entre monedas de forma manual y simular un servicio de un tercero).

Tareas a realizar

1. Leer el documento descriptivo del proyecto:
 - Prioridad: Alta.
 - Descripción: Leer la descripción del proyecto para entender todos los requisitos.
 - Nivel de dificultad: 3/10
 - Estimación de tiempo: 15 min.
2. Crear repositorio en GitHub
 - Prioridad: Alta.
 - Descripción: Crear el repositorio en Github para almacenar todos los contenidos los archivos del proyecto.
 - Nivel de dificultad: 3/10
 - Estimación de tiempo: 10 min.
3. Prototipar:
 - Prioridad: Alta.
 - Descripción: Conceptualizar el proyecto, pensar en cómo se verán las páginas del portal.
 - Nivel de dificultad: 5/10
 - Estimación de tiempo: 1 hr
4. Conceptualización del proyecto
 - Prioridad: Alta.
 - Descripción: Crear las plantillas html a nivel general para cada página del quiz
 - Nivel de dificultad: 4/10
 - Estimación de tiempo: 60 min.
5. Estructura de HTML
 - Prioridad: Alta.
 - Descripción: Crear archivos HTML para la maquetación del portal.
 - Nivel de dificultad: 3/10.
 - Estimación de tiempo:
6. Listado de acciones.
 - Prioridad: Alta.
 - Descripción: Pensar en la lista de acciones que se puede hacer con los botones en la página. Como por ejemplo:
 - Nivel de dificultad: 4/10.
 - Estimación de tiempo: 30 min.
7. Estudio de POO
 - Prioridad: Media.
 - Descripción: Estudio sobre lo básico de cómo trabajar con POO.

- Nivel de dificultad: 6/10.
 - Estimación de tiempo: 2 hr.
8. Creación de Json de productos
- Prioridad: Alta.
 - Descripción: Agregar lista de productos para la página.
 - Nivel de dificultad: 6/10.
 - Estimación de tiempo: 2 hr
9. Cantidad de Producto
- Prioridad: Media.
 - Descripción: Hacer que el usuario pueda elegir la cantidad de ítems que desea comprar.
 - Nivel de dificultad: 8/10.
 - Estimación de tiempo: 4 hr.
10. LocalStorage
- Prioridad: Alta.
 - Descripción: Que el cliente pueda guardar al localStorage todos los ítems que desea comprar.
 - Nivel de dificultad: 8/10.
 - Estimación de tiempo: 4 hr
11. Dinamismo del portal
- Prioridad: Media.
 - Descripción: Crear las funciones necesarias para poder utilizar datos del JSON en la página web
 - Nivel de dificultad: 7/10.
 - Estimación de tiempo: 4 hr.
12. Cambiar de Box
- Prioridad: Media.
 - Descripción: Crear funciones para poder cambiar de un “box” de la página hacia otro, utilizando propiedades como display: none e inline block.
 - Nivel de dificultad: 5/10.
 - Estimación de tiempo: 2 hr.
13. Función para cambiar las cantidades de ítems.
- Prioridad: Alta.
 - Descripción: Crear una función para poder cambiar la cantidad de los ítems a elegir.
 - Nivel de dificultad: 7/10.
 - Estimación de tiempo: 3 hr

14. Función para cambiar tamaños

- Prioridad: Alta.
- Descripción: Crear una función para cambiar el tamaño de los ítems a elegir.
- Nivel de dificultad: 7/10.
- Estimación de tiempo: 2 hr.

15. Función para cambiar moneda

- Prioridad: Media.
- Descripción: Crear una función que permita cambiar de euros a dólares.
- Nivel de dificultad: 6/10.
- Estimación de tiempo: 1 hr.

16. Diseño CSS

- Prioridad: Bajo.
- Descripción: agregarle diseño básico al portal para mejorar su estética.
- Nivel de dificultad: 5/10.
- Estimación de tiempo: 30 min.

17. Preparar presentación

- Prioridad: Alta.
- Descripción: preparar la presentación de los puntos principales del proyecto en base al documento descriptivo del proyecto.
- Nivel de dificultad: 6/10.
- Estimación de tiempo: 1 hr

Cronograma

# Tarea	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
1	X						
2	X						
3	X						
4	X						
5		X					
6		X					
7		X	X	X	X		
8			X				

9			X				
10			X	X			
11				X	X		
12						X	
13						X	
14						X	
15						X	
16						X	X
17							X

Calendario

- **Lunes:**

- Conceptualización y organización del proyecto.

- Hacer esquemas y prototipos en el cuaderno de cómo será el proyecto, que “features” va a incluir, cómo se repartirá el trabajo el resto de los días.

- Escribir la conceptualización y organización del proyecto en la documentación.

- **Martes:**

- Estudio de Programación Orientada a Objetos con JavaScript.

- Maquetación de la página inicial y estilo básico del portal con CSS.

- Documentar el registro de incidencias y lecciones aprendidas del proyecto.

- **Miércoles:**

- Crear las funciones para trasladarte de una página a otra.

-Buscar las etiquetas necesarias

- **Jueves:**

-Trabajar en las funciones necesarias para la creación del sitio web.

-Estudiar más sobre Programación Orientada a Objetos.

- **Viernes:**

-Inicio del proyecto desde cero.

-Reunirme con compañeros para mejor asesoramiento del proyecto.

-Ir anotando aprendizajes e incidencias en la documentación.

- **Sábado:**

-Trabajar en el HTML, maquetación y JavaScript del nuevo proyecto.

-Crear las funciones necesarias para el funcionamiento del proyecto.

- **Domingo:**

-Revisar últimos detalles del proyecto.

-Agregar más estilos al proyecto con CSS.

-Terminar la documentación del proyecto.

Registro de Incidencias

- Dificultad para conceptualizar y preparar los pasos para la creación del proyecto.
- He tenido muchas dificultades con el manejo del estrés, a tal punto que afectaba mi aprendizaje.
- Aún tengo mucha dificultad a la hora de crear funciones, cómo utilizar los loops, etc.
- Tuve que empezar el proyecto de nuevo porque la lógica que había utilizado para la creación del proyecto no fue muy buena.
- A pesar de mis esfuerzos, no hubo una buena organización del proyecto y por ende perdí mucho tiempo.

Lecciones aprendidas

- Esta es mi primera vez utilizando Bootstrap. Nunca lo había usado, y la verdad es que es bastante sencillo, sólo necesito agarrar más experiencia con él.
- Siempre pregúntate, “¿hay una manera más sencilla de hacer esto?”. No te compliques.
- La extensión “highlight matching tag” facilita mucho el trabajo con HTML ya que permite visualizar todo mejor.
- A veces es necesario empezar los proyectos de zero pero con un nuevo enfoque.
- Es bueno reunirse con los compañeros para compartir ideas sobre la conceptualización y organización del proyecto.
- Existen muchas soluciones para resolver un proyecto, cada uno mira todo de una manera muy distinta. Se requiere mucha creatividad para pensar en distintas soluciones para resolver los problemas en un proyecto.
- El concepto de los “box” es muy bueno para trabajar un portal que requiere varias páginas ya que sólo necesitas utilizar un sólo HTML para guardar todas las páginas.

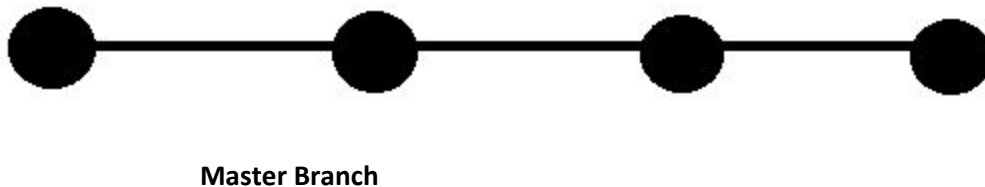
Herramientas utilizadas

- Html para la maquetación, CSS para darle estilo a la página y JavaScript para darle funcionalidad.
- Visual Studio Code: para escribir y editar el código a usar para el proyecto.
- Bootstrap: para facilitar el proceso de darle estilo a la página con CSS.
- Extensiones útiles de VSC que facilitan el trabajo.

Uso de Git

Para este trabajo se utilizó Git de una manera muy sencilla. A medida que se iban realizando cambios al proyecto, se iban guardando los cambios utilizando commits, trabajando sólo con la branch Master.

Como se muestra en la imagen a continuación, sólo se utilizó un Master Branch para hacer todos los cambios al proyecto.



Documentación de riesgos

Es importante tener consciencia de los posibles riesgos a la hora de desarrollar un proyecto. Estos riesgos son capaces de pausar, detener o hasta hacerte tener que reiniciar el proyecto en su totalidad, por lo que debería ser una prioridad tener planes de contingencia en caso de que alguno de los riesgos lleguen a hacerse realidad. Como primer enfoque, la siguiente tabla enumera los posibles riesgos que pueden obstruir el desarrollo del proyecto. Es importante tener en cuenta que la probabilidad sigue una escala de 1 (poco probable) a 3 (muy probable), así como el impacto, que también sigue una escala de 1 (bajo) a 3 (alto). La prioridad se calcula a través del producto de la probabilidad y el impacto:

ID	Riesgo	Consecuencia	Probabilidad	Impacto	Prioridad	Respuesta
1	Perder el ordenador	No se podría trabajar	1	3	3	Mantener el repo actualizado
2	Enfermarse	No se podría trabajar	1	3	3	Cuidar la salud
3	No comprender lo suficiente a tiempo	No podría terminar todo a tiempo	2	3	6	Buscar ayuda de otros
4	No llegar a la meta	Baja calificación	2	3	6	Buscar ayuda de otros

Métricas de Calidad

Para este proyecto se utilizará la validación de páginas de W3S, para mejorar la maquetación del HTML.

El validador de W3S nos muestra lo siguiente:

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Message Filtering

1. **Warning** Empty heading.

From line 42, column 9; to line 42, column 12

```
> <h2></h2>
```

2. **Warning** Empty heading.

From line 59, column 9; to line 59, column 32

```
> <h2 class="text-center"></h2>
```

Estos detalles son difíciles de arreglar ya que estos tags se ponen vacíos para agregarles información con JavaScript.

