

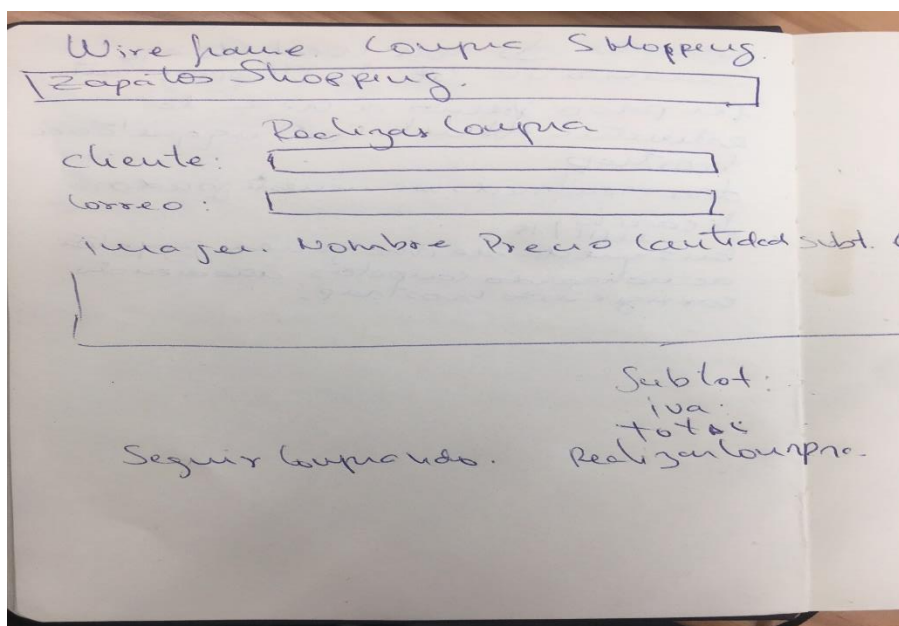
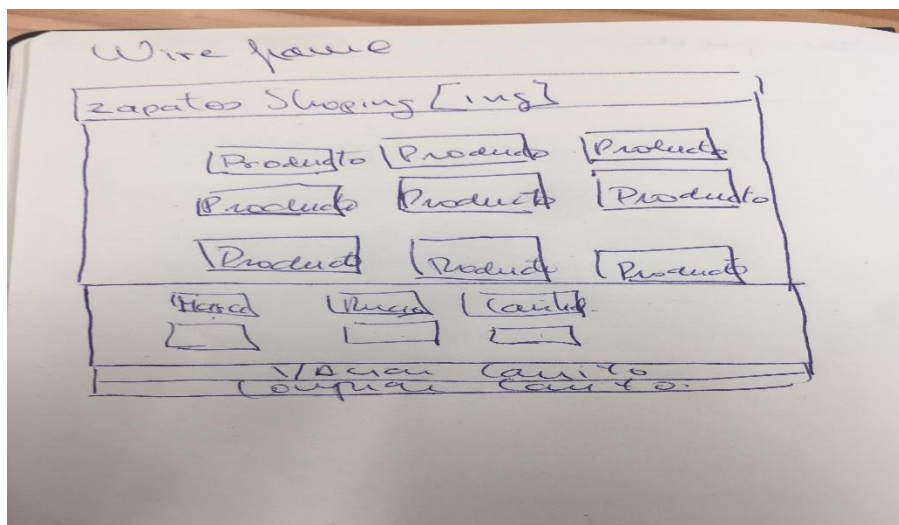
JS Shopping Cart

Objetivo del proyecto

- ✚ Entender cómo funciona un carrito de la compra
- ✚ Mejorar tus conocimientos en JavaScript, HTML y CSS
- ✚ Mejorar tus conocimientos en lógica y programación

ANALISIS Y ORGANIZACIÓN DE PROYECTO:

- ✚ Usamos de referencia la imagen Wireframe



✚ Que acciones a ejecutar son las siguientes

En la página principal del carrito tenemos la siguiente ejecución

Una vez que tengamos las recogidas de datos de JSON en los botones comprar se nos agregue en el carrito los datos recogidos sin tener que añadir más producto al carro.

También tenemos que si no tenemos nada en el carrito que no pase a la siguiente página para procesar la compra.

En la parte del carro tenemos unas clases con métodos para realizar funciones de eliminar agregar al carro también esto incluyendo localStorage guardando, obteniendo, leer y eliminar datos en localStorage.

En la segunda página tenemos la siguiente ejecución.

Estando en esta pagina tenemos que agregar el nombre del cliente y su email recogiendo todos los datos de localStorage que tenía en la parte principal del carrito recalculando el precio.

También tenemos la función de seguir comprando o finalizar compra.

✚ La organización del proyecto se hace de esta manera

Tarea	Prioridad	Horas	Dificultad
Documentación	Alta	4,00	Alta
Organización	Alta	3,00	Alta
Busqueda Previa de informacion	Normal	12,00	Normal
Creacion de repositorio	Baja	0,15	Baja
Documentando Proyecto	Normal	1,00	Normal
Estructura HTML index	Baja	0,30	Baja
Estilos Css	Normal	1,00	Normal
Json Recogidas de datos	Alta	2,00	Alta
Creacion de elementos divs con	Normal	5,00	Alta
Creacion de Clases con su metodos	Alta	8,00	Alta
Creacion de la segunda pagina	Alta	4,00	Normal
Estructura HTML buy.html	Normal	2,00	Normal
Estilos Css	Normal	2,00	Normal
Creacion de metodo llamada a	Alta	3,00	Alta
Creacion de metodo llamada a	Alta	3,00	Alta
Metodo datos de valor	Alta	3,00	Alta
Testing	Normal	1,00	Normal

✚ Diseñar el modelo de clases

Tan solo vamos a realizar una clase con métodos que realice ejecuciones correspondientes

Como llamar los datos JSON

- Leer datos
- Agregar datos
- Obtener datos
- Eliminar datos
- Vaciar Datos

Todos esto también realizando métodos para leer localStorage

- Leer datos localStorage
- Agregar datos localStorage
- Obtener datos localStorage
- Eliminar datos localStorage
- Vaciar datos localStorage

En esta parte también creamos una función para realizar la suma de todos los valores.

Todo el método se guarda en la clase ya creada en este caso **Class Cart**

Estos métodos de guarda en un constructor en JS guardando lo en objeto llamando así los métodos creados

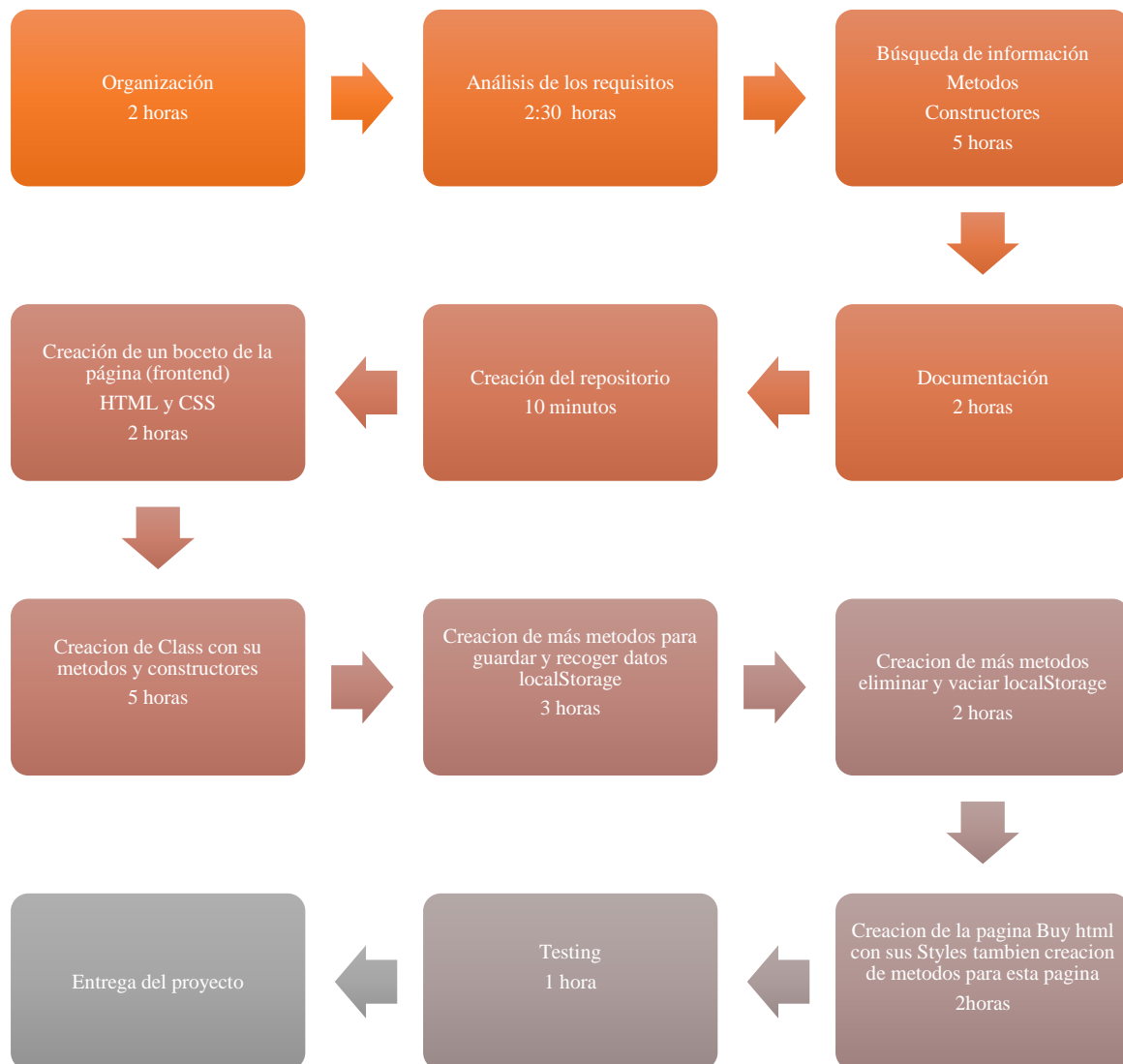
✚ Directorios y ficheros

Se creará en la parte documentos con una carpeta que se llamará [ProyectoJScarritodelaCompra](#)

En la carpeta [ProyectoJScarritodelaCompra](#)

- ✚ Se forma de un index.html y buy.html
- ✚ Se crea una subcarpeta **assets** que vendrá otras subcarpetas CSS, JS, IMG para llevar los datos correspondientes.
- ✚ En la carpeta CSS se creará un style para modificar los estilos básicos ya que usaremos Bootstrap
- ✚ En la carpeta JS se creará distintos ficheros, para el **json**(bd.js) para la creación de los elementos (divs) **métodos**(carrito.js) y la creación del **constructor** (order.js).
- ✚ se creará una carpeta de buy con un fichero JS para así tener personalizado la llamada al objeto creando un constructor correspondiente (process.js) para así separar con el la página buy.html
- ✚ En la carpeta IMG tendremos dos subcarpetas de compra con dos imágenes y otra de zapatos como imagen principal

Seguimiento de calendario



Registro de incidencia

Una lista de lo que fue sucediendo

- Estructura no definida en HTML
- Estilos y maquetación de la pagina
- Ejecución de métodos de la clase
- Petición de los datos localStorage para que se muestre o se elimine
- Ejecución de la petición de los datos de index HTML que vaya a buy.html
- Ejecución de for para la realización del cuenta y aparición de datos.
- Creación de los precios euros y dólares.
- No podrá añadir más cantidades de las que se encuentren disponibles (simular stock)
- Podrá cambiar opciones del producto que pueden influir en el precio e imagen que se me muestra al usuario
- Testing en safari ni internet Explorer no se muestra los divs creados.

Documentación de Riesgos

- Que se mojara la Portátil o que explote mi cabeza.

Documentación acerca del WORKFLOW de git que vas a usar

- Creación Git Hub <https://github.com/robertfox11/JSShoppingCart.git>
- Hacemos commits de la estructura de la página principal.
- Probabilidad de que ocurra 30%
- Impacto en el proyecto 60%
- Posible alternativa (mitigación) Pedir ayuda a compañeros
- Probabilidad de que ocurra 30%
- Impacto en el proyecto 60%
- Posible alternativa (mitigación) Pedir ayuda a compañeros
- No encontrar con facilidad la información relacionada con el proyecto
- Probabilidad de que ocurra 30%
- Impacto en el proyecto 60%
- Posible alternativa (mitigación)
- Pedir ayuda a compañeros

A partir de la realización de la estructura se continuó trabajando solamente en la rama “master”, a través del Workflow “Gitflow”.

Mas información --> <https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow>



Herramientas utilizadas para el proyecto

Son las siguientes

- Sistema operativo Windows
- Navegadores compatibles Chrome, Firefox
- Visual Studio Code

Métricas de calidad

- Se verificó individualmente la calidad a través de las siguientes métricas:
 - Responsive
 - Compatibilidad con los diferentes exploradores
 - Tamaños adecuados de las imágenes
 - Colores acordes a la gama seleccionada.
 - Uniformidad y alineado de cada sección
 - Funcionamiento correcto del JavaScript
 - Indexación correcta a los diferentes enlaces internos y externos

Registro de lecciones aprendidas.

- Aprendiendo más para la realización de estilos CSS con Bootstrap
- Creación de métodos con JS
- Realización de la llamada con addEventListener con el evento DOMContentLoaded
- Creación de elementos div llamando a muchos nodos appendchild