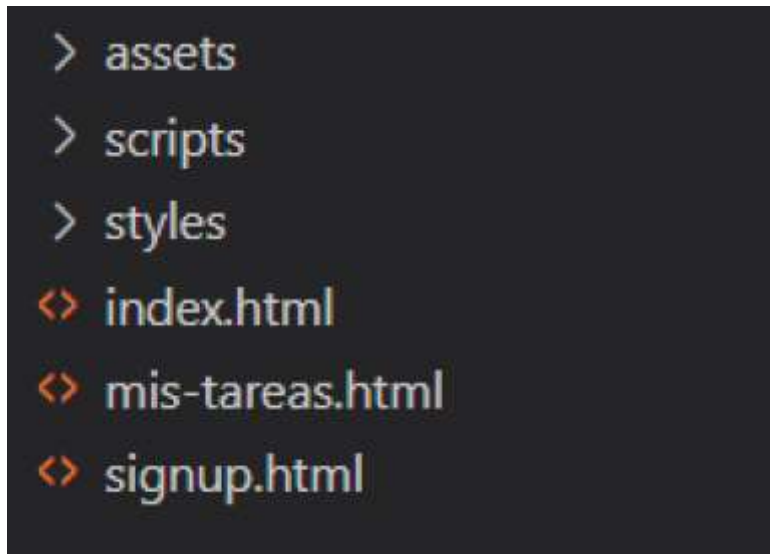


FrontEndII
Profesor: Martín Bonino

Módulo 6. Implementación práctica, animaciones y librerías

Sesión 20. To-Do App: workflow
Abril 19 de 2022

Estructura de la To-Do App:



assets:

Archivos .png y .jpg

scripts:

login.js
signup.js
tasks.js
utils.js

styles:

globalStyles.css
loginSignupStyles.css
tasksStyles.css

(root):

index.html. Los navegadores de manera estándar interpretan al index como la página principal del sitio web.
mis-tareas.html
signup.html

Buenas prácticas

code review: Revisión de código existente

Principio de responsabilidad única

Cada clase, función o módulo debe ser responsable de una tarea específica, y dicha tarea debe estar encapsulada dentro de dicha clase, función o módulo.

Ejemplo:

```
function calcularPromedio(numero1, numero2) {  
    const suma = numero1 + numero2;  
    const promedio = suma / 2;  
    return promedio;  
}
```

//Refactorizando según el principio de responsabilidad única:

```
function sumar(numero1, numero2) {  
    return numero1 + numero2;  
}  
  
function calcularPromedio(numero1, numero2) {  
    const suma = sumar(numero1, numero2);  
    return suma / 2;  
}
```

Keep It Simple Stupid! (KisS)

Cualquier sistema o programa, funciona mejor si se mantiene simple que si se hace complejo. La idea es evitar agregar cualquier capa de complejidad que no sea estrictamente necesaria para el correcto funcionamiento del sistema.

Ejemplo

```
if (nombre === null || nombre === undefined || nombre = '') {  
    console.log('NOMBRE INVÁLIDO');  
} else {  
    console.log(nombre);  
}
```

//Refactorizando:

```
const detalleUsuario = nombre ? nombre : 'NOMBRE INVÁLIDO';  
console.log(detalleUsuario);
```

//Refactorizando de nuevo:

```
const detalleUsuario = nombre || 'NOMBRE INVÁLIDO';  
console.log(detalleusuario); // Javascript evalúa el valor de "nombre"; si es un  
valor de tipo truthy (verdadero), se asigna dicho valor a la variable  
"detalleUsuario"; en caso contrario, se asigna lo que se encuentre a la derecha  
del OR (en este caso, "NOMBRE INVALIDO").
```

Don't repeat yourself (DRY)

Este principio invita a pensar el código de forma tal que se eviten repeticiones innecesarias. Para ello, se debe pensar el código de manera abstracta, con especial énfasis en las funciones que debe cumplir independientemente de un caso concreto. Además, en la medida de lo posible, se debe tener en cuenta la normalización de la información, para evitar redundancia.

Ejemplo:

```
const clientes = ['Juan', 'Pedro', 'Marcelo'];  
const proveedores = ['Esteban', 'Luis', 'José'];
```

```
console.log(clientes[0]);  
console.log(clientes[1]);  
console.log(clientes[2]);
```

```
console.log(proveedores[0]);  
console.log(proveedores[1]);  
console.log(proveedores[2]);
```

// Refactorizando

```
function logItems(array) {  
    array.forEach( item => console.log(item));  
}
```

```
logItems(clientes);  
logItems(proveedores);
```

Sincrónico

Peticiones PUT

- Se puede agregar el id como parámetro en la ruta.
- Las propiedades que cambian se incluyen en el payload. Se puede usar el mismo endpoint para cambiar diferentes propiedades, variando el payload.

Sesión 21. To-Do App: Security

Abril 22 de 2022

Autenticación y Autorización

La autenticación implica verificar la identidad de la persona que accede a una aplicación determinada mediante el uso de sus credenciales. La autorización se ocupa de determinar qué tipo de acciones puede realizar la persona dentro de la aplicación, validando el acceso a los distintos tipos de recursos disponibles.

JSON Web Tokens

JWT es un estándar abierto de codificación, utilizado para transmitir información de manera segura entre dos partes. La información se transmite a través de un objeto JSON, y puede ser verificada ya que el JWT se encuentra firmado digitalmente.

La estructura más simple de un JWT consta de tres partes: Header, Payload y Signature.

Ejemplo: xxxxxx.yyyyyy.zzzzzz

Header:

Contiene la información respecto del tipo de token (JWT), y el algoritmo de encriptación utilizado:

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

Payload:

Contiene la información del usuario pudiendo incluir, por ejemplo, el rol que dicho usuario tiene dentro de la aplicación:

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "admin": true
}
```

Firma:

Es la parte que garantiza la autenticidad de la información incluida en el JWT, permitiendo su verificación.

Cuando una persona inicia sesión en una aplicación determinada, el servidor verifica las credenciales ingresadas y, si son correctos, autentica al usuario dentro de la aplicación, y envía un JWT como respuesta.

El JWT, es entonces almacenado del lado del cliente, y enviado al servidor en cada nueva petición que se realice para acceder a un determinado servicio dentro de la aplicación. Dado que el token contiene la información del usuario (por ejemplo, su rol), el servidor puede acceder a dicha información al recibir la petición, y validar con ello si el usuario se encuentra autorizado a realizarla.

Documentación:

<https://jwt.io/#debugger-io>

Sesión 22. To-Do App: MVP

Abril 21 de 2022

Barra de progreso

<https://codesandbox.io/s/barra-de-carga-8j0pz?file=/index.html>

Ejercicio Pokemons:

<https://github.com/alazzuri/ejercicio-pokemons>

Spinner

Un spinner es un tipo de preloader que puede ser muy útil para un sitio web. Brinda información al usuario sobre el estado de la página al realizar la primera carga, priorizando la interfaz (DOM) y luego la funcionalidad.

Skeleton

Cuando se carga una página, aparece una serie de elementos que no contienen el contenido relevante de la página (que aparecerá segundos después) sino que dichos elementos actúan como una especie de placeholders de la información que la aplicación se encuentra solicitando al servidor.

Esos placeholders son los skeletons. Su uso ha ido ganando popularidad entre los diseñadores y desarrolladores ya que, a diferencia de los spinners y barra de carga, brindan a la persona que accede al sitio, una representación visual del layout y contenido que tendrá la página una vez obtenida la información que se encuentra pendiente de carga. De esta manera, se obtiene una interfaz mucho más amigable y representativa del sitio en su totalidad.

<https://codesandbox.io/s/loading-skeleton-jqskn?file=/styles.css:0-1139>

Ejercicio Skeletons:

<https://github.com/alazzuri/ejercicio-skeletons>

Sesión 23. Evaluación Final
Abril 22 de 2022

Sesión 24. Animaciones
Abril 29 de 2022

Skeleton:

<https://codesandbox.io/s/loading-skeleton-jqskn?file=/styles.css:0-1139>

Sesión 25.

Falta...

Sesión 26. Abril 28 de 2022

Librerías

Algunas librerías para consultar, segun compañeros
kendo, react, storybook

Librerías recomendadas (usadas en la app To-Do):

sweetalert2:

<https://sweetalert2.github.io/#themes>

animate on scroll:

<https://michalsnik.github.io/aos/>

toastify:

<https://apvarun.github.io/toastify-js/>

Otros links:

Métodos HTTP:

[Métodos de petición HTTP - HTTP | MDN](#)

Tutorial de Fetch API:

<https://www.freecodecamp.org/espanol/news/tutorial-de-fetch-api-en-javascript-con-ejemplos-de-js-fetch-post-y-header/>