Persistencia en cliente

Curso 2025/26

Paco Segura

¿Qué es *persistir*?

- Hacer que algo esté disponible de forma permanente entre cambios de contexto.

- Formas:
  - Bases de datos.
  - Almacenamiento en disco: ficheros.
  - Almacenamiento local en una aplicación.

Almacenamiento en cliente

- Cliente = Navegador
- Tres posibilidades:
  - Cookies.
  - Almacenamiento local volátil.
  - Almacenamiento local persistente.

Almacenamiento en cliente

	COOKIES	**LOCAL STORAGE**	**SESSION STORAGE**
Capacidad	4Kb	10mb	5mb
Expira	Fijado manualmente	*"Nunca"*	Al cerrar la pestaña
Localización	Navegador y servidor	Sólo navegador	Sólo navegador
Enviado con peticiones	Si	No	No

Cookies

- Información que recopilamos sobre el usuario y queremos almacenar.

- Se almacenan en Application.

Cookies

- Para crear una cookie empleamos la instrucción document.cookie

- Al crearla es necesario definir un nombre y asignarle un valor:

```
document.cookie = 'Student=John';
```

Name	Value
Student	John

Cookies

- Podemos definir otros atributos, separados por **';'**
  - Expires: para indicar cuando se borrará la cookie.
  - La fecha se indica en formato UTC. Para ello empleamos Date.
  - Ejemplo de una cookie que expire en 30 días:

```
const date = new Date();
let daysToExpire = 30; // Para que la cookie expire en 30 días
date.setTime(date.getTime() + (daysToExpire * 24 * 60 * 60 * 1000));
let expires = "expires=" + date.toUTCString();
let cookie = 'Student=John;' + expires;
document.cookie = cookie; // Student=John;expires=Thu, 20 Nov 2025 16:09:24 GMT
```

Cookies

- Path: para indicar la ruta donde se guardará. Por defecto corresponde a la ruta del documento actual.
- Domain: para indicar el dominio. Por defecto corresponde a la dirección web de la ubicación actual del archivo.
- Otros: max-age, secure, samesite…

Name	Value	Domain	Path	Expires / ...	Size	HttpOnly	Secure	SameSite	Partition K...	Cross Site	Priority
Student	David	127.0.0.1	/	2025-11-2...	12						Medium

Cookies

- Cuando definimos una cookie con un nombre ya indicado, se sobreescribe:

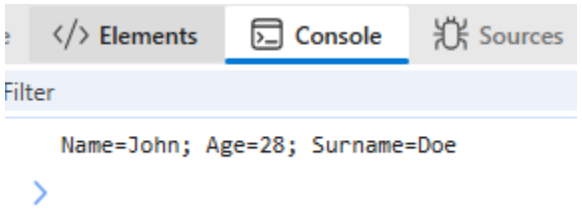
```
document.cookie = 'Student=John;expires=Thu, 20 Nov 2025 16:09:24 GMT';
document.cookie = 'Student=Mary;expires=Thu, 20 Nov 2025 16:09:24 GMT';
document.cookie = 'Student=David;expires=Thu, 20 Nov 2025 16:09:24 GMT';
```

Name	Value
Student	David

Cookies

- Para recuperar las cookies guardadas empleamos la misma instrucción document.cookie:

```
document.cookie = 'Name=John;expires=Thu, 20 Nov 2025 16:09:24 GMT';
document.cookie = 'Surname=Doe;expires=Thu, 20 Nov 2025 16:09:24 GMT';
document.cookie = 'Age=28;expires=Thu, 20 Nov 2025 16:09:24 GMT';
console.log(document.cookie);
```



Name	Value
Age	28
Name	John
Surname	Doe

SessionStorage

- Parte del objeto global “window”.
- A nivel de sesión del navegador (si cerramos el navegador, se descarta la información guardada).
- Persistente entre refrescos (F5) de página.
- API:
  - setItem
  - getItem
  - removeItem
  - clear

- Link

LocalStorage

- También del objeto global “window”.
- Almacenamiento en el navegador.
- Si cerramos el navegador, NO se descarta la información almacenada.
- Persistente entre refrescos de página y entre sesiones del navegador para el mismo sitio web.
- API:
  - setItem
  - getItem
  - removeItem
  - clear

- Link

SessionStorage/ LocalStorage

- Para guardar un único dato en SessionStorage / LocalStorage usamos setItem(*key*, *value*):

```
window.localStorage.setItem('Customer', 'John Doe');
```

- Para obtener un registro guardado en SessionStorage / LocalStorage lo buscamos por su *key*, usando getItem(*key*):

```
let data = window.localStorage.getItem('Customer');
```

SessionStorage/ LocalStorage

- Para guardar varios datos en SessionStorage / LocalStorage, creamos un objeto y lo convertimos a JSON con JSON.stringify(*objeto*):

```
let data = {
   name: 'John',
   surname: 'Doe'
 }
// Convierte Objeto a JSON
window.localStorage.setItem('Customer', JSON.stringify(data));
```

SessionStorage/ LocalStorage

- Para obtener un objeto guardado en SessionStorage / LocalStorage, lo buscamos por su key con getItem(key). Para después trabajar con los datos del objeto, hay que convertirlo de JSON a objeto:

```
let data = window.localStorage.getItem('Customer');
//Convierte JSON a objeto
console.log(JSON.parse(data));
```

SessionStorage/ LocalStorage

- Ejemplo de cómo borrar un único dato en SessionStorage / LocalStorage:

```
window.localStorage.removeItem('Customer');
```

- Ejemplo de cómo borrar todos los datos de SessionStorage / LocalStorage:

```
let data = window.localStorage.clear();
```

Otras tecnologías

- WebSQL
- File Access
- Indexed database

Consideraciones

- *Cache*
  - Si borramos la cache del navegador, en principio se borrará toda la información. No se puede garantizar que un usuario no la borre, y menos en ordenadores que no sean privados.
  - La información de cliente está aislada por sitio.

- Disponibilidad de los motores de persistencia:
  - "if(window.sessionStorage)"

- GDPR:
  - Reglamento General de Protección de Datos / General Data Protection Regulation.
  - Sustituye a la LOPD, desde 25/05/2018
  - Consentimiento, derechos:
    - Portabilidad, olvido, acceso, seguridad (*hackeos*)
  - Nivel europeo.