

# **U5. Almacenamiento en cliente**

Desarrollo Web en Entorno Cliente

# Contenidos

- **Almacenamiento en cliente**
  - ¿Por qué es necesario?
  - Ventajas del almacenamiento en cliente
  - Alternativas para almacenar en cliente
- **Cookies**
- Almacenamiento **en caché**
- Objetos **WebStorage** (almacenamiento web o en DOM)
- **Bases de datos (BBDD):**
  - WebSQL
  - Indexed DB

# Almacenamiento en cliente

## ¿Por qué es necesario?

- Cada vez que nuestro ordenador pide una página a un servidor, éste realiza una conexión nueva e independiente.



- Almacenar información permite crear la **memoria que HTTP no tiene** para:
  1. Crear **sesión** entre páginas
  2. **Personalizar** la web



# Almacenamiento en cliente

- Las principales **ventajas** del almacenamiento en el lado del **cliente** frente al servidor son:
  - **Reduce el volumen de datos** enviados.
  - Conserva los **datos que pertenecen al cliente** en el cliente.
  - Permite trabajar en **modo off-line**.
- La información almacenada en cliente se puede modificar por el cliente.

¿Es buena idea  
almacenar cualquier  
dato en el cliente?



# Almacenamiento en cliente

- **Alternativas para el almacenar en cliente:**

## Cookies

- 50 por dominio y hasta 4KB
- Texto plano (vulnerable a sniffer)

## Caché

- Modo offline
- **Manifest** / Service Workers

## WebStorage

- Mejor rendimiento.
- 5MB y no expira.

## BBDD

- Ventajas uso BBDD (ACID)
- **WebSQL** / IndexedDB

# Cookies

- Las cookies se envían en la cabecera HTML **en cada petición** y contienen:
  - Datos: **propiedad = valor**
  - [Fecha expiración]: **expires = fecha GMT**
    - Sin fecha: se borra cuando se cierra el navegador.
  - [Duración máxima]: **max-age = x segundos.**
  - [Dominio]: **domain=jgaray.es**
    - Sin dominio: aplica al dominio que crea la cookie
  - [Ruta del directorio asociado a la cookie]: **path=**
    - path= / visible para todas las páginas asociadas al dominio.
  - [Seguridad]: **secure** *(si se incluye significa que la cookie sólo se enviará si la conexión es https).*

# Cookies

- Ejemplo cookie:

**//La cookie caducará dentro de 1h**

```
document.cookie = "comidaFavorita=arroz;  
max-age=3600; path=/";
```

**//La cookie caducará el 1/1/2020 a las 12:00**

```
var fechaCaducidad = new Date(2020, 0, 1, 12, 00);
```

```
document.cookie = "colorFavorito=amarillo; expires=" +  
fechaCaducidad.toUTCString(); + ";secure";
```

# Cookies

- **Añadir:** cuando se van creando cookies para distintas propiedades, se van añadiendo a `document.cookie` (no es una propiedad que se sobrescribe).
- **Modificar:** si usamos el mismo nombre de cookie, los datos se sobrescriben.
- **Eliminar:** se borra una cookie modificando la fecha (poner fecha del pasado).

Operaciones con cookies:

[http://www.w3schools.com/js/js\\_cookies.asp](http://www.w3schools.com/js/js_cookies.asp)



# Almacenamiento en caché

- Indicamos que contenidos de los solicitados al servidor se guardarán en cliente para no ser solicitados en sucesivas conexiones. Ventajas:
  - **Reduce el volumen de datos enviados** del servidor (sólo se envían los cambios respecto a lo almacenado en caché).
  - **Aumenta la velocidad de carga** en cliente
  - Es posible trabajar en **modo offline**
- Ejemplo Manifest:  
[http://www.w3schools.com/html/html5\\_app\\_cache.asp](http://www.w3schools.com/html/html5_app_cache.asp)
- Especificación [Service Workers](#) y [ejemplo](#)

# Almacenamiento web

- El almacenamiento web o DOM se realiza localmente en el objeto **window** y puede llegar a **5MB**.
- A diferencia de las cookies:
  - No se envía en cada petición al servidor.
  - No expira
  - Permite ser utilizada en diferentes pestañas y ventanas.
- Existen 2 modalidades: el almacenamiento de sesión o `sessionStorage` y el local o `localStorage`.

# Objetos webStorage

## sessionStorage

- La información se comparte entre páginas del mismo dominio, abiertas en la misma ventana (distintas pestañas).
- Expira con la sesión pero persiste ante caídas.

- Ejemplo:

```
//guardo
```

```
window.sessionStorage.total="75€"
```

```
//consulta
```

```
var t = window.sessionStorage.total;
```

## localStorage

- Permite compartir la información entre ventanas del mismo dominio y protocolo.
- Persiste si cierro la ventana.

- Ejemplo:

```
//guardo
```

```
window.localStorage.ciclo="DAW";
```

```
//consulta
```

```
var ciclo = window.localStorage.ciclo;
```

# Objetos webStorage

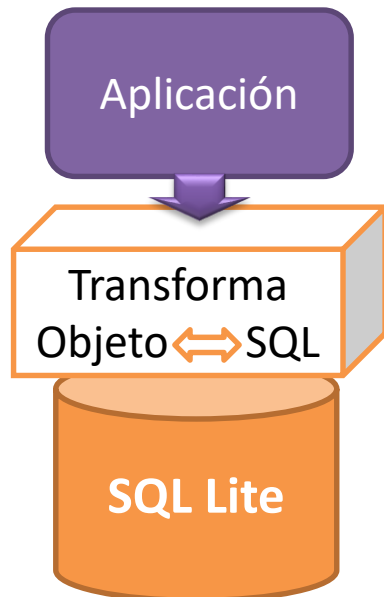
- Los métodos y propiedades asociados a localStorage y sessionStorage son:
  - Almacenar valor de atributo: **setItem**("nombreAtributo", "valor")
  - Obtener valor: **getItem**("nombreAtributo")
  - Borrar elemento: **removeItem**("nombreAtributo")
  - Borrar todos los elementos: **Clear()**
  - **Length**: devuelve el número de atributos almacenados.
- [Ejemplos](#)

# BBDD en cliente

## WebSQL

- Almacena una BBDD SQL en cliente

<https://www.w3.org/TR/webdata-base/>



## Indexed DB

- Almacena una BBDD orientada a objetos en cliente
- <https://www.w3.org/TR/IndexedDB/>

