

DESARROLLO WEB EN ENTORNO CLIENTE

UNIDAD 4 (parte 3/3):

Almacenamiento de datos en el lado cliente

Persistencia de los datos

- Mecanismos de almacenamiento Web del lado del cliente.
- La especificación Web Storage de la W3C.
- Dependiendo el navegador, existirán diferentes estrategias de implantación para persistir los datos.

Persistencia de los datos

- Uno de los pilares de la personalización se encuentra en el concepto de sesión y en la habilidad de almacenar datos del usuario que utiliza el sitio Web en el mismo navegador cliente con el fin de mejorar la experiencia del usuario.
- Si la información pertenece al usuario debe estar en la localización del usuario (es decir, en el navegador cliente). A continuación veremos las diferentes opciones de almacenamiento de datos en los navegadores clientes.

Cookies

- Una `cookie` es información que se almacena en tu equipo cuando visitas un sitio web. Suele ser tus preferencias del sitio o tu estado de inicio de sesión.
- Las cookies surgieron como necesidad ante algunas ausencias tecnológicas del protocolo HTTP con el fin de mantener la información de los carritos de compra virtuales a través de las páginas Web.
- Una cookie es un fichero de texto propio de cada navegador. Por ejemplo, en Firefox, el fichero de las cookies es `cookies.sqlite` y está localizado en un directorio dentro del perfil del navegador:

<https://support.mozilla.org/es/kb/cookies-informacion-que-los-sitios-web-guardan-en->

Cookies

- Una `cookie` es sólo información en texto plano administrable por el mismo usuario y en ningún caso es código fuente interpretable. Parte de la sencillez de las cookies genera inconvenientes:
 - Cada navegador tiene sus propias cookies.
 - Las cookies no diferencian entre usuarios que utilicen el mismo navegador en una misma sesión del sistema operativo. Muchas veces queda almacenada la información de nuestra tarjeta de crédito al realizar una transferencia bancaria.
 - Son vulnerables a los "sniffer" (programas que pueden leer el contenido de peticiones y respuestas HTTP) debido a que se almacenan en texto plano (sin encriptación).
 - Las cookies pueden ser modificadas en el cliente, lo cual podría aprovechar vulnerabilidades del servidor (inyección de SQL).

Utilización de cookies

- Mantener opciones de visualización:
 - Son utilizadas en ocasiones para mantener unas preferencias de visualización.
 - Algunas páginas permiten que el usuario haga una configuración de su página de entrada en el buscador, a través de las cookies el servidor reconoce ciertos aspectos que el usuario configuró y conserva el aspecto.

Utilización de cookies

- Almacenar variables:
 - El servidor puede utilizar las cookies para almacenar variables que se necesiten utilizar en el navegador.
 - Un ejemplo sería una página en la que nos solicitan unos datos, en la siguiente nos solicitan otros datos y así hasta la página final. Los datos de las páginas anteriores se irán almacenando en las cookies hasta que se finaliza el ciclo del formulario y el usuario envía los datos al servidor.
 - Antes del envío al servidor se recuperarán todos los campos del formulario que están guardados en las cookies.
 - Las variables se almacenan con el formato:
"nombreCookie = valorCookie ; expiración ; ruta ; dominio ; seg."

Utilización de cookies

- Realizar un seguimiento de la actividad del usuario:
 - En ocasiones los servidores hacen uso de las *cookies* para almacenar ciertas preferencias y hábitos que el usuario tiene a la hora de navegar.
 - Con esta información, el servidor personaliza sus servicios y publicidad orientándolo a cada cliente en particular.
 - Estos fines no son del todo lícitos si la entidad que realiza estas actividades no avisa al usuario de que está realizando estas acciones.

Utilización de cookies

■ Autenticación:

- Autenticar a los usuarios es uno de los usos más habituales de las cookies.
- A través de las cookies, el navegador guarda los datos del usuario, al realizar una petición al servidor, el navegador envía las cookies junto con la petición.
- Las cookies tienen una fecha de caducidad, cuando pasa un periodo de tiempo establecido, éstas desaparecen (si no la tienen, desaparecen al cerrar el navegador).
- También como mecanismo de seguridad, las aplicaciones Web suelen aplicar un tiempo máximo de inactividad si no se produjo movimiento en la navegación de la aplicación Web, tras el cual las cookies caducan.

Utilización de cookies

- Lectura y escritura de las cookies:
 - Los dos procesos de implementación principales de una cookie son la escritura y la lectura de la misma.
 - A continuación se presentan tres funciones que sirven para:
 - Devolver el valor de una cookie.
 - Escribir una cookie.
 - Comprobar si existe un valor para la cookie.
 - Si se desactivan las cookies del navegador, el código no funciona.

```
document.cookie = "nombreCookie=datosCookie  
    [; expires=horaformatoGMT]  
    [; path=ruta]  
    [; domain=nombreDominio]  
    [; secure]"
```

Utilización de cookies

■ Ejemplo de lectura y escritura de las cookies:

```
<html>
<head>
<script>

function getCookie(c_name) {
    var
i,x,y,ARRcookies=document.cookie.split(";");
    for (i=0;i<ARRcookies.length;i++) {

x=ARRcookies[i].substr(0,ARRcookies[i].index
Of("="));

y=ARRcookies[i].substr(ARRcookies[i].indexOf
("=")+1);
        x=x.replace(/^\s+|\s$/g,"");

        if (x==c_name) {
            return unescape(y);
        }
    }
}
```

```
function setCookie(c_name,value,exdays) {
    var exdate=new Date();
    exdate.setDate(exdate.getDate() + exdays);
    var c_value=escape(value)+((exdays==null) ? "" : ";
expires="+exdate.toUTCString());
    document.cookie=c_name + "=" + c_value;
}
```

```
function checkCookie() {
    var username=getCookie("username");
    if (username!=null && username!="") {
        alert("Bienvenido " + username);
    } else {
        username=prompt("Por favor, Introduzca su
usuario:", "");
        if (username!=null && username!="") {
            setCookie("username",username,365);
        }
    }
}
```

```
</script>
</head>
```

```
<body><input type="button" name="chequeaCookie"
value="Chequear las cookies" onclick="checkCookie();">
</body></html>
```

Web Storage

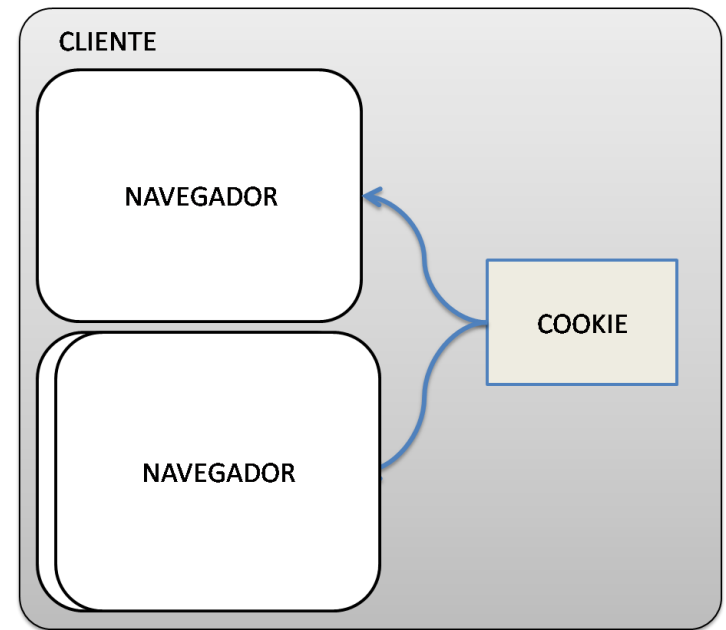
- La especificación actual de Web Storage en HTML5 incluye dos nuevos objetos para el almacenamiento de datos en el cliente: `sessionStorage` y `localStorage`.
- Esta especificación introduce dos mecanismos relacionados para obtener la persistencia de datos de manera estructurada del lado del cliente, similares al de las cookies.
 - El contenido de una cookie es enviado al servidor en cada petición HTTP con tamaño limitado (4 KB).
 - En HTML5 la información solo podrá ser accedida desde el lado del cliente. Mejora de la seguridad.
 - También es posible almacenar gran cantidad de información (más que con las cookies) sin afectar el rendimiento de la aplicación Web.

Web Storage

- **Mejora del rendimiento.** Almacenamiento de datos y envío al servidor por intervalos, en vez de en tiempo real. También al almacenarse datos estáticos que no serán nuevamente obtenidos mediante peticiones al servidor. Además, se almacena por origen (dominio y protocolo), todas las páginas del mismo origen pueden acceder a los mismos datos.
- Con `sessionStorage` no existe relación entre lo almacenado en las diferentes pestañas o ventanas de un mismo navegador. Existe la seguridad de que los datos serán borrados una vez termine la sesión de la ventana que lo ha utilizado.
- Utilización fuera de línea. Con `localStorage` los datos también estarán disponibles entre peticiones y sesiones del navegador. No se pierden datos almacenados entre sesiones.

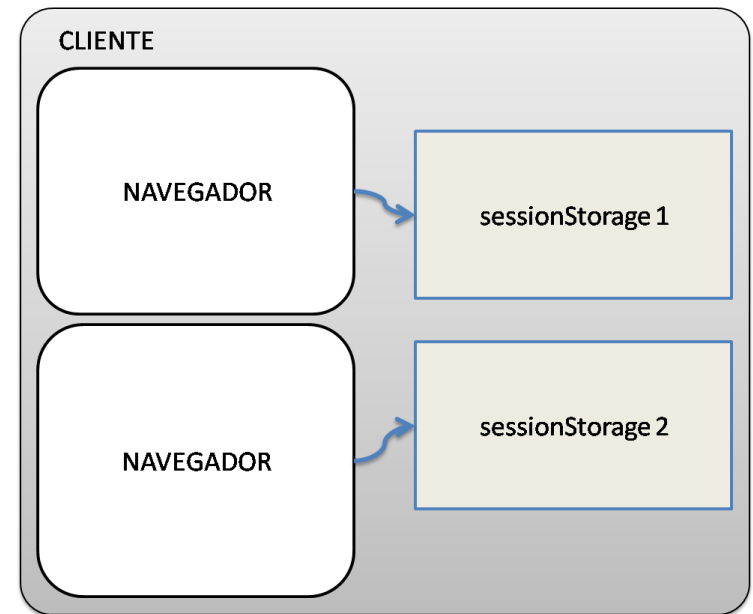
Problemas con las cookies

- En caso de que la aplicación utilice cookies para mantener el estado, existe un problema para realizar múltiples transacciones en diferentes ventanas o pestañas de un mismo navegador si se hace al mismo tiempo.
- Todas las ventanas están asociadas a una misma sesión, por lo que las diferentes instancias del navegador obtendrán información de una misma cookie.



SessionStorage

- El objeto `sessionStorage` se instancia por sesión y ventana, por lo que dos pestañas del navegador abiertas al mismo tiempo y para un mismo sitio Web pueden tener información distinta.
- Al cerrar la sesión se pierde la información. No tiene fecha de expiración.



LocalStorage

- Este objeto se extiende a lo largo de múltiples ventanas y múltiples sesiones (desde siempre).
- Puede ser accedido cada vez que se visita el dominio (los subdominios no son válidos) y todas las sesiones abiertas sobre la misma Web ven la misma información.



- Cualquier tipo de cambio en el almacén debe disparar un evento de tipo `StorageEvent`, de forma que cualquier ventana con acceso al almacén pueda responder al mismo.

Evento StorageEvent

Propiedad	Descripción
url	El dominio asociado con el objeto que ha cambiado.
storageArea	Representa el objeto "localStorage" o "sessionStorage" afectado.
key	La clave del par clave-valor que ha sido agregado, modificado o borrado.
newValue	El nuevo valor asociado con la clave. Será "null" si trata de una eliminación.
oldValue	El antiguo valor.

Métodos de los objetos sessionStorage y localStorage

- Agregar un nuevo par clave-valor:

```
sessionStorage.setItem("maleta", "1");
```

- Obtener el valor en base a la clave:

```
var item = sessionStorage.getItem("maleta");
```

- Eliminar el par clave-valor:

```
var item = sessionStorage.removeItem("maleta");
```

```
var item = sessionStorage.removeItem("1");
```

- El método "clear()" borra todos los elementos de la lista:

```
sessionStorage.clear();
```

Ejercicios

- Realiza todos los ejercicios del bloque 'Ejercicios 4.3' (No entregar)