

# Armado login: cookies

“

Son **archivos** que podemos guardar del lado del **cliente**, es decir, en el **navegador** del usuario.

Las cookies estarán siempre asociadas con el sitio para el que se crearon.



”

# Cookies en Express

A diferencia de la sesión, a las **cookies** les podemos configurar un "tiempo de vida". Es decir, que una cookie dejará de existir cuando **expire** ese tiempo y no cuando el usuario cierre el navegador.

Justamente porque viven del lado del usuario y su información puede leerse, es importante destacar que no debemos almacenar ningún dato sensible en una cookie.

Otro factor a tener en cuenta cuando trabajamos con cookies es que contamos con un límite de espacio (4kb).

“

Para mantener **logueado** a un usuario luego de cerrar el navegador, podemos usar una **cookie** para identificarlo y loguearlo automáticamente la próxima vez que ingrese al sitio.

”



# Implementando cookies

→ Instalamos el módulo `cookie-parser` con npm:

```
>_ npm i cookie-parser
```

- Para **crear** una cookie y **guardar** información en ella, ejecutamos el método `cookie()` sobre el objeto `response`, pasándole dos argumentos:
- ◆ El **nombre** que le queremos asignar a esa cookie.
  - ◆ El **valor** que tendrá.

```
{ } res.cookie('club', 'C. A. Tigre');
```

# Leyendo las cookies

- Para **leer** información de una cookie usamos el objeto `request`, llamando al objeto `cookies`, seguido del nombre de la cookie que definimos anteriormente:

```
{} console.log(req.cookies.club);
```

Cada vez que el cliente hace un pedido a nuestro sitio, envía todas las cookies que estén asociadas con él, por eso es que tiene lógica que las encontremos en el objeto `request`.

DigitalHouse>  
Coding School