



Clase 23. Programación Backend

# ***Cookies, Session y Storage: Parte I***



## ***OBJETIVOS DE LA CLASE***

- Conocer y comprender los conceptos de Cookies, Session y Storage.
- Incorporar de sus usos y aplicaciones.

# ***CRONOGRAMA DEL CURSO***

Clase 22



**Trabajo con datos:  
Normalización**

---

---

---

---

Clase 23



**Cookies, Sesiones,  
storages:  
Parte 1**

Clase 24



**Cookies, Sesiones,  
storages:  
Parte 2**

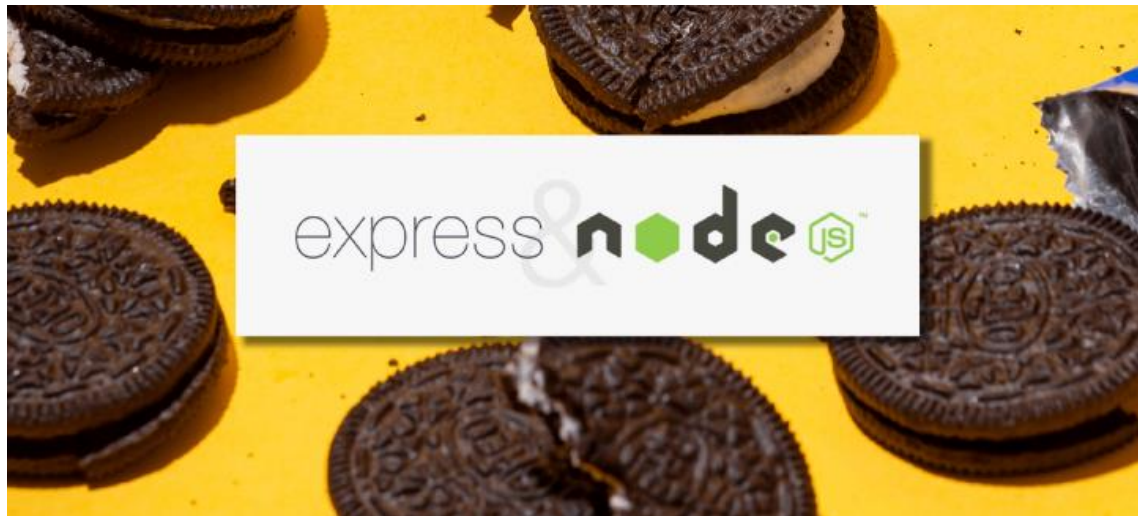
---

***COOKIES***

# ***¿Qué son las Cookies?***



Las Cookies son archivos que podemos guardar del lado del **cliente**, en el navegador del usuario.





# ***Características***

- A las cookies se les puede configurar un **tiempo de vida**. Una vez finalizado el mismo, la cookie se elimina del navegador.
- Al almacenarse del lado del cliente, el espacio con el que se cuenta es limitado, por lo que se recomienda elegir de forma adecuada lo que se vaya a guardar como cookie.
- Hay que recordar que no se deben almacenar datos sensibles en las cookies.

***Vamos al IDE...*** 



***¿Alguna pregunta hasta  
ahora?***



# ***SIGNED COOKIES***



# ***Características***

- A las cookies se les puede agregar un mecanismo de validación que consiste en adjuntar a cada cookie una versión encriptada de su contenido.
- Dicha encriptación se realiza mediante una palabra clave o “secreto” definido del lado del servidor, y desconocido por los clientes.
- El servidor es capaz de verificar si la cookie que se recibe desde el cliente ha sido adulterada o no, chequeando contra la versión encriptada.

# ***cookieParser(secret)***

- **Secret:** string o array de strings que se utiliza para firmar las cookies enviadas, y para analizar las recibidas.
  - Es opcional y, si no se especifica, no firmará ni analizará las cookies recibidas.
  - Si es un string, se utiliza como secret. Si es un array de strings, se firmará la cookie con cada string en el orden provisto (y lo mismo al analizar).



# ***Crear una cookie firmada***

- Para firmar una cookie antes de enviarla al cliente, solo basta con agregar a los dos argumentos usuales (nombre y valor), un tercer argumento de tipo objeto (como se hizo para setear la expiración) con la propiedad “signed” en *true*. `{ signed: true }`
- Las cookies firmadas recibidas, que hayan pasado la verificación de su firma, ya no se encontrarán en *req.cookies*, sino que aparecerán en ***req.signedCookies***. Aquellas que no hayan pasado la verificación, no aparecerán, como si no existieran.



***¿Alguna pregunta hasta  
ahora?***

***Vamos al IDE...*** 



# ***INYECTAR COOKIES EN FRONTEND***

*Tiempo: 10 minutos*



# ***Injectar cookies en frontend***

Realizar un programa de backend que permita gestionar cookies desde el frontend. Para ello:

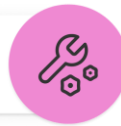
Definir una ruta “cookies”.

Definir un método POST que reciba un objeto con el nombre de la cookie, su valor y el tiempo de duración en segundos, y que genere y guarde dicha cookie.

Definir un método GET que devuelva todas las cookies presentes.

Definir un método DELETE que reciba el nombre de una cookie por parámetro de ruta, y la elimine.





# ***Injectar cookies en frontend***

**NOTA 1:** Utilizar la librería express como estructura de servidor.

**NOTA 2:** Si algún parámetro recibido es inválido, o directamente inexistente, el servidor devolverá un objeto de error.

Ej: { error: 'falta nombre ó valor' } o { error: 'nombre no encontrado' }. Si todo sale bien, devolver el objeto { proceso: 'ok' }.

**NOTA 3:** Si el tiempo no está presente, generar una cookie sin tiempo de expiración.

**NOTA 4:** Generar los request con varios navegadores (Chrome, edge, Firefox) para simular los distintos clientes en forma local.

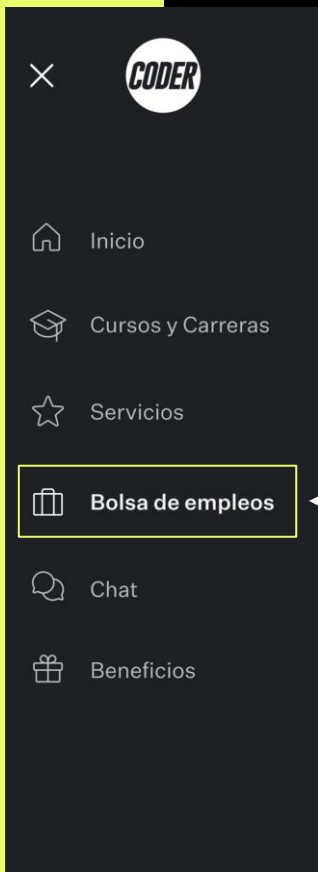


***¿Alguna pregunta hasta  
ahora?***



***BREAK***

**¡5/10 MINUTOS Y VOLVEMOS!**



Nuevo

# ¡Lanzamos la Bolsa de Empleos!

Un espacio para seguir **potenciando tu carrera** y que tengas más **oportunidades de inserción laboral**.

Podrás encontrar la **Bolsa de Empleos** en el menú izquierdo de la plataforma.

Te invitamos a conocerla y ¡postularte a tu futuro trabajo!

Conócela

# ***SESSION MEMORY***

# *¿Qué es Session?*



Session es un paquete de Node, el cual permite que una variable sea accesible desde cualquier lugar del sitio. Se almacena del lado del **servidor**.

Login

Username/Email

Password

Login



# ***Características***

- La información que se quiera guardar en **session** se almacena del lado del servidor.
- Del lado del cliente, se crea un identificador único para poder acceder a esa información desde el navegador.
- Los datos almacenados en **session** se borran al cerrar la ventana del navegador.
- Se utiliza principalmente para guardar los datos de usuario al iniciar sesión.

***Vamos al IDE...*** 





***¿Alguna pregunta hasta  
ahora?***



# ***SESIONES DE USUARIO EN SERVER***

*Tiempo: 15 minutos*

# ***Sesiones de usuario en server***

Desafío  
generico



Realizar un programa de backend que establezca sesiones de usuarios en el servidor.

Cuando un cliente visita el sitio por primera vez en la ruta 'root', se presentará el mensaje de “Te damos la bienvenida”.

Con los siguientes request de ese mismo usuario, deberá aparecer el número de visitas efectuadas. El cliente podrá ingresar por query params el nombre, en cuyo caso se añadirá a los mensajes devuelto.

Por ejemplo: “Bienvenido Juan” o “Juan visitaste la página 3 veces”. Ese nombre sólo se almacenará la primera vez que el cliente visite el sitio.

# ***Sesiones de usuario en server***

Desafío  
generico



Se dispondrá de una ruta 'olvidar' que permita reiniciar el proceso de visitas para el usuario.

En caso de que no haya error, se retornará el mensaje “Hasta luego” más el nombre del cliente (de existir); caso contrario un objeto con el siguiente formato: { error : descripción }.

Luego de requerir esa ruta, el efecto será como el de visitar el sitio por primera vez.

**NOTA1:** Utilizar el middleware express como estructura de servidor.

**NOTA2:** Generar los request con varios navegadores (Chrome, edge, Firefox) para simular los distintos clientes en forma local.

***¿PREGUNTAS?***





# ***¡MUCHAS GRACIAS!***

Resumen de lo visto en clase hoy:

- Cookies
  - Session Memory
- 



***OPINA Y VALORA ESTA CLASE***

***#DEMOCRATIZANDOLAEDUCACIÓN***