

# ¡Javascript!

guayerd beta hub

# Agenda del día



01

## Introducción

Presentación Unidad 3

02

## Almacenamiento

Local Storage:

- getItem
- setItem
- removeItem
- clear

Dev Tools

## JSON

Stringify.  
Parse.

## Sesiones

Páginas públicas.  
Páginas privadas.

03

## Ejercitación

Login y crear sesión.



# daily

¿Cómo venimos?

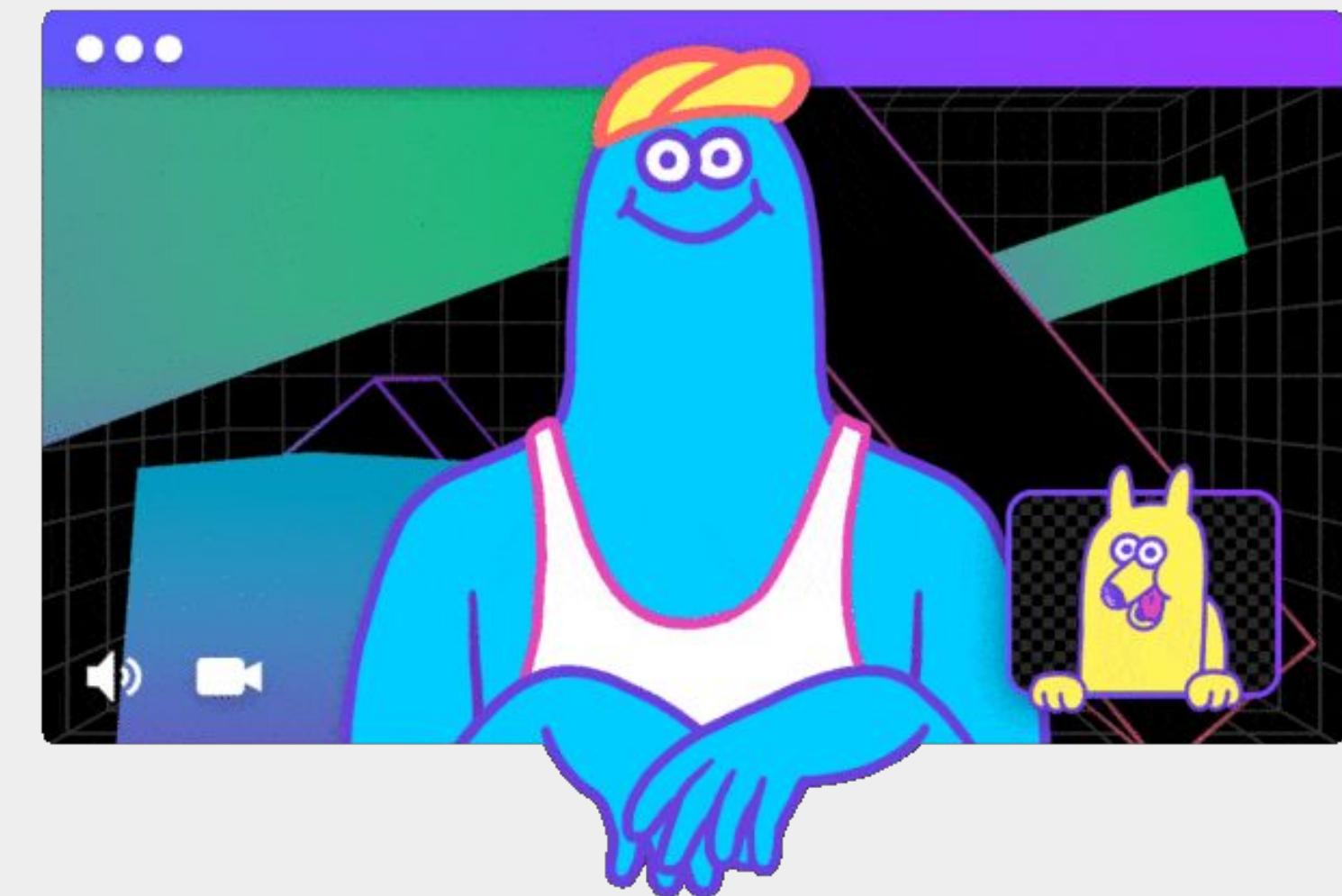
¿Algo nos bloquea?

¿Cómo seguimos?



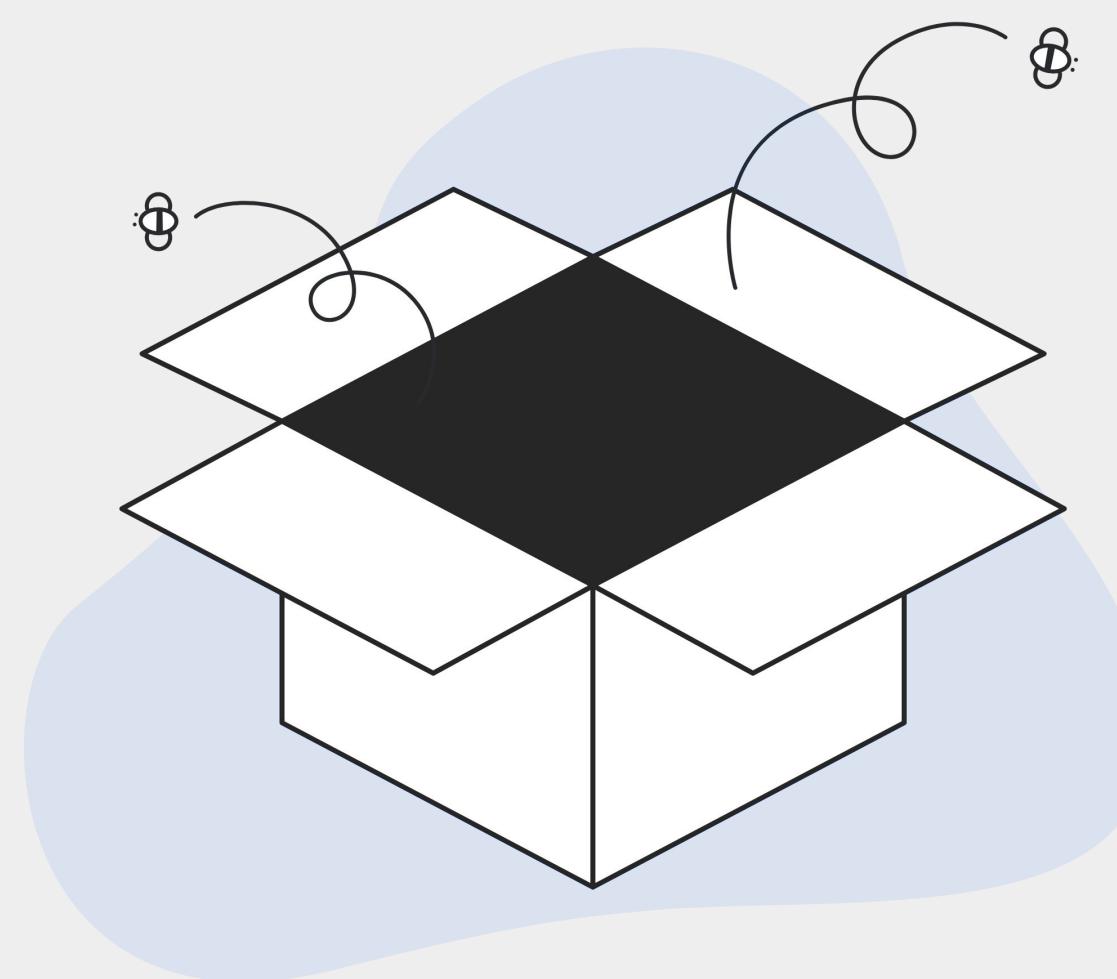
# Almacenamiento local

LocalStorage



SPRINT 3

# LocalStorage



**LocalStorage es parte de la API Web Storage del [navegador](#).**

**Nos permite guardar variables con valores de tipo [string](#), persistentes.**

**Por más que cambiemos de páginas en nuestra aplicación o hasta cerremos el navegador, no se destruirán.**

# Creación de un ítem

Usamos el método `setItem` de `localStorage`. El mismo recibe 2 argumentos: Un string con la clave y un string con el valor.



# Obtención del ítem

Usamos el método getItem de localStorage. El mismo recibe 1 argumento:  
Un string con la clave.

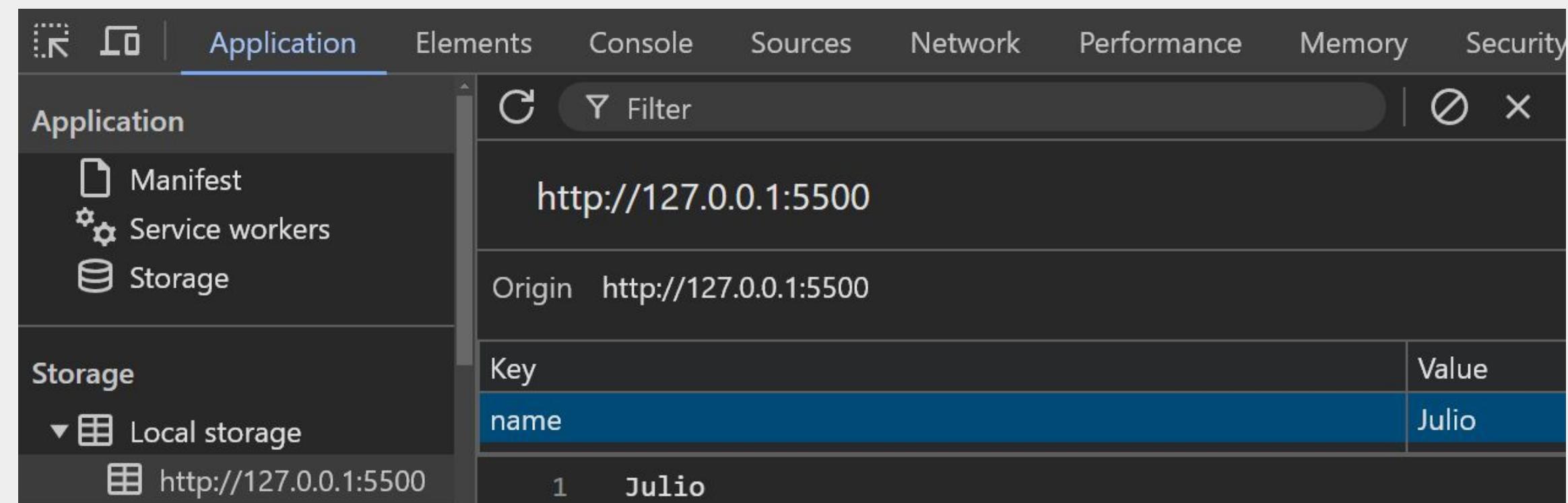
```
localStorage.setItem("name", "Julio");
let name = localStorage.getItem("name");
console.log(name);
```

---

Julio

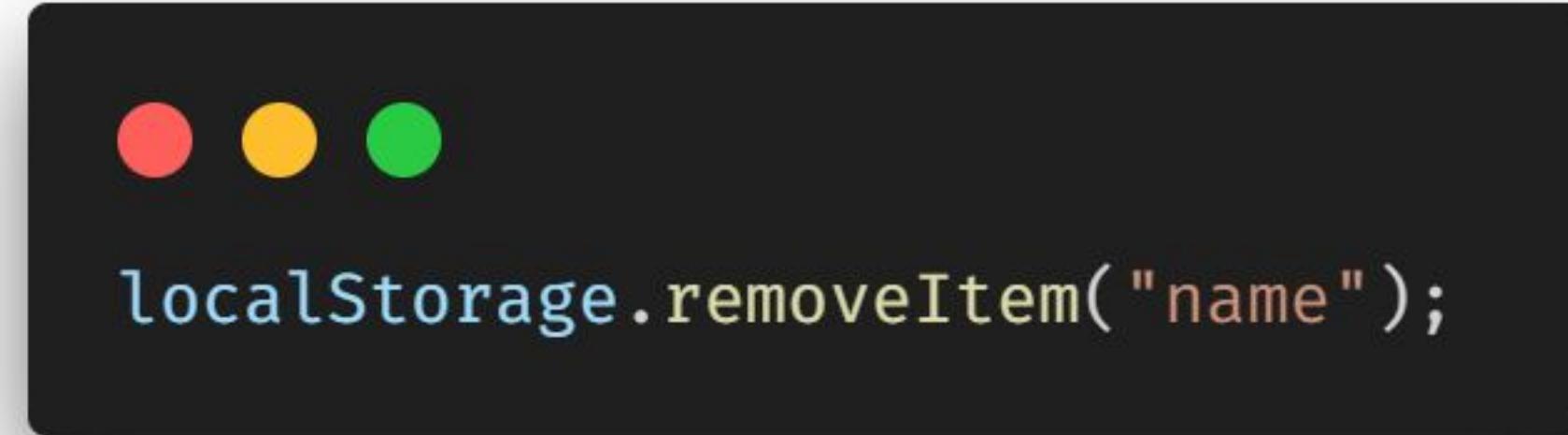
# Ver el ítem en el navegador

Vamos a Dev Tools, a la pestaña Aplicación. En la columna izquierda, en Storage (Almacenamiento), expandimos Local Storage y seleccionamos la URL.



# Eliminar un ítem

Usamos el método `removeItem` de `localStorage`. El mismo recibe 1 argumento:  
Un string con la clave.



# Eliminar todos los ítems

Usamos el método clear de localStorage. Sin argumentos. Es útil para cerrar sesión.



```
localStorage.clear();
```

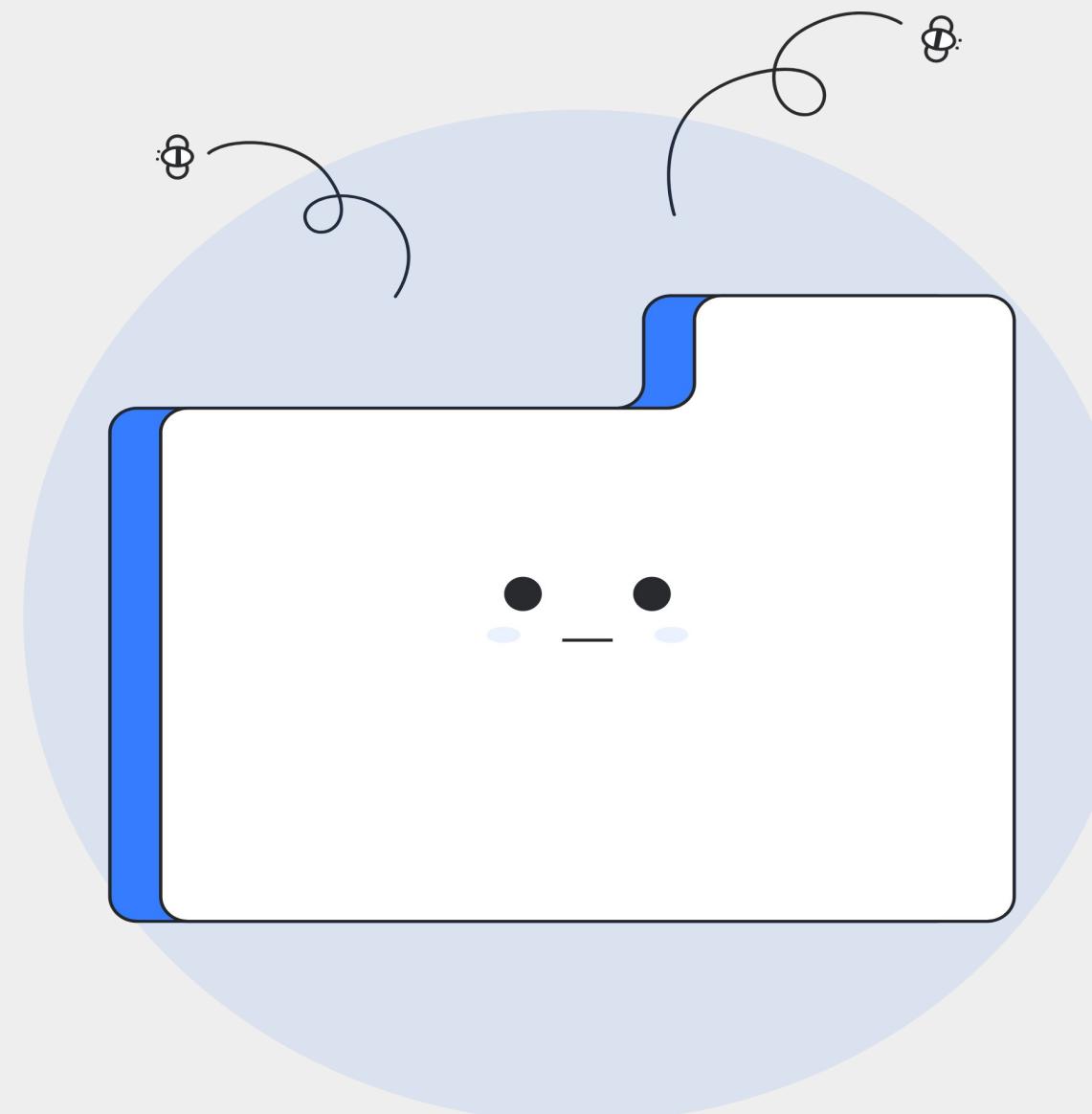
# JSON

Yeah Yeah Yeahs



# SPRINT 3

# JSON



**JavaScript Object Notation** es un formato de texto liviano de intercambio de datos.

Permite comunicar sistemas distintos y, a la vez, ser entendido por personitas. 

Con JSON podremos guardar objetos en **localStorage**.

# JSON.stringify

Usamos este método para convertir un objeto o array Javascript en un string compatible con localStorage.

```
const obj = { name: "Ana" };
const objJSONizado = JSON.stringify(obj);
console.log(typeof objJSONizado, objJSONizado);

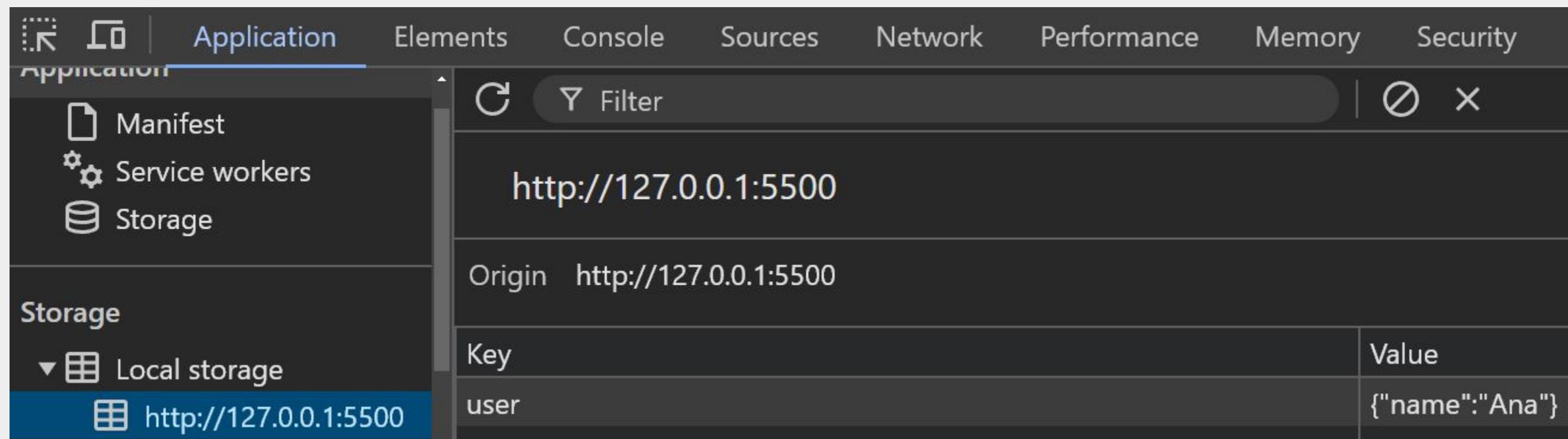
string {"name": "Ana"}
```

# JSON.stringify y localStorage



```
const obj = { name: "Ana" };
const objJSONizado = JSON.stringify(obj);
localStorage.setItem("user", objJSONizado);
```

# JSON.stringify y localStorage



# JSON.parse

Permite reconvertir un string JSON en un objeto o array Javascript.

```
console.log(JSON.parse(localStorage.getItem("user")));  
▼ {name: 'Ana'} ⓘ  
  name: "Ana"  
▶ [[Prototype]]: Object
```

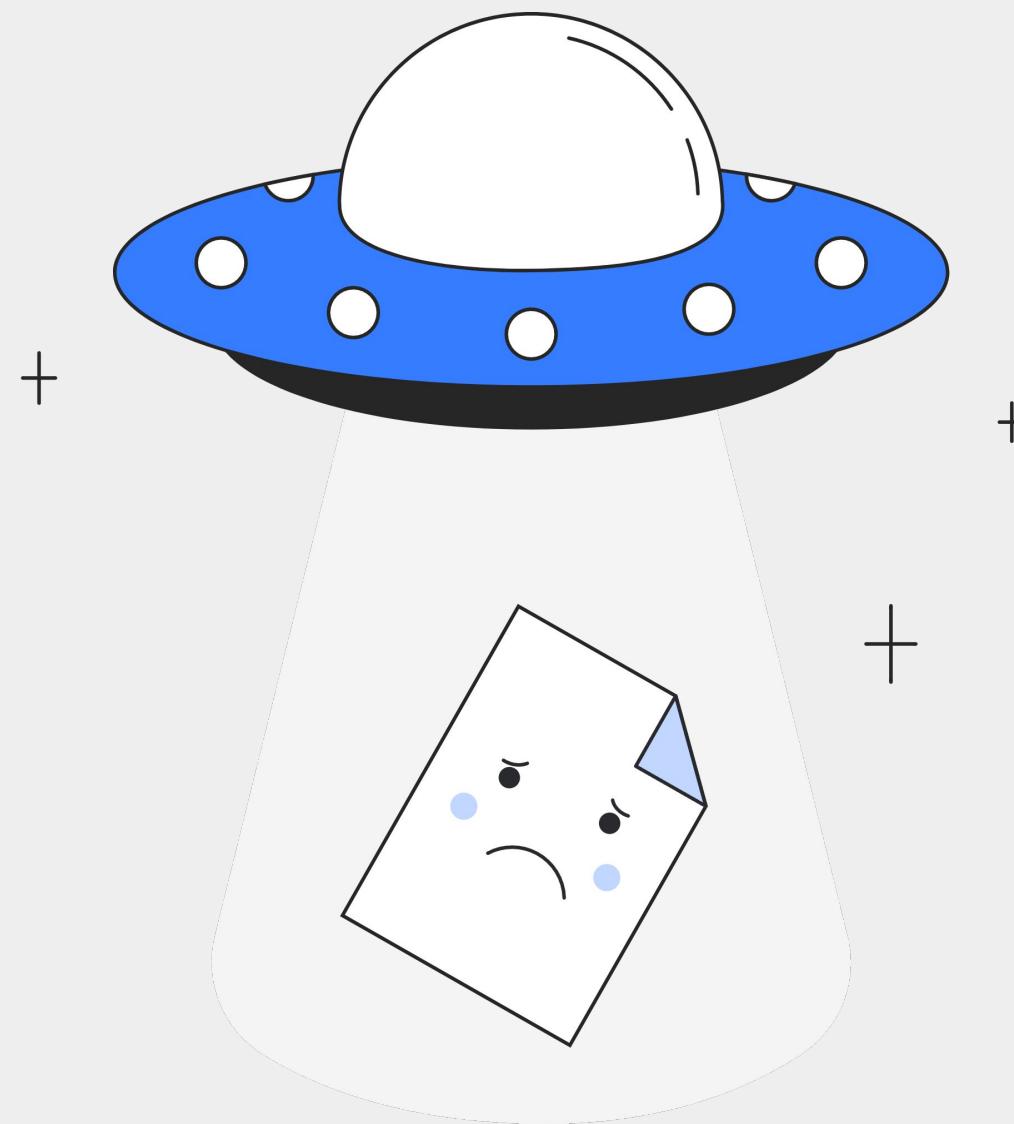
# Sesiones

Oh, tengo páginas privadas!



## SPRINT 3

# Sesiones



Una **sesión** en una App toma datos del usuario. Luego, estos permiten al usuario acceder a contenidos privados.

Una forma común de iniciar sesión es a través de un formulario de **login**.

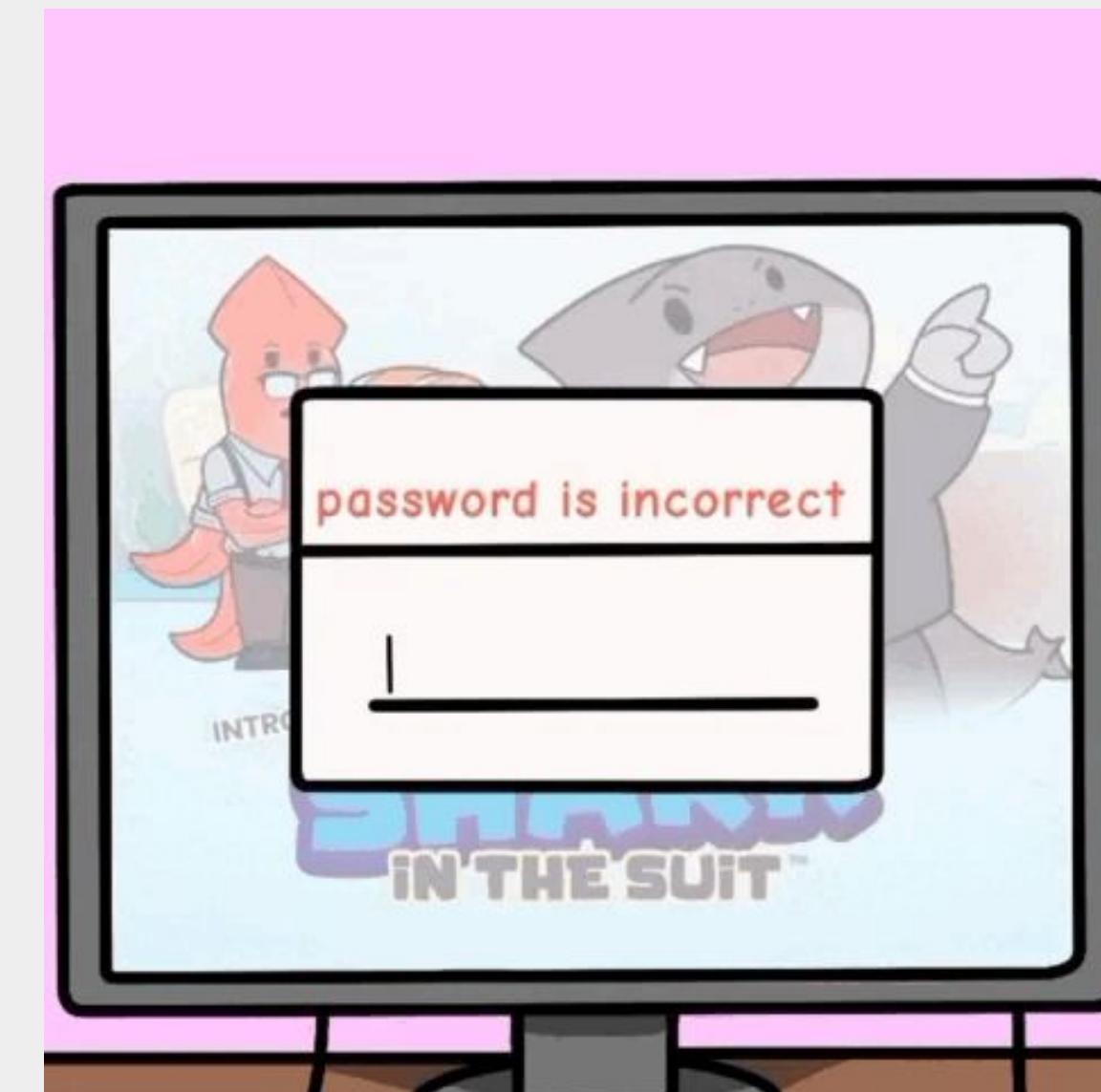
# Página pública

Cualquier ente puede acceder a la misma si tiene la URL.



# Página o contenido privado

Se necesitan permisos de sesión para acceder.



# Redirección de página



```
window.location.href = "./index.html";
```

guayerd





¡Manos a la obra!

**SPRINT 3**

# Login

Entregable



1. Descargar y poner en tu app el login:  
[https://github.com/julioavantt/guayerd\\_login/blob/main/login.html](https://github.com/julioavantt/guayerd_login/blob/main/login.html)
2. Predefinimos los valores correctos de cada campo (input).
3. Mediante eventos obtenemos los valores dados por el usuario.
4. Los comparamos con los valores preestablecidos.
5. Si son iguales redirigimos al index.html y creamos la sesión con localStorage guardando el email.
6. Si son distintos borramos el texto del form y lanzamos un mensaje al usuario.

SPRINT 3

# Logged

Entregable

En la página de login hacer un script con una condición en la que si estás logueado te redirija al index.html



# 🔍 retro

¿Cómo nos fué?

¿Qué cosas no quedaron claras y  
necesitamos repasar la próxima?

