

# MIND HUB.

An aerial night photograph of a city, featuring a multi-lane highway with prominent light trails from vehicles. The city lights are visible in the background, and the overall image has a blue color cast. Decorative elements include pink L-shaped brackets at the top center, white dashed lines at the top right, and white L-shaped brackets at the bottom right.

# Local Storage

# Índice

1 Problemática

2 Definición

3 Métodos



# Local Storage

Como pudimos ver a lo largo de este módulo, utilizando **html, css y JavaScript** desarrollamos aplicaciones de múltiples páginas.

Con el desarrollo de este tipo de aplicaciones, observamos que en el momento que el usuarios navega entre sus páginas las mismas **son recargadas**, junto con la recarga del html al archivo js también le ocurre lo mismo y por consiguiente las variables que este posee declaradas se reinician a su estado inicial de declaración.

Entonces ante esta situación **¿cómo podemos hacer para que la información pueda sobrevivir a esta recarga y ser utilizada en otra página de la aplicación?**

# └localStorage

**localStorage** es un espacio de almacenamiento en la memoria de nuestro navegador, vinculado a la url que lo creó y persiste en este hasta que es eliminado por el usuario mediante un método o manualmente.

Este fue incorporado en HTML5 por la necesidad de evolución de páginas web a aplicaciones web

Para ser almacenado utiliza un modelo de clave valor, muy similar a una variable, y en él podemos almacenar cualquier tipo de dato soportado por JS (string, number, boolean, object, array)

**Dentro del espacio de memoria de nuestro navegador web, contamos con diferentes herramientas, como son también sessionStorage (perdura mientras la ventana esté abierta) o cookies (entre otras cosas podemos incorporar un tiempo de duración)**

# └ Métodos

Para utilizar **localStorage** contamos con una serie de métodos que nos permiten, crearlo, modificarlo, consultarlo o eliminarlo.

	Método	Descripción
<i>Crear</i>	<code>localStorage.setItem("clave",valor)</code>	Guarda un elemento en el local Storage, asignándole una clave,
<i>Obtener</i>	<code>localStorage.getItem("clave")</code>	Busca en el local Storage el elemento cuya clave coincida con el parametro enviado al método
<i>Eliminar</i>	<code>localStorage.removeItem("clave")</code>	Elimina del local Storage el elemento cuya clave coincida con el parametro enviado al método
<i>Eliminar Todo</i>	<code>localStorage.clear()</code>	Elimina todo lo almacenado en el local Storage

¡Muchas gracias!

MIND  
HUB.