




Certified Tech  
Developer

The Ultimate Degree

## Normalizando datos

Hasta ahora, hemos visto de qué manera **JavaScript** nos permite acceder a la información ingresada por las personas en un formulario. Pero ahora, es momento de que vayamos un paso más allá, para descubrir de qué manera esos datos pueden ser manipulados para adecuarse a las necesidades de nuestra aplicación.

Crear una aplicación es un proceso que requiere diversos tipos de análisis, cada uno de los cuales está enfocado en una problemática específica: el diseño, la interacción del usuario con la aplicación, las tecnologías que se van a utilizar, entre otras.

En esta oportunidad, nos enfocaremos en una de dichas problemáticas, la cual resulta relevante para poder construir una aplicación sólida y escalable: **la manipulación de datos.** 

Pero, ¿qué entendemos por manipulación de datos? Como ya sabemos, toda aplicación se construye a partir de datos. Algunos de ellos, se encuentran previamente almacenados en nuestra aplicación y/o en una base de datos; otros, en cambio, son recolectados a medida que las personas interactúan con dicha aplicación.

Pensemos en una aplicación como **Facebook** o **Twitter**. Si queremos acceder por primera vez a alguna de ellas, nuestro primer paso será completar el registro correspondiente. En ese momento, deberemos completar un formulario con una serie de datos, los que serán luego recolectados por dichas aplicaciones y almacenados en su base de datos.

Ahora bien, para poder facilitar las tareas de recolección, almacenamiento y consulta de dichos datos, es necesario que los mismos sean almacenados en un **formato común**. Dicho de otra manera, nuestra base de datos deberá definir un formato específico con el cual deberán ser almacenados los datos ingresados por las distintas personas al momento de registrarse (por ejemplo, puede establecerse que todos los campos de texto sean almacenados en minúscula, sin espacios, puntos ni algún tipo de caracteres especiales).

Pero, pensemos por un segundo la cantidad de personas que se registran en Facebook y Twitter a cada momento. ¿Cómo asegurarnos de que los datos ingresados por todas ellas respeten los criterios establecidos para nuestra base de datos?

Para responder esa pregunta, debemos introducir el concepto de **normalización de datos**.

¿De qué se trata este concepto? Bueno, la normalización de datos no es otra cosa que una serie de procesos, reglas o mecanismos que se utilizan para dar un formato común a los datos recolectados en una aplicación, independientemente de quién sea la persona que lo haya ingresado o la manera en lo que lo haya hecho.

Dicho proceso, puede incluir desde instrucciones que se brindan al usuario cuando se le pide que complete un determinado campo (por ejemplo: *"Ingresa tu número de teléfono sin el 15"*), hasta validaciones y manipulación de los datos recolectados, las que pueden realizarse tanto del lado del cliente como del servidor.

Y, entonces, ¿qué métodos podemos utilizar para manipular estos datos? Veamos un ejemplo: Imaginemos que tenemos un input solicitando a la persona que introduzca sus películas favoritas, separadas por punto y coma.



### Películas favoritas

Ingresa tus pelis favoritas, separadas por punto y coma

Al capturar el input correspondiente, obtendremos el siguiente resultado:

```
const peliculasFavoritas = document.querySelector('#input-peliculas');  
  
console.log(peliculasFavoritas)  
// "Harry Potter;Mi Villano Favorito;Avatar"
```

Ahora bien, supongamos que para poder almacenar dicha información en nuestra base de datos, es necesario que **“normalicemos”** la misma, convirtiéndola en un **array** de películas.

¿De qué manera podemos realizar esta tarea? Por suerte, JavaScript nos provee con el método [split\(\)](#). Este método nos permite dividir los caracteres de un string sobre la base del criterio que deseemos, obteniendo como resultado un **array** que contiene cada uno de los substrings generados. Entonces, volviendo a nuestro array de películas, podríamos utilizar **split** para normalizar la información recolectada:

```
const peliculasNormalizadas = peliculasFavoritas.split(';');  
  
console.log(peliculasNormalizadas)  
// [ 'Harry Potter', 'Mi Villano Favorito', 'Avatar' ]
```

¡Excelente! Ahora nuestra información se encuentra normalizada y lista para ser almacenada. Quizás a esta altura, nos estemos preguntando si este es el único método con el que contamos para normalizar la información. ¡Claro que no! En el próximo video, te mostraremos algunos otros que serán de mucha utilidad para poder llevar adelante esta tarea.