

API REST

CFP

Programador full-stack

Bajas

Pensemos...

Qué vamos a tener en el controlador y en el servicio?

- **[TBC]**

Agregamos endpoints

- Qué verbo se usaba en una API REST para borrar un recurso?
- A que URL responde?
- Qué debería hacer ese end point?
- Donde se programa cada parte?

Controller

```
@Delete('/:index')
```

```
public deleteProducto(@Param('index') index): boolean  
{  
    return this.productoService  
        .deleteProducto(parseInt(index));  
}
```

Service

```
public deleteProducto(position: number): boolean {  
    let removed = this.listaProductos.splice(position, 1);  
    return removed.length == 1;  
}
```

Remueve 1 elemento
desde la posición dada

Splice

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/splice



MDN web docs

moz://a

Technologies ▾

References & Guides ▾

Feedback ▾

Array.prototype.splice()

Jump to: [Syntax](#) [Description](#) [Examples](#) [Specifications](#) [Browser compatibility](#) [See also](#)

[Web technology for developers](#) > [JavaScript](#) >

[JavaScript reference](#) >

[Standard built-in objects](#) > [Array](#) >

[Array.prototype.splice\(\)](#)

Related Topics

[Standard built-in objects](#)

[Array](#)

[Properties](#)

[Array.length](#)

[Array.prototype](#)

The **splice()** method changes the contents of an array by removing or replacing existing elements and/or adding new elements [in place](#).



JavaScript Demo: Array.splice()

```
1 var months = ['Jan', 'March', 'April', 'June'];
2 months.splice(1, 0, 'Feb');
3 // inserts at index 1
4 console.log(months);
5 // expected output: Array ['Jan', 'Feb', 'March', 'April', 'June']
6
7 months.splice(4, 1, 'May');
8 // replaces 1 element at index 4
9 console.log(months);
10 // expected output: Array ['Jan', 'Feb', 'March', 'April', 'May']
11
```

Ejercicio: Qué falta?

Hacer que luego de borrar en el arreglo el archivo quede consistente

TIP: se puede volver a escribir todo el archivo

CFP

Programador full-stack

Borrado de un recurso (DELETE)

Borrado de Tarea

Para enviar el borrado desde el front-end:

- Se usa fetch, similar al POST pero cambiando la ruta
- Necesito poner muchos botones
 - Que cada boton sepa de que elemento es
 - Asignarle el click (el mismo a todos)
 - Que el click pueda diferenciar que botón fue

Necesito “i”, así que tengo que usar un for clásico

Genero un boton “borrar” por cada elemento, marcando cual “i” es

```
function mostrarTablaCompras() {
  html = "";
  for (let i = 0; i < compras.length; i++) {
    let r = compras[i];
    html += `
      ... <td>${r.precio}</td>
          <td><button class="btn-delete-producto"
              pos="${i}">Borrar</button></td>
          </tr> `; //la comilla es el acento inverso
  }

  document.querySelector("#tblCompras").innerHTML = html;

  let botonesBorrar = document
    .querySelectorAll(".btn-delete-producto");
  botonesBorrar.forEach(e => {
    e.addEventListener("click", btnBorrarClick);
  });
}
```

Asigno evento de click

Borrado de Tarea

Uso "this" para saber que boton disparo el borrado

```
async function btnBorrarClick() {  
  let pos = this.getAttribute("pos");  
  let response = await fetch(`/productos/${pos}`, {  
    "method": "DELETE",  
    "headers": {  
      "Content-Type": "application/json"  
    }  
  })  
  load();  
}
```

Resultado

Titulo nuevo

Ejemplo de servidor estático.

<input type="text"/>	<input type="text"/>	<input type="button" value="Agregar"/>	<input type="button" value="Sumar"/>
	Producto Precio		
producto_0	45	<input type="button" value="Borrar"/>	
mariel's demo	500000	<input type="button" value="Borrar"/>	
mariel's demo	500000	<input type="button" value="Borrar"/>	
ma333	5000	<input type="button" value="Borrar"/>	
noticia	1	<input type="button" value="Borrar"/>	