

CFP

Programador

full-stack

Modificaciones

Lógica

- Similar al **POST**
- En lugar de **crear** un producto, vamos a **modificar** uno que ya existe.
- Vamos a necesitar el **:ID** del producto a modificar.
- También los nuevos datos del producto

¿Cual es la URL que vamos a crear?

Combina **parámetros** y el acceso al **body** del request

Controller

```
@Put('/:index')  
public updateProducto(@Body() prod : any,  
@Param('index') index): boolean {  
    return this.productoService  
        .updateProducto(parseInt(index), prod);  
}
```

Service

```
public updateProducto(producto: Producto, position:
number): boolean {
    this.listaProductos[position] = producto;
    return true;
}
```

CFP

Programador full-stack

Actualización de un recurso (PUT)

Actualización de Producto

Para enviar la actualización desde el front-end:

- Se usa fetch, indicando el parámetro de posición en la ruta y los nuevos datos del producto en el cuerpo similar al POST.
- Se transforma la tabla de texto en tabla de inputs.
- Como en el caso del borrado debe haber un boton para actualizar cada producto. Y con las mismas características:
 - Que cada boton sepa de que elemento es.
 - Asignarle el click (el mismo a todos).
 - Que el click pueda diferenciar que botón fue.

JS

```
function mostrarTablaCompras() {
```

```
  html = "";
```

```
  for (let i=0; i<compras.length; i++) {
```

```
    let r = compras[i];
```

```
    html += `<tr>
```

```
      <td><input type="text" value="${r.producto}" id="prod${i}"></td>
```

```
      <td><input type="text" value="${r.precio}" id="prec${i}"></td>
```

```
      <td><button class="btnUpdProd" pos="${i}">Actualizar</button></td>
```

```
    </tr>`;
```

```
  }
```

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

```
  let botonesActualizar = document.querySelectorAll(".btnUpdProd");
```

```
  botonesActualizar.forEach(e => {e.addEventListener("click",  
                                                        btnActualizarClick);});
```

```
}
```

Necesito "i", así que tengo que usar un for clásico

Genero un input por campo y un boton "actualizar" por cada elemento, marcando cual "i" es

Asigno evento de click

Actualización de Producto

```
async function btnActualizarClick() {  
  let pos = this.getAttribute("pos");  
  let renglon = {  
    "producto": document.getElementById(`prod${pos}`).value,  
    "precio": document.getElementById(`prec${pos}`).value  
  };  
  let response = await fetch(`/productos/${pos}`, {  
    "method": "PUT",  
    "headers": {  
      "Content-Type": "application/json"  
    },  
    "body": JSON.stringify(renglon);  
  })  
  load();  
}
```

Uso "this" para saber que boton disparo el borrado

Armo el JSON del elemento desde los input.

Uso el fetch combinando la ruta nueva con la posicion a modificar y el cuerpo con los nuevos datos.

Resultado