

# Bootcamp Crisalis

## Orange Team

Faccipieri, Federico

Fleitas, Gonzalo

Flores, Elio

Haoy, Sebastián

Pepa Degani, Julián



# Impuestos por productos y/o servicios (backend)

Responsable: Facciopieri, Federico

JIRA : [SCRUM-37](#) FrontEnd

## Descripción

Se creó la lógica de backend necesaria para poder administrar los diferentes impuestos.

También se realizó lo necesario para poder asociar un determinado impuesto con los productos y/o servicios correspondientes.

## Requerimientos

Una vez teniendo un impuesto creado debe ser posible poder asignarlo a un producto o servicio. En caso de ser necesario debe ser posible editar o eliminar un determinado impuesto sobre un producto o servicio.

## Vistas del frontend

En la siguiente imagen pueden verse una parte del código del controlador "SellableGoodController", el cual es el controlador justamente de tanto los servicios como productos.

Asimismo en la imagen pueden verse los métodos de agregar un impuesto ("addTax"), eliminar un impuesto ("removeTax"), y traer o leer todos los impuestos ("getTaxes") de un determinado producto o servicio.

```
89      no usages  ± elioflo
90      @PreAuthorize("hasRole('ADMIN')")
91      @Transactional
92      @PostMapping("/addTax")
93  @ > public ResponseEntity<?> addTaxSellableGood(@RequestBody SellableGoodTax data) throws Exception {...}
102
103      no usages  ± elioflo
104      @PreAuthorize("hasRole('ADMIN')")
105      @PostMapping("/removeTax")
106  @ > public ResponseEntity<?> deleteTaxFromSellableGood(@RequestBody SellableGoodTax data) throws Exception {...}
113
114      no usages  ± elioflo
115      @PreAuthorize("hasAnyRole('USER' , 'ADMIN')")
116      @GetMapping("/getTaxes")
117  > public ResponseEntity<List<Tax>> getAllTaxesBySellableGoodId(@RequestBody Long sellableGoodId) throws Exception {...}
124
```