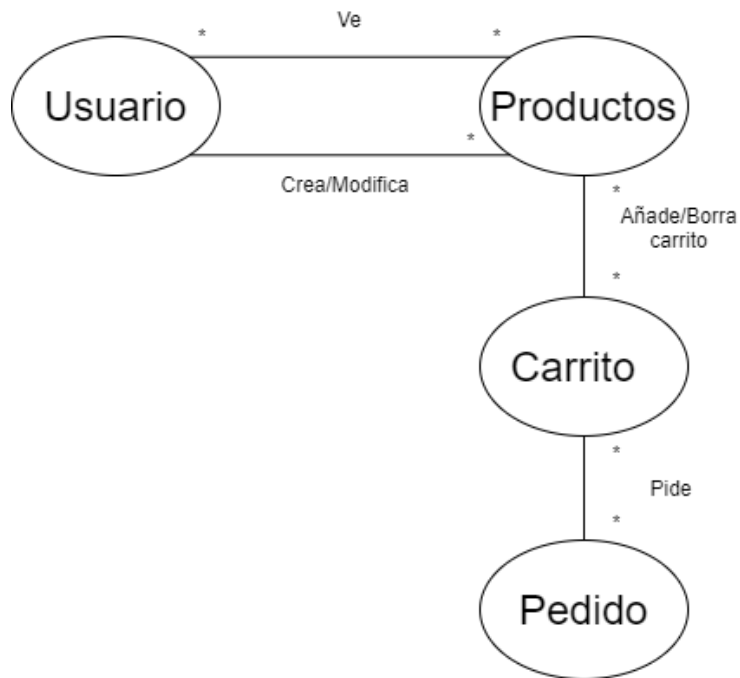


Vestimentas Suaves



Casos de uso simplificados:

- Un usuario debe poder hacer login
- Un usuario debe poder registrarse
- Un usuario sin estar autenticado debe poder ver los productos de forma general
- Un usuario autenticado debe poder ver los productos individualmente
- Un usuario autenticado que además es un administrador debe poder modificar, añadir y borrar productos
- Un usuario autenticado puede ver sus datos y modificarlos
- Un usuario autenticado debe poder añadir y quitar productos al carrito *sin implementar*
- Un usuario autenticado debe poder generar un pedido *sin implementar*

Llamadas al API: Usuarios

```
router.post('/register', async function(pet, resp){
```

Se implementa el caso de uso correspondiente al registro de usuario. Además, he añadido una funcionalidad extra que envía un correo al usuario registrado recordándole que tiene que iniciar sesión con su usuario y contraseña.

```
router.post('/login', async function(pet, resp){
```

Se implementa el caso de uso correspondiente al inicio de sesión.

```
router.get('/users/:id',chequeaJWT, async function(pet, resp){
```

Se implementa el caso de uso que permite a los usuarios autenticados ver su información.

```
router.put('/users/:id',chequeaJWT, async function(pet, resp){
```

Se implementa el caso de uso que permite a los usuarios autenticados modificar su información.

Llamadas al API: Productos

```
router.get('/products', async function(pet, resp){
```

Se implementa el caso de uso correspondiente al listado de productos. Además, he añadido una funcionalidad extra que permite el paginado añadiendo dos parámetros *page* y *limit*.

```
router.get('/products/:id',chequeaJWT, async function(pet, resp){
```

Se implementa el caso de uso que permite a los usuarios autenticados ver los productos de manera individual.

```
router.post('/products', upload.single('productImage'), chequeaJWT, async function(pet, resp){
```

Se implementa el caso de uso que permite a los usuarios autenticados que sean administradores subir productos. Además, he añadido una funcionalidad que permite subir las imágenes de los productos.

```
router.delete('/products/:id', chequeaJWT, async function(pet, resp){
```

Se implementa el caso de uso que permite a los usuarios autenticados que sean administradores borrar productos. Borra también las imágenes almacenadas en el servidor.

```
router.put('/products/:id', upload.single('productImage'), chequeaJWT, async function(pet, resp){
```

Se implementa el caso de uso que permite a los usuarios autenticados que sean administradores modificar productos. Modifica también las imágenes almacenadas en el servidor.

Durante todo el proceso hay una persistencia de datos gracias a una base de datos no relacional (MongoDB). Estos datos están almacenados en un *cluster* (Atlas) y se acceden gracias a moongoose. Está configurado para que se pueda acceder desde cualquier ip. Sin embargo, desde el eduroam no he conseguido acceder.