# Anexo IV - Endpoints

## Productos

## Mostrar producto

Devuelte un objeto ProductDTO con los datos de un producto.

* **URL**

/products

* **Method:**

GET

* **URL Params**

**Required:**

id=[ integer ]

* **Data Params**

None

* **Success Response:**
  + **Code:** 200
  + **Content:** ProductDTO (Java Object from package com.Encom.DTO)
* **Error Response**
  + **Code:** 404 NOT FOUND
  + **Content:** -
  + **Code:** 401 UNAUTHORIZED
  + **Content:** { message: “Access is denied due to invalid credentials.” }
  + **Code:** 403 FORBIDDEN
  + **Content:** { message: “You do not have permission to view this directory or page using the credentials that you supplied.”}

## Listar todos los productos

Devuelte un ArrayList de ProductDTO con los datos de un producto.

* **URL**

/products

* **Method:**

GET

* **URL Params**

**Required:**

None

* **Data Params**

None

* **Success Response:**
  + **Code:** 200
  + **Content:** ArrayList<ProductDTO> (Java Objects from package com.Encom.DTO)
* **Error Response**
  + **Code:** 401 UNAUTHORIZED
  + **Content:** { message: “Access is denied due to invalid credentials.” }
  + **Code:** 403 FORBIDDEN
  + **Content:** { message: “You do not have permission to view this directory or page using the credentials that you supplied.”}

## Listar productos de una categoría

Devuelte un ArrayList de ProductDTO con los datos de un producto de una determinada categoría.

* **URL**

/products

* **Method:**

GET

* **URL Params**

**Required:**

idCategory = [integer]

idSubcategory = [integer]

* **Data Params**

None

* **Success Response:**
  + **Code:** 200
  + **Content:** ArrayList<ProductDTO> (Java Objects from package com.Encom.DTO)
* **Error Response**
  + **Code:** 404 NOT FOUND
  + **Content:** -

## Crear un producto

Devuelte un mensaje de confirmacion.

* **URL**

/products

* **Method:**

POST

* **URL Params**

None

* **Data Params**

**Required:**

action = [“add”]

idCategory = [integer]

idSubcategory = [integer]

name = [String]

description = [String]

price = [BigDecimal]

image = [file]

* **Success Response:**
  + **Code:** 200
  + **Content:** { message: "The product has been successfully added."}
* **Error Response**
  + **Code:** 404
  + **Content:** { message: The product already exist or could not be added."}

## Editar un producto

Devuelte un mensaje de confirmacion.

* **URL**

/products

* **Method:**

POST

* **URL Params**

None

* **Data Params**

**Required:**

action = [“edit”]

id = [integer]

idCategory = [integer]

idSubcategory = [integer]

name = [String]

description = [String]

price = [BigDecimal]

image = [file]

* **Success Response:**
  + **Code:** 200
  + **Content:** { message: "The product has been successfully updated."}
* **Error Response**
  + **Code:** 404
  + **Content:** { message: "The product does not exist or could not be updated."}

## Eliminar un producto

Devuelte un mensaje de confirmacion.

* **URL**

/products

* **Method:**

POST

* **URL Params**

None

* **Data Params**

**Required:**

action = [“delete”]

id = [integer]

* **Success Response:**
  + **Code:** 200
  + **Content:** { message: "The product has been successfully removed."}
* **Error Response**
  + **Code:** 404
  + **Content:** { message: "The product does not exist or could not be removed."}

## Usuarios

## Mostrar usuario

Devuelte un objeto UserDTO con los datos de un usuario.

* **URL**

/users

* **Method:**

GET

* **URL Params**

**Required:**

id=[ integer ]

* **Data Params**

None

* **Success Response:**
  + **Code:** 200
  + **Content:** UserDTO (Java Object from package com.Encom.DTO)
* **Error Response**
  + **Code:** 404 NOT FOUND
  + **Content:** -
  + **Code:** 401 UNAUTHORIZED
  + **Content:** { message: “Access is denied due to invalid credentials.” }
  + **Code:** 403 FORBIDDEN
  + **Content:** { message: “You do not have permission to view this directory or page using the credentials that you supplied.”}

## Listar todos los usuarios

Devuelte un ArrayList de UserDTO con los datos de un producto.

* **URL**

/users

* **Method:**

GET

* **URL Params**

**Required:**

None

* **Data Params**

None

* **Success Response:**
  + **Code:** 200
  + **Content:** ArrayList<UserDTO> (Java Objects from package com.Encom.DTO)
* **Error Response**
  + **Code:** 401 UNAUTHORIZED
  + **Content:** { message: “Access is denied due to invalid credentials.” }
  + **Code:** 403 FORBIDDEN
  + **Content:** { message: “You do not have permission to view this directory or page using the credentials that you supplied.”}

## Crear un usuario

Devuelte un mensaje de confirmacion.

* **URL**

/users

* **Method:**

POST

* **URL Params**

None

* **Data Params**

form = [String]

**Required:**

action = [“add”]

user = [String]

password = [String]

admin = [Boolean]

name = [String]

surname = [String]

dni= [String]

birthdate = [Date]

email= [String]

* **Success Response:**
  + **Code:** 200
  + **Content:** { message: "The user has been successfully added."}
* **Error Response**
  + **Code:** 500
  + **Content:** -

## Editar un usuario

Devuelte un mensaje de confirmacion.

* **URL**

/users

* **Method:**

POST

* **URL Params**

None

* **Data Params**

form = [String]

**Required:**

action = [“edit”]

user = [String]

password = [String]

admin = [Boolean]

name = [String]

surname = [String]

dni= [String]

birthdate = [Date]

email= [String]

* **Success Response:**
  + **Code:** 200
  + **Content:** { message: "The user has been successfully updated."}
* **Error Response**
  + **Code:** 500
  + **Content:** -

## Eliminar un usuario

Devuelte un mensaje de confirmacion.

* **URL**

/users

* **Method:**

POST

* **URL Params**

None

* **Data Params**

**Required:**

action = [“delete”]

id = [integer]

* **Success Response:**
  + **Code:** 200
  + **Content:** { message: "The user has been successfully removed."}
* **Error Response**
  + **Code:** 404
  + **Content:** { message: "The user does not exist or could not be removed."}

## Authenticate

## Login

Autentica las credenciales enviadas e inicia una sesión.

* **URL**

/authenticate

* **Method:**

POST

* **URL Params**

None

* **Data Params**

**Required:**

user = [String]

password = [String]

* **Success Response:**
  + **Code:** 200
  + **Content:** -
* **Error Response**
  + **Code:** 404
  + **Content:** {message: “Invalid user name or password”}

## Sign out

## Logout

Desloguea al usuario actual y borra la sesión.

* **URL**

/logout

* **Method:**

GET

* **URL Params**

None

* **Data Params**

None

* **Success Response:**
  + **Code:** 200
  + **Content:** -
* **Error Response**
  + **Code:** 500
  + **Content:** -