**API PROYECTO TIENDA WEB DOCUMENTACION**

Contenido

[Operaciones Autenticación y registro 2](#_Toc461892941)

[#1 Registro usuario 2](#_Toc461892942)

[#3 Login usuario 2](#_Toc461892943)

[#3 Obtener info usuario 2](#_Toc461892944)

[#4 Modificacion usuario 3](#_Toc461892945)

[#5 Baja usuario 3](#_Toc461892946)

[OPERACIONES CESTA COMPRA 4](#_Toc461892947)

[OPERACIONES TIENDA 5](#_Toc461892948)

[#1 Obtener listado de tiendas 5](#_Toc461892949)

[#2 Obtener tienda especifica 5](#_Toc461892950)

[#3 Crear tienda 5](#_Toc461892951)

[#4 Modificar tienda 5](#_Toc461892952)

[#4 Borrar tienda 6](#_Toc461892953)

[OPERACIONES PRODUCTOS 7](#_Toc461892954)

[OPERACIONES METODOS ENVIO 8](#_Toc461892955)

[OPERACIONES FACTURACION 9](#_Toc461892956)

[OPERACIONES HISTORICO 10](#_Toc461892957)

# Operaciones Autenticación y registro

***Nota: Los usuarios administrativos no tendrán portal de registro, por seguridad los administradores serán añadidos a petición de email directamente en la base de datos***

## #1 Registro usuario

**Descripción:** Enpoint para registro de usuarios, mirar la sección de descripción de la base de datos para saber los campos a mandar en el formulario. Para mejor debug el api manda un mensaje descriptivo en caso de error de parametros

El api a través de este endpoint registra todo tipo de usuario, identificara si el usuario es tipo cliente o tipo tienda automáticamente

**Endpoint:** /api/user/

**Tipo de request:** POST

**REQUEST:** Form con los campos de usuario a rellenar

**Autenticacion Necesaria:** NO

**Return:** JSON con el usuario creado o 404 en caso de que falte algún campo obligatorio

## #3 Login usuario

**Descripcion:** Endpoint para login de usuarios. Todos los usuarios independientemente si son administradores, tiendas o usuarios comunes harán login a través de este endpoint. Por seguridad NO devolverá ninguna descripción en caso de error… solo error 500

**Endpoint:** /api/user/login/user\_name

**Tipo de request:** POST

**Autenticacion Necesaria:** NO

**Request:** form con username y password

**Return:** JSON con el usuario o 500 en caso de autenticación incorrecta, o error

## #3 Obtener info usuario

**Descripcion:** Realmente un usuario solo podrá obtener info de el mismo si esta autenticado o si eres tipo admin, en cuyo caso el token de administración será válido para autenticar cualquier tipo de request.

**Endpoint:** /api/user/id\_usuario

**Tipo de request:** GET

**Autenticacion Necesaria:** SI

**Tipo de autenticación:** Token de usuario (user:user\_token)

**Return:** JSON con el usuario o 404 en caso de autenticación incorrecta o error

## #4 Modificacion usuario

**Descripción:** Ir a la sección de descripción de la base de datos para saber los parámetros editables del usuario. Se esperará un form con los parámetros a editar. Se devolverá un mensaje descriptivo en la respuesta de error para mejor debug

**Endpoint:** /api/user/id\_usuario

**Tipo de request:** PUT

**Autenticacion Necesaria:** SI

**Request:** Campos a modificar del objecto usuario

**Tipo de autenticación:** Token de usuario (user:user\_token)

**Return:** JSON con el usuario o 404 en caso de autenticación incorrecta o error

## #5 Baja usuario

***TODA LA INFORMACION DEL USUARIO SERA ELIMINADA PERMANENTEMENTE***

**Endpoint:** /api/user/id\_usuario

**Tipo de request:** DELETE

**Autenticacion Necesaria:** SI

**Tipo de autenticación:** Token de usuario (user:user\_token)

**Return:** JSON con el mensaje {deleted:ok, user:id\_usuario} o 404 en caso de autenticación incorrecta o error

# OPERACIONES CESTA COMPRA

**Descripción:** Todas las operaciones a la cesta de compra serán autenticadas, se podrá obtener todos, obtener ítem, crear ítem, modificar ítem, eliminar ítem.

# OPERACIONES TIENDA

## #1 Obtener listado de tiendas

**Endpoint:** /api/shop/

**Tipo de request:** Get

**Autenticacion necesaria:** NO

**Return:** JSON Array. Array de objectos tiendas en formato JSON

## #2 Obtener tienda especifica

**Endpoint:** /api/shop/id\_tienda

**Tipo de request:** Get

**Autenticacion necesaria**: NO

**Return:** JSON de objecto tienda

## #3 Crear tienda

**Endpoint:** /api/shop

**Tipo de request:** POST

**Autenticacion Necesaria:** SI

**Tipo de autenticación:** Token en header (user:user\_token)

**Validacion de la autenticaion:** Token debe pertenecer a un usuario valido tipo tienda

**Request:** Form con los campos del objecto tienda a crear

**Return:** JSON con el objecto tienda creada, 404 si hay error

## #4 Modificar tienda

***No será posible eliminar o crear un artículo de la tienda usando este endpoint por seguridad y simplicidad***

**Endpoint:** /api/shop/id\_shop

**Tipo de request:** PUT

**Autenticacion Necesaria:** SI

**Tipo de autenticación:** Token en header (user:user\_token,shop:shop\_token)

**Validacion de la autenticaion:** Token debe pertenecer a un usuario valido tipo tienda, el token de la tienda debe pertenecer al id de la tienda a modificar

**Request:** Form con los campos del objecto tienda a modificar

**Return:** JSON con el objecto tienda modificado, 404 si hay error

## #4 Borrar tienda

***Este endopoint también borrara todos los artículos asociados a la tienda, no borrara el historial de compras por seguimiento***

**Endpoint:** /api/shop/id\_shop

**Tipo de request**: DELETE

**Autenticacion Necesaria: SI**

**Tipo de autenticación:** Token en header (user:user\_token,shop:shop\_token)

**Validacion de la autenticaion:** Token debe pertenecer a un usuario valido tipo tienda, el token de la tienda debe pertenecer al id de la tienda a modificar

**Request:** No es necesario ningún campo adicional

**Return:** JSON con el mensaje {deleted: ”ok”, id: ”id\_tienda”}, 404 si hay error

# OPERACIONES PRODUCTOS

# OPERACIONES METODOS ENVIO

# OPERACIONES FACTURACION

# OPERACIONES HISTORICO