



TUMANGA

TFG 2022

Trabajo realizado por Adrián Arnaiz Fernández

DAW

tumanga.arnaizdev.com

INDEX

1.	Introducción.....	3
2.	Objetivos.....	3
3.	Tecnologías	3
4.	Despliegue	3
5.	Base de datos.....	4
5.1.	Colecciones	4
6.	Backend	15
6.1.	Dependencias	15
6.2.	Inicio	15
6.3.	Rutas.....	16
6.4.	Middlewares	20
6.5.	Información extra	20
7.	Frontend	20
	Usuarios	22
	Productos.....	24
	Pedidos	24
	Comentarios.....	24
	Roles	25
8.	Pasos.....	25
9.	Funcionalidades	25
10.	Anexo.....	41

URL

<https://tumanga.arnaizdev.com>

1. Introducción

El objetivo de este proyecto es crear una aplicación lo más realista posible sobre una tienda de mangas online. Practicar, aprender y poner en marcha todos los conocimientos necesarios que encontré para poder hacer la página lo más profesional posible, simulando que esa aplicación se va a desplegar en una empresa de verdad.

2. Objetivos

- Comprar artículos de una manera lo más rápida posible
- Ver la información de tu cuenta lo más detallada posible
- Vender los artículos fácilmente en el lado de la empresa

3. Tecnologías

Para este proyecto, he usado estas tecnologías:

Backend	Frontend	BBDD	Diseño
NodeJS	ReactJS	MongoDB	FIGMA

4. Despliegue

En este paso use:

Backend	Frontend	BBDD
Render	Netfily	MongoDB Atlas

Estuve pensando en desplegarlo en AWS pero al final decidí no hacerlo ya que fui subiendo las partes individualmente y estas webs serán las que más usare en el futuro.

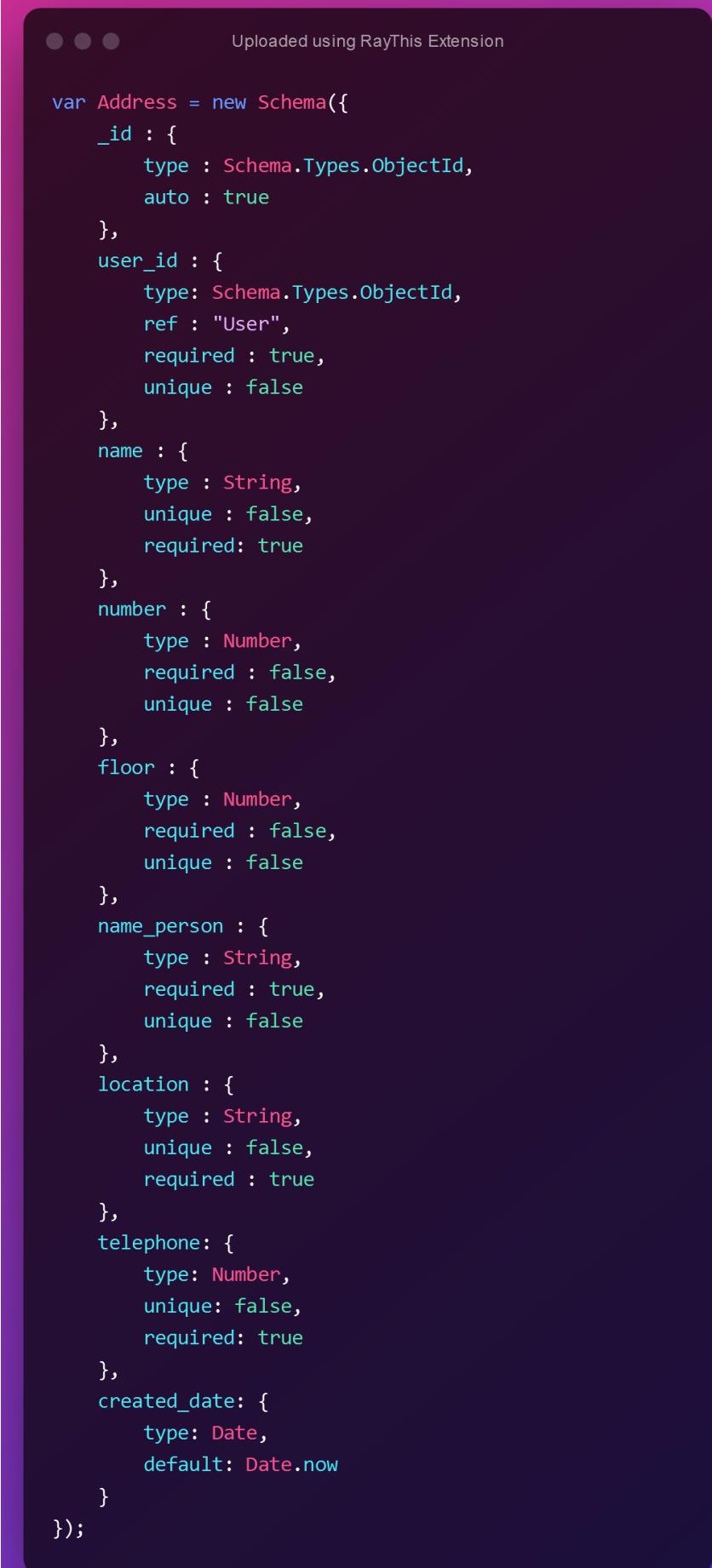
5. Base de datos

Como mencione anteriormente, he usado MongoDB y MongoDB Atlas. Este es una BBDD no relacional. Esta estructura está formada por JSON.

El nombre de la BBDD es TuManga.

5.1. Colecciones

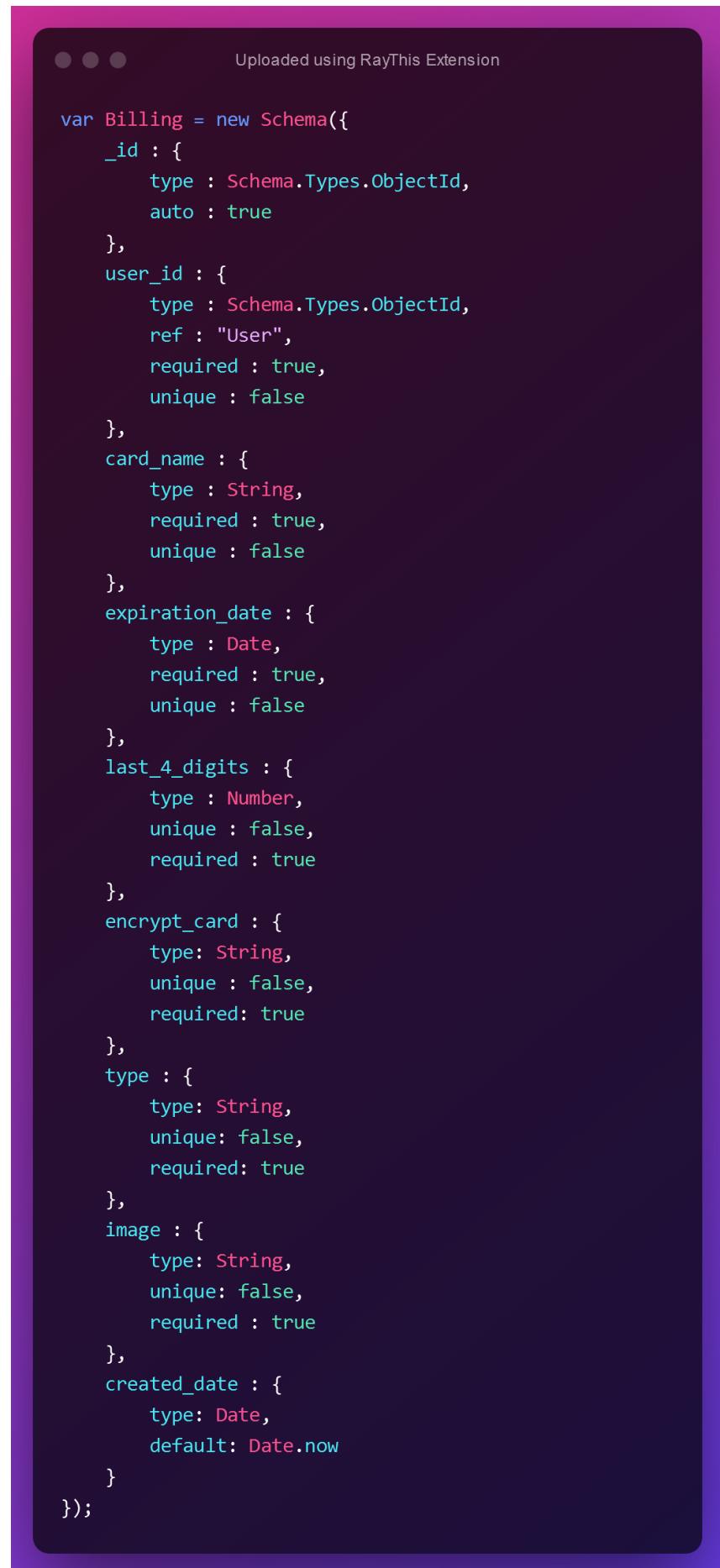
Direcciones



Uploaded using RayThis Extension

```
var Address = new Schema({
  _id : {
    type : Schema.Types.ObjectId,
    auto : true
  },
  user_id : {
    type: Schema.Types.ObjectId,
    ref : "User",
    required : true,
    unique : false
  },
  name : {
    type : String,
    unique : false,
    required: true
  },
  number : {
    type : Number,
    required : false,
    unique : false
  },
  floor : {
    type : Number,
    required : false,
    unique : false
  },
  name_person : {
    type : String,
    required : true,
    unique : false
  },
  location : {
    type : String,
    unique : false,
    required : true
  },
  telephone: {
    type: Number,
    unique: false,
    required: true
  },
  created_date: {
    type: Date,
    default: Date.now
  }
});
```

Facturación



Comentarios

```
● ● ● Uploaded using RayThis Extension

var Comment = new Schema({
  _id : {
    type: Schema.Types.ObjectId,
    auto : true
  },
  user_id : {
    type : Schema.Types.ObjectId,
    ref : "User",
    required : true,
    unique : false
  },
  message : {
    type : String,
    required: true,
    unique : false
  },
  product_id : {
    type : Schema.Types.ObjectId,
    ref : "Product",
    required : true,
    unique : false
  },
  score : {
    type : Number,
    unique : false,
    required : true
  },
  date : {
    type: Date,
    default : Date.now
  },
  name: {
    type: String,
    required: true,
    unique: false
  },
  product_name : {
    type: String,
    required: true,
    unique: false
  }
});
```

Pedidos

```
Uploaded using RayThis Extension

var ItemCart = new Schema({
  product_id: {
    type: Schema.Types.ObjectId,
    ref: "Product",
    unique: false,
    required: false
  },
  quantity : {
    type: Number,
    min: 1,
    required: false,
    unique: false
  },
  price : {
    type: Number,
    required: false,
    unique: false
  },
  name: {
    type: String,
    required: false,
    unique: false
  },
  image : {
    type: String,
    required: false,
    unique: false
  },
  total_price: {
    type: Number,
    required: false,
    unique: false
  }
})
```

```
var Order = new Schema({
  _id: {
    type: Schema.Types.ObjectId,
    auto: true
  },
  id_client: {
    type: Schema.Types.ObjectId,
    ref: "User",
    required: true,
    unique: false
  },
  order_date: {
    type: Date,
    default: Date.now
  },
  send_date: {
    type: Date,
    unique: false
  },
  products: [ItemCart],
  /* products: [
    {
      type: Schema.Types.ObjectId,
      ref: "Product",
      unique: false,
      required: true
    }
  ], */
  delivery_address: {
    type: Schema.Types.ObjectId,
    ref: "Address",
    unique: false,
    required: false
  },
  billing: {
    type: Schema.Types.ObjectId,
    ref: "Billing",
    unique: false,
    required: false
  },
  state: {
    type: String,
    unique: false,
    required: true
  },
  total: {
    type: Number,
    required: true,
    unique: false
  }
});
```

Producto

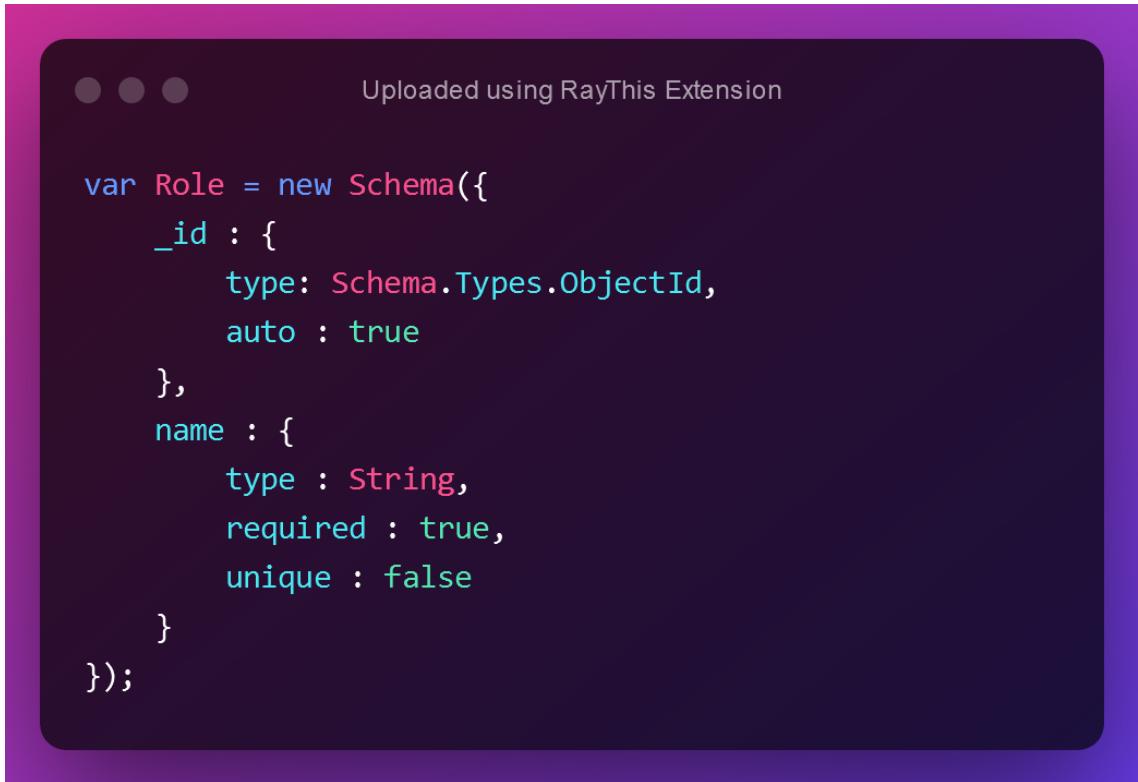
Uploaded using RayThis Extension

```
var Product = new Schema({
  _id: {
    type: Schema.Types.ObjectId,
    auto: true
  },
  name: {
    type: String,
    unique: false,
    required: true
  },
  price: {
    type: Number,
    unique: false,
    required: true
  },
  description: {
    type: String,
    unique: false,
    required: true
  },
  short_description: {
    type: String,
    unique: false,
    required: true
  },
  upload_date: {
    type: Date,
    default: Date.now
  },
  state: {
    type: String,
    unique: false,
    required: true
  },
  stock: {
    type: Number,
    unique: false,
    required: true
  },
  categories: [String],
```

```
● ● ● Product2

type: {
    type: String,
    unique: false,
    required: true
},
image: {
    type: String,
    unique: false,
    required: false,
    default: null
},
number_sales: {
    type: Number,
    unique: false,
    required: true
},
authors: {
    type: String,
    unique: false,
    required: false
},
editorial: {
    type: String,
    unique: false,
    required: false
},
series: {
    type: String,
    unique: false,
    required: false
},
comments: [
    type: Schema.Types.ObjectId,
    ref: "Comment",
    unique: false,
    required: false
]
});
```

Roles



Uploaded using RayThis Extension

```
var Role = new Schema({
  _id : {
    type: Schema.Types.ObjectId,
    auto : true
  },
  name : {
    type : String,
    required : true,
    unique : false
  }
});
```

Usuarios

```
● ● ● User

var User = new Schema({
  _id : {
    type: Schema.Types.ObjectId,
    auto : true
  },
  name : {
    type : String,
    unique : false,
    required : false
  },
  last_name : {
    type : String,
    unique : false,
    required : false
  },
  email : {
    type : String,
    unique : true,
    required : true
  },
  password_hash : {
    type : String,
    unique : false,
    required : true
  },
  register_date : {
    type : Date,
    default : Date.now
  },
  state : {
    type : String,
    unique : false,
    required : true,
    default : "Active"
  },
});
```

```
  ...
  User2

  billing : [
    {
      type : Schema.Types.ObjectId,
      ref : "Billing",
      unique : false,
      required : true
    }
  ],
  cart : [
    {
      type : Schema.Types.ObjectId,
      ref : "Orders",
      unique : false,
      required : true
    }
  ],
  address : [
    {
      type : Schema.Types.ObjectId,
      ref : "Address",
      unique : false,
      required : true
    }
  ],
  role : {
    type : Schema.Types.ObjectId,
    ref : "Role",
    unique : false,
    required: true
  },
  comments : [
    {
      type : Schema.Types.ObjectId,
      ref : "Comment",
      unique: false,
      required : true
    }
  ],
  tokenRecover: {
    type: String,
    required: false,
    unique: false
  }
});
```

6. Backend

El backend consiste en una API REST creada con NodeJS y desplegada en render. Se encarga de la gran mayoría de la lógica de la web.

6.1. Dependencias

- Express -> para poder usar el backend en modo API REST
- Body-parser -> para transformar a JSON todos los datos que llegan a la API
- Mongoose -> para poder conectar el backend con MongoDB
- Connect-multiparty -> Para poder crear archivos temporales (imágenes)
- Validator -> Para poder validar los datos
- Nodemon (DEV) -> para que se reinicie el servidor cuando se realizan cambios. Solo en modo desarrollo
- Bcryptjs -> para encriptar contraseñas y poder trabajar con ellas.
- Jsonwebtoken -> para crear tokens y facilitar la conexión de usuarios
- Cors -> al principio usé esta dependencia para poder configurar el cors automáticamente, pero como me fallaba, al final decidí no usarla

6.2. Inicio

Para el inicio del backend (una vez se arranca por primera vez), cree la siguiente configuración:

- Crear los roles automáticamente
 - Admin
 - Owner
 - Empleado
 - Usuario
- Crear un usuario admin
- Creación de 10 productos

6.3. Rutas

La ruta principal y común es: /api/v1.

```
● ● ● Ruta productos

// * ----- PRODUCTS -----
router.get("/products/new", controllers.getNewProducts); // Get the 8 new products
router.get("/products/mangas/:limit&:skip", controllers.getMangas); // Get mangas or light novel 8 to 8

router.get("/products/merchandising/:limit&:skip", controllers.getMerchandising); // Get merchandising 8 to 8
router.get("/products", middlewares.isUser, controllers.getAllProducts); // Get all products

router.get("/product/:id", controllers.getProduct); // Get one product
```

```
● ● ● Ruta usuarios

// * ----- USER -----
router.post("/user", controllers.createUser); // Create a new user
router.get("/user", middlewares.isUser, controllers.getUser); // Get one user with token
router.patch("/user", middlewares.isUser, controllers.changeUserState); // Change the state of user

router.post("/login", controllers.login); // Login
router.put("/user", middlewares.isUser, controllers.updateUser); // Update the user
router.put("/user/password", middlewares.isUser, controllers.updatePasswords); // Change password

router.delete("/user", middlewares.isUser, controllers.deleteUser); // Delete the user
```

```
● ● ● Ruta pedidos

// * ----- ORDERS -----
router.get("/order/process", middlewares.isUser, controllers.getOrderProcess); // Get the user order with state === "P"
router.post("/order", middlewares.isUser, controllers.createOrder); // Create new order
router.patch("/order", middlewares.isUser, controllers.addProductOrder); // Add product in the order that have state === "P"
router.put("/order", middlewares.isUser, controllers.updateOrder); // Finish the order
router.get("/orders/user", middlewares.isUser, controllers.getUserOrders); // Get all the orders of a user
router.get("/order/cart", middlewares.isUser, controllers.getOrderCart); // Get the shopping cart

router.put("/order/product", middlewares.isUser, controllers.updateShoppingCart); // Delete a product of a cart
router.get("/order/:id", middlewares.isUser, controllers.getOrderId); // Get one order
```

Ruta facturacion

```
// * ----- BILLING -----  
  
router.post("/card" , middlewares.isUser, controllers.createCard); // Create new card  
router.get("/cards/last", middlewares.isUser, controllers.getLastCards); // Get 2 last cards  
router.get("/cards/user", middlewares.isUser, controllers.getUserCards); // Get all the cards of a user  
  
router.delete("/card/:id", middlewares.isUser, controllers.deleteCard); // Delete one card
```

Ruta direcciones

```
// * ----- ADDRESS -----  
  
router.post("/address", middlewares.isUser, controllers.createAddress); // Create address  
router.get("/address/user", middlewares.isUser, controllers.getUserAddress); // Get user address  
  
router.delete("/address/:id", middlewares.isUser, controllers.deleteAddress); // Delete address  
  
router.put("/address/:id", middlewares.isUser, controllers.updateAddress); // Update address  
router.get("/address/last", middlewares.isUser, controllers.getLastAddress); // Get last 2 address of  
an user
```

Ruta comentarios

```
// * ----- COMMENTS -----  
  
router.get("/comments/product/:idProduct", controllers.getCommentsProduct); // Get all the comments of a  
one product  
router.post("/comment", middlewares.isUser, controllers.createComment); // Create a comment  
router.get("/comments/user", middlewares.isUser, controllers.getUserComments); // Get user comments  
  
router.delete("/comment/:idComment", middlewares.isUser, controllers.deleteComment); // Delete a  
comment
```

Rutas: Buscar - Filtrar - Ordenar

```
// * ----- SEARCH -----  
  
router.get("/search/products/:search", controllers.searchProducts); // Search in products  
  
// * -----  
  
// * ----- FILTER -----  
  
router.get("/filter/product/:type/:option&:limit&:skip", controllers.filterProduct); // Filter any product  
router.get("/filters/:type" , controllers.getFilters ); // Get all the filter  
  
// * -----  
  
// * ----- SORT -----  
  
router.get("/sort/products/manga/:option&:limit&:skip", controllers.sortManga); // Sort mangas  
  
router.get("/sort/products/merchandising/:option&:limit&:skip", controllers.sortMerchandising); // Sort merchandising  
  
// * -----
```

Ruta password

```
// * ----- PASSWORD -----  
  
router.post("/email", controllers.sendEmail); // Send email if forgot the password  
router.post("/recover/:token", controllers.recoverPassword); // Change the password
```

Ruta imagenes

```
// * ----- IMAGES -----  
  
router.get("/image/:image", controllers.getImage); // Get a image  
router.post("/image/:id", [middlewares.isEmployee, md_upload],controllers.uploadImage); // Upload a image
```

Ruta admin

```
// * ----- ADMIN -----  
  
router.get("/total/data", middlewares.isEmployee, controllers.totalAdminData); // Get the total orders  
(Admin)  
router.get("/total/users", middlewares.isEmployee, controllers.totalUsers); // Get the total users  
(Admin)  
router.get("/total/earnings", middlewares.isOwner, controllers.totalEarnings); // Get the total  
earnings (Admin)  
router.get("/products/sales/:limit", middlewares.isEmployee, controllers.mostBestsellers); // Get the  
most bestsellers products  
router.get("/admin/orders", middlewares.isEmployee, controllers.getAllOrders); // Get all the orders  
  
router.get("/admin/users/:filter?", middlewares.isEmployee, controllers.getAllUsers); // Get all users  
  
router.get("/admin/cards", middlewares.isEmployee, controllers.getAllCards); // Get all cards  
  
router.get("/admin/roles", middlewares.isAdmin, controllers.getAllRoles); // Get all roles  
router.get("/admin/comments", middlewares.isEmployee, controllers.getAllComments); // Get all comments  
  
router.put("/admin/user/:id", middlewares.isAdmin, controllers.updateUserAdmin); // Update one user  
(Admin)  
router.put("/admin/product/:id", middlewares.isEmployee, controllers.updateProductAdmin); // Update one  
product (Admin)  
router.get("/admin/user/:id", middlewares.isEmployee, controllers.getUserAdmin); // Get one user  
(Admin)  
router.delete("/admin/user/:id", middlewares.isAdmin, controllers.deleteUserAdmin); // Delete one user  
(Admin)  
router.post("/admin/user", middlewares.isAdmin, controllers.createUserAdmin); // Create one user  
  
router.post("/product", middlewares.isEmployee, controllers.newProduct); // Create a new product  
  
router.get("/admin/search/:search&:option", middlewares.isEmployee, controllers.searchAdmin); // Search  
users  
router.delete("/admin/product/:id", middlewares.isEmployee, controllers.deleteProduct); // Delete a  
product  
router.get("/admin/role/search/:search", middlewares.isAdmin, controllers.searchRole); // Search roles  
  
router.post("/admin/role", middlewares.isAdmin, controllers.createRole); // Create role  
router.get("/admin/role/:id", middlewares.isAdmin, controllers.getOneRole); // Get one role  
router.put("/admin/role/:id", middlewares.isAdmin, controllers.updateRole); // Update one role  
  
router.delete("/admin/role/:id", middlewares.isAdmin, controllers.deleteRole); // Delete one role  
  
router.delete("/admin/comment/:id", middlewares.isEmployee, controllers.deleteCommentAdmin); //Delete  
one comment
```

6.4. Middlewares

- isAdmin -> comprobar si el usuario es admin
- isOwner -> comprobar si el usuario es owner o superior
- isEmployee -> comprobar si el usuario es empleado o superior
- isUser -> comprobar si el usuario es usuario o superior

6.5. Información extra

- El email esta guardado siempre en minúsculas, de esta manera nunca podrá haber 2 cuentas con el mismo email
- Creación de un método para poder enviar un correo en el caso de que el usuario no recuerde su contraseña

7. Frontend

En toda la web, se cuenta con un botón que te permite ir de manera automática a la parte superior.

En todo momento la página es responsive.

El frontend esta dividido en 3 partes:

1. Parte pública sin registro:

- En esta parte, sería la página que se vería nada más entrar en la página.
- Al no estar registrado, las opciones están mas limitadas. No se puede añadir nada al carrito.
- Cualquier opción que este disponible sobre tu cuenta, te redireccionara al login automáticamente.
- No existe la opción de poder escribir comentarios en ningún producto.
- En el login, se contará con la posibilidad de cambiar la contraseña.

- Si seleccionas la opción de recordarte, la duración de la validez del JSONWebToken será de 24hrs. Si no se selecciona, será de 2 horas.
 - La contraseña tendrá que contar con (min 6 – máx. 16 caracteres)
2. Parte pública pero registrado:
- En esta parte se te añadirá en el header la opción de carrito de compra, que te mostrará los 3 primeros productos que estén en el carrito y el total.
 - Se añadirá la opción de añadir al carrito cualquier producto que esté disponible
 - Tanto en el menú como en el header, se dará la opción de poder ir a ‘Mi Cuenta’.
 - Ahora en los productos, se dará la posibilidad de poder poner un comentario sobre el artículo
 - Si ya tienes un comentario escrito, podrás eliminarlo desde la sección del producto.
 - ‘Mi Cuenta’ contara con 5 secciones
 - ‘Mi Información’ -> aquí se podrá modificar los nombres y apellidos del usuario, cambiar la contraseña de la cuenta y eliminar o deshabilitar la cuenta.
 - ‘Mis pedidos’ -> Aquí estarán todos los pedidos que hayas realizado anteriormente. Evidentemente solo saldrán los pedidos finalizados. En cada pedido se dará un poco de información, además de una opción que te llevará al resumen entero de ese pedido.
 - ‘Mis direcciones’ -> Aquí saldrán todas las direcciones que tengas registradas en tu cuenta. Podrás editarlas, eliminarlas o hasta añadir nuevas.
 - ‘Mis tarjetas’ -> Aquí se verán todas las tarjetas registradas. Podrás eliminarlas con un solo clic o añadir una nueva.
 - ‘Mis comentarios’ -> Aquí saldrán todos los comentarios que hayas realizado. Podrás eliminarlo también.
 - En el apartado del carrito, tendrás la opción de eliminar un producto.

- Si finalizas el producto, te llevara a la página de pago. En ella, podrás cambiar tanto la dirección de envío como la tarjeta con la que pagaras. De manera automática, te mostrara las 2 ultimas usadas y podrás seleccionarlas o crear una nueva en el momento.
3. La plataforma -> En ella solo tendrán acceso los usuarios que tengan los roles de Admin, Owner o Empleado. Cada rol tendrá unas pequeñas diferencias respecto al resto.
- Cada uno de ellos tendrá la opción de ir a la plataforma desde un nuevo botón que este situado en el menú.
 - Nada más entrar, todos los roles podrán ver estadísticas de la tienda. Ej:
 - N.º total de pedidos realizados
 - N.º total de usuarios registrados
 - N.º total de ganancias (solo admin y owner)
 - Luego podrán ver un grafico sobre los 5 productos más vendidos.
 - Luego tendrán un menú lateral con diferentes opciones:
 - Inicio
 - Usuarios
 - Productos
 - Pedidos
 - Roles (Solo admin)
 - Comentarios

Usuarios

Por cada usuario se podrá ver los siguientes datos:

ID

627c152831add8cbe4634b20

EMAIL

admin@admin.com

NOMBRE

Admin

APELLIDOS

Admin

ESTADO

Activo

FECHA REGISTRO

11/05/2022

ROL

Admin

Nº PEDIDOS

5

Nº DIRECCIONES

1

Nº FACTURACIONES

1

Nº COMENTARIOS

1

Únicamente el usuario con rol admin podrá modificar los usuarios, también podrá crear un nuevo usuario.

Cualquier usuario podrá buscar un usuario mediante la búsqueda superior.

Productos

Respecto a los productos, se podrá ver la información detallada de todos los datos.

Cualquier rol podrá editar o crear un nuevo producto.

A la hora de editar, se podrá cambiar la imagen del producto mientras te muestra la actual.

En ambos casos, se cuenta con un rápido y sencillo mecanismo para poder añadir y eliminar categorías.

Si quieras eliminar un producto, tendrás que darle la opción de editar y ahí aparecerá la opción.

Pedidos

Se podrá ver un resumen bastante básico sobre los pedidos ya realizados.

Evidentemente, solo se mostrarán los pedidos que se hayan ejecutado y pagado.

Comentarios

Se vera un resumen de todos los comentarios que se hayan realizado en la plataforma/web.

Cualquier rol podría eliminarlo, por si se diera el caso de comentarios indebidos o casos similares.

Roles

En esta sección que solo el admin podrá entrar, se vera un resumen de los roles actuales y se podrán editar, eliminar y crear nuevos.

8. Pasos

- Creación del proyecto entero en una hoja.
- Diseño de la interfaz del proyecto con FIGMA
- Creación del backend
- Creación del frontend

9. Funcionalidades

HOME

The screenshot shows the homepage of TuManga. At the top, there's a navigation bar with the logo 'TuManga', a search bar containing 'Busca en nuestro catálogo', and user icons. Below the header is a horizontal row of manga covers: 'Haikyuu!!' (Vols. 33 and 34), 'Sword Art Online Progressive', 'Divergence', 'Haikyuu!! Vol. 45', 'Tokyo Revengers Vol. 7', 'Solo Leveling Vol. 5', and 'One Piece'. A right-pointing arrow indicates more content. On the left side, there are social media sharing icons. The main content area features a section titled 'Novedades' (New Arrivals) with four items: 'Isekai de Cheat Skill wo te ni Shita ore wa, Genjitsu Sekai wo mo Musou Suru ~Level Up wa Jinsel wo Kaeta~' (15€, STOCK), 'The Beginning After The End Divergence' (10€, STOCK), 'Haikyuu!! Vol.45' (10€, STOCK), and 'Haikyuu!! Vol.33' (21€, STOCK). To the right of these is a figure of 'Figura POP! DBZ S8 de Goku (First Appearance) de Dragon Ball' (18€, STOCK).

Home User



Isekai de Cheat Skill wo te ni
Shita ore wa, Genjitsu Sekai wo
mo Musou Suru ~Level Up wa
Jinsei wo Kaeta~

15€ ✓ STOCK



The Beginning After The End

10€ ✓ STOCK



Novedades



Haikyuu!! Vol.45

10€ ✓ STOCK



Haikyuu!! Vol.33

21€ ✓ STOCK



Figura POP! DBZ S8 de Goku
(First Appearance) de Dragon
Ball

18€ ✓ STOCK



Login

¿No tienes cuenta?

Registrarse

Iniciar sesión

Email

Contraseña



Recuérdame

Iniciar sesión

¿Olvidaste la contraseña?

Registro

Registrarse

Registrarse

¿Ya tienes cuenta?

Iniciar sesión

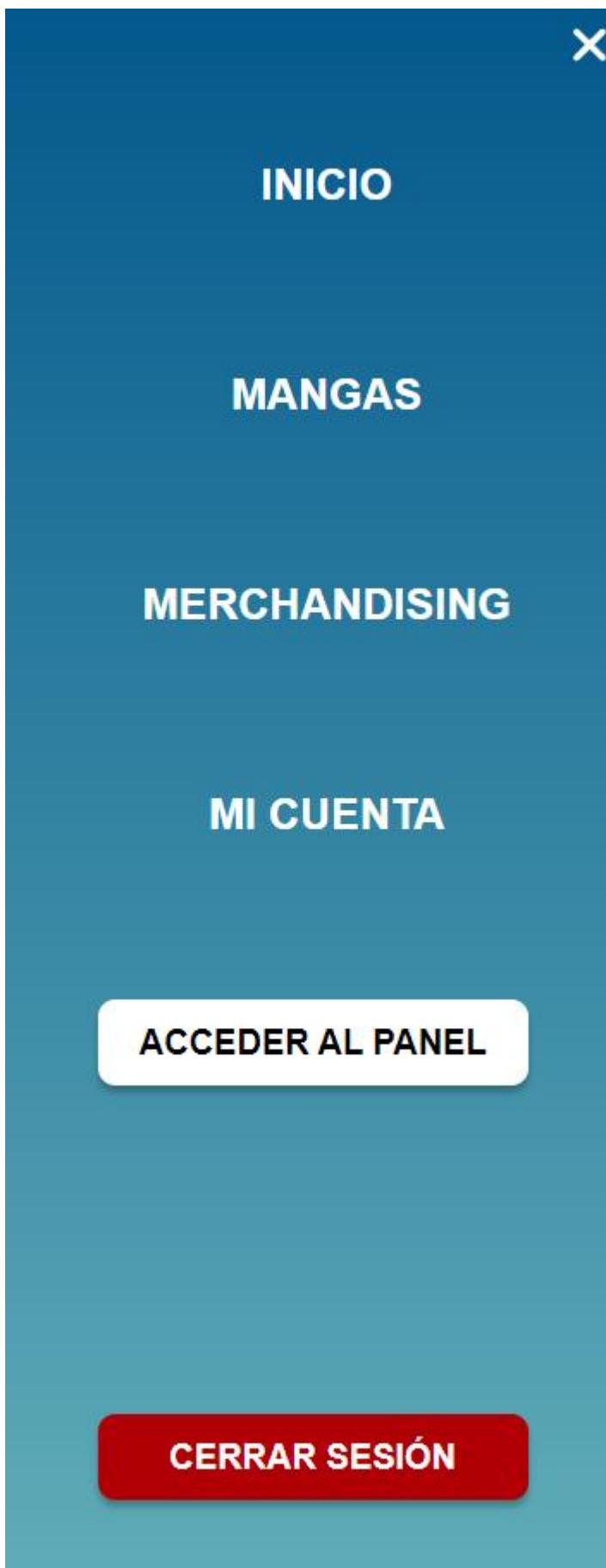
Menu (sin usuario)



Menu (con usuario)



Menu (Admin, Owner o Empleado)



Mangas

TuManga

Busca en nuestro catálogo

MANGAS

Ordenar

Tipos

- Manga
- Novela Ligera

Categorías

- Shonen
- Realidad Virtual
- Seinen
- Acción
- Magia
- Instituto
- Comedia
- Deportes
- Resurrección
- Aventura
- Fantasia
- Harem

Sword Art Online progressive 03
8€ ✓ STOCK

My Hero Academia nº 30
6€ ✓ STOCK

Solo Leveling 5
14€ ✗ AGOTADO

TOKYO REVENGERS 7
16€ ✓ STOCK

HAIKYU!! VOL. 33

THE BEGINNING AFTER THE END
10€ ✓ STOCK

Isekai de Cheat Skill wo te ni Shita ore wa, Genjitsu Sekai wo mo Musou Suru ~Level Up wa Jinsei wo Kaeta~

Mi Cuenta

MI CUENTA

Mi Información

- Mis Pedidos
- Mis Direcciones
- Mis Tarjetas
- Mis Comentarios

Datos de mi cuenta

Nombre: PruebaMod

Apellidos: Si

Guardar

Contraseña

Antigua:

Nueva:

Repetir:

Seguridad:
• Utiliza 6 caracteres o más

Guardar

Eliminar Cuenta

Eliminar

MI CUENTA

Mi Información

Mis Pedidos

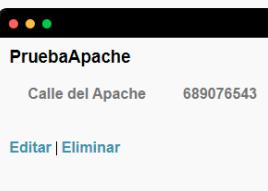
Mis Direcciones

Mis Tarjetas

Mis Comentarios



Mis Direcciones



Mi Información

Mis Pedidos

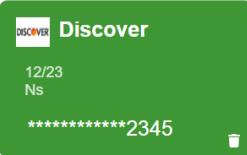
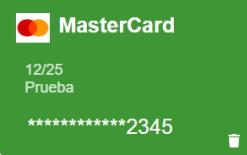
Mis Direcciones

Mis Tarjetas

Mis Comentarios



Mis Tarjetas



Mi Información

Mis Pedidos

Mis Direcciones

Mis Tarjetas

Mis Comentarios

Mis Comentarios



Solo Leveling 5



Tremendo manwha y portada. Por no mencionar al protagonista que esta muy OP

11/05/2022

Mi Información

Mis Pedidos

Mis Direcciones

Mis Tarjetas

Mis Comentarios

Mis Pedidos

Pedido realizado	Total	Enviar a
11 may. 2022	36 €	Dirección no encontrada

[Ver detalles del pedido](#)

Entregado hace un mes



Solo Leveling 5 X1



My Hero Academia n° 30 X1

Buscar

 TuManga

≡

BUSQUEDA: 'SOLO'



Solo Leveling 5
14€ X AGOTADO

Mi Cuenta

- > Mi Cuenta
- > Historial de pedidos
- > Mis Direcciones

Información Legal

- > Aviso Legal
- > Política de cookies
- > Política de privacidad

Contactáanos

- 📞 601 403 189
- ✉️ tumanga@tumanga.com
- Contáctanos para cualquier duda

©Todos los derechos reservados

Carrito en Home

The screenshot shows a mobile application interface for a shopping cart. At the top, there are three icons: a shopping cart with a red notification bubble containing the number 2, a user profile icon, and a menu icon represented by three horizontal lines. Below these, the main content area displays two items:

- Haikyuu!! Vol.45**
Cant: 1
- The Beginning After The End**
Cant: 1

At the bottom of the cart summary, it says **TOTAL 20€**.

Carrito vacío

The screenshot shows a mobile application interface for an empty shopping cart. A large blue header bar contains the word **CARRITO** in white capital letters. Below the header, a message in Spanish reads **Actualmente tu carrito esta vacío**. In the top right corner of the main content area, there is a blue button with the text **Total:0€**.

Actualmente tu carrito esta vacío

Total:0€

Carrito

Total:20€

	Haikyuu!! Vol.45	Cantidad: 1	Precio: 10€	X
	The Beginning After The End	Cantidad: 1	Precio: 10€	X

Finalizar compra

Detalles pedidos y pagar

PAGO

1	Dirección de envío	Adrián Arnaiz Fernández	Cambiar
		Prueba calle 2-Mod	
		789123456	
2	Método de pago	American Express que termina en 7456	Cambiar
3	Revisar productos y opciones de envío		

Entrega estimada: 18 jun 2022

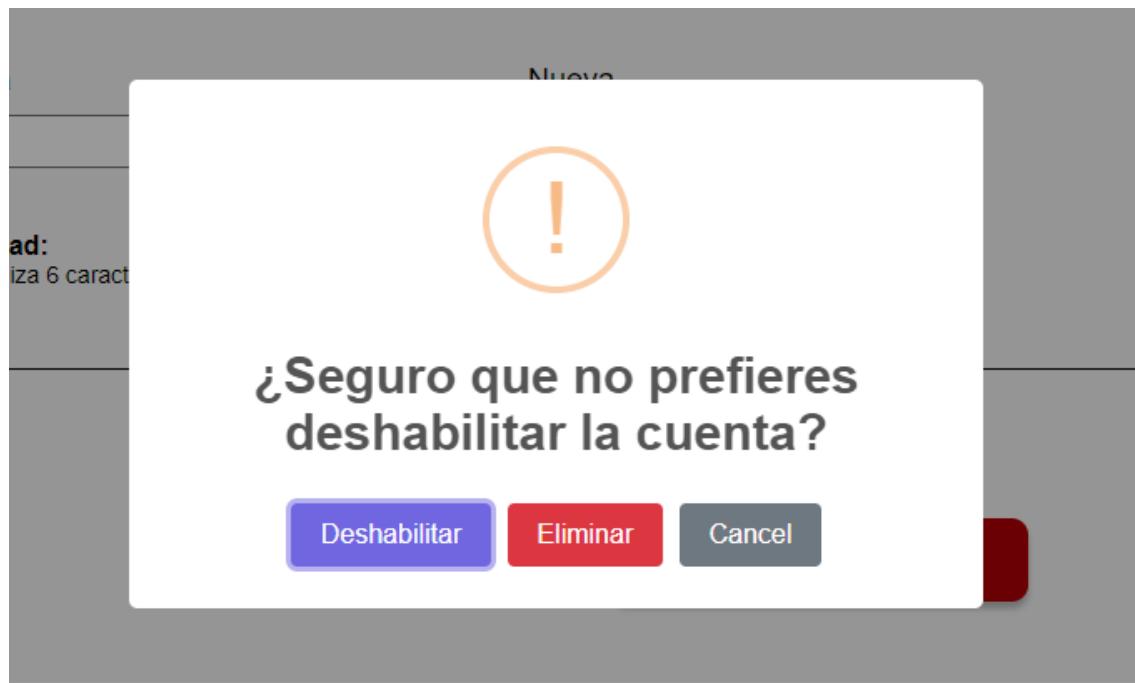
	Haikyuu!! Vol.45	10€	Cant: 1
--	-------------------------	------------	----------------

Finalizar Compra

Resumen del pedido

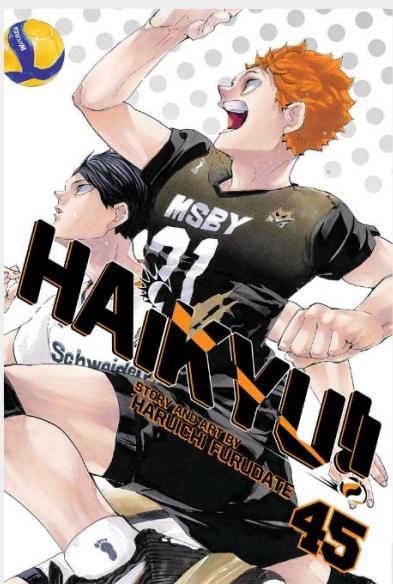
Precio: 20€

Eliminar usuario en Mi cuenta



Detalles producto

PRODUCTO



HAIKYUU!! VOL.45
★★★★★
10 € Disponibilidad: En Stock
[Añadir a la cesta](#)

Editorial: Planeta Cómic
Serie: Haikyuu!
Categorías:
instituto - deportes - comedia

Con el partido de los Adlers y los Black Jackals en pleno apogeo y con años de rivalidad y amistad a flor de piel, Hinata y Kageyama siguen enfrentándose en la cancha. ¿Conseguirá finalmente Hinata declarar una gran victoria sobre Kageyama? ¡No te pierdas el emocionante final! Para el público adolescente.

Comentarios

★★★★★

Escribe tu comentario...

Enviar

Buenisimo

20/05/2022



Tremendo volumen y tremendo mangall! 🎉

Recuperar contraseña

Restablecer la contraseña

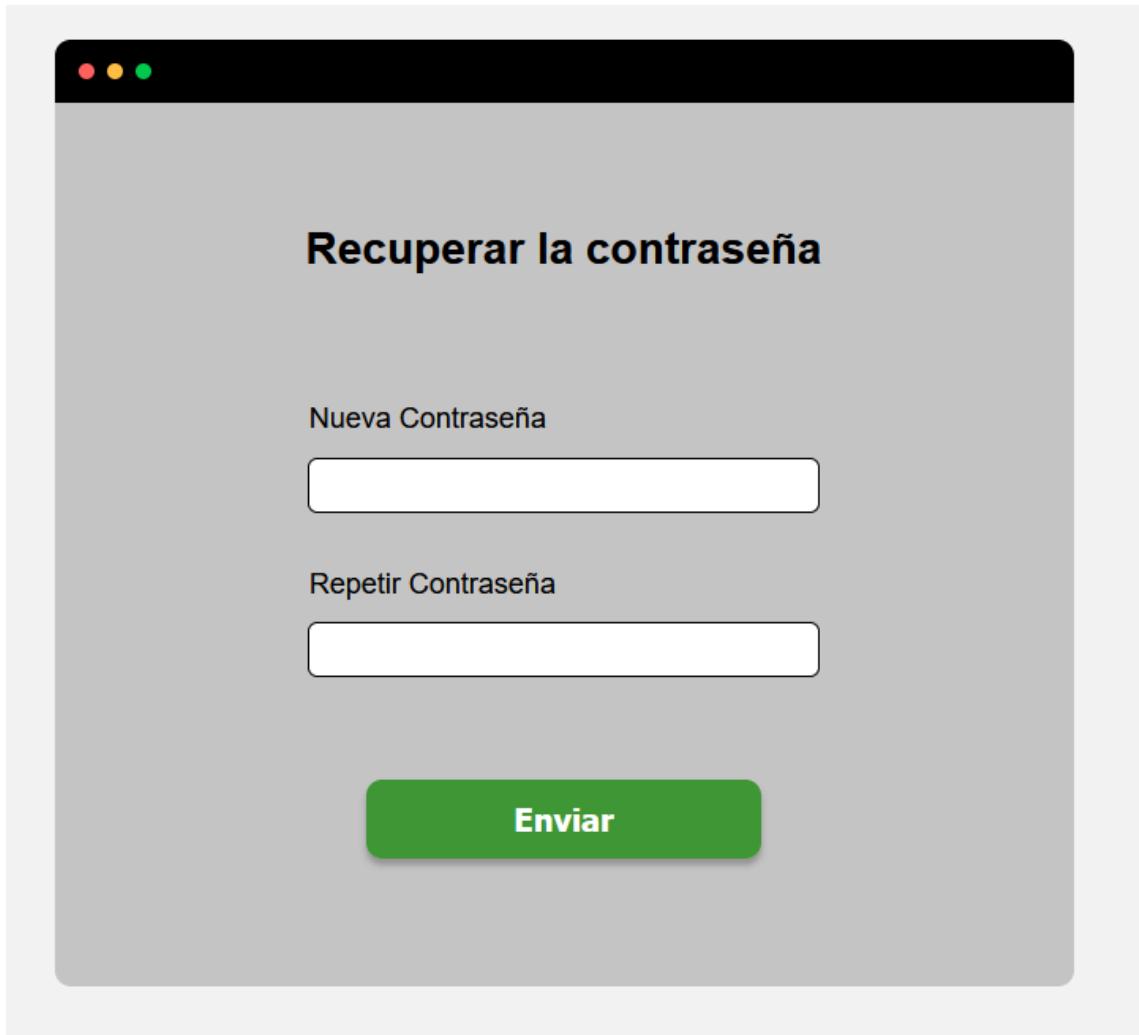
Email

Enviar

Información extra

Recuerde:

- Una vez enviado, se enviará un email para poder restablecerla



Detalles Pedidos

DETALLES PEDIDO

Detalles del pedido

Comprado 26 may. 2022

Dirección de envío	Método de pago
Adrián Arnáiz Fernández Prueba calle 2-Mod Teléfono: 789123456	****7456

Entregado hace 15 días	 Solo Leveling 5	 Haikyuu!! Vol.45
		 Link to item

Plataforma

Admin

- Inicio
- Usuarios
- Productos
- Pedidos
- Roles
- Comentarios

TuManga

VOLVER

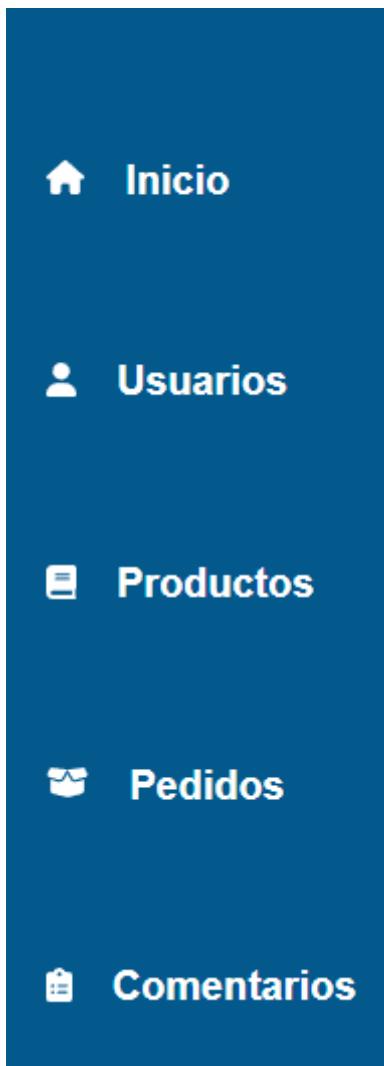
Estadísticas

	Total Pedidos 30		Total Usuarios 9		Ganancias 435€
--	---------------------	--	---------------------	--	-------------------

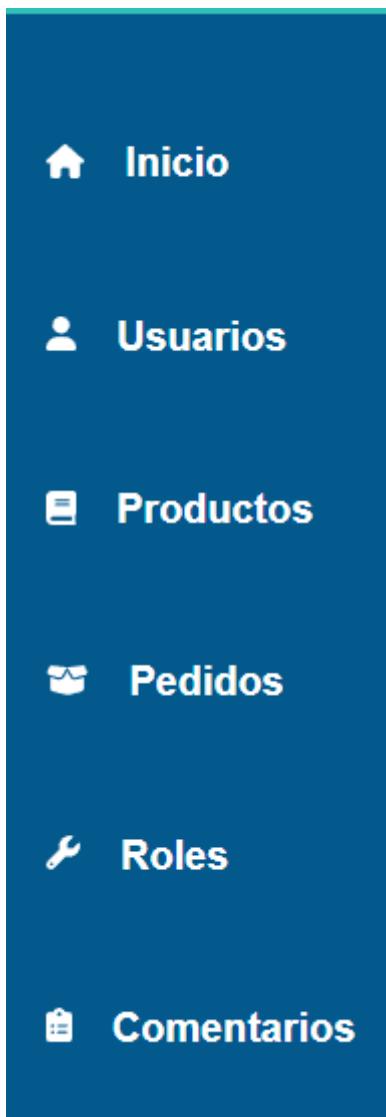
Productos

Solo Leveling 5	Haikyuu! Vol. 45	My Hero Academia nº 30	Camiseta de Goku pequeño	TOKYO REVENGERS 7
-----------------	------------------	------------------------	--------------------------	-------------------

Nav no Admin



Nav Admin



10. Anexo