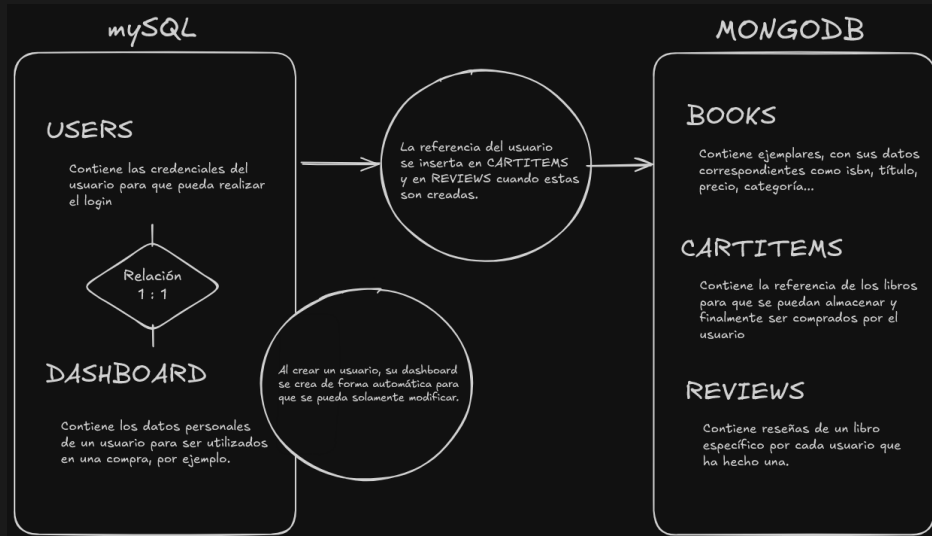




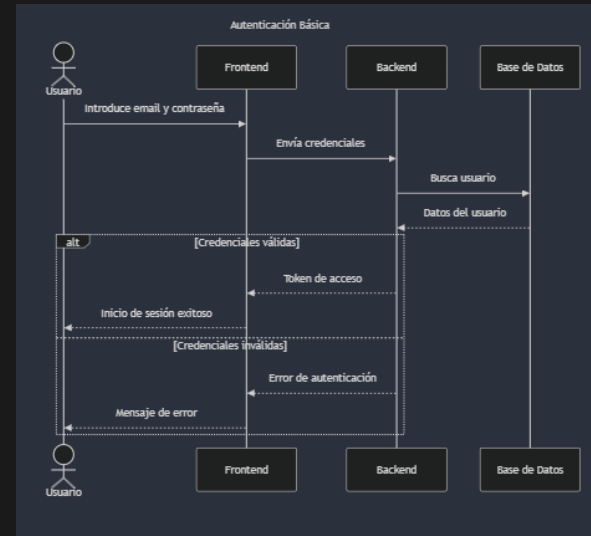
# CULT OF THE READER

## SERVER-SIDE

### BASE DE DATOS



### FLUJO DE AUTENTICACIÓN



- El usuario escribe su email y contraseña.
- El frontend envía esos datos al backend.
- El backend consulta en la base de datos si el usuario existe.
- La base de datos responde con la información del usuario
- Si las credenciales coinciden: Se permite el acceso
- Si las credenciales no coinciden: Se muestra error.

- Users: guarda las credenciales de los usuarios
- Dashboard: guarda los datos personales de los usuarios
- Books: contiene todos los libros con sus respectivos datos
- CartItems: guarda los ejemplares a comprar por el usuario
- Reviews: guarda las reseñas de cada libro hechas por los usuarios



## ESTRUCTURA

server/	
└─ config/	
└─ logger.js	# Logging configuration
└─ mongo.connection.js	# MongoDB connection config
└─ mysql.connection.js	# MySQL connection config
└─ controllers/	
└─ auth.controller.js	# Authentication controller
└─ book.controller.js	# Book operations controller
└─ cartitem.controller.js	# Cart items controller
└─ dashboard.controller.js	# Dashboard controller
└─ review.controller.js	# Reviews controller
└─ docs/	
└─ jest	
└─ book.model.test.js	# Jest tester for CRUD book collection
└─ jmeter	
└─ auth_test.jmx	# Template JMeter
└─ auth.csv	# Template for the JMeter auth test
└─ diagram_UML.png	# Simple UML Diagram
└─ ER_DDBB.png	# Database flow
└─ secuenciaLogin.jpg	# Basic Diagram Sequence Login
└─ middlewares/	
└─ auth.middleware.js	# Authentication middleware
└─ models/	
└─ book.model.js	# Book data model
└─ cartitem.model.js	# Cart item data model
└─ dashboard.model.js	# Dashboard data model
└─ review.model.js	# Review data model
└─ sync.model.js	# Synchronization model
└─ user.model.js	# User data model
└─ mongoDbCollection/	
└─ cultofreader.books.json	# Books collection for MongoDB
└─ routes/	
└─ auth.js	# Authentication routes
└─ book.js	# Book-related routes
└─ cartitem.js	# Cart item routes
└─ .env	# Environment variables
└─ .gitignore	# Git ignore configuration
└─ coffeconfig.md	# Configuration documentation
└─ package-lock.json	# Detailed dependencies
└─ package.json	# Project configuration
└─ README.md	# Project documentation
└─ server.js	# Main server file
└─ swagger.js	# Swagger API configuration

## DOCUMENTACIÓN CON SWAGGER

DEBIDO AL PLAN GRATUITO DE RENDER, PUEDE QUE LA PÁGINA TARDE EN CARGAR



# CLIENT-SIDE

## ESTRUCTURA

```
client-side/
├── cypress                                # test
├── public/
│   ├── favicon-16x16.png                # Small favicon
│   ├── favicon-32x32.png                # Medium favicon
│   └── favicon.ico                       # Main favicon
├── src/
│   ├── assets/
│   │   ├── hero.jpg                    # Main hero image
│   │   └── logo.webp                   # Site logo
│   ├── components/
│   │   ├── Auth/
│   │   │   ├── AuthForm.css            # Authentication form styles
│   │   │   └── AuthForm.jsx            # Authentication form component
│   │   ├── Global/
│   │   │   ├── Cart.css                # Cart styles
│   │   │   ├── Cart.jsx                # Cart component
│   │   │   ├── ErrorMessage.css        # Error message styles
│   │   │   ├── ErrorMessage.jsx        # Error message component
│   │   │   ├── LoadingSpinner.css      # Loading spinner styles
│   │   │   └── LoadingSpinner.jsx       # Loading spinner component
│   │   ├── Home/
│   │   │   ├── BookList.css            # Book list styles
│   │   │   ├── BookList.jsx            # Book list component
│   │   │   ├── Header.css              # Header styles
│   │   │   ├── Header.jsx              # Header component
│   │   │   ├── Hero.css                # Hero section styles
│   │   │   └── Hero.jsx                 # Hero section component
│   │   ├── Product/
│   │   │   ├── BookContent.css         # Book content styles
│   │   │   └── BookContent.jsx         # Book content component
│   │   └── Profile/
│   │       ├── ProfileForm.css          # Profile form styles
│   │       └── ProfileForm.jsx          # Profile form component
│   ├── contexts/
│   │   ├── AuthContext.jsx             # Authentication context
│   │   └── CartContext.jsx             # Shopping cart context
│   ├── hooks/
│   │   └── useCart.jsx                  # Custom cart hook
│   ├── pages/
│   │   ├── Book.jsx                    # Book detail page
│   │   ├── Home.jsx                    # Home page
│   │   ├── Login.jsx                   # Login page
│   │   ├── Profile.jsx                 # User profile page
│   │   └── Register.jsx                # Registration page
│   └── services/
│       └── api.js                       # API services
```

		App.css	# Main application styles
		App.jsx	# Main application component
		index.css	# Global styles
		main.jsx	# React entry point
		.gitignore	# Git ignore configuration
		eslint.config.js	# ESLint configuration
		index.html	# Main HTML file
		package-lock.json	# Dependencies version lock
		package.json	# Project configuration and dependencies
		README.md	# Project documentation
		static.json	# Static files configuration
		vite.config.js	# Vite configuration

## LÓGICA

### PAGES:

- Home: página principal de la web
  - o Header: navegador de la página web
  - o Hero: introducción de la página web
  - o Catálogo: datos simplificados de cada libro y opción de comprar y enlace a su página única
- Book: página específica para un solo libro y sus reseñas
  - o Se visualiza los datos del libro con opción de compra
  - o Se visualiza una sección para insertar una reseña
- Login: autenticación de un usuario
  - o Se usa el email y contraseña
- Register: creación de un usuario en el sistema
  - o Se usa el 'username', email y contraseña
- Profile: zona "privada" de datos personales de un usuario
  - o Permite ver y editar dichos datos

# ¡THAT'S ALL CULTITS!