MARTÍNEZ ABELLÁN, LUCÍA

4273068@alu.murciaeduca.es

CARTAGENITA’S STORIES

Aplicación web para la personalización de cuentos infantiles basados en Cartagena

Proyecto de desarrollo de aplicaciones web - 2425

Convocatoria de octubre a diciembre de 2024



# **CARTAGENITA STORIES: TIENDA ONLINE DE CUENTOS PERSONALIZABLES DE CARTAGENA Y ALREDEDORES MEDIANTE SUSCRIPCIÓN**

Contenido

[**CARTAGENITA STORIES: TIENDA ONLINE DE CUENTOS PERSONALIZABLES DE CARTAGENA Y ALREDEDORES MEDIANTE SUSCRIPCIÓN** 1](#_Toc181984359)

[1. INTRODUCCIÓN 1](#_Toc181984360)

[1.1 Objetivo del Proyecto 1](#_Toc181984361)

[1.2 Descripción General 2](#_Toc181984362)

[2. TECNOLOGÍAS UTILIZADAS 2](#_Toc181984363)

[2.1 Frontend (Cliente) 2](#_Toc181984364)

[2.2 Backend (Servidor) 2](#_Toc181984365)

[2.3 Base de Datos 2](#_Toc181984366)

[2.4 Herramientas de Desarrollo 2](#_Toc181984367)

[3. BASE DE DATOS 3](#_Toc181984368)

[3.1 Modelo Entidad-Relación 3](#_Toc181984369)

[3.2 Descripción de las Tablas 3](#_Toc181984370)

[3.3 Relaciones 4](#_Toc181984371)

[4. Diseño de la marca 5](#_Toc181984372)

# 1. INTRODUCCIÓN

## 1.1 Objetivo del Proyecto

El objetivo principal de este proyecto es desarrollar una aplicación web que permita a los usuarios personalizar cuentos infantiles. La aplicación, llamada "Cartagenita Stories", permite a los usuarios registrarse, iniciar sesión y crear versiones personalizadas de cuentos predefinidos, donde pueden incluir personajes con características específicas.

## 1.2 Descripción General

Cartagenita Stories es una aplicación web que ofrece las siguientes funcionalidades principales:

Registro e inicio de sesión de usuarios

Visualización de cuentos disponibles

Personalización de cuentos con personajes propios

Gestión de perfil de usuario

Visualización de cuentos personalizados creados

La aplicación está construida siguiendo una arquitectura cliente-servidor, donde:

El frontend está desarrollado con React

El backend está implementado con Spring Boot

La base de datos utiliza MySQL

# 2. TECNOLOGÍAS UTILIZADAS

## 2.1 Frontend (Cliente)

* **React**: Framework de JavaScript utilizado para construir la interfaz de usuario.
* **Axios**: Biblioteca para realizar peticiones HTTP al servidor.
* **React Router**: Para la gestión de rutas y navegación en la aplicación.
* **React Bootstrap**: Framework de CSS para el diseño responsivo.
* **HTML5 y CSS3**: Para la estructura y estilos de la aplicación.

## 2.2 Backend (Servidor)

* **Spring Boot**: Framework de Java para el desarrollo del servidor.
* **Spring Security**: Para la gestión de la seguridad y autenticación.
* **JWT (JSON Web Tokens)**: Para la gestión de sesiones de usuario.
* **JPA/Hibernate**: Para el mapeo objeto-relacional con la base de datos.
* **Maven**: Para la gestión de dependencias.

## 2.3 Base de Datos

* **MySQL**: Sistema de gestión de base de datos relacional.

## 2.4 Herramientas de Desarrollo

* **Visual Studio Code**: Como IDE.
* **phpMyAdmin**: Para la gestión y visualización de la base de datos.
* **Git**: Para el control de versiones.

# 3. BASE DE DATOS

## 3.1 Modelo Entidad-Relación

La base de datos está compuesta por las siguientes entidades principales y sus relaciones:

* **usuario**: Almacena la información de los usuarios registrados
* **cuento\_disponible**: Contiene los cuentos base que pueden ser personalizados
* **libro\_personalizado**: Almacena las personalizaciones creadas por los usuarios
* **personaje\_predefinido**: Contiene los modelos de personajes disponibles
* **personaje\_creado**: Almacena los personajes personalizados de cada libro

## 3.2 Descripción de las Tablas

**Usuario:** Esta tabla almacena los datos de registro y acceso de los usuarios.

|  |
| --- |
| CREATE TABLE usuario (  id BIGINT AUTO\_INCREMENT PRIMARY KEY,  nombre VARCHAR(255),  apellidos VARCHAR(255),  email VARCHAR(255),  password VARCHAR(255),  direccion VARCHAR(255),  codigo\_postal VARCHAR(255)  ); |

**Cuento\_disponible**: Contiene los cuentos base que pueden ser personalizados por los usuarios.

|  |
| --- |
| CREATE TABLE cuento\_disponible (  id BIGINT AUTO\_INCREMENT PRIMARY KEY,  titulo VARCHAR(255),  descripcion VARCHAR(255),  imagen\_url VARCHAR(255)  ); |

**Libro\_personalizado**: Almacena las versiones personalizadas de los cuentos creadas por los usuarios.

|  |
| --- |
| CREATE TABLE libro\_personalizado (  id BIGINT AUTO\_INCREMENT PRIMARY KEY,  titulo VARCHAR(255),  fecha\_creacion DATETIME,  usuario\_id BIGINT,  cuento\_id BIGINT,  FOREIGN KEY (usuario\_id) REFERENCES usuario(id),  FOREIGN KEY (cuento\_id) REFERENCES cuento\_disponible(id)  ); |

**Personaje\_predefinido**: Contiene los modelos base de personajes que pueden ser utilizados.

|  |
| --- |
| CREATE TABLE personaje\_predefinido (  id BIGINT AUTO\_INCREMENT PRIMARY KEY,  nombre VARCHAR(255),  tipo VARCHAR(255),  color\_pelo VARCHAR(255),  color\_piel VARCHAR(255),  forma\_pelo VARCHAR(255),  imagen\_url VARCHAR(255)  ); |

**Personaje\_creado**: Almacena los personajes personalizados que forman parte de cada libro.

|  |
| --- |
| CREATE TABLE personaje\_creado (  id BIGINT AUTO\_INCREMENT PRIMARY KEY,  libro\_id BIGINT,  personaje\_id BIGINT,  personaje\_nombre VARCHAR(255),  FOREIGN KEY (libro\_id) REFERENCES libro\_personalizado(id),  FOREIGN KEY (personaje\_id) REFERENCES personaje\_predefinido(id)  ); |

## 3.3 Relaciones

**Usuario - Libro Personalizado (1:N)**

* Un usuario puede crear múltiples libros personalizados
* Cada libro personalizado pertenece a un único usuario

**Cuento Disponible - Libro Personalizado (1:N)**

* Un cuento disponible puede ser base de múltiples libros personalizados
* Cada libro personalizado está basado en un único cuento disponible

**Libro Personalizado - Personaje Creado (1:N)**

* Un libro personalizado puede tener múltiples personajes
* Cada personaje creado pertenece a un único libro

**Personaje Predefinido - Personaje Creado (1:N)**

* Un personaje predefinido puede ser base para múltiples personajes creados
* Cada personaje creado se basa en un único personaje predefinido

4. Diseño de la marca

## 4.1. Logo

El logo representa a la marca mediante un bocadillo de cómic que simboliza la narración de historias y la letra “C” que representa tanto “Cartagena” como “Cuentos”.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| Principal | Monocromático | Negativo | Simple |

## 4.1. Paleta de colores

Se ha empleado una paleta de colores pastel.

|  |  |
| --- | --- |
|  | Colores principales:   * Azul pastel: #A8D8EA (Logo y elementos principales) * Rosa pastel: #FFB9B9 (Botones y elementos de acción) |
| Colores secundarios:   * Verde menta: #D5ECC2 (Elementos de éxito y confirmación) * Lila pastel: #E2D1F9 (Elementos decorativos) * Melocotón: #FFD4B8 (Elementos decorativos) |
| Colores de fondo y texto:   * Blanco crema: #FFF8F0 (Fondos de secciones) * Gris muy suave: #F7F7F7 (Fondos de contenedores) * Gris oscuro: #5A5A5A (Textos) |

## 4.2. Tipografía

Título principal de la aplicación:

* H1: Comic Sans MS 32px Bold #A8D8EA (Azul pastel principal)

Títulos de secciones principales:

* H2: Comic Sans MS 28px Bold #FFB9B9 (Rosa pastel)

Subtítulos de secciones:

* H3: Comic Sans MS 24px Bold #E2D1F9 (Lila pastel)

Títulos de cards y elementos destacados:

* H4: Comic Sans MS 20px Bold FFD4B8 (Melocotón)

Subtítulos en cards y elementos secundarios:

* H5: Comic Sans MS 18px Bold #D5ECC2 (Verde menta)

## Despliegue en RailWay

### Importación de la base de datos a RailWay empleando MySQL Workbench

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Se comprueba que se ha importado correctamente, ejecutanto la consulta: ‘SHOW TABLES’

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

También aparece en la arquitectura de Rail Way

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

### Configurar variables de entorno en Railway

Se recomienda no incluir las credenciales directamente en el código, por lo tanto, las variables de entorno se configuran en railway y se accede a ellas en la aplicación.

En SpringBoot en lugar de usar las credenciales directamente, se emplean las variables de entorno en application.properties

Texto

Descripción generada automáticamente

### Configuración backend

#### Configuración proceso de construcción

Configurar el comando java -jar en Railway para que sepa como ejecutar la aplicación.

En Railway, sobre el proyecto, dentro de Settings, bajo Deploy, se encuentra Custom Start Command:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

## Usar variables de entorno

### Configurar variables de entorno en application.properties en el backend (SpringBoot)

Texto

Descripción generada automáticamente

Se genera un dominio en Railway

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Como frontend y backend se encuentran bajo el mismo dominio, para configurar el servidor y que distinga las rutas de cada uno:

* Las solicitudes a / (y rutas de React como **/home**) sirven los archivos del **frontend**.
* Las solicitudes a **/api/...** se redirigen al **backend**.