

## Book Reviewer

Este proyecto consiste en una aplicación de reseñas de libros que permite a los usuarios explorar, buscar y reseñar libros. Las funcionalidades principales incluyen la navegación por una lista de libros, la búsqueda de libros por título, autor o categoría, y la visualización de detalles de cada libro. Además, los usuarios pueden dejar reseñas, incluyendo una calificación y un comentario en texto, y ver las reseñas de otros usuarios.

### Funcionalidades

#### Página de Inicio:

Muestra una lista de libros disponibles. Permite a los usuarios buscar libros por título, autor o categoría. Detalles del Libro:

Los usuarios pueden ver detalles específicos de un libro, como su título, autor, categoría y resumen. Reseñas de Libros:

Los usuarios autenticados pueden dejar reseñas para los libros, incluyendo una calificación del 1 al 5 y un comentario en texto. Las reseñas pueden ser vistas por otros usuarios, ordenadas por las más recientes. Autenticación y Autorización:

Los usuarios pueden registrarse para obtener una cuenta, iniciar sesión con sus credenciales y cerrar sesión. Solo los usuarios autenticados pueden dejar reseñas para un libro. ## Despliegue

para desplegar de manera local

```
npm start en la carpeta frontend
```

```
dotnet run build en la carpeta backend
```

## Despliegue de un Proyecto React en Vercel

### Prerrequisitos

Antes de comenzar, asegúrese de tener lo siguiente:

1. Un proyecto React listo para ser desplegado.
2. Una cuenta en Vercel.
3. Git instalado y configurado en su máquina.
4. El proyecto React almacenado en un repositorio de GitHub, GitLab o Bitbucket.

### Paso 1: Crear una Cuenta en Vercel

1. Acceda a Vercel y regístrese o inicie sesión con su cuenta existente.

## Paso 2: Importar el Proyecto

1. Una vez iniciada la sesión, haga clic en el botón **New Project**.
2. Conecte su cuenta de GitHub, GitLab o Bitbucket a Vercel si aún no lo ha hecho.
3. Seleccione el repositorio que contiene su proyecto React.

## Paso 3: Configurar el Proyecto

1. Vercel detectará automáticamente que su proyecto es una aplicación React y sugerirá las configuraciones predeterminadas.
2. Revise y confirme la configuración predeterminada. Normalmente, no es necesario cambiar nada, pero asegúrese de que el comando de construcción (**npm run build** o **yarn build**) y la carpeta de salida (**build**) sean correctos.

## Paso 4: Desplegar el Proyecto

1. Haga clic en el botón **Deploy** para iniciar el despliegue.
2. Vercel comenzará a construir y desplegar su proyecto. Este proceso puede tomar unos minutos.

## Paso 5: Revisar y Confirmar el Despliegue

1. Una vez completado el despliegue, verá un mensaje de éxito junto con un enlace a su aplicación desplegada.
2. Haga clic en el enlace para ver su aplicación React en vivo.

## Paso 6: Configuración Adicional (Opcional)

### Configuración de Dominio Personalizado

1. Si desea utilizar un dominio personalizado, acceda a la sección **Domains** en el tablero de su proyecto en Vercel.
2. Añada su dominio personalizado y siga las instrucciones para configurarlo.

### Configuración de Variables de Entorno

1. Si su aplicación React requiere variables de entorno, puede configurarlas en la sección **Environment Variables** del tablero de su proyecto en Vercel.
2. Añada las variables necesarias y sus valores correspondientes.

### Configuración de Redireccionamientos y Reescrituras

1. Si necesita redireccionar o reescribir URLs, acceda a la sección **Redirects** o **Rewrites** en el tablero de su proyecto.
2. Configure las reglas según sea necesario.

## Actualización del Proyecto

Cada vez que realice cambios en su proyecto y los envíe a su repositorio de Git, Vercel construirá y desplegará automáticamente la nueva versión de su aplicación.

## Despliegue de un Proyecto Backend .NET en smarterasp.net

### Prerrequisitos

Antes de comenzar, asegúrese de tener lo siguiente:

1. Un proyecto backend .NET listo para ser desplegado.
2. Una cuenta en SmartASP.NET.
3. Visual Studio 2022 instalado en su máquina.
4. El proyecto .NET almacenado en un repositorio de GitHub, GitLab o Bitbucket.

### Paso 1: Crear una Cuenta en SmartASP.NET

1. Acceda a SmartASP.NET y regístrese o inicie sesión con su cuenta existente.
2. Complete el proceso de registro y active su cuenta si es necesario.

### Paso 2: Configurar un Nuevo Sitio Web

1. Una vez iniciada la sesión, acceda a su panel de control.
2. Navegue a la sección **Websites** y haga clic en **Add Website**.
3. Complete la información requerida para crear un nuevo sitio web, incluyendo el nombre del sitio y el dominio (puede usar un subdominio temporal proporcionado por SmartASP.NET).

### Paso 3: Descargar el Archivo de Publicación XML

1. En el panel de control de su nuevo sitio web, busque la opción para descargar el perfil de publicación.
2. Descargue el archivo XML de configuración de publicación para Visual Studio.

### Paso 4: Publicar el Proyecto desde Visual Studio 2022

1. Abra su proyecto .NET en Visual Studio 2022.
2. Haga clic con el botón derecho en el proyecto en el **Solution Explorer** y seleccione **Publish**.
3. En la ventana de publicación, seleccione **Import Profile**.

4. Importe el archivo XML de configuración de publicación que descargó de SmartASP.NET.
5. Configure cualquier otra opción necesaria en la configuración de publicación.
6. Haga clic en **Publish** para desplegar su proyecto directamente en SmartASP.NET.

### **Paso 5: Configurar la Base de Datos (Opcional)**

1. Si su proyecto .NET utiliza una base de datos, acceda a la sección **Databases** en el panel de control de SmartASP.NET.
2. Cree una nueva base de datos MySQL o SQL Server, según lo que utilice su proyecto.
3. Configure la cadena de conexión en su proyecto .NET para apuntar a la base de datos creada.

### **Paso 6: Configurar el Dominio (Opcional)**

1. Si desea utilizar un dominio personalizado, vaya a la sección **Domains** en el panel de control de SmartASP.NET.
2. Añada su dominio personalizado y siga las instrucciones para configurarlo.
3. Actualice las configuraciones DNS de su dominio para apuntar a los servidores de SmartASP.NET.

### **Paso 7: Verificar y Probar el Despliegue**

1. Una vez que el proyecto se haya publicado correctamente, acceda a la URL de su sitio web (o dominio personalizado) para verificar que su proyecto backend .NET esté funcionando correctamente.
2. Realice pruebas para asegurarse de que todas las funcionalidades de su aplicación estén operativas.

## **Despliegue de una Base de Datos MySQL en Railway**

Este documento proporciona una guía sobre cómo desplegar una base de datos MySQL en Railway sin necesidad de configurar explícitamente un Dockerfile.

### **Prerrequisitos**

Antes de comenzar, asegúrese de tener lo siguiente:

1. Una cuenta en Railway.
2. Los detalles de conexión necesarios para su base de datos MySQL (como la cadena de conexión, nombre de usuario y contraseña).

### **Paso 1: Crear una Cuenta en Railway**

1. Acceda a Railway y regístrese o inicie sesión con su cuenta existente.

### **Paso 2: Crear un Nuevo Proyecto en Railway**

1. Una vez iniciada la sesión, haga clic en el botón **New Project**.

### **Paso 3: Configurar la Base de Datos MySQL en Railway**

1. En la sección **Services**, haga clic en **Add a Service**.
2. Seleccione MySQL como el tipo de servicio.

### **Paso 4: Configurar Variables de Entorno y Detalles de Conexión**

1. Complete los detalles de configuración requeridos para su base de datos MySQL, como el nombre de la base de datos, nombre de usuario y contraseña.
2. Guarde la configuración.

### **Paso 5: Desplegar el Proyecto**

1. Suba su proyecto a un repositorio de GitHub, GitLab o Bitbucket.
2. En Railway, seleccione su repositorio y la rama correspondiente.
3. Haga clic en **Deploy**.

### **Paso 6: Configurar Variables de Entorno en Railway (si es necesario)**

1. Si su proyecto requiere variables de entorno adicionales, puede configurarlas en la sección **Settings** del proyecto en Railway.

### **Paso 7: Verificar y Probar la Conexión**

1. Una vez desplegado, vaya a la sección **Connect** de su proyecto en Railway.
2. Encuentre los detalles de conexión para su base de datos MySQL (como la cadena de conexión) y úselos para conectar su aplicación.

### **Paso 8: Verificar el Despliegue**

1. Asegúrese de que su aplicación puede conectarse correctamente a la base de datos MySQL desplegada en Railway.
2. Realice pruebas para verificar que la base de datos está operativa y que su aplicación funciona correctamente.

## Desarrollo

Tecnologías Utilizadas Frontend:

React: Para la creación de componentes y la gestión del estado de la aplicación.

REST APIs: Para la comunicación entre el frontend y el backend. Backend:

MySQL: Como base de datos relacional para almacenar la información de los libros y las reseñas. .NET: Para desarrollar la lógica del servidor y las API necesarias para el funcionamiento de la aplicación. Implementación:

Vercel: Para el alojamiento web sin servidor. Docker: Para el despliegue y alojamiento de la base de datos Mysql SmartASP.net: Para el alojamiento de el backend .NET CI/CD con GitHub: Para automatizar el proceso de integración y despliegue continuo.

## Tech Stack

**Client:** React, TailwindCSS

**Server:** .NET

**Database** Mysql