**Documentación del Proyecto: Citric Alexandria**

**1. Introducción**

**Título del Proyecto**: Citric Alexandria  
**Descripción del Proyecto**: Citric Alexandria es una página web dedicada a la compra de videojuegos. Los usuarios pueden registrarse, explorar una amplia colección de juegos y realizar compras directamente desde la plataforma.

**2. Requisitos del Sistema**

**Requisitos Funcionales**

1. **Registro de Usuarios**: Los usuarios pueden crear una cuenta en la página web proporcionando un nombre de usuario único, una dirección de correo electrónico válida y una contraseña segura.
2. **Inicio de Sesión**: Los usuarios pueden iniciar sesión en sus cuentas utilizando su nombre de usuario y contraseña.
3. **Exploración de Juegos**: Los usuarios pueden navegar por el catálogo de juegos disponibles, filtrar por categorías y ver detalles como el nombre, la categoría y el precio de cada juego.
4. **Compra de Juegos**: Los usuarios pueden seleccionar juegos del catálogo, añadirlos al carrito de compras y completar la transacción de compra proporcionando la información necesaria de pago.

**Requisitos No Funcionales**

1. **Rendimiento**: La página web debe ser rápida y eficiente en la carga de juegos y en las transacciones de compra.
2. **Seguridad**: Debe implementarse un sistema robusto de autenticación y protección de datos de usuario.
3. **Usabilidad**: La interfaz de usuario debe ser intuitiva y fácil de navegar.
4. **Escalabilidad**: El sistema debe ser capaz de manejar un crecimiento en el número de usuarios y juegos sin degradación del rendimiento.

**3. Diagrama Entidad-Relación (ER)**

1. **Usuario**:
   * Atributos: username, email, password.
   * Nota: Se planea expandir los atributos del usuario en futuras iteraciones del proyecto.
   * Relaciones: Este modelo representa a los usuarios registrados en la página web.
2. **Juego**:
   * Atributos: name, category, price, game\_id.
   * Relaciones: Este modelo representa los juegos disponibles en la página web.
3. **Compra**:
   * Atributos: user (clave foránea a Usuario), game (clave foránea a Juego), fecha (fecha de la compra u otro atributo relevante).
   * Relaciones: Esta entidad representa la relación entre usuarios y juegos comprados.

**4. Descripción de Entidades y Relaciones**

**Usuario**

* **Atributos**: username, email, password.
* **Notas**: Se planea expandir los atributos del usuario en futuras iteraciones del proyecto.

**Juego**

* **Atributos**: name, category, price, game\_id.

**Compra** (implementación en futuras iteraciones)

* **Atributos**: user (clave foránea a Usuario), game (clave foránea a Juego), fecha (fecha de la compra u otro atributo relevante).

**5. Modelos en el Sistema**

**Modelo Relacional**

* **Usuario**: Modelo de Django para representar usuarios registrados en la página web.
* **Juego**: Modelo de Django para representar juegos disponibles en la página web.
* **Compra**: Modelo de Django para representar la relación entre usuarios y juegos comprados.

**6. Descripción del Sistema**

**Arquitectura del Sistema**

El sistema utiliza una arquitectura cliente-servidor:

* **Cliente (Frontend)**:
  + Desarrollado en React.js con soporte de Redux para la gestión del estado.
  + Utiliza Bootstrap para el diseño responsivo y componentes UI.
  + React Router para la navegación entre páginas dentro de la aplicación.
  + Vite como el bundler y servidor de desarrollo para optimizar el rendimiento.
  + HTML y CSS estándar para la estructura y estilización de la interfaz de usuario.
* **Servidor (Backend)**:
  + Implementado con Django y SQLite, cumpliendo con las especificaciones del proyecto para el desarrollo del backend.

**7. Casos de Uso del Sistema**

**Caso de Uso 1: Registro de Usuario**

* **Actor Principal**: Usuario
* **Descripción**: El usuario crea una cuenta en la página web proporcionando un nombre de usuario único, una dirección de correo electrónico válida y una contraseña segura.

**Caso de Uso 2: Inicio de Sesión**

* **Actor Principal**: Usuario
* **Descripción**: El usuario inicia sesión en su cuenta utilizando su nombre de usuario y contraseña.

**Caso de Uso 3: Exploración de Juegos**

* **Actor Principal**: Usuario
* **Descripción**: El usuario navega por el catálogo de juegos disponibles, filtrando por categorías y explorando detalles como el nombre, la categoría y el precio de cada juego.

**Caso de Uso 4: Compra de Juegos**

* **Actor Principal**: Usuario
* **Descripción**: El usuario selecciona un juego disponible en el catálogo, lo añade al carrito de compras y completa la transacción de compra proporcionando la información necesaria de pago.

**Caso de Uso 5: Gestión de Perfil (futuro)**

* **Actor Principal**: Usuario
* **Descripción**: En futuras implementaciones, el usuario podrá acceder a su perfil para ver y editar su información personal, como su nombre de usuario, dirección de correo electrónico y contraseña.

**Caso de Uso 6: Administración de Juegos (futuro)**

* **Actor Principal**: Administrador del Sistema
* **Descripción**: El administrador del sistema puede agregar, modificar o eliminar juegos del catálogo disponible para los usuarios.

**Instrucciones de Uso**

Para iniciar el proyecto, sigue estos pasos:

**Backend**

1. Abre la consola en la carpeta userauth, que es un hijo directo de la carpeta backend.
2. Ejecuta el siguiente comando para iniciar el servidor de desarrollo de Django:

Copy code

python manage.py runserver

**Frontend**

1. Abre la consola en la carpeta citric alexandria, que es un hijo directo de la carpeta frontend.
2. Ejecuta el siguiente comando para iniciar el servidor de desarrollo de React:

arduino

Copy code

npm run dev

**Nota Adicional**

* Si por alguna razón los servidores de Django y React no se conectan correctamente, puedes ajustar la ruta del servidor Django en el archivo App.jsx de la carpeta citric alexandria. Busca la constante URLBACKEND y asegúrate de que apunte a la dirección correcta del servidor Django.