

Temática del proyecto

Tienda de videojuegos

Secciones de la aplicación

- Página principal con 4 artículos destacados
- Página de tienda
 - Página con la información de artículos individuales.
- Página de carrito de la compra
- Página de gestión (solo disponible para admins)
- Formularios de login y registro
- Api para la subida y eliminación de imágenes del servidor

Usuario disponibles

La aplicación cuenta con dos usuarios por defecto que permiten probar los distintos roles disponibles

- Usuario administrador
 - Email: admin@admin.com
 - Contraseña: adminadmin
- Usuario regular
 - Email: usuario@gmail.com
 - Contraseña: 1234

Operaciones CRUD

Dentro del panel de administrador, está disponible un panel con todas las operaciones CRUD para los artículos de la tienda. En el caso de los usuarios, el administrador podrá eliminar usuarios o modificar sus roles. Para el resto de elementos variables de la aplicación (categorías, etiquetas y géneros de videojuegos), el administrador podrá crear nuevos o eliminar los existentes.

Componentes

Los componentes de la aplicación son los siguientes:

- FormArticulos
 - Componente para crear o editar nuevos artículos.
 - Creado para reducir la concentración de código en la propia página de gestión de artículos.
- ImageUploader
 - Componente para gestionar la subida de imágenes.
 - Creado para separar la lógica de subida de imágenes del formulario principal.

- **TablaArticulos**
 - Componente para visualizar los artículos disponibles desde el panel de administración. También permite seleccionar un artículo para su edición o eliminarlo.
 - Creado para reducir la concentración de código en la propia página de gestión de artículos.
- **TablaUsuarios**
 - Componente para visualizar los usuarios de la aplicación, así como la eliminación de usuarios o el cambio de roles de un usuario. No permite cambiar el rol del usuario registrado, para evitar que la aplicación se quede sin administradores.
 - Creado para reducir la concentración de código en la propia página de gestión de usuarios.
- **FormGestion**
 - Formulario genérico para crear nuevos elementos para la base de datos
 - Creado para evitar repetir código para la gestión de elementos simples (etiquetas, categorías, géneros)
- **TablaGestion**
 - Tabla genérica para visualizar elementos para la base de datos o eliminarlos.
 - Creado para evitar repetir código para la gestión de elementos simples (etiquetas, categorías, géneros)
- **WithAuth**
 - Componente envoltorio para gestionar la autorización por roles en las páginas que lo requieran.
 - Creado para el bloqueo de rutas a partir del rol del usuario logueado.
- **Articulo**
 - Componente para representar un artículo en la tienda.
 - Creado para evitar concentración de código en la página de la tienda, así como para ser reutilizado para mostrar los productos destacados en la página de inicio.
- **Filtro**
 - Componente para el filtrado de artículos.
 - Creado para evitar concentración de código y para ser reutilizado tanto en la tabla de gestión de artículos como en la propia tienda
- **Navbar**
 - Componente que actúa de header de la aplicación.
 - Creado para poner en el layout.

Hooks

- `useState`
 - Usado para gestionar variables que pueden cambiar en tiempo de ejecución, afectando al contenido visible en la página.
- `useEffect`
 - Usado para hacer llamadas a la base de datos al inicializar la página o llamar a determinadas funciones cuando ciertos valores cambian.
- `useContext`
 - Usado para crear contextos que comparten información necesaria para toda la aplicación, como la información del usuario logueado o el estado del carrito.

Diagrama de navegación

- Página Principal
 - Registro (/registro)
 - Login (/login)
 - Tienda (/articulos)
 - Artículo (/articulos/[id])
 - Carrito (si estás logueado) (/cart)
 - Gestión (si eres admin) (/gestion)
 - Gestión de artículos (/gestion/articulos)
 - Gestión de categorías (/gestion/categorias)
 - Gestión de etiquetas (/gestion/etiquetas)
 - Gestión de géneros (/gestion/generos)
 - Gestión de usuarios (/gestion/usuarios)
- API
 - PUT imagen (/api/imagen/[nombre])
 - DELETE imagen (/api/imagen/[nombre])

Validaciones

Validaciones básicas en los formularios que comprueban que todos los campos requeridos están correctamente cubiertos.

Persistencia

Usado un json-server para el almacenamiento de datos y una api hecha con NEXTjs para la subida y eliminación de imágenes.