



PROYECTO DAW: NEXOGAMER

Aplicación de intercambio de videojuegos



14 DE DICIEMBRE DE 2023

JOSE VICENTE ROMERO VEIGA

1. INTRODUCCIÓN

NexoGamer nace como una plataforma dinámica dedicada al intercambio y venta de productos relacionados con el mundo de los videojuegos incluyendo, juegos, accesorios y plataformas. La misión primordial es establecer un entorno seguro y accesible donde cualquier aficionado al mundo gamer pueda comprar, vender e intercambiar sus artículos con otros usuarios.

La inspiración de NexoGamer surgió de un recuerdo personal: de niño visitaba una tienda de videojuegos en las galerías que hay en el parque de San Lázaro (Ourense), donde aparte de vender todo tipo de artículos relacionados con el mundo del videojuego, ofrecían la posibilidad de probar juegos, jugar con otras personas, y lo que a mí me fascinaba, la posibilidad de intercambiar juegos por una pequeña tarifa. Este recuerdo es la base de lo que NexoGamer aspira a recrear: un espacio comunitario e interactivo para los aficionados a los videojuegos.

La funcionalidad central de Nexogamer permite a los usuarios registrarse y publicar detalles de los productos que quieren vender o intercambiar. Cada anuncio es acompañado de una descripción detallada, imagen, precio o términos de intercambio, plataforma a la que pertenece, género...

Además, NexoGamer, tiene la intención de crear una comunidad activa alrededor del mundo de los videojuegos. En un futuro se implementará un foro, un sistema de valoración y comentarios de los usuarios, valorar las transacciones para intentar crear un ecosistema de intercambio y venta seguro.

2.OBJETIVOS

2.1 Objetivo principal:

Establecer un espacio digital para la compra, venta e intercambio de productos relacionados con el mundo de los videojuegos. NexoGamer busca ser la aplicación a la que todo el mundo acuda cuando quiera realizar cualquiera de las transacciones comentadas.

2.2 Objetivos secundarios:

1. Fomentar la interacción comunitaria: recrear un espacio comunitario donde los usuarios no solo realicen transacciones, sino también compartir experiencias, consejos...con otros aficionados.
2. Transparencia y confianza en las transacciones: Asegurar que cada anuncio en la plataforma sea claro y detallado, incluyendo descripción completa, imágenes reales, precio justo y términos de intercambio, para fomentar una atmósfera de confianza entre los usuarios.
3. Desarrollo de una plataforma integral: Dado que el tiempo para el proyecto es limitado no se pueden crear todas las funciones que deseó, pero esta plataforma incluirá un foro, un sistema de valoraciones, comentarios, noticias, guías, eventos...y todo lo relacionado con el mundo del videojuego en una misma plataforma.
4. Evolución constante: a través del feedback, buscar nuevas características y mejoras que hagan crecer la plataforma.

3. ESTUDIO DE MERCADO

NexoGamer está dirigido a un público muy diverso. Este mercado incluye desde jóvenes aficionados que buscan las últimas novedades al mejor precio hasta coleccionistas en busca de joyas antiguas. Uno de los aspectos clave del mercado objetivo son los usuarios que buscan alternativas económicas para acceder a juegos o accesorios que por algún motivo le son inalcanzables a precio de artículo nuevo.

En cuanto a la competencia, no encontré una plataforma que sea similar a la base de NexoGamer. Una comparativa podría ser Wallapop, pero que exclusivamente sea de videojuegos no hay ninguna a día de hoy, o por lo menos de habla hispana.

Yo creo que el mercado de segunda mano ha estado en aumento los últimos años, especialmente desde la aparición de la plataforma mencionada anteriormente. A eso sumado al interés creciente que hay por lo retro y la vuelta al pasado, me parece que esta plataforma se aprovecha de todas estas tendencias.

Fortalezas, debilidades, oportunidades y amenazas:

- Fortalezas: NexoGamer cuenta con una interfaz amigable, muy intuitiva y fácil de utilizar.
- Debilidades: Al ser una plataforma nueva es complicado construir desde cero una base de usuarios que permita crecer.
- Oportunidades: Como comenté antes, estamos ante un mercado creciente de reutilizar las cosas y eso ofrece una buena oportunidad.
- Amenazas: alguna regulación sobre el mercado de segunda mano que influya de manera significativa o que surja alguna plataforma con un fuerte apoyo.

Como estrategia de Marketing, lo básico es fuertes campañas en redes sociales, colaboración con creadores de contenido del mundo de los videojuegos...

3.1 PRESUPUESTO:

Ya que NexoGamer es un proyecto de final de grado, lo desarrolle utilizando recursos y herramientas de código abierto. La inversión en el proyecto fue mínima ya que adquirí un dominio y un hosting, cuyo coste fue de unos 10€.

La compra del dominio y el hosting, se realizó con la intención de desplegar el proyecto en un entorno en vivo, pero como cambié de tecnologías, el hosting contratado no disponía de soporte para el Backend, por lo que finalmente me decidí a presentarlo en local.

En conclusión, como el proyecto es con fines académicos, no establecí ningún tipo de presupuesto, ya que el enfoque ha sido aprender y mejorar los conocimientos adquiridos durante el ciclo.

4. DOCUMENTACIÓN TÉCNICA

4.1 Arquitectura y Estructura

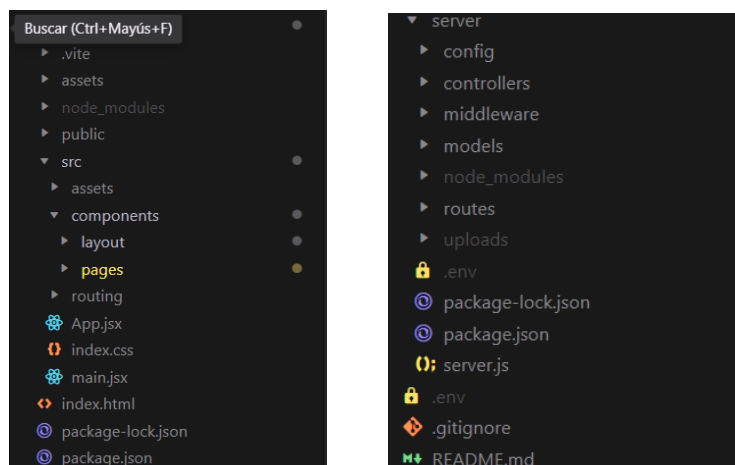
NexoGamer está estructurado en dos directorios principales: client para el Frontend y server para el Backend, reflejando una arquitectura de aplicación web separada.

1. Frontend (Client):

- Carpeta public: Contiene recursos estáticos como imágenes.
- Carpeta src:
 - components: Alberga los componentes de React, divididos en pages para las distintas páginas de la aplicación y layouts para los diseños generales.
- routing: Encargada de la gestión de rutas de la aplicación.
- Archivos principales como App.js y el archivo de estilos globales.

2. Backend (Server):

- Carpeta config: Configuración de la base de datos y Sequelize.
- Carpeta controller: Contiene los controladores para manejar la lógica de negocio.
- Carpeta middleware: Funciones intermedias para el manejo de solicitudes, respuestas y JWT.
- Carpeta models: Define los modelos de datos para artículos y usuarios.
- Carpeta routes: Gestiona las rutas del servidor para las distintas operaciones de la API.



4.1.1 FLUJO DE DATOS Y SOLICITUDES

Las solicitudes desde el frontend se realizan utilizando fetch, interactuando con el backend para operaciones relacionadas con usuarios y artículos. El backend procesa estas solicitudes, interactuando con la base de datos a través de Sequelize para recuperar o almacenar información.

Ejemplo:

```
try {
  const responseArticulo = await fetch(`http://localhost:3000/api/articulos/${id}`);
  if (responseArticulo.ok) {
    const dataArticulo = await responseArticulo.json();
    setArticulo(dataArticulo);

    const responseUsuario = await fetch(`http://localhost:3000/api/usuarios/${dataArticulo.id_usuario}`);
    if (responseUsuario.ok) {
      const dataUsuario = await responseUsuario.json();
      setUsuario(dataUsuario);
    } else {
      console.error('Error al obtener los detalles del usuario');
    }
  } else {
    console.error('Error al obtener el artículo');
  }
}
```

Este ejemplo muestra un proceso de dos pasos en el manejo de solicitudes y el flujo de datos entre el frontend y el backend:

1. Obtención de Detalles de un Artículo:
 - El frontend realiza una solicitud fetch al backend para obtener los detalles de un artículo específico por su id.
 - Si la solicitud es exitosa (responseArticulo.ok), los datos del artículo se obtienen en formato JSON y se almacenan en el estado setArticulo(dataArticulo).
 - Si la solicitud falla, se registra un mensaje de error en la consola.
2. Obtención de Información del Usuario Asociado al Artículo:
 - Tras obtener con éxito los detalles del artículo, se realiza una segunda solicitud fetch al backend para obtener información sobre el usuario que publicó el artículo.
 - Si esta solicitud es exitosa (responseUsuario.ok), se obtienen los datos del usuario en formato JSON y se actualiza el estado setUsuario(dataUsuario).
 - En caso de error, se registra un mensaje de error correspondiente en la consola.

4.2 METODOLOGÍA DE DESARROLLO

El proyecto fue organizado en varios sprints, guiándome principalmente por proyectos que hice en algunos cursos.

SPRINT 0: Preparación:

- Configuración del entorno de desarrollo para utilizar React y Node.js
- Creación de la estructura básica de directorios.
- Configuración de GIT.

SPRINT 1: Diseño de base de datos:

- Diseño inicial de la base de datos (entidades, relaciones...)
- Desarrollo de modelos básicos de datos y conexión a la base de datos.
- Configuración de la autenticación y sistema de sesiones.
- Estructura básica de la aplicación React.
- Configuración del enrutamiento.

SPRINT 2: USUARIO:

- Desarrollo de las funciones de inicio de sesión y registro
- ~~• Implementación de la recuperación de contraseña y verificación (descartado)~~
- ~~• Creación de la interfaz de perfil y opciones de edición (descartado)~~
- Validaciones en el registro y en el inicio de sesión

SPRINT 3: PUBLICACIÓN Y VISUALIZACIÓN DE ARTICULOS:

- Implementación de la funcionalidad de publicación de anuncios.
- Desarrollo de componentes de React para listar artículos de intercambio y segunda mano.
- Configuración de la base de datos y la API para gestionar los artículos.

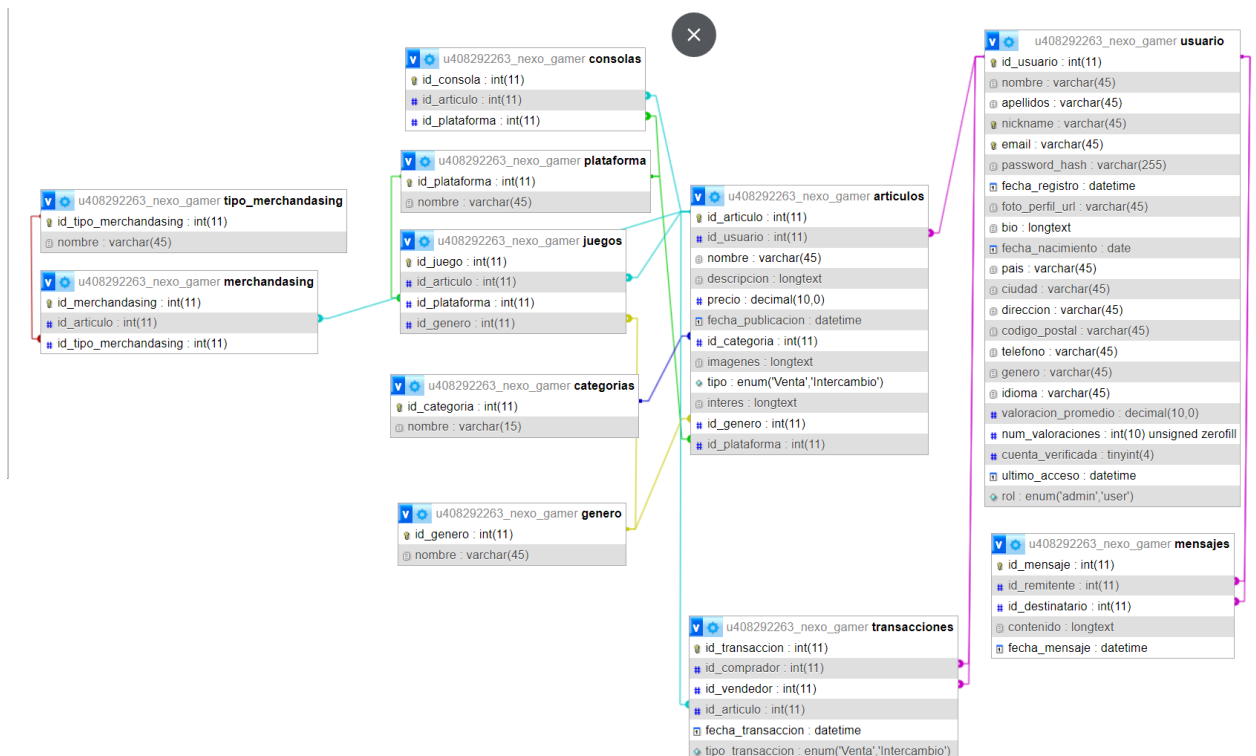
SPRINT 4 DETALLES DE ARTÍCULO:

- Implementación de vista detalles para cada artículo.
- Desarrollo de la lógica para manejar la interacción con los anuncios (ver detalles, editar, borrar).
- Validaciones

SPRINT 5: REVISIÓN Y CORRECCIÓN DE ERRORES:

- Mejora de la interfaz de usuario.
- Corrección de errores de diseño responsivo.
- Corrección de errores en ambos formularios.
- Validación en los formularios.

4.3 MODELO ENTIDAD/RELACIÓN

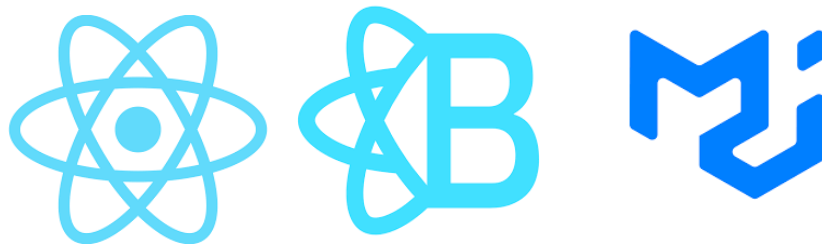


Este es el diseño actual de la base de datos, aunque no se utilizan todos los elementos, al ya estar creados los modelos y los controladores, decidí no modificarla por si algún día continuará con el proyecto.

5. HERRAMIENTAS EMPLEADAS

Para el desarrollo de NexoGamer, elegí el uso de herramientas y tecnologías que aprendí y utilicé en mi formación en centro de trabajo. La sensación que yo tuve es que garantiza una experiencia de usuario optima y un rendimiento y estructura robusta.

Frontend:



- React: Es un framework de JavaScript utilizado para crear una interfaz de usuario dinámica. Permite crear una experiencia de usuario fluida y reactiva.
- React-Bootstrap y Material-UI: Dos bibliotecas de componentes que facilitan el diseño de ciertas partes de la interfaz. Ayuda a acelerar el proceso de desarrollo.

Backend:



- Node.js: De nuevo un framework de JavaScript que tiene capacidad para manejar solicitudes múltiples simultáneamente.
- MySQL: Sistema de gestión de bases de datos relacionales.

Herramientas de desarrollo y diseño:



- Visual Studio Code: Editor de código que soporta múltiples lenguajes y extensiones que facilitan el trabajo.
- Dall-E 3: Inteligencia artificial para la creación de imágenes únicas y atractivas.

6. PRUEBAS REALIZADAS Y RESULTADOS

Realice las pruebas con Lighthouse y esto son los resultados obtenidos:



▲ First Contentful Paint

2,4 s

El primer renderizado con contenido indica el momento en el que se renderiza el primer texto o la primera imagen. [Más información sobre la métrica Primer renderizado con contenido](#)

▲ Renderizado del mayor elemento con contenido

3,8 s

El renderizado del mayor elemento con contenido indica el tiempo que se tarda en dibujar el texto o la imagen de mayor tamaño. [Más información sobre la métrica Renderizado del mayor elemento con contenido](#)

● Total Blocking Time

0 ms

Suma de los periodos, en milisegundos, entre FCP y Time to Interactive cuando la duración de la tarea excede los 50 ms. [Más información sobre la métrica Total Blocking Time](#)

● Cambios de diseño acumulados

0,062

Los cambios de diseño acumulados miden el movimiento de los elementos visibles dentro del viewport. [Más información sobre la métrica Cambios de diseño acumulados](#)

▲ Speed Index

2,4 s

Speed Index indica la rapidez con la que se puede ver el contenido de una página. [Más información sobre la métrica Speed Index](#)



Accesibilidad

Estas comprobaciones permiten identificar oportunidades para [mejorar la accesibilidad de tu aplicación web](#). La detección automática solo puede detectar un subconjunto de problemas y no garantiza la accesibilidad de tu aplicación web, por lo que también te recomendamos que hagas [pruebas manuales](#).

CONTRASTE

▲ Los colores de fondo y de primer plano no tienen una relación de contraste adecuada.



Prácticas recomendadas

CONFIANZA Y SEGURIDAD

○ Asegura que la CSP sea efectiva frente a ataques XSS



SEO

Estas comprobaciones aseguran que tu página siga las recomendaciones básicas sobre optimización en buscadores. Hay muchos otros factores adicionales que Lighthouse no valora aquí y que pueden afectar a tu posicionamiento en las búsquedas, como el rendimiento en las [Métricas web principales](#). [Consulta más información sobre las Directrices básicas de la Búsqueda de Google](#).

PRÁCTICAS RECOMENDADAS DE CONTENIDO

▲ El documento no tiene una metadescripción

7. PROBLEMAS Y SOLUCIONES

El primer problema al que me enfrenté fue que la visión que yo tenía de la plataforma, era muy compleja de plasmar con html, JavaScript y CSS, con php en el back, sobre todo por el poco tiempo que tenía para el proyecto. La solución que adopté fue utilizar las tecnologías que empecé a aprender en la formación en centro de trabajo, React y Node.js que permiten crear componentes y reutilizarlos lo que facilita mucho el trabajo y optimiza el tiempo.

El segundo problema, que me hizo perder bastante tiempo fue una función de edición del anuncio, que no me cargaba los datos previos y el formulario aparecía vacío. La solución fue ir con logs y paciencia ir buscando donde estaba el error o donde se estaba rompiendo el flujo de los datos.

El tercer problema es: subir la aplicación a la red. Como mi idea inicial era con otras tecnologías mi idea era subir el proyecto a un hosting normal, pero creo que el que yo probé, Hostinger, no es compatible con node.js. La solución temporal es presentar el proyecto en local. La segunda solución es buscar un servicio que sea compatible con node.js.

8. MODIFICACIONES Y AMPLIACIONES

Mi idea inicial incluía, muchas más funcionalidades de las que actualmente presenta el proyecto. De hecho, hay bastante código del Backend que no se está utilizando por la previsión que yo tenía.

En primer lugar: sistema de perfiles de usuario con valoraciones y comentarios. Me parece un punto fundamental para la creación de un entorno seguro y confiable.

Foro: me parece el lugar idóneo para las personas que sobre todo quieran compartir o buscar objetos puntuales, experiencias con otros usuarios...

Plataforma de pago y creación de un sistema de intercambio seguro. Esto último es un punto al que todavía no logré encontrar una idea que sea lo suficientemente buena.

Merchandising: incluir esa categoría en venta e intercambio ya que hay muchos artículos interesantes relacionados con el mundo de los videojuegos: Ropa, Funko pops, chapas, tazas...

“Universo NexoGamer”: sería el punto culmen de la aplicación: conseguir unir todas las piezas del mundo gaming en una misma plataforma, aunque centrada en el intercambio y venta, incluir eventos, noticias, guías y todo lo relacionado con este mundo.

9. MANUAL TÉCNICO Y DE INSTALACIÓN

Configuración del Entorno de Desarrollo

Para poner en marcha NexoGamer en tu entorno de desarrollo local, sigue estos pasos:

1. Clonar el Repositorio:

Abre la terminal y ejecuta el siguiente comando para clonar el repositorio:

git clone <https://github.com/Glominictus/NexoGamer>

2. Instalación de Dependencias:

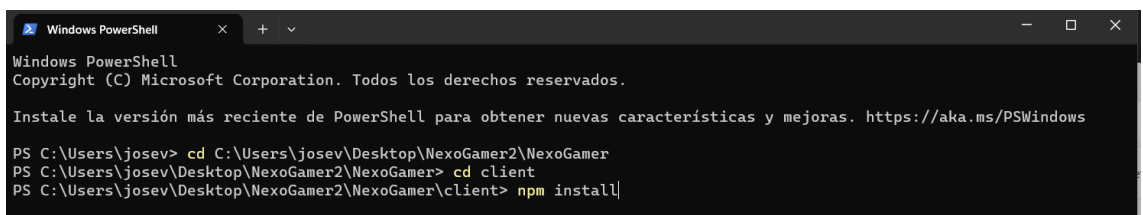
Navega al directorio del proyecto y ejecuta los siguientes comandos para instalar las dependencias necesarias tanto para el servidor como para el cliente.

cd nexo-gamer

#Instalar dependencias para el servidor

cd server

npm install



```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

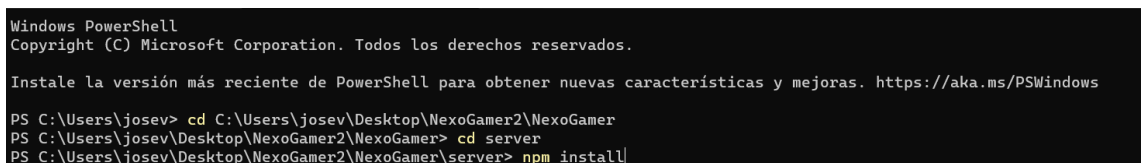
Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\Users\josev> cd C:\Users\josev\Desktop\NexoGamer2\NexoGamer
PS C:\Users\josev\Desktop\NexoGamer2\NexoGamer> cd client
PS C:\Users\josev\Desktop\NexoGamer2\NexoGamer\client> npm install
```

Instalar dependencias para el cliente

cd ../client

npm install



```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\Users\josev> cd C:\Users\josev\Desktop\NexoGamer2\NexoGamer
PS C:\Users\josev\Desktop\NexoGamer2\NexoGamer> cd server
PS C:\Users\josev\Desktop\NexoGamer2\NexoGamer\server> npm install
```


Ejecución del Proyecto:

Para iniciar tanto el servidor como el cliente en modo de desarrollo, ejecuta: `npm run dev`

Esto pondrá en funcionamiento NexoGamer en tu máquina local.

Uso de la Aplicación

Una vez que la aplicación esté en funcionamiento, puedes acceder a la interfaz web de NexoGamer a través de la dirección local que se indica en la consola del cliente.

```
VITE v4.5.0 ready in 1066 ms

→ Local:   http://localhost:5174/
→ Network: use --host to expose
→ press h to show help
```

Dependencias del Proyecto

El proyecto NexoGamer utiliza una variedad de dependencias. Las principales dependencias para el servidor y el cliente:

Servidor (server): `express`, `cors`, `dotenv`, `mysql2`, `sequelize`, `jsonwebtoken`, `bcrypt`, `axios`, entre otras.

Las dependencias de desarrollo incluyen `nodemon`.

Cliente (client): `react`, `react-dom`, `react-router-dom`, `@emotion/react`, `@emotion/styled`, `@mui/material`, `react-bootstrap`, `bootstrap`, `bootstrap-icons`, entre otras.


Las dependencias de desarrollo incluyen `vite`, `eslint`, y plugins relacionados con React y ESLint.

En caso de clonar el proyecto es necesario crear el archivo `.env` en la carpeta `server`, donde están los datos sensibles de la configuración del servidor.

```
DB_HOST=193.203.168.34
DB_USER=u408292263_admin
DB_PASS=Jose220988
DB_NAME=u408292263_nexo_gamer
IMGBB_API_KEY=949b24ffb477062f170dac82ba1186ca
JWT_SECRET=1a3e5c24&%
```


10. MANUAL DE USUARIO

El uso de la aplicación es muy fácil e intuitivo. En la pantalla principal nos encontramos la barra de navegación, el inicio de sesión y registro, y como elemento principal los últimos anuncios subidos, tanto artículos para venta de segunda mano como intercambio.


[Inicio](#)
[Intercambio](#)
[2ª Mano](#)
[Iniciar Sesión](#)
[Registrarse](#)

Últimos Artículos en Venta


Control de Wii



Plataforma: Wii
Precio: 15€

[Ver Detalles](#)


Test de Creación y ruteo



Plataforma: Retro
Precio: 12€

[Ver Detalles](#)

Xbox 360 Con Kinet




Plataforma: Xbox 360
Precio: 0€

[Ver Detalles](#)

Últimos Artículos para Intercambio


The Witcher 3



Plataforma: PlayStation 4
Interés: Elder Ring o Uncharted 4

[Ver Detalles](#)


PDP LVL40 NS Auriculares estéreo con cable



Plataforma: Nintendo Switch
Interés: Cualquier mando edición especial de Nintendo

[Ver Detalles](#)

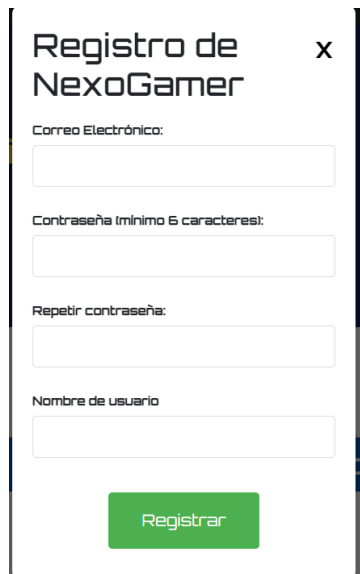
Xbox 360 Edición Los Simpsons



Plataforma: Xbox 360
Interés: Me interesaría la versión Pikachu de la Nintendo64, o alguna edición limitada de Nintendo que aun no tenga.

[Ver Detalles](#)

10.1 REGISTRO



Registro de NexoGamer x

Correo Electrónico:

Contraseña (mínimo 6 caracteres):

Repetir contraseña:

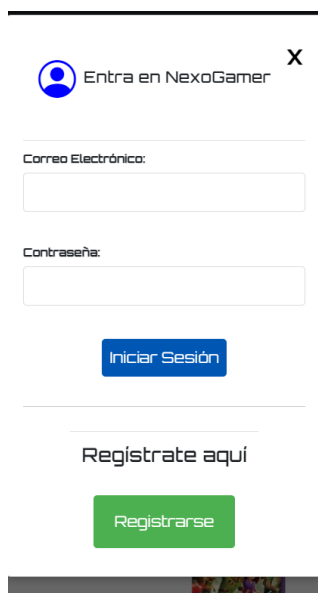
Nombre de usuario:

Registrar

El formulario de registro es muy sencillo, eliges el correo electrónico, una contraseña que repites, y un nombre de usuario. Todos estos campos del formulario tienen validaciones para comprobar que el usuario y/o el correo electrónico no están ya registrados en la base de datos, y que la contraseña es igual en ambos campos.

En caso de querer testear la aplicación se proporciona el siguiente usuario: profesor@test.com contraseña: profesor.

10.2 INICIO DE SESIÓN:



Entra en NexoGamer x

Correo Electrónico:

Contraseña:

Iniciar Sesión

Regístrate aquí

Registrarse

Con la cuenta proporcionada anteriormente se inicia la sesión y nos dará dos opciones, mis anuncios y cerrar sesión. Cerrar sesión, cierra la sesión, y mis anuncios lleva a los anuncios creados y a la opción de crear anuncio.

10.3 MIS ANUNCIOS

Mis anuncios

Crear anuncio

Control de Wii	The Witcher 3	PDP LVL40 NS Auriculares estéreo con cable
		
Plataforma: Wii Precio: 15€	Plataforma: PlayStation 4 Interés: Elder Ring o Uncharted 4	Plataforma: Nintendo Switch Interés: Cualquier mando edición especial de Nintendo
Ver Detalles Editar Borrar	Ver Detalles Editar Borrar	Ver Detalles Editar Borrar

En esta pantalla tenemos la opción de crear anuncio, ver los detalles del anuncio, editar y borrar el anuncio.

10.3.1 CREAR ANUNCIO

Nuevo Anuncio

Tipo de anuncio:

☐ venta ☐ intercambio

Categoría:

Elige la categoría

Plataforma:

Elige la plataforma

Nombre

Descripción:

Imágenes:

[Elegir archivos](#) Ninguno archivo selec.

Enviar Anuncio

El formulario es dinámico, según la opción elegida generará unas opciones u otras. En el caso de elegir venta, aparecerá un campo para poner precio al artículo, y en el caso de elegir intercambio aparecerá un campo para escribir los intereses que tienes. Lo mismo con la categoría, si eliges juego aparecerá un nuevo select para elegir el género del juego. El campo de imagen si no eliges una imagen se insertará una imagen por defecto.

10.3.2 EDITAR ANUNCIO

Es similar al formulario de crear anuncio, solo que aparecerán los datos precargados

Editar Anuncio

Tipo de anuncio:

☒ venta
 ☐ intercambio

Categoría:

Accesorio

▼

Plataforma:

Wii

▼

Nombre:

Control de Wii

Descripción:

blabla

Precio:

15

Imágenes:

Elegir archivos

Ninguno archivo selec.

X

11. CONCLUSIONES

El desarrollo de NexoGamer ha supuesto un gran aprendizaje y un nuevo desafío al utilizar tecnologías que comencé a estudiar hace poco tiempo. La creación de esta plataforma intenta establecer un nuevo nexo de unión entre jugadores, ya sea a través del intercambio o de la venta de artículos de segunda mano.

Aunque por el tiempo límite del proyecto no me ha permitido alcanzar todos los objetivos que a mi me hubiera gustado, si me permitió crear una plataforma accesible y dinámica, con una curva de aprendizaje muy baja, que permite a cualquier usuario utilizar la plataforma. El dinamismo de la página se obtiene gracias al uso de tecnologías como React y Node.js, Frameworks de JavaScript que permite desarrollar una aplicación robusta.

El desarrollo de la aplicación fue dividido en varios Sprints que me permitieran alcanzar los objetivos deseados. A pesar de algunos pequeños baches que aparecieron en el camino, la mayoría tuvo una solución fácil, y otros no se pudieron solucionar por lo comentado anteriormente, el límite de tiempo.

NexoGamer es una propuesta innovadora para el disfrute de la experiencia tanto de jugadores clásicos como más modernos.

Esto es simplemente la base de lo que puede ser una gran aplicación que con futuros desarrollos podría ser una de las aplicaciones más potentes del mundo de los videojuegos.

Con la finalización de este proyecto, me quedo en primer lugar, que he creado una aplicación funcional, y en segundo lugar, y más importante, los conocimientos que he obtenido de dos lenguajes punteros.