



C.P.R. Liceo “La Paz”

Proyecto Fin de Ciclo

Desarrollo de Aplicaciones Multiplataforma

Autor: Fernando Gonzalez Suarez-Noguerol
Tutor: Jesus Ángel Perez-Roca Fernandez

Proyecto Desarrollo de Aplicaciones Web



Resumen

El Proyecto **TCG Market** tiene como objetivo crear una página de compra y venta de productos relacionados con los Juegos de Cartas Coleccionables a.k.a TCG (Trading Card Game). La página web habilita que los usuarios puedan comprar los productos que los vendedores ponen a la venta mediante un sistema de membresía donde pueden ser tanto compradores como vendedores. Ser un comprador solo requerirá de una cuenta en la página web que es totalmente gratuita y usar los métodos de pago aceptados por la página mientras que los vendedores se tendrán que verificar como propietarios de la cuenta haciendo un pequeño trámite bancario. Se generará beneficio mediante el cobro de un pequeño impuesto por cada venta de un 5% y mediante el patrocinio de diferentes marcas.

Abstract

TCG Market is a project whose objective is to make a web page to sell and buy TCG (Trading Card Game) products. The web page allows you to buy products put on sale by sellers which participate in a membership system where they can be buyers and/or sellers. Being a buyer only requires having a free account available to everyone. To be a seller, the user needs to verify themselves as the owner of both web account and bank account through a simple bank procedure. To generate benefit, a tax will be applied for each sale done in the platform and by means of brands sponsoring the web.

Palabras Clave

- **Juego de Cartas Coleccionables:** Mi hobby y el principal foco de este proyecto. Similar a los clásicos cromos pero tienen su propio juego con su propias reglas. La gran variedad permite que tengas diferentes tipos de experiencias y sus amplias comunidades te permiten ampliar tus horizontes
- **Mercado Libre:** Dentro de la web solo se tramitará el impuesto de venta sin coste adicional, es decir, se formará un ecosistema económico fluido, sostenido y usado solo por y para la comunidad.
- **Bases de datos relacional:** Se usará una base de datos relacional para manejar y procesar los datos de la web. Mientras que es un sistema muy rígido, considero que es el más consistente y sólido para este proyecto.
- **SQL:** SQL es el lenguaje de bases de datos por excelencia y es el que mejor soporte me va a dar para desarrollar e implementar mi base de datos en mi proyecto.
- **Java:** Java es el programa de lenguaje que uso para manejar los datos gracias al Framework **Spring** que me permite comunicar la web con la base de datos, tratar los datos de forma efectiva y establecer parámetros de autenticación y autorización.
- **React JavaScript:** Reactjs es uno de los motores más innovadores para diseñar páginas web y su lógica. Es un motor MUY potente, permitiendo crear páginas webs muy potentes.
- **JWT:** Json Web Token es la forma de comunicación para autenticar y autorizar usuarios de la forma más segura a través de un JSON codificado que te decodifica y permite que se usen los datos privadamente. Se usará en el
- **React-Bootstrap:** Una librería de CSS y de React para refinar el código HTML y usar la lógica de componentes de React permitiendo una implementación más profesional y limpia del código.

Dedicatoria o agradecimientos.

Dedicado a mi hermana Carlota, la persona más resiliente que conozco

Dedicado a mi madre, la persona más fuerte que conozco

Dedicado a mi tío Enrique, la persona más resistente que conozco

Dedicado a mi amiga Nu, porque sin ella, no habría dado ni un paso

Dedicado a mi amigo Martín, que nos apoyamos mutuamente contra viento y marea

Dedicado a mi abuela Amalia, que es el pilar de mi familia

Agradecimientos a Mónica, mi profesora, porque me enseñó que los pequeños detalles importan por muy pequeños que sean

Agradecimientos a Rico, mi profesor porque me ayudó a entender que el trabajo del día a día tiene los mejores resultados

Agradecimientos a mis amigos Pedro y Pablo que me han acompañado durante este curso

Dedicado a mi mismo, Fernando, porque, a pesar de todo, he llegado hasta aquí

Sumario

<u>Resumen</u>	3
<u>Abstract</u>	4
<u>Palabras Clave</u>	5
<u>Introducción/motivación.</u>	11
<u>Objetivos.</u>	12
<u>Estado del arte.</u>	13
<u>Caso de estudio.</u>	14
<u>Diagramas.</u>	15
<u>Desarrollo del proyecto</u>	16
<u>Manual Administrador</u>	17
<u>Manual Usuario</u>	18
<u>Viabilidad tecno-económica.</u>	19
<u>Trabajo futuro.</u>	20
<u>Conclusiones.</u>	21
<u>Biblioteca de recursos web y referencias.</u>	22
<u>Anexos.</u>	23

Motivación y Objetivos.

Mi motivación para hacer este proyecto se basa en que los juegos de cartas son mi hobby por excelencia. Tras años jugando y viendo el mercado de estos productos, vi que las alternativas dejan mucho que desear. Por lo tanto, quiero establecer un nuevo estándar de comercio en este sector para que la mayor cantidad de gente pueda disfrutar de mi hobby.

Mis objetivos son dos:

1. **Mejorar las transacciones de cartas:** Hay muchísimos factores que afectan al precio de una carta: Demanda, Rareza, Impresión, Exclusividad y un largo etc. Quiero tratar con estos factores de una manera justa y eficiente. Por ejemplo: Hay gente que abusa de la rareza y hace compra masiva de cartas para después venderlas a costes más altos.
2. **Ofrecer un mercado estable:** Probablemente la parte más difícil del proyecto, es presentar una plataforma preparada para mantener un mercado saludable, estable y disponible para todos.

Son las dos metas que forman el pilar de mi motivación y a las que este proyecto será fiel.

Estado del arte.

Card Trader

La principal razón de empezar este proyecto es esta página web. Personalmente siempre la he usado y puedo decir que mientras tiene muchas fortalezas, también tiene múltiples debilidades graves.

Fortalezas

1. Gran variedad de juegos

Tiene una gran variedad de juegos para catar a un gran público objetivo

2. Rapidez de lanzamiento de juegos

Cuando se lanza un juego, son los más rápidos en sacarlos a mercado

3. Ofrecen servicios premium tales como reunir varios pedidos de distintas partes de Europa y enviarlos directamente y un optimizador de carrito de compra para ver las opciones más baratas.

Debilidades

1. Manejo pobre del listado de producto

Se lista el producto de forma que en juegos donde se repiten nombres, se pisan registros y si quieras añadir una carta con varios nombres, se equivocará la mayoría de las veces en el listado de la carta

2. Catalogo de envío desactualizado y no adaptado según el país

La categorización de envíos está muy maltratada y casi no ofrece opción a envío ordinario en casi ningún país y solo envíos certificados con precios desorbitados

3. Los servicios premium fallan la mayoría de veces.

El envío combinado de reunir los paquetes se cobra aparte por parte de Correos y el optimizador de carrito optimiza teniendo en cuenta intereses exteriores.

4. Los pagos son exactos y no permiten un manejo de crédito. Los reembolsos en cualquier caso no son extraíbles y no permite transferencias siendo bastante menos flexible en las formas de pago.

Tiendas comerciales ([OnlyCards](#), [Alita Comics](#))

Estas tiendas son la representación del juego clásico. Entre rol y de juegos de nicho, los TCGs se abrieron paso a tiendas especializadas para formar comunidad y poder disfrutar del hobby en compañía

Fortalezas

1. Las tiendas comerciales tienen la ventaja de ser una tienda física y poder formar comunidad, algo fundamental en el TCG. Así atendiendo al público más antiguo y pudiendo atraer más público a través de eventos públicos con más cercanía hacia la clientela.
2. Pueden pedir en demanda y atender a sus clientes con un trato personalizado creando un público más fiel.
3. Las tiendas reciben soporte directo de la sección organizativa de los TCGs y con ese soporte, se permiten obtener material promo exclusivo y también pudiendo organizar torneos oficiales con material exclusivo

Debilidades

1. El mantenimiento de la comunidad es exigente y mientras que se especializan en juegos de nicho, no pueden dedicar sus recursos al completo en ofrecer un catálogo de productos especializado en los TCG
2. El mercado secundario que conforman mi web y los otros mercados digitales es competidor directo de las tiendas comerciales y en la mayoría de casos, superior por la flexibilidad y la amplia gama de productos obligando a algunas tiendas a venir a nuestro terreno. (En este caso, Onlycards tiene cuentas en múltiples mercados digitales)

El estado del arte de los TCG es, a día de hoy, amplio pero sigue siendo un arte o hobby de nicho. Ha resurgido gracias al reciente boom de coleccionismo que afectó en las redes sociales y es un modelo de negocio que, al compartir características con las “lootboxes” el modelo de negocio por excelencia en videojuegos, se está empezando a explorar profundamente por su gran rentabilidad a corto plazo. Con este proyecto, espero darle visibilidad a los TCG como producto viable.

Caso de estudio.

Para mejorar la transacción del producto de TCG, debemos tener varios casos presentes:

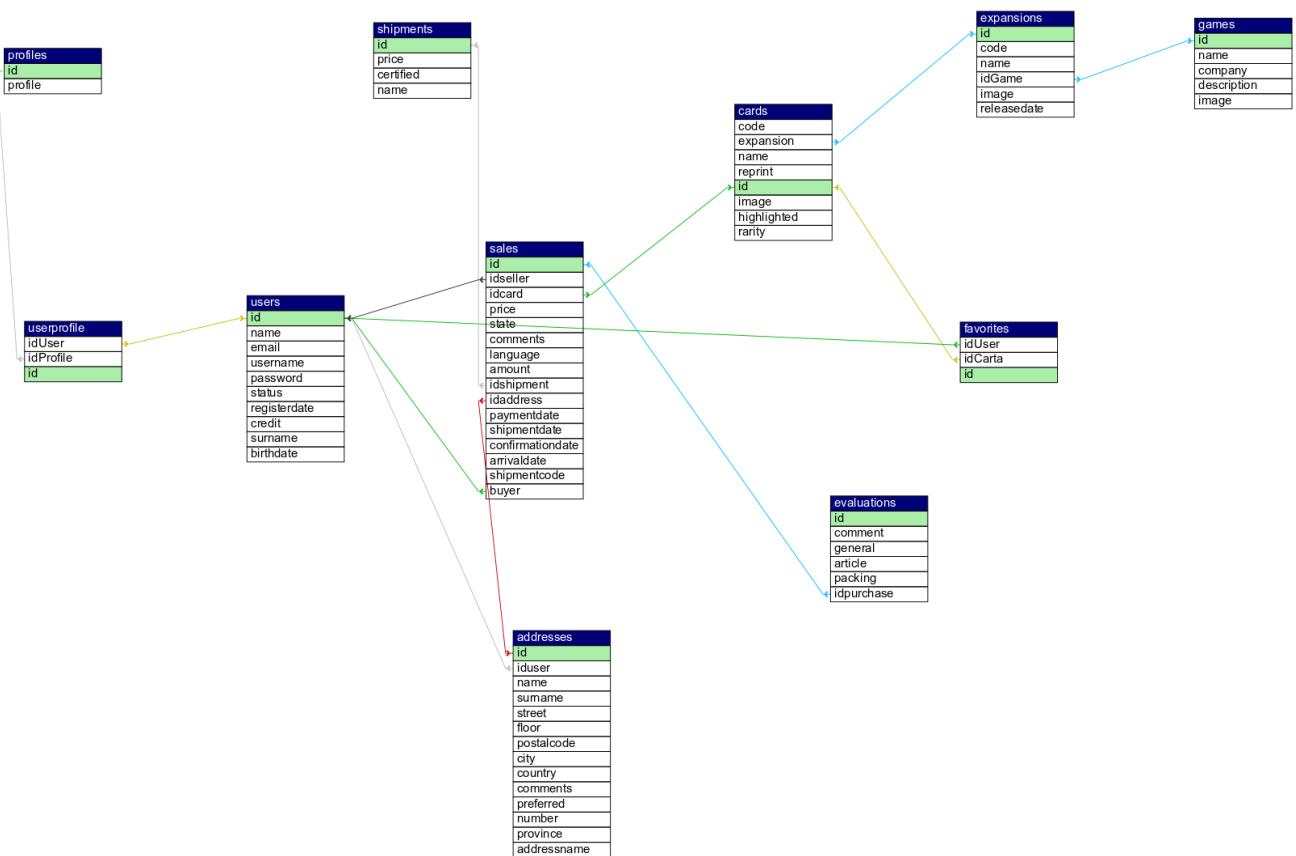
Tipo de producto: El TCG implica que su contenido sea colecciónable implica que va a haber múltiples colecciones de un mismo juego. Las colecciones se presentan en cajas que a su vez contienen sobres y en los sobres habrá un número determinado de cartas. Se debe ofrecer estos 3 productos de forma sencilla y visible: **Cajas, Sobres y Cartas.** Estas son las unidades básicas de TCG.

Mercado: Si se quiere ofrecer un espacio de compra y venta realmente efectivo y fluido, se debe regular la demanda y la oferta de los productos en base a las normas de juego del TCG. Es decir, los TCG establecen unos límites numéricos a la hora de jugar una carta en concreto ej. Solo puedes jugar con 4 cartas con el mismo nombre/código. Cuando una carta en concreto destaca y se codicia entre los jugadores, se suele aprovechar para hacer compra masiva y revender a precios desorbitados. Esta acción se conoce como **Buyout** y es un acontecimiento que se debería evitar a toda costa debido a que daña el mercado del juego y si se complica, podría hasta afectar a la propia comunidad del juego. Por lo tanto, esta situación se debe regular y evitar.

Desarrollo del proyecto.

Base de datos

La lógica de negocio se establece con la manera en la que se va a trabajar la información. Para ello, se empezará estableciendo una base de datos coherente:



Las entidades con las que se trabajará son:

1. Las ventas son el foco central del modelo de negocio y será el pilar del desarrollo de todo el proyecto. Las ventas manejan todo el flujo tanto de capital como de producto.

Los datos de la venta se plasmarán en los siguientes atributos:

1. La carta indica la carta que se vende así marcando la relación entre Carta y Venta.
2. La cantidad indica la cantidad del producto que se vende
3. El lenguaje se indica porque no todos los juegos tramitan o regulan sus juegos en todos los idiomas.
4. El precio.

5. El estado de la carta se indica con una selección de unos atributos en específicos:
 - a. NEAR MINT: Este estado indica que básicamente que el producto está cerca de la perfección
 - b. EXCELLENT: Este estado indica que el producto esta en excelente estado y se diferencia de NEAR MINT por tener ligerísimos defectos sobre el producto
 - c. GOOD: Este estado indica que el producto está bien aunque se pueden notar leves deterioros.
 - d. BAD: Este estado indica que el producto está dañado.
6. Comentario sirve para describir detalles extra del producto que no se contemplen en esta lógica de negocio tales como detalles concretos del estado, si viene un extra con el proyecto

Las ventas necesitan atributos para poder marcar la diferencia entre compra y venta. Una compra que haya sido completada tendrá:

1. Un Comprador que representa la primera relación entre Usuario y Ventas por el cual podremos sacar toda la información del comprador
2. Una fecha de confirmación de compra. No podemos usar la confirmación de pago como concepto para filtrar las compras debido a que la propuesta de negocio es crear un monedero digital donde se pueda gestionar la moneda de forma eficaz y no se efectuará el pago de inmediato a través de una pasarela de pago. Por ende, la confirmación de la compra se usará como anclaje para ver si se ha tramitado de venta a compra

Para manejar el flujo del producto - destinatario, se usarán unos atributos que para hacer el seguimiento de forma efectiva:

1. El pago del producto: Para saber cuando se efectuó el pago del producto.
2. La dirección a enviar del comprador: Para que el vendedor sepa dónde tiene que enviar el producto.
3. El envío que el comprador ha seleccionado para que se envíe el producto.
4. Una fecha de confirmación de envío. A este atributo se le adjudicará la fecha de envío.
5. Un código de seguimiento. Los envíos en el sistema postal pueden ser tanto ordinarios como certificados. En el último caso, a los envíos se les adjudica un código para seguirlos lo cual se debe de implementar sin falta al proyecto.
6. Una fecha de confirmación de llegada. Marca el fin del flujo y permite que una compra sea tramitada como completada y así dar paso a la evaluación de la compra

2. Cartas:

Las cartas son el producto estrella de este proyecto y se presentan de forma sencilla dado que entrar en detalle no entra dentro de las competencias. Se quiere conseguir presentar el producto de forma limpia.

Se representan con los atributos siguientes:

1. Código es el código de la propia carta que es único y se usará para diferenciar reimpresiones del mismo producto.
2. Nombre.
3. Expansión a la que pertenece la carta,
4. Imagen.
5. Destacada para mostrar el producto destacado
6. Rareza es la categoría de clasificación que se usan en la gran mayoría de TCGs aunque es un trabajo en progreso y se implementará en trabajo futuro
7. Reprint es un valor booleano que indica si esta carta es una reimpresión, no que haya sido reimpresa.

3. Expansiones:

Las expansiones son el conjunto de cartas que los TCG usan para manejar su calendario de lanzamientos de producto.

Se representan en los atributos siguientes:

1. Código es el atributo único de la expansión y el que diferencia a una del resto. Este código se implementa de cierta forma en la entidad Carta para crear el código de la mencionada
2. Juego al que pertenece la expansión
3. Nombre
4. Imagen
5. Fecha de lanzamiento

4. Juegos:

Los juegos son la entidad más en la jerarquía y contiene todas las expansiones.

Esta entidad será expandida más en trabajo a futuro dado que las implicaciones a diseñar se van fuera del tiempo disponible. Por ahora es una categorización simple:

Los atributos de Juegos son:

1. Nombre
2. Compañía
3. Descripción del juego
4. Imagen

5. Direcciones:

Las direcciones son el destino donde se deberá enviar el producto designado y el usuario podrá crear cuantas quiera para asignar diferentes compras a diferentes direcciones.

Los atributos son:

1. Usuario al que pertenece la dirección
2. Nombre del destinatario
3. Apellido del destinatario
4. Nombre de la dirección
5. Calle de la dirección
6. Piso de la dirección
7. Código Postal
8. Ciudad
9. Provincia
10. País
11. Comentarios
12. Preferida. Es un trabajo en curso pero es la dirección que se asignará de forma determinada cada vez que el usuario haga un pedido.

6. Envíos:

Los envíos son simplemente una categorización para mostrar los tipos de envíos posibles. Se ampliarán para incluir diferentes pedidos para los distintos países en relación a las direcciones para que sean dinámicos.

Los atributos de Envíos son:

1. Precio
2. Nombre
3. Certificado. Es un booleano para verificar que el envío es certificado y exigir al vendedor que inserte un código de envío.

7. Usuarios:

Los usuarios son las personas que acceden y usan nuestro producto.

Los atributos de Usuarios son:

1. Nombre del usuario
2. Apellido del usuario
3. Nombre de usuario que identificará al usuario al ser un atributo único.
4. Correo del usuario que también identificará al usuario al ser un atributo único dado que el correo es privado y no se comparte.
5. Contraseña de acceso

6. Fecha de registro
7. Fecha de nacimiento
8. Crédito: Esto sería una entidad con sus propios atributos pero se estimó que no daría tiempo a implementarla de forma correcta debido a que toda la lógica de transacciones y transferencias de crédito estarían ligadas a APIs que requerirían datos sólidos para hacer pruebas.

8. Perfiles:

Los perfiles son atributos que se aplican a los Usuarios para determinar la cantidad de acceso que tienen a las distintas capas del proyecto

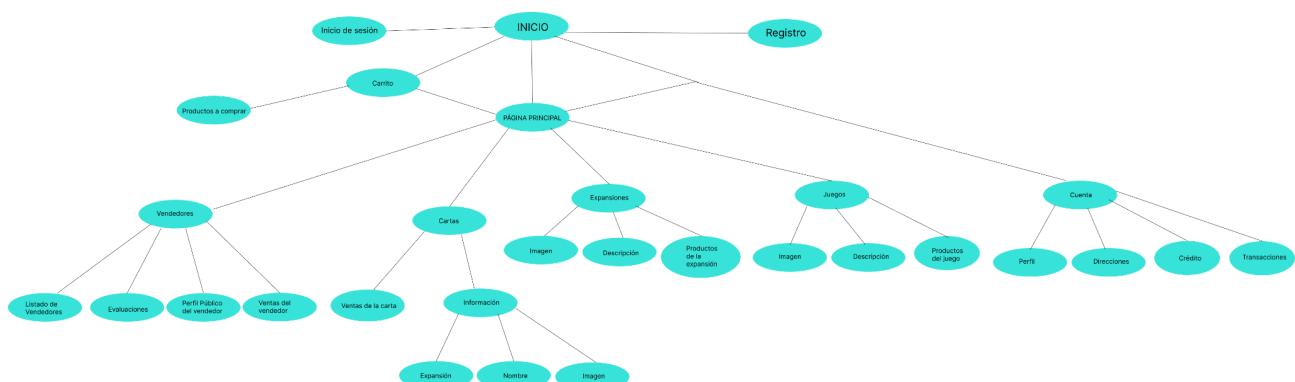
Los atributos son:

1. Perfil.
2. El usuario que se le asigna que pueden ser múltiples en ambos sentidos (Varios usuarios pueden tener varios perfiles) E.j: El usuario Mónica tiene los perfiles USUARIO y VENDEDOR.

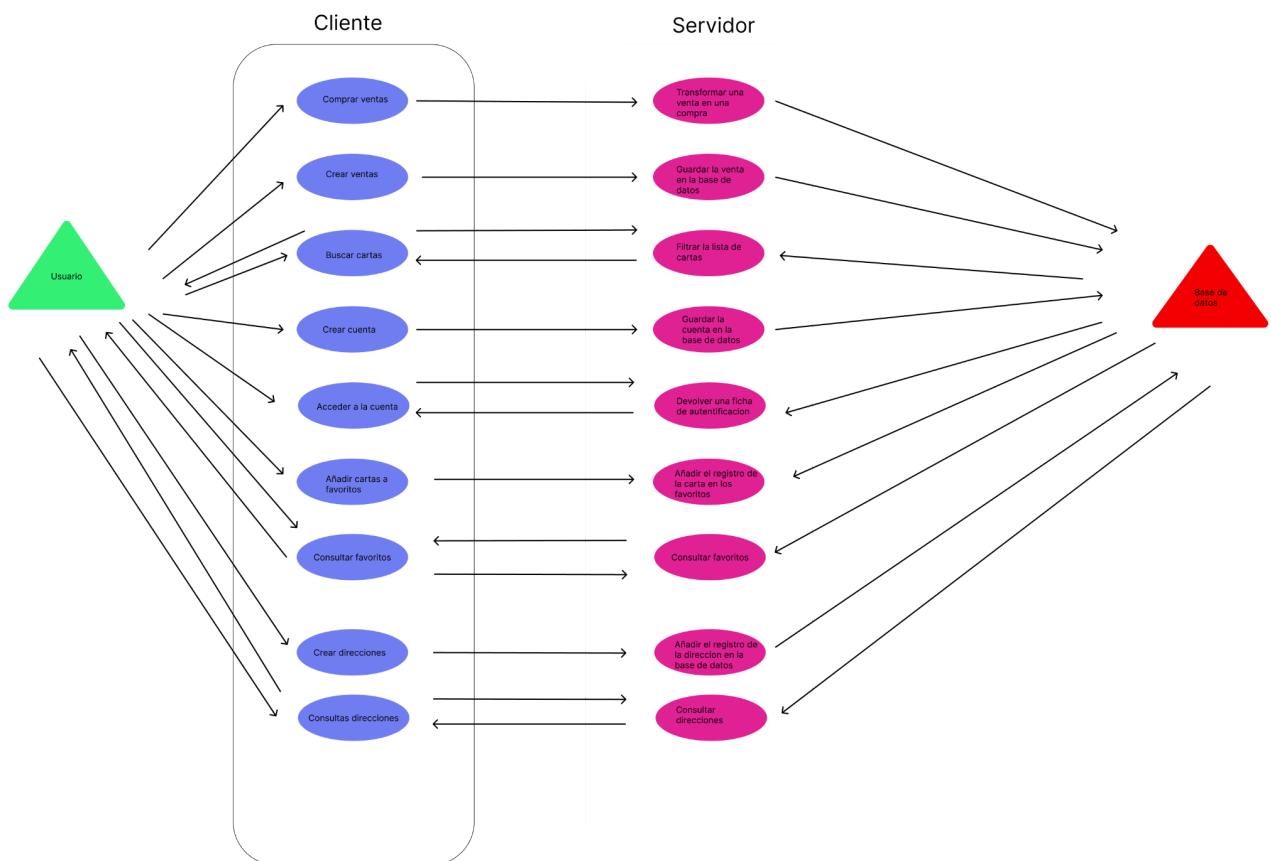
Diseño Web

Una vez diseñada la lógica de la manipulación de los datos, debemos diseñar la página web para que sea atractiva e intuitiva para el usuario. El diseño web que se enseña es el resultado final y se desarrollará una demo que tratará de acercarse de una manera razonable a este por temas temporales y que realmente se quiere mostrar las funcionalidades

Para empezar, se crea la estructura del proyecto en base al diagrama de navegación, al diagrama de casos de uso y a la guía de estilos:



Proyecto Desarrollo de Aplicaciones Web



INICIO

TCGMarket

Nombre de usuario: Contraseña: INICIAR SESIÓN REGISTRARSE 

Juegos Buscar

Bienvenido a TCGMarket, el mejor mercado de cartas de europa

Cardfight Vanguard Digimon Yu-Gi-Oh

TCGMarket es el mercado europeo que te permite comprar y vender cartas de manera fácil

Vende cartas a dos clicks de distancia	Un sistema de protección de pedidos optimizado	Maneja el dinero de forma fácil
La mejor comisión en toda europa	Gran demanda entre los jugadores	Fácil de usar

400.000+ ventas

150.000+ usuarios

1.000.000+ Cartas vendidas

30 Países

Listo para empezar a usar TCGMarket?!

Elige tu juego!

Para el inicio, se ha optado por una introducción básica a la web y su funcionamiento. El nuevo usuario podrá localizar el botón de registro rápidamente o localizar el producto que quiere gracias a la barra de búsqueda.

Menú principal

TCGMarket

Nombre de usuario: Contraseña: INICIAR SESIÓN REGISTRARSE 

Juegos Buscar

[CAROUSEL]

Best Sellers Tendencias Las Mejores Gangas

[Última Expansión] Novedades

[Expansion más reciente] [Segunda expansión más reciente] [Tercera expansión más reciente]

← [Código Última Exp] [Código Última Exp] [Código Última Exp] [Código Última Exp] → Verlo Todo

Listo para empezar a usar TCGMarket?!

Regístrate



Proyecto Desarrollo de Aplicaciones Web

Para el menú principal, se establece una relación visual con el usuario y el producto. Se mostrarán los productos más recientes y los productos más destacados sean por el precio o por su demanda.

Registro

TCGMARKET

Cuenta particular

Cuenta profesional

Nombre *

Nombre

Apellidos

Fecha de nacimiento *

dd/mm/aaaa

Dirección *

Calle *

Calle

Numero de casa / Piso *

Numero de casa / Piso

Código Postal *

Código Postal

Ciudad *

Ciudad

País *

País

Dirección Email *

Correo Electrónico (eg. magnamon@tcgmarket.net)

Elija un nombre de usuario y una contraseña *

Nombre de usuario

Contraseña

Repita su contraseña

REGISTRARSE

Registro

TCGMARKET

Cuenta particular

Cuenta profesional

Nombre *

Nombre

Apellidos

Fecha de nacimiento *

dd/mm/aaaa

Dirección *

Calle *

Calle

Numero de casa / Piso *

Numero de casa / Piso

Código Postal *

Código Postal

Ciudad *

Ciudad

País *

País

Dirección Email *

Correo Electrónico (eg. magnamon@tcgmarket.net)

Número *

Número

Scan del DNI/NIF *

Subir archivo(s)

Scan del registro de empresa *

Subir archivo(s)

Elija un nombre de usuario y una contraseña *

Nombre de usuario

Contraseña

Repita su contraseña

REGISTRARSE



Proyecto Desarrollo de Aplicaciones Web

El registro se centra completamente para que el usuario se registre sin ningún tipo de distracción. El registro se divide en profesional y en particular para preparar casuísticas en el proyecto sobre Vendedores y Vendedores a gran escala. Se incluirá en el trabajo a futuro.

Información Producto

TCGMarket



[Usuario]



Vender



Comprar

Carrito
([Precio total])

≡ Juegos

Buscar

Inicio / [Juego] / [Tipo de Producto] / [Expansion] / [Producto]

[Producto] [Expansion]

	Información	Observaciones
[IMAGEN]	<p>Lore ipsum Lore ipsum</p>	

Vendedor	Información del Producto	Oferta
Lore ipsum	Lore ipsum	Lore ipsum
Lore ipsum	Lore ipsum	Lore ipsum
Lore ipsum	Lore ipsum	Lore ipsum

TLF: 987654321 Avd. Carton
16742 FábricaTCGMarket Marca
Registrada

La información del producto se orienta en dos apartamentos: Información y despliegue de ventas del producto. Las ventas tienen diferentes filtros para múltiples criterios y varias filas indicando todos los atributos de cada venta.

Lista de producto

TCGMarket

[Usuario]

Vender

Comprar

Carrito
([Precio total])



Juegos

Buscar

Home / [Juego] / Lista

LISTA DE [PRODUCTO] DE [EXPANSIÓN]

Edición

Nombre



Nombre Exacto

Solo Disponible

Opciones avanzadas de filtro

Ordenar por:

BUSCAR

99 Resultados

Nombre

Número

Rareza

Disponible Desde

< Página 2 de 8 >

TLF: 987654321

Avd. Carton
16742 Fábrica

TCGMarket Marca
Registrada

La lista de productos es sencilla pero eficaz. Usa paginación y filtros para filtrar la lista.

Menú Expansion



The screenshot shows the TCGMarket website interface. At the top, there is a navigation bar with the brand name "TCGMarket" on the left, followed by icons for "Usuario" (User), "Vender" (Sell), "Comprar" (Buy), and "Carrito" (Cart) with a sub-label "(Precio total)". On the far right of the navigation bar is a downward-pointing arrow icon. Below the navigation bar, there is a search bar with the placeholder text "Buscar". To the left of the search bar is a button labeled "Juegos". The main content area has a dark gray background and features the text "[Expansión]" centered in white.



Información del Set

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer ultricies, nibh in lobortis blandit, justo sapien accumsan arcu, vel tristique massa odio lacinia neque. Donec dapibus mauris sit amet tempor rhoncus. Pellentesque posuere pulvinar odio. Suspendisse sed libero ac massa vulputate dictum. Proin tempus bibendum dui, vel pharetra odio ultricies rhoncus. Nulla dolor ante, lacinia fringilla mi et, congue iaculis diam. Suspendisse molestie, purus ut pretium facilisis, enim lectus pulvinar turpis, vestibulum sagittis tellus quam sit amet nulla.

Proin sed elit mattis, tristique nibh sit amet, fermentum tellus. Suspendisse ultrices nisl nec nibh bibendum mattis. Nullam sit amet tortor varius, congue erat vitae, porta metus. Integer tempor sollicitudin tincidunt. In nec ante in metus fermentum porta. Phasellus ac placerat nibh. Nulla feugiat nisl sagittis suscipit dictum. Vestibulum et aliquet elit, nec blandit nulla. Nunc turpis leo, porta ac eleifend a, ultrices id nibh

Tablas de datos

TCGMarket



[Usuario]

Vender

Comprar

Carrito

{[Precio total]}



Juegos

Buscar

Inicio / Tabla de Datos

Tablas de datos

[Conjunto de datos concreto]

[Conjunto de datos concreto]

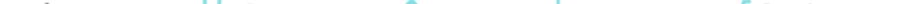
[Conjunto de datos concreto]

TLF: 987654321

Avd. Carton
16742 Fábrica

TCGMarket Marca
Registrada

Resultado de Tablas de datos

 TCGMarket  [Usuario]  Vender  Comprar  Carrito
([Precio total])

 Juegos 

[Inicio](#) / [Tabla de Datos](#) / [Conjunto de datos concretos]

Mejores gangas.

Carta	Precio Actual
Loreum Ipsum	Loreum Ipsum 
Loreum Ipsum	Loreum Ipsum 
Loreum Ipsum	Loreum Ipsum 

TLF: 987654321

Avd. Carton
16742 Fábrica

TCGMarket Marca Registrada

Las tablas de resultado son formas de definir las consultas complejas de la aplicación y delimitar los resultados de forma efectiva. Por ello mostramos un grupo de lista con las consultas y dentro una consulta ya filtrada.

Lista de vendedores

TLF: 987654321

Avd. Carton
16742 Fábrica

TCGMarket Marca Registrada

Información de Vendedor

TCGMarket



[Usuario]



Vender



Comprar



Carrito

([Precio total])



≡ Juegos

Buscar

[Vendedor]

[Tipo de vendedor] [Cuenta de ventas]

Miembro desde [Año de registro]

[Nombre de Establecimiento]

[Calle]

[Nombre del usuario]

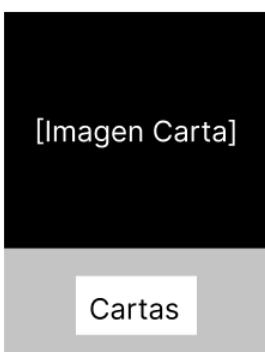
[Edificio]

[Tiempo estimado de llegada
de pedidos]

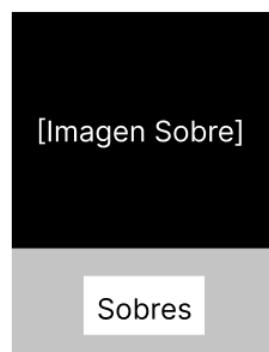
[País]

Inicio / Busqueda de Usuarios / [Nombre de Usuario]

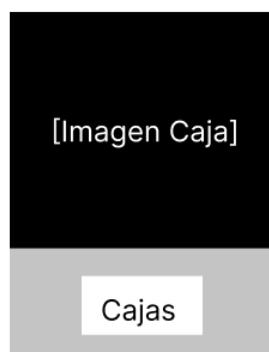
Articulos de [Juego]



Cartas



Sobres



Cajas

Ultimas valoraciones de [Vendedor]

Comprador	Fecha	Comentarios	Eval
Lorem Ipsum	Lorem Ipsum	Lorem Ipsum	✓
Lorem Ipsum	Lorem Ipsum	Lorem Ipsum	✓
Lorem Ipsum	Lorem Ipsum	Lorem Ipsum	✓

Proyecto Desarrollo de Aplicaciones Web

Los vendedores son representados por una tarjeta de perfil diciendo todos sus datos públicos pasando por unos grids mostrando sus productos y al final mostrando sus evaluaciones de sus ventas.



Cuenta del usuario

TCGMarket

Nombre de
usuario:

Contraseña:

INICIAR SESIÓN

REGISTRARSE



Juegos

Buscar

Inicio / Cuenta

Cuenta

Cuenta

[Nombre de Usuario]

[Ver Perfil Público](#)

Perfil

Nombre

[Nombre del Usuario]

Direcciones

Tipo de Cuenta

[Tipo de cuenta]

Crédito

Fecha de nacimiento

[Fecha de nacimiento]

Transacciones

[Estado del usuario]

Direccion

[Nombre de dirección]

[Calle]

[Código Postal] [Ciudad]

[País]

Balance de Cuenta

[Balance de cuenta]

TLF: 987654321

Avd. Carton
16742 Fábrica

TCGMarket Marca
Registrada

Cuenta del usuario

TCGMarket

Nombre de
usuario:

Contraseña:

INICIAR SESIÓN

REGISTRARSE



≡ Juegos

Buscar

Inicio / Cuenta / Direcciones

Direcciones

Cuenta

Perfil

Direcciones

Crédito

Transacciones

[Dirección]

[Nombre del destinatario]
[Calle]
[Código Postal] [Ciudad]
[País]

+ Nueva dirección

TLF: 987654321

Avd. Carton
16742 Fábrica

TCGMarket Marca
Registrada

Cuenta del usuario

TCGMarket

Nombre de usuario:

Contraseña:

INICIAR SESIÓN

REGISTRARSE



≡ Juegos

Buscar

Inicio / Cuenta / Direcciones

Crédito

Cuenta

Perfil

Direcciones

Crédito

Transacciones

Crédito Actual

2,00€

Añadir Crédito

Sacar Crédito

Cuenta Bancaria

Cuenta bancaria asociada a tu cuenta

Propietario de la cuenta [Propietario de la cuenta]

IBAN [IBAN]

Editar Mi Cuenta Bancaria

BIC/SWIFT [BIC/SWIFT]

BORRAR

Nombre del Banco [Nombre del banco]

TLF: 987654321

Avd. Carton
16742 FábricaTCGMarket Marca
Registrada

Cuenta del usuario

TCGMarket

Nombre de usuario:

Contraseña:

INICIAR SESIÓN

REGISTRARSE



≡ Juegos

Buscar

Inicio / Cuenta / Perfil

Perfil

Cuenta

[Nombre de Usuario]

[Ver Perfil Publico](#)

Perfil

Nombre

[Nombre del Usuario]

Direcciones

Tipo de Cuenta

[Tipo de cuenta]

Crédito

Fecha de nacimiento

[Fecha de nacimiento]

Transacciones

¿En Vacaciones?

[Estado del usuario]

Email

[Correo Electrónico]

Contraseña

[Contraseña]

Telefono

[Télefono]

TLF: 987654321

Avd. Carton
16742 FábricaTCGMarket Marca
Registrada

Favoritos

El usuario tiene acceso a un sencillo menú de usuario que le permite permutar entre las 5 secciones de usuario y modificarlas. También podrá acceder a los favoritos en el header en el apartado de usuario.

Compra

TCGMarket

- [\[Usuario\]](#)
- [Vender](#)
- [Comprar](#)
- [Carrito](#) ([Precio total])

Juegos Buscar 

Inicio / Usuario / Compras / [Compra]

Compra [Número de la compra]

[Vendedor]

Resumen	Dirección de entrega	Método de envío seleccionado
Contenido	[Nombre]	[Método de envío]
Valor del pedido	[Calle]	
Gastos de envío	[Codigo Postal] [Ciudad]	[Método de envío] 
Total	[País]	Más opciones de envío

Pagado/Sin Pagar

Entrega estimada:

 Más artículos de este vendedor  Artículos del vendedor de mi lista de wants

[Juego] Cartas Sueltas ([nº productos])

Nombre	Información	Precio
<input checked="" type="checkbox"/> [nº] [Nombre del producto] [Expansion] (V.1)	# [Nº Carta] [Rareza] [Expansion] [Estado] [Idioma]	[Descripción] [Precio] [nº] 

TLF: 987654321 Avd. Carton
16742 Fábrica

TCGMarket Marca
Registrada

Venta

TCGMarket
[Usuario]
 Vender
 Comprar
 Carrito
([Precio total])

≡ Juegos

Buscar

Inicio / Usuario / Venta / [Venta]

Venta [Número de la venta]

[Comprador]

Resumen	Dirección de envío	Método de envío a usar
Contenido	[Nombre]	[Método de envío]
Valor del pedido	[Calle]	
Gastos de envío	[Codigo Postal] [Ciudad]	
Total	[País]	
Pagado/Sin Pagar		
Valoración		Comentarios
General:		
Artículo:		
Empaquetado:		
[Juego] Cartas Sueltas ([nº productos])		
Nombre	Información	Precio
<input checked="" type="checkbox"/> [nº] [Nombre del producto] [Expansion] (V.1)	# [Nº Carta] [Rareza] [Expansion] [Estado] [Idioma]	[Precio] [nº] 

TLF: 987654321 Avd. Carton
16742 Fábrica

TCGMarket Marca
Registrada

Carrito

TCGMarket [Usuario] Vender Comprar Carrito (Precio total) Juegos Buscar

Inicio / Carrito de la compra

Carrito de la compra

[Vendedor]

[Quitar artículos de este vendedor]

Resumen

Contenido

Valor del pedido

Gastos de envío

Total

Método de envío seleccionado

[Método de envío]

[Método de envío]

Más opciones de envío

Entrega estimada:

... Más artículos de este vendedor

Artículos del vendedor de mi lista de wants

[Juego] Cartas Sueltas ([nº productos])

Nombre	Información	Precio
[Imagen] [nº] [Nombre del producto] [Expansion] (V.1)	# [Número Carta] [Rareza] [Expansion] [Estado] [Idioma]	[Precio] [nº] <input checked="" type="button"/>

Dirección de entrega

[Nombre]

[Calle]

[Código Postal] [Ciudad]

[País]

Vista general del carrito: [Precio]

Proceder al pago

[Vendedor] [Precio total]

Número de pedidos

Cantidad de artículos

Valor del pedido

Gastos de envío

Total [Precio Total]

TLF: 987654321 Avd. Cartón
16742 FábricaTCGMarket Marca
Registrada

Para terminar, se define la entidad de venta/compra y su listado en el carro de la compra.

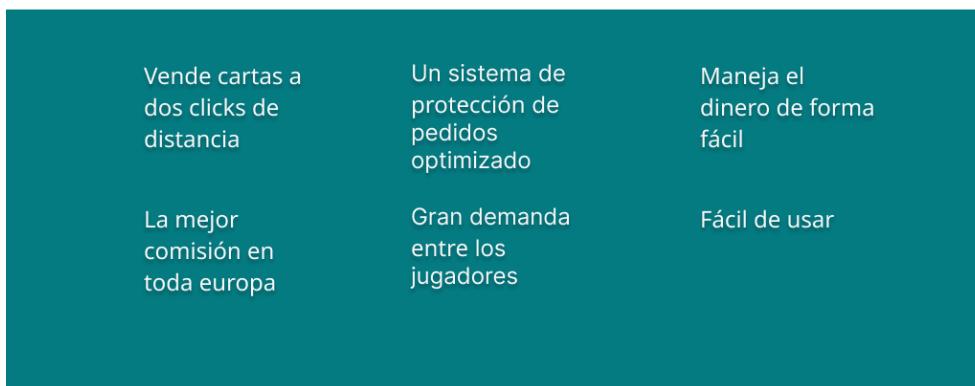
Mock Ups

Indice



The mockup shows the TCGMarket homepage. At the top, there's a navigation bar with 'TCGMarket' on the left, followed by input fields for 'Nombre de usuario' and 'Contraseña', and buttons for 'REGISTRARSE', 'INICIAR SESIÓN', and 'BUSCAR'. Below the header is a search bar with the placeholder 'Buscar en TCGMarket...'. A 'Juegos' button is on the left. On the right, there's a 'BUSCAR' button. The main content area features a teal background with the text 'Bienvenido a TCGMarket, el mejor mercado de cartas de Europa'. Below this are three large cards: 'Vanguard' (Cardfight!! Vanguard), 'Digimon Card Game', and 'Yu-Gi-Oh! Trading Card Game'. Each card has its respective logo and some smaller text.

TCGMarket es el mercado europeo que te permite comprar y vender cartas de manera fácil



This section highlights six key features of TCGMarket:

Vende cartas a dos clicks de distancia	Un sistema de protección de pedidos optimizado	Maneja el dinero de forma fácil
La mejor comisión en toda europa	Gran demanda entre los jugadores	Fácil de usar

400.000+ ventas

150.000+ usuarios

1.000.000+ Cartas vendidas

30 Países



Listo para empezar a usar TCGMarket?!

[Registrate](#)

TCGMarket Teléfono: +34 987654321 Dirección: A Coruña Avd Finisterre Todos los derechos reservados a TCGMarket S.A

Menú Principal

TCGMarket

Nombre de usuario Nombre de usuario Contraseña: Contraseña

[REGISTRARSE](#) [INICIAR SESIÓN](#)

Juegos Buscar en TCGMarket... [BUSCAR](#)

V Clan Collection
Fecha de salida: 4/3/2022



Tendencias

Best Sellers Las Mejores Gangas

					
Crystal Wing Synchr... DUPO-EN068 2,00€					
BT08 Magnamon 12€					
<input checked="" type="checkbox"/> BT08 Magnamon 12€					
BT08 Magnamon 12€					

Novedades

[Verlo Todo](#)

NEW HERO

BT08 Magnamon 12€
<input checked="" type="checkbox"/> BT08 Magnamon 12€
BT08 Magnamon 12€
<input checked="" type="checkbox"/> BT08 Magnamon 12€
BT08 Magnamon 12€
BT08 Magnamon 12€

Expansiones

[Ver Expansiones](#)



NEW HERO



TCGMarket Teléfono: +34 987654321 Dirección: A Coruña Avd Finisterre Todos los derechos reservados a TCGMarket S.A.

Registro

TCGMARKET

Cuenta particular

Cuenta profesional

Nombre *

Nombre

Apellidos

Fecha de nacimiento *

dd/mm/aaaa



Dirección *

Calle*

Numero de casa / Piso*

Calle

NºCasa / Piso

Código Postal*

Ciudad*

Código postal

dd/mm/aaaa

País *

País

Dirección Email *

Email (tcgmarket@example.com)

Elija un nombre de usuario y una contraseña *

Nombre de Usuario

Email

Email

REGISTRARSE

Lista de producto



Nombre de usuario

Contraseña:

[REGISTRARSE](#)

[INICIAR SESIÓN](#)

Inicio/Lista

[Buscar en TCGMarket...](#)

Contraseña:

[REGISTRARSE](#)

INICIAR SESIÓN

Lista de Cartas

Buscar por nombre

Expansión

Todos
▼
 Nombre Exacto

Ordenar por

Todos
▼

BUSCAR

99 Resultados

< Página 2 de 8 >



Información Producto

TCGMarket

Nombre de usuario

Nombre de usuario

Contraseña:

Contraseña

REGISTRARSE

INICIAR SESIÓN

≡ Juegos

Buscar en TCGMarket...

BUSCAR

Inicio / BT08 / Magnamon

Magnamon

BT08 - NEW HERO



Información

Observaciones

Rareza: Super Rare

Número: BT8-038

Impreso en: New Hero

Reimpreso en: BT8

Artículos Disponibles: 9

Desde: 12€

Vendedor	Información del Producto	Oferta
	Lore ipsum	
	Lore ipsum	
	Lore ipsum	

TCGMarket

Teléfono: +34
987654321Dirección: A
Coruña Avd
FinisterreTodos los derechos
reservados a
TCGMarket S.A

Menú Expansion

TCGMarket

Nombre de usuario

Nombre de usuario

Contraseña:

Contraseña

REGISTRARSE

INICIAR SESION

≡ Juegos

Buscar en TCGMarket...

BUSCAR



Cartas



Sobres



Caja

Información del Set

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer ultricies, nibh in lobortis blandit, justo sapien accumsan arcu, vel tristique massa odio lacinia neque. Donec dapibus mauris sit amet tempor rhoncus. Pellentesque posuere pulvinar odio. Suspendisse sed libero ac massa vulputate dictum. Proin tempus bibendum dui, vel pharetra odio ultricies rhoncus. Nulla dolor ante, lacinia fringilla mi et, congue iaculis diam. Suspendisse molestie, purus ut pretium facilisis, enim lectus pulvinar turpis, vestibulum sagittis tellus quam sit amet nulla.

Proin sed elit mattis, tristique nibh sit amet, fermentum tellus. Suspendisse ultrices nisl nec nibh bibendum mattis. Nullam sit amet tortor varius, congue erat vitae, porta metus. Integer tempor sollicitudin tincidunt. In nec ante in metus fermentum porta. Phasellus ac placerat nibh. Nulla feugiat nisl sagittis suscipit dictum. Vestibulum et aliquet elit, nec blandit nulla. Nunc turpis leo, porta ac eleifend a, ultrices id nibh

TCGMarket

Teléfono: +34
987654321Dirección: A
Coruña Avd
FinisterreTodos los derechos
reservados a
TCGMarket S.A

Lista de vendedores

TCGMarket

Nombre de usuario

Nombre de usuario

Contraseña:

Contraseña

REGISTRARSE

INICIAR SESIÓN

≡ Juegos

Buscar en TCGMarket...

BUSCAR

Inicio / Vendedores

BUSQUEDA DE VENDEDORES

Nombre de usuario

País

E.j Fernando

España

BUSCAR

99 Resultados

Nombre de usuario	Miembro desde	Ventas	Artículos disponibles
-------------------	---------------	--------	-----------------------

< Página 2 de 8 >

TCGMarket

Teléfono: +34
987654321Dirección: A
Coruña Avd
FinisterreTodos los derechos
reservados a
TCGMarket S.A

Información de Vendedor

TCGMarket

REGISTRARSE

INICIAR SESIÓN

Juegos

Buscar en TCGMarket...

BUSCAR

Fernando

[Nombre de Establecimiento]
OverlordTheGreat

Particular 500 Ventas
Miembro desde 10/10/2019

Mi casa
Mi piso
España

Tiempo estimado de llegada
de pedidos: 5 días

Inicio / Busqueda de Usuarios / OverlordTheGreat

Articulos



CARTAS

[Imagen Sobre]

SOBRES

[Imagen Caja]

CAJAS

Ultimas valoraciones de Fernando

Comprador	Fecha	Comentarios	Valoraciones
Lorem Ipsum	Lorem Ipsum	Lorem Ipsum	✓
			✓
			✓

Venta

TCGMarket

[REGISTRARSE](#)
[INICIAR SESIÓN](#)

≡ Juegos

[BUSCAR](#)

Inicio / Monica / Venta / Venta #483484]

Venta #483484

Overlord The Great

Resumen	Dirección de entrega	Método de envío requerido
Contenido	2 Artículos Fernando	Certificado - 5,00€
Valor del pedido	24€	Mi Calle
Gastos de envío	5€	23741 Mi Ciudad
Total	29€	España
Entrega estimada:	En 4 días	Más opciones de envío
Pagado	Enviado en:	
	17/04/2022	
Valoración:		
General: Excelente	Comentario	
Artículo: Excelente	Buena entrega, muy amable enviandome las fundas	
Empaquetado: Excelente		

Cartas Sueltas (2)

Nombre	Información	Precio
 Magnamon - BT08 - SR	#038 Near Mint NEW HERO English	12€ 2 ×

TCGMarket
Telefono: +34 987654321
Dirección: A Coruña Avd Finisterre
Todos los derechos reservados a TCGMarket S.A

Compra

TCGMarket

Nombre de usuario

Nombre de usuario

Contraseña:

Contraseña

REGISTRARSE

INICIAR SESIÓN

≡ Juegos

Buscar en TCGMarket...

BUSCAR

Inicio / Overlord The Great / Compras / Compra #345245

Compra #345245

Overlord The Great

Resumen	Dirección de entrega	Método de envío seleccionado
Contenido	2 Artículos Fernando	Certificado - 5€
Valor del pedido	24€	Mi Calle
Gastos de envío	5€	23741 Mi Ciudad
Total	29€	España
Entrega estimada:	En 4 días	

Entrega estimada: En 4 días

Pagado/Sin Pagar

... Más artículos de este vendedor  Artículos del vendedor de mi lista de favoritos

Cartas Sueltas (2)

Nombre	Información	Precio
 Magnamon - BT08 - SR	#038 Near Mint NEW HERO English	En muy buen estado 12€  2 

TCGMarket

Teléfono: +34
987654321

Dirección: A
Coruña Avd
Finisterre

Todos los derechos
reservados a
TCGMarket S.A



Menú de Usuario

TCGMarket

Nombre de usuario

Nombre de usuario

Contraseña:

Contraseña

REGISTRARSE

INICIAR SESION

≡ Juegos

Buscar en TCGMarket...

BUSCAR

Inicio / Cuenta / OverlordTheGreat

Cuenta

Cuenta

Overlord The Great

Ver Perfil Publico

Perfil

Nombre Fernando Gonzalez

Direcciones

Tipo de Cuenta Paritcular

Crédito

Fecha de nacimiento 17/6/1999

Transacciones

Activo

Direccion

Mi casa

Mi calle

19877

Mi Ciudad

Mi País

Balance de Cuenta 90€

TCGMarket

Telefono: +34
987654321Dirección: A
Coruña Avd
FinisterreTodos los derechos
reservados a
TCGMarket S.A

Favoritos

Carrito

TCGMarket
Nombre de usuario
Contraseña:
REGISTRARSE
INICIAR SESIÓN

≡Juegos
Buscar en TCGMarket...
BUSCAR

Inicio / Carrito

Carrito de la compra

Overlord The Great

Resumen

Contenido	2 Artículos
Valor del pedido	24€
Gastos de envío	5€
Total	29€

Entrega estimada: En 4 días

[Quitar artículos de este vendedor](#)

[Método de envío seleccionado](#)

Ordinario

Método de Envío

Más opciones de envío

... Más artículos de este vendedor Artículos del vendedor de mi lista de wants

Cartas Sueltas (2)

Nombre	Información	Precio
Magnamon - BT08 - SR	#038 Near Mint NEW HERO Language	En muy buen estado 12€ 2

Dirección de entrega

Fernando
Mi casa
18236 Mi Piso
Mi País

Vista general del carrito: [Precio]

REALIZAR PAGO

Overlord	24€
The Great	[Precio total]
[Vendedor]	

Número de pedidos	1
Cantidad de artículos	2
Valor del pedido	24€
Gastos de envío	5€

Total	29€
-------	-----

TCGMarket Teléfono: +34 987654321 Dirección: A Coruña Avd Finisterre Todos los derechos reservados a TCGMarket S.A.

Tablas de datos

TCGMarket

Nombre de
usuario

Nombre de
usuario

Contraseña:

Contraseña

REGISTRARSE

INICIAR SESIÓN

≡ Juegos

Buscar en TCGMarket...

BUSCAR

Inicio / Tabla de datos

Tablas de datos

[Conjunto de datos concreto]

[Conjunto de datos concreto]

[Conjunto de datos concreto]

TCGMarket

Teléfono: +34
987654321

Dirección: A
Coruña Avd
Finisterre

Todos los derechos
reservados a
TCGMarket S.A

Resultado de Tablas de datos

TCGMarket

≡ Juegos
Buscar en TCGMarket...
BUSCAR

Inicio /Tabla de datos / Mejores Gangas

Mejores gangas.

Expansion	Carta	Precio Actual
Lorem Ipsum	Lorem Ipsum	Lorem Ipsum
Lorem Ipsum	Lorem Ipsum	Lorem Ipsum
Lorem Ipsum	Lorem Ipsum	Lorem Ipsum

TCGMarket Teléfono: +34 987654321 Dirección: A Coruña Avd Finisterre Todos los derechos reservados a TCGMarket S.A

Una vez finiquitado el diseño, se procede a desarrollar el contorno servidor donde se manipulan los datos.

Back End

El Contorno Servidor o también llamado Back End constituye de varios elementos fundamentales para la comunicación entre servidor y cliente.

El Back End se diseñará en Spring dado que es una de las herramientas más eficientes para hacer un Back End sólido.

Para empezar, Spring adapta la estructura de las bases de datos a Java, es decir, las entidades se convierten en la unidad compleja mínima de Java: Clases

Modelos

Los modelos representan a las entidades en modelo entidad relación de la base de datos.

Son clases cuyos parámetros son adaptaciones de los atributos de la base de datos como se puede ver en la siguiente imagen:

```
@Entity  
@Table(name = "expansions")  
public class Expansion{  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    public Integer id;  
    @Column(unique=true)  
    public String code;  
    public String name;  
    @ManyToOne(cascade = CascadeType.MERGE)  
    @JoinColumn(name="idGame")  
    public Juego juego;  
    public String image;  
    @JsonFormat(pattern="dd-MM-yyyy")  
    public Date releasedate;
```

Spring, al ser completamente interpretativo, es muy sensible en términos de escritura por lo tanto el nombre de los atributos de la clase deben ser iguales a los atributos de la tabla

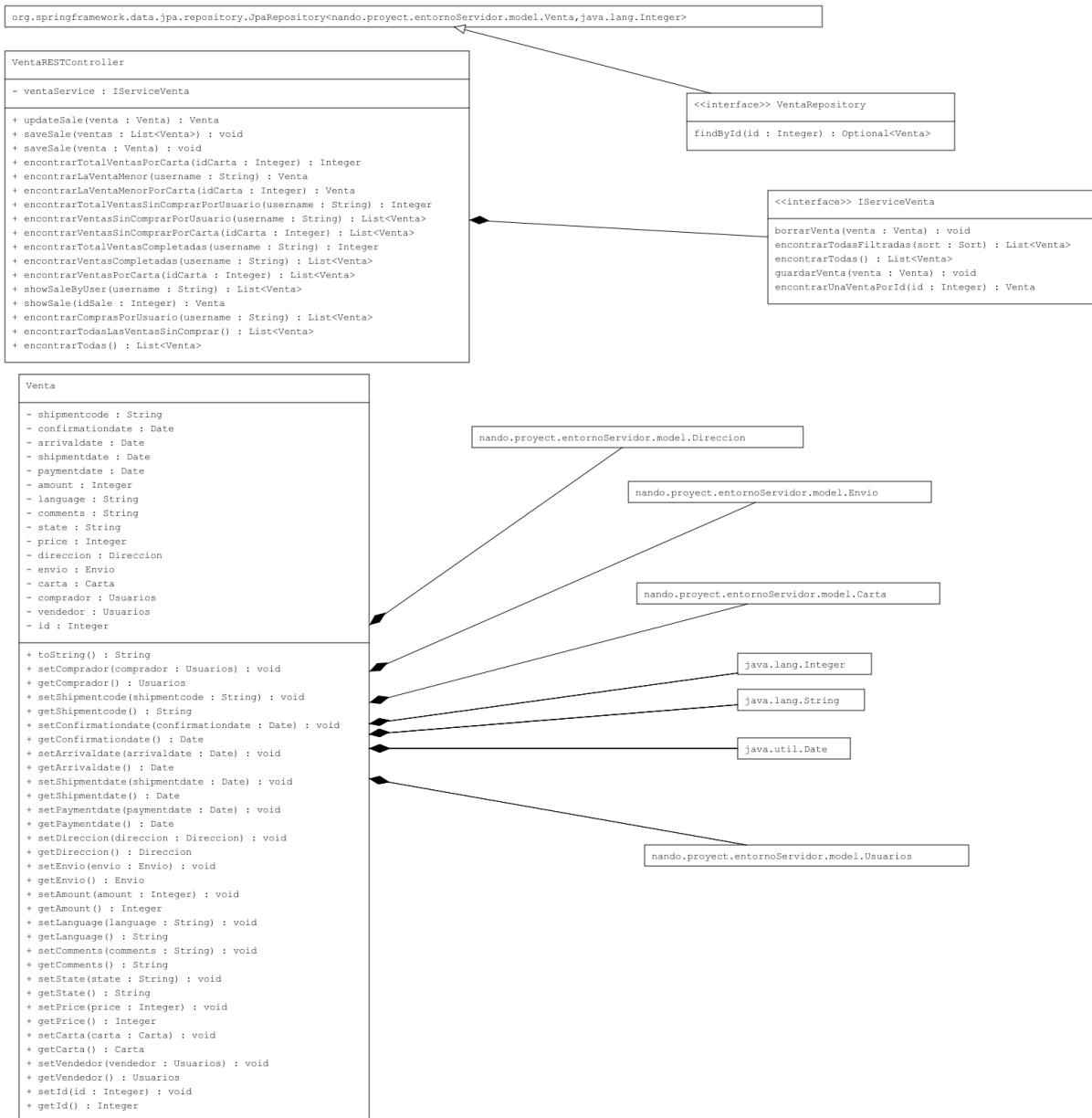
Spring usa las **anotaciones** (E.j: `@Entity`) para que el framework reconozca que parte debe vincular a los elementos de la base de datos.

- `@Entity` indica que la clase es una entidad y `@Table` lo vincula a la tabla
- `@Id` y `@GeneratedValue` genera el Id de la entidad junto a su estrategia de almacenamiento (`Auto_Increment`)
- `@ManyToOne`, `@ManyToMany` y `@OneToOne` introducen la relación de las entidades como objetos para poder interactuar con ellos usando `@JoinColumn` y `@inverseJoinColumn`
- `@Column` sirve para indicar condiciones extra de la columna

Todas las entidades se han hecho de igual forma: Se generan Setters, Getters y los métodos necesarios para formatear los datos e inicializarlos de forma correcta.

Proyecto Desarrollo de Aplicaciones Web

Adjunto ejemplo del UML para visualizarlo mejor:



Una vez establecidos los métodos, se desarrollará los elementos que permitirán hacer las consultas contra la base de datos:

Repositorios

Los repositorios son interfaces diseñadas por Spring para realizar las consultas básicas de una base de datos. Al implementar los modelos de la base de datos, Spring detectará automáticamente las entidades y generará métodos heredados de la interfaz base JDBC aplicándolos a los atributos de tus entidades de tu base de datos usando la Interfaz más avanzada llamada JPA. Los repositorios se detectan usando la anotación de Spring `@Repository`

Como se puede ver aquí:

```
@Repository
public interface CartaRepository extends JpaRepository<Carta, Integer> {
    List<Carta> findAllByExpansion(Integer expansion);
    List<Carta> findByNameContaining(String parteNombre);
    List<Carta> findByCode(String codigo);
    List<Carta> findByReprint(String reprint);
    List<Carta> findByHighlighted(Boolean destacado);

}
```

La interfaz extiende de la entidad padre `JPARepository` que a su vez hereda de `JDBC`, permitiendo hacer consultas sencillas personalizables detectando el nombre de tus entidades. Como Spring usa métodos que interpretan la escritura, es muy sensible a cómo se escriben las entidades en los métodos (Exactamente como en los modelos) así que se debe integrar a la perfección las entidades correspondientes.

Spring tiene palabras clave para poder diseñar consultas varias como por ejemplo:

```
boolean existsByUsername(String username);

boolean existsByEmail(String email);
```

Comprobar que existe un usuario o un email.

Después de los repositorios, pasamos a las clases que preparan el formato de la información y filtra las consultas tiradas contra bases de datos:

Servicios

Los servicios consisten en dos partes: **Las clases y las interfaces**.

Las clases tienen los métodos donde se trata la información y las interfaces solo las implementa, no las muestra. Por ende, no muestra como se trata la información añadiendo una capa de seguridad extra al proyecto. Las clases Servicio se inicializan con una anotación @Service para que Spring la reconozca como tal.

En este momento es cuando sale a lucir una de las mayores características de Spring: **@Autowired**. @Autowired inyecta toda la clase iniciando por completo todos sus métodos y parámetros facilitando la visibilidad del código.

Las clases, como ya se ha visto anteriormente, implementan la interfaz de los servicios y el autowired inicia el repositorio para poder hacer consultas básicas.

```
public interface IServiceVenta {  
    Venta encontrarUnaVentaPorId(Integer id);  
    void guardarVenta(Venta venta);  
    List<Venta> encontrarTodas();  
    List<Venta> encontrarTodasFiltradas(Sort sort);  
    void borrarVenta(Venta venta);  
}
```

```
@Service  
public class ServiceVentaImpl implements IServiceVenta {  
    @Autowired  
    private VentaRepository ventaRepository;
```

```
@Override  
public List<Venta> encontrarTodasFiltradas(Sort sort) {  
    return ventaRepository.findAll(sort);  
}
```

Una vez se trata la información, se enviará al usuario a través de los:

Controladores

Los controladores son las clases que definirán por donde se realizarán las peticiones contra la base de datos. Se deberá usar un mismo lenguaje para que Cliente y Servidor puedan interpretar los datos de las peticiones y ese lenguaje es **JSON (JavaScript Object Notation)**. Es la unidad de comunicación entre Cliente y Servidor más usada actualmente.

Su estructura:

```
{
  "id": 4,
  "usuario": { ... },
  "addressname": "Mi dimension",
  "name": "DIMENSIONAL ROBO",
  "surname": "Amogus",
  "street": "34",
  "number": "5",
  "floor": "2º",
  "postalcode": "15172",
  "city": "A Coruña",
  "province": "Galicia",
  "country": "España",
  "comments": "",
  "preferred": false
}
```

Para que Spring reciba y devuelva objetos JSON se deberá anotar los controladores con `@RestController`.

Al trabajar con peticiones de red, hay varias capas de seguridad básicas que aseguran los ordenadores. Para esta demo se ignorara el CORS (Cross-Origin) y el CSRF (Cross-Site Request Forgery). El CSRF se desactivará en la seguridad pero el CORS se puede desactivar con un anotación `@Cross-Origins`.

Se delimitará por donde se establece la comunicación entre Cliente y Servidor mediante el mapeo de rutas de petición. Son estructuras que delimitan el acceso a un Controlador y se pueden establecer con la anotación `@RequestMapping`

```
@RestController
@RequestMapping("/api/auth")
@CrossOrigin(origins = "http://localhost:3000", methods= {RequestMethod.GET,RequestMethod.POST, RequestMethod.PUT})
public class AuthControlador {
```

Las anotaciones del nivel jerárquico superior se aplican a los métodos de la clase así que no se tendrá que repetir código. Cada método tiene su mapeo de ruta que se combina con la de la clase para formar rutas completas (Ej. Padre: `/api/auth/` y Hijo: `/iniciarsesion` forman `api/auth/iniciarsesion` para poder acceder a la ruta)

Proyecto Desarrollo de Aplicaciones Web

Se inyectan las interfaces de Servicios y se procede a preparar el paquete de JSON:

```
@PostMapping("/registrar")
public ResponseEntity<> registrarUsuario(@RequestBody RegistroDTO registroDTO){
    if(usuarioRepositorio.existsByUsername(registroDTO.getUsername())){
        return new ResponseEntity<>(body: "Ese nombre de usuario ya existe", HttpStatus.BAD_REQUEST);
    }

    if(usuarioRepositorio.existsByEmail(registroDTO.getEmail())){
        return new ResponseEntity<>(body: "Ese email de usuario ya existe", HttpStatus.BAD_REQUEST);
    }

    Usuarios usuario = new Usuarios();
    usuario.setName(registroDTO.getNombre());
    usuario.setSurname(registroDTO.getApellido());
    usuario.setUsername(registroDTO.getUsername());
    usuario.setEmail(registroDTO.getEmail());
    usuario.setBirthdate(registroDTO.getFechaDeNacimiento());
    usuario.setStatus(registroDTO.getStatus());
    usuario.setPassword(passwordEncoder.encode(registroDTO.getPassword()));
    Collection<Perfil> perfiles = new ArrayList<>();
    perfiles.add(perfilRepository.findByProfile(perfil: "USUARIO"));
    usuario.setPerfiles(perfiles);
    usuario.setRegisterdate(new Date());

    usuarioRepositorio.save(usuario);
    return new ResponseEntity<>(body: "Usuario registrado exitosamente", HttpStatus.OK);
}
```

En el ejemplo se usa un DTO: El DTO es un patrón que se define para suavizar la comunicación entre Cliente y Servidor, es decir, formatea los datos para Java para que haya la menor cantidad de errores al pasar cualquier entrada del Cliente:

```
package nando.proyect.entornoServidor.dto;

public class LoginDTO {

    private String usernameOrEmail;
    private String password;

    public String getUsernameOrEmail() {
        return usernameOrEmail;
    }

    public void setUsernameOrEmail(String usernameOrEmail) {
        this.usernameOrEmail = usernameOrEmail;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

}
```

Al usar el `@Autowired`, el dedicar una clase entera a formatear información es más viable que nunca porque se puede inicializar la clase en dos líneas.

Una vez se preparan los paquetes y se devuelven o guardan objetos JSON correctamente, pasaremos a asegurarlo con:

Seguridad

Dado que la comunicación entre Cliente y Servidor es técnicamente abierta, se puede acceder desde cualquier parte a cualquier ruta si es conocedor de estas.

Para evitar esa casuística, se implementará seguridad.

Por ahora, la seguridad en este proyecto se implementará solo para autenticar usuarios y devolver un JWT (Json Web Token) que serán datos personales del usuario codificados para asegurar la información del usuario.

El proceso se seguiría de esta manera:

1. Se recibe el usuario y se inicializa en un DTO que a su vez se inicializa en un Objeto User para formatearlo a un Objeto UserDetails donde se coge el usuario, la contraseña y se mapean los perfiles del usuario

```

@PostMapping("/iniciarSesion")
public ResponseEntity<JWTAuthResponseDTO> authenticateUser(@RequestBody LoginDTO loginDTO){
    Authentication authentication = authenticationManager.authenticate(new UsernamePasswordAuthenticationToken(loginDTO.getUsernameOrEmail(), loginDTO.getPassword()));

    SecurityContextHolder.getContext().setAuthentication(authentication);

    //obtenemos el token del jwtTokenProvider
    String token = jwtTokenProvider.generarToken(authentication);
    return ResponseEntity.ok(new JWTAuthResponseDTO(token));
}

public class JWTAuthResponseDTO {

    private String tokenDeAcceso;
    private String tipoDeToken = "Bearer";

    public JWTAuthResponseDTO(String tokenDeAcceso) {
        super();
        this.tokenDeAcceso = tokenDeAcceso;
    }

    public JWTAuthResponseDTO(String tokenDeAcceso, String tipoDeToken) {
        super();
        this.tokenDeAcceso = tokenDeAcceso;
        this.tipoDeToken = tipoDeToken;
    }

    public String getTokenDeAcceso() {
        return tokenDeAcceso;
    }

    public void setTokenDeAcceso(String tokenDeAcceso) {
        this.tokenDeAcceso = tokenDeAcceso;
    }

    public String getTipoDeToken() {
        return tipoDeToken;
    }
}

```

2. El UserDetails se usa en un método de autenticación inicializado que identificará los datos proporcionados

Proyecto Desarrollo de Aplicaciones Web

```

public class JwtAuthenticationFilter extends OncePerRequestFilter{

    @Autowired
    private JwtTokenProvider jwtTokenProvider;

    @Autowired
    private CustomUserDetailsService customUserDetailsService;

    @Override
    protected void doFilterInternal(HttpServletRequest request, HttpServletResponse response, FilterChain filterChain)
        throws ServletException, IOException {
        //obtenemos el token de la solicitud HTTP
        String token = obtenerJWTdeLaSolicitud(request);

        //validamos el token
        if(StringUtils.hasText(token) && jwtTokenProvider.validarToken(token)) {
            //obtenemos el username del token
            String username = jwtTokenProvider.obtenerUsernameDelJWT(token);

            //cargamos el usuario asociado al token
            UserDetails userDetails = customUserDetailsService.loadUserByUsername(username);
            UsernamePasswordAuthenticationToken authenticationToken = new UsernamePasswordAuthenticationToken(userDetails, credentials: null, userDetails.getAuthorities());
            authenticationToken.setDetails(new WebAuthenticationDetailsSource().buildDetails(request));

            //establecemos la seguridad
            SecurityContextHolder.getContext().setAuthentication(authenticationToken);
        }
    }
}
  
```

3. Devolverá error si son incorrectos pero si son correctos, devolverá un token de acceso que proporcionará los roles necesarios al usuario y devolverá una clave única con la que se filtrarán los datos en la comunicación de Cliente-Servidor (En este caso, se usará Username)

Con la seguridad, se termina el Back End y se pasa al:

Front End

Para el front end, la tecnología que se ha escogido es React JS porque es un motor complicado de usar pero MUY potente permitiendo hacer muchas casuísticas e implementaciones de diseño con JavaScript, HTML y CSS. Se usará la librería de CSS **Bootstrap** para refinar los componentes del front end.

React JS se estructura en componentes. Cada componente implica un código que se ejecuta independientemente de otros elementos y es inmutable salvo por el propio componente.

Mientras que React básico tienes que indicar la inicialización de cada instancia del componente con una clase, React JS te permite usar Hooks, una herramienta que te permite instanciar elementos sin inicializar una clase. Hooks es una parte trascendental en el proyecto porque así se conseguirá que los elementos cambien en base a ciertos componentes y que varíen en base a las adiciones a la base de datos.

También se usa Router-Dom que permite generar rutas dinámicas para pasar información entre componentes dinámicos.

La estructura del Front End es sencilla:

Tenemos un enrutador que envuelve al elemento central de la aplicación permitiendo hacer la aplicación siempre que se acceda a la ruta que hayas determinado:

```
<BrowserRouter>
  <header>
    <Header username={user} cartLength={cartItems.length} />
  </header>
  <Routes>
    <Route path="/" element={<Navigate to='/home' />} />
    <Route path="/home" element={<Home username={user} />} />
    <Route path="/login" element={<Login usuario={user} setUsuario={setUser}>/} />
    <Route path="/signup" element={<Signup />} />
    <Route path="/card/:cardId" element={<CardInfo onAdd={onAdd} username={user} cartItems={cartItems} />} />
    <Route path="/cards/" element={<Cards />} />
    <Route path="/cards/search/:search" element={<SearchResult />} />
    <Route path="/expansions/" element={<Expansions />} />
    <Route path="/expansions/:expansionId" element={<ExpansionInfo />} />
    <Route path="/games/" element={<Games />} />
    <Route path="/games/:gameId" element={<GameInfo />} />
    <Route path="/shoppingCart" element={<ShoppingCart cartItems={cartItems} onRemove={onRemove} clearStorage={CartContext.clearStorage} />} />
    <Route path="/logout" element={<Navigate to='/login' />} />
    <Route path="/account/:user" element={<Account username={user} />} />
    <Route path="/account/:user/address=:addressid" element={<AccountAddressInfo username={user} />} />
    <Route path="/account/:user/:userId/createAddress" element={<AccountAddressCreate username={user} />} />
    <Route path="/favorites/:user" element={<Favorites username={user} />} />
    <Route path="/purchases/:user" element={<Purchases username={user} />} />
    <Route path="/purchases/:user/purchaseId=:purchaseId" element={<PurchaseInfo username={user} />} />
    <Route path="/purchases/:user/gateway" element={<PurchaseGateway cartItems={cartItems} username={user} />} />
    <Route path="/sales/:username" element={<Sales />} />
```

El header está presente junto al footer en todos los elementos y se renderizan desde el primer momento.

Para modificar componentes o variables externas a un componente se deben inicializar en el propio componente donde se vayan a modificar:

```
<Route path="/login" element={<Login usuario={user} setUsuario={setUser}>/}> />
```

Para recibir los datos, uso la función use Effect y use State. La variable a modificar se inició con el use State permitiendo que el atributo sea modificable y el use Effect le asigna un valor a esa instancia.

```
const {game} = useParams();
const [cards, setCards] = useState([]);
const [loading, setLoading] = useState(false);
const [search, setSearch] = useState("");
const [busqueda, setBusqueda] = useState("");

useEffect(() => {
    setLoading(true)
    search === undefined && fetch(`http://localhost:8080/apicartas/cartas`)
        .then(response => response.json())
        .then(cards => setCards(cards))
        .then(() => setLoading(false));
}, []);
if (loading) {
    return <h2>Loading....</h2>
}
```

Aquí se inicia search y cards. Search se modifica con un onChange que se activa cada vez que la entrada de búsqueda se modifica y cada vez que se modifica Search, se modifica Cards con una distinta consulta a la base de datos.

Para mostrar los datos recibidos, mapeo el objeto JSON almacenado en la variable correspondiente y asignar los valores que le corresponde a cada elemento por la key

Proyecto Desarrollo de Aplicaciones Web



```
<Table striped bordered hover responsive="xl" className="pt-3">
  <thead>
    <tr>
      <th className="font-link">Vendedor</th>
      <th className="font-link">Información del producto</th>
      <th className="font-link">Oferta</th>
      <th className="font-link">Precio</th>
    </tr>
  </thead>
  <tbody>
    {sales.map(sale => <tr key={sale.id}>
      <td className="font-link"> <Link to={`/sellers/${sale.vendedor.id}`}>{sale.vendedor.username}</Link></td>
      <td className="font-link">Estado:{sale.state}, Lenguaje: {sale.language}, Comentarios: {sale.comments}</td>
      <td className="font-link">
        {sale.amount}
      </td>
      <td className="font-link">{sale.price}€</td>
      <td className="font-link">
        <Button onClick={() => addToCart(sale)}>Comprar</Button>
      </td>
    </tr>)}
  </tbody>
</Table>
```

```
useEffect(() => {
  fetch(`http://localhost:8080/sale/total/cardId=${carta.id}`)
    .then(response => response.json())
    .then(totalAmount => setTotalAmount(totalAmount))
}, []);
const userCard = (
  <>
    <tr>
      <td>{carta.expansion.juego.name}</td>
      <td className="font-link">
        <Link to={`/expansions/${carta.expansion.id}`}>
          {carta.expansion.code}
        </Link>
      </td>
      <td className="font-link">
        <Link to={`/card/${carta.id}`}>
          {carta.name}
        </Link>
      </td>
      <td className="font-link">{carta.code}</td>
      <td className="font-link">{totalAmount}</td>
      <td className="font-link">{sales.price}€</td>
    </tr>
  </>
)
```

Para tratar la casuística del carrito de la compra y la autorización del usuario, inicio dos variables en el componente padre y los actualizo en los componentes específicos.

Proyecto Desarrollo de Aplicaciones Web

```
const App = () => {

  const [cartItems, setCartItems] = useState([]);
  const [user, setUser] = useState({});

  const onAdd = (product) => {
    const exist = cartItems.find((x) => x.id === product.id);
    console.log(exist);
    if (exist) {
      setCartItems(
        cartItems.map((x) =>
          x.id === product.id ? { ...exist, qty: exist.qty + 1 } : x
        )
      );
    } else {
      setCartItems([...cartItems, { ...product, qty: 1 }]);
    }
  };

  const onRemove = (product) => {
    const exist = cartItems.find((x) => x.id === product.id);
    if (exist.qty === 1) {
      setCartItems(cartItems.filter((x) => x.id !== product.id));
    } else {
      setCartItems(
        cartItems.map((x) =>
          x.id === product.id ? { ...exist, qty: exist.qty - 1 } : x
        )
      );
    }
  };
}
```

Proyecto Desarrollo de Aplicaciones Web

```

<Route path="/card/:cardId" element={<CardInfo onAdd={onAdd} username={user} cartItems={cartItems} />} />

const CardSalesList = (props) => {
  const { onAdd, sales, cartItems } = props;
  const auth = useSelector((state) => state.auth);
  const list = (
    <Table striped bordered hover responsive="xl" className="pt-3">
      <thead>
        <tr>
          <th className="font-link">Vendedor</th>
          <th className="font-link">Información del producto</th>
          <th className="font-link">Oferta</th>
          <th className="font-link">Precio</th>
        </tr>
      </thead>
      <tbody>
        {sales.map(sale => <tr key={sale.id}>
          <td className="font-link"> <Link to={`/sellers/${sale.vendedor.id}`}>{sale.vendedor.username}</Link></td>
          <td className="font-link">Estado:{sale.state}, Lenguaje: {sale.language}, Comentarios: {sale.comments}</td>
          <td className="font-link">
            {sale.amount}
            /* <select class="form-select" aria-label="Elige tu forma de envío">
              {getAmount(sale.amount).map((amount) =>
                <option key={amount} value={amount} onClick={() => setActualAmount(amount)}>{amount}</option>
              )}
            </select> */
          </td>
          <td className="font-link">{sale.price}€</td>
          <td className="font-link">
            <Button onClick={() => addToCart(sale)}>Comprar</Button>
          </td>
        </tr>)
      </tbody>
    </Table>
  );
}

function addToCart(actualSale) {
  const exist = cartItems.find(x => x.id === actualSale.id);
  if(exist){
    Swal.fire(
      'Esta venta ya está en tu carrito!',
      'No puedes repetir ventas',
      'error'
    )
  }else{
    // ...
  }
}

<Route path="/login" element={<Login usuario={user} setUsuario={setUser}> />} />

```

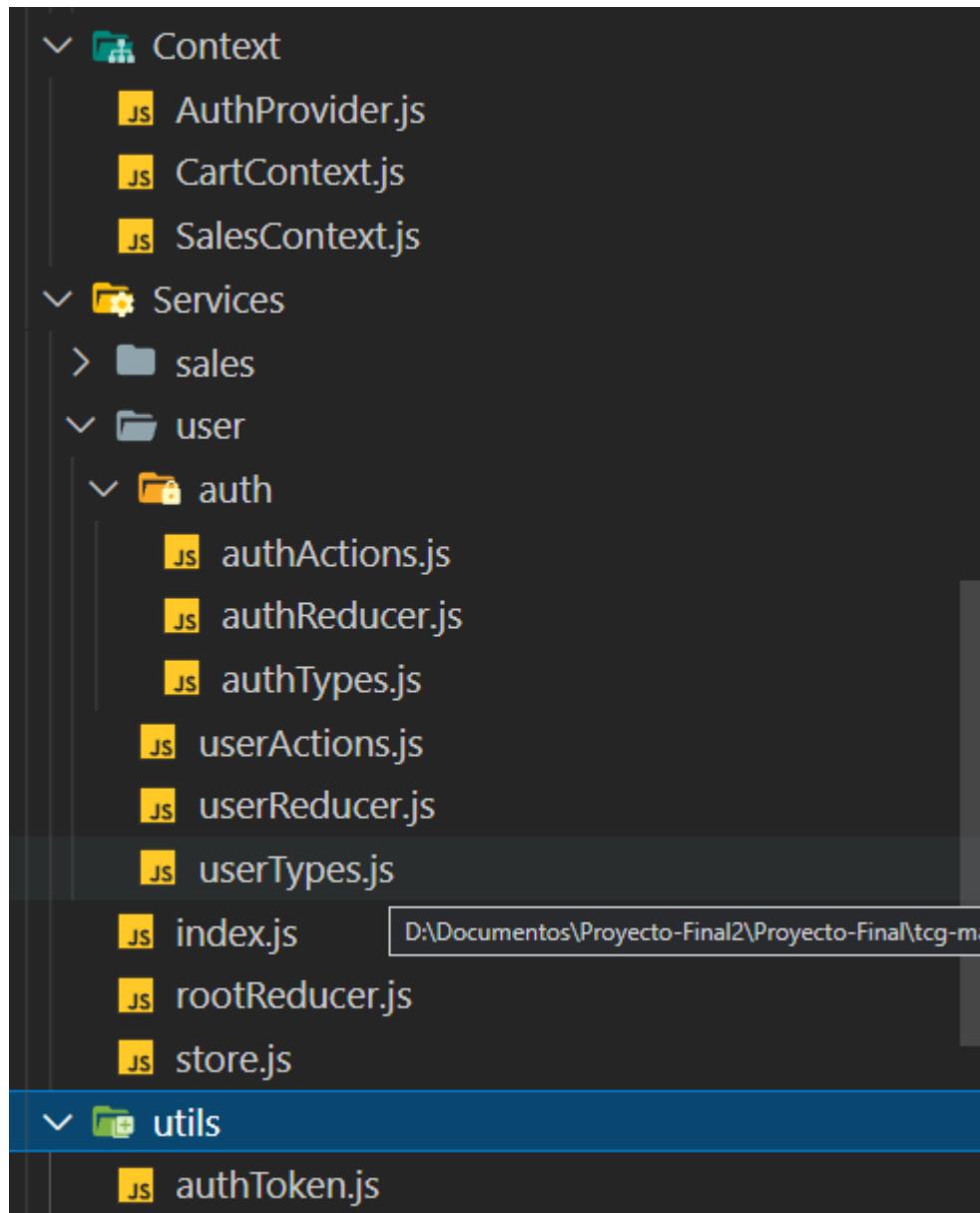
```
const validateUser = () => {
    dispatch(authenticateUser(user.usernameOrEmail, user.password))
        .then((response) =>
            console.log();
            setUsuario(parseJwt(localStorage.getItem("jwtToken")).sub);
            Swal.fire(
                'Has iniciado sesión correctamente',
                'Paselo bien',
                'success'
            )
            setUser(initialState);
        )
        .catch((error) => {
            console.log(error.message);
            setShow(true);
            resetLoginForm();
            Swal.fire(
                'Las credenciales son incorrectas',
                'Intentalo de nuevo',
                'error'
            )
        })
};
```

Para insertar datos en la base de datos, se declara una clase plantilla para poder enviar los datos como Spring los acepta. Dado que tenemos relaciones de N:1 y N:M, los JSON se anidan ergo se tiene que construir el objeto correctamente como un JSON anidado en vez de pasar el ID directamente dado que Spring recibe y devuelve los objetos como JSON.

```
const {username, id} = useParams();
const [cards, setCards] = useState([]);
const swalWithBootstrapButtons = Swal.mixin({
    customClass: {
        confirmButton: 'btn btn-success',
        cancelButton: 'btn btn-danger'
    },
    buttonsStyling: false
});
const initialState = {
    "vendedor": {
        "id": `${id}`,
    },
    "carta": {
        "id": 1,
    },
    "price": "",
    "state": "NEAR_MINT",
    "comments": "",
    "amount": "",
    "language": "SPANISH",
    "envio":null,
    "direccion":null,
};
```

Para manejar el sistema de login, se creó un Contexto. Los contextos en ReactJS son clases que rodean por completo al proyecto y se puede llamar de forma global.

Se comporta de forma similar a los servicios de Java, se llama a los métodos a través de la interfaz pero la herencia hace que no se puedan ver los métodos. Así realizamos la autenticación sin acceder directamente a la información del usuario.



Este conjunto decodifica el JW Token y lo inicializa como variable global accesible para ver si el usuario ha iniciado sesión.

```
const dispatch = useDispatch();
```

```
const auth = useSelector((state) => state.auth);
```

```
{auth.isLoggedIn ? userLinks : guestLinks}
```

Proyecto Desarrollo de Aplicaciones Web

Otra característica que se ha logrado en el proyecto son las alertas estilizadas.

Por ejemplo: Al iniciar sesión con un nuevo usuario, sale que se ha iniciado sesión con éxito:



Has iniciado sesión
correctamente

Paselo bien

OK

Si se pulsase el botón, nos llevará al menú principal donde estaremos con la sesión iniciada.

Aplicando esta librería:

```
import Swal from 'sweetalert2';
```

```
const validateUser = () => {
    dispatch(authenticateUser(user.usernameOrEmail, user.password))
        .then((response) => {
            console.log();
            setUsuario(parseJwt(localStorage.getItem("jwtToken")).sub);
            Swal.fire(
                'Has iniciado sesión correctamente',
                'Paselo bien',
                'success'
            ).then((result) => {
                navigate("/");
            });
        });
    setUser(initialState);
}
.catch((error) => {
    console.log(error.message);
    setShow(true);
    resetLoginForm();
    Swal.fire(
        'Las credenciales son incorrectas',
        'Intentalo de nuevo',
        'error'
    );
});
```

Manual Administrador

Los requisitos para instalar esta aplicación son:

Programa	Versión
Java	11
Git	2.35.1
Spring Tools 4 For Eclipse	4.14.0
Visual Studio Code	1.65
XAMPP	8.1.2

Pasos a seguir

- Importa la base de datos a tu servidor MYSQL
- Modifica el datasource en application.properties para que coincida
- Se debe encender la aplicación de Spring abriendolo en el Spring Tools e iniciando en el menú de Spring Boot
- Después iremos a la carpeta de tcg-market, abrimos una consola y usamos el comando npm start
- El usuario admin de contraseña admin es el usuario con todos los permisos
- Y ya esta! La aplicación esta iniciada y lista para usar

Manual Usuario



Bienvenido a TCG-Market, Invitado

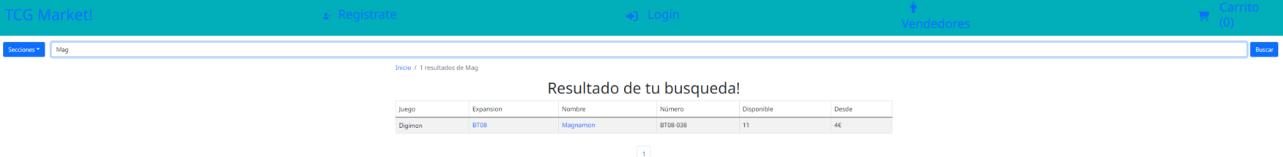
Este es un sitio web para comprar cartas tanto de Yu-Gi-Oh, Cardfight Vanguard y Digimon TCG.

Destacados

Juego	Expansion	Nombre	Número	Disponible	Desde
Vanguard	D-BT01	Vairina Valiente	D-BT01/001	8	4€
Digimon	BT08	Magnamon	BT08-038	11	4€
Vanguard	D-BT01	Diabolos Boys, Eden	D-BT01/004	13	3€
Digimon	BT08	Imperialdramon Fighter Mode	BT08-032	2	10€
Digimon	BT08	Davis Motomiya & Ken Ichijoji	BT08-088	4	6€
Vanguard	D-VS04	Revenget, Raging Fall Dragon Reverse	D-VS04/008	5	7€

El menú principal tiene varios apartados:

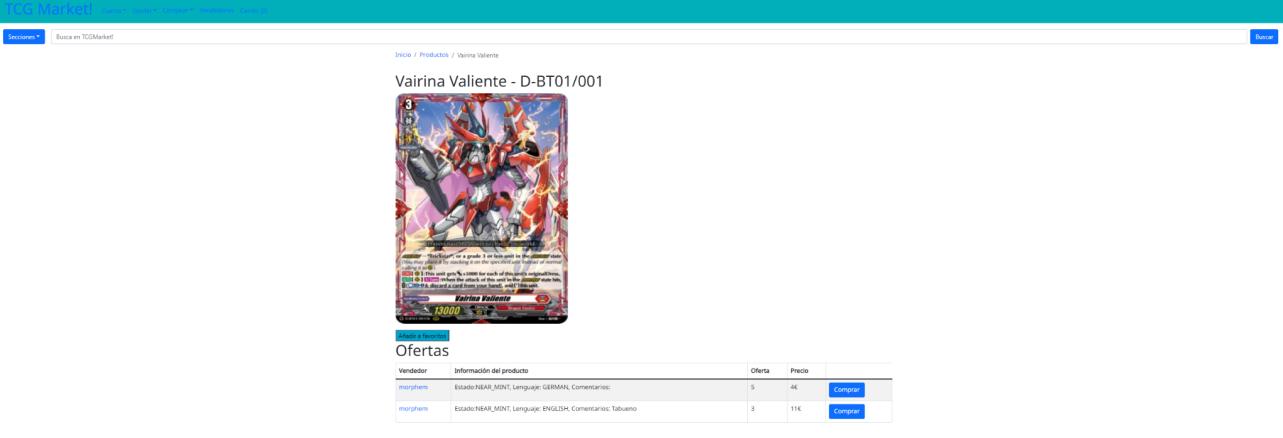
1. El header donde puedes ir a registrarte, a iniciar sesión, ver los vendedores y ver el carrito de tus productos a comprar
2. Puedes elegir secciones que te llevará al apartado que tu deseas entre: **Juegos, Expansiones y Cartas**
3. La barra de búsqueda buscará por nombre de cartas y mostrará una lista:



Resultado de tu búsqueda!

Juego	Expansion	Nombre	Número	Disponible	Desde
Digimon	BT08	Magnamon	BT08-038	11	4€

4. Entre los destacados, puedes acceder a la expansión de la carta y a la propia carta



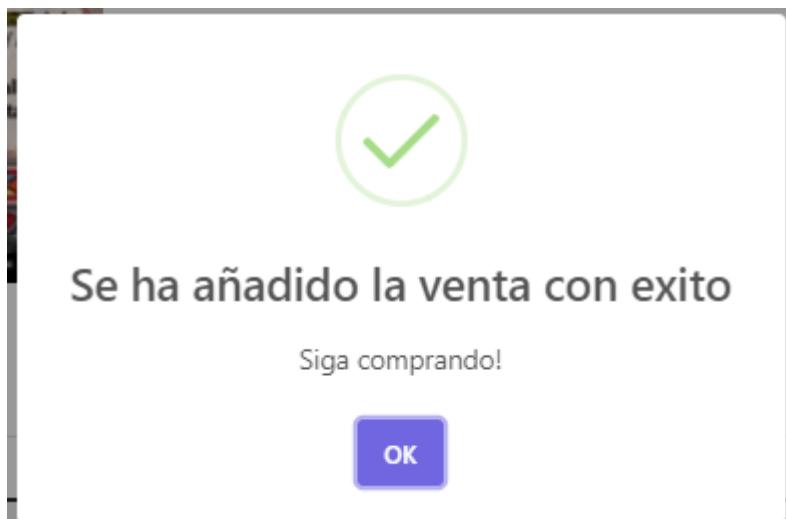
Vairina Valiente - D-BT01/001

Ofertas

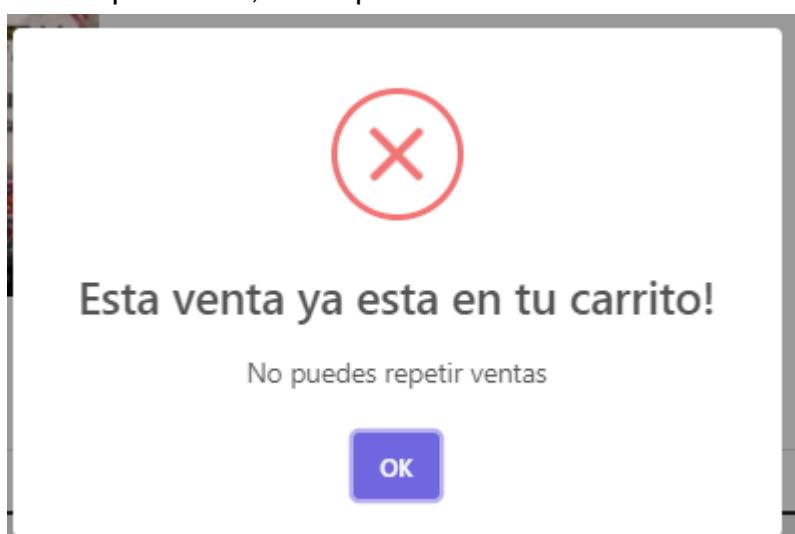
Vendedor	Información del producto	Oferta	Precio
morphem	Estado:NEAR_MINT, Lenguaje: GERMAN, Comentarios:	5	4€
morphem	Estado:NEAR_MINT, Lenguaje: ENGLISH, Comentarios: Tabueno	3	11€

Proyecto Desarrollo de Aplicaciones Web

Donde podrás añadir artículos a la cesta con el botón de comprar y se te confirmará que ha sido introducido en la cesta:



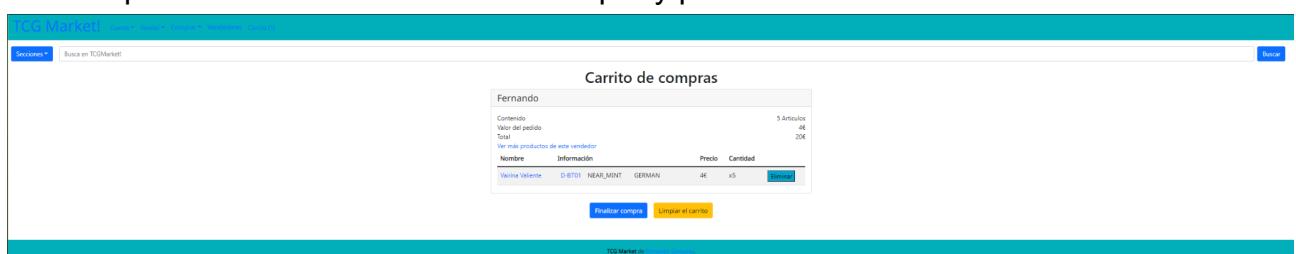
Si se repite venta, no se podrá introducir otra vez



Veremos que el ícono de carrito se modifica:



Al clicar podremos ver el carrito de la compra y poder borrar los items

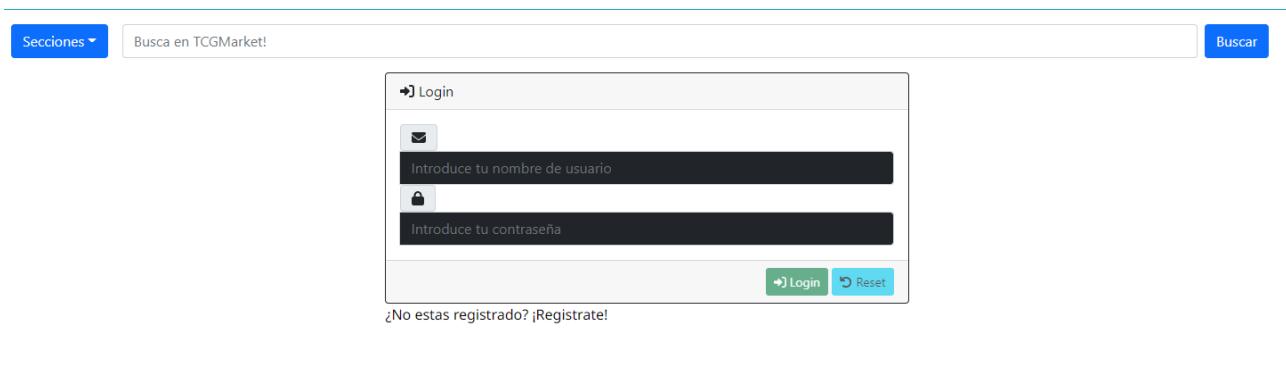



En el apartado de registro, te podrás registrar y te redirigirá a la página de login



The screenshot shows the registration page of the TCG Market website. At the top, there are navigation links for 'Registrate' (Register), 'Login', 'Vendedores' (Sellers), and 'Carrito 0' (Cart). Below the header, there is a search bar and a dropdown menu for 'Secciones'. The main content area is titled 'Formulario de registro' (Registration Form) and contains fields for Name (Nombre and Apellidos), Date of Birth (Fecha de nacimiento), Email (Dirección de correo), Username and Password (Nombre de usuario y contraseña).

Donde introduciendo tu usuario y tu email, podrás iniciar sesión



The screenshot shows the login page of the TCG Market website. At the top, there is a search bar and a dropdown menu for 'Secciones'. The main content area is titled 'Login' and contains fields for 'Introduce tu nombre de usuario' (Enter your username) and 'Introduce tu contraseña' (Enter your password). Below the form, there is a link '¿No estas registrado? ¡Register!' (Not registered? Register!).

Proyecto Desarrollo de Aplicaciones Web

Una vez iniciada la sesión, veremos un nuevo encabezado con varias opciones:

- Cuenta, para ver tus datos personales
 - Cuenta
 - Favoritos
 - Cerrar sesión
- Ventas
 - Ver tus ventas
 - Crear una venta
- Ver los vendedores en general
- El carrito de la compra

TCG Market! | Iniciar sesión | Comprar | Vender | Ayuda

Secciones ▾ Busca en TCGMarket! Buscar



Bienvenido a TCG-Market,
Destacados

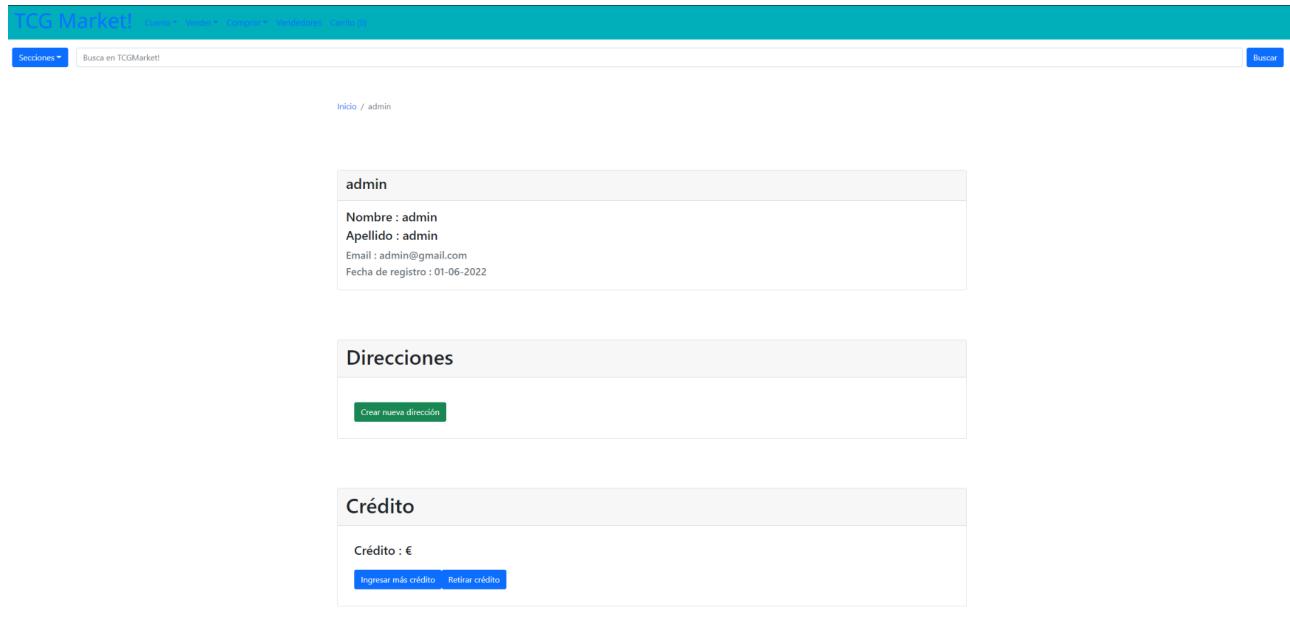
Juego	Expansion	Nombre	Número	Disponible	Desde
Vanguard	D-BT01	Vairina Valiente	D-BT01/001	8	4€
Digimon	BT08	Magnamon	BT08-038	11	4€
Vanguard	D-BT01	Diabolos Boys, Eden	D-BT01/004	13	3€
Digimon	BT08	Imperialdramon Fighter Mode	BT08-032	2	10€
Digimon	BT08	Davis Motomiya & Ken Ichijoji	BT08-088	4	6€
Vanguard	D-VS04	Reenger: Raging Fall Dragon Reverse	D-VS04/008	5	7€

TCG Market de | [Ayuda](#) | [Contacto](#)

Proyecto Desarrollo de Aplicaciones Web

Cuando entramos en cuenta, vemos varios apartados:

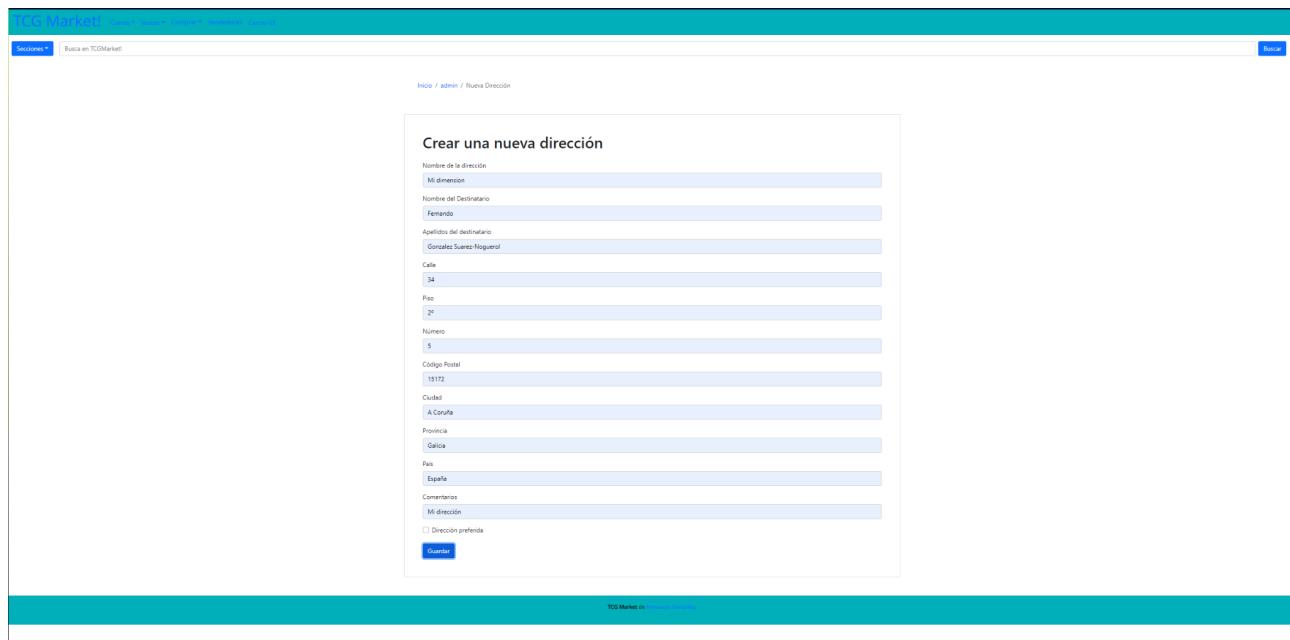
- Información del usuario
- Ver tus direcciones y podremos crear nuevas direcciones



The screenshot displays three main sections of the TCG Market application:

- User Profile:** Shows a user named "admin" with details: Nombre : admin, Apellido : admin, Email : admin@gmail.com, Fecha de registro : 01-06-2022.
- Direcciones (Addresses):** A section titled "Direcciones" with a button labeled "Crear nueva dirección" (Create new address).
- Crédito (Credit):** A section titled "Crédito" showing a balance of € and buttons for "Ingresar más crédito" (Add more credit) and "Retirar crédito" (Withdraw credit).

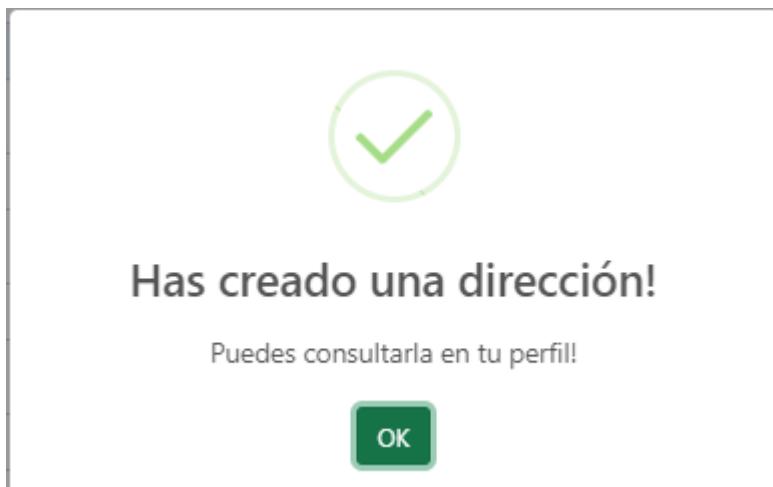
Al clicar en el botón de crear nuevas direcciones, se abrirá un formulario para crear Direcciones:



The screenshot shows a detailed "Crear una nueva dirección" (Create new address) form:

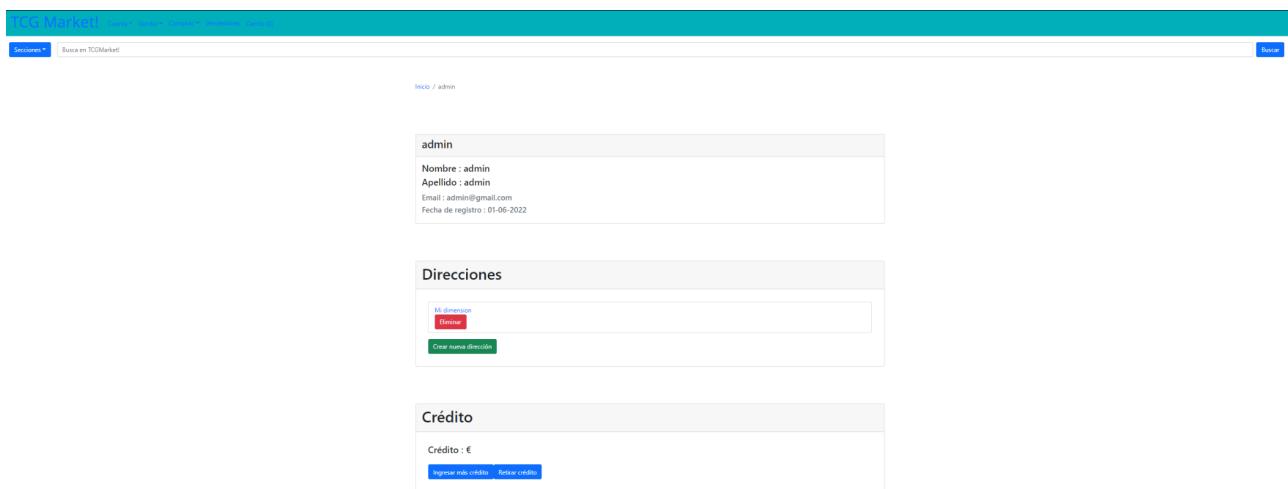
- Nombre de la dirección: Mi dirección
- Nombre del Destinatario: Fernando
- Apellidos del destinatario: González Suárez-Noguero
- Calle: 34
- Piso: 2º
- Número: 5
- Código Postal: 15172
- Ciudad: A Coruña
- Provincia: Galicia
- País: España
- Comentarios: Mi dirección
- Dirección preferida
- Guadar** (Save) button

Se creará una dirección y podremos verla en nuestro perfil

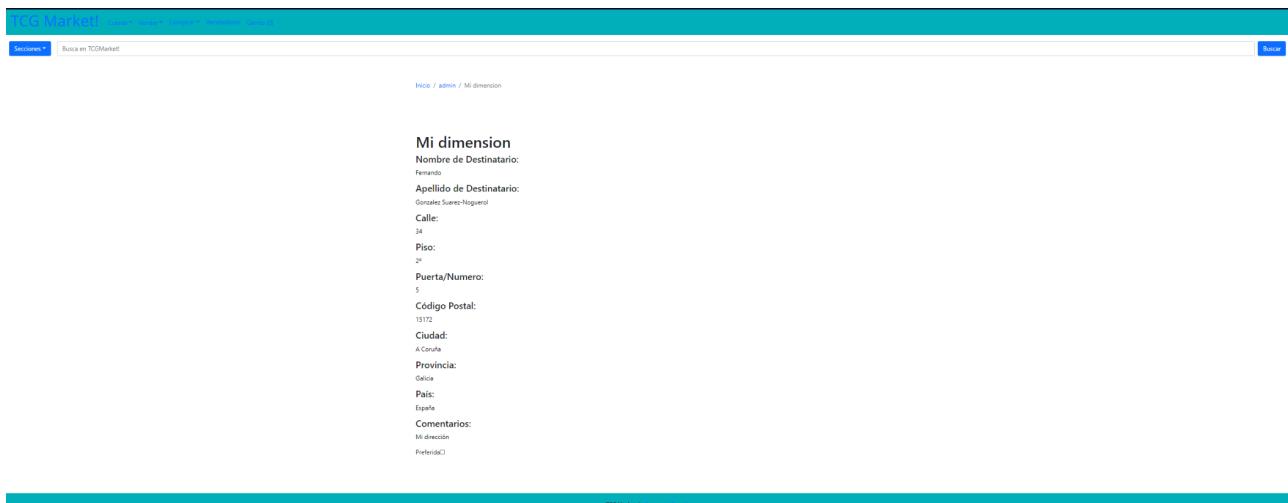


Te mostrará una lista de tus direcciones y una opción a borrar. Te preguntará si estás segur@ y aceptas o cancelas.

Se puede clicar en el nombre de la dirección para verla.



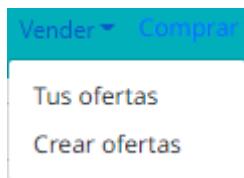
A screenshot of the TCGMarket admin panel. The top navigation bar shows "TCGMarket" and "Administrador". The main content area has three sections: "admin" (user info), "Direcciones" (address list), and "Crédito" (credit). The "admin" section shows a user named "admin" with details: Nombre : admin, Apellido : admin, Email : admin@gmail.com, Fecha de registro : 01-06-2022. The "Direcciones" section shows a table with one row labeled "Mi dimensión" with a "Borrar" button. The "Crédito" section shows a balance of "€" with buttons to "Ingresar más crédito" and "Retirar crédito".



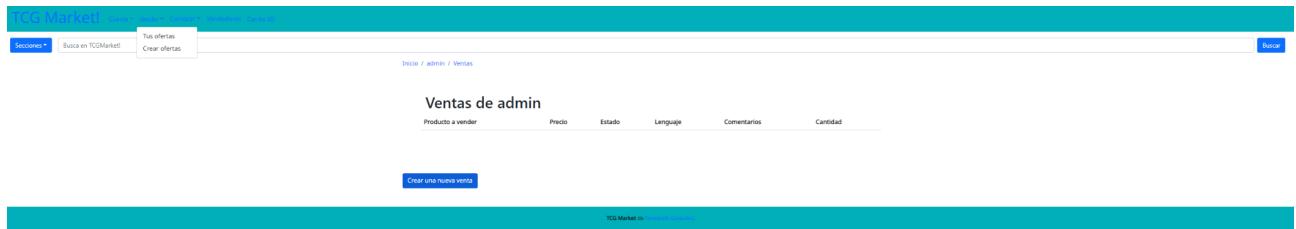
A screenshot of the TCGMarket admin panel showing the "Mi dimensión" address creation form. The form fields include: Nombre de Destinatario: Fernando, Apellido de Destinatario: González Suárez-Noguera, Calle: 34, Piso: 2º, Puerta/Número: 5, Código Postal: 19172, Ciudad: A Coruña, Provincia: Galicia, País: España, Comentarios: Mi dirección Preferida(C), and a "Enviar" button at the bottom.

Proyecto Desarrollo de Aplicaciones Web

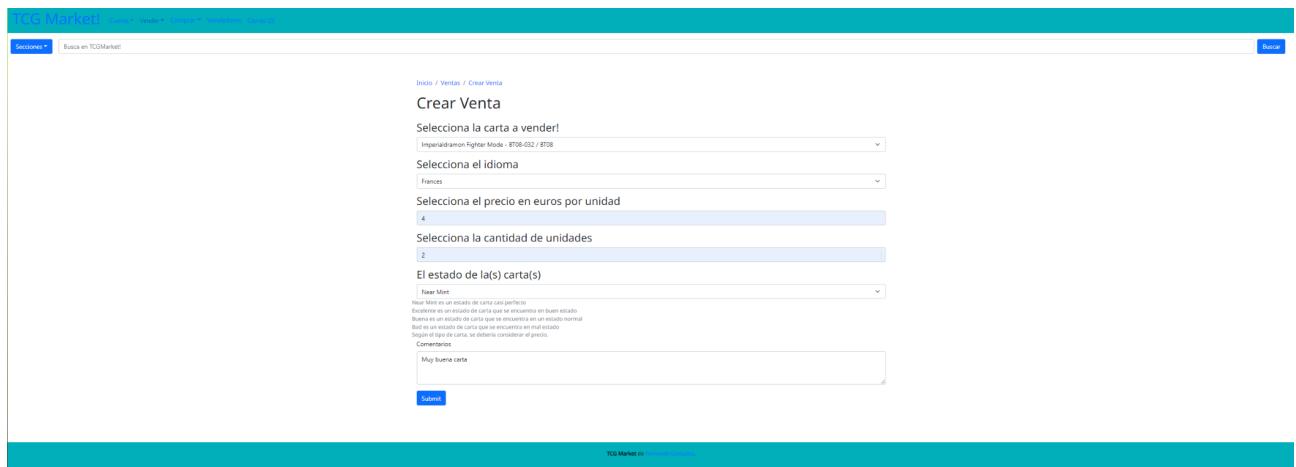
Cuando entramos en el apartado Tus ventas, vemos dos opciones:



- Ver tus ventas, que muestra un listado de las ventas y un botón para crear ventas

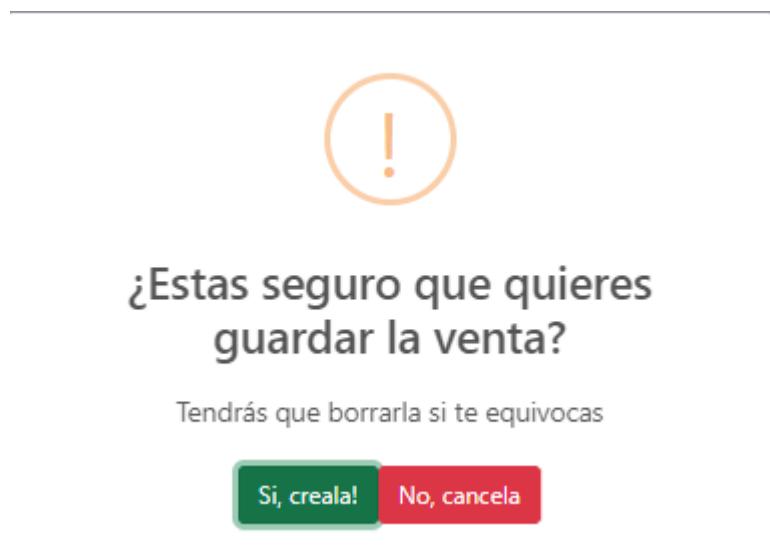


- Crear ventas, se puede acceder tanto desde el listado de opciones como del botón y mostrará un formulario a llenar



Proyecto Desarrollo de Aplicaciones Web

- Cuando rellenamos el listado y confirmamos la creación, nos llevará a las ventas de nuevo



Ventas de admin

Producto a vender	Precio	Estado	Lenguaje	Comentarios	Cantidad
Imperialdramon Fighter Mode	3	NEAR_MINT	SPANISH	Muy buena carta	2

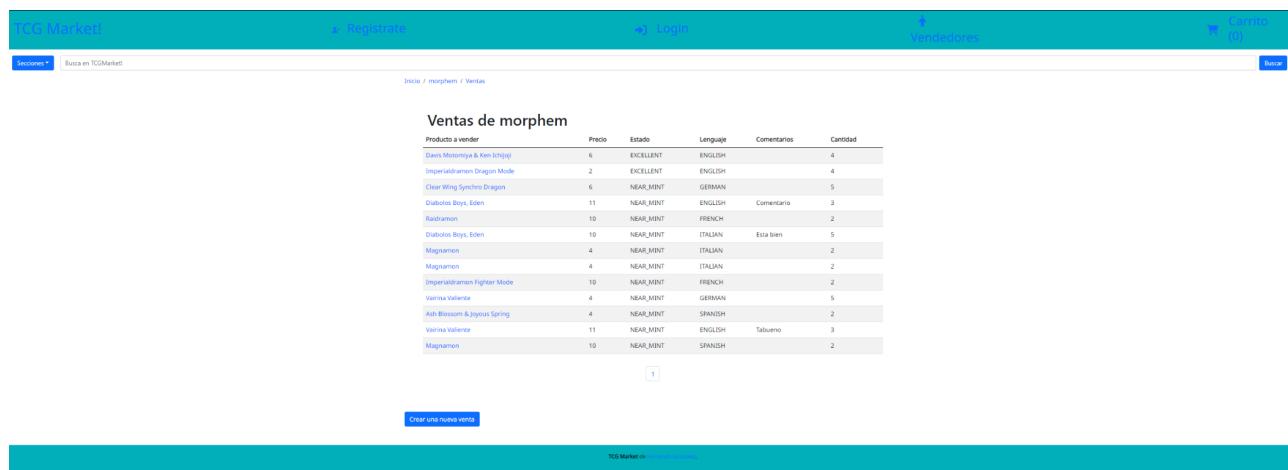
1

Para terminar, veremos la lista de los vendedores:

Nombre de usuario	Fecha de registro	Ventas listadas
monica	09-09-2019	3
morphem	21-05-2022	13
Monica3	01-06-2022	4
admin	01-06-2022	1

Podremos acceder al perfil del vendedor y ver sus ventas

Proyecto Desarrollo de Aplicaciones Web



The screenshot shows a web application interface for 'TCG Market'. At the top, there are navigation links: 'TCG Market', 'Regístrate', 'Login', 'Ventas', 'Centro de ayuda', and 'Búscar'. Below the header, a search bar says 'Busca en TCGMarket!'. The main content area is titled 'Ventas de morphem' and displays a table of products for sale. The columns are 'Producto a vender', 'Precio', 'Estado', 'Lenguaje', 'Comentarios', and 'Cantidad'. The table lists various items like 'Diablos Boys, Eden', 'Diablos Boys, Eden', 'Magnamen', etc., with details such as price (e.g., 6, 2, 11, 10, 4, 4, 10, 4, 4, 4), state (e.g., EXCELLENT, NEAR_MINT), language (e.g., ENGLISH, GERMAN, FRENCH, ITALIAN, SPANISH), comments (e.g., 'Está bien', 'Tabuemo'), and quantity (e.g., 4, 4, 5, 3, 2, 2, 2, 2, 5, 3, 2). A small button 'Crear una nueva venta' is visible at the bottom left of the table.

Viabilidad tecno-económica.

El proyecto para desarrollarse a plan futuro requiere:

- Hosting de aplicaciones java y aplicaciones de JavaScript. Se han barajado opciones y se pagaría
 - El plan de [Hosting JVM Smart para java](#) costaría 149,90€ al año
 - El plan de [Hosting Web](#) costaría 149,90€ al año
- Pagar el desarrollo de la aplicación de React JS y la de java
 - Costaría en un rango de 3.000€ hasta 6.000€ para terminar el desarrollo
- Pagar el mantenimiento de la propia aplicación web y el servidor
 - Costaría entre 1.000€ hasta 3.000€ anuales
- Conseguir que Paypal y/o otras aplicaciones de cartera virtual financie la aplicación y conseguir que su API se implemente perfectamente.

Sería costoso y habría que hacer un estudio de viabilidad económica para ver si el proyecto empezaría con buen pie.

En total, una inversión inicial de 10.000€ y los costes de mantenimiento, sobre 3.300€.

Trabajo futuro.

1. Preparar la base de datos correctamente para que se puedan borrar entidades de forma segura. No se puede eliminar ni direcciones ni ventas porque las relaciones externas lo impiden
2. Preparar más casuísticas: Búsqueda avanzada, filtros, editar objetos y validaciones de los formularios
3. Prepara los endpoints acorde a las casuísticas planteadas.
4. Terminar el checkout porque sin el checkout, este proyecto no es un producto mínimo viable dado que no cumple su finalidad
5. Limpiar el código extensivamente
6. Refinar el diseño: Mejorar la paleta de colores, revisar usabilidad, legibilidad y accesibilidad

Conclusiones.

Para concluir el proyecto, he de decir que el tiempo se me vino encima. Me confié demasiado y no he preparado un proyecto a la altura.

Mientras que el proyecto es muy sencillo, funciona todo lo que he revisado bien y mientras que no es perfecto, cumplirá su función

Confío en mis conocimientos para hacer una presentación detallada y poder hacer una presentación al nivel que no está el proyecto.

Mi conclusión final es que el proyecto es un mero trámite, interesante de hacer pero algo que ocupa un montón de tiempo y las prácticas al mismo tiempo supone una carga realmente pesada para demostrar algo que ya he demostrado con creces.

Aún así, ha sido emocionante desarrollar la aplicación por mi cuenta. Los logros eran triunfos que me alegraban la noche y los fallos, lecciones de las que aprender. Este proyecto va a ser una lección de la que aprender una lección que creía tener aprendida.

Biblioteca de recursos web y referencias.

(2021-2022): Imagenes de Digimon TCG, Bandai, <https://world.digimoncard.com>

(2018-2022): Imagenes de Cardfight Vanguard TCG, Bushiroad,
<https://decklog-en.bushiroad.com>

(2015-2022): Imagenes de Yugioh TCG, Konami, <https://www.yugioh-card.com/es/>