Pixel Portal



Juan Carlos Millán Diez de Ulzurrun

Ángel Olivares Duque

Izar Rodríguez Araujo

**Tutora:** Eva María Alonso Fernández

2º DAW



Índice

[1.1. Análisis de la utilidad y/o necesidad del sistema. Proyección a nivel empresarial.](#_ebx3k7morm2k) 3

[1.2. Identificación del alcance del sistema](#_s0fijxvo51gk) 4

[**2. ANÁLISIS DEL SISTEMA**](#_pl26s36uulhi) **8**

[2.1. Especificación de los requisitos](#_ejvll1kxswc7) 8

[2.2. Especificación de estándares y normas de aplicación en el proyecto](#_5m6mob703xd) 10

[**3. DISEÑO DEL SISTEMA**](#_8e06kzh6848v) **12**

[3.1. Diseño de interfaces de usuario](#_wwe6wm3d70h6) 12

[3.1.1 Paleta de colores](#_wy3qk1qkuxf) 12

[3.1.2 Bocetos](#_l7krtot9ljw5) 13

[3.1.3 Diseño final](#_s5hw93h2kf0r) 16

[3.1.4 Fuentes](#_jwfz74g22tmn) 24

[3.2. Diseño del modelo de datos](#_eqo7i0gvu23g) 25

[Users](#_z8tr0q8ig9uf) 26

[Developers](#_czmmitsb2bev) 26

[Publishers](#_adk45xj2jtjk) 26

[Games](#_thwhw9it5tfl) 26

[Carts](#_8ujcvwolfleg) 27

[Cart\_items](#_ivg0ew4pczar) 27

[Inventory\_users](#_nat2s8eoswby) 27

[Profile\_comments](#_sc2j4arbb3dc) 28

[Friends](#_l0quin6zv4ca) 28

[Reviews](#_mil55ia0de2s) 28

[Game\_genres](#_zg55fgdo1flo) 28

[Game\_genres\_mapping](#_afh5vkdedws) 29

[Posts](#_26l25ero6u7c) 29

[**4. Aspectos relevantes de la implementación**](#_zb4nnsl4qgkb) **29**

[5. Pruebas de accesibilidad y usabilidad 30](#_c04r3js7gqk7)

[5.1 Validación y alta accesibilidad](#_c04r3js7gqk7) 30

[5.2 Test de usuario 30  
  
Objetivo](#_b6kqztedissm) 30

[Duración](#_mwjfg3wpg0s5) 30

[Materiales Necesarios](#_wkg0qkt5chho) 30

[Participante](#_8d7p39ouuckk) 31

[Respuestas del Test de Usuario](#_e053lksuhb0m) 31

[1. Navegación Inicial](#_fb3k65bp3ucd) 31

[2. Exploración de Juegos](#_4p2c89kg6pgh) 31

[3. Carrito de Compras](#_eyuqxk2t3d19) 33

[4. Proceso de Pago](#_ae8chglt8foo) 33

[5. Registro e Inicio de Sesión](#_aqet2wtavhas) 34

[6. Biblioteca de Juegos](#_tmgoq8fuuvzl) 35

[Cuestionario Final](#_sruy5s7ae0yq) 35

[Conclusión](#_v2gvt03nmf1q) 36

[5.3 Test A/B 36](#_3j05xslkk7pe)

[Objetivos](#_u24w6cn9hlq3) 36

[Tasa de Éxito en tareas realizadas](#_4ujrtgkd69xo) 37

[Nivel de Satisfacción (Escala de 1 a 5)](#_r5k51v12oorb) 37

[Conclusión](#_w69e1b6anr15) 38

[**6. Videojuego-demo**](#_k8ioiedlumqt) **38**

[**7. Problemas encontrados**](#_9kh74qeqo6os) **42**

[**8. Conclusiones**](#_ihzy872vvrf2) **43**

[**9. Bibliografía**](#_t1qg6ts9i980) **45**

1. DEFINICIÓN DEL SISTEMA

## 1.1. Análisis de la utilidad y/o necesidad del sistema. Proyección a nivel empresarial.

El presente proyecto se ofrece como una propuesta enmarcada en un contexto *e-commerce* *& social media*. Se trata de un sitio web que posibilita la compra de videojuegos, haciendo de intermediario entre desarrolladores, distribuidores y usuario final o cliente.

Sitio en el que además, todo usuario que así lo desee, puede hacer uso de las herramientas de dinamización social previstas en el mismo: sistema de amigos, búsqueda de usuarios, comentarios en su perfil, etc.

Así, en Pixel Portal aunamos la adquisición de títulos tanto independientes como de aclamado éxito dentro del sector de los videojuegos con una experiencia social en línea a través de la cual no sólo disfrutar de un tiempo para tí, sino también con el resto de usuarios de la plataforma. La decisión depende enteramente del usuario.

El sector del videojuego se encuentra en la cima en lo que a consumo, popularidad y proyección se refiere. Empresas y negocios de todo tipo se asocian con equipos de competición *e-Sport* o patrocinan campañas publicitarias aludiendo a un público cada vez no tan joven con objeto de darse a conocer como joviales, dinámicas, modernas. Por eso creemos firmemente en apostar por un proyecto de estas características, siendo una fórmula probadamente rentable.

Es importante añadir también que las posibilidades de expansión y proyección tanto de las funcionalidades y puntos de interés del proyecto son notables, pudiendo optar por tecnificar más el sitio, profundizar en el apartado social, especializarse en un público en concreto, posicionarse en el ofrecimiento de más servicios más allá de videojuegos, etc.

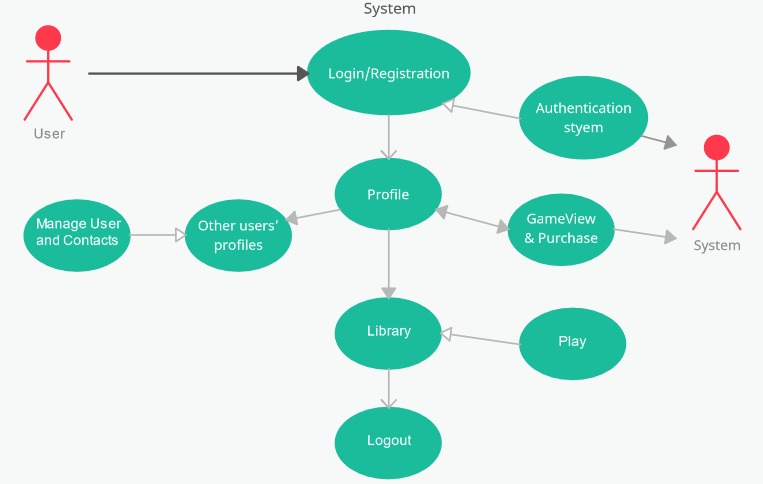
A modo de ejemplo del funcionamiento de la plataforma, **se ha desarrollado la versión demo de un videojuego**, a modo ilustrativo de cómo es la experiencia que hemos diseñado y creemos posee las cualidades para llegar a ser un proyecto viable y lucrativo.

En definitiva, es un servicio ya no sólo útil sino necesario dada la creciente demanda tanto de videojuegos como de interacción social de aquellos que los adquieren. Máxime cuando se trata de un sector en constante crecimiento.

## 1.2. Identificación del alcance del sistema

El principal objetivo del proyecto en sí ha sido la muestra del proyecto anteriormente descrito en un estado de prestaciones limitadas buscado y deseado a modo de prueba para su presentación inicial y *testing* general, a la par que básico.

Los requisitos más inmediatos de cara a una prueba preliminar son un equipo informático completo, ya sea de sobremesa o portátil, si bien se ha tenido en cuenta la hegemonía del dispositivo móvil en el desarrollo. Caso de uso típico:

1.3. Identificación del entorno tecnológico y de las tecnologías a utilizar

El proyecto implica, por necesidad, una combinación de tecnologías y herramientas típicamente asociadas con lo que ha dado en llamarse *frontend* y *backend*. También se ha empleado una base de datos creada *ex profeso* para su comunicación con el *backend* y de necesaria implementación para el funcionamiento del proyecto tal y como fue concebido y ha sido desarrollado.

· *Backend* -> Mientras que el lenguaje empleado ha sido **Java**, el contenido dinámico del servidor ha corrido a cargo de **JavaServer Pages (JSP)**, además de **Java Servlets** para manejar solicitudes/respuestas **HTTP**. El servidor utilizado, **Apache Tomcat,** contiene los servlets. Las

· *Frontend* -> para la estructura de las páginas web se ha usado **HTML5**, **CSS** y **Bootstrap** para los estilos y disposición de los elementos, y **JavaScript** para la manipulación del árbol del DOM y crear dinamizaciones interactivas y reactivas respecto de las indicaciones del usuario.

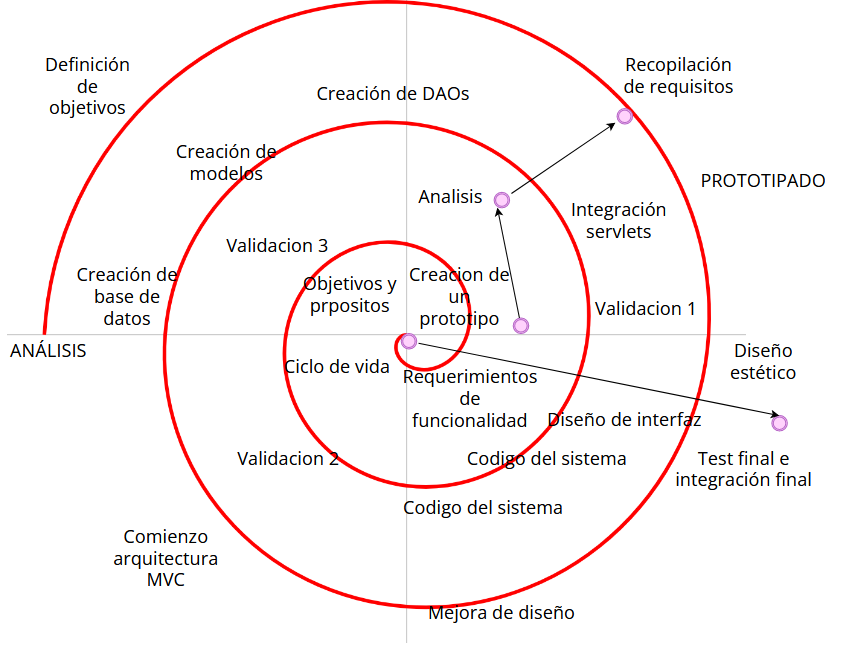
· Base de datos: el lenguaje empleado ha sido **SQL** integrado en **MySQL Workbench.**

Como anexo, **Maven** se ha escogido para la gestión de dependencias y la automatización del proceso de construcción del proyecto en así, además de **Git** para el control de versiones.

Con respecto al entorno de desarrollo, los **IDEs** usados han sido **Visual Studio Code** y **Eclipse for Java EE Developers (versión a fecha de marzo de 2024)**.

1.4. Planificación de las fases del desarrollo del proyecto

· Metodología empleada: se ha llevado a cabo un acercamiento multidisciplinar y polifacético, combinando una metodología **en espiral** con la denominada **lean** (enfoque en el proyecto final como objetivo inamovible).



En el diagrama adjunto, se detalla el proceso recursivo que, a vista general de durante el proceso de desarrollo, se ha efectuado. Es de destacar el manejo de **servlets** allá donde se han requerido para la compartimentalización y separación en módulos independientes todo lo posible de las funcionalidades de la aplicación.

Es de obligada mención el haber recurrido al manido y denostado **cowboy coding** en más situaciones de las que nos hubiera gustado, siendo algo que juzgamos tan honesto mencionar como asegurarnos a nosotros mismos que limitaremos en futuros desarrollos.

Con respecto a las fases de desarrollo y su distribución temporal, *grosso modo*:

* **Planificación**: 1ª y 2ª semana de abril.
* **Análisis**: 3ª semana y ½ de la 4ª semana de abril.
* **Diseño**: a lo largo de todo el desarrollo del proyecto.
* **Implementación**: 2ª, 3ª, 4ª semanas del mes de mayo.
* **Prueba**: 1ª y 2ª semana del mes de junio.

Así, los lenguajes, frameworks y herramientas generales utilizadas han sido:

· **Lenguajes:** Java, JavaScript, HTML, CSS, directivas JSP.

· **Frameworks y librerías:** JQuery, Bootstrap, Bootstrap Icons ( ambos vía CDN).

· **Tecnologías y herramientas de desarrollo:** Apache Tomcat, Maven, Git, Eclipse for Java EE Developers, Visual Studio Code, PostgreSQL/MySQL, JBDC (Java Database Connectivity).

# 2. ANÁLISIS DEL SISTEMA

## 2.1. Especificación de los requisitos

En cuanto a los requisitos para un correcto **desarrollo**, **migrado**, **puesta en marcha**, y **prueba** o **testeo** del proyecto:

· **Requisitos funcionales**: registro y autenticación del/a usuario/a, gestión de los ítems videojuego (visualización en detalle, buscado, filtrado (por género y/o precio) y listado de los juegos además de empresas desarrolladoras y distribuidoras de los mismos; adquisición, despliegue del videojuego desarrollado a modo de demo. También la posibilidad de establecer valoraciones de los títulos adquiridos, así como un sistema de *rating* para manejar su puntuación sobre cinco (5). Posibilidad de edición del perfil con información sobre uno/a mismo/a y establecer una imagen como avatar en la plataforma. Posibilidad de búsqueda de usuarios, listado y añadir como amigos (pudiendo aceptar, denegar y borrar al usuario de amigos), así como escribir comentarios en el perfil de dicho usuario (queda a discreción de dicho usuario borrarlo cuando desee). Queda previsto el contacto con la administración de la plataforma para cualquier duda que pueda surgir sobre el uso o mal funcionamiento de la misma llegado el caso, así como un sistema de recuperación/cambio de contraseña. Todo ello enmarcado en un contexto visualmente atractivo y funcionalmente responsivo para todo tipo de dispositivos.

· **Requisitos no funcionales**: entre ellos destacan la seguridad para evitar inserciones o ataques al sistema, también es requerida una **contraseña compleja** para un registro exitoso. Queda previsto la posibilidad de múltiples usuarios haciendo uso de la página al mismo tiempo. Con respecto a la usabilidad y compatibilidad además de la mantenibilidad del proyecto, se ha comprobado su correcta visualización y uso en diferentes navegadores y tamaños de pantalla para que la experiencia de uso no quede mermada en ningún caso siendo su acceso desde dispositivos ampliamente utilizados por el global de los potenciales consumidores.

## 2.2. Especificación de estándares y normas de aplicación en el proyecto

· **Estándares de código:** el proyecto muestra tanto globalmente como en su distribución interna una estricta aplicación de una arquitectura **Modelo-Vista-Controlador**, como puede verse en los sucesivos *servlets* empleados en él. Los nombres de las variables han sido escogidos para facilitar la lectura, así como una indentación y distribución en línea del mismo agradable a la lectura y al análisis. Es destacable además el uso de **objetos de sesión** para el manejo del usuario aunque la página se haya cerrado; también las restricciones necesarias en las etiquetas ***<input>***han sido codificadas siguiendo los estándares aplicables de la industria.

Ejemplo de uso de objeto de sesión en ProcessPaymentServlet.java


En un sentido más *frontend*, tanto el lenguaje **HTML** puro como aquellas clases de **CSS** y **Bootstrap** se ha mantenido el uso de etiquetas semánticas con objeto de mejorar tanto la accesibilidad como el posicionamiento **SEO**:

Con respecto al archivo **turbodank** correspondiente al juego-demo desarrollado, se ha incluido un archivo **README.txt** con objeto de facilitar una guía para su integración y uso.

Respecto a la aplicación de necesidad de la **arquitectura Modelo-Vista-Controlador,** la hemos aplicado rigurosamente: separando la lógica en tres componentes:

· **Modelo**: representa los datos presentes en la base de datos, siendo cada atributo de clase una columna de la tabla de datos.

· **Vista**: son los archivos JSP que presentan los datos de los modelos y la interacción del usuario con ellos; también están presentes las marcas HTML y CSS junto con el elemento dinamizador en cliente a través de JavaScript. Son las páginas de la web en sí.

· **Controlador**: la comunicación entre modelo y vista, los servlets, que manejan gran parte de la lógica pesada del *backend*  de la aplicación.

También hay archivos **DAO (Data Access Objects)** presentes, encapsulando la lógica del acceso a la base de datos, si ésta conecta eminentemente con los modelos. Así, se posibilitan acciones **CRUD (Create, Read, Update, Delete)** en la vista cuando el/la usuario/a hace uso de ella.

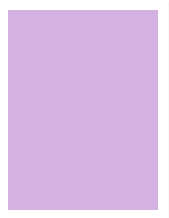
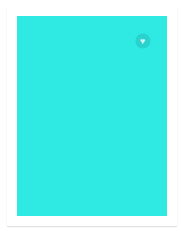
# 3. DISEÑO DEL SISTEMA

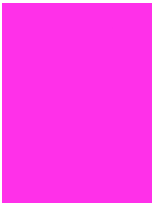
## 3.1. Diseño de interfaces de usuario

### 3.1.1 Paleta de colores Nos hemos decidido por una combinación cromática con objeto de crear una estética moderna y vibrante, atractiva para todo usuario y amante de los videojuegos, optando por una paleta clásica dentro “lo *gamer*” sin haber renunciado a aportaciones propias y originales, habiendo creado una atmósfera [*synthwave*](https://es.wikipedia.org/wiki/Synthwave)con referencias a los neones tan característicos de la estética de los años 80: una experiencia visual evocadora y dinámica. La incorporación de colores neutros como el negro y el blanco asegura la legibilidad y proporciona un equilibrio visual, mientras que los toques de rosa fucsia añaden un elemento de sorpresa y dinamismo.

### Hemos elegido una paleta de colores diseñada para ser accesible para personas con problemas de visión, como puede ser el daltonismo. Los colores seleccionados tienen un alto contraste y combinaciones que aseguran que todos los usuarios puedan distinguir fácilmente los diferentes elementos de la página; esto mejora la usabilidad y garantiza una experiencia inclusiva para todos. En conjunto, esta paleta está diseñada para ofrecer una experiencia de usuario atractiva, efectiva y accesible.

#2EEAE3 #66B3E8 #D4B3E2

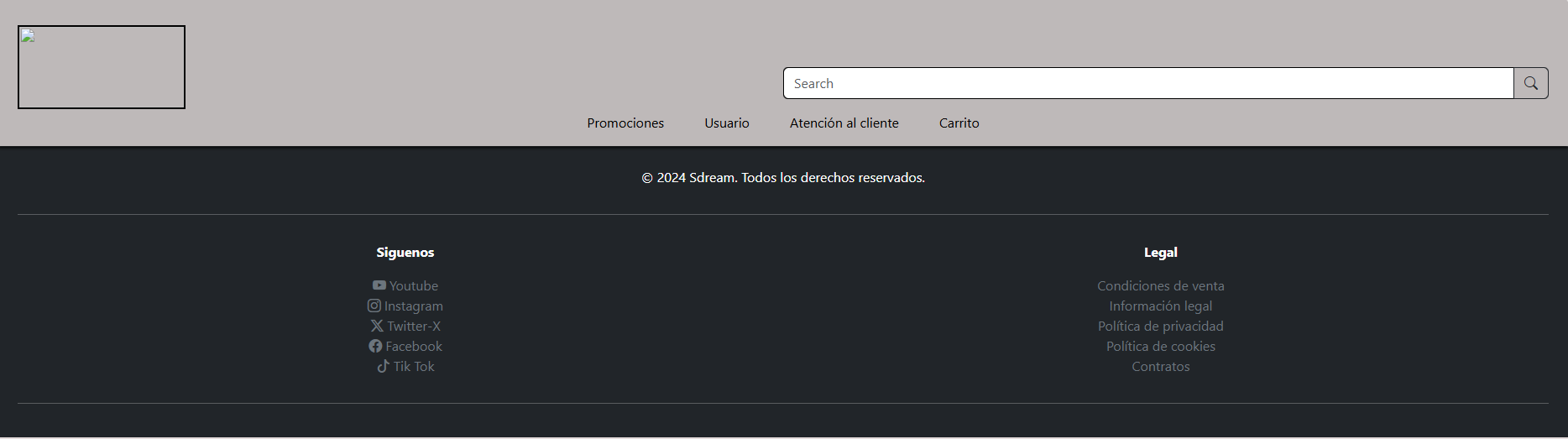


#000000 #FFFFFF #FF30E9  


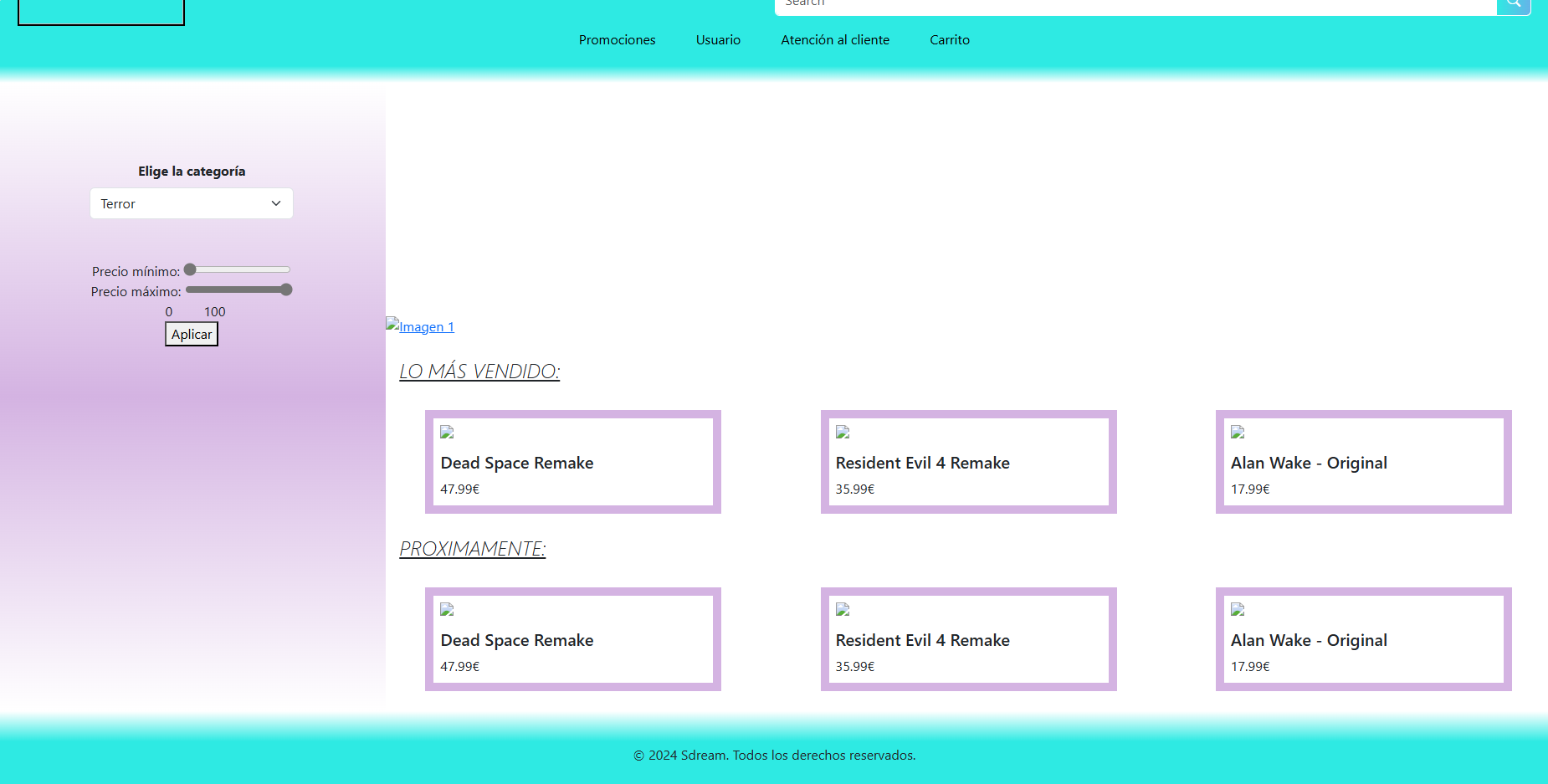
### 

### 3.1.2 Bocetos

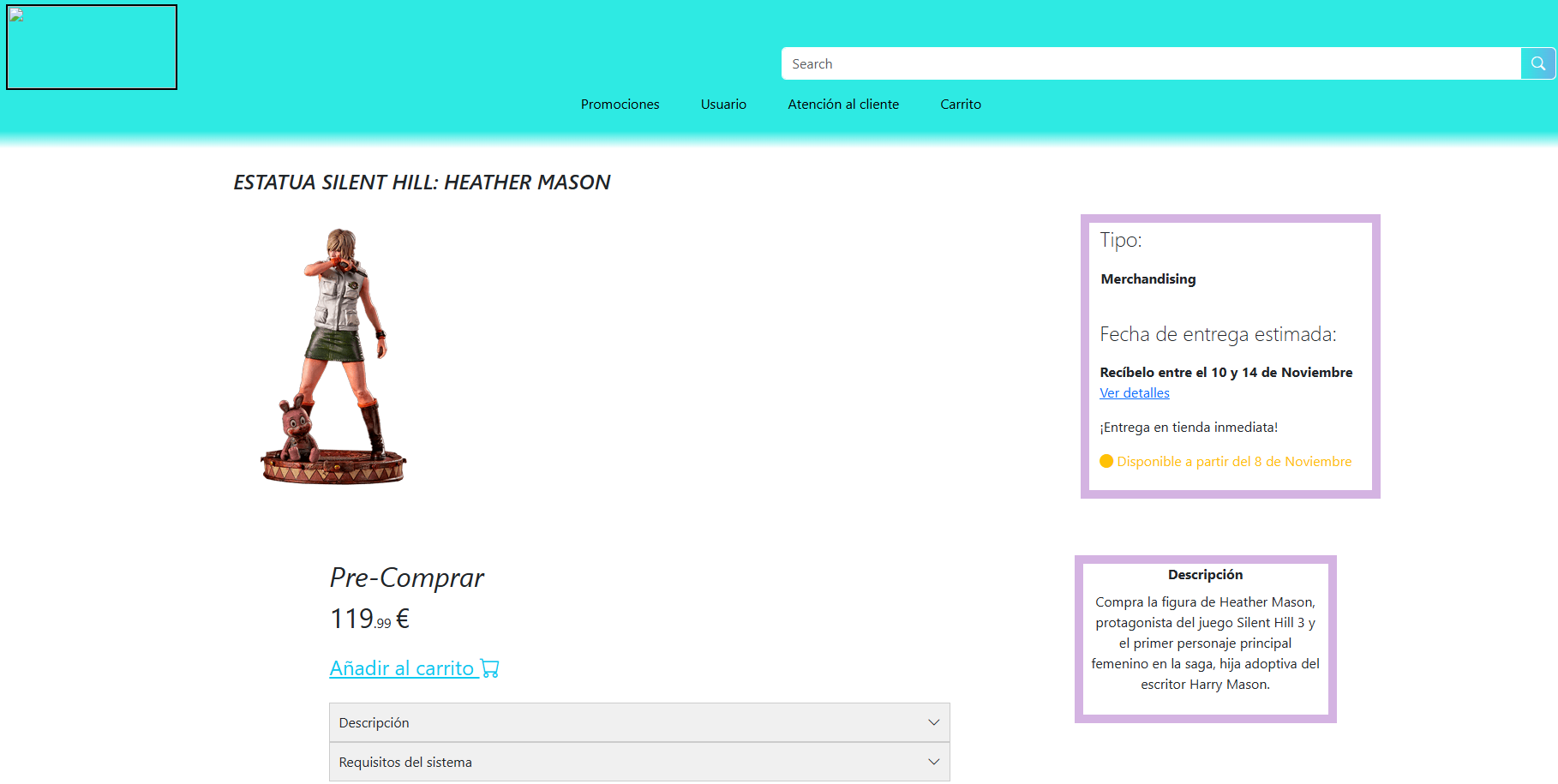
En esta sección se presentan una serie de capturas del prototipado inicial de la página para posteriormente compararlas con el resultado final.

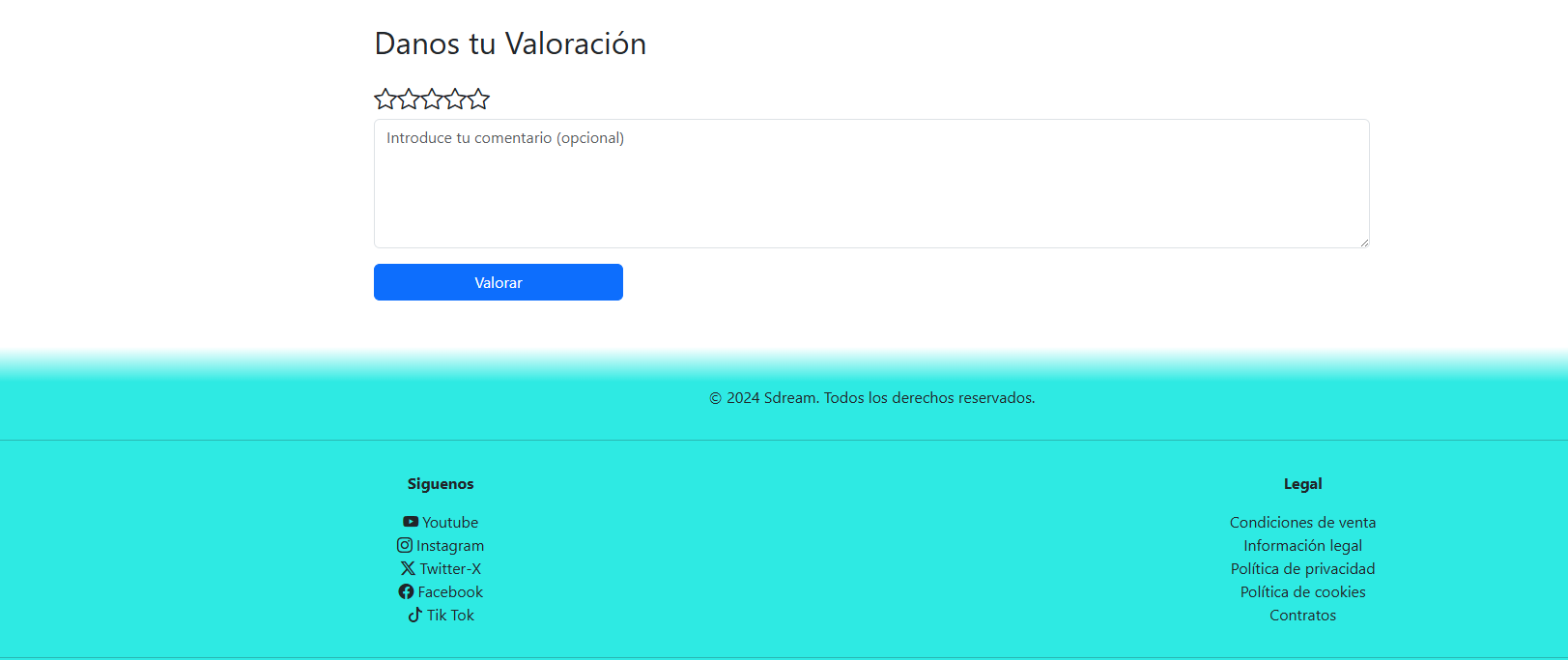
**Boceto de la plantilla inicial para la página:**  


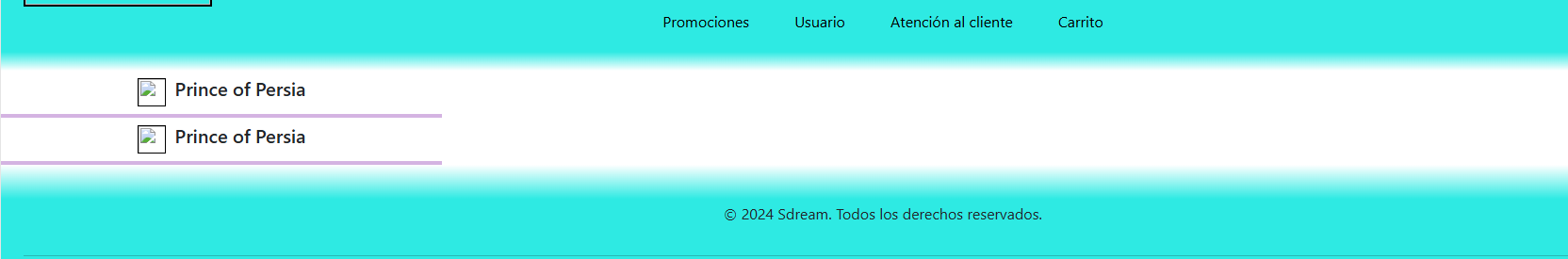
**Boceto del la página *index*:**

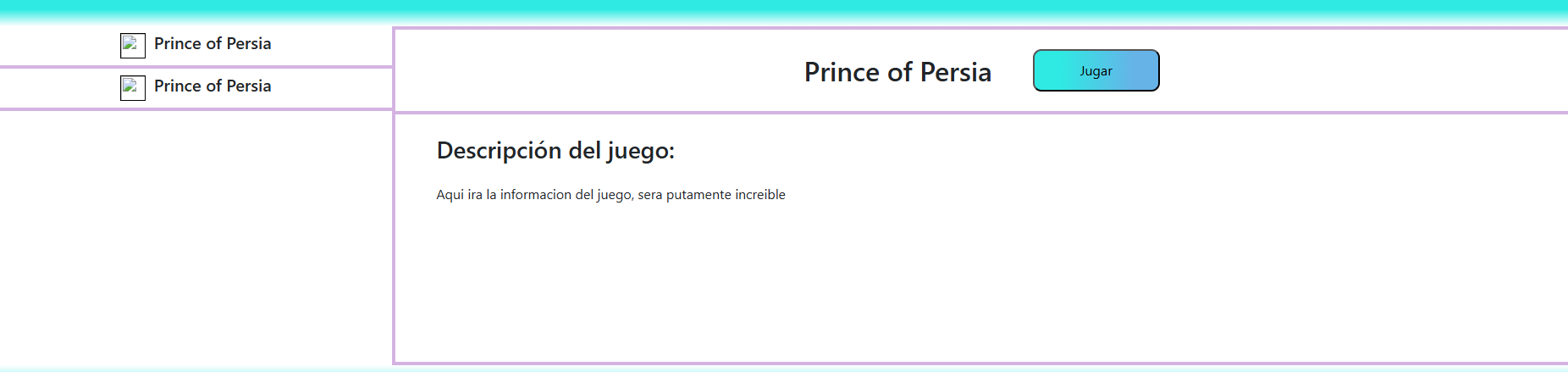


**Boceto del carrito:**   

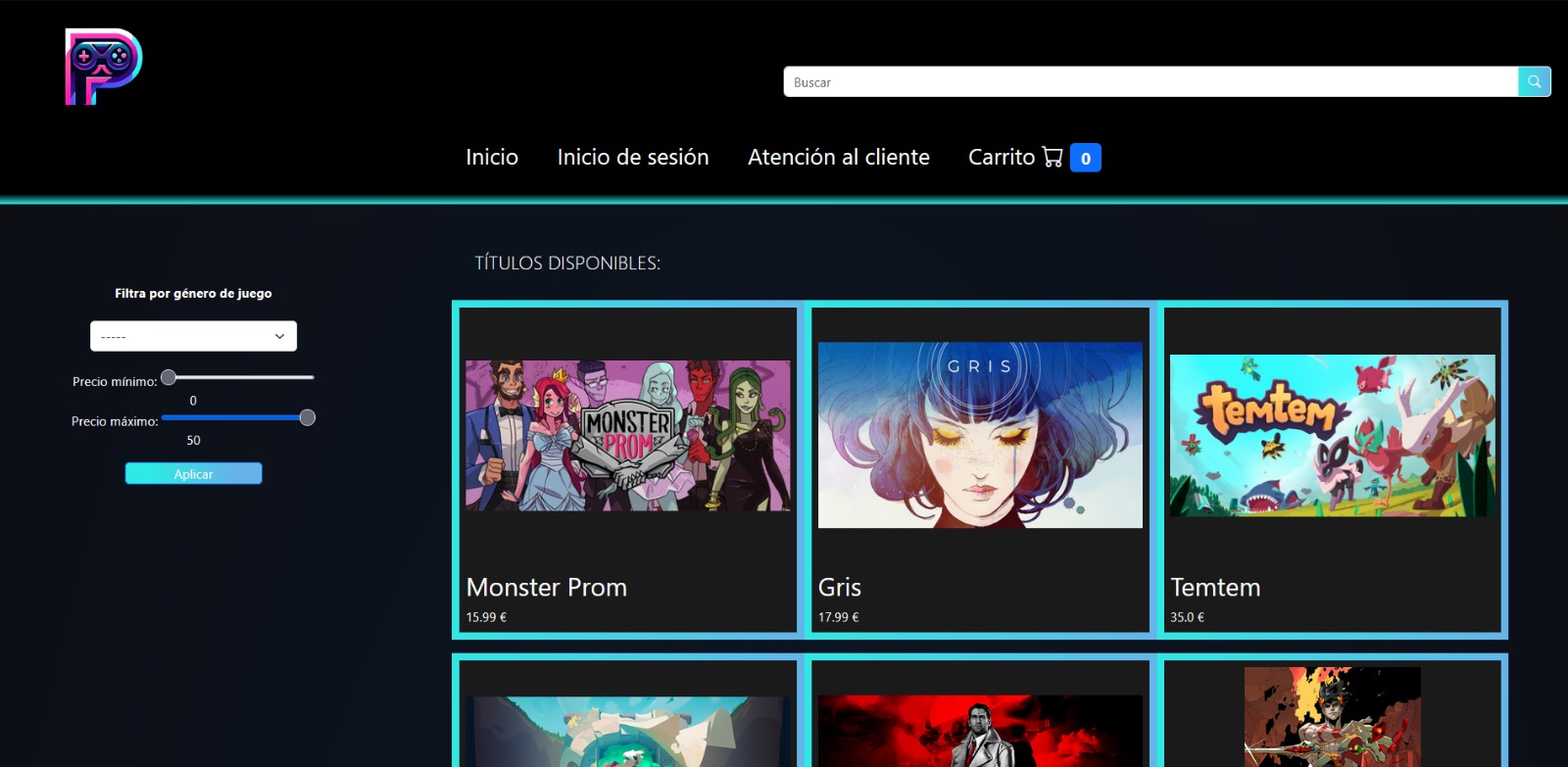

**Boceto de la página de juegos:**



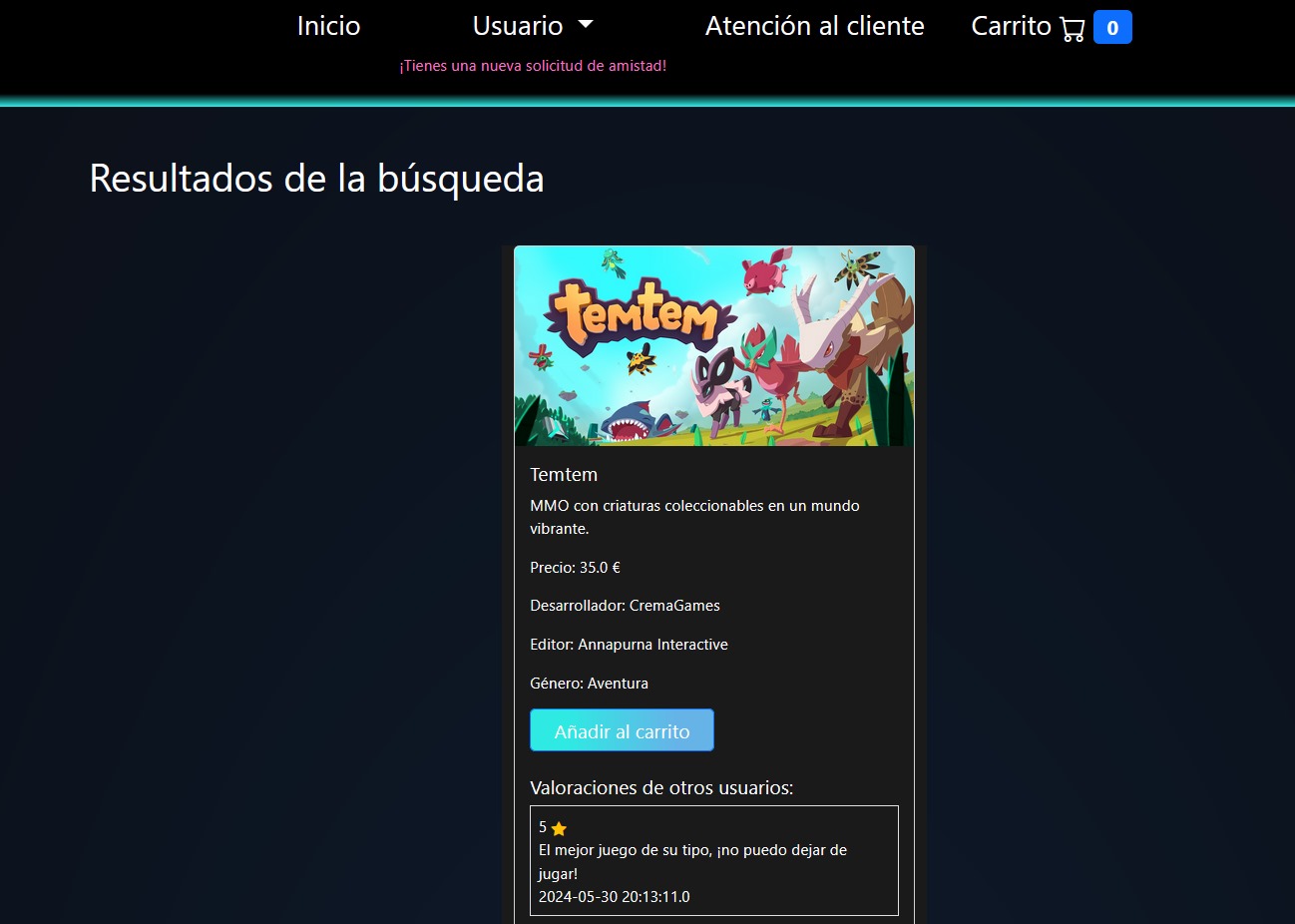
**Boceto de la biblioteca:**  


  
  
**Boceto del formulario de pago:**  
  

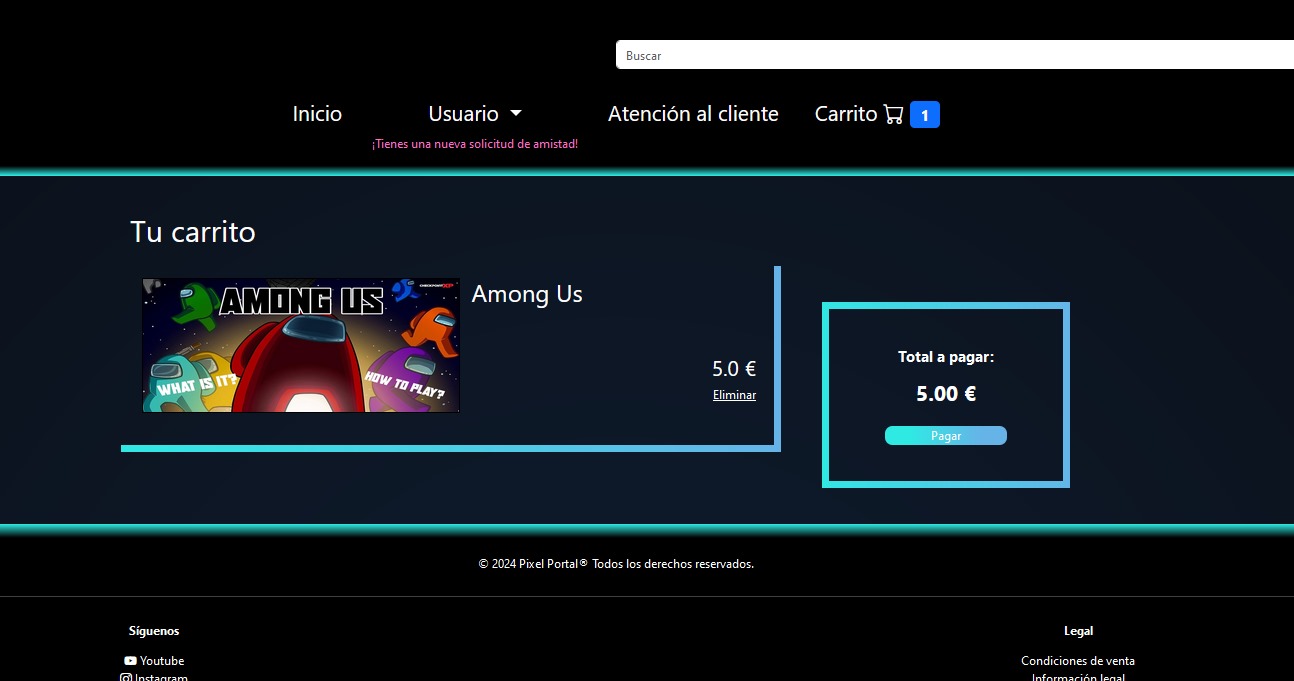

### 3.1.3 Diseño final

**Página de inicio:**  
  
Página de inicio en el cual aparecen todos los juegos de la página, pudiendo filtrar los juegos por género y/o precio. Al aplicar los filtros, se listarán en formato de “cards” o tarjetas aquellos juegos que cumplan la características establecidas en el *aside* de la izquierda.  
  
Aparte, se podrá añadir este juego al carrito previa autenticación.

En cada juego, aparecerán sus reviews y valoración media.



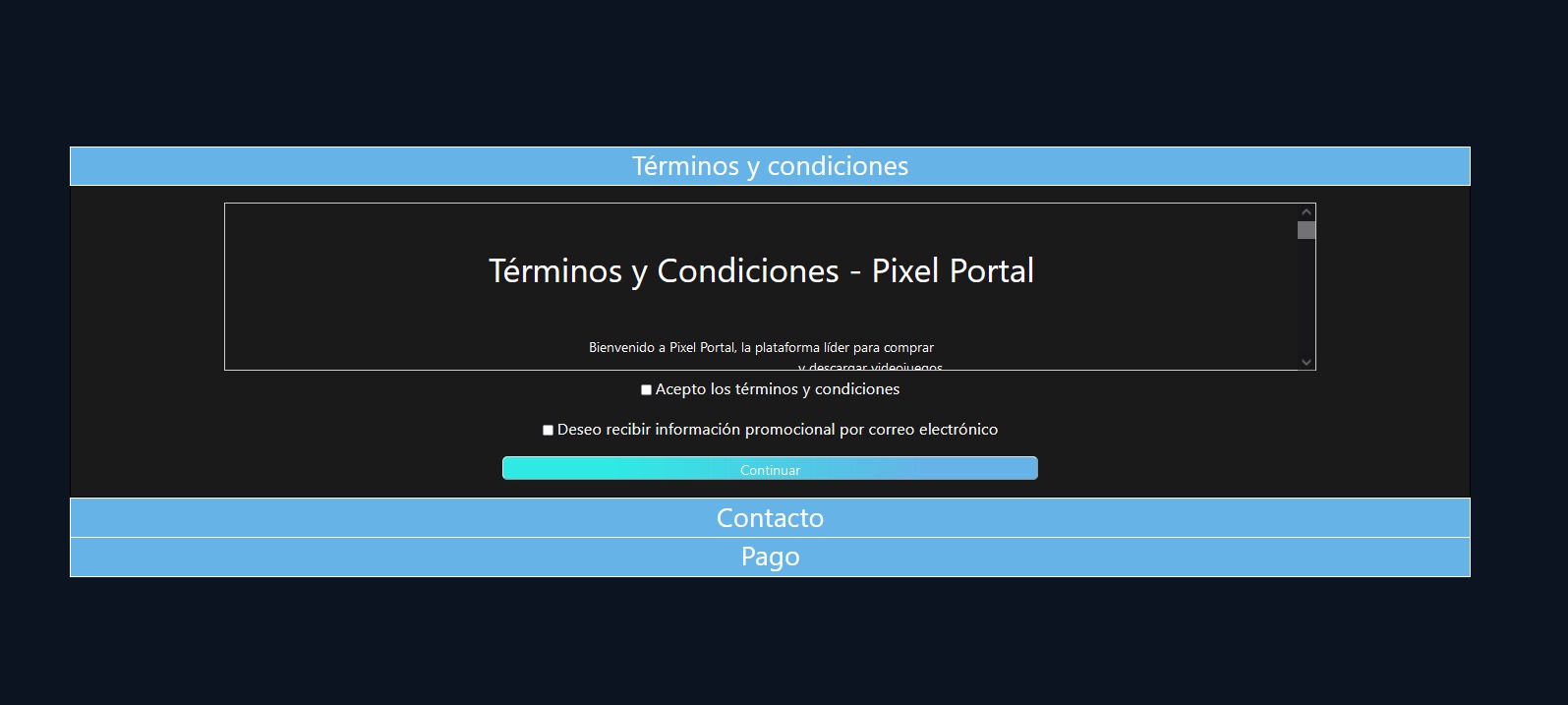
**Página de carrito:**

Al añadir el juego al carrito, te aparecerá ese mismo juego junto a su precio y se le da al usuario la opción de eliminarlo.  
  


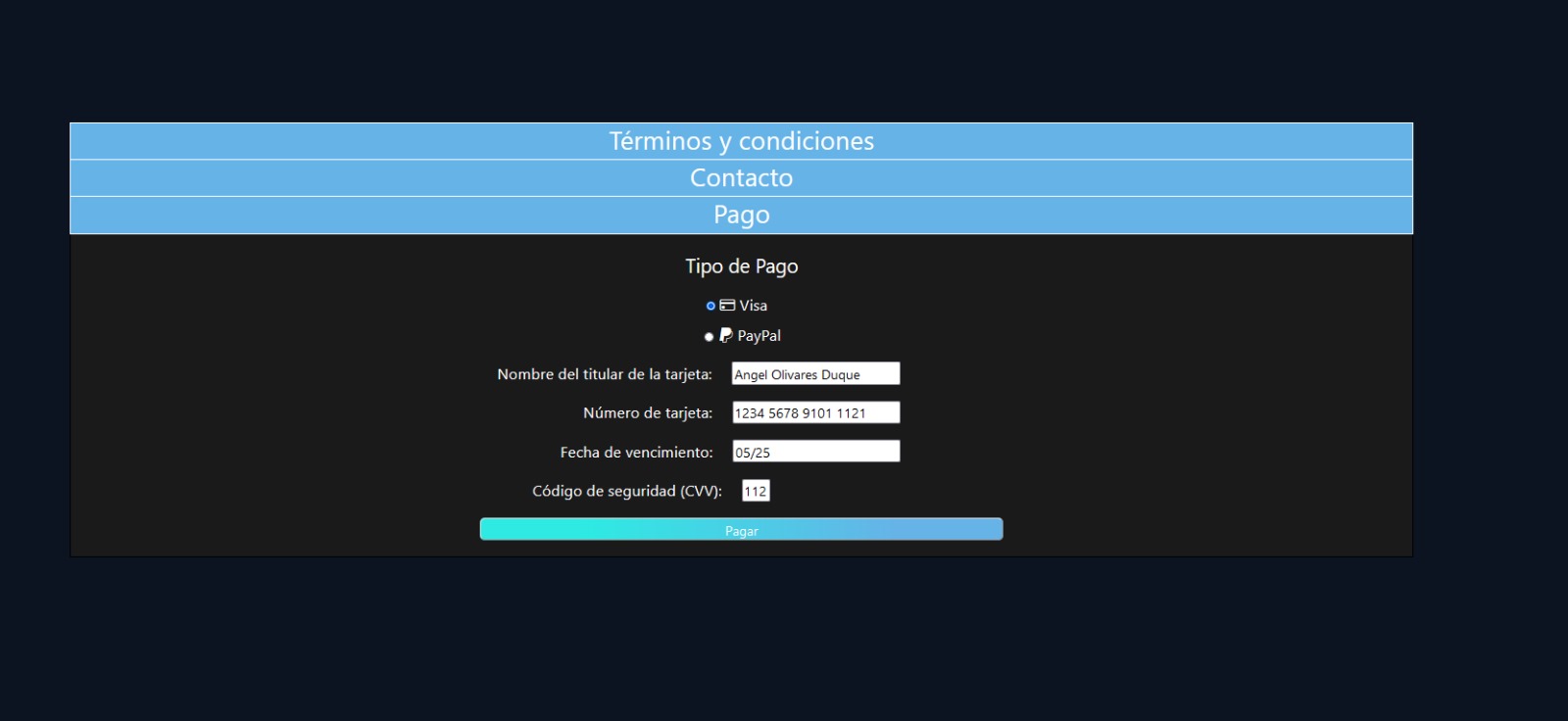
A la derecha, aparecerá la suma de todos los precios que haya en el carrito y al pulsar sobre el botón de pago, el/la usuario/a será redireccionado al formulario ideado para tal efecto.

**Formulario de pago:**

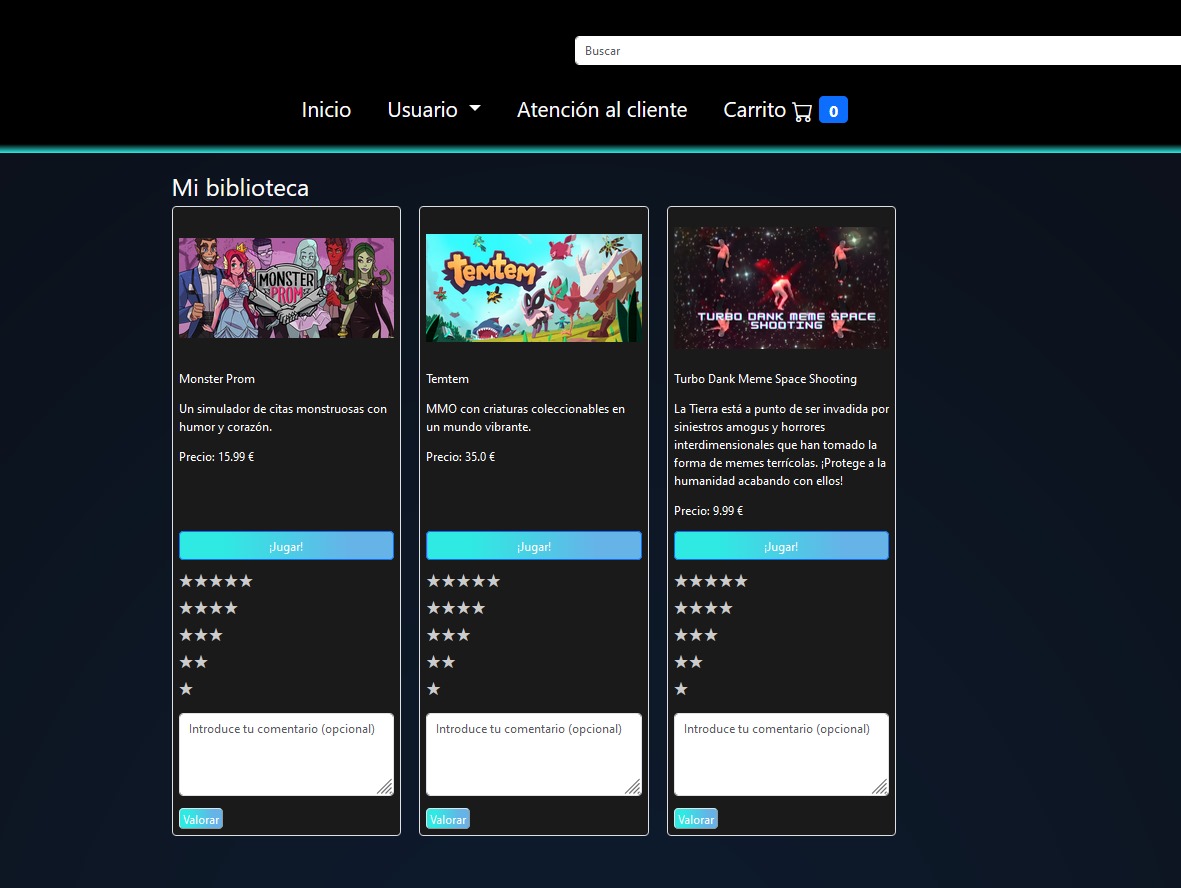
El formulario para pagar que se divide en tres partes:  
  
1. Primer formulario: se le ofrece al usuario a leer los términos y condiciones de la plataforma, que han de ser aceptados para continuar con el proceso.

2. Segundo formulario: se piden los datos del usuario, con obligatoriedad de repetir el email por seguridad y verificación, además de con intención de evitar errores; si ambos campos de correo no coinciden, no se podrá enviar el formulario ni continuar con el proceso.  
  
3. Tercer formulario: se piden los datos de la tarjeta del usuario para poder realizar el pago mediante Visa; o bien se da la opción de pagar a través de PayPal accediendo a su cuenta directamente desde la página.  
  
El número de la tarjeta tendrá por defecto el formato xxxx-xxxx-xxxx-xxxx y el campo para el código CCV se habrá limitado a tres dígitos.  






**Página de biblioteca:**



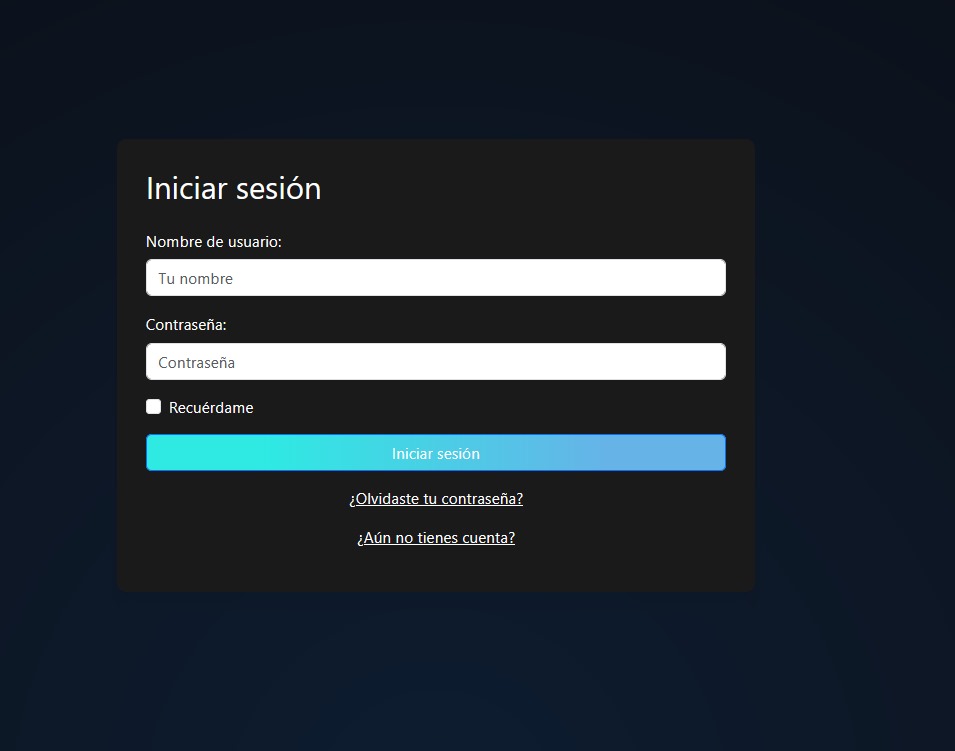
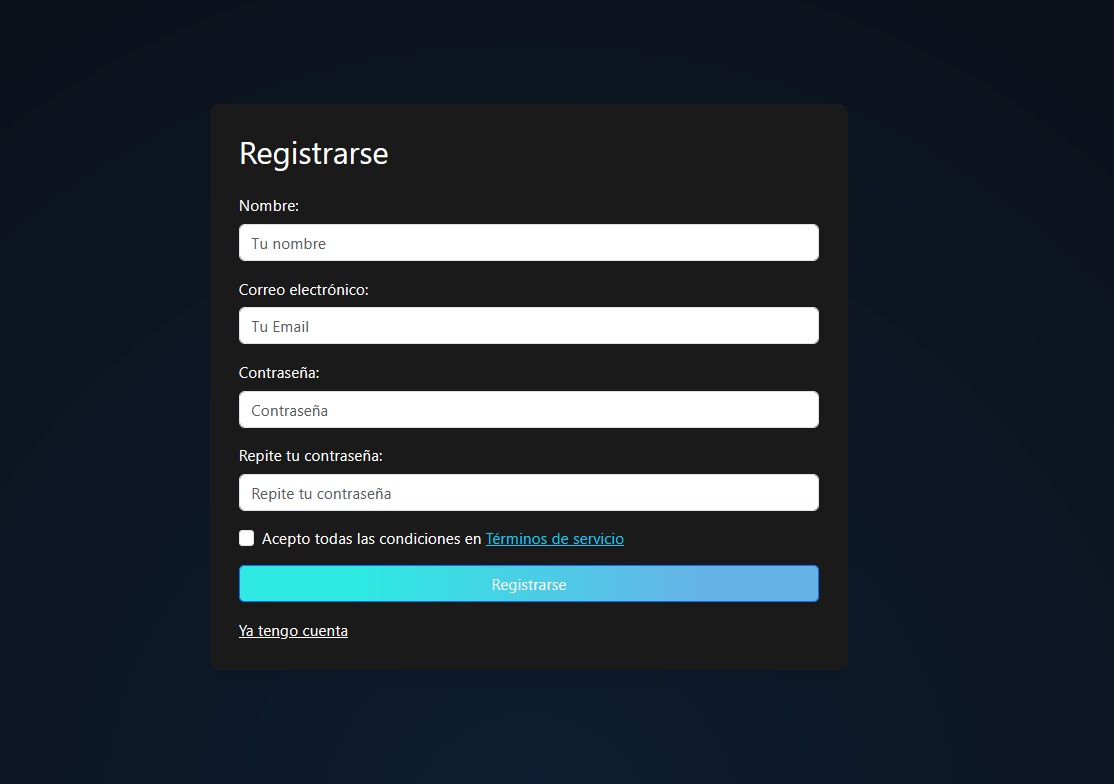
Al rellenar correctamente el formulario de pago, se añadirá correctamente el juego a la biblioteca de ese usuario.  
  
En esta página los juegos aparecerán en formato de “cards” y se le dará la opción al usuario de jugar al juego, valorarlo y escribir una review sobre él.

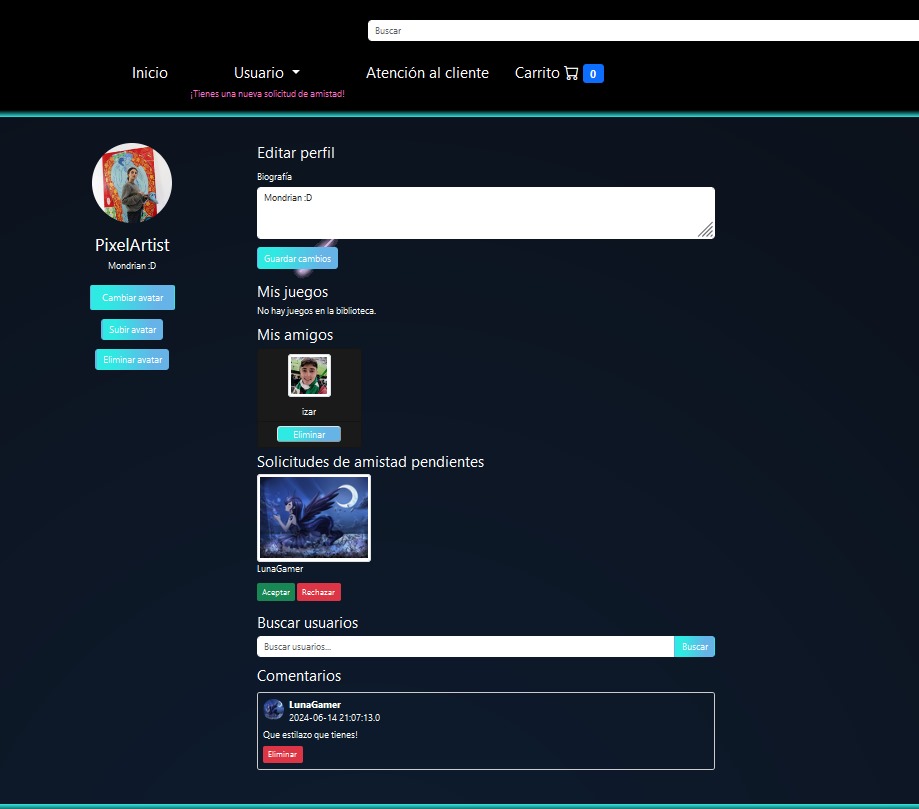
**Formulario para iniciar sesión / registrarse**

Dos formularios conectados entre sí, uno para registrarse y otro para iniciar sesión.  
  
Si el usuario no se registra, no se le proporcionará acceso a comprar juegos y guardarlos en la biblioteca ni acceso a la interfaz de usuario .

Se pedirá la información necesaria para crear una cuenta o registrarse.

Por seguridad, se necesitará repetir la contraseña al registrarse.

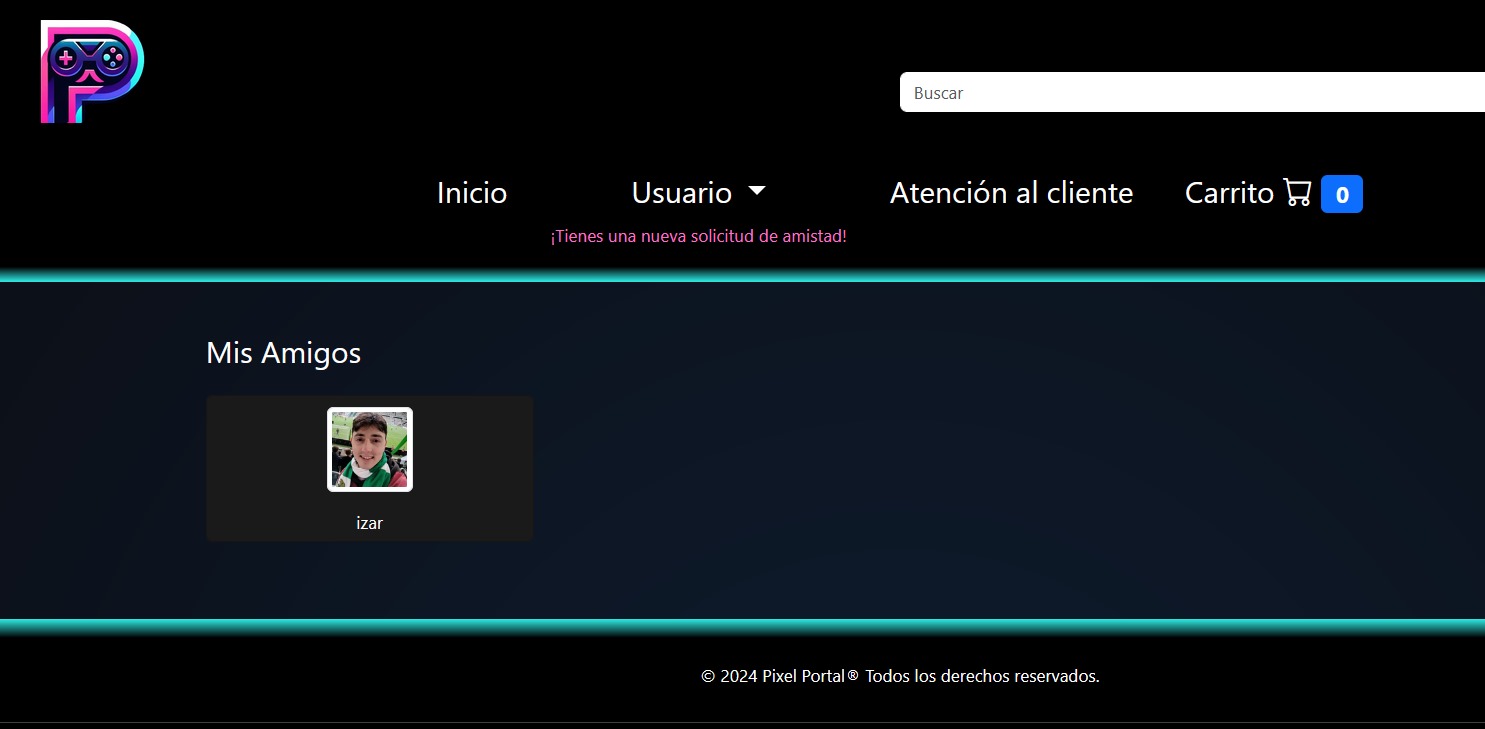
  


**Interfaz de usuario:**  


En esta página aparecerá toda la información de la cuenta

del usuario.  
  
Se podrá redactar una biografía de su cuenta y modificarla cuando el usuario desee, se mostrarán su juegos, los usuarios añadidos a amigos/as que tenga y, a su vez, las solicitudes de amistad pendientes por aceptar o rechazar.  
  
Se permite la búsqueda de usuarios y a su vez, se abre la posibilidad de añadir comentarios en perfiles de otros jugadores.  
  
A su vez, se permitirá también subir una imágen para tenerla de avatar.

Es posible también ver información limitada de otra cuenta.

  
  
  
**Página de soporte:**

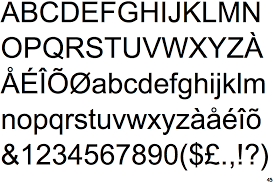
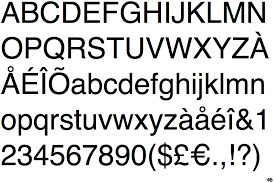
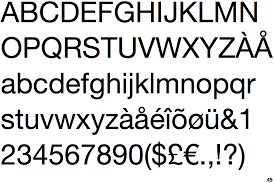
La página de soporte está diseñada para proporcionar a los usuarios toda la ayuda necesaria para resolver cualquier problema o duda que puedan tener.  
  
Será necesaria la información del usuario y la categoría de la consulta, teniendo en cuenta el problema que tenga el usuario.



### 

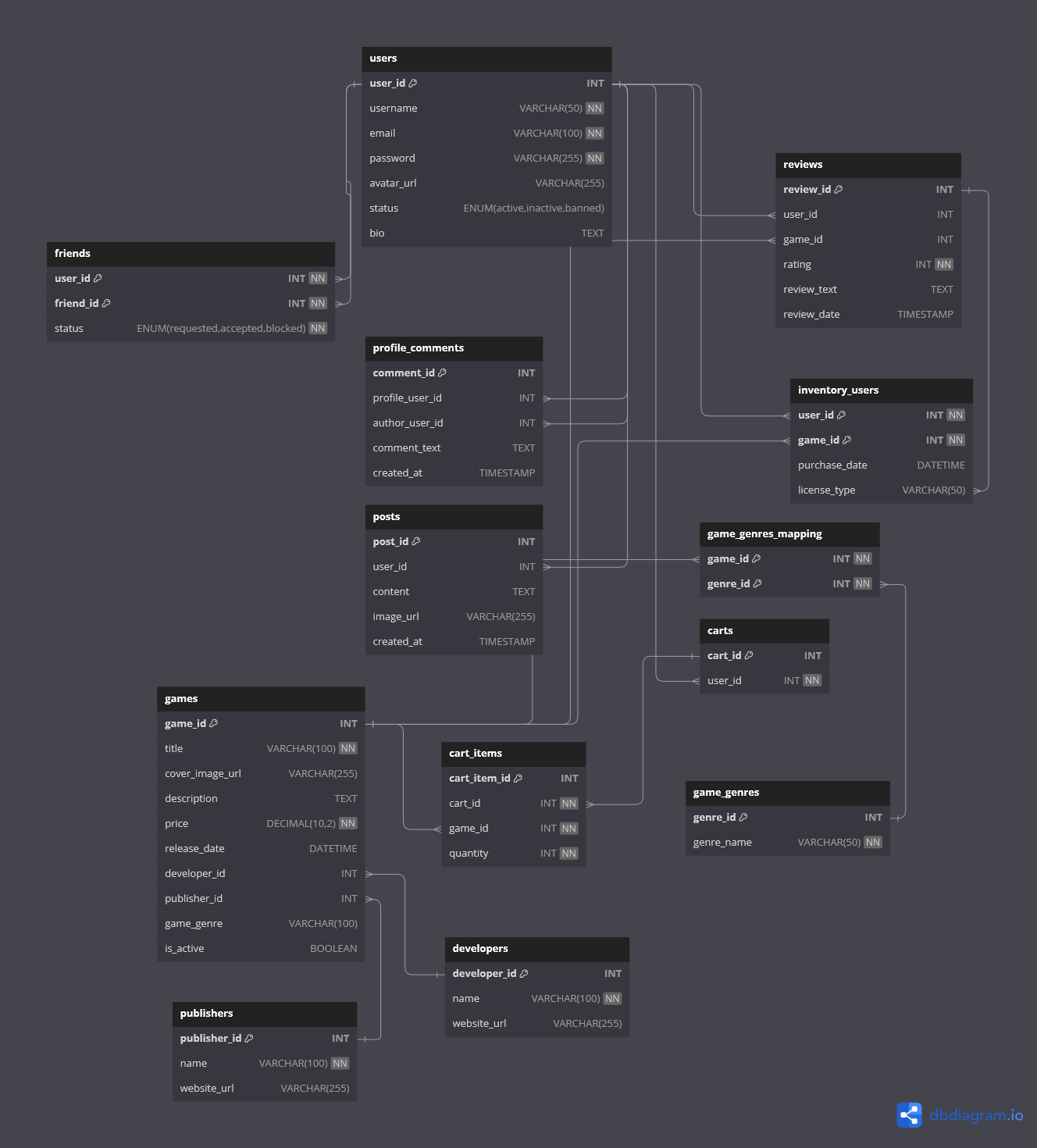
### 3.1.4 Fuentes

Las fuentes utilizadas han sido Helvetica Neue, Helvetica, y Arial. Conscientes de la popularidad y uso estándar y ampliado de las fuentes, la razón de su elección ha sido potenciar la accesibilidad y usabilidad, minimizando el uso de la serifa e intentando al máximo no comprometer ninguna de ambas consideraciones a la hora del desarrollo mientras se mantienen la estética deseada.



## 3.2. Diseño del modelo de datos

El diseño de la base de datos y las relaciones se muestran adjunto y detallados más abajo, habiendo priorizado la integridad y una posterior escalabilidad potencial:



La relación de relaciones entre tablas, valga la redundancia, queda así:

### **Users**

Relaciones:

**inventory\_users**: uno a muchos. Un usuario puede tener todos los títulos que adquiera en su inventario.

**profile\_comments** (author\_user\_id, profile\_user\_id): uno a muchos. Un usuario puede tener infinidad de comentarios en su perfil.

**friends** (user\_id, friend\_id): uno a muchos. Un usuario puede tener tantos amigos como desee, previa aprobación del usuario destino de la invitación.

**reviews**: uno a muchos. Un usuario puede escribir tantas reseñas como juegos tenga.

**posts**: uno a muchos. Un usuario puede crear múltiples publicaciones en distintos perfiles.

### **Developers**

Relaciona **games**: uno a muchos. Un desarrollador puede crear multitud de juegos.

### **Publishers**

Relaciona **games**: uno a muchos. Un distribuidor puede publicar múltiples juegos.

### **Games**

Relaciones:

**developers** (developer\_id): muchos a uno. Un juego es desarrollado por un único desarrollador.

**publishers** (publisher\_id): muchos a uno. Un juego es publicado por un único distribuidor.

**cart\_items** (game\_id): uno a muchos. Un juego puede estar en múltiples carritos pertenecientes a múltiples usuarios, ya que cada usuario/a tiene el suyo.

**inventory\_users** (game\_id): uno a muchos. Un juego puede estar en tantos inventarios como usuarios/as.

**reviews** (game\_id): uno a muchos. Un juego puede tener tantas reseñas como usuarios/as.

**game\_genres\_mapping** (game\_id): uno a muchos. Un juego puede tener múltiples géneros.

**posts** (game\_id): uno a muchos. Un juego puede tener múltiples publicaciones relacionadas.

### **Carts**

Relaciones:

**users** (user\_id): muchos a uno. Un carrito pertenece a un único usuario.

**cart\_items** (cart\_id): uno a muchos. Un carrito puede tener múltiples títulos esperando a la compra final.

### **Cart\_items**

Relaciones:

**carts** (cart\_id): muchos a uno. Un elemento de carrito pertenece a un carrito.

**games** (game\_id): muchos a uno. Un elemento de carrito contiene un juego.

### **Inventory\_users**

Relaciones:

**users** (user\_id): muchos a uno. Un inventario de usuario pertenece a un usuario.

**games** (game\_id): muchos a uno. Un inventario de usuario contiene un único mismo título (Ej. TemTem no podrá estar dos veces).

### **Profile\_comments**

Relaciones:

**users** (profile\_user\_id): muchos a uno. Un comentario de perfil pertenece a un usuario (perfil destino).

**users** (author\_user\_id): muchos a uno. Un comentario de perfil pertenece a un usuario (autor del envío y del comentario en sí).

### **Friends**

Relaciona **users** (user\_id, friend\_id): muchos a uno. Un amigo siempre es un único usuario.

### **Reviews**

Relaciones:

**users** (user\_id): muchos a uno. Una reseña es escrita por un usuario.

**games** (game\_id): muchos a uno. Una reseña es sobre un juego en concreto.

### **Game\_genres**

Relaciona **game\_genres\_mapping** (genre\_id): uno a muchos. Un género puede estar asociado con múltiples juegos.

### **Game\_genres\_mapping**

Relaciones:

**games** (game\_id): muchos a uno. Un mapeo de género de juego pertenece a un juego.

**game\_genres** (genre\_id): muchos a uno. Un mapeo de género de juego pertenece a un género.

### **Posts**

Relaciona **users** (user\_id): muchos a uno. Una publicación es creada por un único usuario.

# 

# 4. Aspectos relevantes de la implementación

La tónica general durante el desarrollo ha sido proporcionar una implementación integral y directa, que no requiriera de grandes medidas o complejidades que, a nuestro modo de ver, son contrarias a los principios por los que nos hemos guiado y en los que nos hemos formado.

Con respecto a la seguridad, al ser un proyecto a modo de demostración de una plataforma *e-commerce* del sector de los videojuegos, aplicar una encriptación y cifrado de contraseñas y de transacciones en la pasarela de pago se revela vital, y contamos con incorporarlo en un futuro.

La autenticación también requiere de futuras mejoras que no dudaremos en implementar: doble factor y guardado en la nube de contraseñas y partidas guardadas en los juegos.

## 5. Pruebas de accesibilidad y usabilidad 5.1 Validación y alta accesibilidad Subimos todas las páginas de la web al validador de consorcio W3C para asegurar la calidad y el cumplimiento de los estándares web. El resultado ha sido perfecto, lo que nos permite garantizar una página con excelente compatibilidad, accesibilidad y rendimiento. Este logro no sólo mejora la experiencia del usuario, sino que también fortalece la integridad técnica del proyecto. Además, hemos procedido a validar la accesibilidad de la página web usando eXaminator, el cual nos proporciona una nota media según el nivel de accesibilidad. Nuestra página ha obtenido una media de 8.32 entre todas las páginas evaluadas. Para asegurar la accesibilidad, hemos prestado especial atención a los detalles: todas las imágenes incluyen texto descriptivo (alt text) y no hay enlaces vacíos,

facilitando la navegación para usuarios con discapacidades visuales y mejora la usabilidad general de la página.

## 5.2 Test de usuario **Objetivo**

Evaluar la usabilidad y la experiencia del usuario en las diferentes secciones de la página web.

#### **Duración**

Aproximadamente 30-45 minutos.

#### **Materiales Necesarios**

* Computadora o dispositivo móvil con acceso a internet
* Papel y bolígrafo o una aplicación para tomar notas

#### **Participante**

* Nombre: Lars Hansson
* Edad: 45 años
* Nivel de experiencia en videojuegos: Intermedio
* Frecuencia de uso de tiendas online de videojuegos: Rara vez

### 

### **Respuestas del Test de Usuario**

#### 

#### 1. Navegación Inicial

**Tarea: abre la página principal de la tienda de videojuegos.**

Pregunta**:** ¿la página carga rápidamente?

Respuesta**:** sí, la página cargó en menos de 3 segundos.

Pregunta: ¿puedes identificar rápidamente la barra de navegación y las categorías principales?

Respuesta**:** sí, la barra de navegación es clara y las categorías están bien definidas.

#### 2. Exploración de Juegos

**Tarea: Navega a la página de los juegos.**

Pregunta: ¿es fácil encontrar la sección de juegos desde la página principal?

Respuesta: sí, hay un enlace directo a la sección de juegos en la barra de navegación.

Pregunta: ¿la página de los juegos está bien organizada y es fácil de navegar?

Respuesta: sí, los juegos están organizados por categorías y hay opciones de filtrado visibles.

Tarea: busca un juego específico utilizando la barra de búsqueda.

Pregunta: ¿el resultado de la búsqueda fue relevante y rápido?

Respuesta: sí, encontré el juego que buscaba inmediatamente.

Tarea: filtra los juegos por categoría (por ejemplo, acción, aventura, etc.).

Pregunta: ¿el sistema de filtrado es intuitivo y efectivo?

Respuesta: sí, los filtros son fáciles de usar y los resultados cambian rápidamente.

#### 

#### 3. Carrito de Compras

**Tarea: Añade un juego al carrito.**

Pregunta: ¿el proceso de añadir un juego al carrito es sencillo?

Respuesta: sí, solo tuve que hacer clic en el botón "Añadir al carrito".

Pregunta: ¿recibiste alguna confirmación visual o notificación al añadir el juego al carrito?

Respuesta: sí, apareció una notificación en la esquina superior derecha.

**Tarea: Revisa el carrito de compras.**

Pregunta: ¿es fácil acceder al carrito de compras?

Respuesta: sí, hay una sección de carrito en la barra de navegación.

Pregunta: ¿la información en el carrito (precio, título del juego, cantidad) es clara y completa?

Respuesta: sí, todo está bien detallado.

#### 4. Proceso de Pago

Tarea: completa el formulario para pagar los juegos.

Pregunta: ¿el formulario de pago es fácil de entender y completar?

Respuesta: sí, el formulario es claro y los campos están bien etiquetados.

Pregunta: ¿el proceso de pago es seguro y confiable?

Respuesta: sí, el proceso parece seguro y confiable, hay indicaciones de seguridad visibles.

#### 5. Registro e Inicio de Sesión

**Tarea: Regístrate en la página web.**

Pregunta: ¿il formulario de registro es fácil de encontrar y usar?

Respuesta: sí, el enlace para registrarse está en la barra de navegación.

Tarea: inicia sesión con las credenciales recién creadas.

Pregunta: ¿el formulario de inicio de sesión es accesible y fácil de usar?

Respuesta: sí, el formulario está conectado al de registro y es fácil de usar.

Pregunta: ¿tuviste algún problema para iniciar sesión?

Respuesta: No, pude iniciar sesión sin problemas.

#### 

#### 

#### 6. Biblioteca de Juegos

**Tarea: Accede a tu biblioteca de juegos.**

Pregunta: ¿es fácil encontrar y acceder a la biblioteca de juegos?

Respuesta: sí, hay un enlace directo a la biblioteca en la barra de navegación.

Pregunta: ¿los juegos en tu biblioteca están bien organizados y son fáciles de gestionar?

Respuesta: sí, están organizados por orden de compra y es fácil ver el estado de cada juego.

#### **Cuestionario Final**

Pregunta: ¿qué fue lo que más te gustó de la página web?

Respuesta: la organización y la claridad de la información. Todo está donde esperaba encontrarlo.

Pregunta: ¿qué fue lo más frustrante?

Respuesta: nada en particular, todo fue bastante fluido.

Pregunta: ¿tienes alguna sugerencia para mejorar la página?

Respuesta: tal vez agregar más ofertas o promociones para poder crear anuncios.

### **Conclusión**

Lars tuvo una experiencia positiva navegando y utilizando las funciones principales de la página web. Sus comentarios destacan la facilidad de uso y la organización clara de la información, lo que es un buen indicativo de la usabilidad de la web.

## 5.3 Test A/B El objetivo de este test A/B ha sido establecer una comparación entre el rendimiento y la usabilidad del diseño final de nuestra página de soporte con varios bocetos preliminares. Queremos determinar cuál de estos diseños proporciona la mejor experiencia de usuario y facilita la resolución de problemas de manera más eficiente.

#### Objetivos

* Medir la satisfacción del usuario con cada diseño.
* Evaluar la eficiencia en la resolución de problemas.
* Identificar preferencias de los usuarios en términos de diseño y funcionalidad.
* Recolectar feedback para futuras mejoras.

Participantes

* Lars Hansson, CEO de [Nrlyze](https://www.nrlyze.se/en/).
* Aitor Arnau, desarrollador web.

Variantes

* Diseño A ( diseño final ): la versión completamente desarrollada y lanzada de la página.
* Diseño B ( bocetos ): bocetos preliminares con diferentes estructuras y funcionalidades.

Resultados

##### Tasa de Éxito en tareas realizadas

* Diseño A: 85%
* Diseño B: 75%

Duración (Promedio)

* Diseño A: 3.5 minutos
* Diseño B: 4.5 minutos

##### Nivel de Satisfacción (Escala de 1 a 5)

* Diseño A: 4.5
* Diseño B: 4.0

Feedback

* Diseño A: usuarios elogiaron la claridad y accesibilidad de la página.
* Diseño B: comentarios sobre la simplicidad pero falta de algunas funcionalidades.

#### **Conclusión**

El Diseño A (diseño final) mostró un mejor rendimiento en todas las métricas clave, destacando en tasa de éxito, tiempo de resolución, y satisfacción del usuario. Esto significa que el diseño A proporciona una experiencia de usuario superior en comparación con los bocetos preliminares.

# 6. Videojuego-demo

*Turbo Dank Meme Space Shooting*

**Introducción**

El juego está diseñado para ejecutarse en navegadores web y ofrece una experiencia dinámica de combate espacial con una estética visual llamativa y una banda sonora que coincide con la sensación que pretende transmitir al jugador.

**Motor del Juego: Construct 3**

Construct 3 es un motor de juegos basado en HTML5 diseñado para la creación de juegos 2D. Ofrece un entorno de desarrollo visual que permite a los usuarios crear juegos sin necesidad de escribir código, aunque también soporta scripting para funcionalidades más avanzadas.

**Requerimientos**

Para jugar, se requiere un navegador web moderno que soporte HTML5. No es necesario instalar software adicional. Gracias a ello, es muy poco probable que el jugador experimente problemas de compatibilidad para poder ejecutar el juego.

Estructura de Archivos

- index.html: archivo principal que carga el juego en el navegador.

- workermain.js: script que maneja la lógica principal del juego.

- style.css: hoja de estilos para la presentación del juego.

- data.json: contiene configuración y datos esenciales del juego.

- images/, icons/, media/: Contienen los recursos gráficos y multimedia del juego.

**Licencia de la Música de Fondo**

La música de fondo del juego es RetroFuture Clean, de Kevin Macleod. La música está sujeta a una licencia Creative Commons Reconocimiento - No Commercial - Compartir Igual (CC BY-NC-SA). Esto significa que se permite el uso, distribución y reproducción de la música de forma gratuita siempre que se otorgue crédito adecuado al autor y no se utilice con fines comerciales.

**Construcción de Eventos**

Construct 3 utiliza un sistema de eventos para definir la lógica del juego. Los eventos son estructuras condicionales que, cuando se cumplen, ejecutan acciones predefinidas. Esto permite crear una variedad de interacciones y comportamientos en el juego sin necesidad de escribir código.

**Lenguaje de Programación Utilizado**

Aunque Construct 3 permite el desarrollo de juegos sin programación directa, también admite el uso de JavaScript para usuarios que desean mayor control y personalización sobre sus proyectos. Esto ofrece una flexibilidad significativa en el desarrollo de características avanzadas y personalizadas.

Cómo Jugar

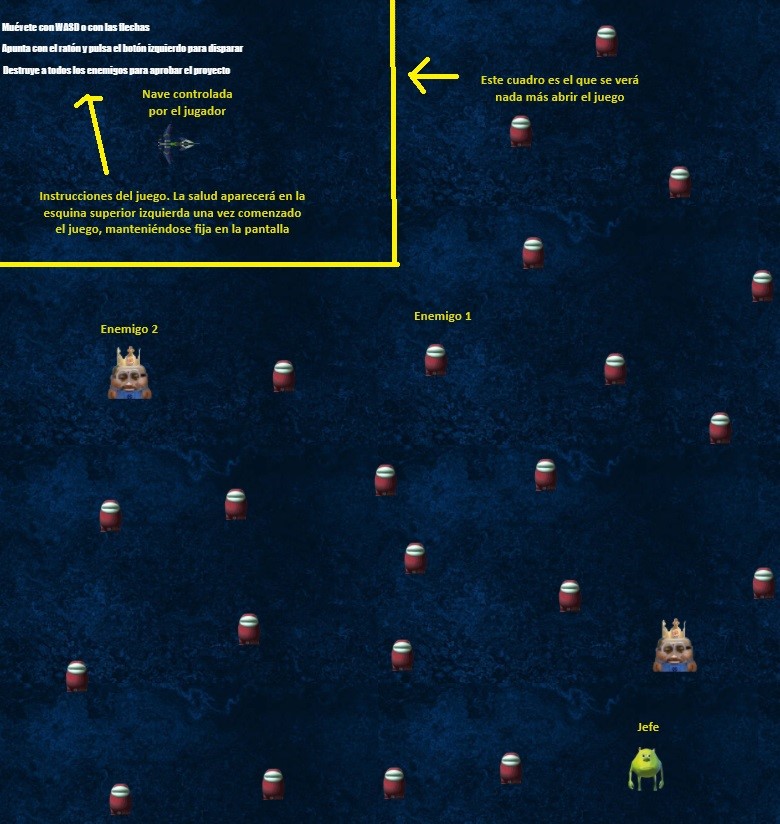
- Inicio del Juego: Acceder al juego a través del archivo `index.html`.

- Controles: Los controles típicos incluyen teclado y mouse o controles táctiles en dispositivos móviles.

- Objetivo: El objetivo del juego es maniobrar la nave espacial para esquivar los proyectiles dirigidos hacia la nave del usuario y derrotar enemigos.

Desarrollo y Personalización

Construct 3 permite modificar o expandir el juego. Esto puede incluir añadir enemigos, efectos especiales, o modificar la banda sonora y gráficos. Las personalizaciones pueden realizarse a través del editor visual del motor o mediante la escritura de código JavaScript para funcionalidades específicas.



El jugador cuenta con una salud inicial de 200 puntos. Cada vez que una bala enemiga (de color verde) le alcance, perderá 5 puntos. Sus balas (de color amarillo) restan 10 puntos por impacto a los enemigos.

El enemigo 1 cuenta con 100 puntos de salud. Además de poder derrotarlo con balas, si el jugador impacta directamente contra él, le derrotará al instante.

El enemigo 2 cuenta con 200 puntos de salud. Si el jugador impacta contra él, no pasará nada.

El jefe cuenta con 500 puntos de salud. Si el jugador impacta contra él, el jugador perderá 10 puntos de salud. Cuanto más tiempo esté en contacto el jugador con el jefe, más salud perderá.

El juego está diseñado para poder ser jugado por cualquier tipo de jugador ya sea casual o experimentado, dándole cierta ventaja teniendo en cuenta su inferioridad numérica, ya que los enemigos disparan una bala por segundo, mientras el jugador dispara 10 balas por segundo.

**Conclusión**

Este videojuego de naves representa una aplicación eficaz del motor Construct 3, demostrando cómo las herramientas de desarrollo visual pueden ser utilizadas para crear juegos interactivos y atractivos con un alcance considerable de personalización y expansión.

# 7. Problemas encontrados

Son numerosos los inconvenientes y problemas a los que nos hemos debido de enfrentar tanto en el *frontend, backend,* desarrollo del videojuego y para con la base de datos, destacando:

·Muchos márgenes y espacios invisibles debido a algunos elementos de bootstrap.

·Relaciones consistentes y coherentes en la base de datos, así como su tamaño desproporcionado inicial.

·Integración de librerías para la encriptación de contraseñas mediante *hash*, como [Bcrypt](https://en.wikipedia.org/wiki/Bcrypt).

·Estilos de CSS y Bootstrap no se aplicaban correctamente en ciertos elementos e inclusive modificaban otros  
  
·Elección de la paleta de colores.

·Falta de información previa para poder desarrollar el juego, a la vez que la elección del motor para desarrollarlo.

·Gestión de herramientas a nivel económico, pues la versión gratuita de *Construct 3* tenía muy pocas prestaciones, con una versión de pago de un precio realmente elevado.

·Motores gratuitos algo más completos, pero mucho más complejos y complicados de entender.

·Numerosos errores y fallos catastróficos, afortunadamente solventados, en la comunicación entre todos los elementos del MVC.

# 

# 8. Conclusiones

Hemos aprendido a crear videojuegos desde cero completamente personalizados, adaptándonos a las condiciones y complicaciones que se nos han presentado, véase errores de código, de ejecución o en el layout. Sin embargo, el resultado ha sido más que satisfactorio.

En el contexto de la lógica trasera, este proyecto ha sido de una inestimable ayuda a la hora de comprender en profundidad el concepto de MVC y llevarlo a cabo satisfactoriamente sin mayores percances. No obstante, hay multitud de elementos que nos hubiera gustado incorporar: despliegue real del proyecto en servidor externo, implementación de chat en tiempo real con usuarios simultáneamente operativos, uso de emojis en el propio chat y comentarios, establecimiento de pasarela de pago funcional y real, autenticación mediante servicios ajenos como, por ejemplo, [Supabase](https://supabase.com/), autenticación de doble factor, y un largo *et cetera* de necesaria mención en lo que a una plataforma *e-commerce* de altísimo potencial respecto a su escalabilidad se refiere.

Es por ello que es un proyecto en el que continuaremos trabajando para mejorar tanto el trabajo en sí como nosotros mismos como desarrolladores.

# 

# 9. Bibliografía

Video Tutorial Construct 3: <https://www.youtube.com/watch?v=QRnjv36XJHs>

Documento Tutorial Juego de Naves en Construct 3: <https://www.calameo.com/read/007124692c64371715358>

Fuente para accesibilidad y estándares:  
<https://www.w3.org>/

Página para animaciones:  
<https://blog.hubspot.com/website/css-animation-examples>

Página para encontrar fuentes:  
<https://fonts.google.com/>

Validadores de html y accesibilidad

<https://validator.w3.org/#validate_by_upload>

<http://examinator.net/>

Página para iconos:

<https://icons.getbootstrap.com/?q=flag>

Backend:

<https://www.tutorialspoint.com/jsp/index.htm>

<https://www.javatpoint.com/jsp-tutorial>

https://chatgpt.com

https://todocodeacademy.com/course/java-web-con-jsp-y-jpa/

<https://www.oracle.com/technical-resources/articles/javase/webappdev.html>

https://rua.ua.es/dspace/bitstream/10045/3300/5/introjsp.pdf

Infinidad de vídeos en https://www.youtube.com/

Material y documentos durante los cursos DAW en

https://www.educa2.madrid.org/web/centro.ies.ellago.madrid