

**PROYECTO DE CICLO DE G.S. DE
DESARROLLO DE APLICACIONES WEB**

CHANGAME



JAVIER GARCÍA MONTERO
CURSO 2023/24

Resumen

Se plantea la oportunidad de diseñar una aplicación de intercambio de trueque de juegos en la medida en que la gente compra cada vez más cosas. Sin embargo, no todos los elementos son necesarios. Es una pena tirarlos. No siempre es posible venderlos de forma rentable. Por tanto, muchos están volviendo al sistema de intercambio tradicional. Una aplicación de intercambio de bienes puede resultar muy útil para modernizar esta práctica.

La posibilidad de realizar donaciones pueden convertirse en toda una plataforma de reputación en la que los usuarios aportan su grano de arena en un objetivo de desarrollo sostenible. La idea de proporcionar una herramienta donde pueden participar en campañas de recogida de juegos y/o aportaciones económica para proporcionar a poblaciones con pocos recursos la posibilidad de disponer de juegos, es uno de los objetivos en lo que se fundamenta esta aplicación.

Estas funcionalidades servirá de punto de partida a un objetivo mayor, como puede ser la incorporación de una aplicación ecommerce de compraventa de juegos.

La aplicación web proporciona un espacio donde un usuario tiene la posibilidad de ofrecer un juego que ya no utiliza, de forma que le permita la posibilidad de obtener a cambio otro de igual o distinta naturaleza. Todo ello, mediante el establecimiento de una serie de premisas que se deben cumplir.

A nivel técnico, se desarrolla una aplicación web híbrida progresiv (PWA) con diseño responsive, de forma que puede usarse a través de un dispositivo móvil, mediante el uso de Laravel como framework de desarrollo con php, un sistema de gestión de bases de de datos como mysql, una biblioteca de diseño como Bootstrap, interfaces de aplicaciones API-REST con JSON y plugins específicos que servirán para la mejora la usabilidad web.

Índice

1	Introducción	4
1.1	Datos del proyecto	4
1.2	Planificación	4
1.2.1	Diagrama de Gantt	4
1.2.2	Planificación semanal del proyecto	5
1.3	Objetivos generales	7
1.4	Elementos del proyecto	8
1.5	Estado de Arte	8
1.6	Público objetivo	8
2	Tecnologías	9
2.1	Metodologías de desarrollo	9
2.2	Herramientas de desarrollo	10
2.2.1	Herramientas de BackEnd	10
2.2.2	Herramientas de FrontEnd	10
2.2.3	Plugins y utilidades para la aplicación de desarrollo	10
2.2.4	Paquetes para el desarrollo de la aplicación	11
2.2.5	Herramientas gráficas	11
2.2.6	Herramientas de gestión de proyecto	11
3	Análisis	12
3.1	Descripción funcional de la aplicación	12
3.2	Diagramas UML	13
3.2.1	Diagrama de contexto	13
3.2.2	Casos de uso	13
4	Diseño	13
4.1	Modelo Entidad/Relación (E/R)	13
4.2	Modelo Relacional	13
4.3	Patrón de arquitectura MVC (Modelo-Vista-Controlador)	13
4.4	Componentes de Interfaz	18
4.4.1	Elementos de Identificación	18
4.4.2	Elementos de Navegación	18
4.4.3	Elementos de Contenidos e Interacción	19
4.5	Guías de estilo	19
4.5.1	Logotipo	19

4.5.2	Favicon	19
4.5.3	Tipografía	19
4.5.4	Colores	19
4.5.5	Botones	22
4.5.6	Input	22
4.5.7	iconos utilizados	22
5	Implementación	22
5.1	Definición de filtrados	22
5.2	Definición de datos Enum	24
5.3	Control de Autorización de acceso	24
5.4	Control de incluir en Favoritos	24
5.5	Control de incluir en Acuerdos	24
5.6	Control de aportaciones en Campañas	24
5.7	Instalación de API MailChimp para Newsletter	24
5.8	Aplicación de paquetes	24
5.8.1	Paquete FileInput	24
5.8.2	Paquete DataTables	24
5.8.3	Paquete Pagination	24
6	Conclusiones	24
7	Dificultades encontradas	24
7.1	Instalación de librería JQuery	24
7.2	Modales de Bootstrap en paquetes instalados	25
7.3	Filtrado en las búsqueda de productos	25
8	Bibliografía	25

1 Introducción

Esta aplicación parte de la idea de que un juego puede tener distintas vidas de uso, o más concretamente, con el principio de que distintas personas puedan utilizar de un mismo juego. Por este motivo, se ha desarrollado una aplicación que proporciona una plataforma donde se puede ofrecer un juego propio, con idea de localizar a alguien que pueda estar interesado en él, de forma que se pueda alcanzar un acuerdo de intercambio. Esta aplicación permite localizar juegos a través de distintos criterios de búsqueda. Uno de los elementos más relevante de la aplicación es la posibilidad de gestión de los productos propios, así como los juegos de aquellas personas potenciales con las que se puede llegar a un acuerdo.

Por otra parte, la posibilidad de acercar los juegos a aquellas personas que tienen menos posibilidades económicas o acceso, se plantea la posibilidad de proporcionar un espacio donde se puedan dar donaciones de juegos a los que ya no se le da uso o, incluso, la posibilidad de dar aportaciones económicas para campañas de recogida de fondos para la compra de juguetes.

Se desarrolla la aplicación atendiendo a la perspectiva de poder ser utilizada en distintos tipos de dispositivos, por lo que se desarrollará mediante un diseño responsive, y en la medida de lo posible, mediante la utilización de frontend adaptado a dispositivos móviles.

Aunque esta aplicación, se ha planteado a partir del intercambio de juguetes entre particulares, se puede extrapolar a distintos tipos de productos.

1.1 Datos del proyecto

Nombra	Apellidos	Título	Ciclo	Año	Centro
Javier	García Montero	Changame (Change Game)	CFGS DAW	2023-2024	I.E.S. Virgen del Carmen

1.2 Planificación

1.2.1 Diagrama de Gantt

Se muestra la planificación del proyecto mediante diagrama de Gantt

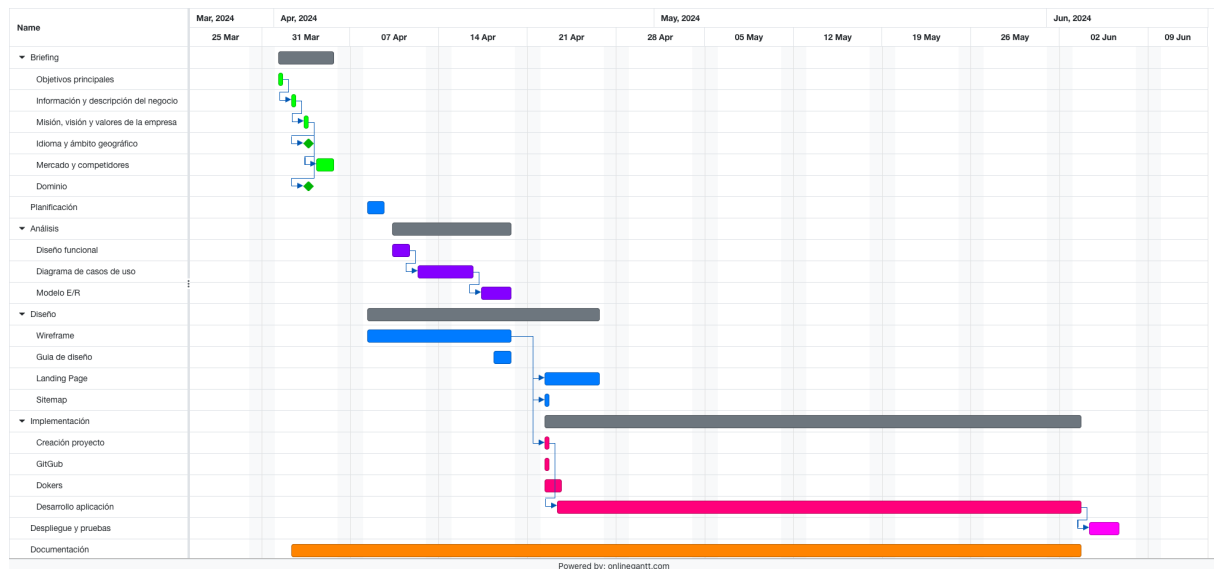


Figura 1: Diagrama de Gantt

1.2.2 Planificación semanal del proyecto

Semana del 1 al 7 de marzo

- Organización del trabajo. Se definen cinco líneas de trabajo.
 1. El análisis funcional y recopilación de los contenidos.
 2. Definición de diseño e interfaz gráfica
 3. Desarrollo de la aplicación
 4. Elaboración de la documentación
- Ideas iniciales (SCRUM)
 1. Seguimiento semanal de avance
 2. Evaluación de dificultades encontradas
 3. Adaptación funcional atendiendo a las dificultades encontradas
 4. Reevaluación funcional de aplicación
 5. Reunión tutor cada 15 días
- Elementos del proyecto
 1. Software
 2. Documentación (memoria)
 3. Presentación
- Preparación de la plantilla del documento

Semana del 8 al 14 de abril

- Título. Changame (*Change Game*)
- Definición de los objetivos iniciales
- Breve resumen
 - Abstract
 - Introducción
- Investigación de estudios y proyectos similares (estado del arte)

Semana del 15 al 21 de abril

- Investigación de estudios y proyectos similares (estado del arte) [D]
- Introducción [D]
- Objetivos definitivos [D] - exactamente qué estamos haciendo
- Material y recursos a utilizar [D]

Semana del 22 al 28 de abril

- Métodos seguidos en el proceso, metodologías, tecnologías (justificación)
- Desarrollo de modelo gráfico funcional (páginas web de contenido)
- Formación y definición del proyecto

Semana del 29 de abril al 5 de mayo

- Resultados iniciales (“bocetos” del programa).
- Verificación y Visto Bueno.
- Analizar si hace falta cambiar algo de los requisitos o tecnologías inicialmente planificadas y explicar si hay algún cambio porqué se ha hecho [D]

Semana del 6 al 19 de mayo

- Resultados intermedios (demo funcional)
- Primera revisión de la documentación para ver que estén todos los puntos necesarios

Semana del 20 al 26 de mayo

- Resultados finales (proyecto terminado: tutorial, aplicación...)
- Segunda revisión del documento [todos los apartados]
- Preparación de la presentación

Semana del 27 de mayo al 9 de junio

- Pulimos los posibles “bugs”

- Entrega del documento final

Semana del 12 al 14 de junio

- Organización de la presentación
- Entrega de la presentación para la exposición
- Presentación de proyectos

1.3 Objetivos generales

Intercambiar juegos puede tener una serie de ventajas tanto para los jóvenes como para sus familias y el medio ambiente.

- **Variedad de experiencia de juego:** Intercambiar juegos permite a los jugadores acceder a una amplia variedad de juegos sin tener que gastar grandes sumas de dinero en comprarlos todos. Esto les brinda la oportunidad de probar diferentes estilos de juegos, géneros y mecánicas de juego.
- **Fomenta la colaboración y la competencia amistosa:** Al intercambiar juegos, los jugadores pueden reunirse para jugar juntos, lo que fomenta la colaboración, la cooperación y la construcción de relaciones sociales positivas. También pueden competir de manera amistosa en los juegos intercambiados, lo que promueve habilidades de resolución de problemas y pensamiento estratégico.
- **Ahorro de dinero:** Intercambiar juegos es una forma económica de expandir la biblioteca de juegos de uno sin tener que comprarlos nuevos. Esto es especialmente útil dado el costo relativamente alto de muchos juegos de mesa y videojuegos.
- **Renovación de la experiencia de juego:** Intercambiar juegos usados puede brindar a los jugadores una forma de renovar su experiencia de juego al obtener juegos nuevos para ellos sin tener que gastar dinero adicional.
- **Promueve la sostenibilidad:** Al intercambiar juegos, se fomenta la reutilización y el reciclaje de juegos usados en lugar de desecharlos. Esto ayuda a reducir el desperdicio y promueve prácticas más sostenibles.
- **Exploración de nuevos intereses:** Intercambiar juegos puede permitir a los jugadores explorar nuevos intereses y géneros de juegos que de otra manera no habrían considerado comprar.

En resumen, intercambiar juegos puede ser una forma emocionante y económica de expandir la colección de juegos de uno, mientras fomenta la socialización, la diversidad de experiencias de juego y prácticas más sostenibles.

Otro aspecto a tener en cuenta es que se desarrollan Objetivos de Desarrollo Sostenible (ODS) diseñados para ser un «plan para lograr un futuro mejor y más sostenible para todos». Los ODS que se abordan en este proyecto van encaminados a los siguientes objetivos.

- ODS 4: “Garantizar una educación inclusiva, equitativa y de calidad y promover oportunidades de aprendizaje durante toda la vida para todos”
- ODS 11: “Lograr que las ciudades sean más inclusivas, seguras, resilientes y sostenibles”
- ODS 12: “Garantizar modalidades de consumo y producción sostenibles”
- ODS 13: “Adoptar medidas urgentes para combatir el cambio climático y sus efectos”



1.4 Elementos del proyecto

En este proyecto se pretende generar una aplicación web funcional, que contiene un landing page de negocio. La documentación del proyecto conlleva una documentación completa de las distintas fases que se han llevado a cabo en el desarrollo de la aplicación web. Esta documentación supondrá la memoria del proyecto.

Además, se aporta una presentación explicativa del desarrollo del proyecto, y un video explicativa del funcionamiento de la aplicación.

1.5 Estado de Arte

Investigación de estudios y proyectos similares

- <https://ludonix.ch/es-ES/juegos/juegos-de-mesa/89>
- <https://www.kidalos.com/>
- <https://www.apli.com/es/juegos/juegos-de-mesa>
- <https://www.bigbuy.eu/>

1.6 Público objetivo

La aplicación está diseñada para ser utilizada dentro del ámbito nacional. Por tanto no es multiidioma, aunque internamente se ha definido el texto prácticamente en su totalidad, con idioma inglés, empleado archivos de configuración de idioma “es.json” para su traducción a español.

El dominio que se puede utilizar para el despliegue de la aplicación puede ser *chargame.es* o *chargame.com* (dominios disponibles actualmente). Por tanto, y puesto que se trata de una herramienta de ayuda que permita contactar con gente para realizar un trueque de productos propios, el **público objetivo** está enfocado en personas que se encuentren próximos dentro de un ámbito geográfico.

2 Tecnologías

2.1 Metodologías de desarrollo

Metodologías híbridas

Actualmente existen muchas metodologías para el desarrollo de software que se agrupan en *metodologías tradicionales* y *metodologías ágiles*.

Las metodologías híbridas incorporan algunas prácticas existentes dentro de las metodologías tradicionales y ágiles. El objetivo es obtener lo mejor de los dos métodos.

Una de las principales ventajas que proporcionan las metodologías de desarrollo, es facilitar el trabajo colaborativo entre un equipo de trabajo, en el que cada miembro, desempeña un rol concreto. En el proyecto que nos ocupa, esta característica se difunina en cuanto que se desarrolla de forma individual.

AGILE

Agile o Manifiesto Agile, es un modelo metodológico que permite mejorar la planificación de proyectos y producción de resultados con la finalidad de evitar la pérdida de tiempo y recursos en las tareas asignadas.

Este método ayuda a mantener la orientación en las directrices del proyecto sin ser tan rígido como otros métodos tradicionales tipo Cascada.

Al tener la capacidad de tomar a tiempo las mejores opciones sin comprometer la integridad del proyecto se incrementa la productividad del mismo, apuntando correctamente al objetivo final.

Entre algunas ventajas de Agile se destacan:

- Mejora la calidad del producto final al minimizar errores en los procesos.
- Genera satisfacción del empleado y conciencia de equipo.
- Permite acortar los ciclos de producción minimizando los tiempos.
- Posee mejor asignación de recursos incentivando la producción.

SCRUM

SCRUM es un modelo que integra un conjunto de prácticas y roles en un marco de trabajo o Framework, donde el desarrollador busca un único producto o servicio en un periodo de tiempo determinado.

La versatilidad de SCRUM permite que los proyectos desarrollados con esta metodología sean más adaptables, interactivos, rápidos, flexibles y eficaces.

Una característica de esta metodología es la división de sus tareas y roles muy bien estructuradas y optimizadas.

- Product Owner/Dueño del producto. Es el responsable del proyecto en cuanto al desarrollo, mantenimiento y priorización de las tareas.
- El Scrum Master. Es la persona responsable de garantizar que el trabajo del equipo se realice según las bases de Scrum.
- Development Team Members/Miembros del Equipo de desarrollo. Son los responsables del desarrollo de cada una de las tareas

2.2 Herramientas de desarrollo

Las herramientas utilizadas para el desarrollo de la aplicación se clasifican en los siguientes apartados:

2.2.1 Herramientas de BackEnd

Las herramientas empleadas para el desarrollo de BackEnd 1. NPM 2. Vite 3. PhpStorm

2.2.2 Herramientas de FrontEnd

Las herramientas empleadas para el desarrollo de FrontEnd son:

1. JQuery
2. Tailwind CSS
3. Bootstrap

2.2.3 Plugins y utilidades para la aplicación de desarrollo

2.2.3.1 Buscador de paquetes

- <https://packagist.org/>
- <https://packalyst.com/>

2.2.3.2 Generador de contenido

- <https://www.mockaroo.com/> Generador de datos
- <https://testingdatagenerator.com/> Testing generator
- <https://unsplash.com/> Generador de imágenes

2.2.4 Paquetes para el desarrollo de la aplicación

- Cargar Archivos Bootstrap File Input
- Editor WYSIWYG Trumbowyg
- Métricas de ejecución Clockwork
- Generador de código en PDF TCPDF
- Generador de gráficos CHART.js

2.2.5 Herramientas gráficas

1. Gimp
2. Fontawesome

2.2.6 Herramientas de gestión de proyecto

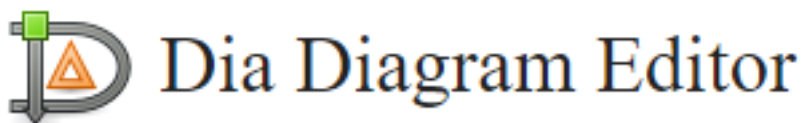
1. Gantt



2. Visualparadigm



3. Dia



3 Análisis

3.1 Descripción funcional de la aplicación

Existen dos roles, un rol de administración y de usuario. De un usuario se tiene la siguiente información: id, email, nickname, nombre, apellidos, teléfono, país, código postal, imagen. Un usuario puede ofrecer uno o muchos juegos. Cuando se ofrece un juego se almacena la fecha y hora.

Del juego se almacenará el código, nombre, descripción, año, estado, edad recomendada, tipo, idioma, número de jugadores, tiempo juego, dificultad, marca. Un juego puede ser al menos de un tipo (categoría). Cada juego es de una marca. De un juego se almacena al menos 1 imagen y máximo 3.

Además, un juego debe tener almacenado, al menos una categoría (tipo). De las categorías se almacena el código. Todos los juguetes disponibles pueden ser vistos por todos, a través de filtros por categorías y atributos. También se puede consultar por localidad extraída del código postal del usuario. No dirección exacta por privacidad.

Todos los juegos disponibles son visibles por todos. Hasta que el usuario no suba un juego no puede optar a cambiar un juego ni mostrar interés por otro. Un usuario puede estar interesado en uno o más juegos (distintos al suyo). Cuando se interesa por un juego, el usuario tiene que haberse identificado y por tanto dato de alta.

Un usuario puede consultar el listado de los juegos de los que tiene interés.

Un usuario puede consultar el listado de sus juegos. En el listado se muestra si alguien tiene interés. Cuando alguien está interesado en un juego de un usuario, ese usuario puede consultar la lista de interesados y por ende sus productos.

Si hay interés mutuo, es decir, un usuario está interesado en un juego de un usuario que tiene interés en un juego del primero, se hace acuerdo.

El acuerdo retira de la lista de disponibles, los productos acordados. De los acuerdos, se guarda producto propio, producto ajeno y código de interesado. Además, permite activar el chat. El acuerdo se puede cerrar o desestimar. Ambas partes deben cerrar el acuerdo, con que desestime uno se rompe el acuerdo. Si no hay acuerdo, los productos vuelven a la lista de juegos disponibles, aunque siguen en lista de interés, el usuario decide si quiere mantener el interés. Se pueden consultar la lista de acuerdos, filtrando por cerrados o pendientes. También se pueden archivar los acuerdos cerrados. Si se archiva un acuerdo, se oculta y se muestra en un listado aparte (acuerdos archivados).

El administrador crea campañas, cada campaña tendrá una duración determinada, una campaña puede ser para recogida de dinero y/o recogida de juguetes. De la campaña se almacena el código, un nombre, descripción, fecha y fin, tipo de campaña (recogida de juguetes y/o dinero). Un usuario puede donar un o más juguetes en una campaña.

Una campaña de recogida puede tener asociado uno o más juguetes de la lista de disponibles, un juguete no puede estar en más de una campaña. El usuario decide si quiere donar un juguete de la lista, cuando se marca para donar, se quita de la lista de disponible.

La campaña de recogida incluye o no servicio de recogida. Una campaña de recogida de dinero se hará mediante aportación económica en una fecha de usuarios, mediante una forma de pago (bizum, tarjeta, transferencia) un importe. Un usuario puede hacer más de una aportación para una campaña de recogida de dinero y una campaña puede tener aportaciones de múltiples usuarios. No se requiere el almacenamiento de la tarjeta de crédito.

3.2 Diagramas UML

3.2.1 Diagrama de contexto

Un Diagrama de Contexto de Sistema en Ingeniería de software e Ingeniería de sistemas es un diagrama que define los límites entre el sistema, o parte del sistema, y su ambiente, mostrando las entidades que interactúan con él.

3.2.2 Casos de uso

Casos de uso General

4 Diseño

4.1 Modelo Entidad/Relación (E/R)

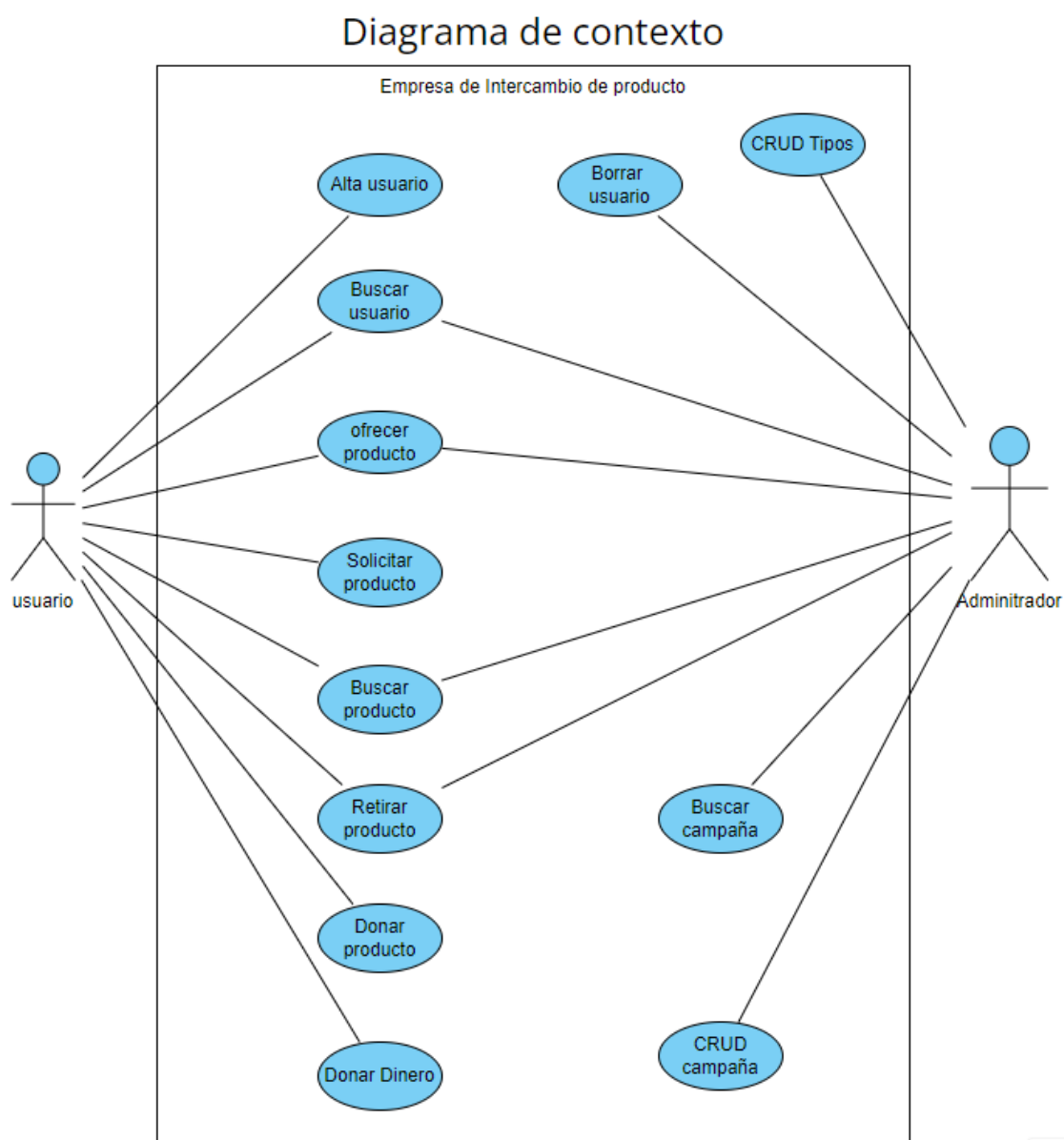
Representa el modelo de datos de forma independiente del DBMS que se utilizará.

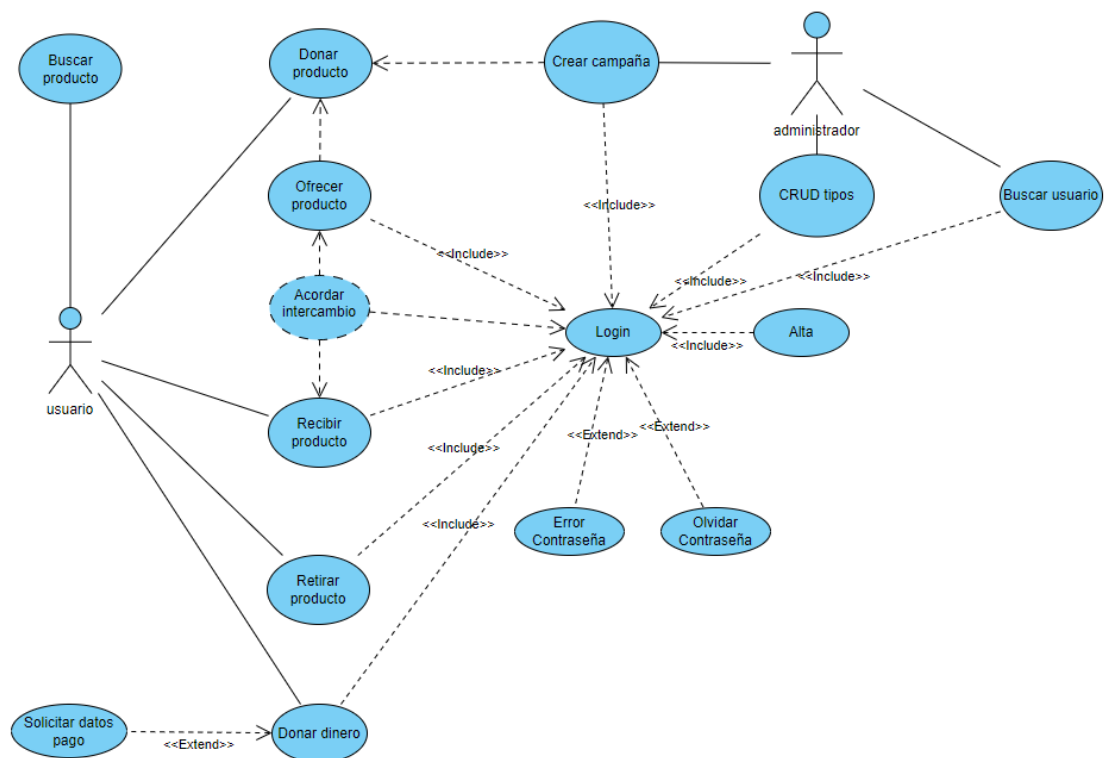
4.2 Modelo Relacional

Esquema canónico (o de base de datos). Representa los datos en un formato más cercano al del ordenador (Modelo lógico)

4.3 Patrón de arquitectura MVC (Modelo-Vista-Controlador)

Modelo-vista-controlador (MVC) es un patrón de arquitectura de software, que separa los datos y principalmente lo que es la lógica de negocio de una aplicación de su representación y el módulo

**Figura 2:** Diagrama de contexto

**Figura 3:** Casos de uso

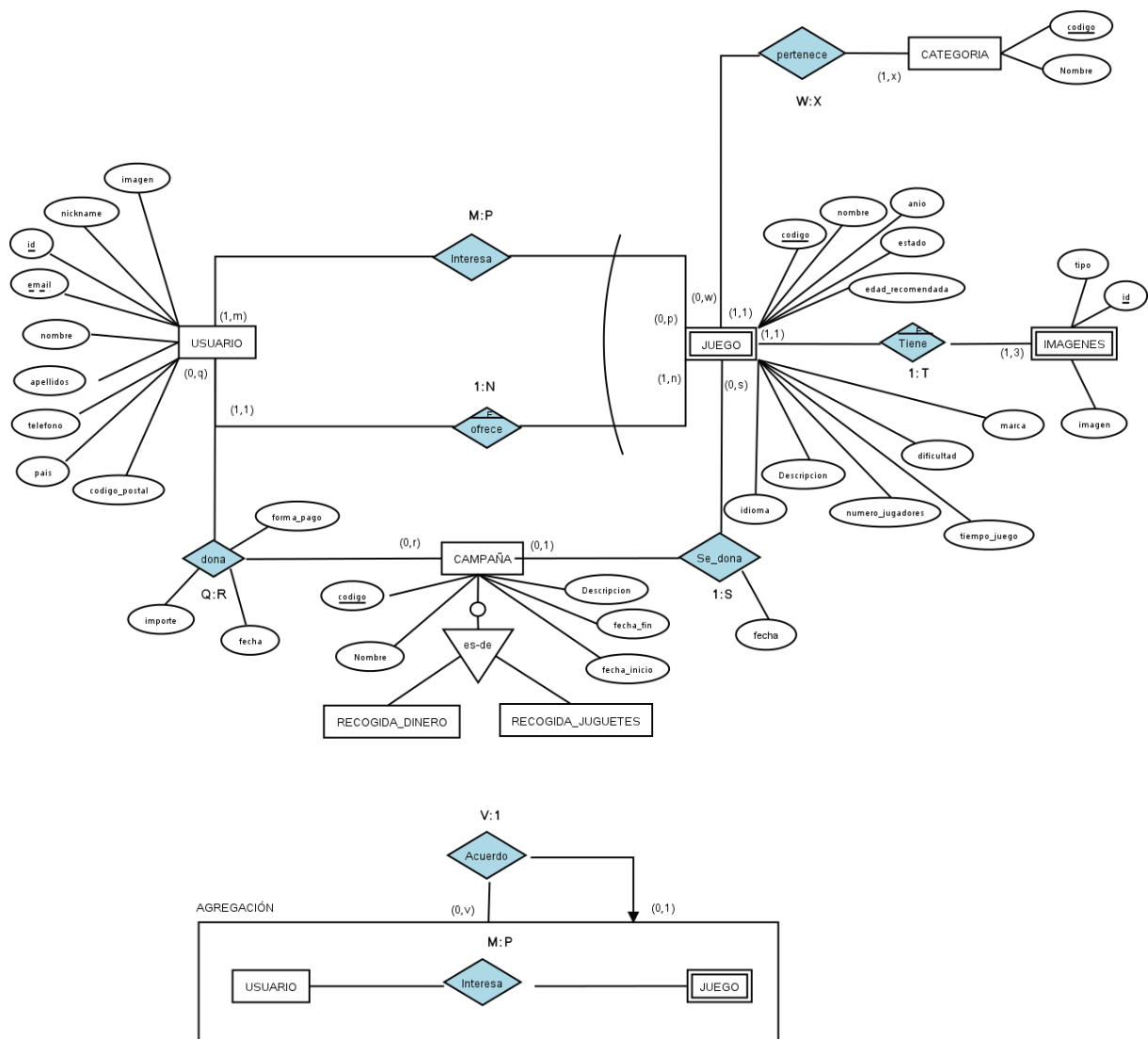
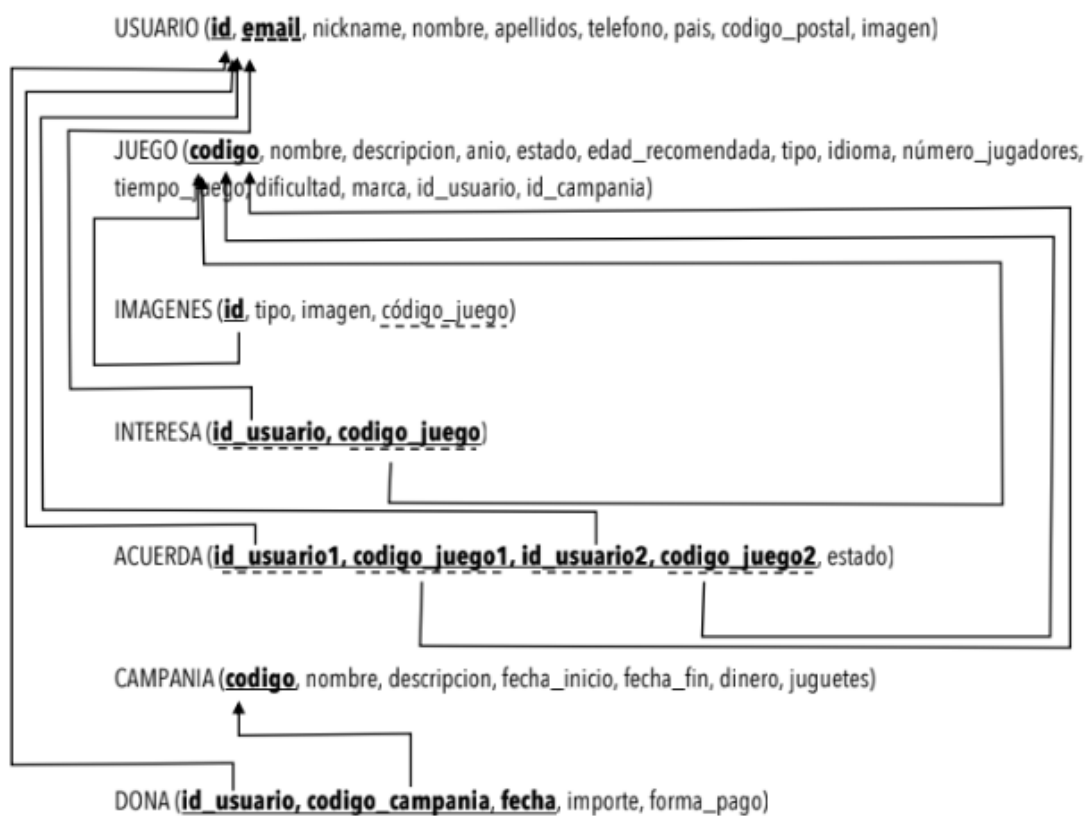


Figura 4: Modelo Entidad-Relación

**Figura 5:** Modelos relacional

encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador; es decir: por un lado define componentes para la representación de la información y, por otro lado, para la interacción del usuario.

Por tanto se ha definido una estructura de contenidos basado en dicho modelo, donde se establece la siguiente estructura de carpetas:

1. Vista
2. Modelo
3. Controllador
4. Interfaz
5. Factory

4.4 Componentes de Interfaz

Para la interfaz de la aplicación se han tenido en cuenta los siguientes elementos de diseño.

4.4.1 Elementos de Identificación

Los elementos de identificación se obtienen a partir de los elementos de navegación, o por acceso a contenido que requiere la autenticación del usuario. En este caso se accede a una nueva venta de verificación de usuario. Además, este debe permitir la recuperación/actualización de la contraseña en el caso de no ser recordada. Se añade a este elemento, la posibilidad de acceso a través de una cuenta usuario de Google+.

4.4.2 Elementos de Navegación

Los elementos de navegación son los que nos permiten acceder a todos los contenidos que se encuentran en las diferentes páginas de un sitio Web. Pero, lo que no hemos dicho, es que si queremos que nuestra página sea usable, el usuario debe conseguir navegar por la página sin perderse y sin tener la sensación de perderse. Para conseguirlo, el sistema de navegación debería constar de una serie de elementos:

- Elemento de regreso a la portada.
- Menú de secciones y/o áreas de interés.
- Información sobre la ubicación del usuario dentro del sitio Web.

[Aportar](#)[Catálogo](#)[Iniciar sesión](#)

Figura 6: Menú de navegación

Es importante para el usuario tener algún elemento que le permita volver al principio sin necesidad de utilizar la herramienta “ir hacia atrás o regresar” del navegador. Este problema suele resolverse empleando un enlace en el **logotipo de la empresa** que se sitúa normalmente en parte superior izquierda de cada una de las páginas que componen el sitio Web.

4.4.3 Elementos de Contenidos e Interacción

El contenido es la parte esencial de una página Web. Es importante que los contenidos estén expresados en un lenguaje claro y conciso y, presentados en un formato agradable y de fácil lectura. Además, si el sitio Web está formado por muchas páginas, el contenido debe situarse siempre en la misma ubicación. También es importante evitar que el usuario tenga que hacer grandes desplazamientos durante la lectura de los contenidos. Siempre es mejor dividir el contenido en más de una página y enlazar unas con otras.

la zona de contenido que a su vez se divide en dos zonas enmarcadas con elipses: el Título del contenido y el Contenido propiamente dicho.

4.5 Guías de estilo

4.5.1 Logotipo

4.5.2 Favicon

4.5.3 Tipografía

Google Fonts RaelWay

4.5.4 Colores

Con el fin de proporcionar contraste en la visualización de los elementos, se ha optado por utilizar colores complementarios, análogos y monocromáticos. Paleta de colores empleado en los distintos elementos de la página web se puede ver en la siguiente imagen.



Figura 7: Logotipo



Figura 8: Favicon

<p><i>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</i></p> <p>ExtraLight 200</p> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>	<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p> <p>Regular 400 Italic</p> <p><i>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</i></p>
<p>ExtraLight 200 Italic</p> <p><i>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</i></p>	<p>Medium 500</p> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
<p>Light 300</p> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>	<p>Medium 500 Italic</p> <p><i>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</i></p>
<p>Light 300 Italic</p> <p><i>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</i></p>	<p>SemiBold 600</p> <p><i>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</i></p>

Figura 9: Fuede Raleway



Figura 10: Paleta de colores

4.5.5 Botones

4.5.6 Input

Nombre*

Figura 11: Input

4.5.7 iconos utilizados



Figura 12: Iconos

5 Implementación

5.1 Definición de filtrados

Uno de los problemas que nos hemos encontrado es determinar los campos por los que se quiere establece una consulta de productos. Esta dificultad se acentúa en el hecho de optar por un búsqueda de datos a medida que se introducen valores en los campos. Se podría haber hecho con alternativas (javascript, jquery,..) Sin embargo, se opta por establecer un formulario individual para cada campo, de forma si se ha realizado alguna búsqueda por algún otro campo previamente, este se introduce com campo oculta para volver a ser en viado via *request*. Se ha definido, por tanto, un componente visual denominado `<x-form.input-search>` al que se le pasa el nombre campo, el tipo, la doniminación y el array de campos a filtrar (viene desde el **contolador ProductController**)

```
1 $filters = ['search', 'category', 'owner', 'from_age', 'play_time', 'release_year'];
2 return view( 'products.index', [
3     'filters' => $filters,
4     'products' => Product::latest()->filter(
```

```
5         request($filters)
6     )->paginate(8)->withQueryString()
7 ];
```

método index de clase ProductController.php

Las consultas de filtrado se hacen a través del **modelo Product**

```
1 $query->when($filters['from_age'] ?? false, fn($query, $from_age)
2     => $query->where(fn($query)
3     => $query->where('from_age', '>=', $from_age)
4     )
5 );
```

Por último tenemos es el componente que se incorpora dentro la **vista Index** de consulta de productos.

```
1 @props(['name', 'type', 'text', 'filters'])
2
3 <form method="GET" action="/products/{ {{ http_build_query(request()->
4     except('category', 'page')) }} ">
5     @foreach ($filters as $filter)
6         @if (request($filter) && $filter != $name)
7             <input type="hidden" name="{{ $filter }}" value="{{ request(
8                 $filter) }}" />
9         @endif
10    @endforeach
11    <div class="relative lg:inline-flex items-center rounded-xl gap-2">
12        <x-form.text-input id="{{ $name }}" class="block mt-1 w-full"
13            type="{{ $type }}" name="{{ $name }}"
14            autofocus autocomplete="{{ $name }}"
15            placeholder=" {{ __( $text) }}"
16            value="{{ request($name) }}" />
17    </div>
18 </form>
```

Componente de vista input-search

5.2 Definición de datos Enum

5.3 Control de Autorización de acceso

5.4 Control de incluir en Favoritos

5.5 Control de incluir en Acuerdos

5.6 Control de aportaciones en Campañas

5.7 Instalación de API MailChimp para Newsletter

5.8 Aplicación de paquetes

5.8.1 Paquete FileInput

5.8.2 Paquete DataTables

5.8.3 Paquete Pagination

6 Conclusiones

Cosas que hemos aprendido nuevas, cosas que haríamos de otra manera con lo que sabemos, qué haremos después...

7 Dificultades encontradas

Planteamos dificultades que nos hemos ido encontrando durante el desarrollo del aplicativo.

7.1 Instalación de librería JQuery

La instalación y configuración de JQuery se ha realizado conjuntamente con la bootstrap. Teniendo en cuenta que Laravel se ha instalando sobre la agrupación de código de Vite, esta configuración conlleva realizar los siguientes pasos.

```
1 // Configuración de resources\js\app.js
2
3 import jQuery from 'jquery';
4 window.jQuery = window.$ = jQuery;
```

```
5
6 import './bootstrap';
7 import * as bootstrap from 'bootstrap';
```

```
1 // Configuración de resources\sass\app.scss
2
3 @import "bootstrap/scss/bootstrap";
```

la llamada a la configuración se realiza a través de la sentencia

```
1 @vite(['resources/css/app.css', 'resources/js/app.js'])
```

El problema radica en que cuando se hace una llamada en un archivo público de javascript, el sistema no reconoce la llamada a la función

```
1 $(document).ready(function (event) {
2     // código javascript jquery
3 });
```

Para solucionar el problema se requiere que la llamada a script propio dentro del código se debe hacer como un módulo, de forma que el render de vite lo interprete en el orden correcto.

```
1 <script type="module" src="/js/app.js"></script>
```

7.2 Modales de Bootstrap en paquetes instalados

7.3 Filtrado en las búsqueda de productos

Multitud de campos para filtrar campos.

8 Bibliografía

- Objetivos de desarrollo sostenible (ODS) <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>
- Documentación Tailwind CSS <https://tailwindcss.com/docs/installation>
- The PHP Framework for Web Artisan <https://laravel.com/docs/11.x>
- Metodología Híbrida para Desarrollo de Software https://www.iiis.org/CDs2012/CD2012IMC/CIC_IC_2012/PapersPdf/CB153YB.pdf