

Memoria Proyecto

DEMO de Gestor de contenido
Tienda ONLINE de cartas TCG

Resumen	5
Abstract	7
1. Introducción	8
1.1 Justificación del proyecto	9
1.2 Caso de estudio	9
1.3 Estructura del documento	10
1.4 Descripción general	11
1.5 Objetivos	12
1.6 Requisitos	13
2. Análisis	14
2.1 Casos de uso	14
2.2 Diagrama de clases	16
2.3 Diagramas de secuencia	17
3. Diseño	20
3.1 Capa presentación	21
3.1.1 Login	22
3.1.2 Navegador	23
3.1.3 Sección Lista	24
3.1.4 Carta	25
3.1.4 Sección Admin	26
3.2 Capa Lógica	28
3.3 Capa de datos	31
4. Implementación	32
4.1 Tecnologías	32
4.1.1 JAVA	32
4.1.2 HTML	33
4.1.3 CSS	34
4.1.4 JAVASCRIPT	35
4.1.5 SQL	36

4.2 Tecnologías	37
4.2.1 Tecnologías visuales	37
4.2.2 Hardware	37
4.3 Detalles de implementación	
4.3.1 Ejemplo del Funcionamiento del buscador del INDEX	
4.3.2 Ejemplo creación de una carta	
4.3.3 La clase Carta	
4.3.4 Conexión a la Base de datos	
4.3.5 JSON	
4.3.6 Funciones	

5. Plan de proyecto

5.1 Equipo de trabajo

5.2 Plan de trabajo

5.3 Análisis de costes

6. Trabajos futuros

7. Conclusiones

Anexos

Anexo.1 Recursos utilizados

Anexo.2 Bibliografía

RESUMEN

Nuestra tienda online de cartas de Digimon cuenta con un gestor totalmente editable a través de menús y formularios, diseñados con las tecnologías web más actuales como HTML, JAVA, JAVASCRIPT y MySQL. Hemos priorizado el dinamismo en su contenido y personalización, asegurando así una experiencia única para nuestros usuarios. Además, nos hemos esforzado por posicionar el portal de manera óptima en internet.

Buscamos que nuestro sitio sea escalable, permitiendo la fácil inserción de nuevos objetivos y elementos. Aunque en este caso presentamos una tienda online de cartas como ejemplo, estamos comprometidos a adaptar nuestra plataforma para satisfacer las necesidades específicas de los amantes de Digimon. ¡Explora nuestro catálogo y únete a la aventura digital!

ABSTRACT

Our Digimon card online store features a fully editable manager through menus and forms, designed with the latest web technologies such as HTML, JAVA, JAVASCRIPT and MySQL. We have prioritized dynamism in its content and customization, ensuring a unique experience for our users. Additionally, we have made efforts to position the portal optimally on the internet.

We aim for our site to be scalable, allowing for easy insertion of new modules that may be created in the future. While in this case, we showcase a online cards shop as an example, we are committed to adapting our platform to meet the specific needs of Digimon enthusiasts. Explore our catalog and join the digital adventure!

1. Introducción

En el mundo actual del comercio electrónico, las empresas necesitan mantener una presencia en Internet sólida y actualizada. Aunque existen numerosas soluciones para establecer una tienda en línea, es común recurrir a plataformas de comercio electrónico como Shopify, CardMarket, entre otras. Estas plataformas ofrecen la facilidad de actualizar y gestionar productos mediante paneles de administración, pero en ocasiones requieren de cierta capacitación previa para su manejo eficiente.

En este proyecto, se propone el desarrollo de un gestor de tiendas en línea con requisitos específicos que no se encuentran completamente satisfechos por las plataformas existentes. Por un lado, se busca que el uso del gestor sea intuitivo y accesible incluso para usuarios sin experiencia previa en el manejo de tiendas en línea. Se pretende que la gestión de productos sea sencilla y directa, sin necesidad de una curva de aprendizaje extensa.

Además, se plantea la posibilidad de crear múltiples cartas de otras colecciones en línea con estructuras y diseños personalizados, alojadas en diferentes carpetas del servidor. Esto permitiría dirigir cada tienda mediante dominios individuales, proporcionando una experiencia única para cada producto o línea de productos. Cada producto podría tener su propia identidad visual y funcionalidades específicas, adaptadas a las necesidades de los clientes o productos que representan.

El gestor de tiendas en línea también contaría con una página principal que funcionaría como un catálogo centralizado, mostrando una lista de productos disponibles y un sistema de búsqueda para facilitar la navegación de los usuarios."

1.1 Justificación del proyecto.

Un proyecto de tienda en línea de venta de cartas de Digimon sería justificado por el creciente interés en la nostalgia de los años 90 y el resurgimiento de la franquicia Digimon. La popularidad de los juegos, series de televisión y películas recientes de Digimon ha creado una demanda renovada de productos relacionados, como las cartas coleccionables. Una tienda en línea dedicada a la venta de cartas de Digimon proporcionaría a los aficionados un acceso conveniente a una amplia variedad de cartas para coleccionar, intercambiar y jugar. Además, aprovecharía el crecimiento del mercado de coleccionistas y aficionados a los juegos de cartas en línea, ofreciendo una plataforma para comprar productos auténticos y de alta calidad. La tienda podría diferenciarse ofreciendo cartas raras, ediciones limitadas y paquetes temáticos exclusivos para atraer a clientes y fomentar la lealtad a la marca. Además, podría incluir contenido adicional como guías de juego, noticias sobre eventos y torneos, y tutoriales para ayudar a los clientes a sacar el máximo provecho de su experiencia con las cartas de Digimon. La tienda en línea también sería una oportunidad para crear una comunidad en línea donde los aficionados puedan compartir sus colecciones, intercambiar estrategias y conectarse con otros fans de Digimon de todo el mundo. En resumen, un proyecto de tienda en línea de venta de cartas de Digimon capitalizaría el interés creciente en la franquicia, proporcionando a los aficionados un destino centralizado y accesible para satisfacer su pasión por coleccionar y jugar con las cartas de Digimon.

1.2 Caso de estudio.

Después de detectar un resurgimiento en el interés por la franquicia Digimon debido al lanzamiento de una nueva serie de televisión y varios juegos populares, una empresa de comercio electrónico especializada en productos de entretenimiento decide lanzar una tienda en línea dedicada exclusivamente a la venta de cartas de Digimon. Utilizando técnicas de marketing digital dirigidas a la comunidad de fans existente y aprovechando las redes sociales para generar anticipación y promoción antes del lanzamiento, la tienda en línea se abre con

una amplia selección de cartas de Digimon, incluyendo raras y ediciones limitadas.

1.3 Estructura del documento.

Esta memoria se ha estructurado de la siguiente forma:

Introducción: En este segmento se detalla el proyecto y se explican en términos generales sus elementos, así como sus objetivos.

Análisis: Explicación de la estructuración del proyecto y sus capacidades, junto con los esquemas UML más destacados.

Diseño: Estratos de la arquitectura de la aplicación (interfaz, lógico y de almacenamiento en base de datos).

Ejecución: Las tecnologías aplicadas y la descripción de las herramientas utilizadas en la construcción.

Conclusiones: Reflexión desde una perspectiva personal acerca de la aplicación y su propósito.

Referencias: Listado de los recursos documentales consultados para la elaboración de este informe.

Apéndices: Datos adicionales para ampliar alguno de los aspectos tratados en los apartados previos.

Manual del usuario: Descripción sobre cómo emplear el gestor, con asistencia e ilustraciones para facilitar su comprensión.

1.4 Descripción general

La estructura de la tienda online se basa en una parte pública que es el front-end y una parte privada o administrativa que es el back-end.

El front-end es la parte de un sistema informático o sitio web que los usuarios ven y con la que interactúan directamente. Es la interfaz gráfica y funcionalidad que se muestra en el navegador web o en una aplicación, permitiendo a los usuarios realizar acciones y recibir información de manera visual.

El back-end es la parte de un sistema informático o sitio web que opera detrás de escena y no es visible para los usuarios. Se encarga de procesar datos, gestionar la lógica de la aplicación y comunicarse con la base de datos. En resumen, es el "cerebro" de la aplicación, maneja la lógica y la funcionalidad que permite que el front-end funcione correctamente.

Aquí van un par de fotos de la página para ilustrar

1.5 Objetivos

El objetivo de crear una tienda en línea de cartas es satisfacer la demanda de los clientes que buscan adquirir cartas de manera conveniente y accesible a través de Internet, creando modificado y borrando los diferentes elementos que tenemos a disposición.

- Creación de una interfaz fácil, sencilla y amigable.
- Dar visibilidad al producto de manera directa.
- Dar soporte a diferentes dispositivos.
- Facilitar la edición y manipulación de la información.

Aquí van un par de fotos de la página para ilustrar

1.6 Requisitos

1. Conexión a Internet: Se requiere acceso a una conexión de Internet estable para cargar y navegar por la página web de la tienda de cartas en línea.
2. Dispositivo compatible: Es necesario contar con un dispositivo compatible, como una computadora, tableta o smartphone, con un navegador web actualizado para acceder a la tienda en línea y visualizar su contenido.
3. Capacidad de navegación segura: Los usuarios deben tener la capacidad de navegar de manera segura en la página web de la tienda de cartas en línea, lo que implica contar con un navegador web que admita protocolos de seguridad como HTTPS y SSL para proteger la información personal y financiera.
4. Habilitar JavaScript: La funcionalidad de la tienda en línea puede depender del uso de JavaScript, por lo que los usuarios deben tener habilitado este lenguaje de programación en su navegador para una experiencia completa.
5. Método de pago: Para realizar compras en la tienda de cartas en línea, los usuarios necesitan contar con un método de pago válido, como tarjetas de crédito/débito, PayPal u otros sistemas de pago en línea aceptados por la tienda.

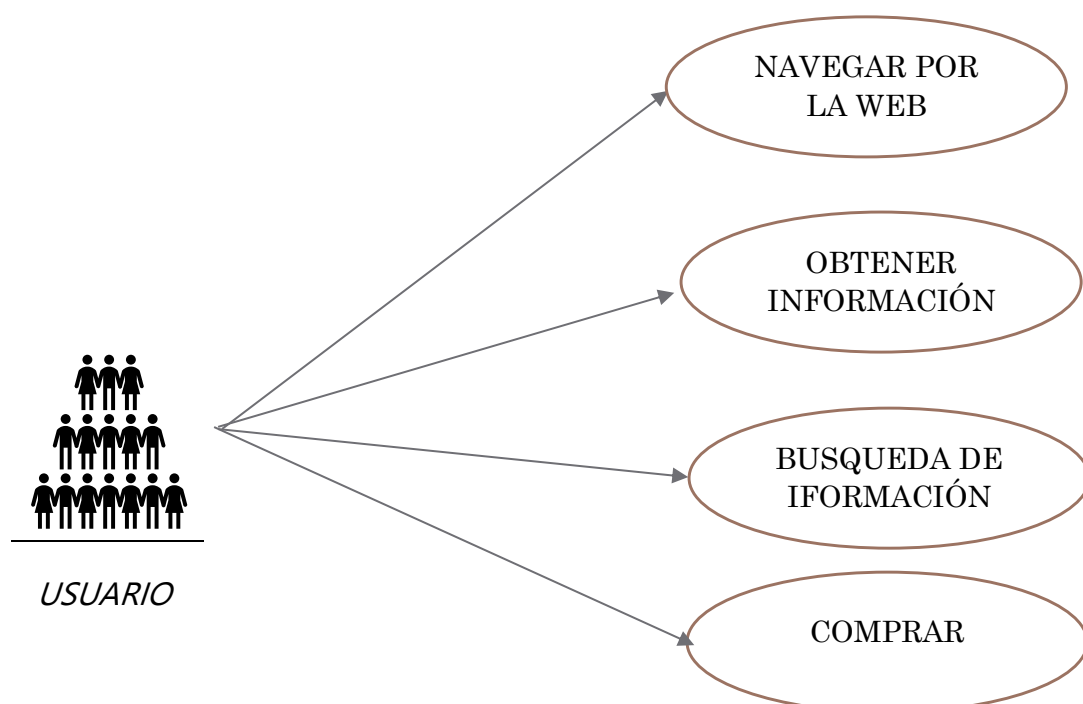
2. Análisis

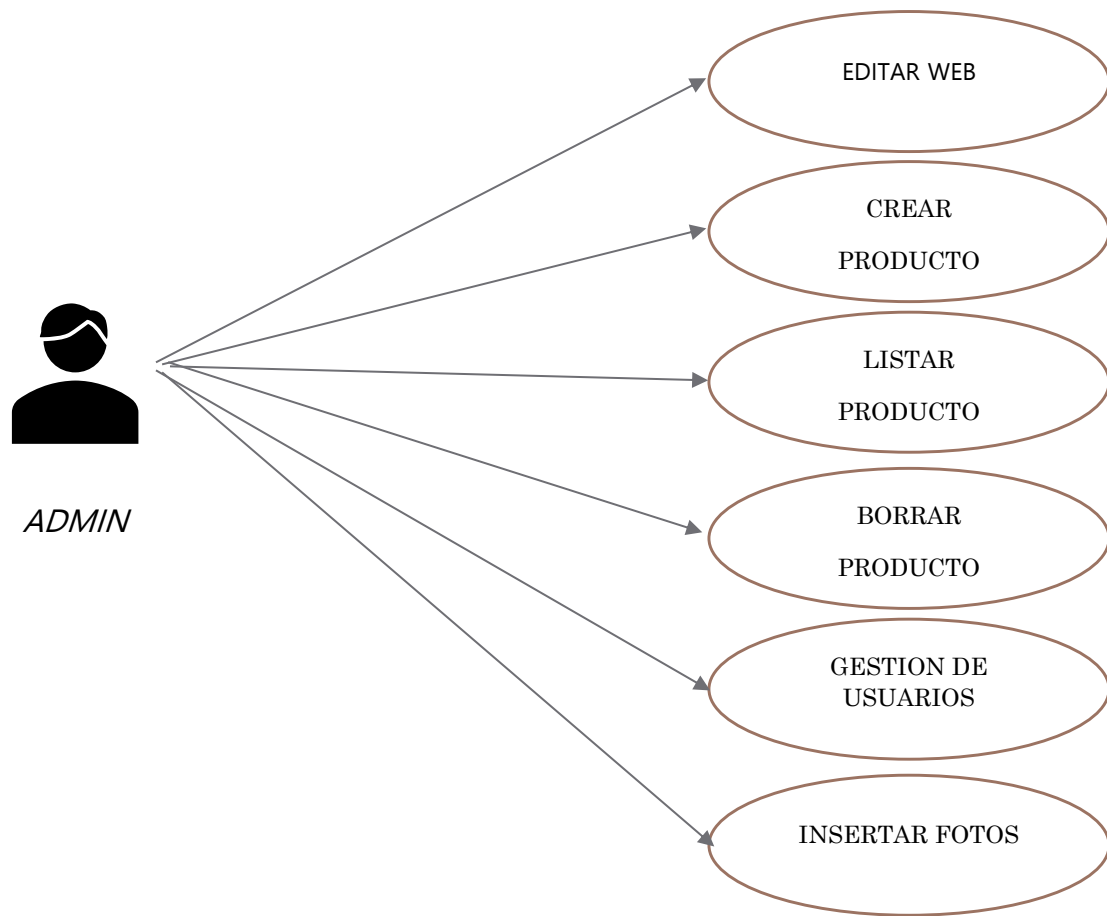
En este capítulo se comenta el análisis del proyecto con diagramas UML de los diferentes actores que interactúan con el proyecto y sus acciones con los elementos que veremos a continuación.

2.1 Casos de uso

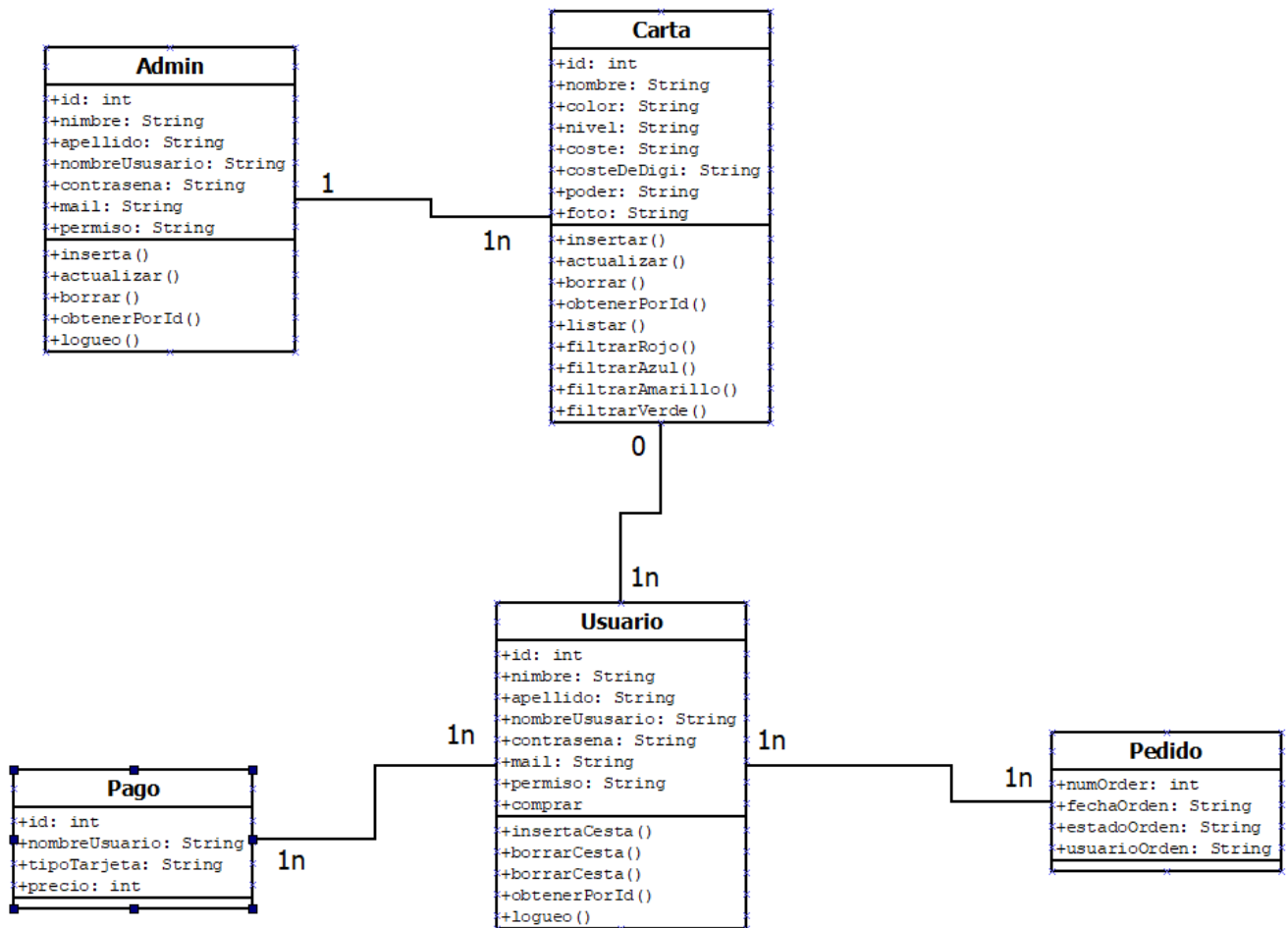
Un escenario de uso describe las acciones o actividades requeridas para ejecutar un determinado proceso. Los roles o entidades que intervienen en un escenario de uso se conocen como actores. En el ámbito de la ingeniería de software, un escenario de uso representa una serie de interacciones planificadas entre un sistema y sus actores, desencadenadas por un evento iniciado por un actor principal, y dirigidas al sistema en sí.

ACTOR ANÓNIMO NIVEL USUARIO



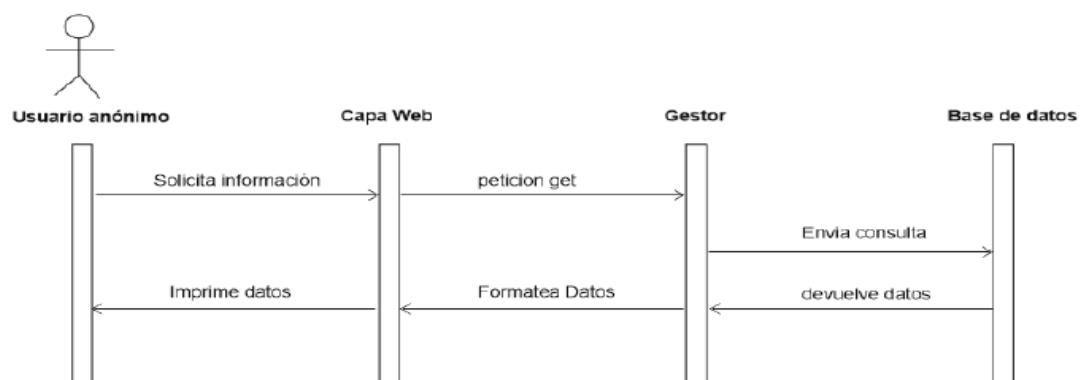


2.2 Diagrama de clases

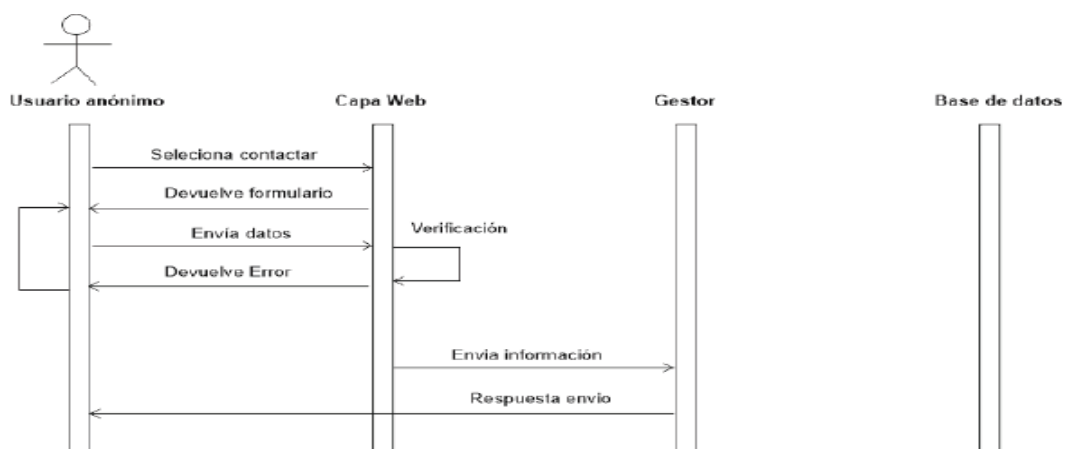


2.3 Diagrama de secuencia

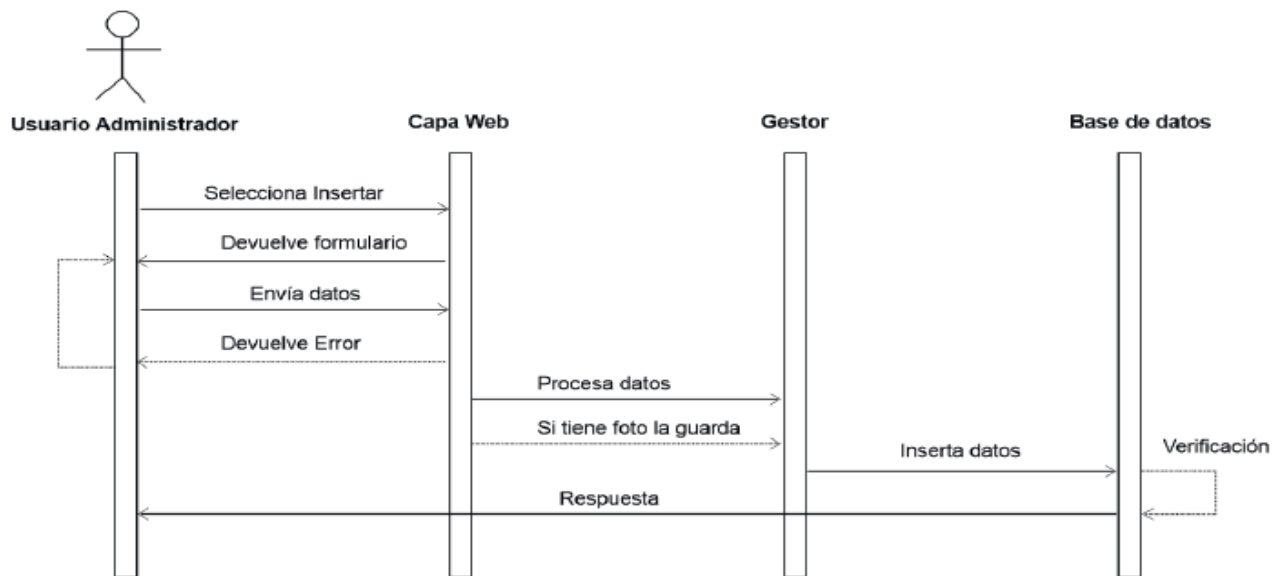
Un diagrama de secuencia es una herramienta de modelado en UML que muestra cómo los objetos interactúan en una secuencia de eventos a lo largo del tiempo. Representa la interacción entre diferentes partes de un sistema a través de mensajes enviados entre objetos. Los objetos se colocan en una línea de tiempo vertical y las flechas indican la dirección de los mensajes enviados entre ellos, mostrando el orden en que ocurren las interacciones. Es útil para visualizar el flujo de control entre objetos y entender el comportamiento dinámico de un sistema.



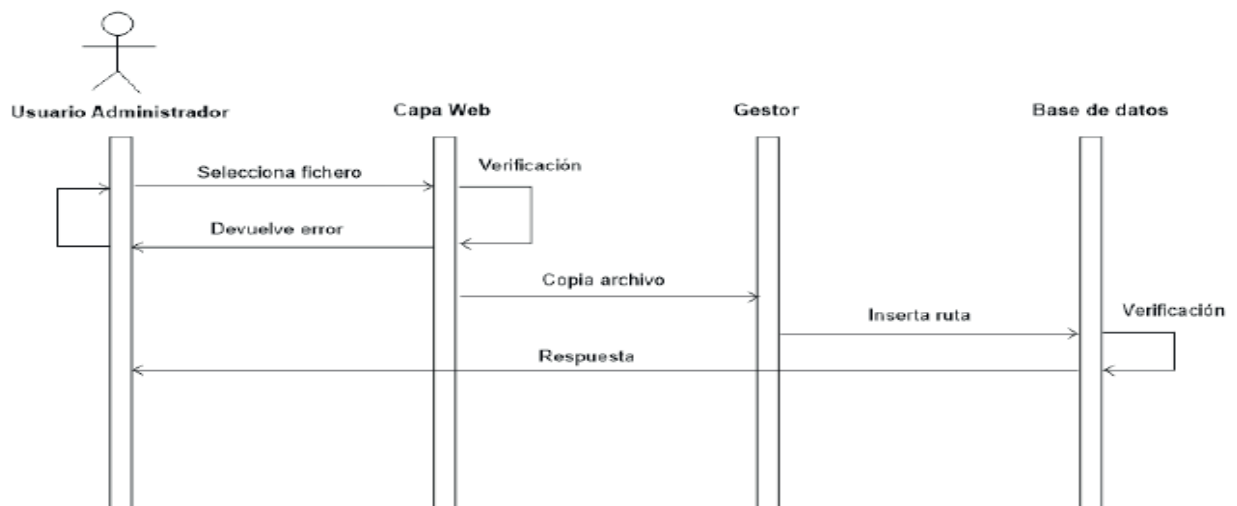
Visualizar contenido.



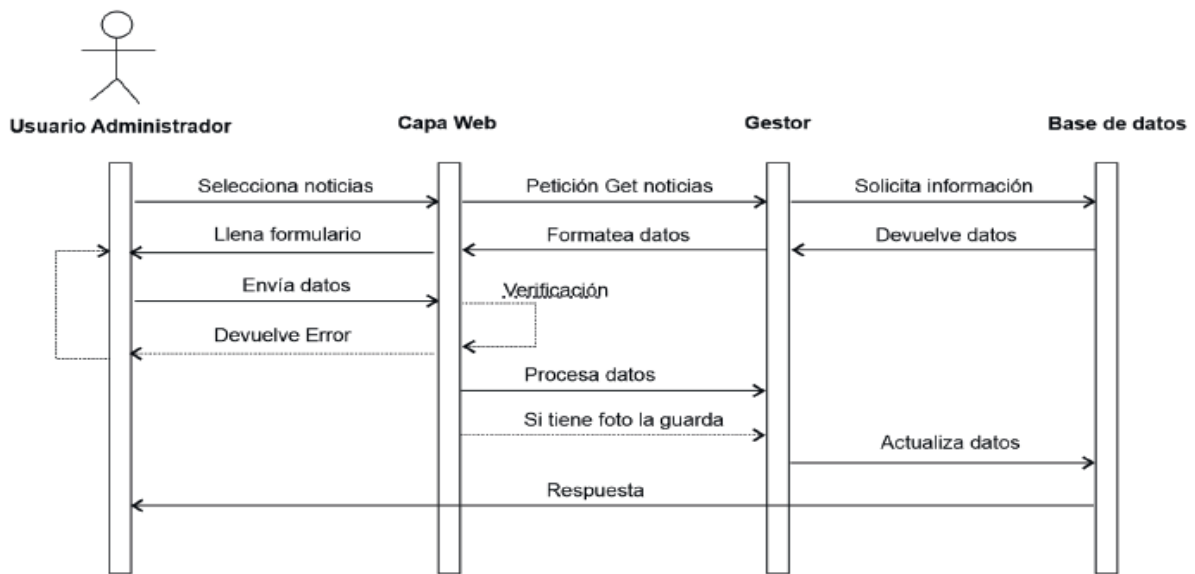
Solicitud de información.



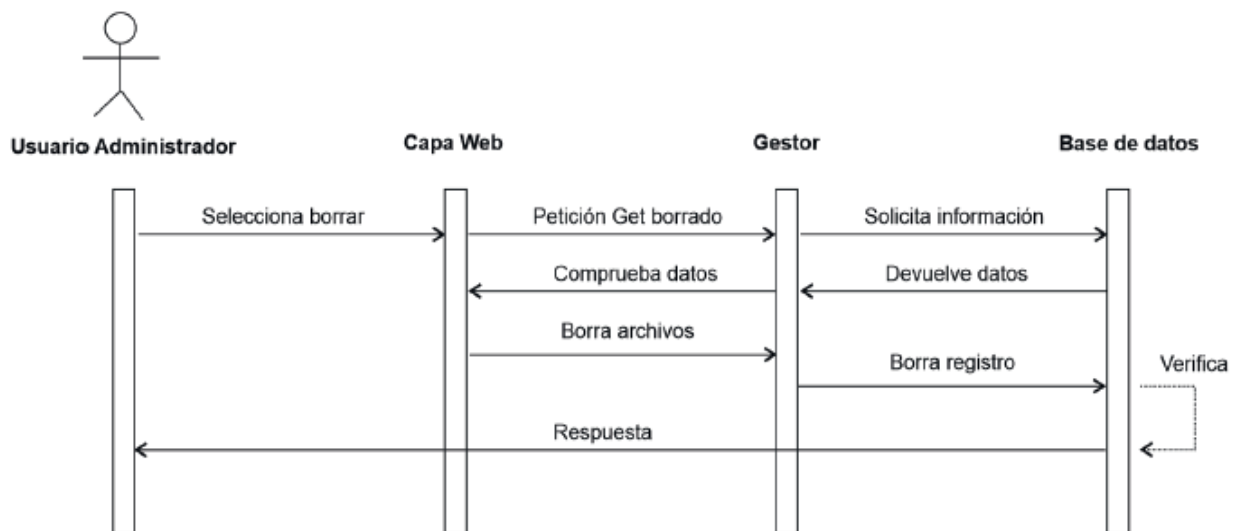
Insertar contenido.



Subir archivo foto.



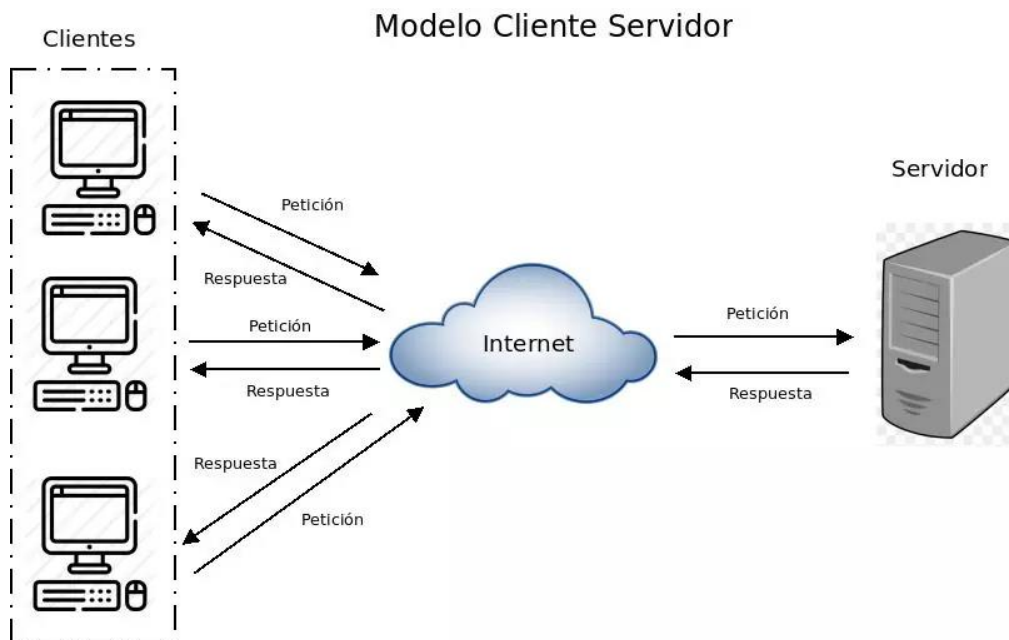
Actualizar datos.



Borrar datos.

3. Diseño

El portal utiliza un sistema donde hay una división clara de responsabilidades entre el servidor y los usuarios. El servidor provee los servicios y recursos solicitados por los usuarios, quienes son los clientes. Los usuarios, a través de sus terminales, hacen demandas al servidor, que las procesa y devuelve resultados o servicios según lo solicitado. Este modelo distribuido asegura una distribución eficiente de las tareas y recursos, permitiendo una interacción fluida entre los usuarios y el sistema.



La arquitectura de tres capas incluye un nivel inicial (Capa de presentación) que representa la parte visual y gráfica.

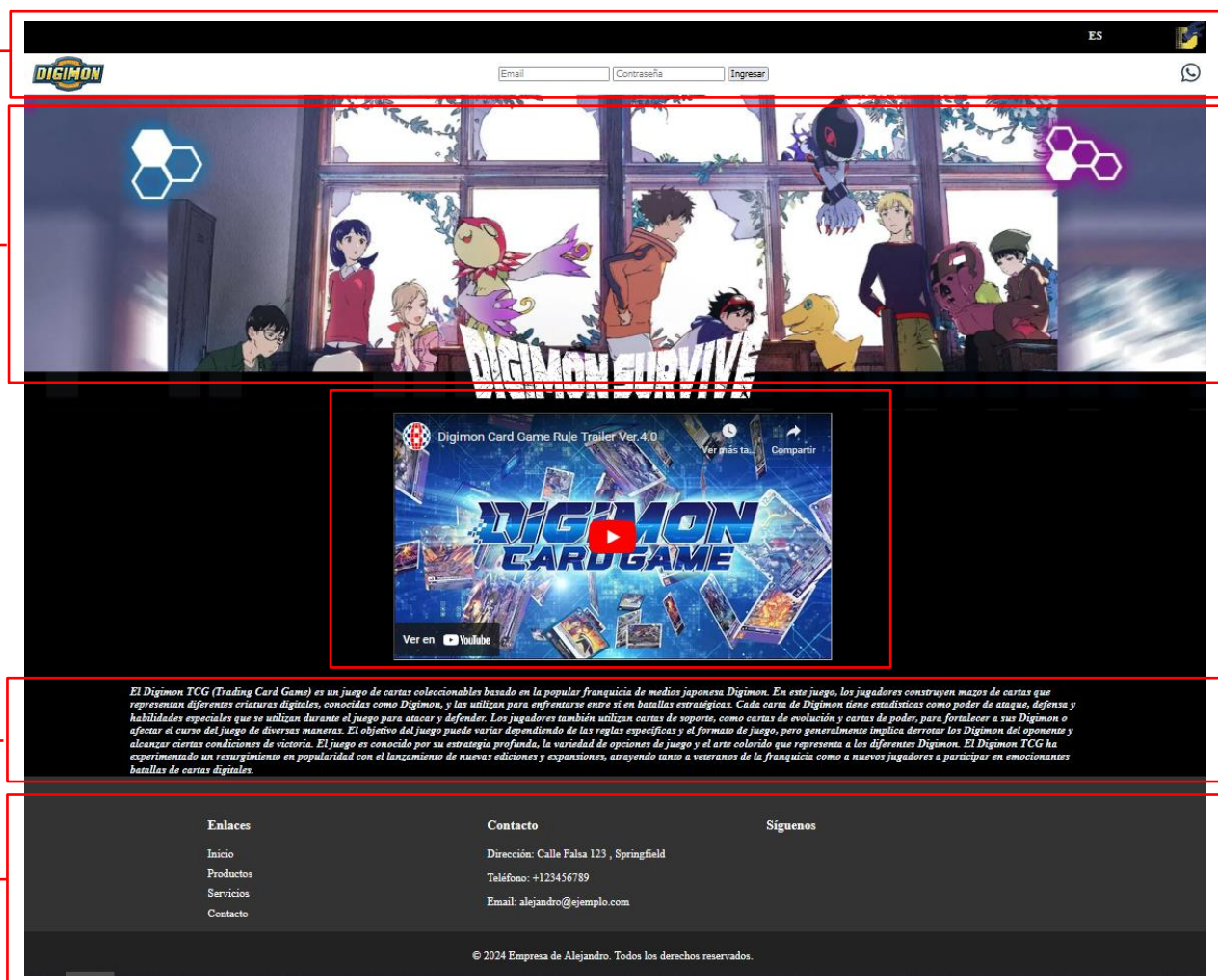
Un nivel intermedio (Capa lógica) que conecta al cliente, que solicita recursos a través de una interfaz de usuario o un navegador, con el servidor de datos. Este nivel intermedio, llamado software intermedio, facilita los recursos solicitados al cliente utilizando otro servidor.

El servidor de datos, la última capa (Capa de datos), suministra los datos necesarios para que el servidor de aplicaciones pueda procesar y proporcionar el servicio solicitado por el cliente.

3.1 Capa de presentación

La capa de presentación es la parte de un sistema o aplicación que se encarga de interactuar directamente con el usuario.

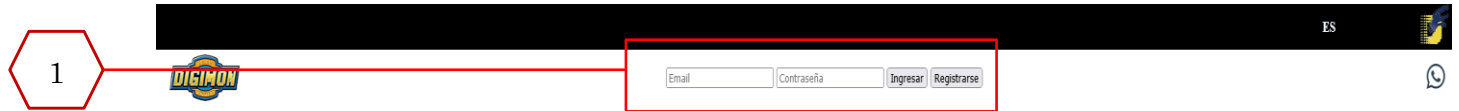
Mostrare capturas de pantalla de la interfaz d usuario y explicare las diferentes partes parte de la página según vaya navegando.



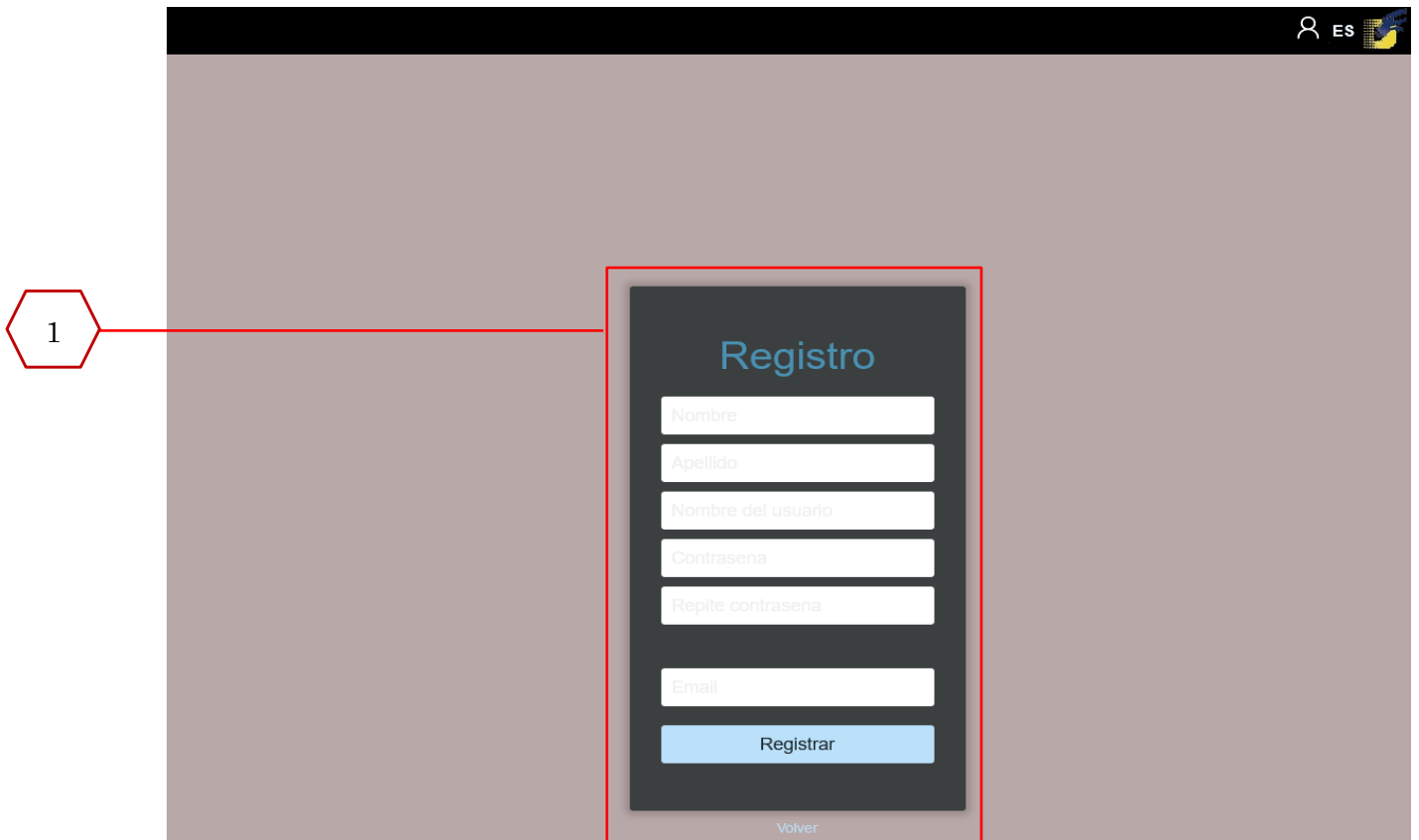
1. Navegador: En el que se mostrara el menú de navegación por el que usuario interactuara de forma rápida por la página.
2. Foto principal: Muestra una ilustración que va a caracterizar a la tienda online.
3. Video principal: Enlace de un video para explicar las mecánicas del juego de las cartas que se van a poder comprar.
4. Pie de página: Enlaces e información de la empresa o tienda física.

3.1.1 Login.

Lo primero que debería hacer el usuario sería insertar su Email y contraseña para tener permiso de ver las diferentes opciones de navegación.

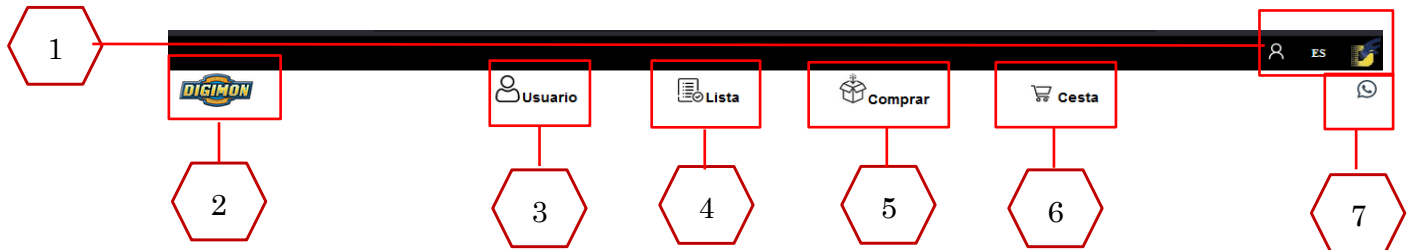


1. Registro (Login): El usuario tendrá la obligación de estar en la base de datos. Si no estuviera aún registrado, tendría que pulsar el botón de REGISTRARSE. Una vez pulsado pasaría a la siguiente web. Si el usuario tiene permiso de "Admin" se redirigiría a otra web personal.



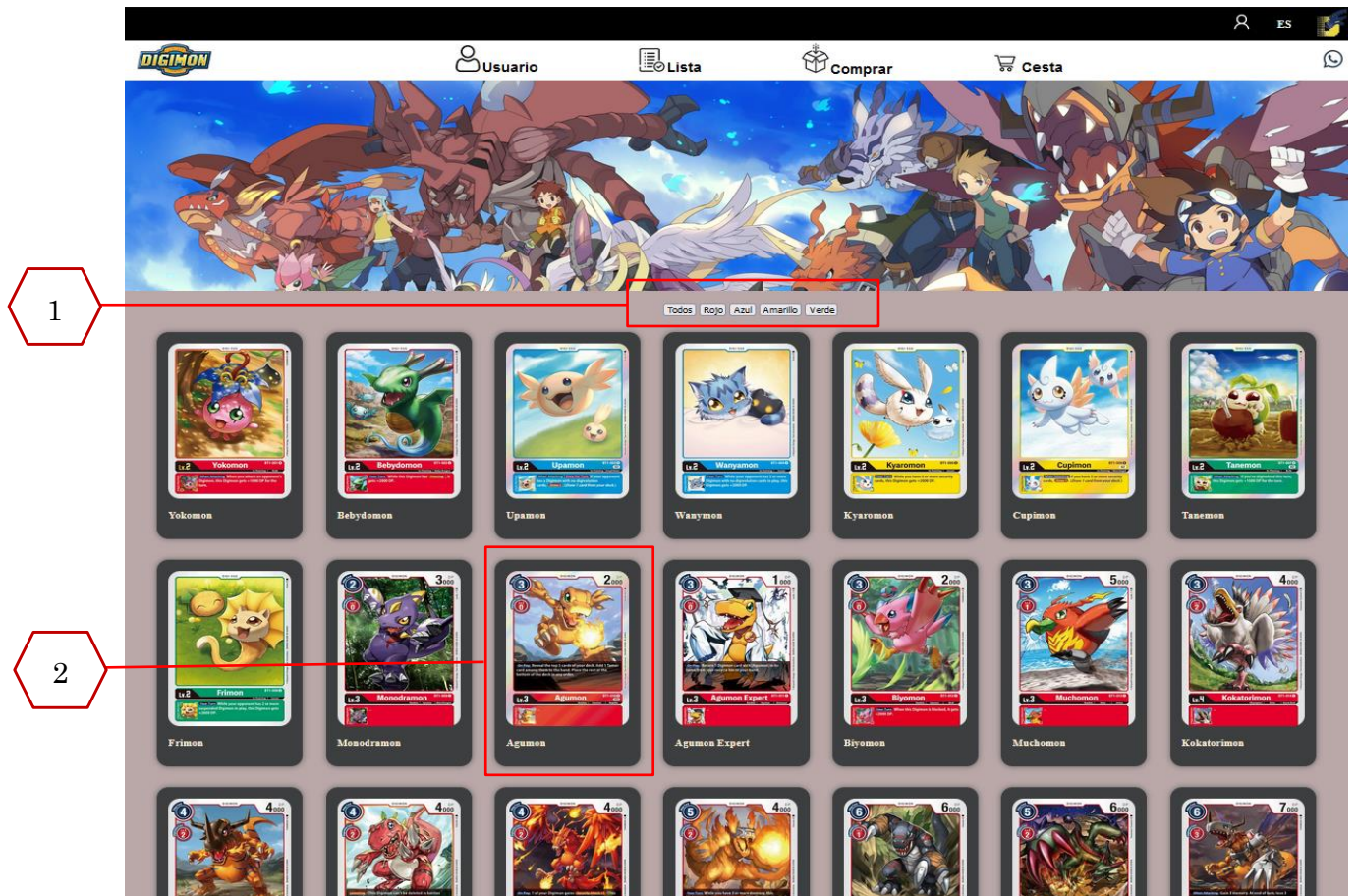
1. Nuevo Usuario: El usuario se registra en la base de datos y será redirigido a la página principal para ya ingresar mail y contraseña.

3.1.2 Navegador.



1. Índice de navegación: El usuario podrá cerrar sesión, cambiar el idioma e ir al índice de nuevo.
2. Icono principal: Dirige a la página principal.
3. Web usuario: El usuario podrá modificar sus datos personales.
4. Lista de cartas: Información de todas las cartas hasta el momento metidas en la base de datos.
5. Comprar: Lista de cartas en formato columna para ingresar la carta en la cesta del usuario.
6. Cesta: Una vez metidas las cartas que el usuario necesite, puedas comprarlas.
7. Social: Enlace a la aplicación de Telegram de la Tienda.

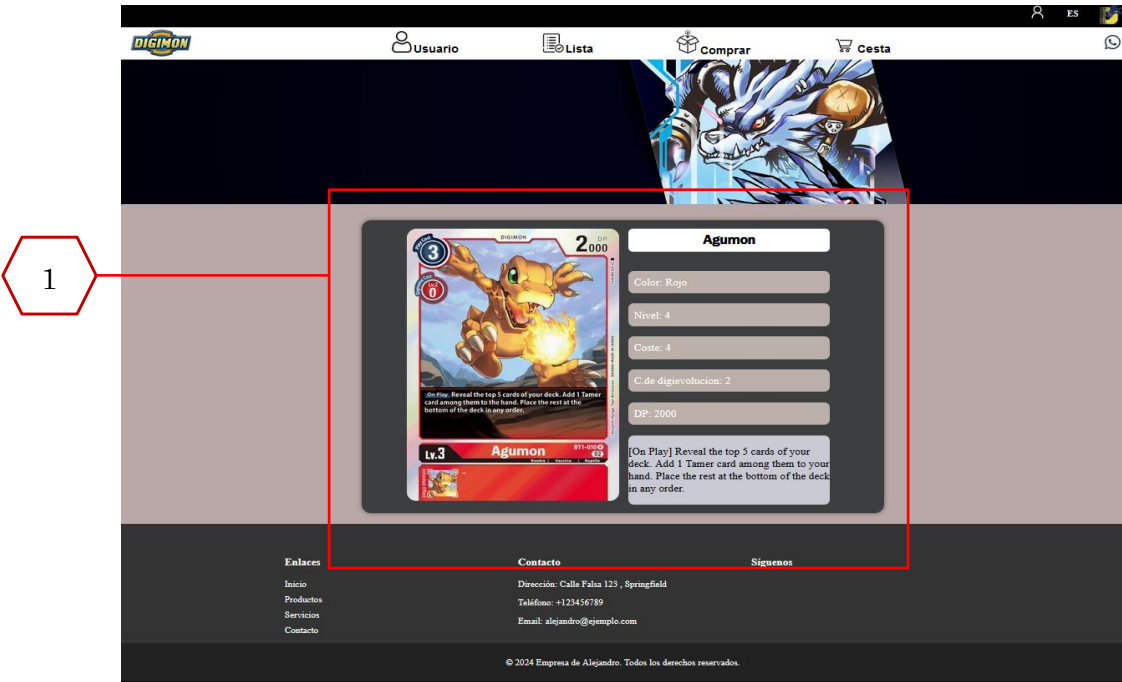
3.1.3 Sección Lista



1.Filtro: El usuario podrá filtrar entre las cartas según su atributo color.

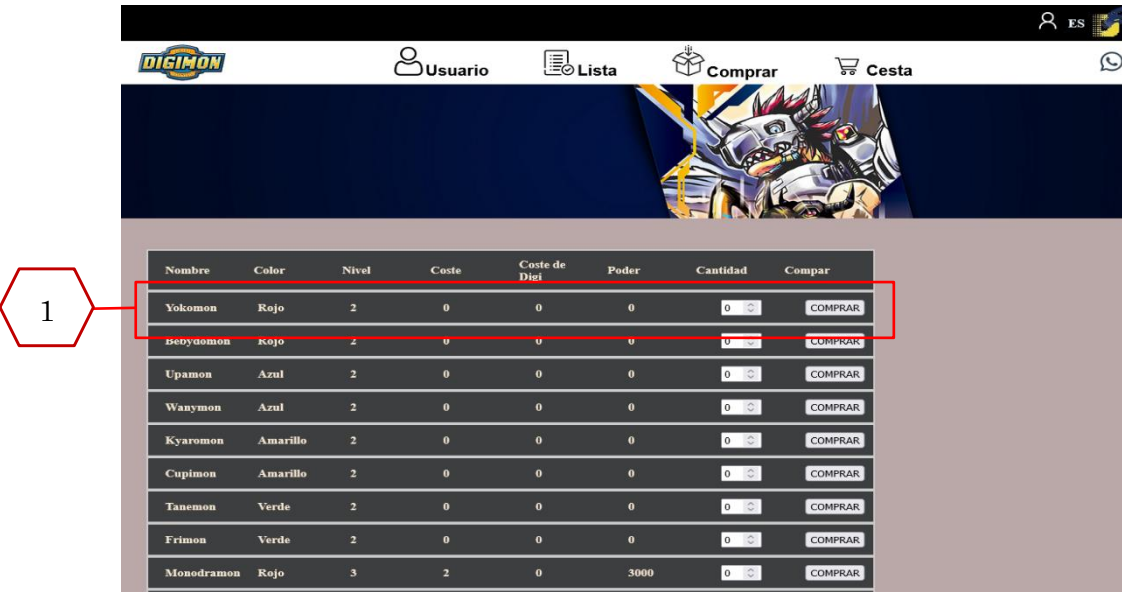
2.Carta: Imagen de la carta. Lleva al enlace con información específica de esta.

3.1.4 Carta



1.Carta: Muestra atributos y características de la carta de la lista y su descripción.

3.1.5 Sección Comprar



1.Compra de la carta: Se introduce de la carta que se desea, una cantidad de cartas para meter en la cesta.

3.1.6 Sección de Admin

Por la parte de back-en tenemos diferentes gestores de creación, edición y borrado tanto de los usuarios registrados como de las cartas listadas.

La barra de navegación cambiara según los permisos del usuario. Estas son las 3 interfaces que necesita el administrador.

The screenshot displays the Admin interface. At the top, there is a navigation bar with the DIGIMON logo and three main menu items: 'Listar usuarios' (with a user icon), 'Crear carta' (with a card icon), and 'Editar carta' (with a document icon). The main content area has a blue header with a white Digimon character icon on the right. Below this is a table with user data:

ID	Nombre	Apellido	Nombre de Usuario	Contraseña	Email	Acción 1
26	1234	1234	1234	81dc9bdb52d04dc20036dbd8313ed055	1234	Borrar
28	as	as	as	f970c2767d0cfe75876ea857892e319b	as	Borrar

Below the table, there is a footer section with three columns: 'Enlaces' (Inicio, Productos, Servicios, Contacto), 'Contacto' (Dirección: Calle Falsa 123, Springfield; Teléfono: +123456789; Email: alejandro@ejemplo.com), and 'Siguenos'. At the very bottom, a copyright notice reads: '© 2024 Empresa de Alejandro. Todos los derechos reservados.'

La creación y edición se hacen en dos paginas diferentes en las que además de permitir meter información de la carta, también permite adjuntar la foto necesaria y insertarla tanto a la base de datos como a nuestro proyecto desde cualquier directorio.

ES

DIGIMON

Listar usuarios Crear carta Editar carta

Nombre:

Color:







Nivel:

Coste:

Coste de digievolución:

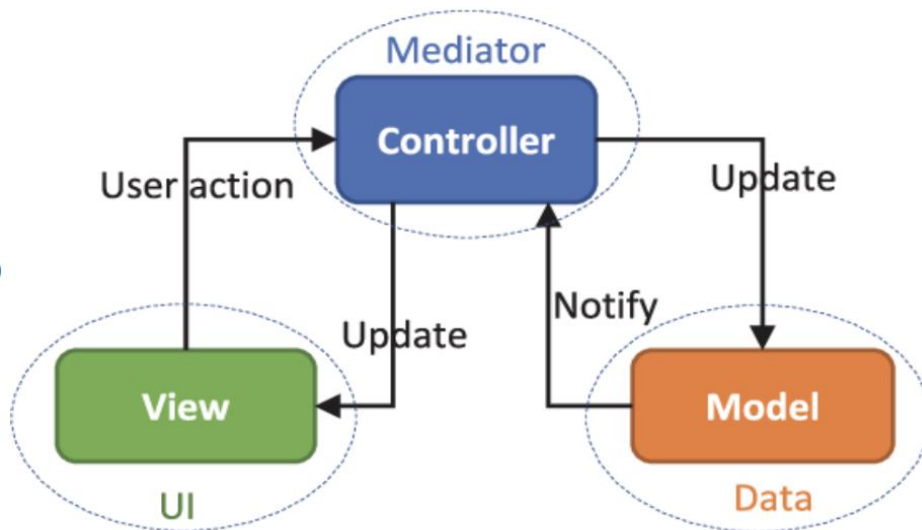
Poder:

Foto: No se ha seleccionado ningún archivo.

<div>  <div>  Listar usuarios  Crear carta  Editar carta </div> <div>  ES </div> </div>									
									
ID	Nombre	Color	Nivel	Coste	Coste de Digi	Poder	Foto	Accion 1	Accion 2
1	Yokomon	Rojo	2	0	0	0	BT1-001.jpg	Editar	Borrar
2	Bebydomon	Rojo	2	0	0	0	BT1-002.jpg	Editar	Borrar
3	Upamon	Azul	2	0	0	0	BT1-003.jpg	Editar	Borrar
4	Wanymon	Azul	2	0	0	0	BT1-004.jpg	Editar	Borrar
5	Kyaromon	Amarillo	2	0	0	0	BT1-005.jpg	Editar	Borrar
6	Cupimon	Amarillo	2	0	0	0	BT1-006.jpg	Editar	Borrar
7	Tanemon	Verde	2	0	0	0	BT1-007.jpg	Editar	Borrar
8	Frimon	Verde	2	0	0	0	BT1-008.jpg	Editar	Borrar
9	Monodramon	Rojo	3	2	0	3000	BT1-009.jpg	Editar	Borrar
10	Agumon	Rojo	3	3	0	2000	BT1-010.jpg	Editar	Borrar
11	Agumon Expert	Rojo	3	3	0	1000	BT1-011.jpg	Editar	Borrar
12	Biyomon	Rojo	3	3	0	2000	BT1-012.jpg	Editar	Borrar

3.2 Capa lógica o controlador

La capa lógica, dentro del contexto de desarrollo de software, es una parte fundamental de la arquitectura de un sistema. Esta capa se encarga de definir los comportamientos, reglas y procesos que gobiernan la interacción entre la interfaz de usuario y los datos almacenados en la capa de persistencia (como una base de datos)



Model-View-Control(MVC) design pattern.

En el contexto de un patrón de diseño como Modelo-Vista-Controlador (MVC), la capa lógica se separa de la capa de presentación (interfaz de usuario) y la capa de datos (persistencia). En el MVC, las clases y métodos dentro de la capa lógica representan la lógica de negocio y la manipulación de datos independiente de cómo se muestran o almacenan los datos.

Ejemplos.

```

public class Carta {
    private int id;
    private String nombre;
    private String color;
    private String nivel;
    private String coste;
    private String costeDeDigi;
    private String poder;
    private String foto;

    public Carta() {
    }

    public Carta(String nombre, String color, String nivel, String coste, String costeDeDigi, String poder, String foto) {
        this.nombre = nombre;
        this.color = color;
        this.nivel = nivel;
        this.coste = coste;
        this.costeDeDigi = costeDeDigi;
        this.poder = poder;
        this.foto = foto;
    }

    public Carta(int id, String nombre, String color, String nivel, String coste, String costeDeDigi, String poder, String foto) {
        this.id = id;
        this.nombre = nombre;
        this.color = color;
        this.nivel = nivel;
        this.coste = coste;
        this.costeDeDigi = costeDeDigi;
        this.poder = poder;
        this.foto = foto;
    }
}

```

```

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub

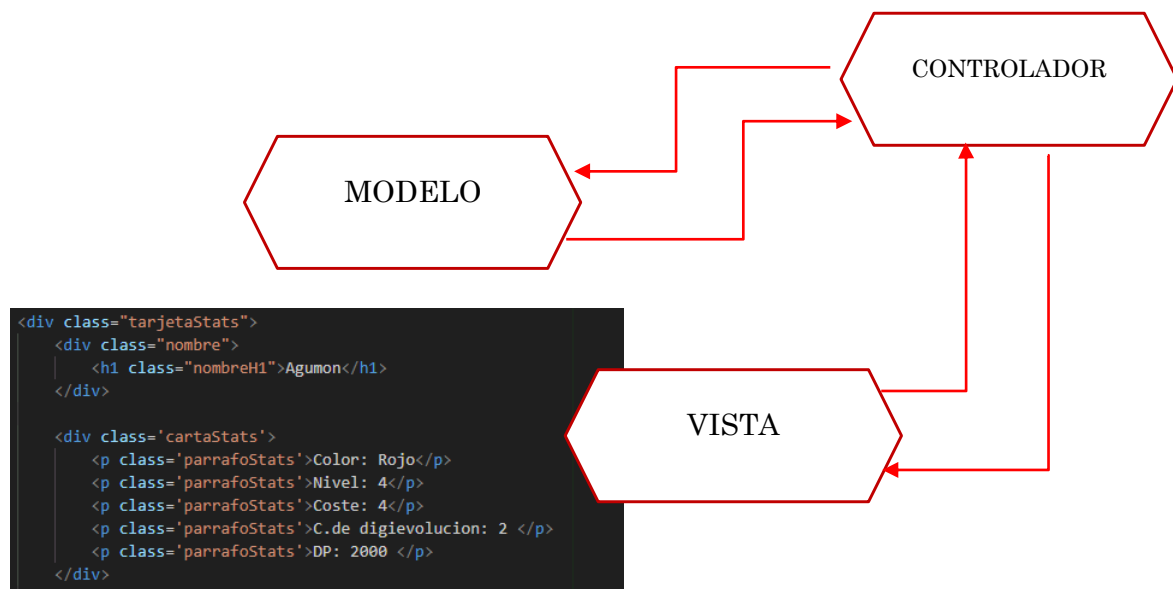
    String mail = request.getParameter("mail");
    String contrasena = request.getParameter("contrasena");

    Usuario u = new Usuario();
    u.setMail(mail);

    try {
        if(u.logueo(contrasena)) {
            session = request.getSession();
            session.setAttribute("id", u.getId());
            session.setAttribute("permiso", u.getPermiso());
            System.out.println(u.toString());

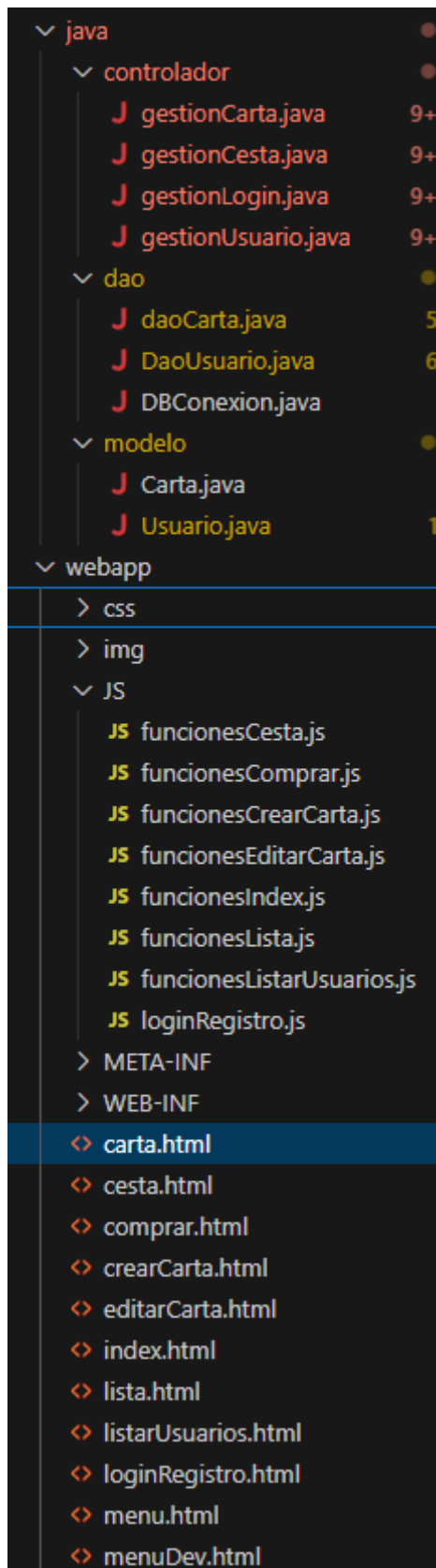
            if (u.getPermiso().equals(anObject:"a")) {
                response.sendRedirect("menu.html");
            } else {
                response.sendRedirect("menuDev.html");
            }
        } else {
            response.sendRedirect("error.html");
        }
    }
}

```



Con estos tres un ejemplo de código se ve como interactúan entre las tres partes el MVC.

Esto es un ejemplo que desgranaremos más adelante.



Archivos de la parte
CONTROLADOR del
proyecto



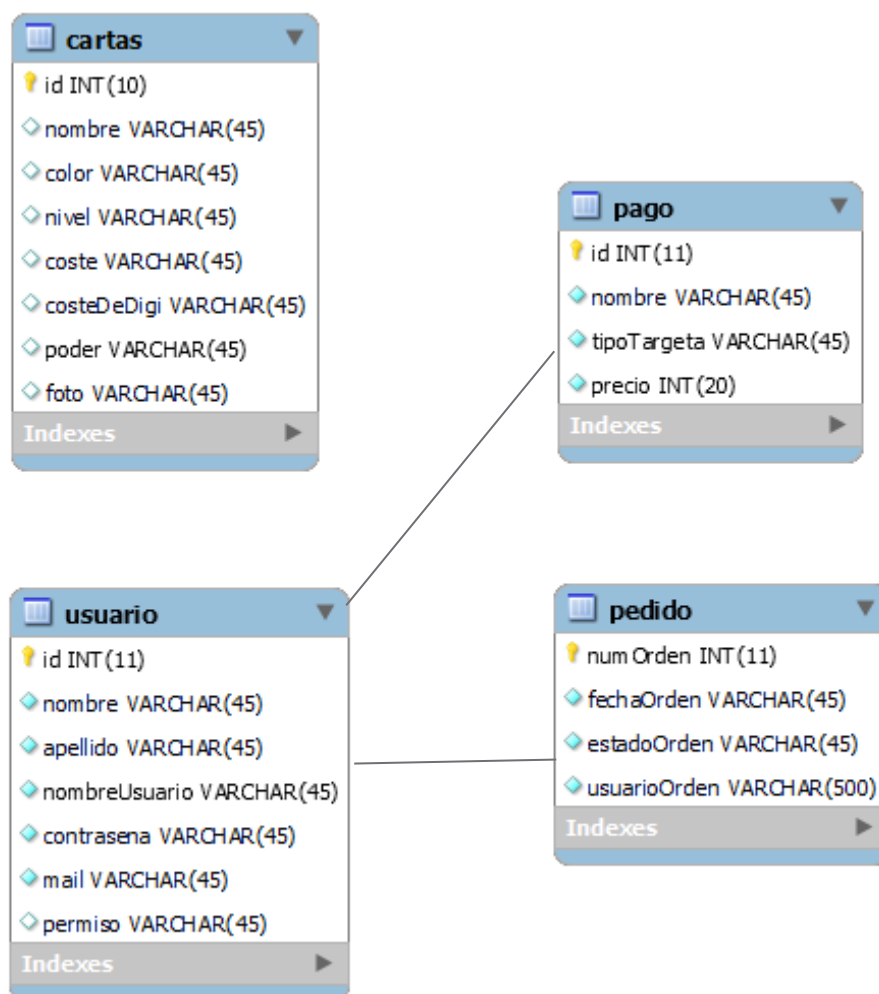
Archivos de la parte
MODELO del
proyecto



Archivos de la parte
VISTA del proyecto

3.3 Capa de datos

La capa de datos es la parte de la arquitectura de software donde se almacenan y manipulan los datos de la aplicación, separada de la lógica de negocio y la interfaz de usuario, asegurando la integridad y persistencia de la información.



4. Implementación

4.1 Tecnologías.

4.1.1 JAVA.

La tecnología de Java es una plataforma de desarrollo versátil y robusta que permite crear una amplia gama de aplicaciones, desde aplicaciones de escritorio hasta sistemas empresariales y aplicaciones web. Utiliza un enfoque de "write once, run anywhere" (escribe una vez, ejecuta en cualquier lugar), lo que significa que el código Java puede ejecutarse en múltiples plataformas sin necesidad de modificaciones. Java se ha destacado por su seguridad, portabilidad y rendimiento, lo que lo convierte en una opción popular para desarrolladores en una variedad de industrias.

Además, la comunidad de Java es extensa y activa, lo que garantiza un amplio soporte y una gran cantidad de recursos disponibles para los desarrolladores.

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("¡Hola, mundo!");  
    }  
}
```



4.1.2 HTML.

HTML, o HyperText Markup Language, es el lenguaje estándar utilizado para crear y diseñar páginas web. Utiliza una estructura de marcado que define la estructura y el contenido de una página web mediante etiquetas. Estas etiquetas permiten al desarrollador especificar diferentes elementos como encabezados, párrafos, enlaces, imágenes y más. HTML es un lenguaje de marcado simple y fácil de aprender, lo que lo convierte en la base fundamental para la creación de sitios web. Junto con CSS (Cascading Style Sheets) y JavaScript, HTML forma la tríada básica de tecnologías utilizadas en el desarrollo web moderno.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ejemplo de HTML Básico</title>
</head>
<body>

  <h1>¡Hola, mundo!</h1>

  <p>Este es un ejemplo básico de una página web utilizando HTML.</p>

</body>
</html>
```



4.1.3 CSS

CSS, o Cascading Style Sheets, es un lenguaje utilizado para describir la presentación de un documento HTML (o XML). Permite definir el aspecto visual de los elementos de una página web, como el color, la tipografía, el diseño y otros aspectos relacionados con la apariencia. CSS utiliza reglas que se aplican a elementos específicos o conjuntos de elementos en un documento HTML, lo que proporciona un control preciso sobre cómo se visualiza el contenido en diferentes dispositivos y tamaños de pantalla. Al separar el contenido de su presentación, CSS promueve la creación de sitios web más accesibles, mantenibles y consistentes en términos de diseño. Con la evolución de CSS y sus capacidades, los desarrolladores pueden crear diseños más complejos y adaptativos, lo que contribuye a una experiencia de usuario más rica y atractiva en la web.

```
<style>
  body {
    font-family: Arial, sans-serif;
    background-color: #f0f0f0;
    color: #333;
    margin: 0;
    padding: 20px;
  }

  h1 {
    color: #0066cc;
  }

  p {
    font-size: 18px;
    line-height: 1.5;
  }
</style>
```



4.1.4 JavaScript

JavaScript es un lenguaje de programación de alto nivel, interpretado por el navegador web, que se utiliza para agregar interactividad y dinamismo a las páginas web. A diferencia de HTML y CSS, que se centran en la estructura y el estilo, respectivamente, JavaScript se enfoca en la funcionalidad y el comportamiento de una página web. Permite a los desarrolladores crear aplicaciones web complejas, manipular el contenido de la página en tiempo real, responder a acciones del usuario, realizar solicitudes de red asincrónicas y mucho más. JavaScript es un lenguaje versátil que se ha expandido más allá del navegador web, con frameworks y librerías que permiten el desarrollo tanto de aplicaciones web como de aplicaciones móviles y de escritorio. Su capacidad para interactuar con HTML y CSS lo convierte en una herramienta poderosa para el desarrollo web moderno.

```
<body>
  <script>
    alert("¡Hola, mundo!");
  </script>
</body>
```



4.1.5 SQL

SQL, o Structured Query Language, es un lenguaje de programación diseñado para gestionar y manipular bases de datos relacionales. Proporciona un conjunto de comandos que permiten realizar diversas operaciones, como la creación, modificación y eliminación de tablas y datos, así como la consulta y la actualización de información almacenada en la base de datos. SQL se utiliza en una amplia gama de aplicaciones y entornos, desde sistemas de gestión de bases de datos (DBMS) como MySQL, PostgreSQL y Oracle, hasta aplicaciones web y empresariales. Su sintaxis es clara y fácil de aprender, lo que lo hace accesible tanto para desarrolladores novatos como experimentados. Además, SQL es un estándar ampliamente aceptado en la industria, lo que garantiza la portabilidad y la interoperabilidad entre diferentes sistemas y tecnologías de bases de datos.

```
1 • create database clinica;
2 • use clinica;
3 • create table cliente(
4     cod_cliente char(5) not null,
5     nombre varchar(30) not null,
6     apellidos varchar(30) not null,
7     direccion varchar(30),
8     telefono varchar(15),
9     primary key(cod_cliente)
10 );
11
12 • INSERT INTO CLIENTE(COD_CLIENTE,NOMBRE,APELLIDOS,TELEFONO)
13 VALUES ('C_1','Luis','Pérez Pérez',33344455);
14 • INSERT INTO CLIENTE(COD_CLIENTE,NOMBRE,APELLIDOS,DIRECCION)
15 VALUES ('C_2','Juan','García López','Del Valle');
16 • INSERT INTO CLIENTE(COD_CLIENTE,NOMBRE,APELLIDOS,DIRECCION,TELEFONO)
17 VALUES ('C_3','Rosa','Sánchez Gil','Paseo Recoletos',8888888);
18 • INSERT INTO CLIENTE(COD_CLIENTE,NOMBRE,APELLIDOS,DIRECCION,TELEFONO)
19 VALUES ('C_4','Isabel','Rodríguez Pla','De la Luz',5555555);
20
```



4.2 Tecnologías.

4.2.1 Tecnologías visuales.

Para este proyecto he utilizado la los exploradores web que normalmente se utilizan. Para el testeo de la tecnología JAVA he utilizado Firefox y para el testeo y desarrollo de la tecnología JAVASCRIPT HTML y CSS he utilizado el explorador Chrome.

Esto me ha facilitado el trabajo por ejemplo en el vaciado de cache en diferentes lapsus de tiempo.



4.2.2 Hardware.

El proyecto esta corriendo en un ordenador personal CPU con los siguientes componentes:

- Microprocessor: AMD Ryzen 7 1800X Eight-Core Processor 3.60 GHz.
- RAM: 16,0 GB DDR4.
- Tarjeta gráfica: NVIDIA 2070 Super.
- Memoria: SSD 1 T.
- Placa base: MSI AB350-Gaming 3.
- SO: Microsoft Windows 10 Pro

