



**DATOS DEL ESTUDIANTE**

Apellidos y Nombres:	Pachas Saavedra Jesús Omar	ID:	001514827
Dirección Zonal/CFP:	Ica – Ayacucho / CFP CHINCHA		
Carrera:	Ingeniería de Software con Inteligencia Artificial	Semestre:	V
Curso/ Mód. Formativo:	FULLSTACK DEVELOPER SOFTWARE		
Tema de Trabajo Final:	Implementación de un software web		

**1. INFORMACIÓN****▪ Identifica la problemática del caso práctico propuesto.**

Elaboración de una web usando node.js para implementar un software de ventas con un catalogo incluido donde se puedan apreciar los productos registrados

**▪ Identifica propuesta de solución y evidencias.**

Propongo un software con un navbar que contendrá varias pestañas, entre las principales una pestaña admin y otra catálogos, donde en la pestaña admin podre realizar las operación crud y en la pestaña catalogo mostrar los registros, también se podrá aplicar filtros para una búsqueda fácil de un producto en especifico

▪ **Respuestas a preguntas guía**

Durante el análisis y estudio del caso práctico, debes obtener las respuestas a las interrogantes:

<b>Pregunta 01</b>	<b>¿Cuáles son los principales desafíos operativos que enfrenta la botica "FarmaPlus" debido a la falta de un sistema de gestión de inventarios automatizado?</b>
<p>Entre sus desafíos están:</p> <ul style="list-style-type: none"> <li>• <b>Errores humanos:</b> Registros manuales por lo cual puede surgir equivocaciones.</li> <li>• <b>Desabastecimiento frecuente:</b> Falta de alertas para reponer productos agotados.</li> <li>• <b>Tiempos de espera largos:</b> Búsqueda manual de productos en inventario, demorando la atención.</li> </ul>	
<b>Pregunta 02</b>	<b>¿Qué características específicas debe incluir el software web para atender las necesidades de control de stock y atención al cliente en "FarmaPlus"?</b>
<p>El software debe de tener:</p> <ul style="list-style-type: none"> <li>• <b>Control de stock en tiempo real:</b> Actualización automática al vender/comprar.</li> <li>• <b>Alertas de reposición:</b> Notificaciones cuando el stock esté bajo.</li> <li>• <b>Registro de ventas integrado:</b> Facturación rápida con descuento automático.</li> <li>• <b>Interfaz de búsqueda rápida:</b> Filtros por nombre, categoría o código de producto.</li> </ul>	
<b>Pregunta 03</b>	<b>¿De qué manera la implementación de un software web para la gestión de inventarios puede impactar en la reducción del desabastecimiento y la mejora del servicio al cliente?</b>
<p>El software reducirá el desabastecimiento mediante alertas automáticas que avisan cuando el stock llega a niveles mínimos, permitiendo reposiciones oportunas. Además, agilizará el servicio al cliente al mostrar en tiempo real la disponibilidad de productos, eliminando búsquedas manuales y reduciendo los tiempos de espera durante la compra.</p>	
<b>Pregunta 04</b>	<b>¿Qué beneficios adicionales, aparte del control de inventarios, podría ofrecer el software para incrementar la competitividad de "FarmaPlus" en su mercado?</b>
<p>Aparte del control de inventario, el software ofrecerá ventajas como optimización de compras (con reportes de productos más vendidos) y una imagen más profesional al adoptar tecnología digital, lo que puede atraer más clientes y diferenciar a "FarmaPlus" de la competencia.</p>	
<b>Pregunta 05</b>	<b>¿Qué métricas se pueden utilizar para evaluar la efectividad del software en mejorar los tiempos de respuesta al cliente y reducir los errores en la gestión de inventario?</b>
<p>Para medir el éxito del sistema, se usarán indicadores como:</p> <ul style="list-style-type: none"> <li>• <b>Tiempo promedio de atención.</b></li> <li>• <b>Nivel de stock crítico.</b></li> <li>• <b>Errores en inventario.</b></li> <li>• <b>Ventas recuperadas</b> (aumento por disponibilidad de productos),</li> <li>• <b>Satisfacción del cliente</b> (encuestas).</li> </ul>	

## 2. PLANIFICACIÓN DEL TRABAJO

### ▪ Cronograma de actividades:

N°	ACTIVIDADES	CRONOGRAMA					
01	Diseño de la base de datos						
02	Procedimiento crud para la primera tabla						
03	Procedimiento crud para las demás tables						
04	Implementación del catalogo						

### ▪ Lista de recursos necesarios:

1. MÁQUINAS Y EQUIPOS	
Descripción	Cantidad
Laptop	

2. HERRAMIENTAS E INSTRUMENTOS	
Descripción	Cantidad
Bootstrap 5	
SweetAlert	
Visual studio code	
Node.js	

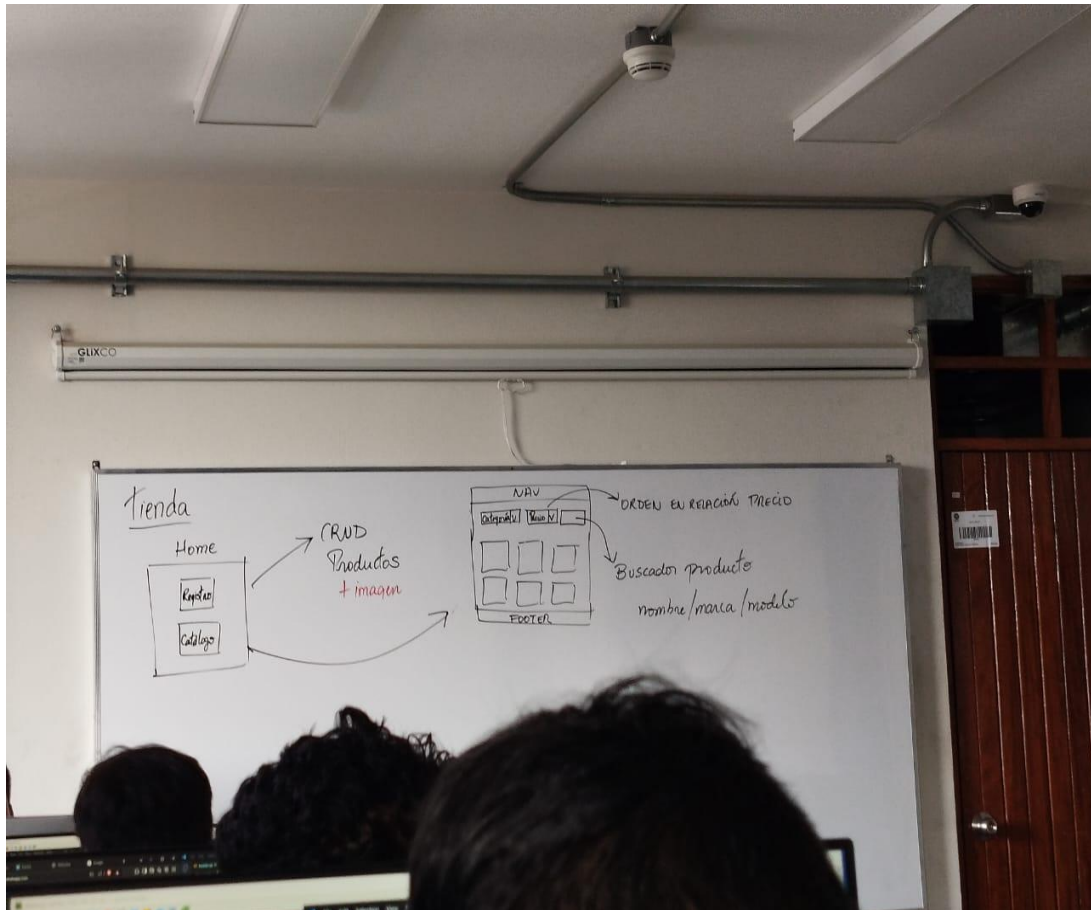
3. MATERIALES E INSUMOS	
Descripción	Cantidad
Material realizado en clase (proyectos)	

### 3. DECIDIR PROPUESTA

- Describe la propuesta determinada para la solución del caso práctico

#### PROPUESTA DE SOLUCIÓN

Para resolver el enunciado elaborare un software de una tienda, donde implementare un apartado administrador donde se pueden subir registros al catalogo, en el catalogo mostrare el productor con la imagen correspondiente, también se podrán aplicar algunos filtros para una mejor búsqueda del usuario.



## 4. EJECUTAR

- Resolver el caso práctico, utilizando como referencia el problema propuesto y las preguntas guía proporcionadas para orientar el desarrollo.
- Fundamentar sus propuestas en los conocimientos adquiridos a lo largo del curso, aplicando lo aprendido en las tareas y operaciones descritas en los contenidos curriculares.

**INSTRUCCIONES:** Ser lo más explícito posible. Los gráficos ayudan a transmitir mejor las ideas. Tomar en cuenta los aspectos de calidad, medio ambiente y SHI.

[illegible]

## DIBUJO / ESQUEMA / DIAGRAMA DE PROPUESTA

(Adicionar las páginas que sean necesarias)

Para elaborar el software usare node.js con su framework express, implementare el proceso crud a todas las tablas, esto para que al agregar un producto, categoría o marca nueva, los filtros se puedan realizar de forma dinámica, de esta manera el usuario podrá agregar productos de diferentes marcas y categorías las veces que quiera sin tener problemas en los filtros o la búsqueda.

### 1. Implementación de la base de datos

Como primer paso implementare la base de datos, también realizare y probare algunas consultas para no tener problemas al momento de realizar las rutas o los archivos.ejs

```

1 • CREATE DATABASE mundodigital;
2 • USE mundodigital;
3
4 • CREATE TABLE marcas(
5   idmarca INT PRIMARY KEY AUTO_INCREMENT,
6   marca VARCHAR(50) NOT NULL
7 ) ENGINE = INNODB;
8
9 • CREATE TABLE categorias(
10  idcategoria INT PRIMARY KEY AUTO_INCREMENT,
11  categoria VARCHAR(50) NOT NULL
12 ) ENGINE = INNODB;
13
14 • CREATE TABLE videojuegos(
15  idvideojuego INT PRIMARY KEY AUTO_INCREMENT,
16  titulo VARCHAR(100) NOT NULL,
17  descripcion VARCHAR(180) NOT NULL,
18  precio DECIMAL(10,2) NOT NULL,
19  flanzamiento CHAR(4) NOT NULL,
20  peso VARCHAR(50) NOT NULL,
21  edadrec INT NOT NULL,
22  idmarca INT,
23  idcategoria INT,
24  imagen VARCHAR(280),
25  FOREIGN KEY (idmarca) REFERENCES marcas(idmarca),
26  FOREIGN KEY (idcategoria) REFERENCES categorias(idcategoria)
27 ) ENGINE = INNODB;

```

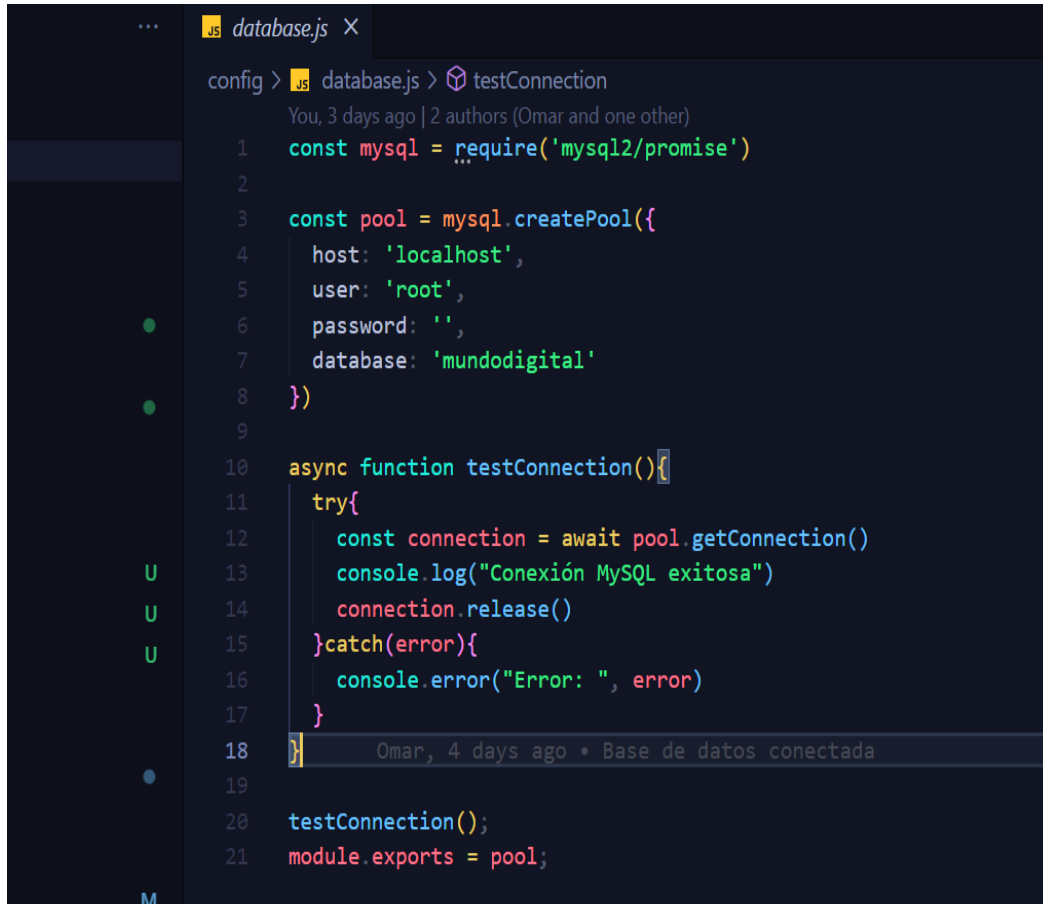
```

28
29 • INSERT INTO marcas (marca) VALUES
30   ('PS'),
31   ('Nintendo Switch'),
32   ('Xbox');
33
34 • INSERT INTO categorias (categoria) VALUES
35   ('Acción'),
36   ('Deportes'),
37   ('RPG');
38
39
40 • INSERT INTO videojuegos (titulo, descripcion, precio, flanzamiento, peso, edadrec, idmarca, idcategoria, imagen) VALUES
41   ('God of War Ragnarök', 'Kratos y Atreus en un viaje épico por los Nuevos Reinos', 69.99, '2022', '106 GB', 18, 1, 1, 'gow.jpg'),
42   ('The Legend of Zelda: Tears of the Kingdom', 'Link explora los cielos y descubre misterios en Hyrule', 59.99, '2023', '16.3 GB', 12, 2, 1, 'zelda.jpg'),
43   ('Forza Horizon 5', 'Experiencia de carreras en mundo abierto en México', 59.99, '2021', '110 GB', 3, 3, 2, 'forzah.jpg');
44
45 • SELECT * FROM videojuegos;
46
47
48
49

```

## 2. Configuración de la conexión

Implementare la conexión dentro de mi proyecto mundo-digital en una carpeta config, dentro tendré 3 archivos entre ellos multer, que me servirá para poder realizar el formulario con imágenes y poder darle el funcionamiento correcto y otro archivo llamado database.js, que contendrá la conexión a la base de datos

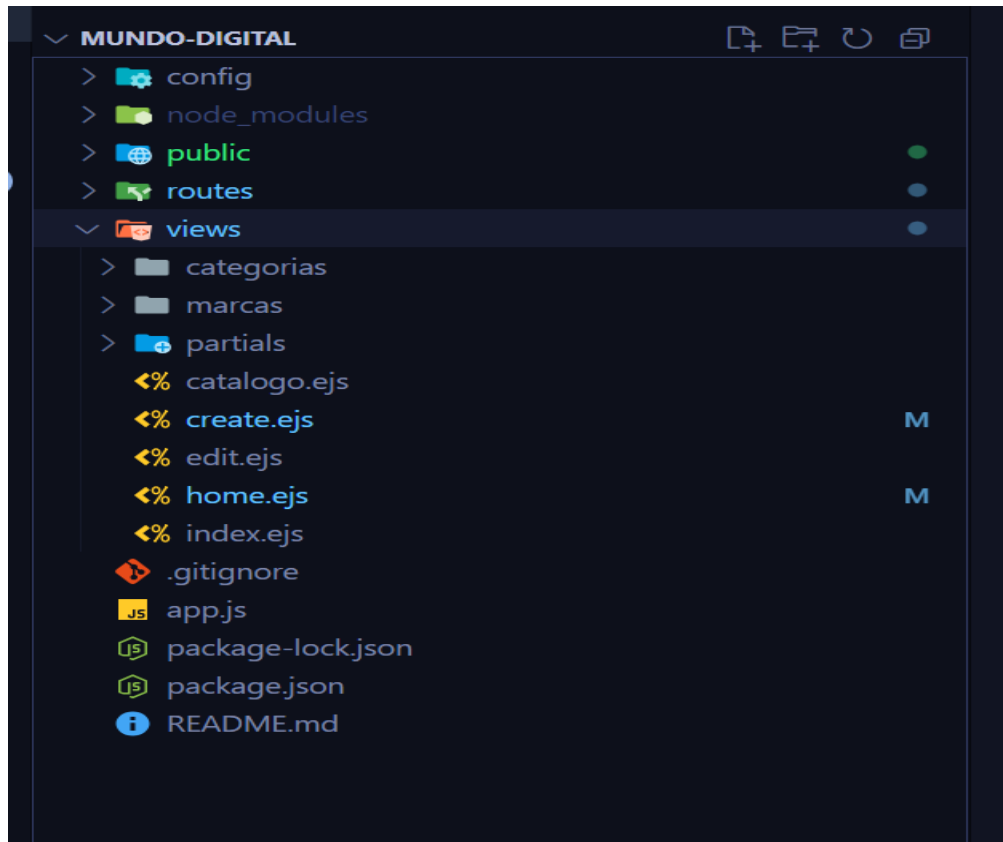


```
... database.js X
config > database.js > testConnection
You, 3 days ago | 2 authors (Omar and one other)
1  const mysql = require('mysql2/promise')
2
3  const pool = mysql.createPool({
4    host: 'localhost',
5    user: 'root',
6    password: '',
7    database: 'mundodigital'
8  })
9
10 async function testConnection(){
11   try{
12     const connection = await pool.getConnection()
13     console.log("Conexión MySQL exitosa")
14     connection.release()
15   }catch(error){
16     console.error("Error: ", error)
17   }
18   Omar, 4 days ago • Base de datos conectada
19
20 testConnection();
21 module.exports = pool;
```



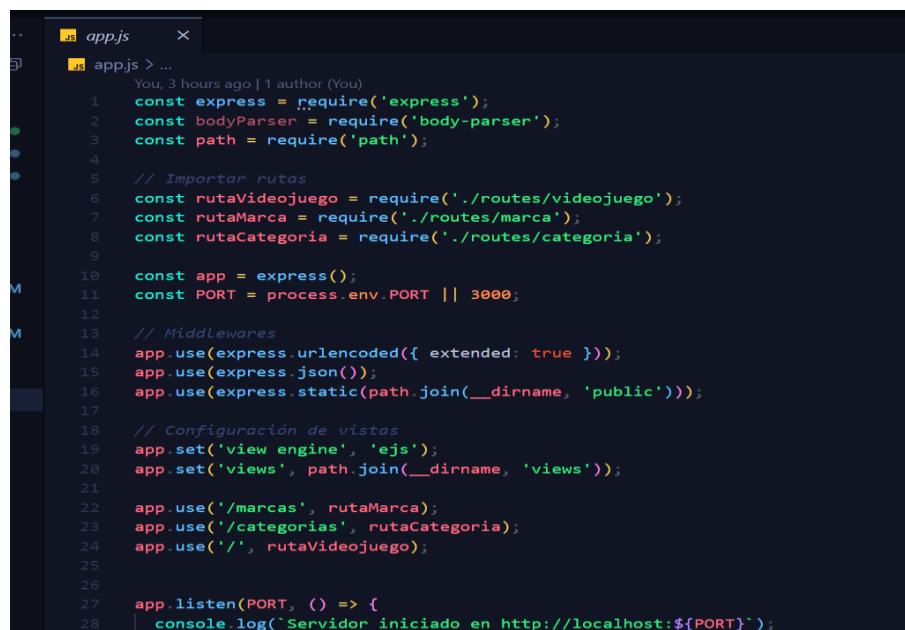
### 3. Estructura de las carpetas

Implementé una estructura organizada para mi proyecto, que facilita la gestión del código y la separación de responsabilidades. La estructura de carpetas es la siguiente:



### 4. Configuración del servidor

El archivo principal de mi servidor lo implemente en app.js. Aquí configuré Express, las rutas y el middleware necesario para manejar las solicitudes HTTP.



## 5. Rutas principales

Las rutas del proyecto las definí en el archivo videojuego.js Este archivo contiene las rutas para listar, filtrar, agregar, editar y eliminar videojuegos, también tiene incluida la ruta para mostrar el catálogo para mostrar los videojuegos en los cards y los filtros para las búsquedas.

```

videojuego.js M X
routes > videojuego.js > router.post('/create') callback
const db = require('../config/database');
const upload = require('../config/multer');
const fs = require('fs');
const path = require('path');

//ruta para obtener el total de registros en home
router.get('/', async (req, res) => {
  try {
    const [vjuegosResult] = await db.query('SELECT COUNT(*) AS total FROM videoj');
    const totalJuegos = vjuegosResult[0].total;

    const [marcasResult] = await db.query('SELECT COUNT(*) AS total FROM marcas');
    const totalMarcas = marcasResult[0].total;

    const [categoriasResult] = await db.query('SELECT COUNT(*) AS total FROM cat');
    const totalCategorias = categoriasResult[0].total;

    res.render('home', {
      total: totalJuegos,
      totalmarcas: totalMarcas,
      totalcategorias: totalCategorias
    });
  } catch (error) {
    console.error('Error:', error);
    res.status(500).send('Error al cargar los datos');
  }
});

// Ruta para el listado de registros
router.get('/index', async (req, res) => {
  try {
    const query = `
      SELECT
        C.idvideojuego,

```

```

      res.redirect('/');
    }
  });
  router.get('/catalogo', async (req, res) => {
    try {
      const { titulo, marca, categoria } = req.query;

      let query = `
        SELECT
          C.idvideojuego,
          C.titulo,
          C.descripcion,
          C.precio,
          C.flanzamiento,
          C.peso,
          C.edadrec,
          C.imagen,
          M.marca,
          CA.categoria
        FROM videojuegos C
        INNER JOIN marcas M ON C.idmarca = M.idmarca
        INNER JOIN categorias CA ON C.idcategoria = CA.idcategoria
        WHERE 1=1
      `;

      const params = [];


      if (titulo) {
        query += ` AND C.titulo LIKE ?`;
        params.push(`%${titulo}%`);
      }

      if (marca) {

```

## 6. Configuración de subida de imágenes

Implementé la configuración para manejar la subida de imágenes al servidor utilizando la biblioteca Multer. Esto permite que los usuarios puedan subir imágenes personalizadas para los videojuegos al momento de registrarlos o editarlos.

```
config >  multer.js > ...  
You, 2 hours ago | 1 author (You)  
1  const multer = require('multer');  
2  const path = require('path');  
3  
4  // Configuración de almacenamiento  
5  const storage = multer.diskStorage({  
6    destination: (req, file, cb) => {  
7      cb(null, path.join(__dirname, '../public/images/videojuegos'));  
8    },  
9    filename: (req, file, cb) => {  
10      const uniqueSuffix = Date.now() + '-' + Math.round(Math.random() * 1E9);  
11      cb(null, file.fieldname + '-' + uniqueSuffix + path.extname(file.originalname));  
12    }  
13  });  
14  
15  const fileFilter = (req, file, cb) => {  
16    if (file.mimetype.startsWith('image/')) {  
17      cb(null, true);  
18    } else {  
19      cb(new Error('Solo se permiten imágenes'), false);  
20    }  
21  };  
22  
23  const upload = multer({  
24    storage: storage,  
25    fileFilter: fileFilter,  
26    limits: { fileSize: 5 * 1024 * 1024 } // Límite  
27  });  
28  
29  module.exports = upload;
```

## 7. Vista catalogo.ejs

En esta que vista implemente el catálogo de videojuegos. Los usuarios pueden ver los videojuegos en tarjetas con detalles como título, descripción, precio, marca y categoría. Además, incluye un formulario para buscar y filtrar los videojuegos según diferentes criterios.

```

views > <% catalogo.ejs > html > body > ? > div.container.mt-4 > form.mb-4 > div.row.g-2 >
2  <html lang="es">
3  <head>
4
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <title>Catálogo de Videojuegos</title>
7    <link rel="stylesheet" href="/public/css/styles.css">
8  </head>
9  <body>
10   <%- include('partials/header') %>
11
12   <div class="container mt-4">
13     <h1 class="text-center text-primary mb-4">Catálogo de Videojuegos</h1>
14
15     <!-- Formulario de búsqueda -->
16     <form action="/catalogo" method="GET" class="mb-4">
17       <div class="row g-2">
18         <div class="col-md-4">
19           <input type="text" name="titulo" class="form-control" placeholder="Busc
20         </div>
21         <div class="col-md-4">
22           <select name="marca" class="form-select">
23             <option value="">Todas las marcas</option>
24             <% marcas.forEach(marca => { %>
25               <option value="<%= marca.marca %>"><%= marca.marca %></option>
26             <% }> %>
27           </select>
28         </div>
29         <div class="col-md-4">
30           <select name="categoria" class="form-select">

```

## 8. Vista para agregar videojuegos

En este vista implemente un formulario que permite al usuario agregar nuevos videojuegos al catálogo. Incluye un formulario donde se pueden ingresar los datos del videojuego, como título, descripción, precio, imagen, marca y categoría.

## 9. Vista para editar videojuegos

```

1  <%- include('partials/header') %>
2
3  <form action="/update/<%= videojuego.idvideojuego %>" id="formulario-edicion" aut
4  <div class="card mt-2">
5    <div class="card-header"><strong>Actualización de Videojuego</strong></div>
6
7    <div class="card-body">
8
9      <input type="hidden" name="imagenActual" value="<%= videojuego.imagen %>">
10
11      <div class="row g-2">
12        <div class="col-md-6 mb-2">
13          <div class="form-floating">
14            <select name="idmarca" id="idmarca" class="form-select" required>
15              <option value="" disabled>Selecione una marca</option>
16              <%= marcas.forEach(marca => { %>
17                <option value="<%= marca.idmarca %>" <%= marca.idmarca == videoj
18                  <%= marca.marca %>
19              </option>
20              <%= }) %>
21            </select>
22            <label>Marca</label>
23          </div>
24        </div>
25        <div class="col-md-6 mb-2">
26          <div class="form-floating">
27            <select name="idcategoria" id="idcategoria" class="form-select" requi
28              <option value="" disabled>Selecione una categoria</option>
29              <%= categorias.forEach(cat => { %>
30                <option value="<%= cat.idcategoria %>" <%= cat.idcategoria == vic
31                  <%= cat.categoria %>
32              </option>
33              <%= }) %>

```

## Actualización de Videojuego

Marca

PS5

Categoría

Acción

Título del Videojuego

God of War Ragnarök

Descripción

Kratos y Atreus en un viaje épico por los Nuevos Reinos

Precio (\$)

69,99

Año de Lanzamiento

2022

Edad Recomendada

18

Peso (GB)

106 GB

Subir Nueva Imagen

Seleccionar archivo

Ningún archivo seleccionado

Guardar Cambios

Cancelar




## 10. Ruta para listar videojuegos



En este paso implementé la ruta que permite listar todos los videojuegos registrados en el sistema. Esta ruta es fundamental para mostrar los datos en la vista del catálogo y para que los usuarios puedan ver toda la información disponible.

```

<% index.ejs >
views > <% index.ejs > <div.container.mt-4 > <div.text-end
You, 3 hours ago | 1 author (You)
1 <%- include('partials/header') %>
2
3 <div class="container mt-4">
4 <div class="table-responsive">
5 <table class="table table-bordered table-hover table-sm align-middle">
6 <thead class="table-primary text-center">
7 <tr>
8 <th>ID</th>
9 <th>Título</th>
10 <th>Descripción</th>
11 <th>Precio</th>
12 <th>Fecha de Lanzamiento</th>
13 <th>Almacenamiento</th>
14 <th>Edad Recomendada</th>
15 <th>Marca</th>
16 <th>Categoría</th>
17 <th>Imagen</th>
18 <th>Operaciones</th>
19 </tr>
20 </thead>
21 <tbody>
22 <% videojuegos.forEach(videojuego => { %>
23 <tr>
24 <td class="text-nowrap text-center"><%= videojuego.idvideojuego %></td>
25 <td class="text-nowrap"><%= videojuego.titulo %></td>
26 <td class="text-nowrap"><%= videojuego.descripcion %></td>
27 <td class="text-nowrap">$ / <%= Number(videojuego.precio).toFixed(2) %><
28 <td class="text-nowrap text-center"><%= videojuego.flanzamiento %></td>
29 <td class="text-nowrap"><%= videojuego.peso %> kg</td>
30 <td class="text-nowrap"><%= videojuego.edadrec %></td>
31 <td class="text-nowrap"><%= videojuego.marca %></td>
32 <td class="text-nowrap"><%= videojuego.categoria %></td>
33 <td class="text-nowrap text-center">
34 
36 <td class="text-nowrap">
37 <div class="btn-group" role="group">

```

Descripción	Precio	Fecha de Lanzamiento	Almacenamiento	Edad Recomendada	Marca	Categoría	Imagen	Operaciones
Kratos y Atreus en un viaje épico por los Nuevos Reinos	S/ 69.99	2022	106 GB kg	18	PS5	Acción		<a href="#">Editar</a> <a href="#">Eliminar</a>
Link explora los cielos y descubre misterios en Hyrule	S/ 59.99	2023	16.3 GB kg	12	Nintendo Switch	Acción		<a href="#">Editar</a> <a href="#">Eliminar</a>
eflds	S/ 1200.00	2024	25 gb kg	14	PS5	Deportes		<a href="#">Editar</a> <a href="#">Eliminar</a>

[Registrar nueva consola](#)

## 11. Proceso crud para marcas y modelos

Siguiendo la misma lógica, pude implementar el proceso crud para las tablas marcas, categorías, como ambas tablas contienen solo un campo, me resultado fácil de realizar, con estas 2 tablas culminadas, el usuario ya puede agregar las marcas que quiera, como también las categorías que quiera, los filtros que contienen categorías y marcas, se modificaran según la cantidad de categorías y marcas que exitan en estas tablas

Estos procesos los dividí en dos carpetas: marcas y categorías. Dentro de estas carpetas se encuentran los archivos necesarios para manejar las rutas, vistas y lógica del CRUD.

Con el proceso CRUD para marcas y categorías, el sistema es más flexible, ya que los usuarios pueden gestionar estas tablas según sus necesidades. Esto asegura que los filtros de búsqueda en el catálogo se actualicen automáticamente con las nuevas marcas y categorías registradas.


## Catálogo de Videojuegos

Todas las marcas

Todas las categorías

[Filtrar](#)

God of War Ragnarök



PS5 ACCIÓN

Kratos y Atreus en un viaje épico por los Nuevos Reinos

Precio: S/ 69.99


Lanzamiento: 2022

Peso: 106 GB kg

Edad Recomendada: 18 años

Adquirir

The Legend of Zelda: Tears of the Kingdom



NINTENDO SWITCH ACCIÓN

Link explora los cielos y descubre misterios en Hyrule

Precio: S/ 59.99


Lanzamiento: 2023

Peso: 16.3 GB kg

Edad Recomendada: 12 años

Adquirir

Forza



PS5 DEPORTES

eflds

Precio: S/ 1200.00

Lanzamiento: 2024

Peso: 25 gb kg

Edad Recomendada: 14 años

Adquirir

Marcas		
ID	Marca	Acciones
1	PS5	<a href="#">Editar</a> <a href="#">Eliminar</a>
2	Nintendo Switch	<a href="#">Editar</a> <a href="#">Eliminar</a>
3	Xbox	<a href="#">Editar</a> <a href="#">Eliminar</a>
4	Sega	<a href="#">Editar</a> <a href="#">Eliminar</a>
5	Android	<a href="#">Editar</a> <a href="#">Eliminar</a>

[Volver](#)

Categorías		
ID	Categoría	Acciones
1	Acción	<a href="#">Editar</a> <a href="#">Eliminar</a>
2	Deportes	<a href="#">Editar</a> <a href="#">Eliminar</a>
3	RPG	<a href="#">Editar</a> <a href="#">Eliminar</a>
4	Terror	<a href="#">Editar</a> <a href="#">Eliminar</a>

[Volver](#)

ACCIÓN home Catálogo



Videoguegos registrados:

3

[Agregar videogame](#)

[Ver videogame](#)



Total de marcas:

5

[Agregar marca](#)

[Ver marcas](#)



Total de categorías:

4

[Agregar categoría](#)

[Ver categorías](#)



- **Verificar el cumplimiento de los procesos desarrollados en la propuesta de solución del caso práctico.**

<b>EVIDENCIAS</b>	<b>CUMPLE</b>	<b>NO CUMPLE</b>
• ¿Se identificó claramente la problemática del caso práctico?	<input checked="" type="checkbox"/>	<input type="checkbox"/>
• ¿Se desarrolló las condiciones de los requerimientos solicitados?	<input checked="" type="checkbox"/>	<input type="checkbox"/>
• ¿Se formularon respuestas claras y fundamentadas a todas las preguntas guía?	<input checked="" type="checkbox"/>	<input type="checkbox"/>
• ¿Se elaboró un cronograma claro de actividades a ejecutar?	<input checked="" type="checkbox"/>	<input type="checkbox"/>
• ¿Se identificaron y listaron los recursos (máquinas, equipos, herramientas, materiales) necesarios para ejecutar la propuesta?	<input type="checkbox"/>	<input checked="" type="checkbox"/>
• ¿Se ejecutó la propuesta de acuerdo con la planificación y cronograma establecidos?	<input checked="" type="checkbox"/>	<input type="checkbox"/>
• ¿Se describieron todas las operaciones y pasos seguidos para garantizar la correcta ejecución?	<input type="checkbox"/>	<input checked="" type="checkbox"/>
• ¿Se consideran las normativas técnicas, de seguridad y medio ambiente en la propuesta de solución?	<input type="checkbox"/>	<input checked="" type="checkbox"/>
• ¿La propuesta es pertinente con los requerimientos solicitados?	<input checked="" type="checkbox"/>	<input type="checkbox"/>
• ¿Se evaluó la viabilidad de la propuesta para un contexto real?	<input type="checkbox"/>	<input checked="" type="checkbox"/>

## 6. VALORAR

- Califica el impacto que representa la propuesta de solución ante la situación planteada en el caso práctico.

CRITERIO DE EVALUACIÓN	DESCRIPCIÓN DEL CRITERIO	PUNTUACIÓN MÁXIMA	PUNTAJE CALIFICADO POR EL ESTUDIANTE
Identificación del problema	Claridad en la identificación del problema planteado.	3	
Relevancia de la propuesta de solución	La propuesta responde adecuadamente al problema planteado y es relevante para el contexto del caso práctico.	8	
Viabilidad técnica	La solución es técnicamente factible, tomando en cuenta los recursos y conocimientos disponibles.	6	
Cumplimiento de Normas	La solución cumple con todas las normas técnicas de seguridad, higiene y medio ambiente.	3	
<b>PUNTAJE TOTAL</b>		<b>20</b>	

