

DOCUMENTACIÓN DEL PROYECTO CSS



Índice:

1. Introducción: quienes sois
2. Requisitos del proyecto
3. Distribución de Tareas
4. Herramientas utilizadas
5. Diagramas
6. XML
7. Documentación JS
8. Evolución del tablero Kanban
9. Descripción de arquitectura y código
10. Conclusiones miembros del equipo

1. Introducción: quienes sois

Somos el grupo CSS, conformado por Robert Munteanu, Hugo Bisbal, Raúl Herrar y Alan Castillo. En esta presentación, les explicaremos el desarrollo de nuestro proyecto: la creación de una página web funcional y bien estructurada.

Nuestro principal objetivo es diseñar y desarrollar una plataforma web que no solo sea visualmente atractiva, sino también intuitiva y eficiente en su funcionamiento. Para lograrlo, hemos aplicado conocimientos de diseño, programación y usabilidad web, asegurando que el sitio sea accesible y ofrezca al usuario funcionalidades mas allá de la simple búsqueda como el uso de favoritos.

A lo largo de esta exposición, detallaremos cada fase del proceso, desde la planificación hasta la implementación, destacando los retos que enfrentamos y las soluciones que aplicamos. ¡Esperamos que nuestra presentación sea de su interés!

2. Requisitos del proyecto

El objetivo principal de este proyecto es desarrollar y presentar una página web dinámica utilizando HTML, CSS y JavaScript, que permita la visualización de datos de manera estructurada y eficiente. Para lograrlo, hemos implementado una arquitectura que integra diferentes tecnologías con el fin de ofrecer una experiencia de usuario fluida y funcional.

Uno de los aspectos clave de este desarrollo es la utilización de una API previamente traducida a XML, la cual servirá como fuente de datos para nuestro sistema. Esta API ha sido cargada en una base de datos SQL alojada en un servidor local, lo que nos permite gestionar y almacenar la información de manera organizada y optimizada para futuras consultas.

Además, mediante el uso de consultas SQL, nuestra página web podrá extraer, filtrar y mostrar la información de forma dinámica y en tiempo real, garantizando que los usuarios accedan a los datos de manera rápida y efectiva.

Este proyecto no solo busca demostrar habilidades en el desarrollo web y gestión de bases de datos, sino también integrar múltiples tecnologías para construir una solución funcional que combine diseño, interactividad y manejo eficiente de la información.

3. Distribución de Tareas

Aunque en nuestro grupo queremos que todos formemos parte en mayor o menor medida de todos los campos del proyecto, sabemos que esto no es lo mas eficiente ni mucho menos por lo que, aunque todos vamos a revisar el trabajo de todos los integrantes, a dar nuestras opiniones y a realizar modificaciones en otros ámbitos, hemos optado por segmentar el proyecto y hacer responsable a cada uno de nosotros de cada parte en la que hemos segmentado el proyecto:

En el front-end realizando la mayor parte de los cambios usando HTML, CSS Y JS, estarán tanto Robert como Raúl, creando la mayor parte de el lado visual del proyecto web y buscando las diferentes maneras para que sea más atractiva visualmente y se pueda adaptar de forma correcta con nuestra API, que en este caso es Marvel.

En la parte más técnica como de documentación estamos tanto Alan Castillo como Hugo Bisbal, siendo Hugo el responsable de la creación de los diferentes diagramas en los que hemos basado la utilidad de nuestro proyecto y siendo Alan el responsable de la creación de la base de datos y sus diferentes diagramas. Posteriormente los dos trabajaran para integrar el archivo xml sacado de la API para así cargarlo en la base de datos creada en el sqlworkbench y que esta se conecte con nuestra página web.

Finalmente agregar que como se ha comentado con anterioridad, todos los integrantes participaremos en diferentes tareas por lo que es normal ver que por ejemplo hay tareas que Raúl ha podida realizar sobre la base de datos y tareas que por ejemplo Alan ha hecho en la parte de JavaScript.

4. Herramientas utilizadas

A lo largo del desarrollo del proyecto, hemos empleado una serie de herramientas y tecnologías que nos han permitido organizar, desarrollar y ejecutar cada parte de la aplicación web de manera eficiente.

Para la gestión del proyecto y organización del equipo, hemos utilizado Jira, una herramienta que nos ha permitido mantener un control constante sobre las tareas, su progreso y asignación a los diferentes miembros del grupo, así como realizar un seguimiento claro del estado de cada parte del desarrollo.

En cuanto al desarrollo front-end, se ha hecho uso de HTML, CSS y JavaScript, tecnologías esenciales para la estructura, el diseño y la interactividad de nuestra página web.

- Con HTML se ha estructurado el contenido.
- Con CSS se ha trabajado el diseño y estilo visual, cuidando aspectos como la paleta de colores, tipografías y disposición de elementos.
- Con JavaScript se ha implementado la lógica del lado del cliente, incluyendo funcionalidades como la interacción con el usuario, la conexión con la API y la gestión dinámica del contenido.

Para el back-end y gestión de datos, se ha utilizado:

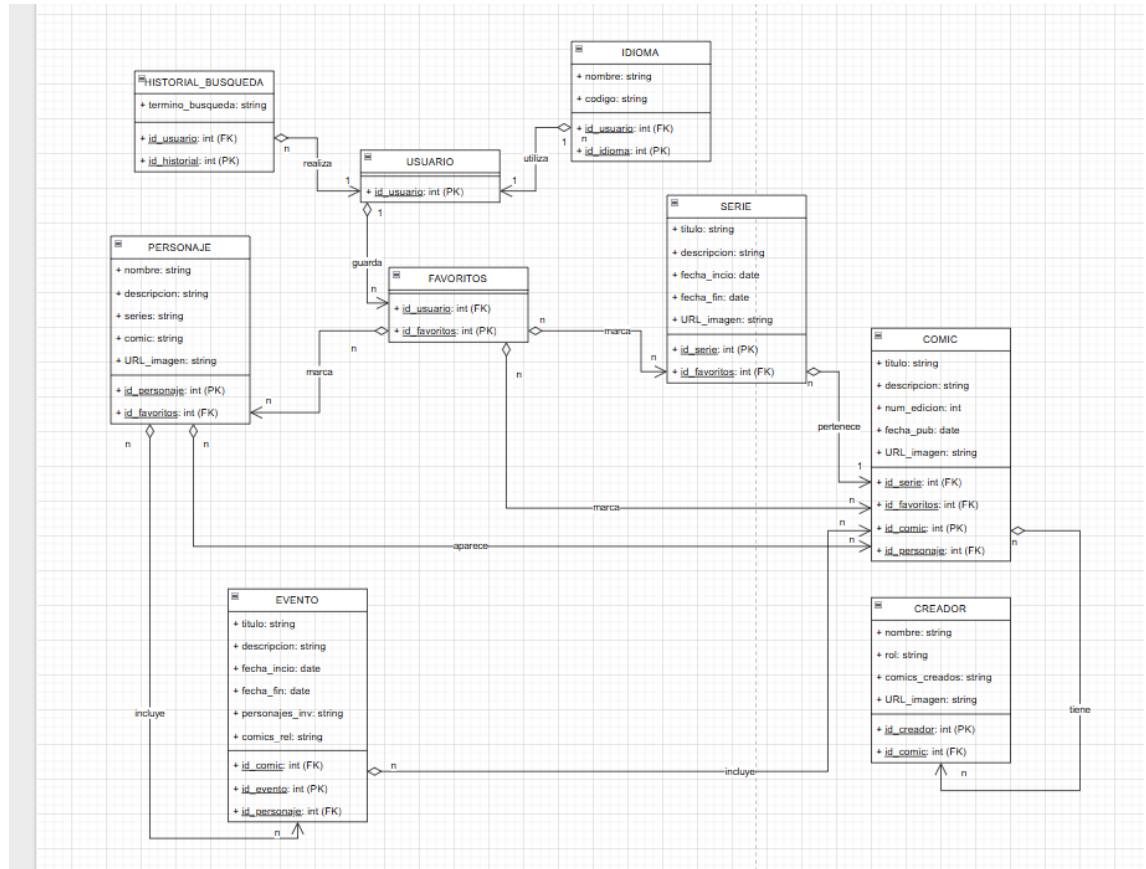
- Java, como lenguaje de programación para desarrollar la lógica de negocio y facilitar la conexión entre la base de datos y el front-end.
- SQL, para la creación y gestión de la base de datos en la que se almacena la información extraída de la API traducida a XML. La base de datos se ha gestionado utilizando MySQL Workbench, lo cual ha facilitado la administración y la visualización de las tablas y relaciones.

Además, se ha trabajado con archivos en formato XML, que contienen los datos originales provenientes de la API de Marvel. Estos han sido adaptados e integrados dentro de la base de datos para poder ser consultados y mostrados de forma dinámica en la web.

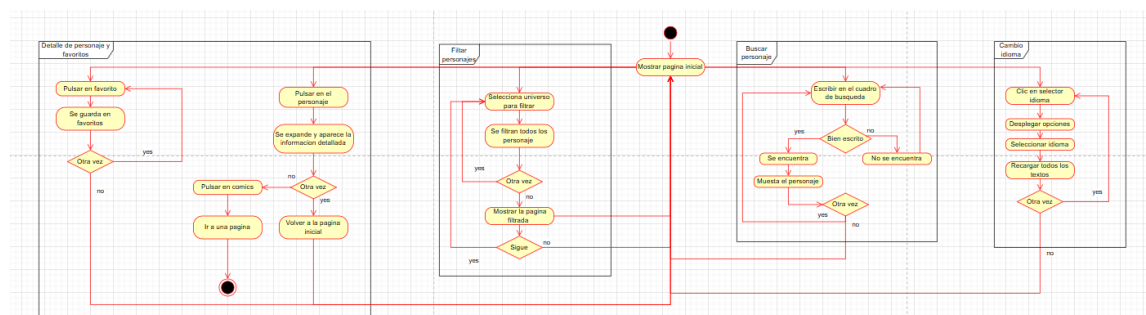
5. Diagramas

Para organizar y representar de forma clara la estructura y funcionamiento interno de nuestro proyecto, hemos elaborado una serie de diagramas UML que nos han servido de guía durante las fases de diseño y desarrollo.

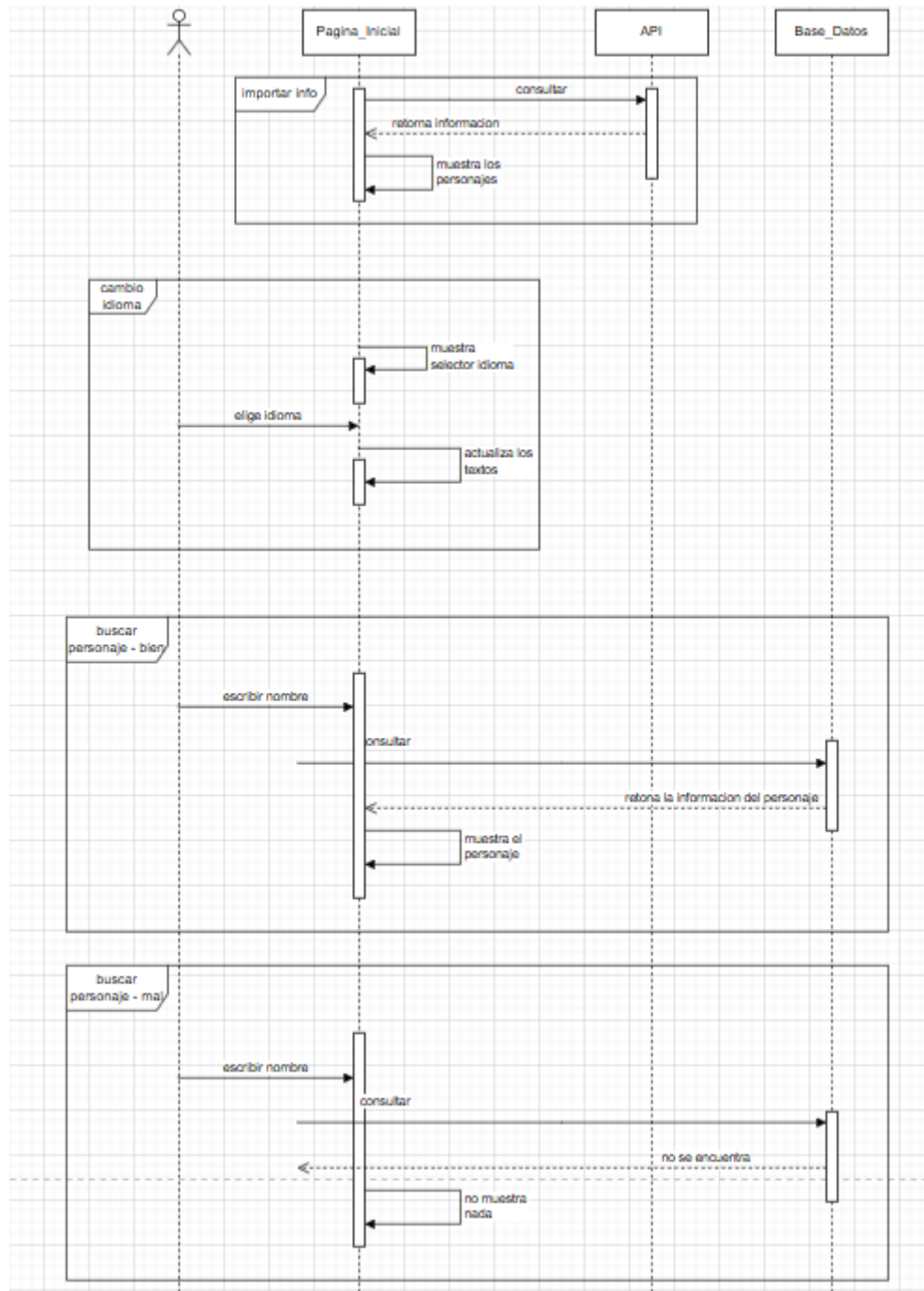
- **Diagrama de Clases:** Refleja las clases principales utilizadas en nuestro proyecto, así como sus atributos, métodos y relaciones. Este diagrama ha sido esencial para definir la arquitectura del código en Java y entender cómo interactúan las distintas partes del sistema.

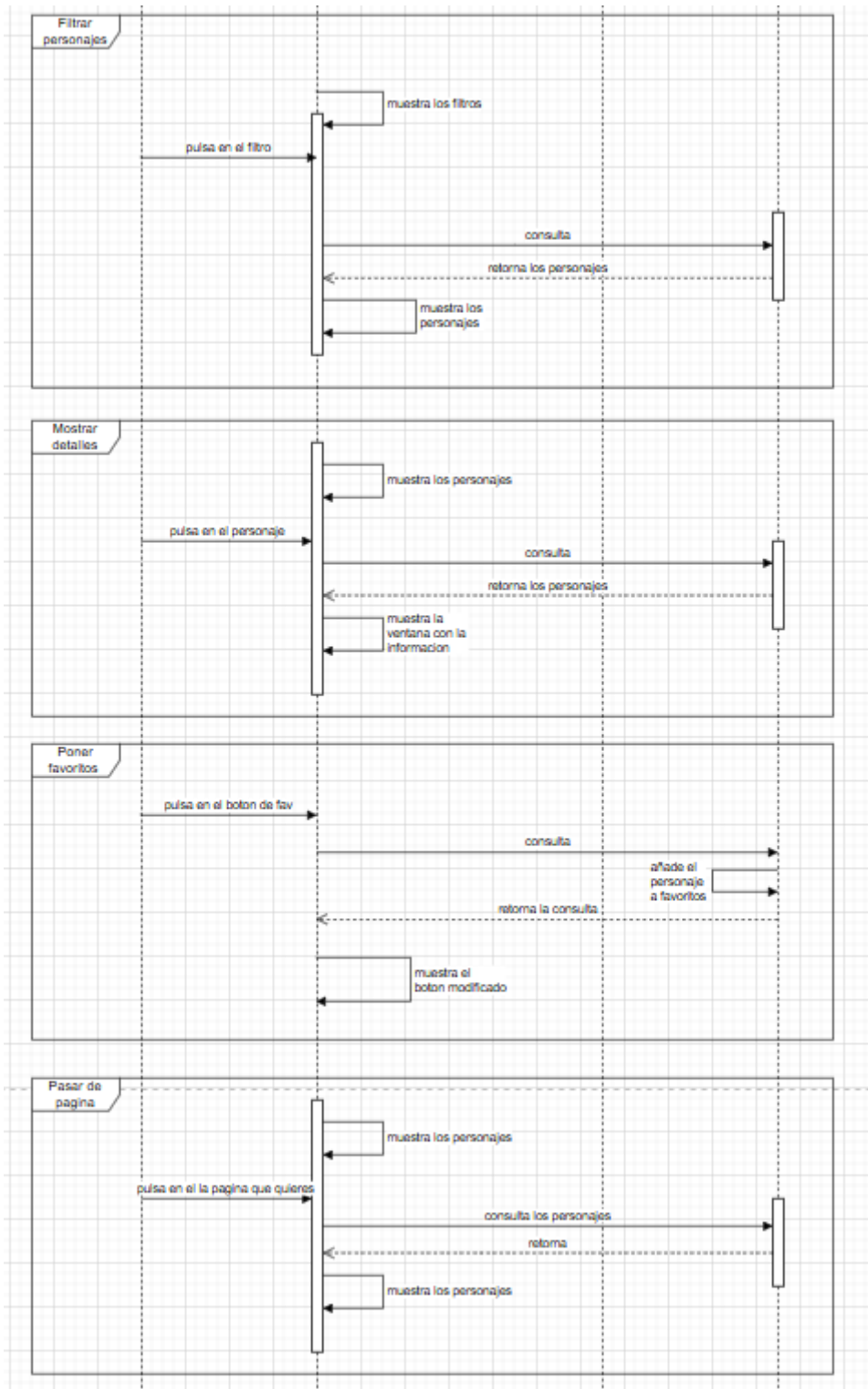


- **Diagrama de Actividades:** Representa el flujo de operaciones y decisiones que realiza el usuario dentro de la aplicación, permitiendo visualizar el comportamiento dinámico del sistema. Nos ha ayudado especialmente a planificar las rutas del usuario dentro de la página web.



- Diagrama de Secuencia:** Muestra cómo se comunican los diferentes componentes del sistema a lo largo del tiempo, desde que se realiza una solicitud hasta que se genera una respuesta. Este diagrama ha sido clave para entender la interacción entre el front-end, el back-end y la base de datos.





Estos tres diagramas no solo han servido como apoyo visual para el equipo, sino que han facilitado la planificación y la detección de posibles errores antes de pasar a la fase de codificación.

6. XML

Uno de los pasos clave en el desarrollo de nuestro proyecto ha sido el tratamiento y conversión de los datos obtenidos desde la API de Marvel. Dicha API nos proporciona la información en formato JSON, por lo que fue necesario realizar una conversión a formato XML, que es el tipo de archivo requerido para su posterior integración con nuestra base de datos.

Para ello, se ha implementado una estructura en JavaScript que permite transformar los datos JSON en un archivo XML con la jerarquía y etiquetas adecuadas. Esta conversión no solo fue fundamental para mantener un formato estándar, sino también para facilitar la carga del contenido en la base de datos.

Una vez generado el archivo XML, se utilizó como fuente para alimentar la base de datos SQL que sirve de núcleo a nuestra página web. Gracias a esta integración, la información que inicialmente estaba disponible en la API ahora puede ser consultada directamente desde nuestro servidor en local, mejorando el rendimiento y permitiendo un mayor control sobre los datos mostrados al usuario.

Este proceso ha demostrado la importancia de manejar distintos formatos de datos y adaptarlos según las necesidades técnicas del proyecto, conectando eficazmente el trabajo del front-end con la lógica del back-end y la estructura de datos.

```
<?xml version="1.0" encoding="UTF-8" ?>
<characters>
  <id_persona>101134</id_persona>
  <nombre>B. Bam</nombre>
  <description>
    <description>B. Bam</description>
  </description>
  <url_image>http://i.annihil.us/u/prod/marvel/i/mg/c/eb/535fecbb0704.jpg</url_image>
</characters>
  <id_persona>1017100</id_persona>
  <nombre>A-Bomb</nombre>
  <description>
    <description>A-Bomb has been Hulk's best bud since day one, but now he's more than a friend...he's a teammate! Transformed by a gamma energy explosion, A-Bomb's thick, armored skin is just as strong and powerful as it is blue. And when he curls into action, he uses it like a giant bowling ball of destruction!</description>
  </description>
  <url_image>http://i.annihil.us/u/prod/marvel/i/mg/3/20/521235d6e916.jpg</url_image>
</characters>
  <id_persona>1009144</id_persona>
  <nombre>A.I.M.</nombre>
  <description>
    <description>AIM is a terrorist organization bent on destroying the world.</description>
  </description>
  <url_image>http://i.annihil.us/u/prod/marvel/i/mg/6/20/5260f21f2bec.jpg</url_image>
</characters>
  <id_persona>1010009</id_persona>
  <nombre>Aaron Stack</nombre>
  <description>
    <description>http://i.annihil.us/u/prod/marvel/i/mg/b/40/image_not_available.jpg</url_image>
  </description>
</characters>
  <id_persona>1009146</id_persona>
  <nombre>Abomination (Bill Blonsky)</nombre>
  <description>
    <description>formerly known as Bill Blonsky, a spy of Soviet Yugoslavian origin working for the KGB, the Abomination gained his powers after receiving a dose of gamma radiation similar to that which transformed Bruce Banner into the Incredible Hulk.</description>
  </description>
  <url_image>http://i.annihil.us/u/prod/marvel/i/mg/9/50/4ce18091cf04.jpg</url_image>
</characters>
  <id_persona>1016023</id_persona>
  <nombre>Abomination (Ultimate)</nombre>
  <description>
    <description>http://i.annihil.us/u/prod/marvel/i/mg/b/40/image_not_available.jpg</url_image>
  </description>
</characters>
  <id_persona>1009148</id_persona>
  <nombre>Absorbing Man</nombre>
  <description>
    <description>http://i.annihil.us/u/prod/marvel/i/mg/1/b0/5269670709fb7.jpg</url_image>
  </description>
</characters>
  <id_persona>1009149</id_persona>
  <nombre>Abyss</nombre>
  <description>
    <description>http://i.annihil.us/u/prod/marvel/i/mg/3/30/535feb462a64.jpg</url_image>
  </description>
</characters>
  <id_persona>1010003</id_persona>
  <nombre>Abyss (Age of Apocalypse)</nombre>
  <description>
    <description>http://i.annihil.us/u/prod/marvel/i/mg/3/00/4c00358ec7540.jpg</url_image>
  </description>
</characters>
  <id_persona>1011266</id_persona>
  <nombre>Adam Destine</nombre>
</characters>
```

7. Documentación JS

La mayor parte del desarrollo técnico de nuestro proyecto se ha realizado utilizando **JavaScript**, siendo este el lenguaje principal tanto para la lógica del lado del cliente como para la manipulación dinámica del contenido, la comunicación con la base de datos y la interacción con la API.

Durante el proceso de programación, se ha hecho un esfuerzo por mantener una **documentación clara, ordenada y funcional del código JavaScript**, lo cual ha sido fundamental para facilitar la comprensión entre los distintos miembros del equipo y garantizar la mantenibilidad del proyecto.

Cada función y bloque de código cuenta con **comentarios explicativos** que describen su propósito, entradas, salidas y posibles excepciones. Además, se ha seguido una estructura coherente de nombres de variables y funciones, lo que ha ayudado a mantener la limpieza y legibilidad del código.

Gracias a esta correcta documentación y uso de JavaScript, hemos podido integrar de manera eficiente todos los elementos del sistema:

- Conexión con la base de datos a través de llamadas
- Conversión del JSON a XML.
- Implementación de funcionalidades dinámicas como el guardado de favoritos o los filtros de búsqueda.

En resumen, JavaScript ha sido una herramienta clave en el éxito del proyecto, y su adecuada documentación ha permitido que el desarrollo sea más fluido, colaborativo y profesional.

```

// Genera hash para autenticación de la API
async function generateHash(ts) {
  await loadCryptoJS();
  return CryptoJS.MD5(ts.toString() + privateKey + publicKey).toString();
}

// Crea un selector de idioma mejorado
function createLanguageSelector() {
  if (document.getElementById('language-selector-improved')) {
    return document.getElementById('language-selector-improved');
  }

  const existingSelector = document.getElementById('language-selector');
  let parent = existingSelector ? existingSelector.parentNode : null;

  if (!parent) {
    const header = document.querySelector('header');
    const newContainer = document.createElement('div');
    newContainer.id = 'language-selector';
    newContainer.className = 'language-selector-wrapper';

    if (header) {
      header.appendChild(newContainer);
    } else {
      document.body.insertBefore(newContainer, document.body.firstChild);
    }

    parent = newContainer.parentNode;
  }

  const container = document.createElement('div');
  container.className = 'language-selector-container';

  const select = document.createElement('select');
  select.id = 'language-selector-improved';
  select.className = 'language-selector';

  const languages = [
    { code: 'es', name: 'Español', flag: 'ES' },
    { code: 'en', name: 'English', flag: 'GB' },
    { code: 'fr', name: 'Français', flag: 'FR' },
    { code: 'ro', name: 'Română', flag: 'RO' }
  ]

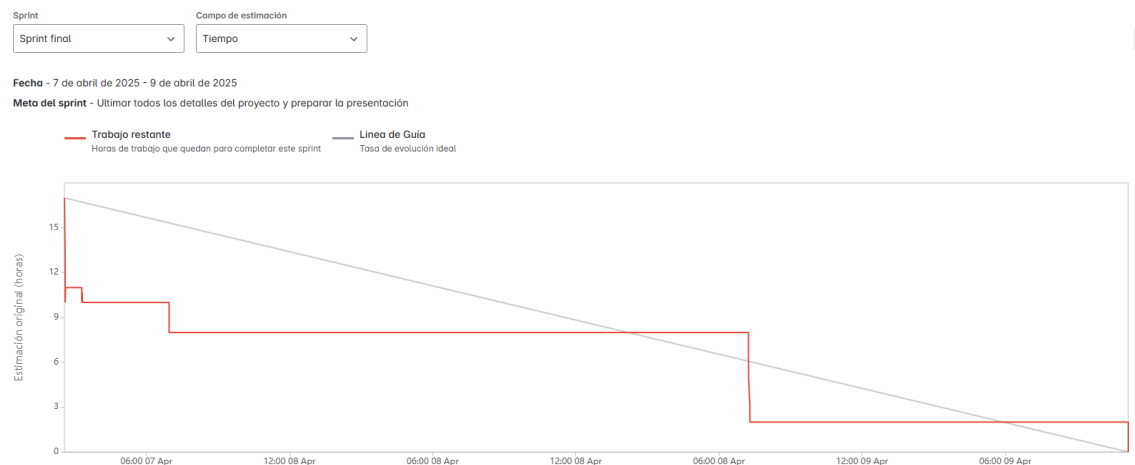
```

8. Evolución del tablero Kanban

Para llevar un control eficiente sobre el desarrollo del proyecto y garantizar una buena organización dentro del equipo, hemos utilizado un **tablero Kanban** implementado directamente en la plataforma **Jira**.

Este tablero nos ha permitido **visualizar de forma clara el estado de cada tarea**, así como su asignación a los distintos miembros del grupo. Desde el inicio del proyecto, el Kanban se estructuró en varias columnas que reflejan el flujo de trabajo que hemos seguido:

- **Pendiente**
- **En proceso**
- **Finalizado**



Cada tarea ha sido registrada con su descripción correspondiente, prioridad, fecha límite y miembro responsable. Gracias a esta metodología, hemos podido:

- Mantener una visión general del avance del proyecto.
- Identificar rápidamente cuellos de botella o tareas atascadas.
- Reasignar funciones en función de la carga de trabajo.
- Mejorar la comunicación interna del grupo.

El uso de Jira y su tablero Kanban ha sido clave para dividir correctamente las tareas, distribuir el trabajo de forma equitativa y asegurarnos de cumplir con los tiempos de entrega.

9. Descripción de arquitectura y código

La arquitectura de nuestro proyecto ha sido diseñada con una clara división entre las distintas capas: presentación (front-end), lógica (JavaScript/Java) y persistencia de datos (SQL). Esto nos ha permitido trabajar de forma ordenada, reutilizar componentes y facilitar el mantenimiento del código.

A continuación, se incluirán imágenes representativas del código utilizado en los distintos lenguajes y tecnologías implementadas. Estas capturas nos ayudarán a ilustrar cómo está estructurado el proyecto y explicar el papel que juega cada parte de este.

Código HTML

En el HTML se ha definido la estructura base de la página web, organizando el contenido en secciones claras y accesibles.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Marvel Universe Explorer</title>
  <link rel="icon" type="image/x-icon" href="marvelfav.png">
  <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css" rel="stylesheet">
  <script src="https://cdnjs.cloudflare.com/ajax/libs/crypto-js/3.1.9-1/crypto-js.js"></script>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <!-- La sección `<header>` está creando el encabezado de una página web para la aplicación Marvel Universe Explorer. -->
  <header>
    <div class="idioma">
      <label for="language-selector" data-i18n="idioma"></label>
      <select id="language-selector">
        <option value="es">Español</option>
        <option value="en">English</option>
        <option value="fr">Français</option>
        <option value="ro">Română</option>
      </select>
    </div><br><br>
    <div class="header-content container">
      <h1 class="marvel-logo" data-i18n="header.title"></h1>
      <p class="subtitle" data-i18n="header.subtitle"></p>
    </div>
    <div class="barrabuscador">
      <input type="text" id="buscador" data-i18n-placeholder="search.placeholder" />
      <button type="button" id="search" data-i18n="search.button"></button>
    </div>
  </header>
```

Código CSS

Con CSS hemos trabajado el diseño visual de la página, creando una interfaz atractiva, moderna y coherente con los objetivos del proyecto.

```
:root {
  --rojito: #e62429;
  --oscurito: #202020;
  --casi-negro: #151515;
  --blanquito: #f0f0f0;
}

* {
  box-sizing: border-box;
  margin: 0px;
  padding: 0px;
}

body {
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
  background-color: var(--casi-negro);
  color: var(--blanquito);
}

main {
  margin-bottom: 300px;
}

.container {
  margin: 0 auto;
  padding: 0 1rem;
}

header {
  background: linear-gradient(135deg, #000000, #2d2d2d);
  padding: 2rem 0;
  position: relative;
}

.header-content {
  position: relative;
  z-index: 2;
  text-align: center;
}

.marvel-logo {
  font-size: 3.5rem;
  font-weight: 900;
  color: var(--blanquito);
  text-transform: uppercase;
  letter-spacing: 2px;
  margin-bottom: 5px;
  text-shadow: 3px 3px 0px rgba(0, 0, 0, 0.8);
}
```

Código Java

En java hemos utilizado lo ofrecido por el profesor (el jdbc) para implementar la base de datos en el proyecto y cargar el servidor local.

```
public class ServidorMarvel {
    public static void main(String[] args) throws Exception {
        // Crear servidor en el puerto 8080
        HttpServer server = HttpServer.create(new java.net.InetSocketAddress(port:8080), backlog:0);

        // Endpoints REST
        server.createContext(path:"/api/characters", new CharactersHandler());
        server.createContext(path:"/api/comics", new ComicsHandler());
        server.createContext(path:"/api/series", new SeriesHandler());
        server.createContext(path:"/api/events", new EventsHandler());
        server.createContext(path:"/api/favorites", new FavoritesHandler());

        // Endpoint para verificar si la API está funcionando
        server.createContext(path:"/api", new ApiStatusHandler());

        // Endpoint para acceso a archivos estáticos (opcional)
        server.createContext(path:"/", new StaticFileHandler());

        server.start();
        System.out.println(x:"🚀 Servidor Marvel API iniciado en http://localhost:8080");
        System.out.println(x:"✅ Endpoints disponibles:");
        System.out.println(x:"  - /api/characters - Obtener personajes");
        System.out.println(x:"  - /api/comics - Obtener cómics");
        System.out.println(x:"  - /api/series - Obtener series");
        System.out.println(x:"  - /api/events - Obtener eventos");
        System.out.println(x:"  - /api/favorites - Gestionar favoritos");
    }

    // Clase auxiliar para verificar el estado de la API
    static class ApiStatusHandler extends BaseHandler {
        @Override
        public void handle(HttpExchange exchange) throws IOException {
            if ("GET".equals(exchange.getRequestMethod())) {
                String jsonResponse = "{\"message\":\"API de Marvel conectada correctamente\"}";
                sendJsonResponse(exchange, jsonResponse);
            } else if ("OPTIONS".equals(exchange.getRequestMethod())) {
                handleOptionsRequest(exchange);
            } else {
                sendError(exchange, code:405, message:"Método no permitido");
            }
        }
    }
}
```

Código SQL

El código SQL ha sido esencial para crear y gestionar la base de datos. Se han creado tablas bien estructuradas, relaciones entre ellas y consultas eficientes para mostrar los datos relevantes en la web.

```
1 • CREATE DATABASE Marvel_BBDD;
2
3 • use Marvel_BBDD;
4
5 -- Tabla USUARIO
6 • CREATE TABLE USUARIO (
7     id_usuario INT PRIMARY KEY,
8     nombre VARCHAR(100),
9     email VARCHAR(100)
10 );
11
12 -- Tabla HISTORIAL_BUSQUEDA
13 • CREATE TABLE HISTORIAL_BUSQUEDA (
14     id_historial INT PRIMARY KEY,
15     id_usuario INT,
16     termino_busqueda VARCHAR(255),
17     fecha_busqueda TIMESTAMP,
18     FOREIGN KEY (id_usuario) REFERENCES USUARIO(id_usuario)
19 );
20
21 -- Tabla IDIOMA
22 • CREATE TABLE IDIOMA (
23     id_idioma INT PRIMARY KEY,
24     nombre VARCHAR(100),
25     codigo VARCHAR(10)
26 );
27
28 -- Tabla SERIE
29 • CREATE TABLE SERIE (
30     id_serie INT PRIMARY KEY,
31     nombre VARCHAR(255),
```


10. Conclusión personal de los miembros del equipo

A lo largo de este proyecto, uno de los aspectos más destacados ha sido la mejora del trabajo en equipo. Aunque nos hemos enfrentado a varios retos y problemas a lo largo del proceso, como diferencias de opiniones sobre la estructura o dificultades técnicas, hemos aprendido a colaborar de manera más eficiente, a comunicarnos de forma más clara y a encontrar soluciones conjuntas.

Pese a los obstáculos, hemos logrado avanzar en el desarrollo del proyecto y cumplir con los objetivos propuestos desde el inicio. El proyecto nos ha permitido conocer mejor nuestras fortalezas y debilidades como equipo y ha mejorado nuestra capacidad para tomar decisiones y gestionar el tiempo y las tareas de forma más organizada.

Estamos contentos con el resultado final y creemos que hemos conseguido el objetivo de este proyecto, que era desarrollar una página web funcional, intuitiva y bien estructurada. Además, hemos adquirido habilidades valiosas en el uso de diversas tecnologías y herramientas, como JavaScript, HTML, CSS, SQL y Jira, que sin duda serán útiles en proyectos futuros.

En resumen, este proyecto ha sido una experiencia enriquecedora tanto a nivel técnico como personal, y estamos satisfechos con lo que hemos logrado.