DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

PROGRAMACIÓN INTEGRATIVA

Nombre

Jean Michael Saltos Palacios.

NRC

16496

Fecha

20/06/2024

Tarea 1.2

1. Introducción

Este informe describe el desarrollo de una aplicación web para la gestión de una biblioteca, utilizando los principios de la Programación Orientada a Objetos (POO) en JavaScript. La aplicación permite a los usuarios agregar libros con sus respectivos autores y capítulos, y visualizar la lista de libros disponibles.

2. Objetivos

- Implementar una aplicación web utilizando OOP en JavaScript.
- Aplicar los conceptos de generalización, asociación, agregación y composición.
- Desarrollar una interfaz de usuario elegante y funcional.
- Proveer un manual detallado del proyecto.

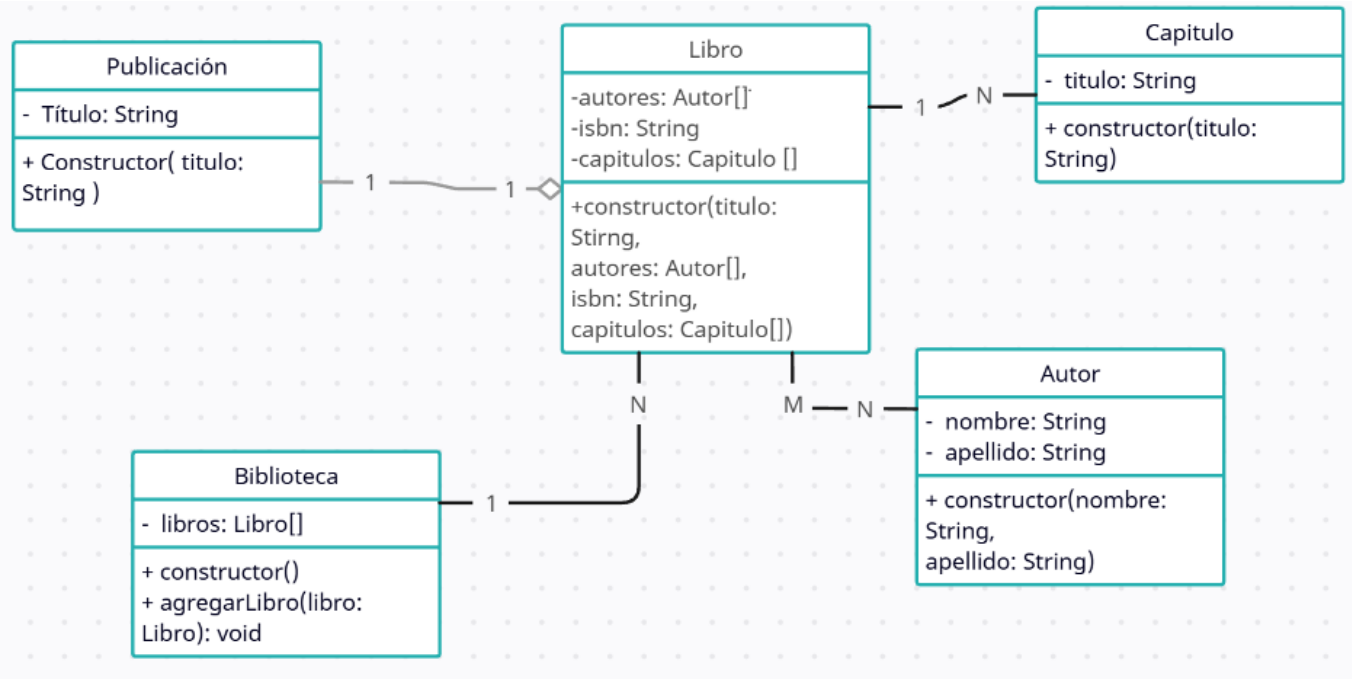
3. Modelo de Clases

Clases y Relaciones

1. **Publicacion**: Clase base para todas las publicaciones.
    - Propiedades: `titulo`
2. **Libro**: Representa un libro en la biblioteca y hereda de `Publicacion`.
    - Propiedades: `titulo`, `autores`, `isbn`, `capitulos`
3. **Autor**: Representa un autor que puede haber escrito varios libros.
    - Propiedades: `nombre`, `apellido`
4. **Capitulo**: Representa un capítulo de un libro.
    - Propiedades: `titulo`
5. **Biblioteca**: Representa una colección de libros.
    - Propiedades: `libros`

Relaciones

- **Generalización**: `Libro` hereda de `Publicacion`.
- **Asociación**: `Libro` se asocia con `Autor`.
- **Agregación**: `Biblioteca` contiene una colección de `Libro`.
- **Composición**: `Libro` contiene una colección de `Capitulo`.



4. Estructura del Proyecto

```
Tarea1.2/
├── index.html
├── css/
│   └── styles.css
└── js/
    └── app.js
```

5. Implementación del Proyecto

HTML

El archivo `index.html` define la estructura básica de la página web.

TAREA1.2 > index.html > ...

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Gestión de Biblioteca</title>
    <link rel="stylesheet" href="css/styles.css">
</head>
<body>
    <div class="container">
        <h1>Gestión de Biblioteca</h1>
        <div class="form-container">
            <h2>Agregar Libro</h2>
            <form id="formLibro">
                <input type="text" id="titulo" placeholder="Título" required>
                <input type="text" id="autores" placeholder="Autores (separados por coma)" required>
                <input type="text" id="isbn" placeholder="ISBN" required>
                <input type="text" id="capitulos" placeholder="Capítulos (separados por coma)" required>
                <button type="submit">Agregar Libro</button>
            </form>
        </div>

        <div class="list-container">
            <h2>Libros Disponibles</h2>
            <ul id="listaLibros"></ul>
        </div>
    </div>
    <script src="js/app.js"></script>
</body>
```

CSS

El archivo `css/styles.css` mejora la apariencia de la aplicación.

```css
body {
    font-family: 'Arial', sans-serif;
    background-color: #f4f4f4;
    margin: 0;
    padding: 0;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
}

.container {
    background-color: #fff;
    padding: 20px;
    border-radius: 8px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    width: 80%;
    max-width: 600px;
}
```

JavaScript

El archivo `js/app.js` contiene la lógica de la aplicación.

```javascript
document.addEventListener('DOMContentLoaded', () => {
    const biblioteca = new Biblioteca();

    const formLibro = document.getElementById('formLibro');
    const listaLibros = document.getElementById('listaLibros');

    formLibro.addEventListener('submit', (e) => {
        e.preventDefault();
        const titulo = document.getElementById('titulo').value;
        const autoresTexto = document.getElementById('autores').value;
        const isbn = document.getElementById('isbn').value;
        const capitulosTexto = document.getElementById('capitulos').value;

        const autores = autoresTexto.split(',').map(nombre => {
            const [nombreAutor, apellidoAutor] = nombre.trim().split(' ');
            return new Autor(nombreAutor, apellidoAutor);
        });

        const capitulos = capitulosTexto.split(',').map(nombre => new Capitulo(nombre.trim()));

        const nuevoLibro = new Libro(titulo, autores, isbn, capitulos);
        biblioteca.agregarLibro(nuevoLibro);

        actualizarListaLibros();
    });
```

6. Ejemplo de Uso

1. **Abrir el Proyecto**: Abre el archivo `index.html` en tu navegador.
2. **Agregar un Libro**:
    - Rellena los campos "Título", "Autores" (separados por coma), "ISBN" y "Capítulos" (separados por coma).
    - Haz clic en "Agregar Libro".
3. **Verificar la Lista de Libros**: Los libros agregados se mostrarán en la sección "Libros Disponibles".

7. Conclusión

Este proyecto demuestra cómo implementar una aplicación web de gestión de biblioteca utilizando los principios de la Programación Orientada a Objetos (OOP) en JavaScript. Hemos cubierto la creación de clases y la implementación de relaciones de generalización, asociación, agregación y composición. Este enfoque modular facilita la expansión y el mantenimiento del código.