



Universidad Internacional del Ecuador

Carrera: Ingeniería en Software Modalidad En Línea
Materia: PROGRAMACION ORIENTADA A OBJETOS
Estudiante: SALAZAR CADENA JHONNATAN FRANCISCO
Fecha: 25 de enero del 2026

SISTEMA DE GESTIÓN DE LIBROS ELECTRÓNICOS

Docente: PALACIOS MOROCHO MILTON RICARDO

Tabla de contenido

SISTEMA DE GESTIÓN DE LIBROS ELECTRÓNICOS	1
Introducción.....	3
Descripción General del Sistema.....	3
Objetivos del sistema.....	4
Objetivo General.....	4
Objetivos Específicos	4
Alcance del Proyecto	4
Alcance Funcional	4
Alcance No Funcional	5
Módulos del sistema	5
Módulo de Gestión de Libros	5
Módulo de Persistencia de Datos	6
Módulo de Interfaz de Usuario	7
Estructura del sistema basada en programación Orientada a Objetos	8
Uso de Estructuras (Structs) como Entidades del Sistema.....	8
Separación de Responsabilidades mediante Paquetes	8
Encapsulación de la Lógica del Sistema	9
Relación entre Programación Orientada a Objetos y Programación Funcional	9
Beneficios del Diseño Orientado a Objetos en el Sistema	10
Modelo de Datos: Estructura Libro	10
Identificación y Documentación de Paquetes Utilizados	11
Paquetes Estándar del Lenguaje Go	11
Paquetes Propios del Proyecto.....	12
Paquetes de Terceros	13
Gestión de Dependencias.....	13
Importancia de la Documentación de Paquetes	13
Enfoque de Programación Funcional	13
Conclusiones.....	14
Bibliografía.....	14
URL web repositorio de Github	15



Introducción

En la actualidad, la gestión de información digital se ha convertido en un elemento fundamental dentro de las organizaciones y entornos educativos. El crecimiento acelerado del contenido digital ha generado la necesidad de contar con sistemas que permitan organizar, almacenar y administrar información de manera eficiente y segura. En este contexto, los libros electrónicos representan una de las formas más comunes de distribución del conocimiento, lo que hace indispensable la existencia de sistemas de gestión especializados para su control.

El presente documento describe la planeación del desarrollo de un **Sistema de Gestión de Libros Electrónicos**, el cual permitirá administrar un catálogo digital mediante operaciones básicas como registro, consulta, búsqueda, edición y eliminación de libros. Este sistema será desarrollado utilizando el lenguaje de programación **Go**, aplicando principios de **Programación Orientada a Objetos**, el uso de **paquetes** y un enfoque basado en **programación funcional**.

La planeación del software es una etapa clave dentro del ciclo de desarrollo, ya que permite definir claramente los objetivos, el alcance, la estructura del sistema y los módulos que lo conformarán. (Ramirez, 2024). A través de esta fase se busca establecer una base sólida que facilite la implementación posterior del sistema, reduciendo errores y mejorando la organización del código.

Descripción General del Sistema

El Sistema de Gestión de Libros Electrónicos es una aplicación de tipo **consola**, diseñada para facilitar la administración de información relacionada con libros digitales. El sistema permitirá almacenar datos como título, autor, categoría, año

de publicación, formato del libro y disponibilidad, utilizando un archivo en formato **JSON** como medio de persistencia de datos.

Este sistema está orientado a usuarios que necesiten llevar un control básico de una biblioteca digital, como estudiantes, docentes o pequeñas organizaciones. No se contempla la lectura directa de los libros dentro del sistema, sino únicamente la gestión de su información.

Desde el punto de vista técnico, el sistema será desarrollado en Go, aprovechando su simplicidad, eficiencia y estructura modular. Se aplicará una correcta separación de responsabilidades mediante el uso de paquetes, permitiendo que cada componente del sistema tenga una función específica.

Objetivos del sistema

Objetivo General

Desarrollar un sistema de gestión de libros electrónicos utilizando el lenguaje de programación Go, que permita administrar un catálogo digital mediante operaciones básicas de gestión, aplicando conceptos de Programación Orientada a Objetos, programación funcional y organización modular del código.

Objetivos Específicos

- Identificar los módulos necesarios para la gestión de libros electrónicos.
- Diseñar la estructura del sistema utilizando paquetes y funciones en Go.
- Implementar operaciones básicas de gestión de libros (CRUD).
- Aplicar principios de programación funcional mediante el uso de funciones independientes y reutilizables.
- Almacenar la información de los libros en un archivo JSON para garantizar la persistencia de datos.
- Facilitar la interacción del usuario con el sistema a través de un menú en consola.

Alcance del Proyecto

Alcance Funcional

El sistema incluirá las siguientes funcionalidades:

- Registro de libros electrónicos.

- Listado general de libros almacenados.
- Búsqueda de libros por título, autor o categoría.
- Edición de información de libros existentes.
- Eliminación de libros del sistema.
- Cambio del estado de disponibilidad de un libro.
- Guardado de la información en un archivo JSON.
- Carga de información desde un archivo JSON.

Alcance No Funcional

- El sistema no incluirá autenticación de usuarios.
- No se implementará una interfaz gráfica ni web.
- No se utilizará una base de datos relacional.
- No se permitirá la lectura o descarga directa de los libros.
- El sistema estará limitado a un entorno local.

Módulos del sistema

El Sistema de Gestión de Libros Electrónicos ha sido diseñado bajo una arquitectura modular, lo que permite dividir el software en componentes independientes, cada uno con responsabilidades claramente definidas. Esta organización facilita el mantenimiento del sistema, mejora la legibilidad del código y permite futuras ampliaciones sin afectar el funcionamiento general del software.

Cada módulo cumple una función específica dentro del sistema y se comunica con los demás a través de estructuras de datos y funciones, siguiendo los principios de la Programación Orientada a Objetos adaptada al lenguaje Go y un enfoque de programación funcional.

Módulo de Gestión de Libros

El Módulo de Gestión de Libros constituye el núcleo principal del sistema, ya que se encarga de administrar toda la información relacionada con los libros electrónicos. Su función principal es manejar las operaciones básicas de gestión, permitiendo al usuario registrar, modificar, eliminar y consultar los libros almacenados en el sistema.

Este módulo trabaja directamente con la estructura de datos **Libro**, la cual representa la entidad principal del sistema. A través de funciones específicas, el módulo valida la información ingresada por el usuario, asegurando que los datos cumplan con los requisitos mínimos antes de ser almacenados o modificados.

Entre las funcionalidades principales que ofrece este módulo se encuentran:

- Registro de nuevos libros electrónicos, incluyendo datos como título, autor, categoría, año de publicación, formato y disponibilidad.
- Listado completo de los libros registrados, mostrando su información de forma clara y ordenada.
- Búsqueda de libros mediante criterios como título, autor o categoría, facilitando la localización de información específica.
- Edición de los datos de un libro previamente registrado, permitiendo actualizar su información cuando sea necesario.
- Eliminación de libros del sistema cuando ya no sean requeridos.
- Cambio del estado de disponibilidad de un libro, permitiendo indicar si se encuentra disponible o no.

Desde el punto de vista técnico, este módulo implementa sus funcionalidades mediante funciones independientes que reciben como entrada una colección de libros y devuelven resultados actualizados. Este enfoque promueve la reutilización del código y reduce la dependencia entre componentes, alineándose con los principios de la programación funcional.

Módulo de Persistencia de Datos

El Módulo de Persistencia de Datos tiene como objetivo garantizar la conservación de la información del sistema entre distintas ejecuciones de la aplicación. Para ello, utiliza archivos en formato **JSON** como mecanismo de almacenamiento, permitiendo guardar y recuperar los datos de los libros electrónicos de manera estructurada y legible.

Este módulo se encarga de transformar las estructuras de datos utilizadas en el sistema en un formato que pueda ser almacenado en un archivo, así como de realizar el proceso inverso al cargar la información. De esta manera, se evita la pérdida de datos al cerrar la aplicación y se asegura la continuidad de la información registrada por el usuario.

Las principales responsabilidades de este módulo incluyen:

- Guardar la lista de libros electrónicos en un archivo JSON.
- Leer y cargar los datos almacenados en el archivo JSON al iniciar el sistema.
- Manejar posibles errores relacionados con la lectura o escritura de archivos.
- Garantizar la integridad de los datos almacenados.

El uso de JSON como formato de almacenamiento ofrece ventajas como simplicidad, portabilidad y facilidad de comprensión, lo que resulta adecuado para un sistema de gestión de alcance básico. (Erickson, 2024). Además, este módulo se mantiene independiente del resto del sistema, permitiendo cambiar el mecanismo de persistencia en el futuro sin afectar los demás módulos.

Módulo de Interfaz de Usuario

El Módulo de Interfaz de Usuario es el encargado de gestionar la interacción entre el sistema y el usuario final. En este proyecto, la interfaz se implementa mediante un menú en consola, el cual presenta de forma clara las diferentes opciones disponibles dentro del sistema.

Este módulo permite al usuario seleccionar las funcionalidades que desea ejecutar, ingresar los datos requeridos para cada operación y visualizar los resultados obtenidos. Además, se encarga de validar las opciones seleccionadas y de mostrar mensajes informativos o de error cuando sea necesario.

Entre las funciones principales de este módulo se encuentran:

- Mostrar el menú principal del sistema con todas las opciones disponibles.
- Capturar la entrada del usuario desde la consola.
- Validar las opciones seleccionadas para evitar errores de ejecución.
- Mostrar los resultados de las operaciones realizadas por los otros módulos.
- Proporcionar una experiencia de uso sencilla e intuitiva.

La separación de la interfaz de usuario del resto de la lógica del sistema permite que los cambios en la forma de interacción (por ejemplo, una futura interfaz gráfica o web) puedan realizarse sin modificar los módulos de gestión y persistencia. Esto refuerza la modularidad del sistema y facilita su evolución a largo plazo.

Estructura del sistema basada en programación Orientada a Objetos

El diseño del Sistema de Gestión de Libros Electrónicos se fundamenta en los principios de la **Programación Orientada a Objetos (POO)**, adaptados a las características propias del lenguaje de programación Go. Aunque Go no implementa la Programación Orientada a Objetos de manera tradicional como otros lenguajes orientados a clases, sí proporciona mecanismos que permiten estructurar sistemas de forma organizada y modular mediante el uso de **estructuras (structs), métodos y paquetes**.

La aplicación de estos principios permite modelar el sistema de acuerdo con el dominio del problema, facilitando la comprensión del código, su mantenimiento y su posible extensión en el futuro. En este proyecto, la POO se utiliza principalmente para representar las entidades del sistema y definir responsabilidades claras para cada componente.

Uso de Estructuras (Structs) como Entidades del Sistema

En el sistema desarrollado, las estructuras cumplen el rol de las clases dentro del paradigma orientado a objetos. La estructura principal es **Libro**, la cual representa la entidad central del sistema de gestión.

Cada estructura encapsula los atributos que describen a un libro electrónico, tales como identificador, título, autor, categoría, año de publicación, formato y disponibilidad. Esta encapsulación permite agrupar datos relacionados dentro de una misma entidad, facilitando su manipulación y reduciendo la complejidad del código.

El uso de estructuras permite crear múltiples instancias de libros, las cuales pueden ser almacenadas y gestionadas dentro de colecciones, como listas o arreglos dinámicos. De esta forma, el sistema puede manejar un conjunto variable de libros de manera eficiente.

Separación de Responsabilidades mediante Paquetes

El sistema ha sido diseñado utilizando una estructura modular basada en **paquetes**, donde cada paquete representa una parte específica de la

funcionalidad del sistema. Esta separación de responsabilidades es uno de los principios fundamentales de la Programación Orientada a Objetos y contribuye a la organización del código.

Cada paquete contiene funciones y estructuras relacionadas con una responsabilidad específica, evitando que la lógica del sistema se encuentre mezclada en un solo archivo. Esta organización mejora la legibilidad del código y facilita su mantenimiento.

Los principales paquetes definidos en el sistema son:

- **Modelo:** Contiene las estructuras que representan las entidades del sistema.
- **Gestión:** Incluye la lógica relacionada con la administración de libros.
- **Persistencia:** Maneja el almacenamiento y recuperación de datos.
- **Interfaz:** Gestiona la interacción con el usuario.

Esta estructura permite que cada paquete evolucione de forma independiente, manteniendo un bajo acoplamiento entre los componentes del sistema.

Encapsulación de la Lógica del Sistema

La encapsulación es un principio clave de la Programación Orientada a Objetos que consiste en ocultar los detalles internos de un componente y exponer únicamente lo necesario para su uso. En el sistema desarrollado, la encapsulación se logra mediante la definición de funciones específicas dentro de cada paquete.

Las funciones encargadas de gestionar los libros, por ejemplo, se encuentran aisladas dentro del módulo correspondiente, evitando que otros módulos accedan directamente a los datos internos sin pasar por las reglas establecidas. Esto permite controlar el flujo de información y reducir errores.

Además, la encapsulación facilita la implementación de validaciones y reglas de negocio, asegurando que los datos ingresados cumplan con los requisitos definidos antes de ser procesados o almacenados.

Relación entre Programación Orientada a Objetos y Programación Funcional

El diseño del sistema combina principios de la Programación Orientada a Objetos con un enfoque de **programación funcional**, aprovechando las ventajas de ambos paradigmas. Mientras la POO se utiliza para modelar las entidades y

estructurar el sistema, la programación funcional se emplea para implementar la lógica de negocio mediante funciones independientes.

Las funciones del sistema reciben estructuras como parámetros y devuelven resultados sin modificar directamente el estado global, lo que mejora la claridad del código y facilita su prueba. Esta combinación permite construir un sistema robusto, modular y fácil de mantener.

Beneficios del Diseño Orientado a Objetos en el Sistema

La aplicación de principios de Programación Orientada a Objetos en el Sistema de Gestión de Libros Electrónicos aporta múltiples beneficios, entre los cuales se destacan:

- Mejor organización y estructura del código.
- Facilidad para identificar y corregir errores.
- Posibilidad de reutilizar componentes en otros proyectos.
- Mayor facilidad para extender el sistema con nuevas funcionalidades.
- Mejor comprensión del sistema por parte de otros desarrolladores.

En conclusión, el uso de una estructura basada en Programación Orientada a Objetos, adaptada al lenguaje Go, permite desarrollar un sistema bien organizado, coherente y alineado con los objetivos académicos del proyecto.

Modelo de Datos: Estructura Libro

La estructura Libro representará la entidad principal del sistema. Sus atributos serán:

- **ID:** Identificador único del libro.
- **Título:** Nombre del libro electrónico.
- **Autor:** Autor del libro.
- **Categoría:** Clasificación del libro.
- **Año:** Año de publicación.
- **Formato:** Tipo de archivo (PDF, EPUB, etc.).
- **URL o Ruta:** Ubicación del archivo.
- **Disponible:** Estado del libro (disponible o no disponible).

Esta estructura permitirá gestionar de forma clara y organizada la información de cada libro dentro del sistema.

Identificación y Documentación de Paquetes Utilizados

Durante el desarrollo del Sistema de Gestión de Libros Electrónicos se realiza una correcta identificación de los paquetes que serán incorporados al proyecto, con el objetivo de garantizar una organización adecuada del código, facilitar su mantenimiento y cumplir con los principios de modularidad establecidos en la Programación Orientada a Objetos.

El lenguaje de programación Go promueve el uso de paquetes como una forma de agrupar código relacionado, permitiendo la reutilización de funciones y una clara separación de responsabilidades dentro del sistema. (Herrera, 2024). En este proyecto se utilizan tanto paquetes estándar del lenguaje como paquetes definidos por el propio desarrollador, sin necesidad de incorporar librerías de terceros, lo cual contribuye a mantener la simplicidad del sistema.

Paquetes Estándar del Lenguaje Go

Los paquetes estándar de Go proporcionan funcionalidades básicas necesarias para el funcionamiento del sistema. A continuación, se describen los paquetes utilizados y su propósito dentro del proyecto:

- **fmt**

Se utiliza para mostrar información en la consola y para la lectura de datos ingresados por el usuario. Este paquete es fundamental para la interacción básica entre el sistema y el usuario.

- **os**

Permite el manejo de archivos y directorios del sistema operativo. En el proyecto se emplea para crear, abrir y cerrar archivos donde se almacena la información de los libros electrónicos.

- **encoding/json**

Se utiliza para convertir las estructuras de datos del sistema a formato JSON y viceversa. Este paquete es clave para la persistencia de datos, ya que permite guardar y recuperar la información de los libros en un archivo de manera estructurada.

- **bufio**
Facilita la lectura de datos desde la entrada estándar, permitiendo capturar información ingresada por el usuario de forma eficiente y segura.
- **strings**
Se emplea para el procesamiento de cadenas de texto, como la eliminación de espacios innecesarios y la comparación de datos durante las búsquedas.
- **strconv**
Permite la conversión de datos de tipo cadena a tipos numéricos y booleanos, lo cual es necesario al procesar la información ingresada por el usuario.
- **time** *(opcional)*
Puede utilizarse para registrar la fecha de creación o modificación de un libro electrónico, aportando un mejor control de la información.

El uso de estos paquetes estándar garantiza estabilidad, seguridad y compatibilidad, ya que forman parte de la biblioteca oficial del lenguaje Go.

Paquetes Propios del Proyecto

Además de los paquetes estándar, el sistema utiliza paquetes creados específicamente para el proyecto, los cuales permiten organizar el código de acuerdo con las funcionalidades del sistema. Estos paquetes son:

- **model**
Contiene las estructuras que representan las entidades del sistema, principalmente la estructura Libro. Su función es definir el modelo de datos que será utilizado por los demás módulos.
- **catalog**
Incluye las funciones encargadas de la gestión de libros electrónicos, tales como agregar, buscar, editar y eliminar libros. Este paquete concentra la lógica de negocio del sistema.
- **storage**
Se encarga de la persistencia de datos, implementando las funciones necesarias para guardar y cargar la información desde archivos JSON.
- **ui**
Gestiona la interacción con el usuario mediante el menú de consola. Este

paquete se encarga de mostrar opciones, capturar entradas y presentar los resultados de las operaciones.

La utilización de paquetes propios permite mantener una estructura clara y coherente, facilitando la comprensión del sistema y el trabajo colaborativo.

Paquetes de Terceros

En esta etapa del desarrollo no se han incorporado paquetes de terceros, ya que el sistema puede ser implementado completamente utilizando los paquetes estándar de Go. Esta decisión responde al objetivo de enfocarse en los conceptos fundamentales del lenguaje y evitar dependencias innecesarias.

Sin embargo, el diseño modular del sistema permite que, en futuras versiones, se puedan integrar paquetes externos para ampliar sus funcionalidades. Un ejemplo de esto podría ser el uso de librerías para la generación de identificadores únicos o la implementación de interfaces gráficas.

Gestión de Dependencias

La gestión de dependencias del proyecto se realiza mediante el sistema de módulos de Go, utilizando el archivo **go.mod**, el cual permite definir el nombre del módulo y controlar las versiones de los paquetes utilizados. Esta configuración facilita la portabilidad del proyecto y asegura que el entorno de desarrollo sea consistente.

Importancia de la Documentación de Paquetes

La correcta identificación y documentación de los paquetes utilizados en el sistema es fundamental para garantizar la calidad del software. Esta práctica permite comprender el rol de cada componente, facilita el mantenimiento del sistema y asegura el cumplimiento de los requerimientos establecidos en la actividad académica.

Enfoque de Programación Funcional

El desarrollo del sistema se realizará utilizando un enfoque de **programación funcional**, basado en la creación de funciones independientes que reciben datos de entrada y devuelven resultados sin generar efectos secundarios innecesarios.

Las operaciones del sistema, como agregar, buscar o eliminar libros, serán implementadas como funciones que operan sobre estructuras de datos, promoviendo la reutilización del código y facilitando su mantenimiento.

Este enfoque permite una mejor organización del sistema y una mayor claridad en la lógica de negocio, cumpliendo con los principios vistos en la asignatura.

Conclusiones

En el desarrollo de este proyecto pude comprender la importancia de la etapa de planeación del software, ya que definir claramente los objetivos, el alcance y los módulos del sistema permitió establecer una base sólida para la construcción del Sistema de Gestión de Libros Electrónicos. Esta planificación facilitó la organización del código y redujo posibles errores durante el proceso de desarrollo.

A través de la implementación del sistema en el lenguaje Go, logré aplicar de manera práctica los conceptos de Programación Orientada a Objetos y programación funcional, utilizando estructuras, paquetes y funciones independientes. Esto me permitió entender cómo estos paradigmas pueden combinarse para crear sistemas modulares, reutilizables y fáciles de mantener, aun cuando el lenguaje no maneja la POO de forma tradicional.

Finalmente, considero que este proyecto fortaleció mis habilidades en el diseño y organización de software, especialmente en el uso de paquetes, manejo de persistencia de datos mediante archivos JSON y control de versiones con GitHub. El Sistema de Gestión de Libros Electrónicos representa una solución funcional y bien estructurada, alineada con los objetivos académicos de la asignatura y con buenas prácticas de desarrollo de software.

Bibliografía

Erickson, J. (4 de abril de 2024). *Oracle*. Obtenido de Oracle:

<https://www.oracle.com/latam/database/what-is-json/#:~:text=din%C3%A1micas%20e%20interactivas.-,%C2%BFPor%20qu%C3%A9%20se%20utiliza%20JSON?,necesidad%20de%20cambios%20de%20c%C3%B3digo>.

Herrera, J. (3 de enero de 2024). *Medium*. Obtenido de Medium:

<https://medium.com/@jaime.herrera.mz/orientaci%C3%B3n-a-objetos-en-go-b74a3a2a5a58>

Ramirez, O. (29 de julio de 2024). *Bambu Mobile*. Obtenido de Bambu Mobile:
<https://bambu-mobile.com/ciclo-de-desarrollo-de-software/#:~:text=hasta%20el%20mantenimiento.-,Planificaci%C3%B3n,desarrollar%20y%20c%C3%B3mo%20se%20lograr%C3%A1.>

URL web repositorio de Github

https://github.com/Jhonna555/Jhonna555-Sistema_de_Gestion_de_libros_Electronicos.git