



Informe Técnico Taller 5

Asignatura: Programación Avanzada

Paralelo: C2

Nombre del Profesor: Álvaro Castillo

Nombre del Ayudante: Helmer Pizarro

Nombre de los Estudiantes:

- Cristian Álvarez
- Christian Aquilar

Fecha de entrega

09 de julio, 2023



Introducción

En este informe se entregará una revisión parte por parte, clase por clase de cómo funciona el programa que se nos encomendó codificar:

“BiblioTech es una empresa que busca modernizar su sistema de préstamo de libros y gestión de la biblioteca. Para ello, ha establecido una alianza con los estudiantes de Programación Avanzada del Departamento de Ingeniería de Sistemas y Computación de la Universidad Católica del Norte ya que ya habían trabajado antes con ellos en su primer sistema. Debido a que la empresa busca mejorar su anterior sistema, se busca implementar una interfaz gráfica para que la gestión de su inventario y proceso de arriendos y devoluciones sea más rápido y efectivo. Por ese motivo les encomendaron la tarea a los estudiantes el realizar este formulario más avanzado. A continuación, se entregan los detalles del formulario: La empresa ha proporcionado a los estudiantes dos archivos de texto (.txt) donde están los datos de los usuarios (trabajadores) y libros del sistema. Adicional a esto, la empresa les entrega a los estudiantes el código para leer los archivos para que no pierdan tiempo en realizarlo.”



Clase BiblioTech:

- La clase BiblioTech extiende de JFrame, lo que significa que es una ventana de la interfaz gráfica de usuario (GUI) en Java.
- Los componentes de la GUI, como JPasswordField, JTextField, JButton y JPanel, se declaran variables de instancia en la clase.
- Dentro del constructor de la clase BiblioTech, se inicializan las listas de usuarios y libros utilizando métodos de la clase Utils (suponemos que esta clase contiene métodos para leer archivos de usuarios y libros).
- Se agregan ActionListeners a los botones abrir y cerrar. Estos ActionListeners especifican qué hacer cuando se hace clic en los botones.
- El método cerrarPrograma() se invoca cuando se hace clic en el botón "cerrar". Muestra un cuadro de diálogo de confirmación y cierra el programa si el usuario confirma.
- El método iniciarSesion() se invoca cuando se hace clic en el botón "abrir". Obtiene el RUT y la contraseña ingresados, verifica si están vacíos y luego busca un usuario en la lista de usuarios que coincida con el RUT ingresado. Si encuentra un usuario y la contraseña coincide, se muestra el menú principal llamando al método mostrarMenuPrincipal(). De lo contrario, se muestra un mensaje de error.
- El método mostrarMenuPrincipal() crea una instancia de la clase MenuPrincipal (suponemos que esta clase representa el menú principal de la aplicación) y la hace visible. Luego, se cierra la ventana actual de inicio de sesión.
- El método buscarUsuario() busca un usuario en la lista de usuarios que coincida con el RUT ingresado.
- El método mostrarMensajeError() muestra un cuadro de diálogo de error con el mensaje proporcionado.



Clase MenuPrincipal:

- La clase MenuPrincipal extiende de JFrame, lo que significa que es una ventana de la interfaz gráfica de usuario (GUI) en Java.
- Los componentes de la GUI, como JButton, JPanel y JButton, se declaran como variables de instancia en la clase.
- Se crean las listas de usuarios, libros y registros utilizando los métodos de la clase Utils (suponemos que esta clase contiene métodos para leer archivos de usuarios, libros y registros).
- Se configuran los ActionListeners para los botones del menú. Cada botón tiene asociado un método que se ejecuta cuando se hace clic en el botón correspondiente.
- El panel del menú se configura con un diseño de cuadrícula y se agregan los botones al panel
- El panel del menú se agrega al contenido de la ventana.
- El constructor de la clase MenuPrincipal recibe las listas de usuarios y libros como parámetros.
- El método verPerfil() se ejecuta cuando se hace clic en el botón "Ver Perfil". Crea una instancia de la clase VerPerfil y la hace visible, pasando la lista de registros y la instancia actual de MenuPrincipal.
- El método obtenerRegistrosUsuario() busca los registros asociados al usuario actual y devuelve una lista de registros específicos del usuario.
- El método cerrarPerfil() se ejecuta cuando se hace clic en el botón "Cerrar Sesión". Muestra un cuadro de diálogo de confirmación y, si se confirma, registra los registros y cierra el programa.
- Los métodos buscarLibro(), devolverLibro(), agregarLibro() y prestarLibro() manejan las acciones correspondientes al hacer clic en los botones del menú. Cada método muestra un cuadro de diálogo para obtener información relevante y ejecuta la lógica correspondiente utilizando los métodos de la clase Utils.
- El método buscarLibroPorISBN() busca un libro en la lista de libros que coincide con el ISBN proporcionado.
- El método devolverLibroPorISBN() busca un libro en la lista de libros que coincide con el ISBN proporcionado y, si hay copias disponibles, incrementa el stock del libro devuelto.
- El método agregarLibro() muestra cuadros de diálogo para ingresar los detalles de un nuevo libro y lo agrega a la lista de libros si no existe previamente.
- El método prestarLibroPorISBN() busca un libro en la lista de libros que coincide con el ISBN proporcionado y, si hay copias disponibles, decrementa el stock del libro prestado.



Clase BuscarLibro:

- La clase BuscarLibro extiende de JFrame, lo que significa que es una ventana de la interfaz gráfica de usuario (GUI) en Java.
- Los componentes de la GUI, como JLabel, JTextField, JButton y JList, se declaran como variables de instancia en la clase.
- El constructor de la clase BuscarLibro recibe la lista de libros disponibles y una instancia de la clase MenuPrincipal.
- Se configuran los componentes de la GUI, como el título de la ventana, el tamaño, la operación de cierre, la ubicación y el diseño del panel principal.
- Se definen los componentes de la GUI, como la etiqueta "ISBN", el campo de texto para buscar el ISBN, los botones de búsqueda y volver, y la lista para mostrar los resultados.
- Se configuran los ActionListeners para los botones de búsqueda y volver. Cuando se hace clic en el botón de búsqueda, se ejecuta el método buscarLibro(), y cuando se hace clic en el botón de volver, se ejecuta el método volverAlMenuPrincipal().
- El método buscarLibro() obtiene el ISBN ingresado en el campo de texto y busca el libro correspondiente en la lista de libros utilizando el método buscarLibroPorISBN(). Si se encuentra el libro, se muestra la información del libro en la lista.
- El método volverAlMenuPrincipal() muestra la ventana del menú principal (MenuPrincipal) y cierra la ventana de búsqueda de libros (BuscarLibro).
- El método buscarLibroPorISBN() busca un libro en la lista de libros que coincide con el ISBN proporcionado.



Clase VerPerfil:

- La clase VerPerfil extiende de JFrame, lo que significa que es una ventana de la interfaz gráfica de usuario (GUI) en Java.
- Los componentes de la GUI, como JList y JButton, se declaran como variables de instancia en la clase.
- El constructor de la clase VerPerfil recibe la lista de registros de préstamos y una instancia de la clase MenuPrincipal.
- Se configuran los aspectos de la ventana, como el título, el tamaño y la operación de cierre.
- Se crea y configura el panel principal de la GUI.
- Se crea un modelo de lista (DefaultListModel<String>) para almacenar la información del perfil del usuario.
- Se crea un componente JList utilizando el modelo de lista y se configura para permitir la selección de un solo elemento.
- Se crea un JScrollPane para agregar capacidad de desplazamiento al JList y se agrega al panel principal.
- Se crea un botón "Volver al menú principal" y se configura un ActionListener para volver a la ventana del menú principal cuando se hace clic en el botón.
- Se agregan los componentes al panel principal.
- Se agrega el panel principal a la ventana.
- Se obtiene el RUT del usuario actualmente activo (en este caso, se utiliza un RUT fijo para simplificar el ejemplo) y se busca el objeto Usuario correspondiente en la lista de usuarios utilizando el método obtenerUsuarioActual().
- Se recorren los registros de préstamo y se agrega la información relevante al modelo de lista para mostrarla en la GUI.
- Se muestra la ventana.
- (La idea era que, al clickear "Ver Perfil", se desplegara una nueva pantalla con los datos del usuario logeado, destacando lo que serían nombre, rut, etc.)



Clase Libro:

La clase Libro representa un libro en un sistema de biblioteca. Tiene los siguientes atributos:

- isbn (String): el número de identificación único del libro.
- titulo (String): el título del libro.
- autor (String): el autor del libro.
- categoria (String): la categoría a la que pertenece el libro.
- cantidadPaginas (int): la cantidad de páginas del libro.
- stock (int): la cantidad de copias disponibles en el inventario.
- La clase tiene un constructor que recibe los valores iniciales para todos los atributos y los asigna a las variables de instancia correspondientes. Si alguno de los parámetros del constructor no es válido (por ejemplo, es null, está vacío o es un número negativo), se lanza una excepción `IllegalArgumentException` con un mensaje de error específico.
- La clase Libro también proporciona métodos getter y setter para acceder y modificar los atributos del libro. Los métodos getter devuelven los valores de los atributos, y los métodos setter permiten establecer nuevos valores para los atributos.
- Además, la clase Libro tiene dos métodos adicionales:
- `stockBajo()`: disminuye el valor del atributo stock en 1. Se puede utilizar para indicar que se ha devuelto una copia del libro.
- `incrementarStock()`: incrementa el valor del atributo stock en 1. Se puede utilizar para indicar que se ha agregado una nueva copia del libro.

Estos métodos son útiles para mantener actualizado el stock del libro en el sistema de biblioteca.



Clase Registro:

La clase Registro en el paquete Model representa un registro de transacción en el sistema de biblioteca. Contiene los siguientes atributos:

- rut (String): el RUT del usuario involucrado en la transacción.
- nombre (String): el nombre del usuario.
- apellido (String): el apellido del usuario.
- isbn (String): el ISBN del libro involucrado en la transacción.
- nombreLibro (String): el nombre del libro.
- tipoTransaccion (String): el tipo de transacción realizada (por ejemplo, "préstamo" o "devolución").

La clase proporciona un constructor que recibe los valores iniciales para todos los atributos y los asigna a las variables de instancia correspondientes.



Clase Usuario:

La clase Usuario en el paquete Model representa un usuario del sistema de biblioteca. Contiene los siguientes atributos:

- rut (String): el RUT del usuario.
- nombre (String): el nombre del usuario.
- apellido (String): el apellido del usuario.
- contraseña (String): la contraseña del usuario.

La clase proporciona un constructor que recibe los valores iniciales para todos los atributos y los asigna a las variables de instancia correspondientes.



Clase Utils:

La clase Utils en el paquete Model contiene métodos de utilidad para leer y escribir archivos de libros, usuarios y registros. Estos métodos son utilizados para cargar datos desde archivos y almacenar registros en un archivo.

La clase proporciona los siguientes métodos:

- leerArchivoLibros(): Lee el archivo "libros.txt" y devuelve una lista de objetos Libro. Cada línea del archivo representa un libro y contiene información como el ISBN, título, autor, categoría, cantidad de copias y precio.
- leerArchivoUsuarios(): Lee el archivo "usuarios.txt" y devuelve una lista de objetos Usuario. Cada línea del archivo representa un usuario y contiene información como el RUT, nombre, apellido y contraseña.
- leerArchivosRegistro(): Lee el archivo "registro.txt" y devuelve una lista de objetos Registro. Cada línea del archivo representa un registro de transacción y contiene información como el RUT del vendedor, nombre, apellido, ISBN del libro, nombre del libro y tipo de transacción.
- registrarRegistros(List<Registro> registros): Registra una lista de registros en el archivo "registro.txt". Cada registro se agrega como una nueva línea en el archivo.

Estos métodos utilizan las clases Libro, Usuario y Registro del paquete Model para crear instancias de los objetos correspondientes con los datos leídos desde los archivos.



Clase Main:

En la clase "Main", se crea una instancia de la clase "SistemImpl", que implementa la interfaz "Sistema". La instancia de "SistemImpl" se utiliza para iniciar el programa llamando al método "startProgram()".

La clase "Main" puede realizar otras tareas adicionales de configuración inicial si es necesario, como la configuración de parámetros o la inicialización de componentes. Es común que la clase "Main" sea responsable de crear la instancia inicial de la aplicación y establecer cualquier estado inicial necesario antes de que la aplicación comience a funcionar.

Interfaz Sistema:

La interfaz "Sistema" define un contrato para las clases que implementan dicha interfaz. Su función principal es proporcionar un conjunto de métodos que deben ser implementados por cualquier clase que desee actuar como un sistema dentro del programa.

Clase SistemImpl:

La clase "SistemImpl" es una implementación de la interfaz "Sistema". Su función principal es proporcionar la lógica y la funcionalidad específica para el sistema del programa.

En la clase "SistemImpl", se implementa el método "starProgram()" definido en la interfaz "Sistema". Este método inicia el programa y realiza las configuraciones iniciales necesarias. En este caso, se crea una instancia de la interfaz gráfica "BiblioTech" y se muestra al usuario.

Además, la clase "SistemImpl" tiene la responsabilidad de administrar las listas de usuarios, libros y registros. Se crean listas vacías para almacenar los usuarios, libros y registros, y se puede realizar el manejo de estos datos según sea necesario.

