



Universidad Autónoma de Nuevo León
Facultad de Ciencias Físico-Matemáticas



Laboratorio de bases de datos

Producto Integrador de Aprendizaje: Manual técnico

Alumnos: Manuel Alejandro García Medina 2177573

Alexis Ibarra Rodriguez 2094647

Angel Emiliano Carmona Lozano 2069119

Yael Abiezer García Guerra 2050107

Paquetes.....	3
Clases.....	3
Model.....	3
Util.....	6
Controller.....	7
Main.....	11
View.....	12
Jerarquía de clases y Resources.....	14

Paquetes

El proyecto se divide en los paquetes data y PIA. Los paquetes dentro de PIA contienen las clases principales que hacen que funcione el programa

Clases

Model

El proyecto basa su funcionalidad en estas clases: **ClaseGrupal**, **Cliente**, **Inventario**, **Membresia** y **UsuarioEmpleado**. A continuación se enseñan sus métodos y miembros:

Cliente

Clase usada para el registro de un cliente, contiene los siguientes campos:

Nombre del miembro	Tipo
id	private String
serialVersionUID	private static final long
nombre	private String
apellido	private String
email	private String
fechaRegistro	private LocalDate
puntosFidelidad	Integer
membresiaActual	Membresia

Tipo	Nombre del método	Descripción
String	getNombreCompleto()	Retorna el nombre completo del usuario.
void	agregarPuntos(int puntos)	Agrega puntos de fidelidad al usuario.
String	toString()	Obtiene la información del usuario.

El resto de los métodos son getters y setters de los campos mencionados anteriormente.

Inventario

Clase encargada de registrar el inventario (equipamiento, pesas, etc.) del gimnasio.

Nombre del miembro	Tipo
id	private String
serialVersionUID	private static final long
nombre	private String
cantidad	private long

Los métodos son getters y setters de los miembros mencionados anteriormente.

Membresia

Nombre del miembro	Tipo
serialVersionUID	private static final long
TipoMembresia { BASICA, PREMIUM, ESTUDIANTE, ANUAL }	public enum
tipo	private TipoMembresia
fechaInicio	private LocalDate
fechaFin	private LocalDate
costo	private double
activa	private boolean
actualizable	private boolean

Tipo	Nombre del método	Descripción
public boolean	esValida()	Retorna un valor booleano si la membresía todavía es válida.
public void	cancelar()	Cancela la membresía

		cambiando el miembro activa a false.
public long	diasRestantes()	Obtiene los días restantes de la membresía.
public int	getMeses()	Obtiene los meses restantes de la membresía.

Los otros métodos son getters y setters de los campos mostrados anteriormente.

UsuarioEmpleado

Pertenece al registro de los empleados del gimnasio.

Nombre del miembro	Tipo
serialVersionUID	private static final long
username	private String
nombreCompleto	private String
rol	private String

Tipo	Nombre del método	Descripción
public boolean	autenticar(String user, String pass)	Retorna un valor booleano asegurando que el nombre y pass del miembro son correctas.
public String	getNombreCompleto()	Regresa el nombre completo del usuario
public String	getRol()	Regresa el rol del usuario
public String	toString()	Regresa la información del usuario.

ClaseGrupal

Nombre del miembro	Tipo
serialVersionUID	private static final long
id	private String

descripcion	private String
date	private LocalDate
hour	private LocalDate

Los métodos son getters y setters de los miembros mencionados anteriormente.

Util

Contiene la interfaz Gestionador, la excepción GymException y la clase Serializador:

Gestionador<T>

La interfaz se hizo para que las clases gestionadoras dentro del paquete de controladores siguieran los métodos que se muestran a continuación

Tipo	Nombre del método	Descripción
void	registrar(T obj)	Registra el objeto hacia una base de datos.
void	eliminar(String id)	Elimina el objeto de la base de datos
void	actualizar(T obj)	Actualiza el objeto deseado.
Optional<T>	buscar(String id)	Busca el objeto dentro de la base de datos.
List<T>	getLista()	Regresa una lista de los objetos registrados dentro de la base de datos.

GymException

Es el manejador de excepciones por defecto de la aplicación.

Serializador<T>

Es la clase encargada de guardar los objetos dentro de los archivos .ser respectivos (clientes, empleados, inventarios y fechas de calendario).

Tipo	Nombre del método	Descripción
------	-------------------	-------------

public void	guardar(String archivo, List<T> lista)	Dada la dirección del archivo .ser y una lista de objetos, la guarda en el archivo.
public List<T>	cargar(String archivo)	Carga un arreglo que contiene la lista de los empleados registrados en la aplicación.

Controller

Se encarga de realizar todas las operaciones de la aplicación para que pueda funcionar, se compone de varias clases que tienen la función de enseñar la vista de cierta ventana, guardar objetos en la base de datos, etc.

CalendarioDeClase implements Gestionador<ClaseGrupal>

Se encarga de gestionar las actividades realizadas en el calendario de actividades.

Tipo	Nombre del método	Descripción
void	registrar(ClaseGrupal clase)	Registra la clase hacia el archivo de clases.
void	eliminar(String id)	Elimina la clase de la base de datos
void	actualizar(ClaseGrupal claseModificada)	Actualiza la clase deseada.
Optional<ClaseGrupal>	buscar(String id)	Busca la clase dentro de la base de datos.
List<ClaseGrupal>	getLista()	Regresa una lista de las clases registradas dentro de la base de datos.
void	guardarClases()	Guarda las clases dentro del archivo de la base de datos.

ControlAccesol Barra

Se maneja de registrar las entradas y salidas de los clientes en el gimnasio, y las despliega en la terminal y aplicación.

Contiene una variable llamada registroAccesos, que registra cada entrada y salida.

Tipo	Nombre del método	Descripción
public boolean	validarEntrada(Cliente cliente)	Registra la entrada del cliente, y añade 10 puntos de fidelidad.
public void	registrarSalida(Cliente cliente)	Registra la salida del cliente.
private void	registrarEvento(Cliente c, String tipo)	Registra la entrada, salida o cualquier evento que se quiera añadir.
public List<String>	gethistorialAccesos()	Regresa los accesos que se hicieron en la sesión.

GeneradorReportesA extends Task<Boolean>

Contiene un solo método public boolean call(), que se encarga de generar los reportes de inventarios y de usuarios registrados, a través de multithreading. La generación de reportes adicionalmente conecta los hilos a una barra de progreso en la pestaña de reportes, para demostrar el progreso de la generación de los reportes.

GestionInventario implements Gestionador<Inventario>

Tipo	Nombre del método	Descripción
void	registrar(ClaseGrupal clase)	Registra el objeto al inventario.
void	eliminar(String id)	Elimina el objeto de la base de datos.
void	actualizar(ClaseGrupal claseModificada)	Actualiza el objeto.

Optional<ClaseGrupal>	buscar(String id)	Busca el objeto en el inventario.
List<ClaseGrupal>	getLista()	Regresa una lista de los objetos registrados.
void	guardarInventarios()	Guarda los objetos dentro del archivo del inventario.

LoginController

Usado para inicializar el diseño FXML de la vista de login.

Tipo	Nombre del método	Descripción
void	onLoginClick()	Usado para cambiar de escena del login a la escena principal
void	initialize()	Inicializa el diseño FXML

ProcesadorPagos4647

Clase que procesa los pagos de la membresía, simulando un sistema de pagos por medio de una función random para desplegar si el pago ha sido aceptado o rechazado.

Tipo	Nombre del método	Descripción
public double	calcularTotalConCentavos(double precioBase, double descuento)	Calcula el descuento del pago, si es que es aplicable.
public boolean	procesarPagoConTarjeta(String numeroTarjeta, double monto)	Simulación del pago con tarjeta, usando un delay para emular la aceptación del pago.
public void	generarRecibo(String nombreCliente, Membresia mem, double total)	Imprime el recibo usando datos del cliente y la membresía.

SistemaMembresias1412

Controlador que maneja las membresías y el sistema de pagos de éstas.

Tipo	Nombre del método	Descripción
private Membresia	calcularCostoMembresia(TipoMembresia tipo, int meses)	Genera el costo de la membresía dependiendo del tipo de membresía escogido.
public void	inscribirCliente(Cliente cliente, TipoMembresia tipo, int meses, String tarjetaSimulada)	Inscribe a un cliente con una membresía del tipo deseado, emulando el proceso del pago.
public void	renovarMembresia(Cliente cliente, int mesesExtras, String tarjetaSimulada, double descuento)	Renueva la membresía, aplicando descuento si los puntos de fidelidad el usuario cumplen con la condición.

GestionClientesIbarra

Contiene los autenticadores para asegurar que el usuario registrado pueda hacer un login correcto en la aplicación.

Tipo	Nombre del método	Descripción
public static boolean	validarNombre(String nombre)	Valida que el formato del nombre de usuario sea correcto.
private boolean	validarCorreo(String correo)	Valida que el formato del correo sea correcto.
void	registrar(Cliente c)	Registra el cliente a la base de datos
void	eliminar(String id)	Elimina el cliente de la base de datos.
void	actualizar(Cliente clienteModificado)	Actualiza el cliente.
Optional<Cliente>	buscar(String id)	Busca el cliente en la base de datos.
List<Cliente>	getLista()	Regresa una lista de los clientes registrados.
private void	guardarClientes()	Guarda los clientes dentro de la base de datos.

private void	guardarEmpleados()	Guarda los empleados dentro de la base de datos.
public UsuarioEmpleado	autenticar()	Autentica al empleado registrado en la base de datos en el login.

Main

Launcher

Contiene un wrapper hacia la clase MainApp usado para iniciar el proyecto.

MainApp

Contiene los métodos para ejecutar la aplicación y las vistas principales (clientes, membresías, inventario, accesos, calendario y reportes).

Tipo	Nombre del método	Descripción
public void	start(PrimaryStage stage)	Inicia los gestores y la vista del login.
private void	cargarVistaLogin()	Carga la vista del login
public void	setUsuarioLogeado(UsuarioEmpleado usuario)	Carga el usuario que se registró en la sesión.
public void	mostrarVistaPrincipal()	Carga la vista principal.
private Scene	crearVistaPrincipal(Stage primaryStage)	Crea la vista principal con todas las pestañas cargadas
public static void	mostrarAlerta(Alert.AlertType type, String title, String content)	Muestra un popup de alerta customizado.
public static void	showDiscountPopup(Cliente cliente, TextArea logArea)	Muestra un popup en caso de que la membresía tenga un descuento aplicable.
public static void	actualizarVistaClientes(TableViewController<Cliente> tableView)	Actualiza la tabla de clientes al momento de realizar un cambio.

View

Las siguientes clases se enfocan solamente en crear la vista de las pestañas.

ClienteView

Tipo	Nombre del método	Descripción
public static BorderPane	CRUDVistaClientes()	Crea la vista de la base de datos de los clientes.
private static void	CreateClient(Table<Cliente> tableView)	Añade funcionalidad para el botón de Añadir Cliente.
private static void	UpdateClient(Table<Cliente> tableView)	Añade funcionalidad para el botón de Actualizar Cliente.
private static void	deleteClient(Table<Cliente> tableView)	Añade funcionalidad para el botón de Eliminar Cliente.

MembresiasView

Tiene el método crearVistaMembresias() que carga la pestaña para la asignación de membresías.

VistaCalendario

Tipo	Nombre del método	Descripción
public static BorderPane	crearVistaCalendarioDeClases()	Crea la vista del calendario de clases..
private static void	showCalendario()	Obtiene el calendario de cada mes.
private static void	createClaseGrupal(Table<ClaseGrupal> claseGrupalTableView)	Crea una clase grupal en una fecha dada.
private static void	updateClase(Table<ClaseGrupal> claseGrupalTableView)	Actualiza una clase grupal seleccionada en el calendario

private static void	deleteClase(tableView<Client e> tableView)	Borra una clase grupal seleccionada en el calendario
private static void	mostrarClasesEnTabla(Local Date fecha)	Muestra las clases grupales registradas en el calendario.

VistaControlAcceso

Contiene el método crearVistaControlAcceso(), que crea la vista de esa pestaña.

VistaInventario

Genera la vista de la pestaña de inventario.

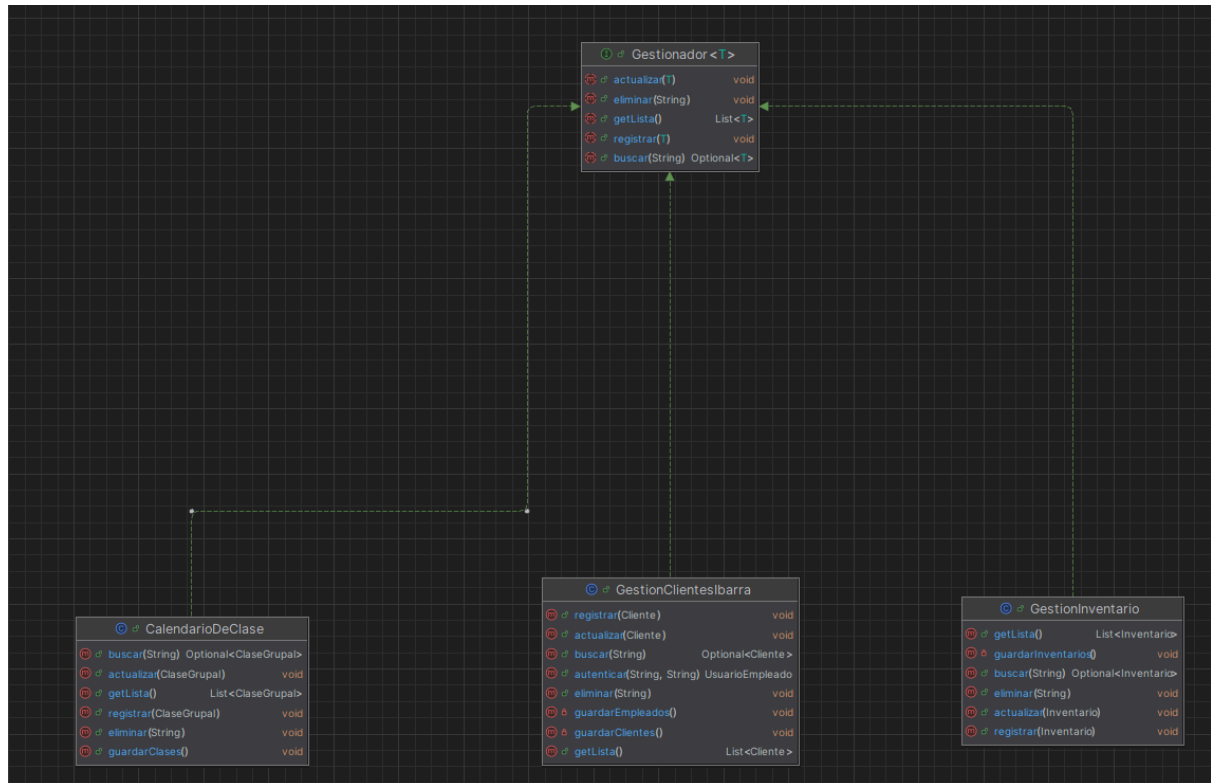
Tipo	Nombre del método	Descripción
public static BorderPane	crearVistaInventario()	Crea la pestaña del inventario en la vista principal.
public static void	crearInventario(tableView <Inventario> tablaInventario)	Funcionalidad del botón para agregar inventarios.
public static void	actualizarInventario(tableView<Inventario> tablaInventario)	Funcionalidad del botón para actualizar el inventario.
private static void	eliminarInventario(tableView<Inventario> tablaInventario)	Funcionalidad del botón para eliminar el inventario.

VistaReportes

La clase contiene un solo método, que sirve para crear la vista de la pesatña de reportes y su funcionalidad.

Jerarquía de clases

El proyecto no contiene una jerarquía, ya que todas las clases no extienden sus métodos a otras, pero existe la interfaz Gestionador:



Que asegura que las clases gestionadoras puedan realizar los métodos y lineamientos impuestos por la interfaz.

Resources

Los recursos de la aplicación son usados para guardar imágenes, diseños FXML y los estilos CSS. Tiene diferentes imágenes para errores, validaciones, para los íconos de la aplicación, etc.