



SISTEMA ADMINISTRATIVO Y DE CONTROL DE ACCESO

MANUAL DE USUARIO

Av.Colon 1665
Oficina N°5
2235054684
Mar del plata,Argentina
gymcontrol@soporte.com
www.gymcontrol.com

Contenido

I.	Introducción.....	1
1.	Objetivos.....	1
2.	Requerimientos.....	1
II.	Opciones del sistema.....	2
1.	Ingreso al sistema.....	2
2.	Funciones de Turnos.....	3
3.	Funciones de Tienda.....	3
4.	Funciones de Usuarios.....	4
5.	Funciones de Contabilidad	4

I. Introducción

1. Objetivo

Brindar a dueños y empleados de gimnasios herramientas de contabilidad, administración y control de acceso de clientes basado en las restricciones por el COVID-19.

2. Requerimientos

- Versiones de 64 bits de Microsoft Windows 10, 8
- 2 GB de RAM como mínimo, 8 GB recomendados
- 2,5 GB de espacio en disco duro, SSD recomendado
- JDK 1.8 desde 2016.1
- 1024x768 mínimo de resolución de pantalla

II.Opciones del sistema

El presente manual está organizado de acuerdo a la secuencia del código en la clase Main.

1. Ingreso al sistema

En este bloque se registran clientes, profesores y turnos con datos generados por los desarrolladores a modo de testeo.

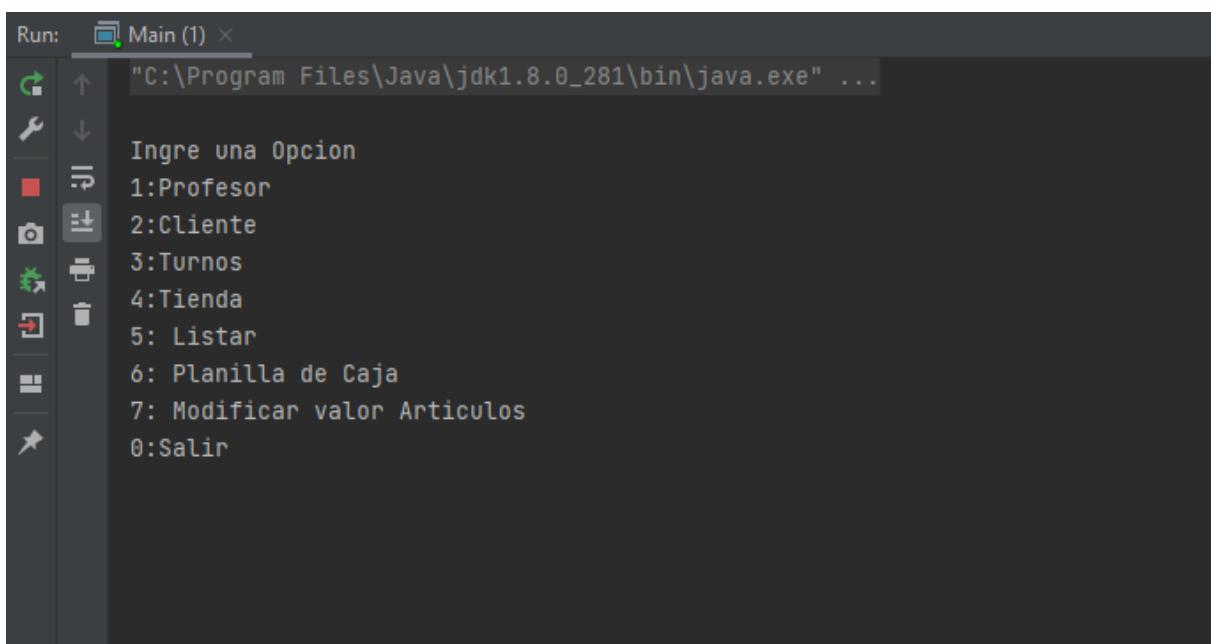
Por otro lado se inicia un menú de opciones que mediante las teclas 1/2/3/4/5/6/7 llevará a las distintas funcionalidades del sistema.

```
gym.AgregarPersona(new Profesor( nombre: "Hector", apellido: "Sosa", dni: 1, genero: 'H', edad: 35, celular: 223421122, Disciplin
gym.AgregarPersona(new Profesor( nombre: "Jimena", apellido: "Fierro", dni: 2, genero: 'M', edad: 35, celular: 223421122, Disciplin
gym.AgregarPersona(new Profesor( nombre: "Lucas", apellido: "Lopez", dni: 3, genero: 'H', edad: 35, celular: 223421122, Disciplin
gym.AgregarPersona(new Profesor( nombre: "Pedro", apellido: "Sgalla", dni: 4, genero: 'H', edad: 35, celular: 223421122, Disciplin

gym.AgregarPersona(new Cliente( nombre: "Juan", apellido: "Perez", dni: 5, genero: 'H', edad: 18, celular: 223421122, LocalDate.r
gym.AgregarPersona(new Cliente( nombre: "Juana", apellido: "Matinez", dni: 6, genero: 'M', edad: 18, celular: 223421122, LocalDat
gym.AgregarPersona(new Cliente( nombre: "Pablo", apellido: "Jose", dni: 7, genero: 'H', edad: 18, celular: 223421122, LocalDate.r
gym.AgregarPersona(new Cliente( nombre: "Gonzalo", apellido: "Perez", dni: 8, genero: 'H', edad: 18, celular: 223421122, LocalDat
gym.AgregarPersona(new Cliente( nombre: "Gaston", apellido: "Loel", dni: 9, genero: 'H', edad: 18, celular: 223421122, LocalDate.
gym.AgregarPersona(new Cliente( nombre: "Franco", apellido: "Valiente", dni: 10, genero: 'H', edad: 18, celular: 223421122, Local
gym.AgregarPersona(new Cliente( nombre: "Lucia", apellido: "Benoffi", dni: 11, genero: 'M', edad: 18, celular: 223421122, LocalD
gym.AgregarPersona(new Cliente( nombre: "Pedro", apellido: "Lopez", dni: 12, genero: 'H', edad: 18, celular: 223421122, LocalD

gym.AgregarTurno(new Turno( horario: 9, new Profesor( nombre: "Hector", apellido: "Sosa", dni: 1, genero: 'H', edad: 35, celular: 22
gym.AgregarTurno(new Turno( horario: 10, new Profesor( nombre: "Jimena", apellido: "Fierro", dni: 2, genero: 'M', edad: 35, celular: 22
gym.AgregarTurno(new Turno( horario: 11, new Profesor( nombre: "Lucas", apellido: "Lopez", dni: 3, genero: 'H', edad: 35, celular: 22
gym.AgregarTurno(new Turno( horario: 12, new Profesor( nombre: "Pedro", apellido: "Sgalla", dni: 4, genero: 'H', edad: 35, celular: 22
```

Clientes,profesores y turnos.



Menú de opciones.

2. Funciones de Turnos

```
public boolean Inscribirse_A_Turnos(Integer horario, Persona persona) {}
```

Inscribe a un cliente en un turno. Comprueba el pago de la cuota y que el turno no esté lleno.

```
public void Cambiar_De_Turnos(Persona persona, Integer horarioActual, Integer horarioNuevo) {}
```

Cambia a un cliente de turno. Cancela su turno anterior y lo agrega al nuevo.

```
public void Cancelar_Turno(Integer horarioActual, Persona persona)
```

Cancela el turno.

3. Funciones de Tienda

```
public void Ver_Tienda() {}
```

Muestra el estado actual de la tienda: sus productos, con stock y precio, y la caja.

```
public void Vender_Producto() {}
```

Da a seleccionar el producto y la cantidad a vender, comprueba que el producto exista y que el stock sea suficiente.

```
public void Agregar_Producto() {}
```

Agrega un producto a la tienda si no existe ya un producto con ese nombre.

```
public void Quitar_Producto() {}
```

Remueve un producto de la tienda.

```
public void Retirar_De_Caja() {}
```

Permite actualizar el saldo de la caja tras un retiro de dinero.

```
public void Reponer_Caja() {}
```

Permite actualizar el saldo de la caja tras reponer dinero.

4. Funciones de Manejo de usuarios

```
public void Dar_De_baja() {}
```

Introduce a los clientes con deuda en una lista de deudores.

```
public void Dar_De_alta() {}
```

Pasa a los usuarios que pagaron su deuda a la lista de clientes por defecto.

```
public Persona BuscarClientePorDNI() {}
```

Busca un cliente mediante un dni introducido por el admin.

```
public void Borrar_Cliente() {}
```

Borra un cliente mediante un dni introducido por el admin.

5. Funciones de Contabilidad

```
public void listar() {}
```

Muestra los movimientos(ingresos, egresos)que se produjeron el día de la fecha.

```
public void cierreCaja() {}
```

Permite al operario retirar el dinero producido en el día y aumenta la fecha en un día para al registrar nuevos movimientos hacerlo en ese dia.

```
public void initBilletera() {}
```

Reinicia la caja.

```
public boolean pagar(int debito) {}
```

Resta de la caja el valor pasado por parámetro.

```
public void agregar(int credito) {}
```

Agrega a la caja el valor pasado por parámetro.

```
public void verSaldo() {}
```

Muestra el saldo.