



DAMA Sports



Proyecto

Hugo Garrido Rojo, Rebeca Cerón Camps, José Andrés Cava Torres

Índice

1.Modelo de clases	Diagrama simple de Socio
2.Servicio ClubDeportivo	Métodos principales y reglas
3.Reglas Negocio	Disponibilidad, bajas, tiempos válidos
4.Persistencia	Formato ficheros, lectura/escritura, lecciones aprendidas
5.Flujo de Git	Ramas, revisiones
6.Caso práctico	1 caso exitoso + 1 caso con validación.

Modelo de clases (UML)

Socio
- ID_SOCIO : string - dni : string - nombre : string - apellido : string - telefono : string - email : string
+ getIdSocio(): string + getDni(): string + getNombre(): string + getApellidos(): string + getTelefono(): string + getEmail(): string + setDni(in dni: string) + setNombre(in nombre: string) + setApellidos(in apellidos: string) + setTelefono(in telefono: string) + setEmail(in email: string)

Pista
- ID_PISTA : string - deporte : string - descripcion : string - disponible : string
+ getIdPista(): string + isDisponible(): boolean + setDisponible(in disponible: string) + getDescripcion(): string + setDescripcion(in descripcion: string) + getDeporte(): string + setDeporte(in deporte: string)

ClubDeportivo
+ pistas : string [*] + socios : string [*] + reservas : string [*]
+ agregarSocio(in socio: Socio) + agregarPista(in pista: Pista) + agregarReserva(in reserva: Reserva) + getSocios(): Socio [*] + getPistas(): Pista [*] + getReservas(): Reserva [*] + cargarDatosSocio() + guardarDatosSocio(in socio: Socio) + cargarDatosPista() + guardarDatosPista(in pista: Pista) + cargarDatosReserva() + guardarDatosReserva(in reserva: Reserva) + guardarDatosGlobales() + altaSocio(in socio: Socio) + bajaSocio(in idSocio: string) + altaPista(in pista: Pista) + cambiarDisponibilidadPista(in idPista: string, in disponible: boolean) + crearReserva(in reserva: Reserva): boolean + cancelarReserva(in idReserva: string)

Reserva
- ID_RESERVA : string - idSocio : string - idPista : string - fecha : date - horainicio : date - duracionMin : integer - precio : double
+ getIdReserva(): string + getIdSocio(): string + setIdSocio(in idSocio: string) + getIdPista(): string + setIdPista(in idPista: string) + getFecha(): date + setFecha(in fecha: date) + getHorainicio(): date + setHorainicio(in horainicio: string) + getDuracionMin(): integer + setDuracionMin(in duracionMin: integer) + getPrecio(): double + setPrecio(in precio: double)



Clases: Socio, Pista y Reserva

```
public class Socio {  👤rebecaccamps +1
    private final String idSocio;
    private String dni;
    private String nombre;
    private String apellidos;
    private String telefono;
    private String email;
```

- Identificador único
- Cada cliente del club
- No mismo socio con ID

```
public class Reserva {  👤rebecaccamps +1
    private final String idReserva;
    private String idSocio;
    private String idPista;
    private LocalDate fecha;
    private LocalTime horaInicio;
    private int duracionMin;
    private double precio;
```

- Cada pista tiene un estado de disponibilidad
- Se puede marcar como no disponible

```
public class Pista {  👤rebecaccamps
    private final String idPista;
    private String deporte;
    private String descripcion;
    private boolean disponible;
```

- Conecta Socios y Pistas
- Valida que estén disponibles y no se solapen horas

Clase ClubDeportivo

- Núcleo del sistema
- Contiene las colecciones de Socios, Pistas y Reservas
- Contiene toda la lógica

Clase: ClubDeportivo

`ArrayList<Socio> socios`

`ArrayList<Pista> pistas`

`ArrayList<Reserva> reservas`

Métodos principales

`altaSocio()`

`bajaSocio()`

`cambiarDisponibilidadPista()`

`crearReserva()`

`cancelarReserva()`

`guardarDatos()` (De todos los objetos)

```
//Dar de alta a un Socio
public boolean altaSocio(Socio socio) { @rebecacamps
    for (Socio s : socios) {
        if (s.getIdSocio().equals(socio.getIdSocio())) {
            return false;
        }
    }
    socios.add(socio);
    return true;
}
```

Reglas de Negocio

- No se puede crear un Socio con un id ya existente
- Se deben rellenar los campos obligatorios *

- La fecha de reserva no puede ser anterior a la actual
- La pista tiene que estar disponible para poder reservar

Persistencia de datos

Contenido:

- Lectura y escritura al presionar el botón de guardado.



- Formato de ficheros: .txt

Archivos:

- socios.txt, pistas.txt, reservas.txt

Formato:

- SOCIOS:
id,dni,nombre,apellidos,telefono,email
- RESERVA:
idReserva,idSocio,idPista,fecha,horaInicio,duracionMin,precio
- PISTA:
idPista,deporte,descripcion,disponible



Método Cargar

```
//Cargar los datos a un fichero reservas
public void cargarDatosReserva() throws IOException {
    King-Yike567

    FileReader fr = new FileReader(fileName: "reservas.txt");
    BufferedReader br = new BufferedReader(fr);
    String linea = br.readLine();

    while (linea != null) {

        String[] datos = linea.split(regex: ",");

        if (datos.length == 7) {
            try {
                Reserva r = new Reserva(datos[0], datos[1], datos[2], LocalDate.parse(datos[3]),
                    LocalDateTime.parse(datos[4]), Integer.parseInt(datos[5]), Double.parseDouble(datos[6]));
                reservas.add(r);
            } catch (Exception e) {
                System.out.println("Error al procesar la reserva: " + linea + " - " + e.getMessage());
            }
        } else {
            System.out.println("Error: Formato incorrecto en linea de reserva: " + linea);
        }

        linea = br.readLine();
    }

    br.close();
    fr.close();
}
```

Método Guardar

```
public void guardarDatosReserva(Reserva r) throws IOException {
    King-Yike567

    FileWriter fw = new FileWriter(fileName: "reservas.txt", append: true);
    BufferedWriter bw = new BufferedWriter(fw);

    bw.write(stri: r.getIdReserva() + "," + r.getIdSocio() + "," + r.getIdPista() + "," + r.getFecha()
        + "," + r.getHoraInicio() + "," + r.getDuracionMin() + "," + r.getPrecio() + "\n");

    bw.flush();
    bw.close();
    fw.close();
}
```

Flujo de Git

Título: Flujo de Trabajo con Git

Contenido:

- Repositorio: **dam-green-u1-equipoN**
- Ramas: Master, Rebeca, Hugo, José.
- Pull Requests y revisiones de código
- Commits con mensajes claros

ImplementacionVista
K1ng-Yike567 • 7 days ago
CambioPrueba2
K1ng-Yike567 • 7 days ago
CambioPrueba
K1ng-Yike567 • 7 days ago
modificado condicion precio de pista
rebecaccamps • 10 days ago
Documentos basicos csv integrados
rebecaccamps • 10 days ago
Actualizado ClubDeportivo
rebecaccamps • 10 days ago
Dar de baja un Socio
rebecaccamps • 10 days ago
Estructura modificada y alta de socio
rebecaccamps • 10 days ago
Estructura básica realizada
rebecaccamps • 10 days ago

CambioPequeño

14 changed files

idea/misc.xml

```
idea/misc.xml @@ -1,6 +1,6 @@
DAMA Sports\src\...\SocioFormView.java 1 1 <?xml version="1.0" encoding="UTF-8"?>
2 2 <project version="4">
out\production\...\ClubDeportivo.class 3 - <component name="ProjectRootManager" version="2" languageLevel
out\production\dam-ads-\Pista.class 3 + <component name="ProjectRootManager" version="2" languageLevel
out\production\dam-a-\Reserva.class 3 + <component name="ProjectRootManager" version="2" languageLevel
out\production\dam-ads-\Socio.class 3 + <component name="ProjectRootManager" version="2" languageLevel
out\production\dam-\MainApp.class 4 4 <output url="file://$PROJECT_DIR$/out" />
out\production\...\BajaSocioView.class 5 5 </component>
out\...\CambiarDisponibilidadView.class 6 6 </project>
out\produ...\CancelarReservaView.class
out\production...\DashboardView.class
out\production\...\SocioFormView.class
out\production\...\ReservaFormView.class
out\production\...\SocioFormView.class
```


Caso práctico

- Reservar con una fecha anterior a la de hoy (da error)
- Intentar Reservar una pista no disponible
- Intentar dar de alta Socio con mismo id
- Guardado de datos de un Socio con solo campo ID.



★ ¡Muchas gracias! ★

¿ALGUNA PREGUNTA?