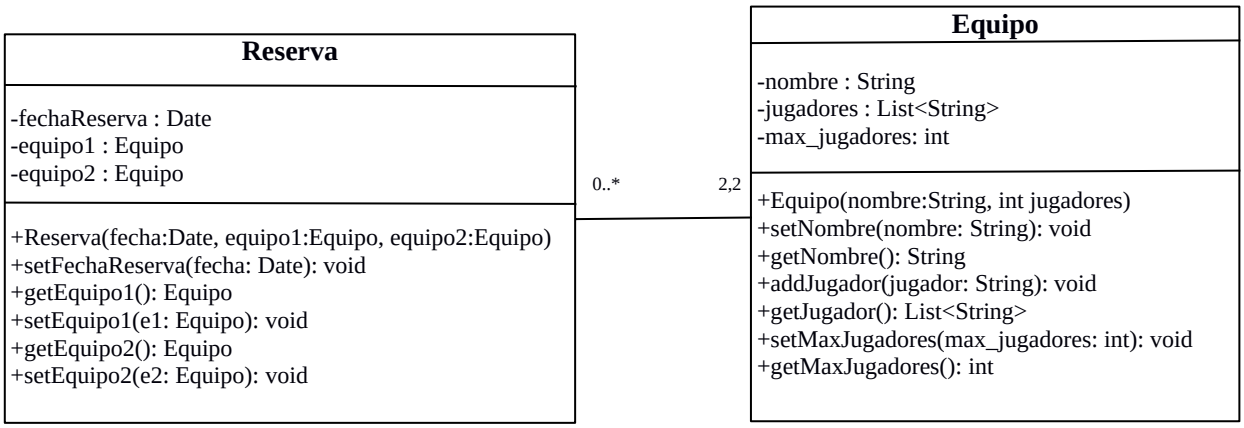
Examen 111Mil – Asignación de Canchas de Futbol 5

El programa 111Mil ha observado que tanto los instructores como alumnos del programa han trabajado y estudiado mucho pero tanto estudio y aprendizaje ha provocado un poco de cansancio. Por esta razón, desea ofrecerles un espacio de esparcimiento y recreación, no sólo para que se relajen sino también para que encuentren otro espacio de charla y asi fomentar las relaciones humanas tan importantes para los futuros programadores. Para ello tiene un club, Club 111Mil, el cual posee varias canchas de fútbol 5 y donde los integrantes del curso de 111Mil pueden asistir para jugar un partido.

Obviamente, es necesario programar un sistema que ayude con la reservas de las canchas ya que todos están muy interesados en relajarse y divertirse. Quiénes mejores que los mismos alumnos de 111Mil para implementar el sistema de asginación de las canchas de fútbol 5 correspondiente.

Ejercicio 1. Implementar desde el diagrama de clases

Se le ha dado al equipo de programadores un conjunto de clases, otros programadores comenzaron con la codificación pero nos piden que la completemos. Por lo que el equipo 111Mil debe completar la implementación de los métodos y la definición de las variables que faltan. Por favor, ayuda a completar la implementación de los métodos que faltan y en la definición de las variables.




```
public class Reserva {
   private Date fechaReserva;
   private Equipo equipo1,equipo2;

   public Reserva(Date dReserva, Equipo dEquipo1, Equipo dEquipo2) {

      this.fechaReserva = dReserva;
      this.equipo1 = dEquipo1;
      this.equipo2 = dEquipo2;
   }

      // AGREGAR LOS MÉTODOS QUE FALTAN
```

SOLUCION POSIBLE

```
public void setFechaReserva(Date fechaReserva) {
      this.fechaReserva = dateReserva;
   }

   public Equipo getEquipo1() {
      return this.equipo1;
   }

   public void setEquipo1(Equipo dEquipo) {
      this.equipo1 = dEquipo;
   }

   public Equipo getEquipo2() {
      return this.equipo2;
   }

   public void setEquipo2(Equipo dEquipo) {
      this.equipo2 = dEquipo;
   }
```

```
}

public class Equipo {
   private String name;
   private List<String> jugadores;
```

// AGREGAR LA DEFINICIÓN DE VARIABLES QUE FALTAN

SOLUCION POSIBLE

```
 private int max_jugadores = 0;
```

```
public Equipo(String name, int jugadores) {
   this.name = name;
   this.max_jugadores = jugadores
   }

   public void addJugador(String jugador) {
      this.jugadores.add(jugador);
   }

   public List<String> getJugador() {
      return this.jugadores;
   }
```

// AGREGAR LOS MÉTODOS QUE FALTAN

SOLUCION POSIBLE

```
    public void setNombre(String nombre) {
      this.nombre = nombre;
   }

   public String getNombre() {
      return this.nombre;
```

```
  }

  public void setMaxJugadores(int max_jugadores) {
     this.max_jugadores  = max_jugadores;
  }

  public int getMaxJugadores() {
     return this.max_jugadores ;
  }
```

}

Ejercicio 2. Implementar un método a partir de un enunciado

A Juan, el presidente del Club 111Mil, le gustó la primera implementación del programa que leresuelve sus problemas y se le ocurrió agregar otra funcionalidad al mismo:

- En la clase Equipo, al momento de agregar un jugador a un equipo (método addJugador), verificar que el equipo no supere el máximo definido para el equipo en cuestión (representado por el atributo max_jugadores). Codificar en Java todos los métodos necesarios para satisfacer el nuevo requisito de Juan. Indicar a qué clase (Equipo o Reserva) corresponde cada método de los codificados.

SOLUCION POSIBLE

```
public void addJugador(String dni)
  {
        if (hayEquipo())
                this.jugadores.add(dni);
  }

private boolean hayEquipo() {
        return (this.jugadores.size()<=this.max_jugadores);
        }
```

Ejercicio 3. Extender la funcionalidad de un método a partir del enunciado.

Dado el siguiente método de la clase Club, extenderlo para que en el momento de realizar una reserva se verifique que los 2 equipos no contengan jugadores repetidos. Por ejemplo, los siguientes equipos son válidos:

Equipo A: Juan, Pedro, Carlos, Federico y Javier.

Equipo B: Martín, Pablo, Sebastián, Gonzalo y Lucas.

Sin embargo, estos equipos no son válidos para una reserva:

Equipo A: **Juan**, Pedro, **Carlos**, Federico y Javier.

Equipo B: Martín, **Juan**, Sebastián, **Carlos** y Lucas.

```
public boolean realizarReserva (Date fecha, Cancha cancha, Equipo e1, Equipo e2)
   {
        if (!(hayJugadoresRepetidos(e1, e2))){
            Reserva nueva = new Reserva ();
            nueva.setDateReserva(fecha);
             nueva.setEquipo1(e1);
             nueva.setEquipo2(e2);
             cancha.addReserva(nueva);
             this.addCancha(cancha);
             return true;
     } else
             return false;
 }


private boolean hayJugadoresRepetidos(Equipo e1, Equipo e2) {
       // Codificar este método
```

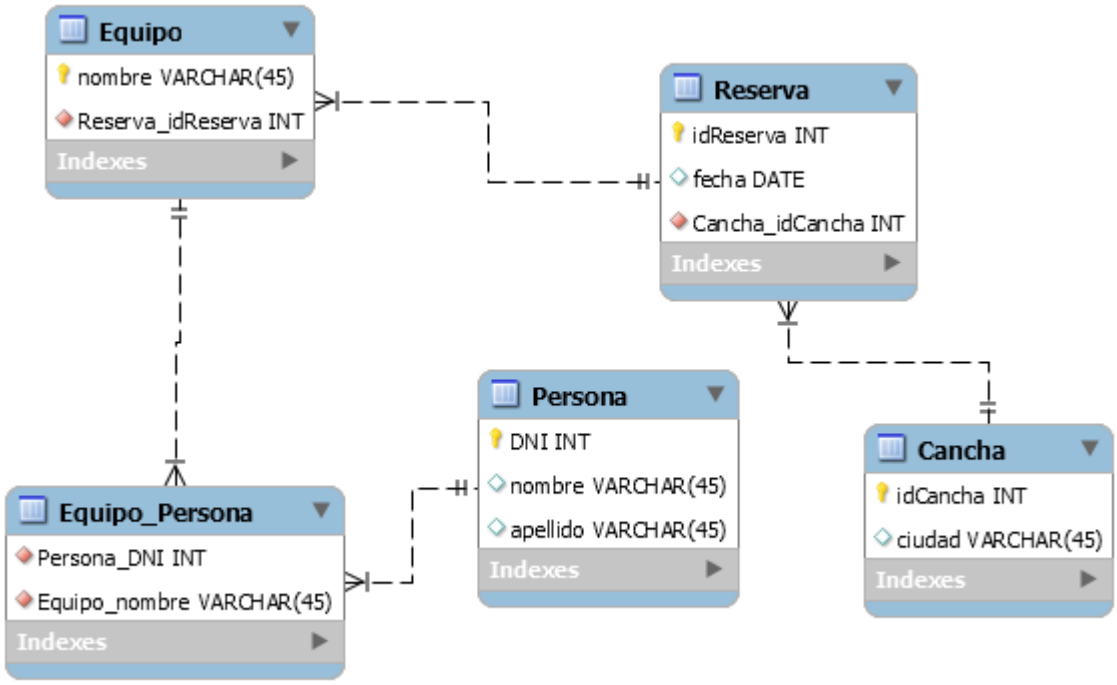
Codificá en Java el método hayJugadoresRepetidos, que recibe dos listas con los nombres de los jugadores, verifica que no haya jugadores repetidos en las listas y devuelve false si no hay jugadores repetidos y true si encuentra jugadores repetidos.

Posible Solución

```
private boolean hayJugadoresRepetidos(Equipo e1, Equipo e2) {

        //comparar ambas listas de jugadores
        List<String> jugadores1 = e1.getJugadores();
        List<String> jugadores2 = e2.getJugadores();
        boolean repetidos = false;

        Iterator<String> it_jugadores1 = jugadores1.iterator();
        Iterator<String> it_jugadores2 = jugadores2.iterator();

        while (it_jugadores1.hasNext() && !repetidos){
                while (it_jugadores2.hasNext()&& !repetidos)
                  if (it_jugadores1.next().equals(it_jugadores2.next()))
                        repetidos=true;
                 }
                        return repetidos;
        }
```

Ejercicio 4. Interpretación de DER.

Algunos alumnos del curso de 111Mil ha realizado un siguiente DER pero el profesor tiene algunas dudas sobre el mismo. A partir del DER, **responda Verdadero (V) o Falso (F)** a las siguientes afirmaciones (asignar a cada afirmación un valor V o F. Todas las afirmaciones tienen que tener un valor asignado. Puede haber varios V y varios F):



SOLUCION

Entre las tablas Reserva y Cancha existe una relación 1 a 1	F
Entre las tablas Equipo y Reserva existe una relación N a 1	V
Entre las tablas Persona y Equipo_Persona existe una relación N a N	F
La clave primaria de la tabla Reserva es Cancha_idCancha	F
La clave foránea de la tabla Equipo es Reserva_idReserva	V
La tabla Cancha no tiene clave foránea	V
La tabla Equipo no tiene clave primaria	F

Ejercicio 5. Escribir una consulta SQL.

Los responsables del programa 111MIL desean saber cuántas reservas se han hecho por ciudad ordenadas en orden decreciente. Esto les permitirá tener la información necesaria para saber si tiene que abrir o cerrar algunos clubes.

Por favor, ayudales a realizar la consulta SQL:

SOLUCION POSIBLE

```
SELECT Cancha.ciudad, COUNT(*) FROM Reserva, Cancha
    WHERE (Cancha.idCancha=Reserva.Cancha_idCancha)
    GROUP BY Cancha.ciudad
    ORDER BY Cancha.ciudad DESC;
```