

# Bienvenido a tu Certificación (18/12 - 19 hs)

¡Ya estás un paso más cerca de ser un Analista del Conocimiento - Dimensión Programador!

El examen consta de 5 ejercicios basados en los conocimientos exigidos por el 111 Mil y por la industria, para convertirte en programador junior. Tené en cuenta y leé con mucha atención las siguientes pautas para la correcta realización y aprobación del examen:

- 1- El examen tiene una duración máxima de 2 (dos) horas. Deberás enviar los resultados antes de cumplir ese tiempo.
- 2- Es necesario responder TODOS los ejercicios para poder aprobar el examen.
- 3- El resultado es APROBADO o DESAPROBADO, no tiene puntaje.
- 4- Los veedores estarán presentes para verificar que el examen se tome en las condiciones adecuadas.
- 5- Está prohibido utilizar el chat, el correo electrónico (fuera del uso normal para abrir este formulario), o cualquier página web que no sea este Google Form.
- 6- Tené presente que el teclado no reconoce la tecla "TAB" para escribir código Java, por lo que deberás usar 3 (tres) veces la tecla "ESPACIO" para poner sangrías (indentar/tabular) al alinear el código.
- 7- En caso de no encontrar el tipo de una tecla, podrás usar el mapa de caracteres, que podés encontrarlo en: "Tecla Windows + R" y en el cuadro de texto escribir "charmap", y apretar "Enter". También podés encontrarlo en "Inicio--> Accesorios ---> Herramientas del Sistema ---> Mapa de caracteres".
- 8- El examen es individual. No podrás consultar con tus compañeros sobre ningún tema.
- 9- Cuando en el punto siguiente el formulario te pida la dirección de correo electrónico, tenés que declarar el que usaste para inscribirte a la certificación.

En los próximos días recibirás un correo electrónico con los resultados del examen.

Mucha suerte y ¡a trabajar en los ejercicios!

**\*Obligatorio**

**Dirección de correo electrónico \***

Tu dirección de correo electrónico

Apellidos \*

Tu respuesta

Nombres \*

Tu respuesta

DNI (con puntos) \*

Tu respuesta

Sede de Certificación \*

Elige



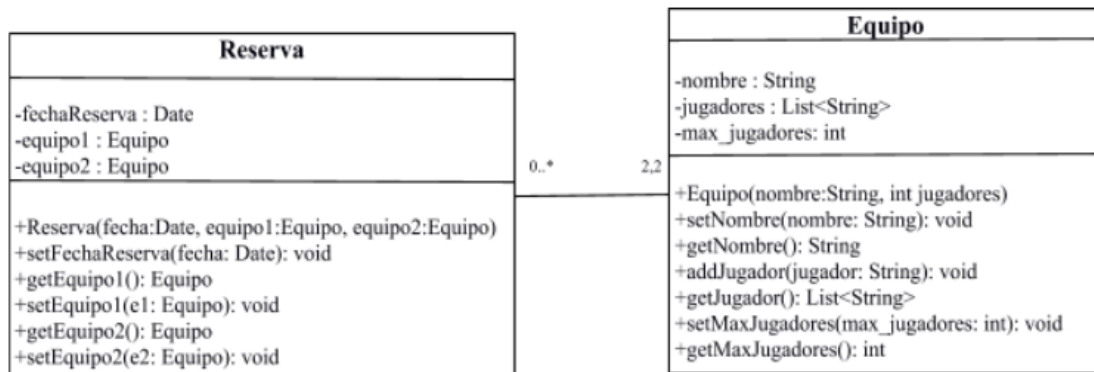
## Contexto - Asignación de Canchas de Futbol 5

El programa 111Mil ha observado que tanto los instructores como los 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 así 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 asignación de las canchas de fútbol 5 correspondiente.

## Ejercicio 1 - Implementar desde el diagrama de clases

Un grupo de programadores comenzaron con la codificación y nos solicitan, al equipo 111Mil, completar la misma con la implementación (código Java) de los métodos y la definición de las variables que faltan. Por favor, ayuda a completar la implementación de los métodos que faltan y la definición de las variables.



## 1a- Completar:

```

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

}
  
```

Tu respuesta

## 1b- Completar:

```

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

// AGREGAR LA DEFINICION DE VARIABLES QUE FALTAN

Tu respuesta

1c- Completar:

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

Tu respuesta

## 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 le resuelve 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.

Tu respuesta

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

El siguiente método realizarReserva reserva una cancha para dos equipos. Ahora habría que agregarle funcionalidad 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.setFechaReserva(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.

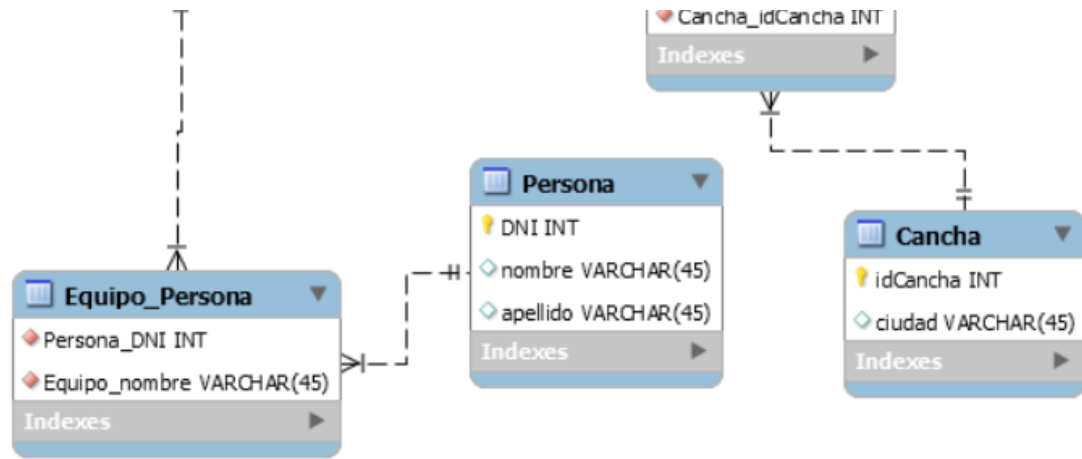
Tu respuesta

## Ejercicio 4 - Interpretación de DER

Algunos alumnos del curso de 111Mil han realizado el siguiente Diagrama de Entidades y Relaciones (DER) pero el profesor tiene algunas dudas sobre el mismo.

A partir del DER, responda Verdadero (V) o Falso (F) a las siguientes afirmaciones (todas las afirmaciones tienen que tener un valor asignado. Puede haber varios V y varios F):





	VERDADERO	FALSO
A- Entre las tablas Reserva y Cancha existe una relación 1 a 1	<input type="radio"/>	<input type="radio"/>
Entre las tablas Equipo y Reserva existe una relación N a 1	<input type="radio"/>	<input type="radio"/>
C- Entre las tablas Persona y Equipo_Persona existe una relación N a N	<input type="radio"/>	<input type="radio"/>
D- La clave primaria de la tabla Reserva es Cancha_idCancha	<input type="radio"/>	<input type="radio"/>
E- La clave foránea de la tabla Equipo es Reserva_idReserva	<input type="radio"/>	<input type="radio"/>
F- La tabla Cancha no tiene clave foránea	<input type="radio"/>	<input type="radio"/>
G- La tabla Equipo no tiene clave primaria	<input type="radio"/>	<input type="radio"/>

## 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 tienen que abrir o cerrar algunos clubes. Por favor, ayúdalos a realizar la consulta SQL:

Tu respuesta

Te recordamos revisar tu examen antes de presionar el botón "ENVIAR", ya que una vez que lo hagas, tus respuestas serán enviadas y no podrás modificarlas.

Éxitos!

Página 1 de 1

ENVIAR

Nunca envíes contraseñas a través de Formularios de Google.

Este contenido no ha sido creado ni aprobado por Google. [Informar sobre abusos](#) - [Condiciones del servicio](#) - [Otros términos](#)

Google Formularios