**Práctica 1**

Para la primera práctica se han elaborado clases para representar un sistema de reserva de pistas de baloncesto. Durante el desarrollo se han tomado ciertas decisiones sobre la implementación del programa:

Ejercicio 1:

* El atributo *usoMaterial* de la clase Material se define como verdadero si es para uso en interiores y falso en el caso contrario.
* El atributo *tipo* de la clase Pista se define como verdadero en caso de que la pista sea interior y falso en caso de que sea exterior.
* Los atributos de tipo “enum” se definen aparte para comodidad de uso en las diferentes clases. Se define la opción NONE para su uso en constructores y errores.
* Se añade el atributo *dni* en la clase Jugador para facilitar su identificación

Ejercicio 2:

* Se crea la clase FileManager para poder leer y escribir en ficheros de texto. Esta clase incluye funciones para guardar con diferente formato los datos de las clases requeridas por los gestores (Jugadores, Materiales, Pistas…)
* Se agregaron mensajes de logs que notifican al usuario sobre la acción realizada (agregado, modificado, cancelado).
* Se crea la clase Reserva, utilizando el patrón de diseño “Factoría”, de la cual heredan “ReservaAdultos”, “ReservaFamiliar” y “ReservaInfantil”. Después añadimos otra factoría (ReservaTypeFactory) de la cual heredan ReservaBonoFactory y ReservaIndividualFactory.

Ejercicio 3:

* En los gestores se implementaron métodos que verifican la existencia de elementos (como pistas o reservas) antes de agregarlos o modificarlos.
* Se crea un main principal (Main.java) que, mediante un menú, deriva a un menú particular para cada gestor (MainUsuarios.java, MainPistas.java y MainReservas.java)

Todas las clases que hemos creado a lo largo de la elaboración de esta práctica se han agrupado en el paquete que hemos llamado es.uco.practica1, aunque creemos que, para un futuro podría ser más conveniente modificar esta distribución

**Dificultades encontradas**

En general ha sido una práctica compleja debido a que ningún miembro del grupo había trabajado antes con el lenguaje Java ni estábamos familiarizados con el IDE Eclipse.

Somos conscientes de que el código necesita arreglos y bastante pulido, pero debido a la carga de trabajo de otras asignaturas nos ha sido complicado disponer del tiempo que requiere la elaboración de esta práctica, por lo que seguiremos trabajando para que podamos mejorar el código que adjuntamos en esta entrega.

**Referencias**

* https://chatgpt.com
* <https://stackoverflow.com/questions/10631715/how-to-split-a-comma-separated-string>
* <https://www.oracle.com/technical-resources/articles/java/javadoc-tool.html>

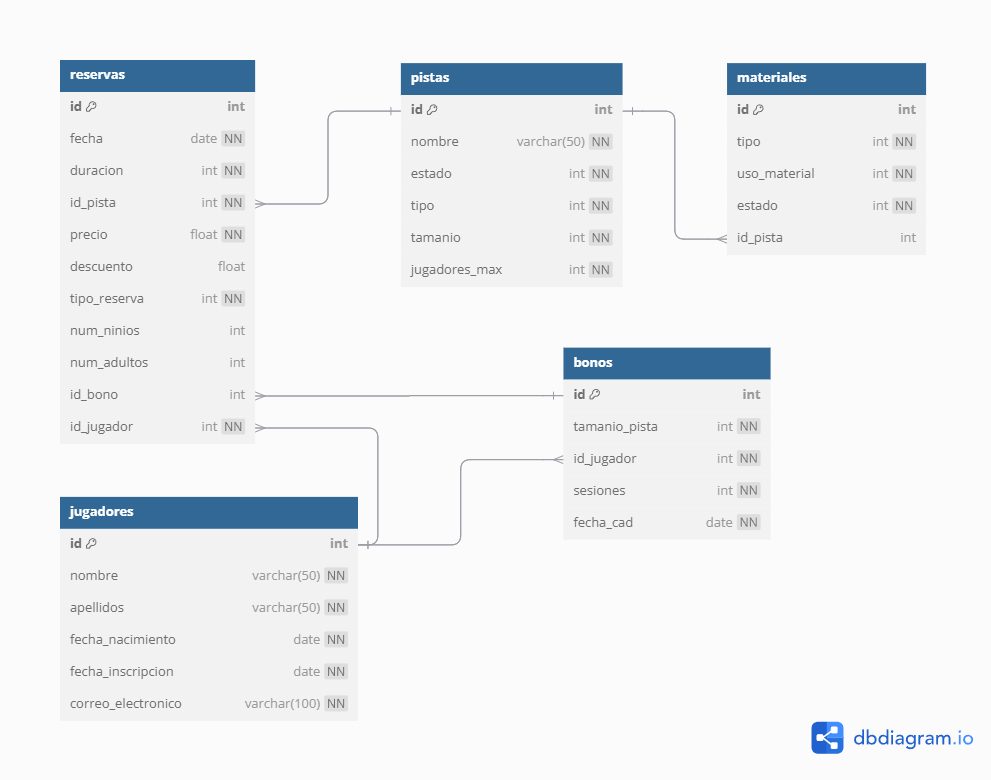
**Práctica 2**

Para la segunda práctica (…)Durante el desarrollo se han tomado ciertas decisiones sobre la implementación del programa:

Ejercicio 1:

* (…)

**Modelo relacional de la BBDD:**



**Modelo Entidad-Relación:**

**Entidades y sus atributos:**

1. **Jugadores**
   * id (PK)
   * nombre
   * apellidos
   * fecha\_nacimiento
   * fecha\_inscripcion
   * correo\_electronico
2. **Pistas**
   * id (PK)
   * nombre
   * estado
   * tipo
   * tamanio
   * jugadores\_max
3. **Materiales**
   * id (PK)
   * tipo
   * uso\_material
   * estado
   * id\_pista (FK) → **Pistas.id**
4. **Bonos**
   * id (PK)
   * tamanio\_pista
   * id\_jugador (FK) → **Jugadores.id**
   * sesiones
   * fecha\_cad
5. **Reservas**
   * id (PK)
   * fecha
   * duracion
   * id\_pista (FK) → **Pistas.id**
   * precio
   * descuento
   * tipo\_reserva
   * num\_ninios
   * num\_adultos
   * id\_bono (FK) → **Bonos.id**
   * id\_jugador (FK) → **Jugadores.id**

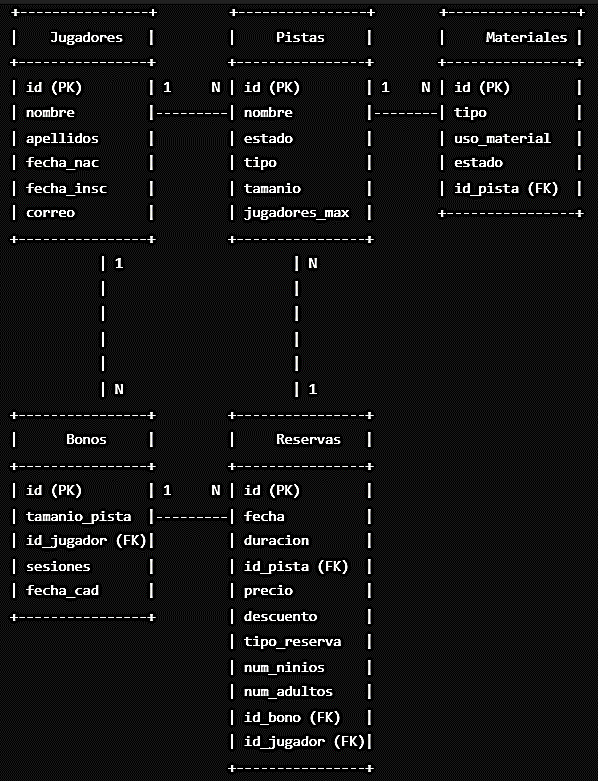
**Relaciones:**

* **Jugadores** a **Bonos**: Un jugador puede tener uno o más bonos. Relación 1 (Un jugador a varios bonos).
* **Jugadores** a **Reservas**: Un jugador puede realizar varias reservas. Relación 1 (Un jugador a varias reservas).
* **Pistas** a **Materiales**: Una pista puede tener varios materiales. Relación 1 (Una pista a varios materiales).
* **Pistas** a **Reservas**: Una pista puede ser reservada varias veces. Relación 1 (Una pista a varias reservas).
* **Bonos** a **Reservas**: Una reserva puede estar asociada a un bono, y un bono puede estar asociado a varias reservas. Relación 1 (Un bono a varias reservas).

**Esquema de relaciones:**

1. **Jugadores** (1) — (N) **Bonos**
2. **Jugadores** (1) — (N) **Reservas**
3. **Pistas** (1) — (N) **Materiales**
4. **Pistas** (1) — (N) **Reservas**
5. **Bonos** (1) — (N) **Reservas**

**Diagrama ER (Conceptual)**



Ejercicio 2:

* (…)

**Dificultades encontradas**

En