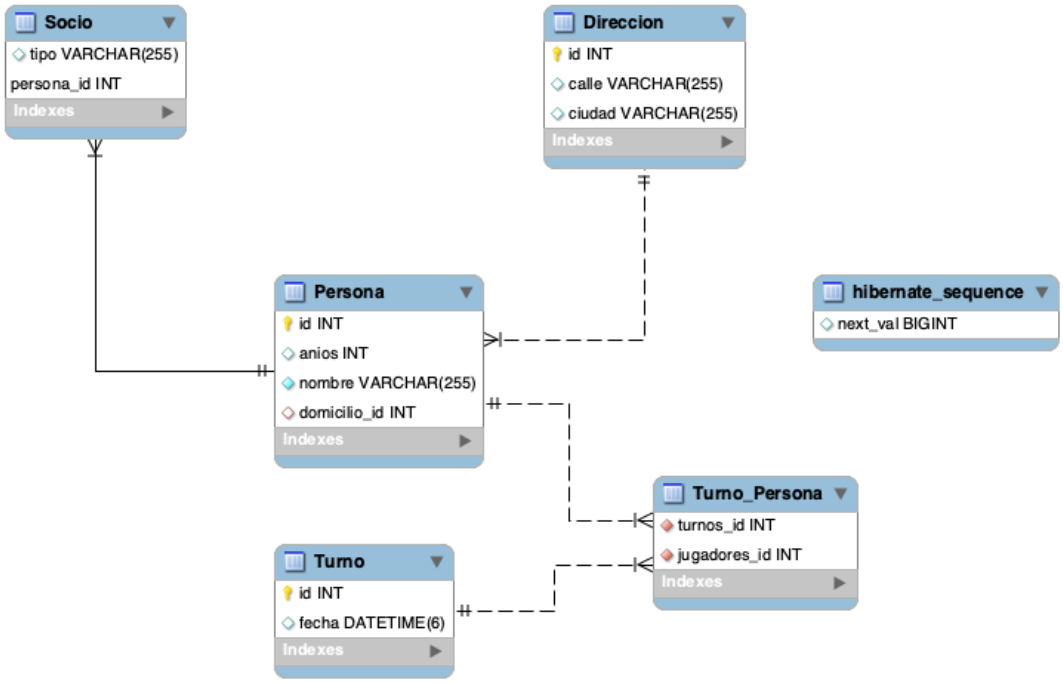
Trabajo Práctico No. 2: Anotaciones JPA y consultas JPQL

1) En base al Video 5: JPA-Relaciones entre entidades, implementar los mapeos de las entidades Turno, Dirección, Persona, y Socio a tablas en una base de datos relacional (Derby o MySQL), así como las relaciones entre ellos. Para testear, probar casos de altas, bajas y modificaciones de entidades.
   Video 5: https://www.loom.com/share/5433281ad87342dcaafad8b787e0eeba



2) En base al ejercicio anterior, implementar las siguientes consultas:
   a) recuperar todas las personas asignadas a un turno.
   b) recuperar todas las personas asignadas a un turno, y marcar cuales de ellas son socios.
   c) recuperar todas las personas que viven en una ciudad predeterminada.
   d) En los casos anteriores, evaluar que sucede al utilizar las opciones FetchType.LAZY o FetchType.EAGER en las anotaciones. ¿Nota alguna diferencia?, ¿a qué se debe?

3) Suponga un sistema para gestionar torneos de futbol 7, donde las principales entidades son: Equipo, Jugador, y Torneo. Cada equipo incluye 7 jugadores (uno de ellos debe ser arquero),

hasta 3 suplentes, y 1 director técnico. Cada equipo puede o no representar a una entidad o firma comercial. Los jugadores desempeñan una posición fija dentro del equipo (por ej., arquero, defensa, mediocampo, delantera). Un jugador solo puede estar fichado en un equipo. Un torneo tiene un nombre e incluye un conjunto prefijado de equipos.

a) realizar un modelamiento orientado a objetos de las entidades y sus relaciones.
b) diseñar un DER que permita mapear el modelo orientado a objetos anterior.
c) implementar el mapeo entre a) y b) utilizando anotaciones de JPA.
d) implementar servicios para altas, bajas y modificaciones de las distintas entidades
e) implementar servicios para agregar jugadores a equipos, e inscribir equipos a torneos
f) implementar servicios para buscar todos los jugadores de un equipo, y todos los jugadores de un torneo.

4) Extender el diseño anterior con la siguiente información. Al inicio del torneo se establecen los encuentros (todos contra todos). A medida que se van jugando los encuentros de cada fecha, se va llevado un registro del resultado de cada partido, de los goles que anotó cada jugador, y de los puntos que obtuvo cada equipo en una partida que jugó. Luego de cada partido, puede que se reporten jugadores lesionados, o expulsados, que no podrán jugar partidos por un período de tiempo determinado. En caso que un equipo no pueda completar su formación, se declarará ganador al equipo contrario.
   a) adecuar el modelo de objetos, y también el esquema de tablas.
   b) agregar las consultas correspondientes para registrar la nueva información en los objetos y tablas del sistema
   c) implementar un servicio que genere la tabla de posiciones, a una fecha dada.
   d) Implementar un servicio que genere la tabla de goleadores, a una fecha dada.

5) Extender el diseño anterior considerando que los equipos se dividen en grupos de igual cantidad de equipos, y dentro de cada grupo se juega e la modalidad todos contra todos. Una vez concluida la fase de grupos, se seleccionan los 2 mejores equipos de cada grupo, y se arman llaves de eliminación, que se van jugando hasta que uno de los equipos resulta ganador.
   a) adecuar el modelo de objetos, y también el esquema de tablas.
   b) agregar las consultas correspondientes para registrar la nueva información en los objetos y tablas del sistema
   c) ajustar el diseño de los servicios del ejercicio anterior, para adecuarse al esquema de grupos.

Nota: para testear los ejercicios 3-5 se recomienda definir un conjunto de datos básico que contenga: nombres de jugadores y equipos.

Ejercicio Integrador

1) Considere el diseño de un registro de estudiantes, con la siguiente información: nombres, apellido, edad, género, número de documento, ciudad de residencia, número de libreta universitaria, carrera(s) en la que está inscripto, antigüedad en cada una de esas carreras, y si se graduó o no. Diseñar el diagrama de objetos y el diagrama DER correspondiente.

2) Implementar consultas para:
   a) dar de alta un estudiante
   b) matricular un estudiante en una carrera
   c) recuperar todos los estudiantes, y especificar algún criterio de ordenamiento simple.
   d) recuperar un estudiante, en base a su número de libreta universitaria.
   e) recuperar todos los estudiantes, en base a su género.
   f) recuperar las carreras con estudiantes inscriptos, y ordenar por cantidad de inscriptos.
   g) recuperar los estudiantes de una determinada carrera, filtrado por ciudad de residencia.

3) Generar un reporte de las carreras, que para cada carrera incluya información de los inscriptos y egresados por año. Se deben ordenar las carreras alfabéticamente, y presentar los años de manera cronológica.

Nota: las consultas deben ser resueltas mayormente en JPQL, y no en código Java.