Universidad FASTA – Facultad de Ingeniería - Ingeniería Informática
Programación B - 28/10/2022 - 2do. Parcial – Evaluación Práctica
Apellido y Nombre: **NOTA Práctica:**

Los creadores del juego *Racer* han lanzado la versión online de su popular juego de escritorio. Los jugadores de esta versión online se suscriben por internet y juegan partidas contra otros jugadores. El jugador simplemente se conecta a la web de *Racer* con su usuario y contraseña e indica su voluntad de participar de una partida. La plataforma web, una vez que reúne a 4 jugadores (pueden ser novatos o expertos) que quieren jugar, simplemente da inicio a una partida. Las reglas del juego son las mismas que las del juego en su versión escritorio.

Nota:	Toda la dinámica de suscripción de jugadores, ingreso a la plataforma, jugada de partidas y registro de estadísticas de cada jugador, exceden el alcance de este examen. Simplemente se explica el contexto a los efectos de una mejor comprensión del problema a resolver. El alumno no tiene que actualizar las estadísticas, sólo accede a la Base de Datos de estadística de los jugadores para elaborar un ranking.

Una base de datos PostgreSQL llamada *racerDB* almacena en una tabla *player,* los datos de los jugadores registrados (nombre de usuario, contraseña, nivel [E=experto / N=novato], país de origen del jugador). De cada jugador se almacenan en el mismo registro, las estadísticas de todas las partidas que jugó: cantidad de partidas jugadas, cantidad de partidas ganadas, cantidad de preguntas contestadas, cantidad de respuestas contestadas correctamente y cantidad total de jugadas que necesitó para culminar dichas partidas. El campo del país de origen del jugador guarda integridad referencial con la tabla *country*, que contiene registros con los distintos países del mundo.

Una vez finalizada cada partida, la plataforma web actualiza las estadísticas de los jugadores que participaron de la misma.

Los creadores de *Racer* nos dan acceso a esa Base de Datos de estadísticas de los jugadores y nos piden elaborar un ranking de jugadores, sabiendo que la fórmula para calcular el puntaje varía si el jugador es novato o experto, de acuerdo a las siguientes fórmulas (a mayor puntaje, mejor posición en el ranking):

- Puntaje jugador *experto*:

  Cant. de partidas jugadas +
  *Cant. de partidas ganadas * 100 +*
  Cant. de preguntas realizadas +
  *Cant. de preguntas contestadas correctamente * 2*

- Puntaje jugador *novato*:

  Cant. de partidas jugadas +
  *Cant. de partidas ganadas * 125 +*
  Cant. de preguntas realizadas +
  *Cant. de preguntas contestadas correctamente * 3 +*
  Promedio de jugadas por partida

Para mostrar el ranking, al crear la clase *RaceController* se deberán inicializar las listas de Países y la de Jugadores haciendo lo siguiente:

- **Extraer de la Base de Datos, los datos de los países con la siguiente consulta:**

```
SELECT countryid, name FROM country
```

  Con cada registro, crear un objeto de tipo *Country* y almacenarlo en una lista o hash de países

- **Extraer de la Base de Datos, los datos de los jugadores con la siguiente consulta:**

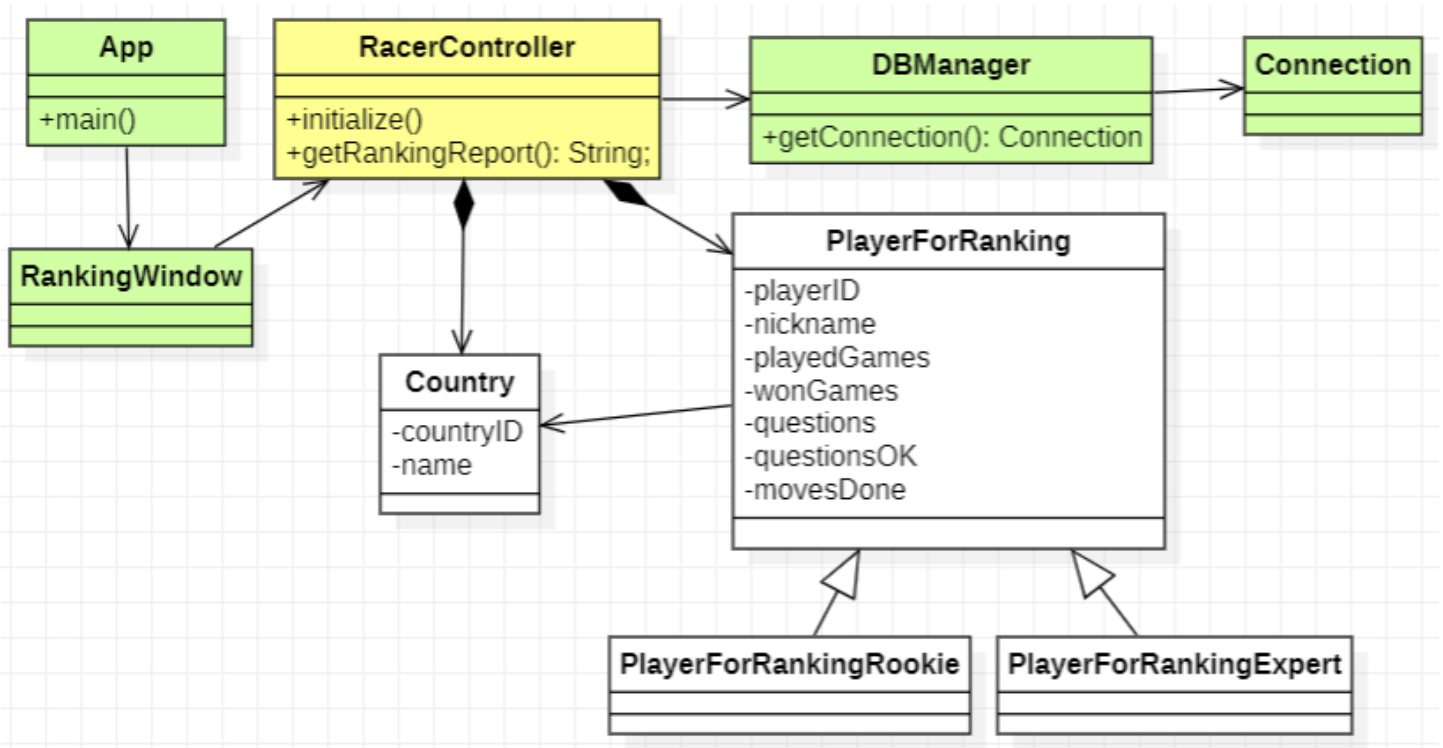
```
SELECT playerID, nickname, level, playedGames, wonGames,
       questions, questionsOK, movesDone, countryID
  FROM player
```

  Con cada registro, crear un jugador del tipo adecuado. El atributo *country* del jugador debe asignarse con el objeto correspondiente, obteniéndolo de la lista de países llenada previamente. El jugador creado, almacenarlo en una única lista de jugadores.

- **Obtener el ranking de jugadores y mostrarlo en la ventana,** detallando polimórficamente, para cada jugador, su *nickname*, su país de origen, el nivel (experto o novato), partidas jugadas, partidas ganadas, cantidad de preguntas, cantidad de preguntas respondidas correctamente, cantidad de movidas totales, promedio de movidas por partida y el puntaje. **Identificar y mostrar al final, al jugador de mayor puntaje:**

```
=================================================================================================
RANKING jugadores Racer
-------------------------------------------------------------------------------------------------
Nickname     Nivel  Pais de origen     PJ     PG     #?     OK     #Movidas     MpP    Puntaje
=================================================================================================
the best      Experto     Alemania     10     2      100    80     250          25     470
el genio      Novato      Argentina    5      0      200    50     150          30     385
twinpick      Novato      Uruguay      30     5      400    300    990          33     1988
mary pop      Experto     Argentina    60     20     500    350    1380         23     3260
intrepid      Experto     Brasil       20     1      220    80     760          38     500
-------------------------------------------------------------------------------------------------
Cantidad de Jugadores: 5
El ranking lo encabeza mary pop
```

Diagrama de Clases



Observaciones

- Importar el proyecto dado desde *File -> import -> Existing Maven project*
- En el Diagrama de Clases se muestran en verde las clases entregadas por la cátedra que no requieren modificación porque están completas. En amarillo, las clases que se entregan parcialmente desarrolladas. En blanco, las que debe desarrollar íntegramente el alumno.
- Las clases en blanco muestran únicamente los atributos de las clases. Se deberán **incorporar los métodos que se consideren apropiados para resolver el ejercicio**.
- En PostgreSQL se debe crear una base de datos con el nombre *racerDB* y luego, para crear las tablas y cargarle datos, se deberá utilizar el script *create.SQL*
- **Resolver aplicando necesariamente los principios del paradigma POO: encapsulamiento, herencia y polimorfismo.**
- **Este examen se aprueba con nota mínima de 5. Se alcanzará esta nota habiendo resuelto de forma correcta, mínimamente el 50% de la evaluación teórica y el 50% de la evaluación práctica.**