



TALLER N°2 – Base de Datos

Docente: Eric Ross Cortés

Ayudante: Sebastián Álvarez T.

Fecha Entrega del Enunciado : Jueves 28 de Mayo

Fecha de Término : Lunes 25 de Junio

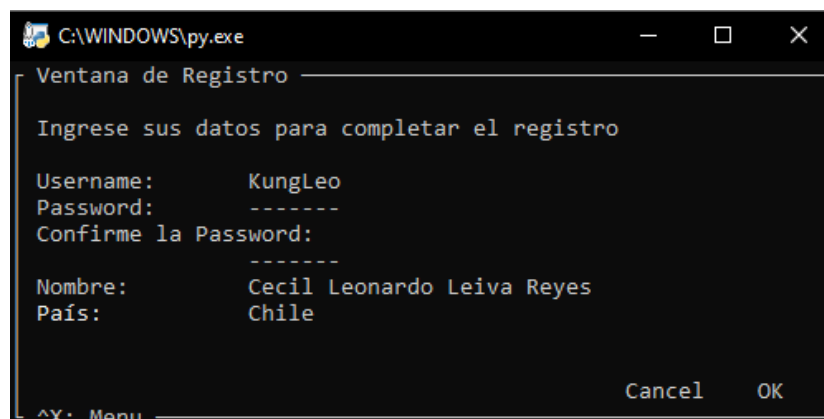
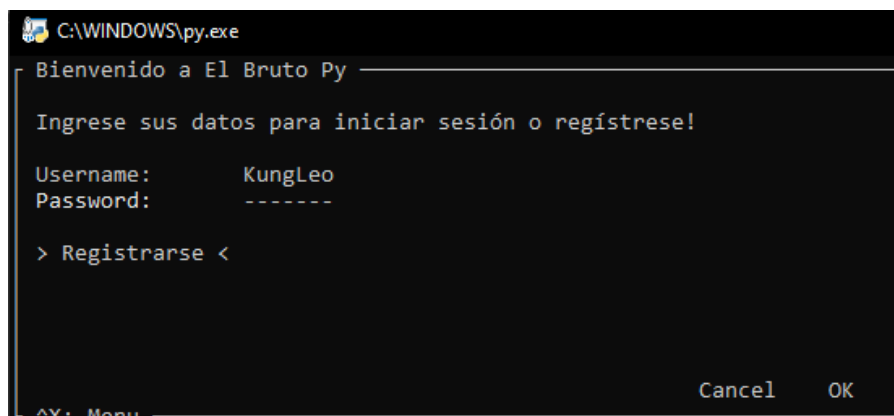
OBJETIVO

- Diseñar una base de datos a través del MER y su posterior transformación al modelo relacional.
- Desarrollar un programa en Python que pueda conectarse a la base de datos de PostgreSQL y que satisfaga los requerimientos del problema.

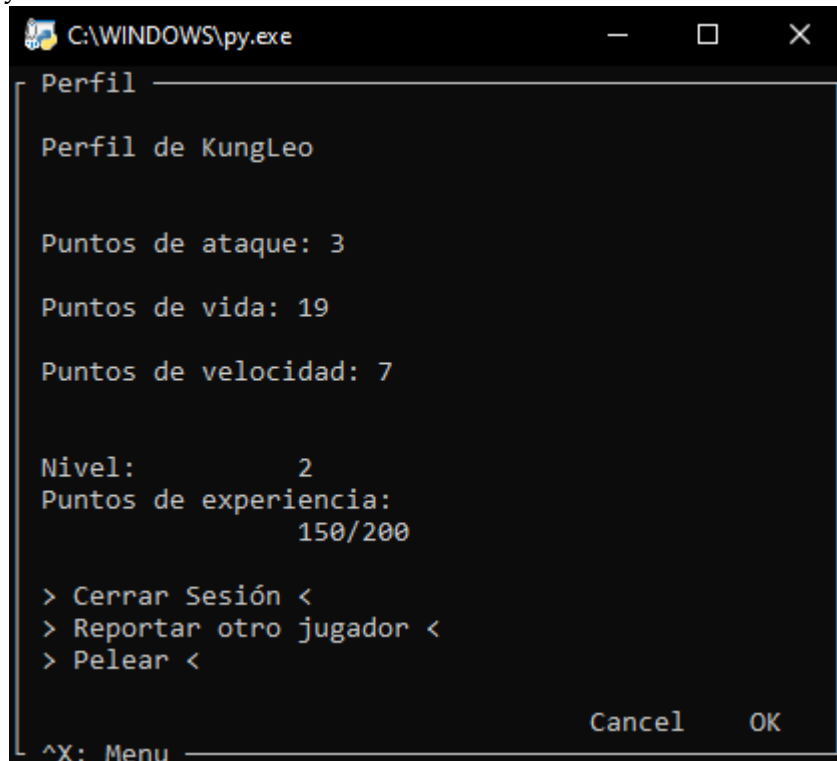
DOMINIO DEL PROBLEMA

El Bruto es un videojuego flash de internet, un pionero de los juegos online flash. Hoy como es sabido sigue existiendo, sin embargo, Flash Player dejará de funcionar pronto. Usted como fan de El Bruto quiere desarrollar un juego similar pero mucho más reducido para jugar con sus amigos y otros usuarios en línea. Cabe mencionar que este juego que usted creará solo es un prototipo que en el futuro puede mejorar.

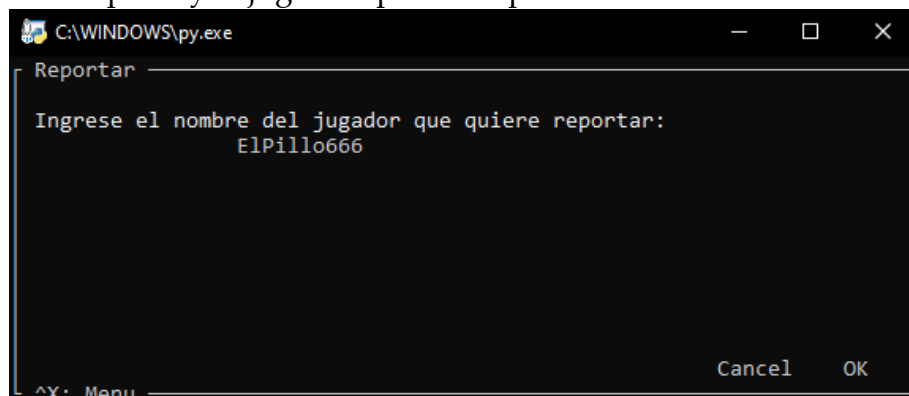
El simulador del juego inicialmente debe contar con un login, donde se soliciten el nickname del jugador y su contraseña para iniciar sesión, en caso de que un usuario no esté registrado, este podrá registrarse. De los **jugadores** interesa su nombre completo, nickname, contraseña y país al cual pertenece. A continuación, se muestra una imagen de cómo debería verse el login y el registro de usuario:



1. **Perfil:** Una vez iniciada la sesión, el jugador podrá ver su perfil con las estadísticas de su Avatar. En esta ventana también se podrá elegir la opción de luchar. Las estadísticas del Avatar se generarán aleatoriamente cuando el jugador se registre. El avatar cuenta con puntos de ataque, puntos de vida y velocidad. También cuenta con puntos de experiencia. Todas estas características deben mostrarse en el perfil. Debe haber una opción de Luchar, una de reportar jugador y otra de Cerrar Sesión.

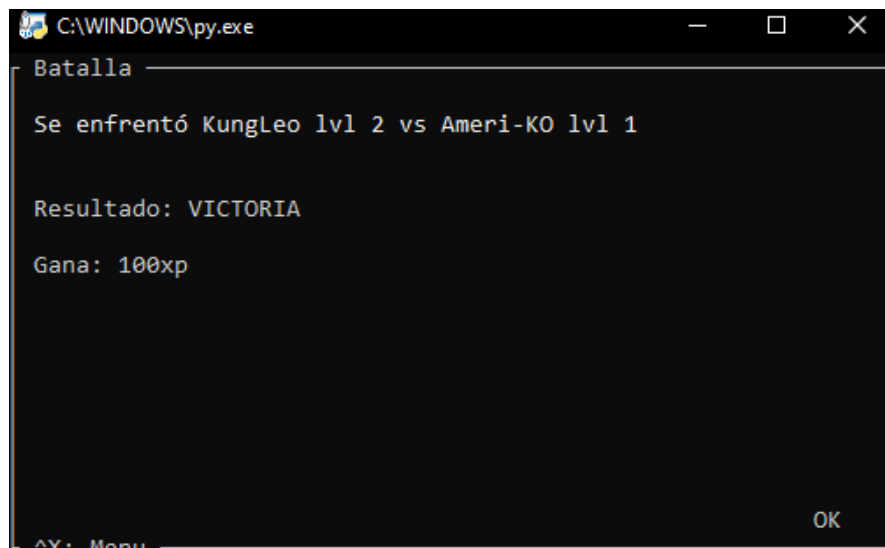


2. **Generación del Avatar:** Cuando el jugador se registre, automáticamente obtendrá un avatar con estadísticas determinadas aleatoriamente. Puede obtener un ataque entre 1 y 3, puntos de vida entre 10 y 20, y velocidad entre 1 y 10.
3. **Reportar Jugador:** Al seleccionar esta opción debe abrirse una ventana en donde se pida el nombre del jugador a reportar y debe contar con la opción de cancelar (volver atrás) o aceptar. Cada reporte debe quedar almacenado en la base de datos. Se debe tener la información del jugador que hizo el reporte y el jugador que fue reportado.



4. **Lucha:** Cuando el jugador selecciona luchar, será emparejado aleatoriamente con otro jugador registrado en el juego que tenga como máximo 1 nivel de diferencia (si mi personaje es nivel 2 puede enfrentarse a personajes de nivel 1, 2 o 3). El proceso de lucha funciona de la siguiente forma:

- Los jugadores se atacarán entre turnos, primero ataca uno, luego el otro.
- El jugador con más velocidad atacará primero.
- Cada vez que un jugador ataca, el otro jugador recibe daño a sus puntos de vida igual al ataque del jugador que lo atacó.
- Dependiendo de la velocidad del jugador, tendrá una probabilidad de esquivar el ataque. (Cada punto de velocidad representa un 1% de probabilidad de esquivar, el máximo será 80%). Esto influirá en cómo se resuelve la pelea.
- Al finalizar el combate, si el jugador ganó la pelea ganará 100 puntos de experiencia. Si perdió solo ganará 20.
- Un jugador solo podrá pelear 5 veces al día (cree una función que reinicie las oportunidades cada día). Tip: Puede registrar para el jugador la cantidad de oportunidades que tiene disponible y guardar qué día fue la última vez que se abrió el juego.
- No necesita animar ni representar las peleas, todo este proceso será automático. Cuando el jugador elija pelear, en pantalla debe aparecer contra quien se enfrentó (el nombre del jugador, y el nivel) y posteriormente en pantalla debe decir quién ganó y cuántos puntos de experiencia recibió.




5. **Subir de nivel:** Los jugadores siempre empezarán en nivel 1 al crear su personaje. Cada nivel tiene una cantidad de experiencia necesaria para alcanzarlo. Esta cantidad se calculará de la siguiente forma: $100 + (\text{nivel del personaje} * 50)$.

Entonces, para alcanzar el nivel 2 son necesarios $100 + (1*50) = 150$ puntos de experiencia.

Para alcanzar el nivel 30 son necesarios $100 + (29*50) = 1550$ puntos de experiencia.

Cada vez que se sube de nivel, el avatar del jugador obtiene puntos extra en sus características de manera aleatoria. Puede recibir +1 punto en ataque, o +3 puntos de velocidad, o +3 puntos de vida. Solo puede recibir una de estas bonificaciones aleatoriamente cuando suba de nivel.

6. **Administrador:** Los administradores pueden banear jugadores reportados. Al iniciar sesión el Administrador verá la lista de todos los jugadores y cuántas veces han sido reportados (La lista está ordenada por la cantidad de reportes de cada usuario). Éste también tendrá una opción de banear a un jugador escribiendo su nickname. Un jugador baneado no puede iniciar sesión. Es importante mencionar que los administradores no tienen un avatar para jugar.



```
C:\WINDOWS\py.exe
- Perfil Administrador -
Bienvenido Applo

Lista de Jugadores:

ElPillo666 - Cantidad de reportes: 10
Ameri-KO - Cantidad de reportes: 7
Vardoc1 - Cantidad de reportes: 1
KungLeo - Cantidad de reportes: 0

> Banear un Jugador <
> Cerrar Sesión <

- ^X: Menu -
```

CONSIDERACIONES

- Para los datos que se obtengan de manera aleatoria utilice funciones Random en Python.
- Utilice la consola de la terminal para mostrar los menús, opciones, etc.
- El archivo zip (o rar) debe llamarse **apellidointegrante1_apellidointegrante2.zip**. Ejemplo: Araya_Orellana.zip
- La base de datos debe llamarse apellidointegrante1_apellidointegrante2.
- El taller debe ser realizado en grupos de 2 alumnos.
- Las consultas se realizarán al correo del ayudante. (**Ayudante:** Sebastián Álvarez Trujillo, **correo:** sebastian.alvarez@alumnos.ucn.cl)
- En caso de copia (códigos completamente iguales) **la nota será 1.0** para todos los que entregaron el mismo código y el caso se presentará ante el consejo.
- Ver capítulo 4 del curso para la ayuda con las sentencias sql que necesite.

FECHAS DE ENTREGA

Entrega N.º 1: Jueves 4 de Junio (23:59). DER y MR con documentación. (20%).

Debe entregarse vía educa.

Entrega N.º 2: Jueves 11 de Junio (23:59). Avance - Conexión de Python con PostgreSQL. (40%).

DEBE CUMPLIR:

- Login
- Registro de usuario

El nombre del archivo debe ser **apellidointegrante1_apellidointegrante2.zip** (o .rar)

Entrega N.º 3: Lunes 25 de Junio (23:59).

Sistema completo conectado con PostgreSQL. (40%)

Debe satisfacer todos los requerimientos.

Haga un backup de la base de datos y adjúntela al Zip con el programa.

El nombre del Archivo debe ser **apellidointegrante1_apellidointegrante2.zip** (o .rar).