

Mecanismos de sincronización empleados

Para el programa se utilizó como protagonista un simple mutex que permitió generar una sección crítica en la carrera de caballos, garantizando así que estos no serían dados como ganadores hasta que todos los demás caballos hayan terminado de moverse virtualmente. Por otra parte, se empleó la espera activa en el proceso principal para darle una sensación de plataforma real al programa, teniendo “tiempos de carga”.

La parte interesante de este programa es que puede ser visto como un problema de **productor – consumidor** llevado a un caso real y que puede ser muy útil en caso de que se desee entrar en el mundo del desarrollo de aplicaciones web y juegos en línea.

Lógica de operación

- El estado compartido de este programa se encuentra en las variables que almacenan la posición horizontal de cada uno de los caballos, las cuales sirven para verificar al ganador de la carrera.
- Lo primero que se realiza en cada hilo es el inicio del hilo que mueve a los caballos, este se encarga de hacer operaciones completamente aleatorias (con funciones de la API de Java) y posteriormente notifica al hilo que se encarga de realizar las verificaciones para que este pueda tener en cuenta todos los casos y así dar un veredicto.
- La interacción entre ambos hilos es simple, ya que uno se encarga de notificar al otro y así es como se protege la sección crítica del código.

Desarrollo de entorno

- **Lenguaje de programación:** Java SE 8
- **Entorno de desarrollo y pruebas:** NetBeans 8.2 en Windows 11
- **Bibliotecas:** todas las bibliotecas empleadas (diseñadas por mí) se incluyen en el código fuente del programa, además de las incluidas en el API de Java.

Instrucciones de ejecución y uso:

Para ejecutar este programa, acudir a la carpeta ‘Casino (Programa)’ la cual contiene un archivo .jar que sirve para realizar una ejecución más limpia y amigable, de forma que este archivo debería estar libre de problemas de compatibilidad ya que se ejecuta directamente en la JVM.

El comando para ejecutar el programa es:

```
java -jar "Casino.jar"
```

Una vez ejecutado el programa, es importante aclarar que este tiene unos sencillos menús que buscan ser intuitivos, de forma que se sienta una experiencia de plataforma web de Casino. Se tiene un registro e inicio de sesión de usuarios, sin embargo, para fines de pruebas se tiene un usuario **administrador** que tiene todos los privilegios posibles (tales como borrar usuarios).

Los datos de inicio de sesión de este usuario son:

Usuario: admin

Contraseña: cs2021-01

La carpeta 'save' de este mismo folder se encarga de almacenar todos los usuarios creados, **NO se debe borrar.**

Datos importantes

Algunos de los módulos empleados en este programa (tales como excepciones o los sistemas para el inicio de sesión) fueron retomados de proyectos pasados y reestructurados para que fueran consecuentes con este proyecto. La "base de datos" que se tiene es una idea propia la cual funciona a partir de generación de archivos de objetos los cuales guardan los objetos de tipo usuario y los almacenan en archivos binarios, los cuales son abiertos en cada ejecución del programa y retoman la información de todos estos, además que en todo momento se realiza un autoguardado de los datos (todo esto puede ser visto en el módulo de ManejoArchivos en la clase del mismo nombre).

Toda esta parte "rimbombante" fue puesta con fines de mejora estética, por lo que para observar realmente la implementación paralela se incita al profesor a revisar el archivo "Caballos" del modulo "Juegos" en donde se encuentra el manejo del mutex en la aplicación.

Los otros miembros que se mencionan en el programa eran parte de una idea para tener diversos juegos paralelizables, sin embargo, por una mala administración del tiempo estos fueron desechados y son parte solamente de líneas fantasma que puede que no haya alcanzado a borrar.

Ejecución exitosa

```
D:\Ricardo Ruelas\Facultad de Ingeniería\6to Semestre\Sistemas Operativos\sistop-2022-2\proyectos\2\RuelasRicardo\Casino
(Programa)>java -jar "Casino.jar"

¿Qué deseas realizar?
1.- Iniciar sesion
2.- Registrarse
3.- Salir
Opcion: 1

Usuario: admin
Contraseña: cs2021-01

-> Bienvenido Admin, entrando al sistema...

Centro de control
1.- Caballos
2.- Administrar miembros
3.- Salir
Opcion: 1
```

```
****Bienvenido al juego de carrera de caballos****

Reglas:
1) Seleccionar un caballo de entre los disponibles
2) Se tiene que apostar al menos 100 fichas
3) Los miembros Silver tendran 2 veces su apuesta mas un bono adicional del 50% de su apuesta
4) Los miembros Golden tendran 3 veces su apuesta mas un bono adicional de 75% de su apuesta
5) Los miembros Diamond tendran 5 veces su apuesta mas un bono adicional del 200% de su apuesta
-> Disfruta

Caballos disponibles:
1.- Duncan: ⚡
2.- Sardinilla: ☸
3.- Rosinante: ⚡
4.- Spirit: 🐎
0.- Volver al menú principal
Opcion: 2
Apuesta: 300
```

```
Duncan ->
Sardinilla ->
Rosinante ->
Spirit ->

Duncan gano!

Suerte para la proxima!
Caballos disponibles:
1.- Duncan: +
2.- Sardinilla: ++
3.- Rosinante: +++
4.- Spirit: ■
0.- Volver al menú principal
Opcion: |
```

Nota: debido a que Linux y Windows difieren en sus comandos para limpiar pantalla, se realiza una limpieza “manual” que queda horrible a la vista pero que fue desde mi perspectiva la mejor solución en cuanto a compatibilidad.

Nota 2: en la ejecución mostrada se encuentra el uso principal del programa, sin embargo, todas las demás opciones son completamente funcionales ya que en un principio busque crear una solida plataforma de casino con juegos estables en cuanto a concurrencia, por lo que la creación de usuarios y la eliminación de los mismos desde el administrador son opciones que quedan a elección del profesor en caso de que las desee probar.