Centro Académico de Limón

Tecnológico de Costa Rica

Lenguajes de programación

Proyecto 3

Kakuros

Elaborado por:

Jeremy Madrigal Portilla

Carné: 2019258245

Randall Zumbado Huertas

Carné: 2019082664

Para:

Allan Rodríguez Dávila

Alajuela, 30 de noviembre de 2020

Manual de usuario

Instrucciones de compilación

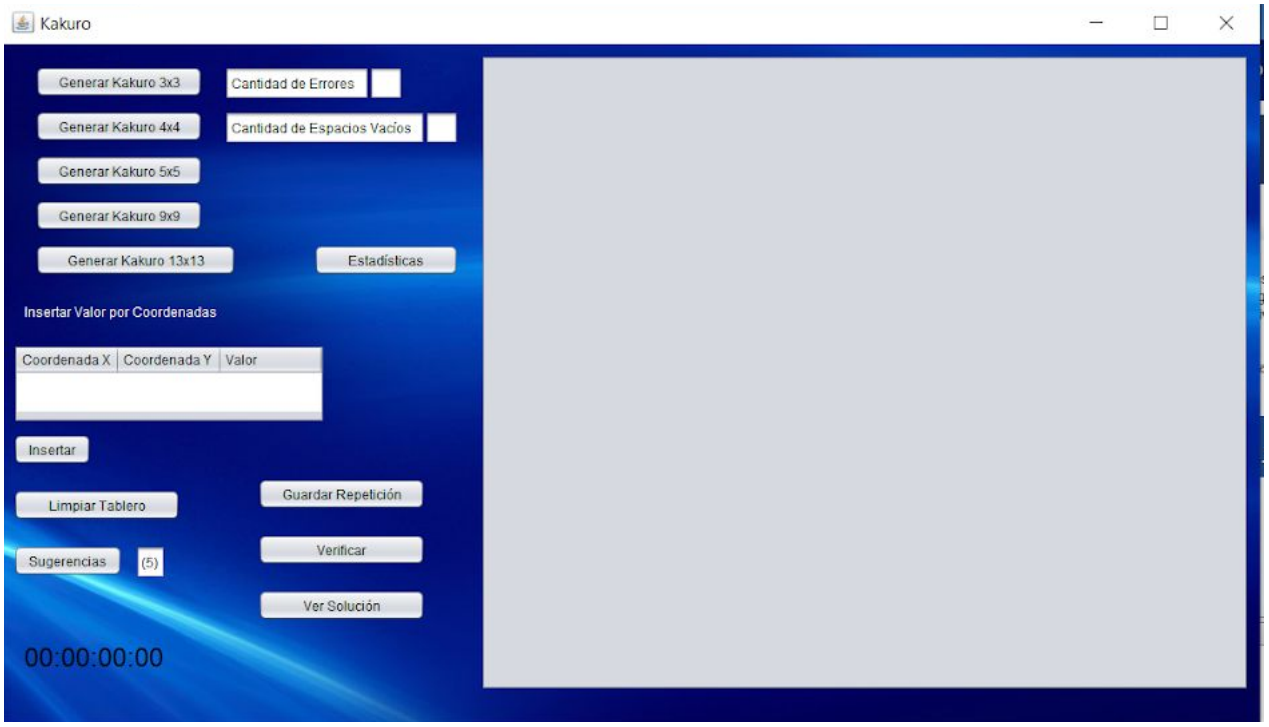
Antes de su compilación es importante establecer las rutas necesarias para poder utilizar la librería jpl en nuestro programa, esto fue aplicado por nosotros en windows utilizando netbeans entonces no tiene instrucciones de compilación como tal.

Ejecución

Para ser ejecutado es tan sencillo como utilizar F6, aunque esto podría ser diferente dependiendo del computador o utilizar el botón de Run Project para darle inicio a nuestro proyecto.

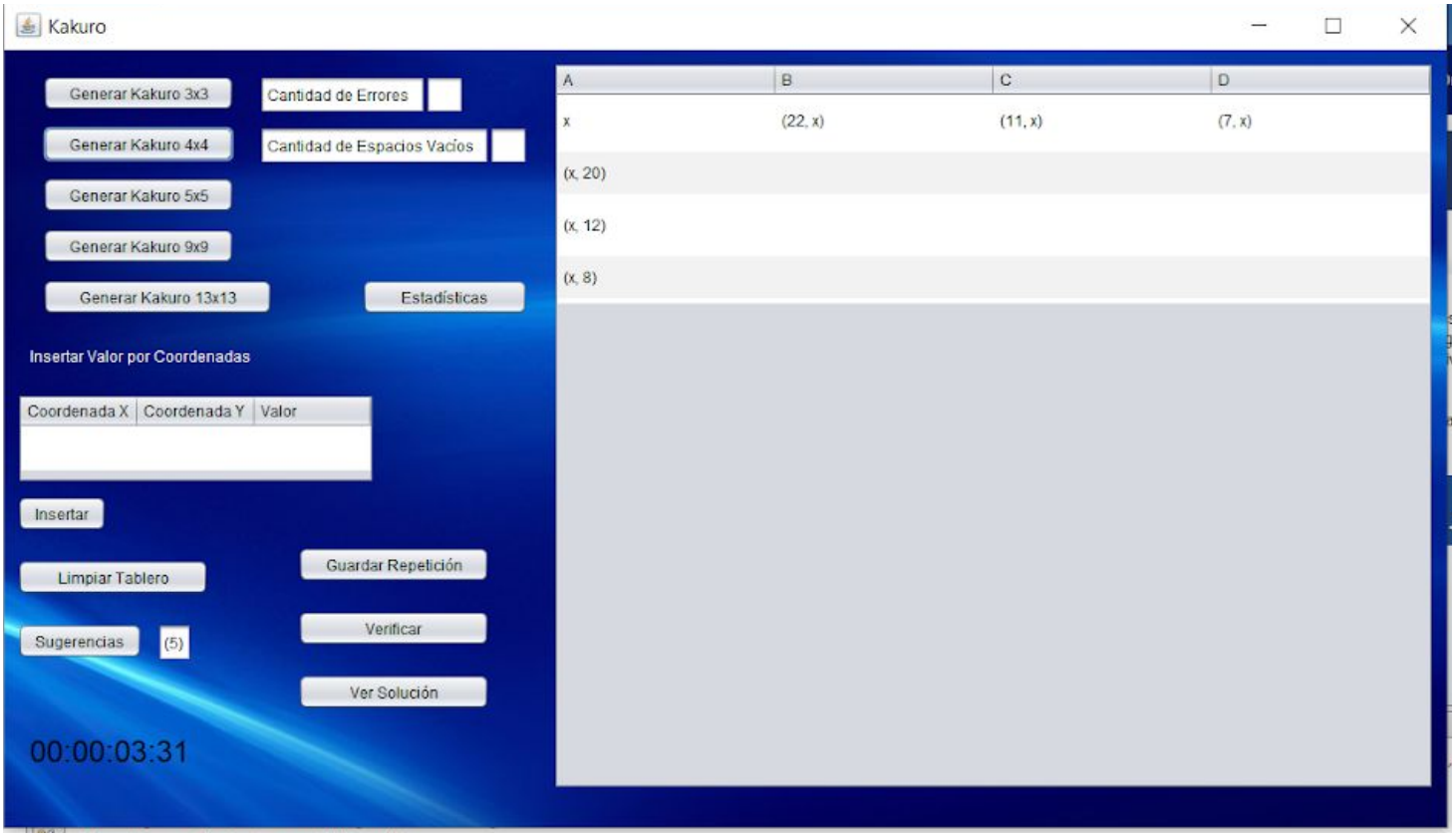
Uso

1. Al iniciar el programa se le desplegará una ventana de la siguiente manera:



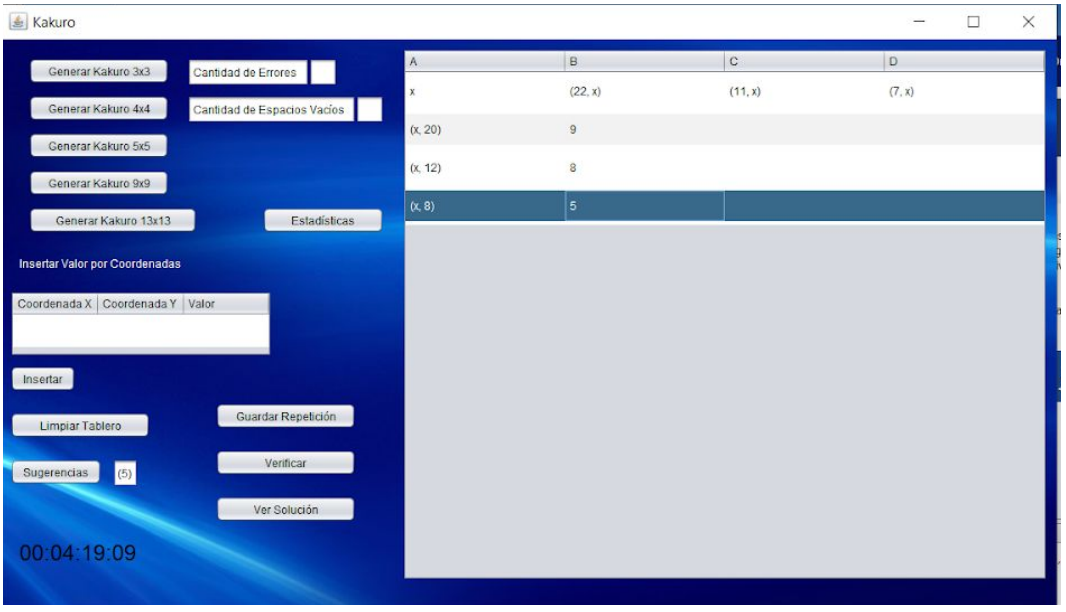
Se pueden observar todas las distintas opciones que hay para el juego, y sólo hace falta seleccionar un tamaño de tablero para empezar.

2. Al seleccionar un tamaño de tablero tiene 5 opciones que serían uno de 3x3, uno de 4x4, uno de 5x5, uno de 9x9 y el de mayor tamaño que es 13x13, para elegir uno sólo debe darle click al botón, con lo cuál desplegará lo siguiente:



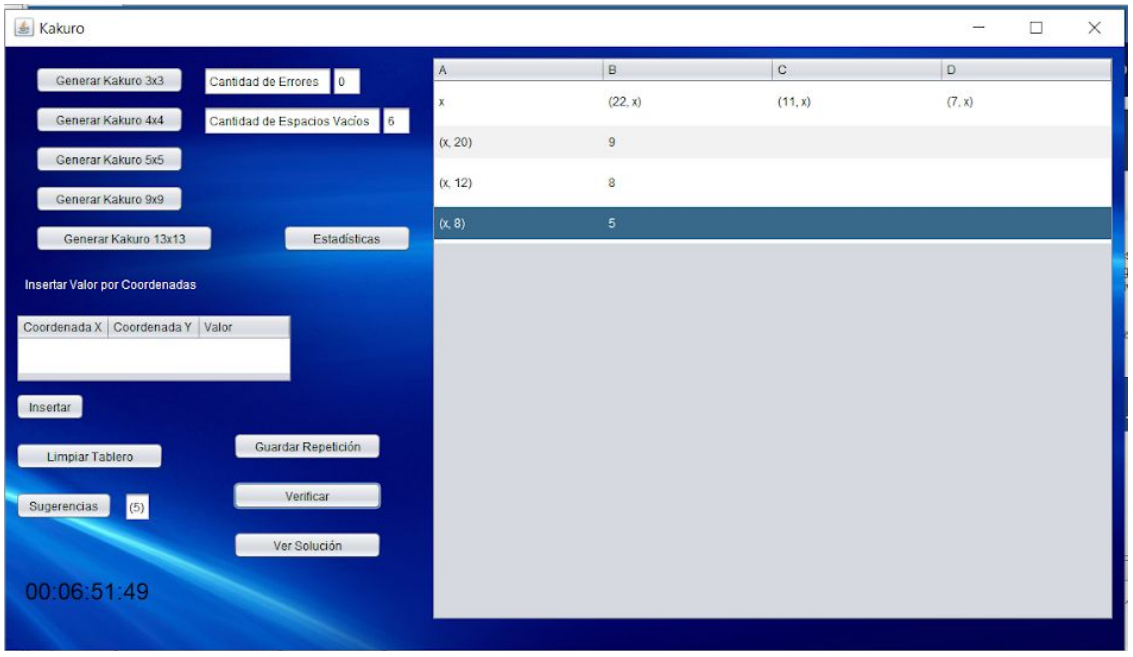
En este caso seleccioné el cuadro de 4x4, entonces se despliega uno que cumpla con esa condición, además de eso se inicia el cronómetro.

3. Se ingresan los datos que uno considere son la solución, para esto se debe seleccionar la casilla y luego seleccionar el número con click derecho, otra opción es insertar por coordenadas.



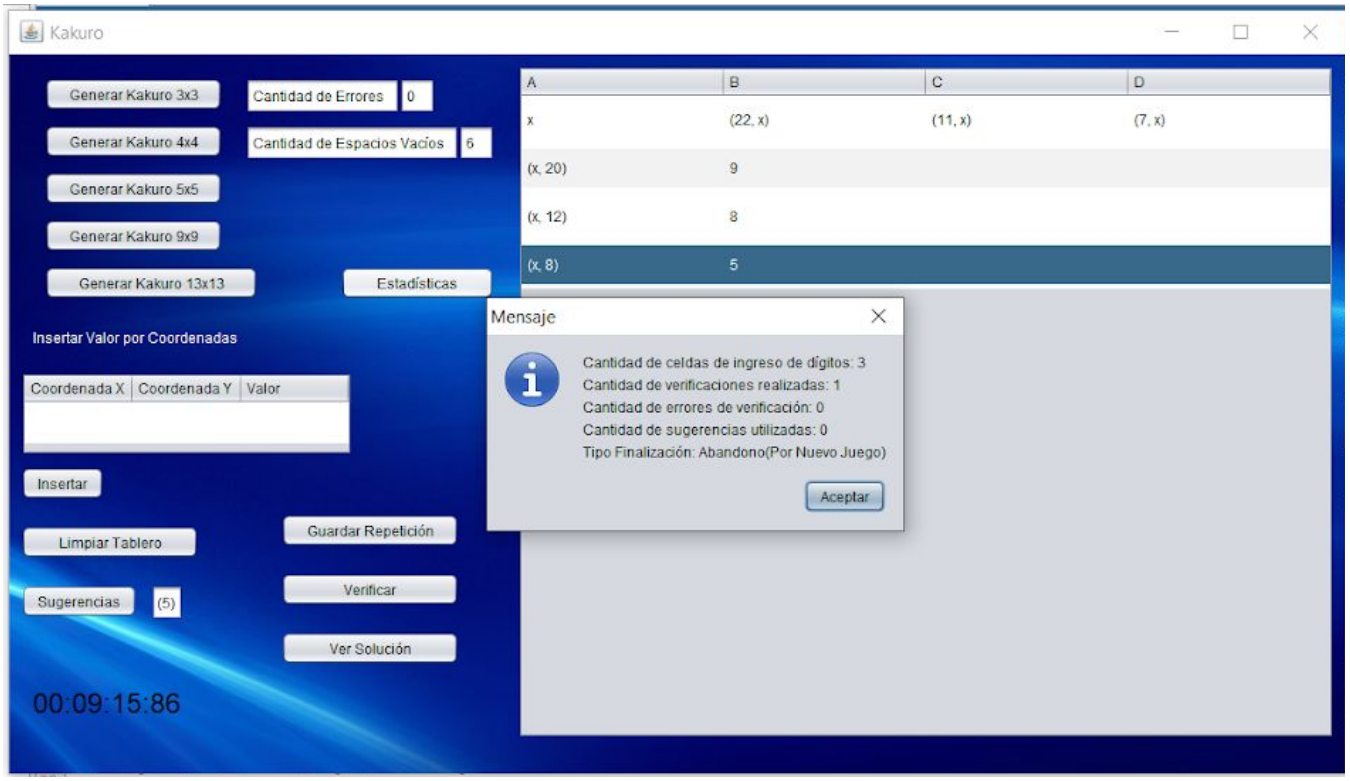
Ahora se tendrán distintas posibilidades, se pueden introducir más números, verificar los que ya tenemos, ver la solución y rendirnos, elegir las sugerencias, limpiar el tablero, guardar una repetición, mostrar las estadísticas del tablero actual.

4. En caso de verificar debemos tocar el botón de verificar para ver qué tan bien vamos en la partida, en este caso se verá así:



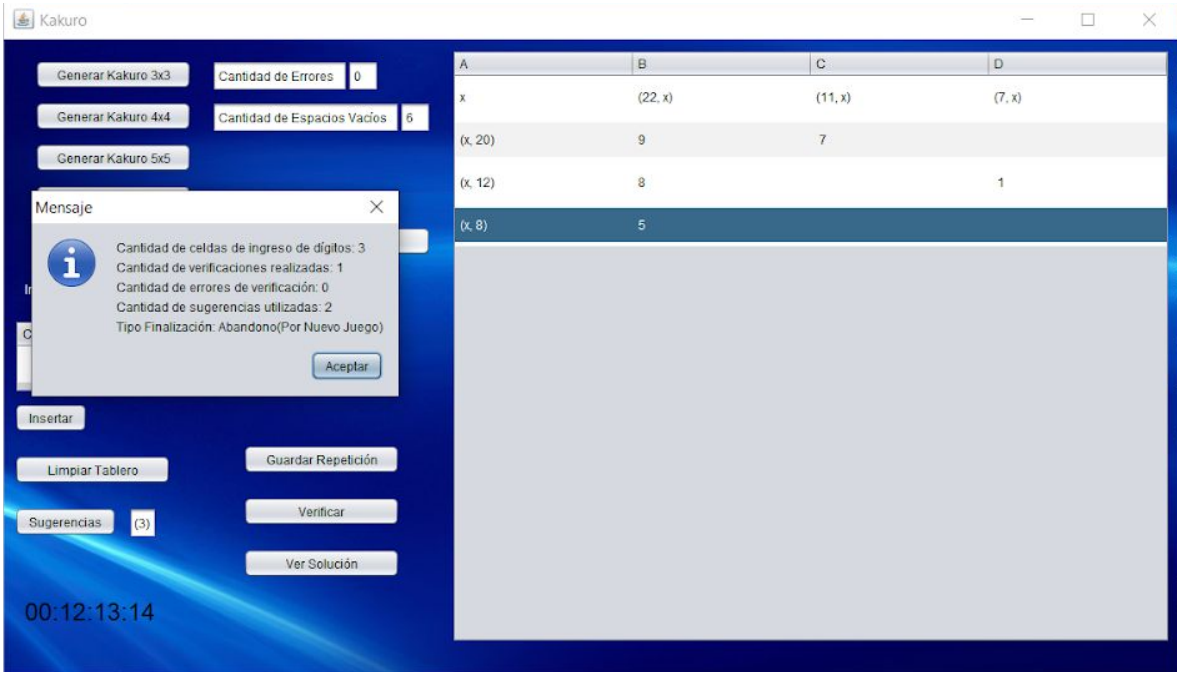
Donde dice cantidad de errores y cantidad de espacios vacíos se puede observar que todo lo que pusimos es correcto pero aún me faltan seis respuestas, tenga en cuenta que al llegar estos dos contadores a cero ganará la partida.

5. En el caso de mostrar las estadísticas se desplegará lo siguiente:



Se puede observar que ya ingresé 3 dígitos y que realicé una verificación, además no he tenido errores ni he utilizado sugerencias aún.

6. En el caso de pedir una sugerencia el sistema brinda en un espacio vacío una respuesta correcta y se le reduce la posibilidad de pedir sugerencias



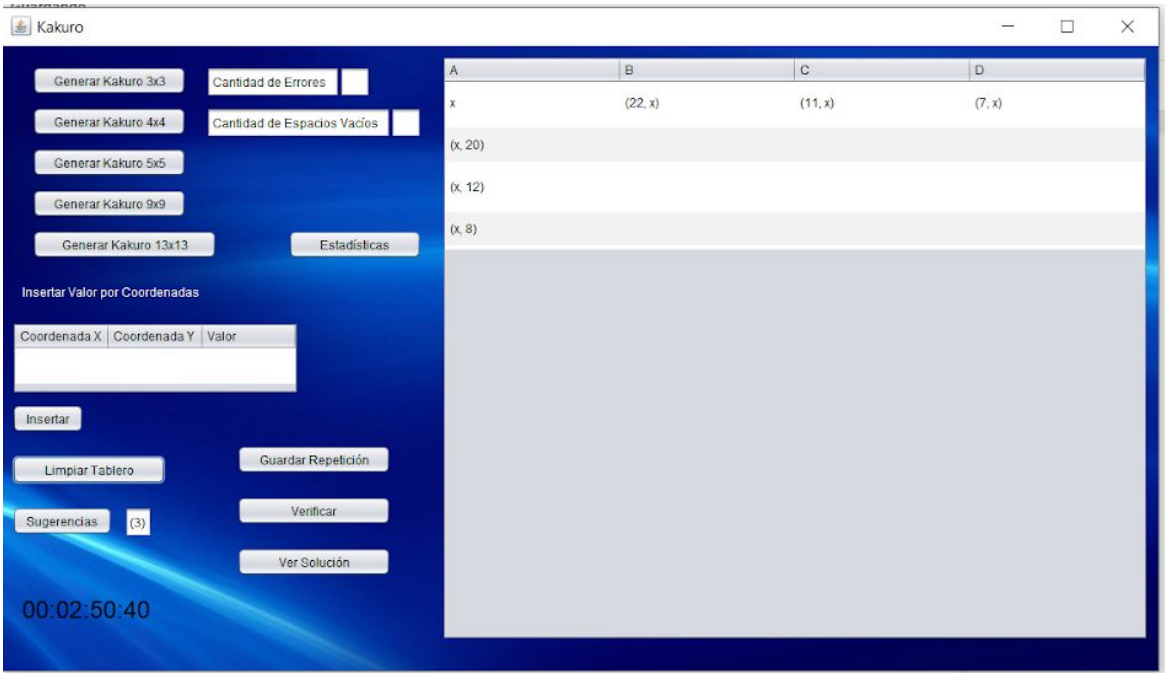
Como se observa se pidieron dos sugerencias y se redujo el número de sugerencias posibles, al este llegar a cero no nos será posible pedir más.

7. Al presionar guardar repetición se tomará una foto de lo que tenemos actualmente en nuestro juego y se verá así:

x	(22, x)	(11, x)	(7, x)
(x, 20)	9	7	
(x, 12)	8		1
(x, 8)	5		

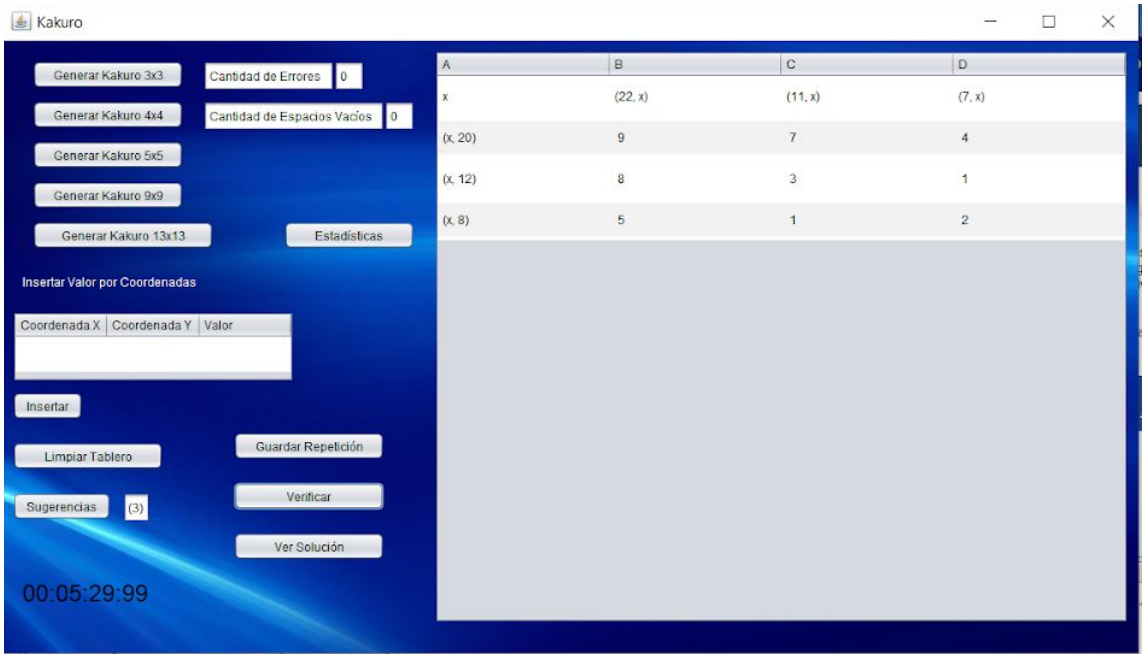
La imagen anterior se guarda en la carpeta de nuestro programa y se llama ultima_jugada, con esto podemos ver tableros que nos salieran previamente.

8. Al presionar limpiar tablero el tablero vuelve a la posición inicial, de esta manera puede volver a intentar ganar.



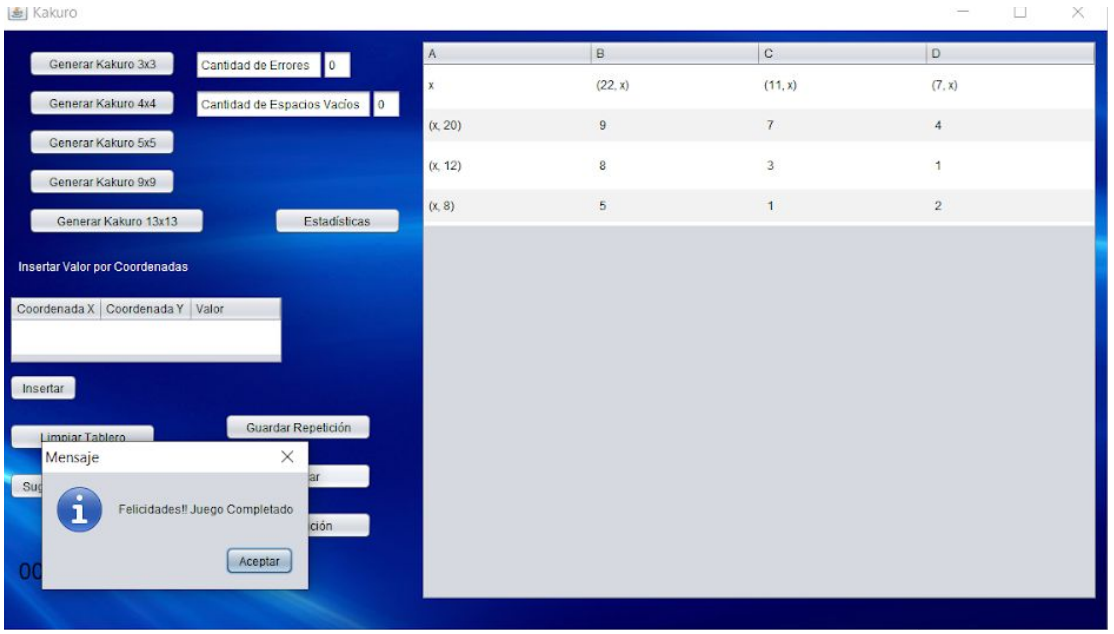
Se puede observar que es el mismo tablero pero con las casillas vacías.

9. Para rendirse se puede presionar ver solución, esto nos mostrará cómo debíamos llenar el tablero



De esta manera ya se ha terminado el juego.

10. Para terminar se puede presionar Verificar para ver si ya hemos ganado.



Se puede observar que ya el usuario terminó el juego correctamente, esto además detiene el cronómetro.

Pruebas de funcionalidad

Presentar tablero

Se muestra el tablero correctamente al presionar su botón respectivo

Generar nuevo juego

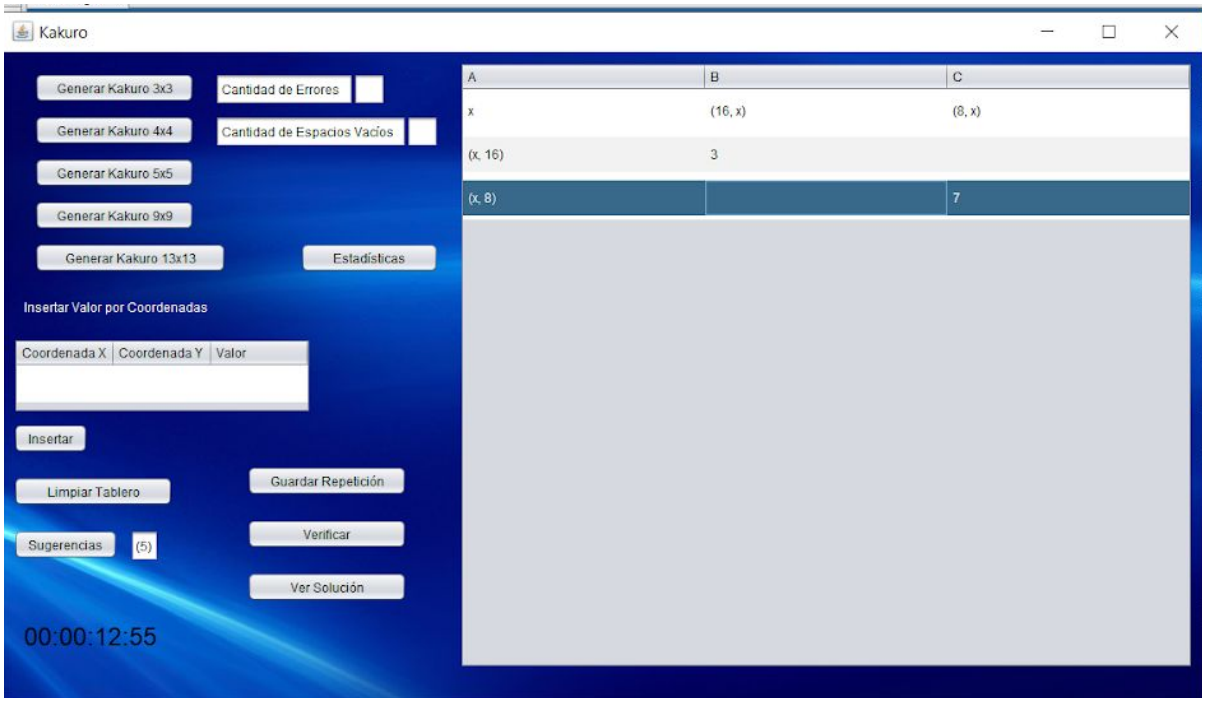
Se crea una parte de la matriz aleatoriamente cuando llamamos la regla en prolog

```
?- creaMatriz(3,3,Res).
Res = [[2, x, 9], [4, 5, 9], [x, x, 5]].
```

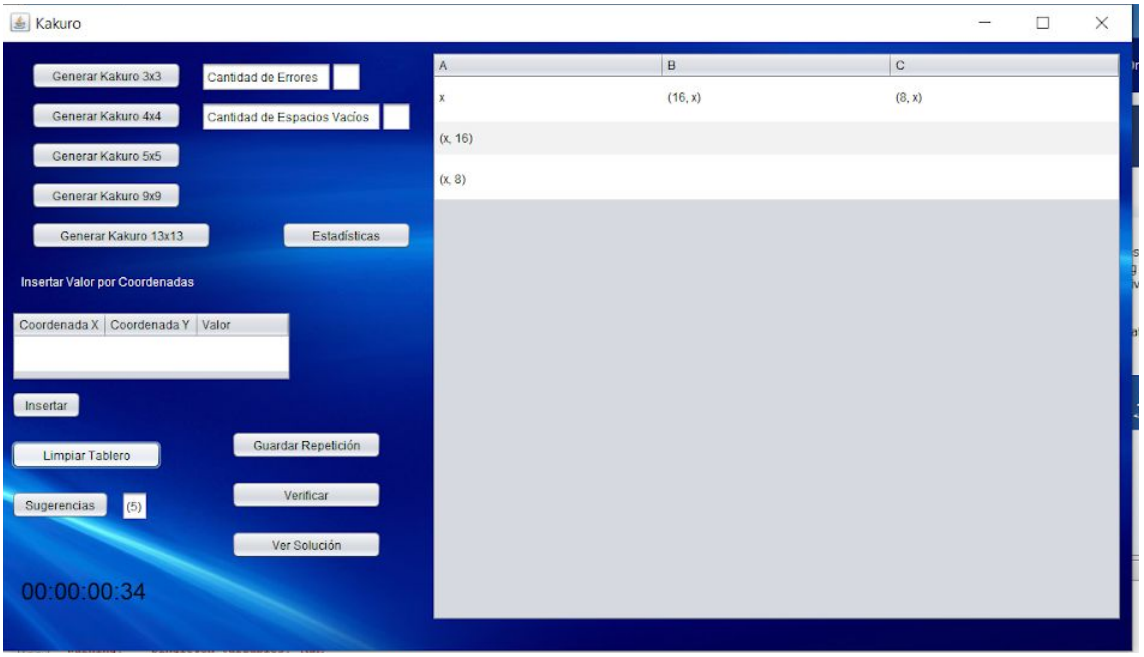
No es la matriz completamente terminada solamente una parte

Limpiar tablero

El tablero pasa de tener respuestas como se muestra a continuación a no tenerlas.

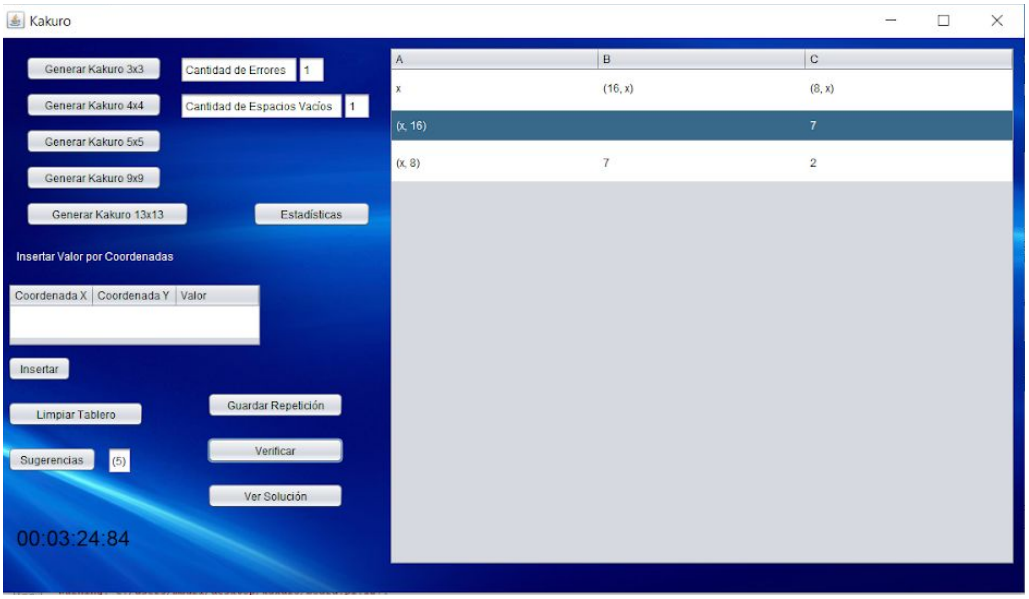


La siguiente imagen es después de presionar el botón de limpiar.

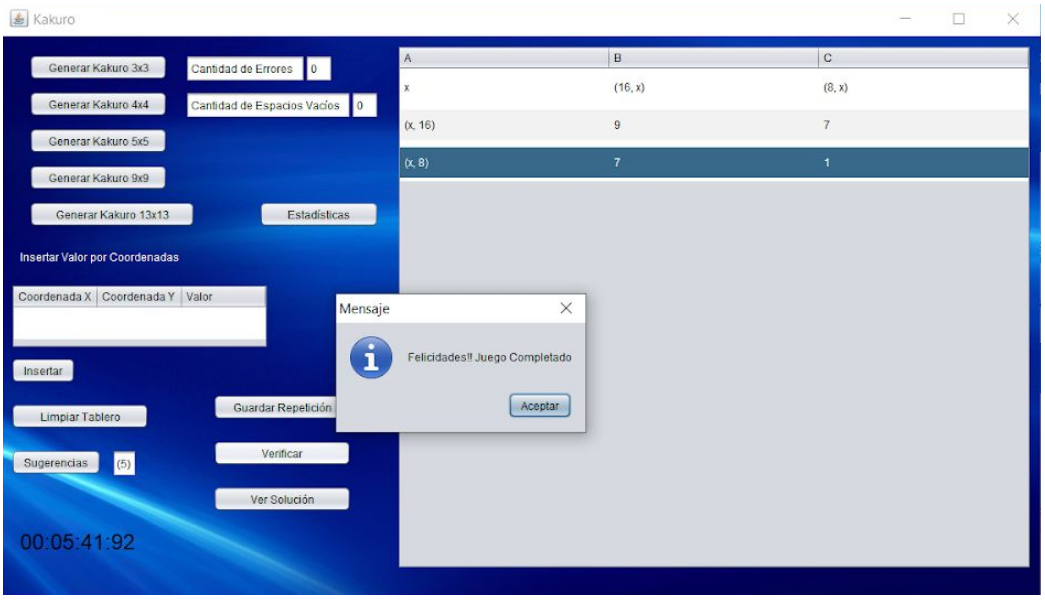


Verificar tablero

Se puede presionar verificar tablero para ver respuestas correctas e incorrectas y campos vacíos.



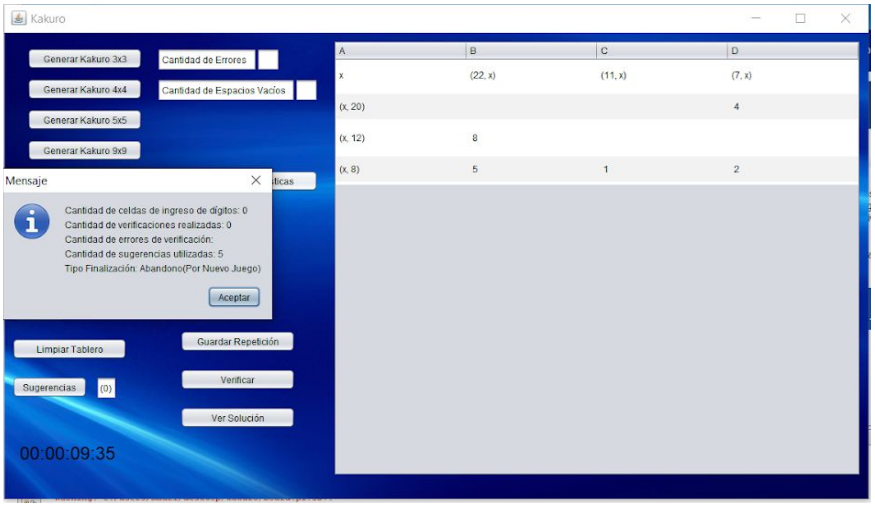
Se puede observar que tenemos dos errores y una casilla faltante, la casilla faltante es la que debería sumar 16 junto con 7 y el error sería el dos ya que su suma con 7 no es igual a 8, en caso de cambiar ambos dígitos por un 9 y un 1 respectivamente se ganará el juego de la siguiente manera:



Como se puede ver se detiene el cronómetro y el jeugo termina exitosamente.

Pistas

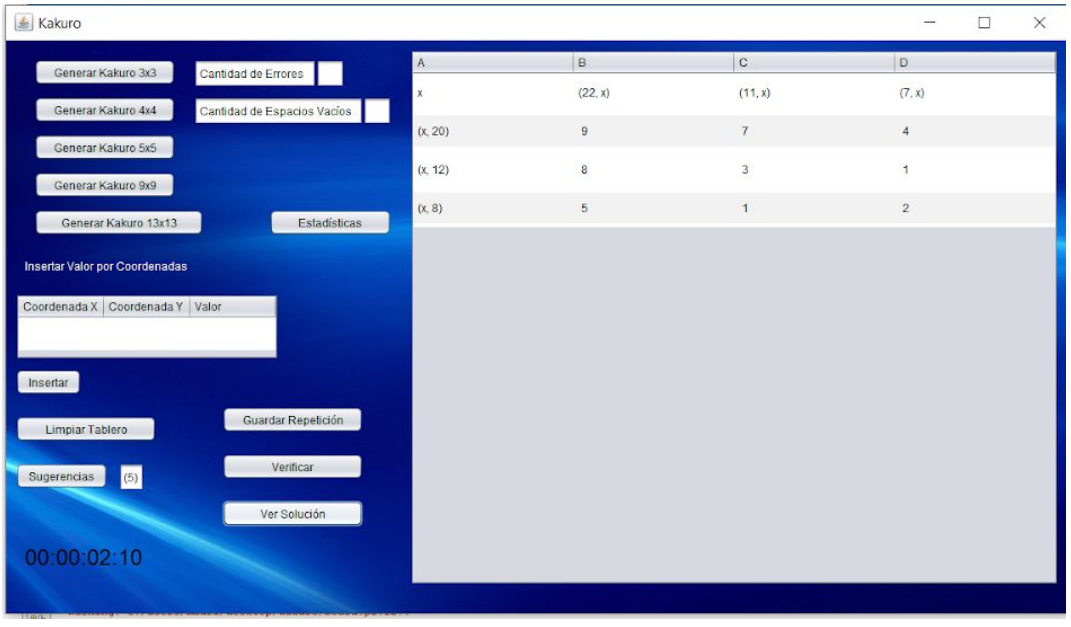
Para las pistas simplemente se presiona el botón, a continuación lo presionaré 5 veces en la tabla de 4x4



Se puede ver que ahora hay 0 sugerencias restantes y por medio de las estadísticas se ven las 5 utilizadas, es importante saber que las casillas donde se usa una sugerencia ya no serán editables hasta el fin del tablero.

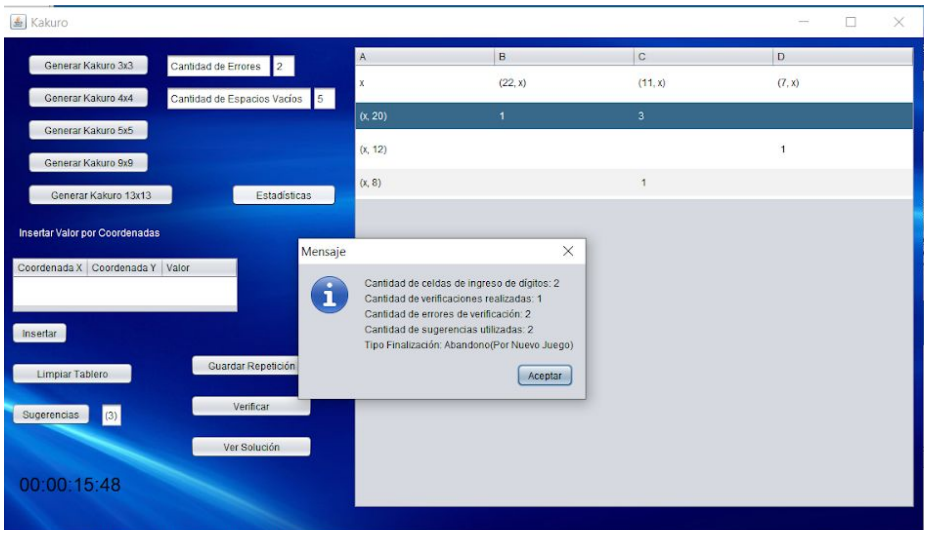
Ver solución

Si se presiona ver solución el usuario se rinde y se despliegan todos los números en sus casillas



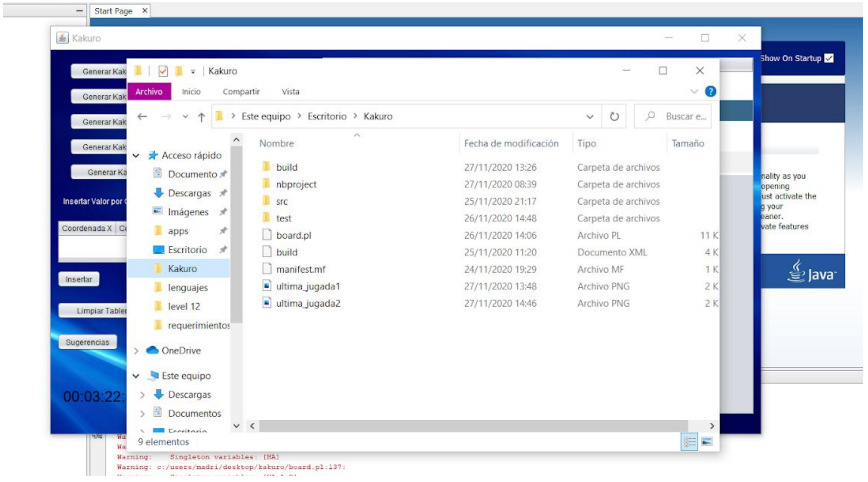
Ver estadísticas

Como anteriormente se mostró se puede utilizar para ver un resumen de cómo ha estado nuestra partida y se verá de la siguiente forma:



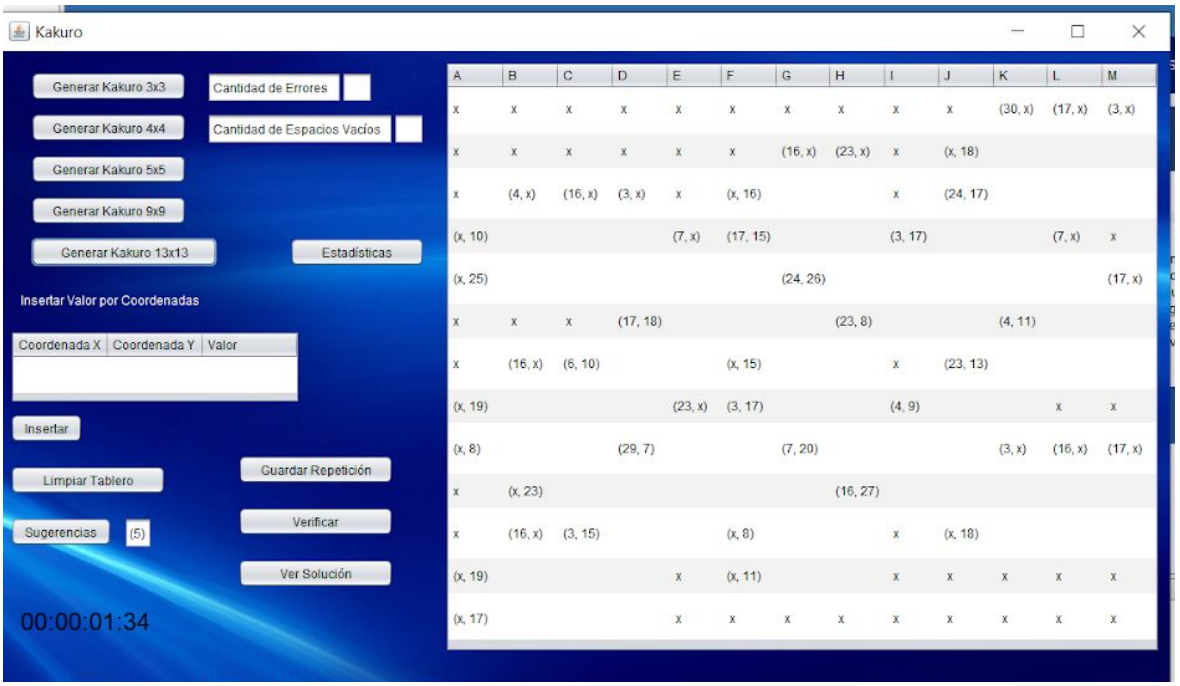
Almacenar repetición

Se almacena un pantallazo en la carpeta del programa, esto como se vio anteriormente sólo guarda el tablero.



Redimensionar tablero

Cada botón despliega un tamaño de tablero diferente por el ejemplo el 13x13 sería así:



Implementar cronómetro

El cronómetro se ha observado en cada pantallazo anterior, este se encuentra en la esquina inferior izquierda y se ve así:

Descripción del problema

Se debe implementar un kakuro utilizando prolog y java. Este proyecto requiere de un conocimiento detallado de las reglas del juego de kakuro. Antes de iniciar el trabajo debe comprender por completo el funcionamiento del juego, ya que lo que se desea es reproducir el comportamiento del mismo.

Se debe investigar a profundidad los kakuros que se encuentran disponibles en internet, Las funciones que debe tener este sistema son:

Presentar tablero, generar nuevo juego, limpiar tablero, verificar tablero, pistas, ver solución, ver estadísticas, almacenar repetición, redimensionar tablero, implementar cronómetro.

Diseño del programa

Decisiones de diseño

Decidimos utilizar una matriz para almacenar el tablero, de esta manera se nos simplificó el uso de head y tail en prolog para poder recorrer el tablero y realizar las operaciones necesarias, decidimos utilizar java swing para realizar la interfaz ya que se nos facilita más su uso que el resto de herramientas para realizar interfaces, con el uso de java swing podemos usar eventos que realicen las operaciones que queremos. Decidimos almacenar x en el tablero para representar los campos donde no se puede escribir, decidimos almacenar tuplas con las reglas a seguir y decidimos almacenar de una vez los números que corresponden a la solución del tablero de esta manera se facilita su verificación y en java sólo se oculta lo que no ocupemos.

Algoritmos usados

No fue necesario el uso de algoritmos, sin embargo para la verificación de la matriz utilizamos un código para recorrer que fue bastante útil, este dividía la matriz en listas y las listas en elementos por medio de su head y tail, luego se realizaban las comparaciones necesarias para contar la cantidad de valores que siguen sin llenar y aquellos que tienen una respuesta errónea.

Bibliotecas usadas

Se utilizó random para dar la aleatoriedad querida a la generación de un nuevo juego, esta se utilizó para sacar números aleatoriamente que sirvieron como indicadores, se utilizó awt para realizar los eventos de cada elemento de la interfaz, de esta manera al presionar un botón empieza un evento, además esta biblioteca nos brindó la opción de tomar screenshots entonces la utilizamos para almacenar la repetición, junto a esta se utilizó io para pasar las imágenes a formato png y controlar excepciones del programa, en conjunto con estas dos anteriores se utilizó imageio para la exportación de la imágen. Otra biblioteca utilizada fue util que fue principalmente utilizada para la elaboración de listas y el uso de randoms, por último se utilizó swing para el desarrollo de la interfaz.

Análisis de resultados

Objetivos alcanzados

Los objetivos que logramos alcanzar fueron: presentar el tablero en pantalla, una parte de generar nuevo juego, limpiar el tablero, verificar el tablero correctamente, brindar pistas, ver solución de tablero, ver estadísticas por tablero, almacenar repetición de jugada, redimensionar tablero a voluntad, implementar cronómetro correctamente.

Objetivos no alcanzados

No se logró alcanzar el objetivo de generar nuevo juego completamente, se pudo generar casillas no jugables aleatoriamente, y se pudieron generar los números de la respuesta aleatoriamente.

Razones de objetivos no alcanzados

Se complicó la lógica del resto del programa cuando intentamos implementar nuevo juego, tanto que complicó incluso la entrega efectiva del proyecto.

Bitácora

Autogenerada, se encuentra presente en el git.
Link para acceder al repositorio: https://github.com/tocapb/proyecto3_lenguajes.git