

TECNOLÓGICO DE COSTA RICA
Taller de Programación
Grupo 5

Título del trabajo:

PROYECTO:
JUEGO DE LÓGICA KAKURO

Estudiante:

Yared Moisés Guido Fallas
2025087987

Profesor:

William Mata Rodriguez

I Semestre 2025

Contenido

Objetivos.....	1
Definición	2
Requerimientos.....	3
A) Opción Jugar	3
B) Opción Configurar.....	9
C) Ayuda	9
D) Acerca de	9
E) Salir	9
PARTIDAS DE JUEGO	9
Marco Teórico.....	12
Control de Versiones y Colaboración	12
¿Qué es?.....	12
Importancia.....	12
Softwares de este tipo.....	12
Git.....	12
Subversion (SVN).....	13
Mercurial.....	13
Diferencia entre Git y GitHub	13
¿Qué es GitHub?.....	13
Diferencias entre Git y GitHub	13
Conclusiones	15
Problemas encontrados y soluciones a los mismos	15
Problema para hacer la cuadrícula	15
Estadística de Tiempos.....	16
Lista de revisión del proyecto	17

Objetivos

- Continuar aplicando el ciclo completo de la metodología general de desarrollo de programas a situaciones de mayor alcance: entender el problema, diseñar algoritmos, codificar algoritmos, probar y evaluar programas.

- Aplicar y reforzar aspectos del lenguaje

- Uso de diversos componentes del lenguaje
- Uso de programación imperativa y estructurada (estructuras: secuencia, condicional, repetición de procesos)
- Manejo de la técnica de iteración para repetición de procesos
- Manipulación de estructuras de datos nativas de Python
- Diseño y manipulación de TDA (Tipos de Datos Abstractos) como las pilas

- Aplicar buenas prácticas de programación básicas:

- Documentación interna y externa del programa
- Nombres significativos
- Usar un estilo de programación
- Desarrollo y reutilización de funciones
- Desarrollo y reutilización de módulos si fueran necesarios
- Uso de la técnica divide y vencerás (dividir el problema en partes)
- Validación de datos
- Manejo de errores y excepciones
- Pruebas
- Uso de software de control de versiones: Git

- Uso de archivos para mantener la persistencia de la información

- Continuar usando interfaces gráficas de usuario (GUI: Graphical User Interface) en Python con el módulo tkinter

- **Fomentar la habilidad de investigación:** los aspectos investigados deben ser explicados detalladamente en la documentación del proyecto. Para este proyecto los temas mínimos a investigar son:

- Software de control de versiones y colaboración: ¿A qué se refiere este software y su importancia en el desarrollo de software?
- Mencionar y explicar brevemente tres software de ese tipo
- Diferencia entre Git y GitHub.
- Explicar las características de Git que usó en este proyecto.
- Nuevos componentes de tkinter usados en este proyecto

Definición

Kakuro es un pasatiempo de lógica, tipo crucigrama matemático ya que en lugar de completarlo con palabras usando pistas verbales se completa con números usando pistas numéricas.

Consiste en rellenar con números del 1 al 9 las casillas vacías de una cuadrícula. La cuadrícula la podemos ver como una matriz de dos dimensiones. Aunque la cuadrícula puede tener diferentes tamaños, para efectos de este programa vamos a trabajar una cuadrícula de 9 x 9, es decir 9 filas y 9 columnas.

En una casilla, ya sea en su parte derecha o en su parte izquierda, puede aparecer un número al cual se le llama clave numérica. Si la clave está en la parte derecha de la casilla corresponde a una clave de fila, si aparece a la izquierda corresponde a una clave de columna. El objetivo es que la suma de los números seguidos en forma horizontal sea igual a su clave numérica de fila, y la suma de los números seguidos en forma vertical sea igual a su clave numérica de columna. No se puede repetir un mismo número para las casillas de la clave numérica, ya sea a nivel de fila o de columna.

Ejemplo de un tablero de Kakuro: la imagen a la izquierda muestra una partida inicial de juego y a la derecha la solución:

		13	12				7	10
	14			4	11	17	4	
	36							
12			10				25	14
3			20	11	20			
17					8	6		
	11	7	13			4	10	
28								
6				8				

		13	12				7	10
	14	5	9	4	11	17	4	1
	36	2	3	1	8	9	6	7
12	4	8	10	3	2	5	25	14
3	2	1	20	11	20	1	3	9
17	1	3	9	4	8	6	1	5
	11	7	13	7	1	5	4	10
28	7	5	4	6	3	1	2	
6	4	2			8	3	5	

Requerimientos

El programa tendrá un menú principal desde el cual se accederán las funcionalidades del programa. Usted puede agregar otras funcionalidades que vayan a mejorar el producto. La interfaz de las cuadrículas del tablero debe ser igual a la indicada en esta especificación. En los demás elementos de la ventana de juego puede hacer cambios a esta interfaz manteniendo la funcionalidad especificada.

El menú principal contendrá las siguientes opciones.

A) Opción Jugar

Esta opción permite jugar el Kakuro. Cuando se da esta opción se muestra una pantalla como la siguiente según la configuración del juego.

KAKURO

Jugador: (String de 1 a 40 caracteres máximo)

	25	44		20	10		39	16
10			10			8		
16			16			12		
		26				14		
29								
7				3				12
	17			12				
			18	6				
	42							
11								
17			5			11		
9			11			6		

1

2

3

4


5

6

7

8

9



INICIAR JUEGO

DESHACER JUGADA

BORRAR JUEGO

GUARDAR JUEGO

RÉCORDS

REHACER JUGADA

TERMINAR JUEGO

CARGAR JUEGO

Horas	Minutos	Segundos
00	00	00

NIVEL FÁCIL

Uso de los botones de la opción jugar:



Cuando el jugador hace clic en este botón inicia el juego. El jugador selecciona un número con un clic en el panel de números y luego clic en la casilla de la cuadrícula en donde quiere ponerlo y le sea permitido. Cuando selecciona un número, éste se señala de alguna forma que usted decide, por ejemplo, puede ponerse en otro color como se muestra con el número 3, y luego cuando el número es colocado en alguna casilla se le quita la seña, en este ejemplo se le quitaría el color. Cuando el jugador pone un número se deben hacer las validaciones para que la jugada cumpla con las reglas del juego, de lo contrario se le envía alguno de estos mensajes:

- JUGADA NO ES VÁLIDA PORQUE EL NÚMERO YA ESTÁ EN SU GRUPO DE FILA
- JUGADA NO ES VÁLIDA PORQUE EL NÚMERO YA ESTÁ EN SU GRUPO DE COLUMNA
- JUGADA NO ES VÁLIDA PORQUE LA SUMA DE LA FILA ES XX Y LA CLAVE NUMÉRICA ES YY
- JUGADA NO ES VÁLIDA PORQUE LA SUMA DE LA COLUMNA ES XX Y LA CLAVE NUMÉRICA ES YY

XX: se cambia por la suma de los números de la fila o de la columna.

YY: se cambia por la clave numérica respectiva.

El juego termina automáticamente cuando el jugador llena todas las casillas de la cuadrícula, ahí para el Cronómetro o el Temporizador y despliega el mensaje ¡EXCELENTE JUGADOR! TERMINÓ EL JUEGO CON ÉXITO.

En este momento el jugador se registra en los Récords. Los Récords es un archivo donde el programa registra las marcas que van haciendo los jugadores en cada nivel de dificultad. Cada marca tiene estos datos: el nombre del jugador y el tiempo (horas, minutos, segundos) que un jugador tardó en completar un juego. Note que si usa el Temporizador hay que calcular la duración del juego.

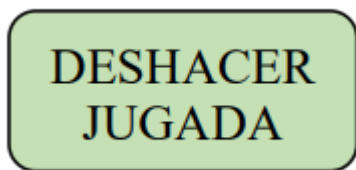
Cuando un juego es completado el programa borra la información del juego actual excepto el nombre y pone nuevamente otro juego de tal forma que el programa quede listo para que vuelva a hacer clic en este botón.

El ícono del borrador (debajo del panel de números) debe permitir eliminar el contenido de una casilla.

- Las casillas que tienen claves o las que no se pueden usar no se pueden modificar durante el juego.

- Antes de iniciar el juego el jugador debe dar un nombre.
- Luego de dar el botón INICIAR JUEGO, este botón se deshabilita.
- En caso de haber configurado la opción de Temporizador, el jugador puede dejar el tiempo configurado o modificarlo antes de INICIAR JUEGO. El tiempo empieza a correr cuando le den INICIAR JUEGO. Cuando se usa el Cronómetro o el Temporizador debe mantenerse actualizado en el monitor en tiempo real.
- En caso de no usar el Cronómetro o el Temporizador no debe aparecer el reloj en la pantalla.
- En caso de haber configurado la opción de Temporizador y éste llegue a 0 y el juego no haya terminado se envía el mensaje TIEMPO EXPIRADO. ¿DESEA CONTINUAR EL MISMO JUEGO (SI/NO)?. Si responde SI entonces el Temporizador pasa a ser cronómetro inicializado con el tiempo que se había establecido en el Temporizador. Por ejemplo, si el Temporizador estaba para 1 hora y 30 minutos, ahora el cronómetro debe marcar que ya ha pasado 1 hora y 30 minutos y sigue contando el tiempo. Si responde NO el juego finaliza regresando a la opción de Jugar.
- En caso de usar el Cronómetro las horas pueden llegar hasta 2, los minutos entre 0 y 59 y los segundos entre 0 y 59. Hay que realizar estas validaciones.
- En caso de no existir alguna partida para el nivel seleccionado se da el mensaje NO HAY PARTIDAS PARA ESTE NIVEL. Luego lo envía al menú principal.
- En caso de dar clic en una casilla y no haya seleccionado previamente un número se envía el error FALTA QUE SELECCIONE EL NÚMERO.

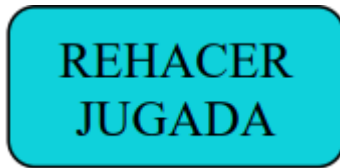
En caso de que ocurra un error el programa enviará un mensaje de error. El error se mantendrá en pantalla hasta que el usuario le indique al programa que continúe para que pueda corregir la situación.



Una jugada realizada es cuando se pone un número en una casilla o cuando se borra. Puede llevarse una pila de jugadas realizadas:

- Operación push: cuando se realiza una de esas jugadas.
- Operación pop: cuando el jugador pide deshacer una jugada se deshace la última en la pila. ¿Cuántas jugadas se pueden deshacer? Todas las que se encuentren en la pila de jugadas.

Un pop en esta pila también causar un push en la pila del siguiente botón.

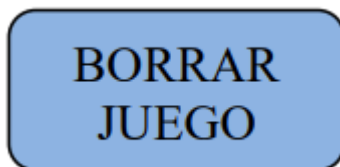


Este botón se puede usar solamente cuando se hayan deshecho jugadas. Rehace o reconstruye la última jugada que se deshizo. ¿Cuántas jugadas se pueden rehacer? Todas las que se hayan deshecho.

Puede llevarse una pila de jugadas deshechas:

- Operación push cuando se deshace una jugada.
- Operación pop cuando el jugador pide rehacer jugada.

Un pop en esta pila también causa un push en la pila del botón anterior.



Cuando el jugador selecciona esta opción se le pregunta

¿ESTÁ SEGURO DE BORRAR EL JUEGO (SI/NO)?

Si responde SI borra todas las jugadas que ha realizado y pone nuevamente el mismo juego de tal forma que el programa quede listo para que vuelva a hacer clic en el botón INICIAR JUEGO.

Si responde NO sigue jugando con el mismo juego.

Se puede seleccionar esta opción solamente si el juego ha iniciado de lo contrario hay que enviar el mensaje NO SE HA INICIADO EL JUEGO.



Cuando el jugador selecciona esta opción se le pregunta

¿ESTÁ SEGURO DE TERMINAR EL JUEGO (SI/NO)?

Si responde SI termina de inmediato el juego y se vuelve a mostrar otro juego como si estuviera entrando a la opción de Jugar.

Si responde NO sigue jugando con el mismo juego.

Se puede seleccionar esta opción solamente si el juego ha iniciado de lo contrario hay que enviar el mensaje NO SE HA INICIADO EL JUEGO.



Esta opción se puede usar en cualquier momento. Detiene el reloj si se está usando. Los récords se guardan en el archivo "kakuro2025_récords".

¿Qué información se despliega? Se da al jugador estas opciones:

Récords o Todos los niveles

 o Nivel fácil

 o Nivel medio

 o Nivel difícil

 o Nivel experto

Jugadores o Todos los jugadores

 o Yo

Por omisión se muestran los récords de todos los jugadores del nivel que se está jugando, pero el jugador puede escoger la información según esas opciones. Si el jugador escoge la opción Yo, le lleva directamente a su información mostrando el lugar que ocupa en ese récord. En cada nivel se despliega la información ascendentemente por tiempo.

NIVEL FÁCIL	JUGADOR	TIEMPO
	1- Nombre jugador	1:30:15
	2- Nombre jugador	1:32:55

NIVEL MEDIO

1- Nombre jugador	1:30:15
-------------------	---------

2- Nombre jugador	1:32:55
-------------------	---------

NIVEL DIFÍCIL

1- Nombre jugador	1:30:15
-------------------	---------

2- Nombre jugador	1:32:55
-------------------	---------

NIVEL EXPERTO

1- Nombre jugador	1:30:15
-------------------	---------

2- Nombre jugador	1:32:55
-------------------	---------

Luego de que el usuario vea esta información el programa regresa a donde estaba jugando y sigue el conteo en el reloj en caso de que proceda.



Este botón se puede usar en cualquier momento que el juego haya iniciado. El reloj se detiene. Guarda en el archivo "kakuro2025_juego_actual" toda la información del juego actual de tal forma que posteriormente el juego pueda reiniciarse en donde quedó. Este archivo solo va a contener una partida por cada jugador, en caso de que haya una partida guardada en el archivo para ese jugador, se borra y se guarda la del momento. Luego de guardarlo se le pregunta al usuario:

¿VA A CONTINUAR JUGANDO (SI/NO)?

Si responde SI sigue jugando el mismo juego.

Si responde NO regresa al menú principal.



Este botón se puede usar solamente cuando un juego no se haya iniciado. Trae del archivo "kakuro2025_juego_actual" el juego que tenga registrado con la configuración y lo pone en la pantalla como el juego actual. El juego continúa cuando el jugador usa el botón de INICIAR JUEGO.

B) Opción Configurar

Esta opción es para indicar las condiciones con que se va a jugar. Contiene los siguientes datos que se van a guardar en el archivo “kakuro2025_configuración”: (los valores por omisión –o default- están señalados con el círculo en rojo)

1. Nivel: ☒ Fácil
 ☐ Medio
 ☐ Difícil
 ☐ Experto

2. Reloj: ☒ Cronómetro
 ☐ Temporizador
 ☐ No usar reloj

Horas	Minutos	Segundos
0	15	0

En caso de seleccionar Temporizador opcionalmente puede dar el tiempo o dejarlo en blanco para que antes de iniciar el juego hagan el registro del tiempo a usar. Para el temporizador las horas pueden estar entre 0 y 2, los minutos entre 0 y 59 y los segundos entre 0 y 59. El Temporizador debe tener al menos uno de estos valores ya sea aquí o en la opción de Jugar. Hay que realizar estas validaciones y enviar los mensajes respectivos en caso de errores.

C) Ayuda

Esta opción la usaremos para que el usuario pueda ver el Manual de Usuario (explicado más adelante) directamente en la computadora. Cuando la opción es seleccionada el manual se despliega de inmediato.

D) Acerca de

Esta opción la usaremos para desplegar información “Acerca del programa” donde colocaremos al menos los datos del nombre del programa, la versión, la fecha de creación y el autor.

E) Salir

Esta opción se usa para salir de este programa (también se puede salir con el botón de cerrar “X” en la interfaz gráfica).

PARTIDAS DE JUEGO

Están guardadas en un archivo JSON de nombre “kakuro2025_partidas.json”. Contendrá la información para completar una partida que permite iniciar el juego.

El programa toma las partidas registradas y debe seleccionar aleatoriamente una partida para desplegarla en el área de juego según el nivel de dificultad configurado. Python tiene funciones para generar números aleatorios que pueden servir para seleccionar alguna de las partidas de tal forma que siempre se elija una al azar. Puede usar otros algoritmos para esta selección aleatoria de partidas. Documente cuál algoritmo de selección aleatoria usó. En una misma corrida del programa, si hay n partidas para un nivel, primero se deben escoger las n partidas aleatorias de ese nivel antes de volver a repetir las.

Estructura del archivo json: lista ([]) donde cada elemento es un diccionario con la información de una partida para iniciar el juego. Cada partida tendrá tres elementos:

- “nivel_de_dificultad”: FÁCIL, MEDIO, DIFÍCIL, EXPERTO
- “partida”: número de partida dentro del nivel de dificultad iniciando en 1
- “claves”: es una lista de diccionarios, donde cada diccionario tiene:

“tipo_de_clave”: “F” para fila, “C” para columna

“fila”: número de fila donde va la clave

“columna”: número de columna donde va la clave

“clave”: valor de la clave

“casillas”: cantidad de casillas a partir de la fila y columna para completar la clave

Las casillas con los valores “clave” y “casillas” en 0 no se pueden usar. Ejemplo con algunas casillas para la partida de ejemplo en la opción Jugar:

```
[ ← inicio del archivo
  {
    "nivel_de_dificultad": "FÁCIL",          ← inicio de una partida
    "partida": 1,
    "claves": [ ← inicio de la información de una partida
      {
        "tipo_de_clave": "F",
        "fila": 2,
        "columna": 1,
        "clave": 10,
        "casillas": 2
      },
      {
        "tipo_de_clave": "F",
        "fila": 5,
        "columna": 1,
        "clave": 7,
        "casillas": 3
      },
    ]
  }
]
```

```

        {
            "tipo_de_clave" : "C",
            "fila": 1,
            "columna": 2,
            "clave": 25,
            "casillas" : 4
        },
        {
            "tipo_de_clave" : "C",
            "fila": 1,
            "columna": 3,
            "clave": 44,
            "casillas" : 8
        },
        {
            "tipo_de_clave" : "F",
            "fila": 6,
            "columna": 1,
            "clave": 0,
            "casillas" : 0
        },
        ...
    ] ← fin de la información de una partida

}, ← fin de una partida
{
    "nivel_de_dificultad" : "FÁCIL",
    "partida": 2,
    "claves": [
        ...
    ]
},
...
] ← fin del archivo

```

Para pruebas llene este archivo con al menos 3 partidas por cada nivel.

Marco Teórico

Control de Versiones y Colaboración

¿Qué es?

El control de versiones es una práctica fundamental en el desarrollo de software que permite registrar y gestionar los cambios realizados en el código fuente de un proyecto a lo largo del tiempo. Este sistema permite llevar un historial detallado de todas las modificaciones, facilitando así la identificación de errores, la comparación entre versiones anteriores y actuales, y la posibilidad de restaurar versiones previas si es necesario.

Además, el control de versiones favorece la colaboración entre múltiples desarrolladores, ya que permite trabajar en paralelo sin sobrescribir el trabajo de los demás. Cada desarrollador puede trabajar en su propia copia del proyecto, y luego integrar sus cambios de forma segura y organizada. Esto hace que los proyectos sean más escalables, organizados y menos propensos a errores causados por conflictos de código.

Importancia

El control de versiones es vital para proyectos de software, ya que proporciona seguridad, trazabilidad y control sobre el desarrollo. Permite saber quién hizo qué cambio, cuándo y por qué, lo que es crucial tanto para resolver errores como para documentar el progreso del equipo. También facilita el trabajo con ramas, lo que permite desarrollar nuevas funciones sin afectar la versión principal del software.

En proyectos grandes o colaborativos, la ausencia de un sistema de control de versiones puede causar pérdidas de información, conflictos en el código y desorganización general. Por eso, esta herramienta no solo mejora la eficiencia del desarrollo, sino que también promueve una cultura de trabajo colaborativo, seguro y profesional.

Softwares de este tipo

Git

Git es un sistema de control de versiones distribuido, creado por Linus Torvalds. A diferencia de los sistemas centralizados, Git permite que cada desarrollador tenga una copia completa del historial del proyecto, lo que ofrece mayor independencia y seguridad. Es muy eficiente para manejar grandes volúmenes de cambios y es ampliamente utilizado en el desarrollo ágil y proyectos de código abierto.

Una de sus características más potentes es la gestión de ramas (branches), que permite experimentar o desarrollar nuevas funcionalidades sin afectar el código

principal. También es altamente personalizable y rápido, lo que lo convierte en el estándar de facto en el desarrollo de software moderno.

Subversion (SVN)

Subversion, también conocido como SVN, es un sistema de control de versiones centralizado, desarrollado por Apache. A diferencia de Git, en SVN todos los usuarios trabajan conectados a un servidor central, lo que permite un control más directo y centralizado de los cambios. Esto puede ser beneficioso en entornos corporativos donde se requiere una supervisión más estricta del código.

SVN es sencillo de usar y ofrece una buena trazabilidad de los cambios. Aunque ha sido superado en popularidad por Git, todavía es utilizado en muchos proyectos institucionales o en empresas que prefieren la simplicidad y el control centralizado sobre las ventajas del modelo distribuido.

Mercurial

Mercurial es otro sistema de control de versiones distribuido, al igual que Git, pero diseñado con un enfoque en la simplicidad y facilidad de uso. Es muy rápido, confiable y especialmente útil en proyectos grandes con muchos archivos y colaboradores. Su interfaz más amigable y comandos más predecibles lo convierten en una buena opción para desarrolladores que buscan eficiencia sin tanta complejidad.

Aunque no es tan popular como Git, Mercurial ha sido utilizado por grandes organizaciones como Mozilla. Su modelo distribuido permite a los usuarios trabajar sin conexión y sincronizar cambios después, al igual que Git, pero con una curva de aprendizaje más suave.

Diferencia entre Git y GitHub

¿Qué es GitHub?

GitHub es una plataforma en línea que permite almacenar y gestionar repositorios de código usando Git como base. Es decir, GitHub no es un sistema de control de versiones, sino una plataforma que facilita el uso de Git a través de una interfaz web amigable. Permite compartir proyectos, colaborar en equipo, gestionar solicitudes de cambio (pull requests), revisar código, reportar errores y documentar proyectos.

GitHub ofrece además herramientas adicionales como integración con CI/CD (integración y despliegue continuo), wikis, gestión de proyectos y perfiles públicos, lo que la convierte en una comunidad activa donde miles de desarrolladores colaboran y publican código abierto o privado.

Diferencias entre Git y GitHub

La principal diferencia es que Git es la herramienta que gestiona versiones localmente en tu computadora, mientras que GitHub es un servicio en la nube que permite alojar repositorios Git y colaborar con otros. Git puede usarse sin GitHub, pero GitHub no funciona sin Git. Git es el motor que gestiona los cambios en el código; GitHub es el entorno donde esos cambios se comparten y organizan.

Otra diferencia clave es que Git es una tecnología de línea de comandos (aunque existen interfaces gráficas), mientras que GitHub ofrece una interfaz web visual para gestionar proyectos, visualizar ramas, colaborar y revisar código. En resumen: Git es la herramienta; GitHub es el espacio donde esa herramienta cobra vida en comunidad.

Conclusiones

Problemas encontrados y soluciones a los mismos

Problema para hacer la cuadrícula

Inicialmente, tuve problemas para construir la cuadrícula del juego de Kakuro, ya que no estaba seguro de cuál widget de Tkinter era el más adecuado para esta tarea. Probé con varias opciones, pero ninguna me brindaba el control y funcionalidad que necesitaba para que el usuario pudiera interactuar fácilmente con cada celda. Finalmente, opté por usar botones, ya que estos permiten una interacción directa con eventos como clics, además de facilitar la ejecución de funciones asociadas a cada celda. Esta elección también me dio un control total sobre el contenido textual que aparece en cada botón, lo cual es esencial para mostrar los números del juego o dejarlos vacíos según el caso. Aunque no era la solución más convencional para una cuadrícula, resultó ser efectiva y flexible para este tipo de proyecto.

Estadística de Tiempos

Actividad Realizada	Horas
Análisis del problema	2
Diseño de algoritmos	4
Investigación del funcionamiento de Tkinter	6
Investigación del funcionamiento de Git	2
Investigación de temas para Marco Teórico	5
Programación	43
Documentación interna	3
Pruebas	3
Elaboración del manual de usuario	2
Elaboración de documentación del proyecto	5
TOTAL	75

Lista de revisión del proyecto

Concepto	Puntos originales	Avance 100%/0	Puntos obtenidos	Análisis de resultados
Opción Jugar: despliegue ventana de juego	15	100%	15	
Seleccionar partida	5	100%	5	
Botón Iniciar Juego	15	100%	15	
Creación del archivo de Récor ds	5	100%	5	
Borrar casillas	2	100%	2	
Botón Deshacer Jugada	8	100%	8	
Botón Rehacer Jugada	8	100%	8	
Botón Borrar Juego	2	100%	2	
Botón Terminar Juego	2	100%	5	
Botón Récor ds	8	100%	8	
Botón Guardar Juego	5	100%	5	
Botón Cargar Juego (incluye el despliegue del mismo)	10	100%	10	
Opción Configurar	5	100%	5	
Ayuda en el programa: Manual de usuario	5	100	5	
Cronómetro o Temporizador en tiempo real	5	100%	5	
TOTAL	100			
Partes desarrolladas adicionalmente				