



Instituto Tecnológico de Costa Rica

Campus Tecnológico Central Cartago

Escuela de Computación

Taller de Programación (IC-1803)

Programa 3

Estudiante:

Jose Pablo Hidalgo Navarro / 2020178017

Profesor:

William Mata Rodríguez

Grupo 3

Fecha de entrega: 29/06/2021

Primer Semestre / Semana 19

2021

Índice (Tabla de Contenidos)

Definición del proyecto	3
Requerimientos del programa	6
Jugar	6
Botón Iniciar Juego	7
Botón Borrar Jugada	8
Botón Terminar Juego	8
Botón Borrar Juego	8
Botón Top 10	10
Botón Guardar Juego	11
Botón Cargar Juego	11
Configurar	11
Ayuda	11
Acerca de	13
Salir	13
Partidas del juego	14
Temas investigados	15
Librerías (usos nuevos)	17
Time (función .sleep())	17
MessageBox (más tipos de mensajes)	18
Nuevas funciones tkinter	19
Función .update()	19
Bloquear Entrys	20
Software controlador de versiones	21
Utilidad	21
Git Hub	22
Conclusión	23
Problemas encontrados en el desarrollo y soluciones de los mismos	23
Aprendizajes obtenidos	24
Diseño y explicación de la solución	25
Estadísticas de tiempo	26
Lista de revisión del proyecto	27
Bibliografía	28

DEFINICIÓN DEL PROYECTO: FUTOSHIKI

Futoshiki es un pasatiempo de lógica originario de Japón, desarrollado por Tamaki Seto en 2001. Su nombre significa “desigualdad”. La información de este juego se tomó de wikipedia (<https://en.wikipedia.org/wiki/Futoshiki>). Puede jugarlo en línea en <https://es.goobix.com/juegos-en-linea/futoshiki/>.

En este pasatiempo hay que llenar con dígitos las casillas de una cuadrícula de tal forma que cada dígito no se repita ni en la fila ni en la columna a que pertenece y los dígitos cumplan con restricciones de desigualdad: mayor que ($>$) o menor que ($<$). Las restricciones de desigualdad pueden estar a nivel de filas (entre dos casillas horizontales) o a nivel de columnas (entre dos casillas verticales). La cuadrícula debe tener la misma cantidad de filas y de columnas.

En caso de que la cuadrícula fuera de tamaño 5 x 5, los dígitos a colocar estarían entre 1 y el tamaño de la cuadrícula, en este caso 5. Cada juego tiene de forma predeterminada las restricciones de desigualdad que el jugador debe cumplir y algunos dígitos fijos como se muestra en la figura 1:

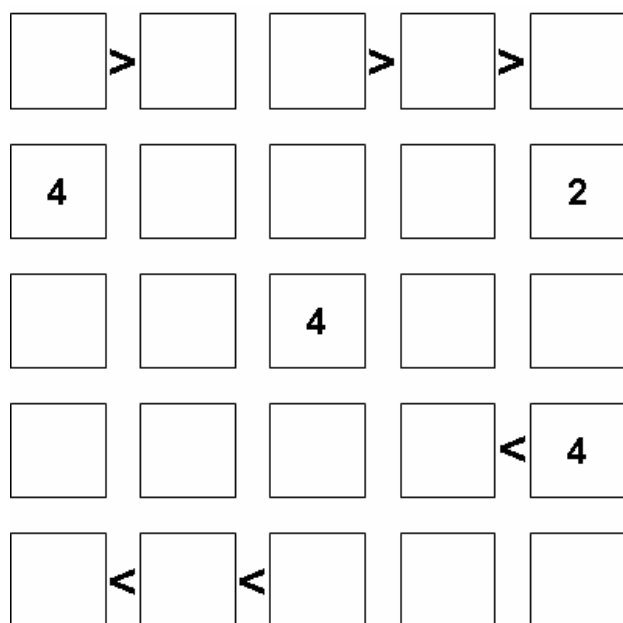


Figura 1: ejemplo de un Futoshiki 5 x 5 (By Gandalf61 at the English Wikipedia / CC BY-SA, <http://creativecommons.org/licenses/by-sa/3.0/>)

Programa 3 (30%): Juego Futoshiki

La solución al juego anterior sería:

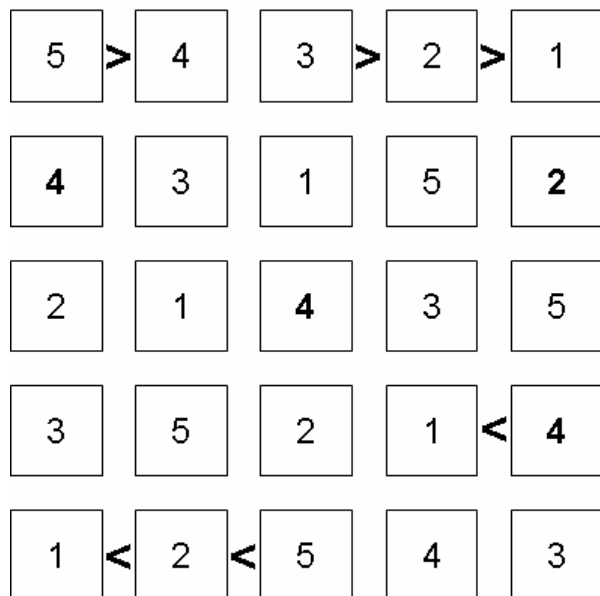
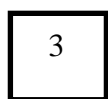
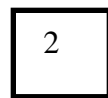


Figura 2: Futoshiki terminado (By Gandalf61 at the English Wikipedia / CC BY-SA, <http://creativecommons.org/licenses/by-sa/3.0/>).

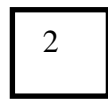
El juego puede tener restricciones a nivel de columna, por ejemplo:



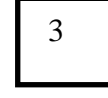
v



Se lee: $3 > 2$



Λ



Se lee: $2 < 3$

El programa usará una GUI (Graphical User Interface) que inicia con un menú principal desde el cual se accederá la funcionalidad del programa, es decir, lo que el programa hace. Usted puede agregar otras funcionalidades que vayan a mejorar el producto. Puede hacer cambios a la interfaz previamente acordados con el profesor.

Objetivos del proyecto de programación:

- Aplicar el ciclo completo de la metodología general de desarrollo de programas a situaciones de mayor alcance.
 - Aplicar y reforzar conceptos de programación y del lenguaje Python 3.
 - o Uso de diversos componentes del lenguaje.
 - o Desarrollo y reutilización de funciones.
 - o Uso de estructuras condicionales y de repetición de procesos. Opcionalmente puede usar recursión en las partes que considere apropiadas.
 - o Diseño y uso de estructuras de datos nativas de Python.
 - o Diseño y uso de TDA (Tipos de Datos Abstractos).
 - o Uso de archivos.
 - Aplicar buenas prácticas de programación: documentación interna y externa del programa, reutilización de código, nombres significativos, eficiencia del programa, evaluar alternativas, uso de técnica divide y vencerás (dividir el problema en partes, desarrollar cada una de esas partes), etc.
 - Validación de los datos de entrada: todos los datos de entrada se deben validar según restricciones que se indican en cada uno de ellos.
 - Fomentar la investigación en el estudiante: aquellos temas no tratados en el curso pero que los necesita para hacer el proyecto. Dichos temas deben ser explicados detalladamente en la documentación del proyecto. Entre los tópicos a investigar para este proyecto específico están:
 - o Nuevas características de tkinter exploradas para este proyecto.
 - o Software de control de versiones: importancia en ingeniería del software.
 - o Software de control de versiones usado: Git u otro, explicar las funciones usadas.
 - o Cualquier otro aspecto necesario para ofrecer su solución.
- Usar algún software de control de versiones de software, por ejemplo, Git u otro que usted decida. EN LA SEMANA SE DARÁ UN TALLER DEL SOFTWARE GIT HUB.**

REQUERIMIENTOS DEL PROGRAMA

A. Jugar

Esta opción permite jugar el Futoshiki con un tamaño de cuadrícula 5 x 5. Cuando se da esta opción se muestra una pantalla como la siguiente considerando la configuración (nivel del juego, uso del reloj, posición del panel de dígitos).

FUTOSHIKI

NIVEL FÁCIL

Nombre del jugador: _____

	>		>		>		1
4					2		2
		4					3
				<	4		4
	<		<				5

INICIAR JUEGO

BORRAR JUGADA

TERMINAR JUEGO

BORRAR JUEGO

TOP 10

Horas	Minutos	Segundos
00	00	00

GUARDAR JUEGO

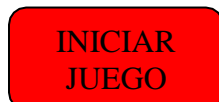
CARGAR JUEGO

El programa tiene una serie de partidas que previamente han sido registradas y de ahí selecciona aleatoriamente una partida según el nivel de dificultad configurado. Python tiene

Programa 3 (30%): Juego Futoshiki

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. En el diseño de la solución documente el algoritmo de selección aleatoria usado aquí.

Uso de los botones:



Cuando el jugador pica este botón se inicia el juego. Una vez que se inicia el juego, el jugador selecciona un dígito picándolo en el panel de dígitos (círculos a la derecha de la partida) y luego pica en la casilla de la cuadrícula en donde quiere ponerlo. Cuando pica el dígito se marca su círculo con un color verde como se muestra en el ejemplo. El dígito permanece marcado, disponible para ponerlo en otra casilla. Si quiere seleccionar otro dígito lo pica y el dígito anterior se desmarca. Cuando el jugador pone un dígito 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 ELEMENTO YA ESTÁ EN LA FILA
- JUGADA NO ES VÁLIDA PORQUE EL ELEMENTO YA ESTÁ EN LA COLUMNA
- JUGADA NO ES VÁLIDA PORQUE NO CUMPLE CON LA RESTRICCIÓN DE MAYOR
- JUGADA NO ES VÁLIDA PORQUE NO CUMPLE CON LA RESTRICCIÓN DE MENOR
- JUGADA NO ES VÁLIDA PORQUE ESTE ES UN DÍGITO FIJO

En estos casos ponga en color rojo la casilla que no está permitiendo la jugada para que el jugador la identifique de inmediato. El mensaje es enviado y el programa se detiene para que el jugador vea ese mensaje, cuando quiera seguir jugando debe dar la tecla <ENTER> u otra tecla y se quita el color rojo de la casilla con error.

El juego termina cuando el jugador llena todas las casillas de la cuadrícula de forma correcta. En este caso el reloj o el timer se detienen automáticamente y se despliega un mensaje de felicitación, por ejemplo: ¡EXCELENTE! JUEGO TERMINADO CON ÉXITO.

En este momento debe determinar si este jugador debe registrarlo en el Top 10. El Top 10 es un archivo donde el programa registra las mejores 10 marcas por cada nivel de dificultad (los jugadores que tardan menos en completar el juego). Si tenemos las 10 marcas y el jugador actual hace un mejor tiempo que esas marcas, hay que eliminar la marca con mayor tiempo para seguir teniendo un máximo de 10 marcas por nivel. La marca contiene el nombre del jugador y el tiempo (horas, minutos, segundos) que un jugador tardó en completar un

Programa 3 (30%): Juego Futoshiki

juego. Note que si usa el timer hay que calcular la duración del juego. Cuando un juego es terminado con éxito el programa regresa a la opción de Jugar.

Otras consideraciones:

- Antes de iniciar el juego el jugador debe dar un nombre (string de 1 a 20 caracteres). Este nombre no debe estar en el TOP 10 de cualquier nivel.
- Luego de dar el botón INICIAR JUEGO, este botón se deshabilita.
- En caso de haber configurado la opción de Timer, el jugador puede dejar el tiempo registrado en la configuración o modificarlo antes de INICIAR JUEGO. El tiempo empieza a correr cuando le den INICIAR JUEGO.
- En caso de no usar el reloj o el timer no debe aparecer esa parte en la pantalla. Para el uso del timer alguna de sus partes (horas, minutos, segundos) debe ser mayor a cero.
- En caso de haber configurado la opción de Timer y éste llegue a 0 y el juego no haya terminado se envía el mensaje TIEMPO EXPIRADO. ¿DESEA CONTINUAR EL MISMO JUEGO (SI O NO)??. Si responde SI entonces el timer pasa a ser reloj inicializado con el tiempo que se había establecido en el timer. Por ejemplo si el timer estaba para 1 hora y 30 minutos, ahora el reloj 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 reloj hay que validar los valores dados de horas, minutos y segundos. En la configuración están los rangos de estos datos.
- 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 picar una casilla y no haya seleccionado previamente un dígito se envía el error FALTA QUE SELECCIONE UN DÍGITO.

En caso de que ocurra un error el programa enviará un mensaje respectivo. El error se mantendrá en pantalla hasta que el usuario presione la tecla <Enter> para continuar. Con esto se pretende que el usuario vea el mensaje de error y luego decida cuando continuar.

**BORRAR
JUGADA**

Elimina la última jugada dejando la casilla vacía. Puede borrar todas las jugadas que ha hecho. Use un TDA pila para registrar la información de las jugadas que van sucediendo de tal manera que pueda implementar esta funcionalidad. Cada vez que se hace una jugada se agrega a la pila (fila y columna de la jugada), y si seleccionan este botón, se toma la última jugada agregada en la pila (el manejo de la pila es tipo LIFO: Last In First Out, último en

Programa 3 (30%): Juego Futoshiki

entrar primero en salir) y se borra la casilla respectiva de la cuadrícula. La jugada se quita de la pila. En el diseño de la solución describa la estructura del TDA usado y el algoritmo de manipulación.

Otras consideraciones:

- En caso de que ya no hayan más jugadas para borrar según la pila hay que enviar el mensaje NO HAY MÁS JUGADAS PARA BORRAR.
- Se puede seleccionar esta opción solamente si el juego ha iniciado, antes de ello la opción permanece deshabilitada.
- La pila de jugadas se reinicia en cada juego.

**TERMINAR
JUEGO**

Cuando el jugador selecciona esta opción se le pregunta

¿ESTÁ SEGURO DE TERMINAR EL JUEGO (SI o 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, antes de ello la opción permanece deshabilitada.

**BORRAR
JUEGO**

Cuando el jugador selecciona esta opción se le pregunta

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

Si responde SI vuelve a la opción de Jugar usando la misma partida pero eliminando todas las jugadas que hizo.

Si responde NO sigue jugando con el mismo juego.

Se puede seleccionar esta opción solamente si el juego ha iniciado, antes de ello la opción permanece deshabilitada.

TOP
10

Esta opción se puede usar en cualquier momento. Detiene el reloj si lo está usando. Despliega una sola pantalla con los registros de los mejores 10 primeros jugadores por cada nivel: aquellos que hicieron menos tiempo para completar el juego. En caso de no tener los 10 jugadores en algún nivel se despliegan los que se tengan. El Top 10 se guarda en el archivo "futoshiki2021top10.dat".

Represente el Top 10 con 3 listas, una por cada nivel de dificultad, los elementos de cada lista son tuplas con la información del registro.

TOP 10

<i>NIVEL DIFÍCIL:</i>	<i>JUGADOR</i>	<i>TIEMPO</i>
	1- Nombre jugador	1:30:15
	2- Nombre jugador	1:32:55
	...	
	10-	

<i>NIVEL INTERMEDIO:</i>	<i>JUGADOR</i>	<i>TIEMPO</i>
	1- Nombre jugador	1:10:21
	2- Nombre jugador	1:35:55
	...	
	10-	

<i>NIVEL FÁCIL:</i>	<i>JUGADOR</i>	<i>TIEMPO</i>
	1- Nombre jugador	0:7:23
	2- Nombre jugador	0:10:55
	...	
	10-	

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

GUARDAR JUEGO

Este botón se puede usar en cualquier momento que el juego haya iniciado, antes de ello permanece inhabilitado. Guarda en el archivo "futoshiki2021juegoactual.dat" todo el estado del juego actual: configuración, cuadrícula, nombre del jugador, etc. El objetivo es que el jugador pueda en cualquier momento guardar el juego y posteriormente continuarlo en el punto donde hizo el guardado del juego. Este archivo solo va a contener una partida. En caso de que haya una partida en el archivo, se borra y se guarda la del momento.

CARGAR JUEGO

Este botón se puede usar solamente cuando un juego no se haya iniciado, luego de ello permanece inhabilitado. Trae del archivo "futoshiki2021juegoactual.dat" el juego que fue guardado y lo pone en la pantalla como el juego actual con exactamente el mismo estado que tenía cuando fue guardado. El juego continúa cuando el jugador usa el botón de INICIAR JUEGO.

B. 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 "futoshiki2021configuración.dat" : (los valores por omisión –o default- están señalados con el círculo en rojo)

1. Nivel: ☒ Fácil
o Intermedio
o Difícil

2. Reloj: ☒ Si
o No
o Timer

Horas	Minutos	Segundos
0	30	0

Para el timer las horas pueden estar entre 0 y 2, los minutos entre 0 y 59 y los segundos entre 0 y 59. El timer debe tener al menos uno de estos valores. Hay que realizar estas validaciones y enviar los mensajes respectivos en caso de errores.

La medición del tiempo es tiempo real.

3. Posición en la ventana del panel de dígitos: ☐ Derecha
o Izquierda

Ayuda

Esta opción la usaremos para que el usuario pueda ver el Manual de Usuario

Programa 3 (30%): Juego Futoshiki

directamente en la computadora (despliega el pdf respectivo).

Acerca de

Puede poner esta opción para desplegar información “Acerca del programa” donde pondremos al menos los datos del nombre del programa, la versión, la fecha de creación y el autor.

Salir

Puede poner esta opción para salir del programa. También se puede salir con la opción de cerrar “X” en la GUI.

PARTIDAS DEL JUEGO

Registre partidas de este juego en el archivo “futoshiki2021partidas.dat”. Las partidas se van a almacenar en tres listas:

- Lista de partidas de nivel fácil
- Lista de partidas de nivel intermedio
- Lista de partidas de nivel difícil

Cada elemento de estas listas representa una partida por medio de una tupla. Y cada partida se componen de tuplas conteniendo una restricción o un dígito fijo de la partida.

Estructura de una tupla con restricciones:

(tipo de restricción: para filas “>” o “<”; para columnas “v” o “^”,
 posición de la casilla izquierda o superior de la desigualdad: índice de fila, índice de columna)

Estructura de una tupla con dígitos fijos:

(dígito: “1”, “2”, “3”,...,
 posición del dígito: índice de fila, índice de columna)

Contesta partida de ejemplo de nivel fácil:

	>			>		>	
4							2
			4				
					<	4	
	<		<				

tendríamos la siguiente estructura:

```
[
    # lista de partidas de nivel fácil:
    (
        # tupla de la partida número 1
        (">", 0, 0), (">", 0, 2), (">", 0, 3),      # fila 1
        ("4", 1, 0), ("2", 1, 4),                  # fila 2
        ("4", 2, 2),                                # fila 3
        ("<", 3, 3), ("4", 3, 4),                    # fila 4
        ("<", 4, 0), ("<", 4, 1)                    # fila 5
    ),
    # fin de la partida número 1
    ...
    # otras partidas
]
```

Instituto Tecnológico de Costa Rica
Escuela de Computación
Carrera: Ingeniería en Computación
Curso: Taller de Programación (IC-1803)
Semestre: I 2021
Programa 2 (30%): FUTOSHIKI

Busque algunas partidas, al menos 3 por cada nivel, y grábelas directamente en el archivo.

Temas Investigados

Librerías (usos nuevos)

- * Time (controlador de tiempo)
- * MessageBox (más tipos de mensajes)

Nuevas funciones de tkinter

- * Función update()
- * StringVal()

Software controlador de versiones

- * Git Hub

MARCO TEORÍCO

Librerías (usos nuevos)

Time (función sleep()):

¿Qué hace?

La función sleep suspende la ejecución de un proceso durante un periodo de tiempo indicado por el programador.

¿Cómo funciona?

En primera instancia, se debe de importar la librería time. Después, dentro de una función y un ciclo se asigna como parámetro la cantidad de segundos que debe esperar el ciclo para volver a “activarse”.

```
import time

while True:

    print("cada segundo")

    time.sleep(1)
```

Al ejecutar ese anterior código, se imprimirá “cada segundo”, cada segundo transcurrido.

```
cada segundo
cada segundo
cada segundo
```

MessageBox (askyesno):

¿Qué hace?

Dentro de la librería de messagebox existen varios tipos de mensajes, uno de esos es el askyesno.

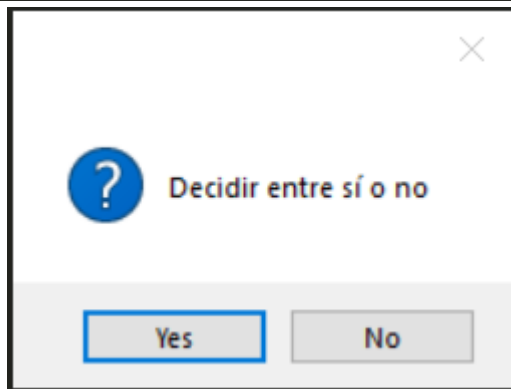
¿Cómo funciona?

Primero se debe de llamar la librería de messagebox. Posteriormente, se fija una variable con un mensaje de este tipo, al seleccionar Yes or No desde el apartado gráfico se retornará un True o un False respectivamente a la respuesta. Después el programador puede usar esa respuesta a su conveniencia.

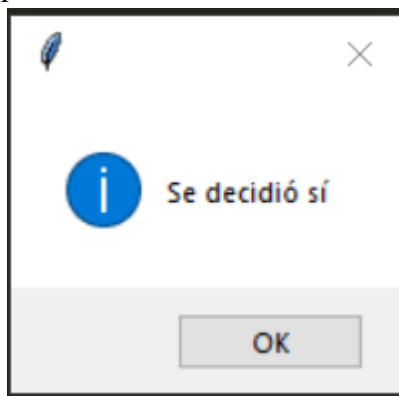
```
from tkinter import messagebox

siono=messagebox.askyesno("", "Decidir entre sí o no")

if siono==True:
    messagebox.showinfo("", "Se decidió sí")
else:
    messagebox.showinfo("", "Se decidió no")
```



Se clickeó en el botón “Yes”, por lo que se retornó un True.



Nuevas funciones tkinter

Función update():

¿Qué hace?

Esta función “recarga” los valores activos en una ventana, y si alguno de estos presentar cambios los enseña.

¿Cómo funciona?

Esta función debe estar después del nombre de la ventana, por ejemplo: root.update().

Lo que hará es recargar los valores previamente dados en esa ventana. Si esta función es metida dentro de un ciclo está constantemente recargando los valores dados, es por esto que sirve muy bien para crear relojes.

“Bloquear” Entrys:

¿Qué hace?

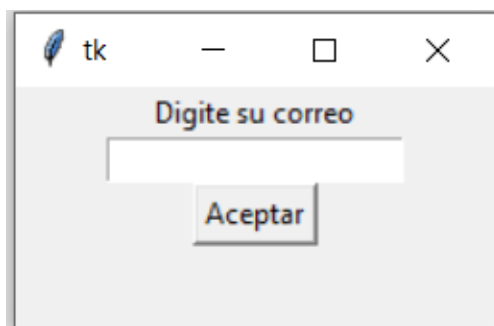
Se ponen las entradas en un modo de “solo lectura”, de esta forma, el usuario no puede modificar el texto que hay dentro de la entrada.

¿Cómo funciona?

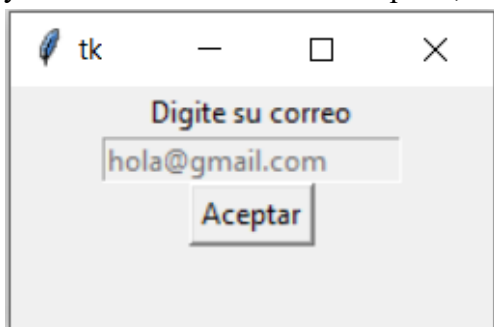
Se debe llamar a la función tkinter. Crear una entrada y algo para ejecutar un comando (como un botón), una vez una vez este comando sea ejecutado (la entry debe ser enviada como parámetro), la entrada pasará a modo lectura.

```
def bloquear(entrada):  
    entrada.config(state="disable")  
  
import tkinter  
  
ventana=tkinter.Tk()  
ventana.geometry("200x100")  
  
lcorreo=tkinter.Label(ventana,text="Digite su correo")  
lcorreo.pack()  
  
entrada=tkinter.Entry(ventana)  
entrada.pack()  
  
boton=tkinter.Button(ventana,text="Aceptar",command=lambda: [bloquear(entrada)])  
boton.pack()  
  
ventana.mainloop()
```

Al ejecutar este código, la GUI se verá así:



Al digitar un correo en la entrada y clickeando en el botón “Aceptar”, la entrada pasará a modo lectura.



Software controlador de versiones

Utilidad:

¿Qué hace?

Son sistemas de monitoreo ante cambios en archivos. A su vez, ayudan en la gestión de estos.

¿Cómo funciona?

Estos controladores monitorean cualquier acción de adición, sustracción o modificación de datos dentro de un archivo, al ocurrir esto este indica los cambios ocurridos y pide un tipo de justificación al usuario (etiqueta). Lo bueno de estos programas es que pueden ser manipulados por varias personas al mismo tiempo, permitiendo el trabajo “en la nube”. Existen a su vez, controladores especiales para código informático de tal magnitud que debe ser corrido en servidores especiales y no en computadores comunes.

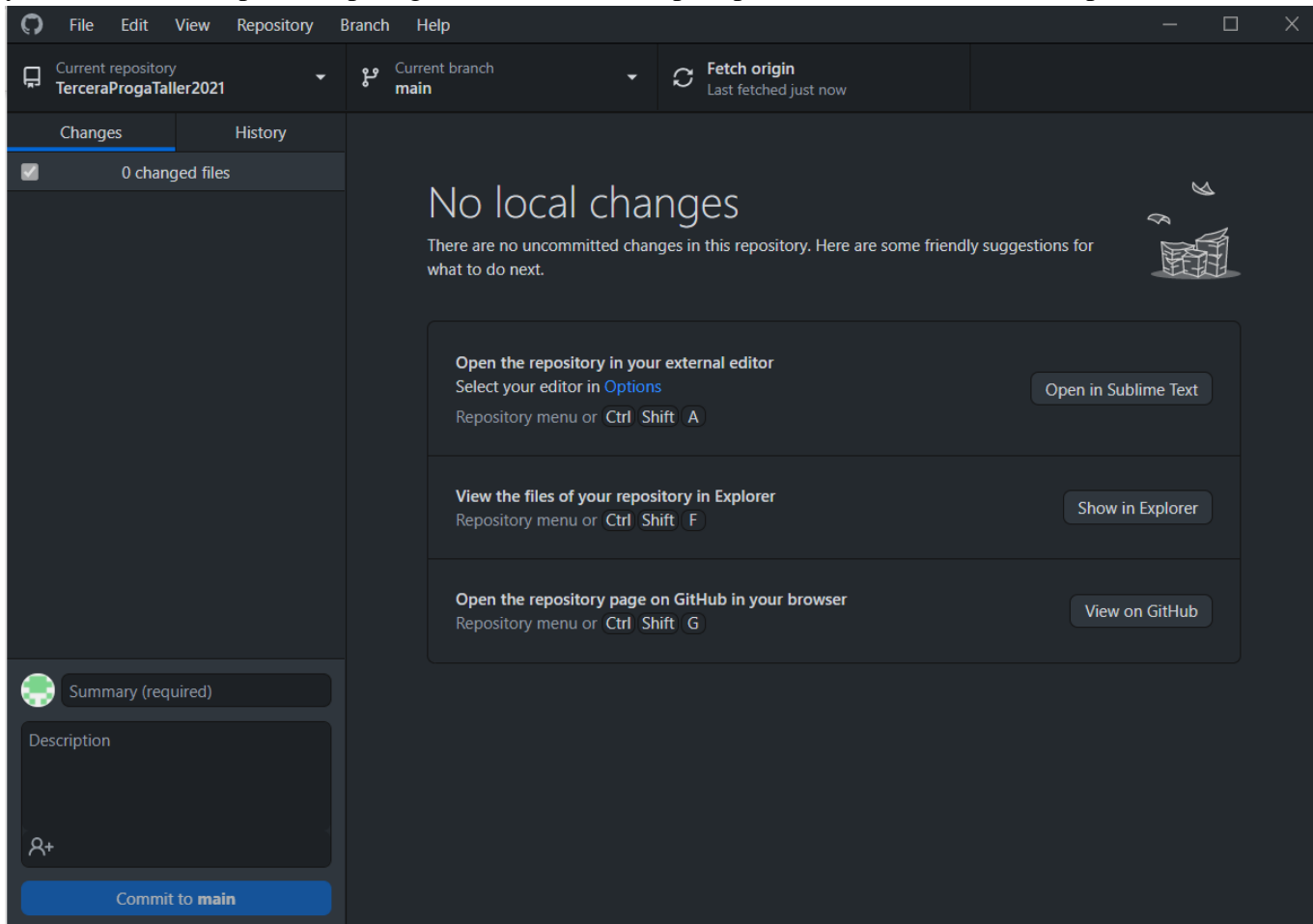
Git Hub:

¿Qué hace?

GitHub es uno de los controladores de versiones más conocidos en el mundo. Cuentan a su vez con una página web tipo repositorio de códigos donde existe una infinidad de proyectos informáticos ya desarrollados o bien en este proceso.

¿Cómo funciona?

Al instalar la aplicación “GitHub Desktop”, se debe ingresar con la cuenta de la página <https://github.com/>, una vez hecho esto, podemos empezar a crear nuestros repositorios. Estos deben estar anclados a alguna carpeta de la computadora, por lo tanto, cada que haya un cambio en un archivo de esta carpeta, se deberá agregar una etiqueta y un comentario (opcional) para guardar los cambios, para posteriormente subirlos al repositorio en línea.



Repositorio en github del proyecto: <https://github.com/Elmono56/TerceraProgaTaller2021>

Conclusiones

Problemas encontrados en el desarrollo y soluciones de los mismos

Durante el desarrollo del programa hubo varios problemas, uno de estos fue el temporizador, en específico, el control del tiempo para medir el paso de los segundos, por lo que se investigó y se dio con una función desconocida hasta el momento de la librería time. Esta función permite “parar” los ciclos durante un periodo de tiempo determinado, gracias a esto se podía hacer las restas o sumas correspondientes al reloj, sin embargo, la pantalla se congelaba y se indagó en el problema, este fue solucionado con la implementación de la función update, de la librería de tkinter.

Otro problema que se presentó fue en la pantalla de juego, ya que los jugadores podrían cambiar su nombre a otro durante la propia partida, incluso saltándose las restricciones de este. Por lo que se pensó una solución y se tenía, simplemente se tuvo que buscar el cómo bloquear una entrada de tkinter. Una vez esta bloqueada, no se pueden realizar cambios en su texto. Por lo que, una vez empezada la partida, esta ventana se bloquea y el jugador no podrá modificar el nombre que puso.

A su vez, hubo otro problema respectivo a la pantalla de juego. Este constaba que en ciertos botones se necesitaba que el jugador tomara decisiones de sí o no, por lo que se procedió a investigar métodos para realizar estas decisiones en un apartado gráfico. En un principio se planteó la posibilidad de crear una ventana pequeña para esto, sin embargo, la solución fue más sencilla de lo pensado; la librería messagebox cuenta con un tipo de mensaje el cual es para recibir una respuesta de sí o de no, por lo que se procedió a buscar un método para obtener esta respuesta del usuario, constó con hacer una variable que contuviese al mensaje para luego evaluarla.

Aprendizajes obtenidos

Se aprendió a crear un temporizador de dos maneras, la primera de tipo pseudocódigo y la segunda en el lenguaje de Python.

Un buen proceso de programación (entender el problema, diseño de algoritmos, programación, entre otros) especialmente en un programa que requiera de interfaz gráfica, disminuye considerablemente las horas de trabajo, ya que al ir con una idea del cómo se va a montar este apartado, sus elementos y los roles que va a tener cada elemento, así como sus funciones de requerirlas, se agiliza demasiado el proceso de programación final.

Se reforzó la idea que el intentar e insistir en que algo sirva no es lo óptimo, en internet existen muchas guías, muchos foros y muchas páginas con información y solución a los problemas que se encuentran en esta labor de la programación. Por lo que, en varios casos, es muy recomendable dar una chequeada en el internet ante un problema extraño.

Diseño y explicación de las soluciones

Estructuras de datos

Dentro del programa, los datos se van a guardar en variables globales, esto para facilitar su llamada en donde se requieran, sin tener que enviarlas como un parámetro.

En el caso de la configuración, los datos default o bien los seleccionados por el usuario son guardados en una lista en el formato [dificultad (valores de 1 a 3 dependiendo de la opción escogida), tiempo (valores de 1 a 3 dependiendo de la opción escogida), horas, minutos, segundos, lado del panel (1 o 2 dependiendo de la opción escogida)]. Si se seleccionó sin tiempo o solo con reloj, los valores de horas, minutos y segundos serán 0.

La variable pila, va almacenando en una lista los movimientos realizados (y su valor), de tal manera que su última posición es la última jugada realizada.

La variable datos almacena los espacios de las cuadrículas.

La utilidad de la variable cont es ir contando la cantidad de movimientos realizados, para lograr moverse dentro de los espacios de la cuadrícula.

La variable partida, se inicializa con los mismos datos de la lista de configuración, sin embargo, al guardarse una partida pasa a tener los datos de configuración, el nombre del jugador, el contador de movimientos y los datos de la lista de pila.

Por último, la variable booleana tiempocorriendo es utilizada para parar el tiempo si esta se iguala a False en algún proceso.

Se emplearon listas para contener las respuestas correctas del mapa de cada nivel.

Archivos

Se creó el archivo de futoshiki2021configuración.dat, el cual administra los datos de la configuración.

Aspectos principales

El programa cuenta con 3 ventanas gráficas.

Se decidió hacer la cuadrícula con entradas.

El programa cuenta con una amplia cantidad de variables globales.

Estadísticas de tiempo

Actividad Realizada	Horas
Análisis del problema	15
Diseño de algoritmos	15
Investigación de ...	20
Programación	65
Documentación interna	5
Pruebas	15
Elaboración del manual de usuario	5
Elaboración de documentación del proyecto	10
Etc.	
TOTAL	145

LISTA DE REVISIÓN DEL PROYECTO

Concepto	Puntos originales	Avance 100%/0	Puntos obtenidos	Análisis de resultados
Opción Configurar	6	100%	6	
Despliegue y manipulación de la ventana del juego: cuadrícula con sus restricciones y dígitos fijos. Incluye el despliegue de partidas.	12	90%	11	No se implementaron números fijos.
Despliegue y manipulación de la ventana del juego: otros elementos	6	100%	6	
Botón Iniciar Juego	10	100%	10	
Crear Top 10	12	0%	0	Se complicó su programación.
Botón Borrar Jugada	5	100%	5	
Botón Terminar Juego	2	100%	2	
Botón Borrar Juego	2	100%	2	
Botón Top 10	10	0%	0	Se complicó su programación.
Botón Guardar Juego	5	100%	5	
Botón Cargar Juego (incluye el despliegue del mismo)	15	100%	15	
Ayuda (manual de usuario)	5	100%	5	
Reloj tiempo real	5	100%	5	
Timer tiempo real	5	100%	5	
TOTAL	100	77%	77	
Partes desarrolladas adicionalmente	3	100%	3 Total 80	Botón Acerca de.

Bibliografía

Tkinter - Read only Entry Widget - GeeksforGeeks. (2020, March 30). GeeksforGeeks.

<https://www.geeksforgeeks.org/tkinter-read-only-entry-widget/?ref=rp>

Bijay Kumar Sahoo. (2020, December 28). *Python Tkinter MessageBox + Examples - Python Guides*. Python Guides.

<https://pythonguides.com/python-tkinter-messagebox/>

How to create a timer using tkinter? (2021). Tutorialspoint.com. [https://www.tutorialspoint.com/how-to-create-a-timer-](https://www.tutorialspoint.com/how-to-create-a-timer-using-tkinter)

[using-tkinter](https://www.tutorialspoint.com/how-to-create-a-timer-using-tkinter)

time — Time access and conversions — Python 3.9.6 documentation. (2021). Python.org.

<https://docs.python.org/3/library/time.html#time.sleep>

Atlassian. (2021). *Software de control de versiones: descripción general / Bitbucket*. Bitbucket.

<https://bitbucket.org/product/es/version-control-software>

Red de aprendizaje. (2018). *¿Cómo crear un repositorio controlador de versiones Git en GitHub 2018?* [YouTube

Video]. In *YouTube*. <https://www.youtube.com/watch?v=GCmSBXz--j0>