

Instituto Tecnológico de Costa Rica

Sede Central Cartago

Escuela de Ingeniería en Computación

Taller de programación

Programa III

“Sudoku”

Diana María Calderón Mora, 2021093558

Natalia Sofía Rodríguez Solano, 2021033307

GR-1

Docente:

William Mata Rodríguez

II Semestre

2021

Tabla de Contenidos

Enunciado del Proyecto	3
Temas investigados	5
Software de control de versiones	5
¿Qué es?	5
Ventajas de control de versiones.....	5
Importancia en ingeniería de software.....	5
Funciones utilizadas del software de control de versiones utilizadas en el proyecto.	6
Nuevas características exploradas de Tkinter	6
Config ():	6
Clases:	6
Generación de archivo PDF:	8
Subprocess- Generar procesos adicionales:	8
Archivos .dat():.....	9
Conclusiones:	9
Lista de revisión del proyecto	10
Bibliografía:.....	11

Enunciado del Proyecto

Sudoku es un pasatiempo que se popularizó internacionalmente después del año 2000 cuando diversos periódicos empezaron a publicarlo en la sección de pasatiempos.

Consiste en llenar con dígitos del 1 al 9 cada una de las casillas de una cuadrícula que tiene un total de 9 x 9 casillas. La cuadrícula la podemos ver como una matriz de 9 filas y 9 columnas. A la vez esta cuadrícula se divide en subcuadrículas de 3 x 3 casillas. Para empezar a jugar hay algunos dígitos fijos en la cuadrícula, lo cual determina el nivel de dificultad del juego. Aunque usualmente se usan dígitos, lo que interesa es que sean grupos de 9 elementos diferentes: por ejemplo 9 letras, 9 colores, 9 frutas, etc.

La regla de este juego es que un mismo elemento no se puede repetir:

- ✚ En una misma fila (casillas horizontales)
- ✚ En una misma columna (casillas verticales)
- ✚ En las subcuadrículas (3x3)

Además, los elementos que aparecen al inicio del juego quedan fijos, no se pueden cambiar.

Ejemplo de un Sudoku: ya tiene los dígitos preestablecidos para empezar a jugar.

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

¿Se puede llenar la casilla de la fila 3 columna 1 con un 4? ¿Por qué?

¿Se puede llenar la casilla de la fila 4 columna 7 con un 8? ¿Por qué?

¿Se puede llenar la casilla de la fila 1 columna 6 con un 9? ¿Por qué?

¿Se puede llenar la casilla de la fila 9 columna 5 con un 6? ¿Por qué?

El programa tendrá un menú principal usando botones (Buttons widget) para acceder las funcionalidades del programa, es decir, lo que el programa hace. Usted puede agregar otras funcionalidades que vayan a mejorar el producto.

Objetivos del proyecto:

- ✚ Reforzar los conocimientos acerca de la metodología general de desarrollo de programas de mayor alcance:
 - Entender el problema.
 - Diseñar algoritmo (plantear la solución por partes, luego desarrollar cada una).
 - Codificar algoritmo.
 - Probar y evaluar el programa.
- ✚ Aplicar y reforzar aspectos del lenguaje Python 3:
 - Uso de diversos componentes del lenguaje.
 - Uso de estructuras condicionales y de repetición de procesos.
 - Desarrollo de funciones.
 - Utilización de estructuras de datos nativas de Python y TDA (Tipos de Datos Abstractos).
 - Uso de clases (class-objetos). Solo en determinadas funcionalidades.
- ✚ Ampliar el conocimiento acerca del desarrollo de GUI en Python.
- ✚ Aplicar buenas prácticas de programación básicas: documentación interna y externa del programa, nombres significativos, uso y reutilización de funciones.
- ✚ Validación de los datos de entrada: todos los datos de entrada se deben validar según restricciones que se indican en cada uno de ellos.
- ✚ Usar archivos de datos.
- ✚ Usar algún software de control de versiones de software, por ejemplo, Git o algún otro que usted decida.
- ✚ Fomentar en el estudiante la investigación: temas no tratados en el curso, pero necesarios para hacer el proyecto. Dichos temas deben ser explicados detalladamente en la documentación del proyecto. Entre los tópicos a investigar están:
 - Software de control de versiones:
 - ✓ ¿Qué es?
 - ✓ Importancia en ingeniería de software.
 - ✓ Explicar las funciones utilizadas del software de control de versiones usado en el proyecto.
 - Nuevas características de tkinter exploradas en este proyecto.
 - Cualquier otro aspecto necesario para ofrecer su solución.

Temas investigados

Software de control de versiones

¿Qué es?

Los sistemas de control de versiones o también conocido como "control de código fuente", es un software que ayudan a realizar un seguimiento de los cambios realizados un código a lo largo del tiempo; a medida que un desarrollador edita el código, el sistema de control de versiones toma una instantánea de los archivos. Seguidamente, guarda esa instantánea de forma permanente para que se pueda recuperar más adelante si es necesario.

Sin control de versiones, los desarrolladores tienen la tentación de mantener varias copias de código en su equipo, lo cual es peligroso porque es fácil cambiar o eliminar un archivo en una copia de código incorrecta, lo que pondría en juego la pérdida del trabajo. Los sistemas de control de versiones resuelven este problema administrando todas las versiones del código, pero presentando al equipo una sola versión a la vez.

Para casi todos los proyectos de software, el código fuente es un activo valioso cuyo valor debe protegerse: para la mayoría de equipos de software, el código fuente es un repositorio del conocimiento de valor incalculable y de la comprensión sobre el dominio del problema que los desarrolladores han recopilado y perfeccionado con un esfuerzo cuidadoso; el control de versiones protege el código fuente tanto de las catástrofes como del deterioro casual de los errores humanos y las consecuencias accidentales.

Ventajas de control de versiones

- ✚ Creación de flujos de trabajo: Los flujos de trabajo de control de versiones evitan el caos de todos los usuarios que usan su propio proceso de desarrollo con herramientas diferentes e incompatibles.; los sistemas de control de versiones proporcionan permisos y cumplimiento de procesos para que todos permanezcan en la misma página.
- ✚ Trabajar con versiones: Cada versión tiene una descripción que corresponden a los cambios en la versión, para posteriormente saber cómo corregir un error o agregar una característica; dichas descripciones ayudan al equipo a seguir los cambios en el código por versión en lugar de por cambios de archivos individuales. El código almacenado en versiones se puede ver y restaurar desde el control de versiones en cualquier momento según sea necesario, lo que facilita la base del nuevo trabajo fuera de cualquier versión del código.
- ✚ Código juntos: El control de versiones sincroniza las versiones y se encarga de asegurarse de que los cambios no entren en conflicto con otros cambios de otros. El equipo se basa en el control de versiones para ayudar a resolver y evitar conflictos, incluso cuando los usuarios realicen cambios al mismo tiempo.
- ✚ Mantener un historial: El control de versiones mantiene un historial de cambios a medida que el equipo guarda nuevas versiones del código. Este historial se puede revisar para averiguar quién, por qué y cuándo se realizaron los cambios; el historial proporciona a los equipos la confianza de experimentar, ya que es fácil revertir a una versión anterior buena en cualquier momento; permite que cualquier usuario base trabaje desde cualquier versión del código.
- ✚ Automatización de tareas: Las características de automatización del control de versiones ahorran tiempo y generan resultados coherentes; automatice las pruebas, el análisis de código y la implementación cuando se guarden nuevas versiones en el control de versiones.

Importancia en ingeniería de software

La ingeniería de software se ha convertido en una de las áreas clave en el auge de la tecnología en empresas y sociedad.

La ingeniería de software abarca un amplio abanico de campos de la informática y la ciencia de la computación. Esta tecnología cumple ahora dos funciones, es el producto, pero también sirve como vehículo que transporta el producto; sirve de base de control para el producto como motor, como un sistema operativo o un sistema de gestión de redes.

Dicha permite una serie de ventajas e importancia ya que permite facilitar el control en el proceso de desarrollo de software, de igual forma permite suministrar a los desarrolladores las bases para construir software de alta calidad en una forma eficiente; permite definir una disciplina que garantice la producción y el mantenimiento de los productos software desarrollados en el plazo fijado y dentro del costo estimado.

En la construcción y desarrollo de proyectos se aplican métodos y técnicas para resolver los problemas, la informática aporta herramientas y procedimientos sobre los que se apoya la ingeniería de software de igual forma permite mejorar la calidad de los productos de software y aumentar la productividad.

Hay muchas cosas que pueden tardar tiempo como desarrollador; la reproducción de errores, el aprendizaje de nuevas herramientas y la adición de nuevas características o contenido son algunos ejemplos. A medida que las demandas de los usuarios se escalan verticalmente, el control de versiones ayuda a los equipos a trabajar juntos y enviarse a tiempo; la ingeniería en software es muy importante ya que con ella se puede analizar, diseñar, programar y aplicar un software de manera más correcta y organizada, cumpliendo con todas las especificaciones del cliente y el usuario final. Lo anterior es posible gracias a los objetivos que esta propone.

Funciones utilizadas del software de control de versiones utilizadas en el proyecto.

- ✚ Crear repositorio en el software de control de funciones (en este caso se utilizó GitHub).
- ✚ Subir archivos del programa, diferentes versiones del programa.
- ✚ Ver los cambios realizados en el repositorio: Para tener la posibilidad de ver los cambios, se ha hecho un repositorio, por lo que se utiliza el comando “git status”.
- ✚ Añadir cambios: para poder subir el archivo modificado al repositorio alojado en GitHub, lo primero que se debió hacer fue ir a la instrucción “git add”, seguido de la ruta hasta llegar al archivo que se quiso subir.
- ✚ Compartir el repositorio con otro usuario, con el equipo de trabajo.
- ✚ Se pudieron descargar las versiones ya creadas para su revisión y modificación.

Nuevas características exploradas de Tkinter

Durante el desarrollo del programa se han observado y manipulado algunas características de Python como lo son las siguientes:

Config ():

“Config ()” se utiliza para acceder a los atributos de un objeto después de su inicialización.

Clases:

Las clases proveen una forma de empaquetar datos y funcionalidad juntos; al crear una nueva clase, se crea un nuevo tipo de objeto, permitiendo crear nuevas instancias de ese tipo. Cada instancia de clase puede contener atributos adjuntos para mantener su estado; dichas instancias de la clase también pueden tener métodos (definidos por su clase) para modificar su estado.

Comparado con otros lenguajes de programación, el mecanismo de clases de Python agrega clases con un mínimo de nuevas sintaxis y semánticas. Es una mezcla de los mecanismos de clases encontrados en C++ y Modula-3. Las clases de Python proveen todas las características normales de la Programación Orientada a Objetos: el mecanismo de la herencia de clases permite múltiples clases base, una clase derivada puede

sobre escribir cualquier método de su(s) clase(s) base, y un método puede llamar al método de la clase base con el mismo nombre. Los objetos pueden tener una cantidad arbitraria de datos de cualquier tipo; de igual forma que con los módulos, las clases participan de la naturaleza dinámica de Python: se crean en tiempo de ejecución, y pueden modificarse luego de la creación.

Como he se ha mencionado anteriormente, las clases y los objetos sirven para crear tipos de datos definidos por el usuario, y la crear instancias de una clase hace relación a la creación de objetos de ese tipo. Las clases y los objetos son considerados los principales bloques de desarrollo para Python, el cual es un lenguaje de programación orientado a objetos.

La estructura de clase más simple en Python luciría de la siguiente manera:

```
class ClassName:
    statements
```

La definición de una clase comienza con la palabra clave “class” y “className” el cual se refiere a el nombre de la clase (identificador); hay que tomar en cuenta que el nombre de la clase sigue las mismas reglas que los nombres de variables en Python, es decir, sólo pueden comenzar con una letra o un subrayado _, y de igual forma sólo pueden contener letras, números o guiones bajos, además, según PEP 8 (Guía de estilo para la programación en Python), se recomienda que los nombres de las clases estén capitalizados.

Seguidamente se debe definir una clase, y luego se podrá crear una instancia (objeto) de esta clase. Como ejemplo de dicho proceso se puede ver las siguientes imágenes:

```
class Person:
    pass
```

```
jorge = Person()
```

Métodos:

Los métodos son cómo funciones en Python, ya que se definen con la palabra clave “def” y cuentan con el mismo formato que las funciones; dentro de la clase, se define un método que imprima lo que se requiere en el programa.

```
class Person:
    name = ''
    school = ''

    def print_name(self):
        print self.name

    def print_school(self):
        print self.school

jorge = Person()
jorge.name = 'Jorge'
jorge.school = 'Universidad de la vida'
jorge.print_name()
jorge.print_school()
```

Como se pudo ver en el ejemplo anterior hay un método que imprime el nombre (name) y la escuela (school) de una persona (Person).

La principal diferencia entre las funciones y los métodos es que los métodos necesitan tener un argumento convenientemente llamado “self”, que se refiere al objeto del método que está siendo llamado. Es importante resaltar que, en la llamada al método, no se requiere pasar “self” como un argumento, Python se encargará de dicha función (nota: si no se pone el “self” como argumento, Python arrojará un error como el que se muestra a continuación:

```
Traceback (most recent call last):
  File "test.py", line 14, in
    jorge.print_name()
TypeError: print_name() takes no arguments (1 given)
```

Inicialización:

Anteriormente, se explicó inicializar los atributos, dándoles un valor vacío "", más la forma más elegante de inicializar variables con sus valores predeterminados; es un método especial, con nombre `__init__` (el método se considera especial y será tratado de forma especial, es por eso que tiene subrayados dobles al principio y al final). Se utilizaría de la siguiente manera:

```
class Person:
    def __init__(self, n, s):
        self.name = n
        self.school = s

    def print_name(self):
        print self.name

    def print_school(self):
        print self.school

jorge = Person('Jorge', 'Universidad de la vida')

jorge.print_name()
jorge.print_school()
```

Las clases brindan la capacidad de crear tipos de datos, y con los objetos permite crear instancias de estos tipos de datos; de igual forma las clases se componen de atributos (propiedades) y métodos que son acciones que realizamos en dichos atributos.

Generación de archivo PDF:

Subprocess- Generar procesos adicionales:

El módulo estándar subprocess permite invocar procesos desde Python y comunicarse con ellos: enviar datos a la entrada (stdin) y recibir la información de salida (stdout); además, esperar a que el proceso finalice o bien terminarlo prematuramente, y obtener el valor de retorno. Resulta ideal y es el método recomendado para ejecutar comandos del sistema operativo o lanzar programas (en lugar de la tradicional `os.system()`) y opcionalmente interactuar con ellos.

La forma más sencilla de ejecutar un comando o invocar un proceso es vía la función `call()` (desde Python 2.4 hasta 3.4) o `run()` (3.5+).

El módulo **subprocess** es una parte de un proceso que se ejecuta como si fuera un proceso completo; son usados para dividir procesos en unidades más pequeñas; este de igual forma permite iniciar aplicaciones o procesos distintos durante la ejecución de un programa.

Archivos .dat():

Los archivos en el formato DAT se adjuntan con el .dat extensión y se pueden crear, abrir y editar utilizando la aplicación Microsoft Windows asociado o el software de Mac que se utilizó para crear el archivo DAT. Estos archivos DAT son archivos de datos genéricos que también se pueden abrir utilizando cualquier aplicación de edición de texto de Microsoft Windows o cualquier editor de texto elaborado por las plataformas Mac. Estos archivos DAT pueden ser codificados en formato de texto plano, mientras que algunos archivos DAT se implementan con especificaciones de codificación binaria. Los usuarios de sistemas basados en Microsoft Windows pueden utilizar el editor de texto Bloc de notas de Microsoft para abrir un archivo DAT, mientras que los usuarios de Mac pueden utilizar cualquier editor de texto de Apple para hacer esto, aunque es aconsejable para determinar el software específico que se desarrolló para crear o abrir y ver el contenido almacenado en un archivo DAT. Por ejemplo, algunos VCDs contienen archivos DAT que se pueden abrir y ver el uso de Microsoft Windows Media Player, y al abrir estos archivos DAT con Microsoft Bloc de notas, se muestran los códigos binarios y no el contenido de vídeo almacenados en el archivo DAT.

Conclusiones:

Durante el desarrollo del proyecto se encontraban ciertas dificultades como lo son las siguientes; al inicio no se encontraba el método para crear los botones sin tener que crearlos uno por uno; otro problema encontrado fue que a la hora de elegir un tipo de partida al inicio se utilizó “random”, mas este método que generó errores, por lo que se tuvo que buscar una manera alternativa la cual fue utilizar el “randit()” para corregir dicho error.

Se tuvo investigar para saber como desarrollar las funciones de reloj y el timer que se requería como parte de las opciones requeridas en el programa. Se tuvieron que crear varias funciones en las clases para poder acceder a los objetos

En cuanto a lo que fueron aprendizajes mediante la investigación de métodos se supo cómo manipular y utilizar métodos como lo son las clases, lectura y escritura de diferentes tipos de archivos, y lo que fue la creación de pdf, desde el código fuente.

Estadísticas de tiempos

Actividad Realizada	Horas
Análisis del proyecto - Leer y entender el proyecto - Etc.	2
Diseño de algoritmos	20
Investigación de ...	20
Programación	80
Documentación interna	4
Pruebas	10
Elaboración del manual de usuario	5
Elaboración de documentación del proyecto	5
Etc.	-
TOTAL	142

Lista de revisión del proyecto

Concepto	Puntos	Puntos obtenidos	Avance 100/%/0	Análisis de resultados
Menú	1	1	100	-
Opción Jugar (despliegue del juego)	15	15	100	-
Botón Iniciar Juego	10	10	100	-
Crear Top X	5	5	100	-
Botón Deshacer Jugada	8	8	100	-
Botón Rehacer Jugada	8	8	100	-
Botón Terminar Juego	3	3	100	-
Botón Borrar Juego	3	3	100	-
Botón Top X	10	10	100	-
Formato PDF	2	2	100	-
Botón Guardar Juego	4	4	100	-
Botón Cargar Juego incluyendo su despliegue	8	8	100	-
Opción Configurar	8	8	100	-
Ayuda (manual de usuario)	5	5	100	-
Acerca de	1	1	100	-
Salir	1	1	100	-
Reloj y Timer tiempo real	8	8	100	-
TOTAL	100	100	100	-
Partes desarrolladas adicionalmente	-	-	-	-

Bibliografía:

Microsoft (19 de mayo de 2021). ¿Qué es el control de versiones? Microsoft. Recuperado de: <https://docs.microsoft.com/es-es/devops/develop/git/what-is-version-control>

Atlassian Bitbucket (s.f). ¿Qué es el control de versiones? Atlassian Bitbucket. Recuperado de: <https://www.atlassian.com/es/git/tutorials/what-is-version-control>

Unir (6 de abril de 2021). Ingeniería de software: qué es, objetivos y funciones del ingeniero. Unir. Recuperado de: <https://mexico.unir.net/ingenieria/noticias/ingenieria-de-software-que-es-objetivos/>

González. L, (21 marzo de 2019). Herramientas de control de versiones. ¿Por qué debes usarlas? Dinahosting. Recuperado de: <https://dinahosting.com/blog/herramientas-de-control-de-versiones/>

Acens (23 de julio de 2015). Control de versiones: Git y GitHub. Acens. Recuperado de: <https://www.acens.com/comunicacion/white-papers/control-versiones-git-github/>

Stackoverflow (febrero de 2018). Utility of config() in python tkinter. Stackoverflow. Recuperado de: <https://stackoverflow.com/questions/42511123/utility-of-config-in-python-tkinter/42511195>

Programacion.net (s.f). Cómo funcionan las clases y objetos en Python. Programacion.net. Recuperado de: https://programacion.net/articulo/como_funcionan_las_clases_y_objetos_en_python_1505

Python (2021). 9. Clases. Python. Recuperado de: <https://docs.python.org/es/3/tutorial/classes.html>

El módulo Python3 de la semana (2019). subprocess — Generar procesos adicionales. El módulo Python3 de la semana. Recuperado de: <https://rico-schmidt.name/pymotw-3/subprocess/index.html>

Unipython (s.f). Subprocesos en Python. Unipython. Recuperado de: <https://unipython.com/subprocesos-python/>

Recursos Python (1 de diciembre de 2017). subprocess – Creación y comunicación con procesos. Recursos Python. Recuperado de: <https://recursospython.com/guias-y-manuales/subprocess-creacion-y-comunicacion-con-procesos/>

Adrián. L (19 de octubre de 2019). Archivos .dat: ¿Qué son estos archivos y cómo puedo abrirlos? Profesional review. Recuperado de: <https://www.profesionalreview.com/2019/10/19/archivos-dat-que-son-estos-archivos-y-como-puedo-abrirlos/>