

**Instituto Tecnológico de Costa Rica**

**Escuela de Computación, Ingeniería en Computación**



**Curso IC-1803**

**Taller de Programación**

**Grupo 03**

Proyecto Programado #1

"2048"

**Elaborado por:**

Katerine Guzmán Flores

2019390523

**Profesor del curso:**

William Mata Rodriguez

**Semestre II**

**2019**

# Contenido

<b>Contenido</b>	<b>2</b>
<b>Enunciado del proyecto</b>	<b>3</b>
<b>Importancia de la documentación de software</b>	<b>11</b>
Tipos de documentación e importancia del software	11
Importancia de la documentación	11
<b>Programación por eventos.</b>	<b>11</b>
¿Qué son los eventos?	12
<b>Uso de interfaz gráfica de usuario (GUI: Graphical User Interface) en el desarrollo de software.</b>	<b>12</b>
Desarrollo de GUI con tkinter	13
<b>Explicación de las funciones del “debugger” del IDE que está usando.</b>	<b>14</b>
The Debugger	14
Stepping	14
Globals Area	14
Locals Area	14
The Go and Quit Buttons	14
Break Points	15
<b>Comandos de Tkinter utilizados:</b>	<b>15</b>
<b>Conclusiones</b>	<b>18</b>
<b>Tabla Evaluaciones</b>	<b>20</b>
<b>Referencias</b>	<b>22</b>

# Enunciado del proyecto

2048 es un juego en línea y para móviles creado en el año 2014 por el desarrollador web italiano Gabriele Cirulli a la edad de 19 años. El objetivo del juego es deslizar casillas en una cuadrícula para combinarlas y crear una casilla con el número 2048. Se le considera un rompecabezas de deslizamiento y se han creado diversas versiones. Esta es una imagen de un juego en progreso:



(fuente: De Gabriele Cirulli - <https://github.com/gabrielecirulli/2048>, MIT, <https://commons.wikimedia.org/w/index.php?curid=31695218>)

## ¿ Cómo se juega ?

2048 se juega en una cuadrícula o tablero de 4 x 4, matriz con 4 filas y 4 columnas.

Al inicio del juego aparecen aleatoriamente en cualquier parte del tablero dos números: 2 y 2 o 2 y 4.

A partir de aquí se utilizan las teclas direccionales (izquierda, derecha, arriba, abajo) para mover las casillas, las cuales se deslizan en su totalidad por el tablero.

Al usar una direccional, todos los números en las casillas se mueven en esa dirección mientras haya casillas vacías (las que no tienen un número), si en ese movimiento dos casillas con el mismo número quedan juntas entonces esas casillas se funden en una sola cuyo número será el equivalente a la suma de ambas casillas. Por ejemplo, si dos casillas con el número 8 quedan juntas al hacer el movimiento, se combinarán en una sola casilla con el valor 16. Sin embargo, la casilla resultante no puede combinarse con otra en una misma jugada.

Después de realizar una jugada aparecerá en alguna casilla vacía el valor 2 o el valor 4, todo de forma aleatoria.

El juego registra el puntaje del jugador. El puntaje comienza en 0 y cuando dos casillas se juntan el puntaje aumenta en el valor de la casilla resultante. También se despliega el mejor puntaje.

El juego termina bajo dos condiciones:

- El jugador pierde cuando no pueda hacer más movimientos
- El jugador gana cuando logra que una casilla llegue al número 2048 (de allí el nombre del juego) como en esta partida:



Puede jugar una versión aquí: <https://juego2048.es/>

Fuente de esta información: [https://es.wikipedia.org/wiki/2048\\_\(videojuego\)](https://es.wikipedia.org/wiki/2048_(videojuego))

El programa tendrá un menú principal (tipo el IDE de Python) desde el cual se accederán las funcionalidades del programa (lo que el programa hace):

- A- Jugar
- B- Configurar
- C- Ayuda
- D- Acerca de
- E- Salir

Usted puede agregar otras funcionalidades que vayan a mejorar el producto. En la interfaz gráfica ponga atención a los diferentes elementos como son los tamaños de letras, colores, formas, menús, botones, cuadros de texto, etc. Puede hacer cambios a la interfaz siempre y cuando cumpla con los requerimientos del programa que se indican seguidamente.

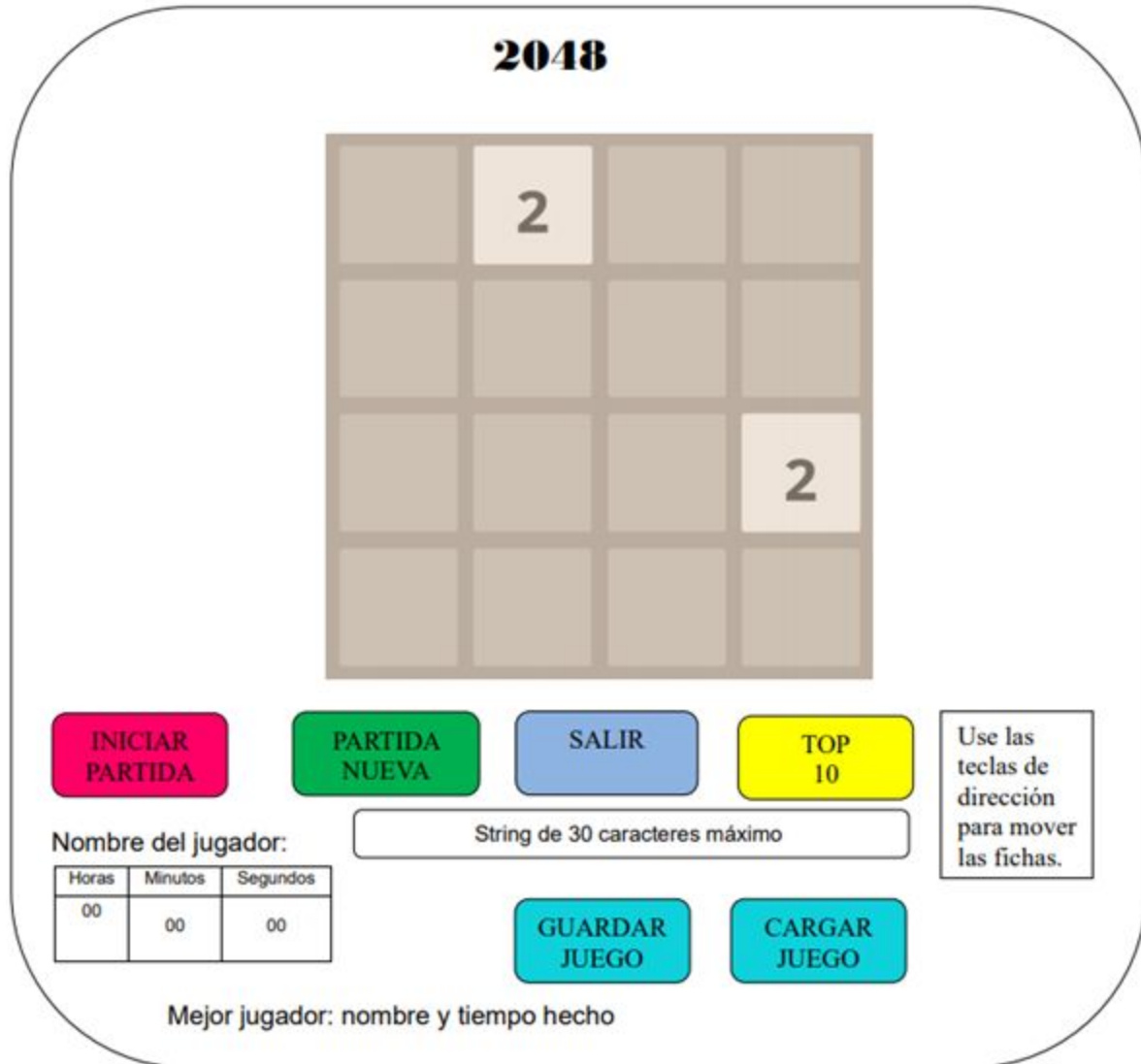
Esta versión a implementar usará tiempos para registrar los mejores jugadores en lugar de llevar puntajes.

Tanto para el tablero como para cada número (2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048) debe asignar un color diferente. En las imágenes anteriores vea que las casillas con diferentes números tienen diferentes colores.

## FUNCIONALIDADES DEL PROGRAMA

### A) Jugar

Esta opción permite jugar el 2048. Cuando escoge esta opción se muestra una pantalla como la siguiente según la configuración del juego.





Uso de las opciones por medio de botones:



Cuando el jugador pica este botón se inicia el juego. Note que antes de iniciar la partida ya se han colocado dos números en forma aleatoria dentro del tablero: un 2 y otro 2 o un 2 y un 4.

A partir de aquí se deben usar solamente las teclas de las flechas direccionales ( flecha izquierda, flecha derecha, flecha arriba, flecha abajo) para hacer los movimientos del juego tal como se explicó anteriormente. Aparte de la flechas se permite usar los botones de opciones.

Para esta versión el juego va a terminar bajo estas condiciones:

- Cuando el jugador no pueda hacer movimientos, es decir, no se ha formado el 2048 y no hay casillas disponibles para juntar con otras o para que el juego ponga un número más. En este caso debe enviar un mensaje avisando de la situación y luego vuelve a una nueva partida.
- Cuando el jugador forma una casilla con el número 2048. Desplegar un mensaje similar a ¡ EXCELENTE ! JUEGO COMPLETADO. Ponga además algún sonido (por ejemplo: aplausos, una canción, etc.). En caso necesario también hay que detener el reloj o el timer.

También determine si este jugador debe registrarlo en el Top 10. El Top 10 es un registro de las mejores 10 marcas (los jugadores que tarden 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. 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 timer hay que calcular la duración del juego. Cuando un juego es completado el programa regresa a la opción de Jugar. El Top 10 hay que guardarlo en un archivo para mantener la información de manera permanente.

- Cuando se está usando timer y termina el tiempo asignado. 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 LA MISMA PARTIDA (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.

Otras consideraciones:

- Para iniciar el juego el jugador debe dar un nombre (string de 30 caracteres máximo, mínimo 1 caracter).
- Luego de dar el botón INICIAR PARTIDA, este botón se deshabilita.
- En caso de haber configurado la opción de Timer, el jugador puede dejar el tiempo registrado en la opción CONFIGURAR o modificarlo antes de seleccionar el botón INICIAR PARTIDA. El tiempo empieza a correr cuando le den INICIAR PARTIDA. NOTA: cuando se usa el Timer o el reloj este debe mantenerse actualizado en tiempo real en el monitor. Hacer las validaciones indicadas para establecer un Timer.
- En caso de no usar el reloj o el timer no debe aparecer en la pantalla..

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.



Cuando el jugador selecciona esta opción se le pregunta

¿ ESTA SEGURO DE TERMINAR LA PARTIDA ACTUAL (SI o NO) ?

Si responde SI termina de inmediato la partida y se vuelve a mostrar otra como si estuviera entrando a la opción de Jugar.

Si responde NO sigue jugando con la misma partida.

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.

SALIR

Cuando el jugador selecciona esta opción se le pregunta

¿ ESTA SEGURO DE TERMINAR EL JUEGO (SI o NO) ?

Si responde SI el programa termina.

Si responde NO sigue jugando con la misma partida.

TOP  
10

Esta opción se puede usar en cualquier momento. Detiene el reloj si se está usando. Despliega una pantalla con los marcas de los mejores 10 primeros jugadores: aquellos que hicieron menos tiempo para completar el juego. En caso de no tener los 10 jugadores se despliegan los que se tengan. El Top 10 se guarda en el archivo "2048top10.dat".

### **TOP 10**

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

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



**GUARDAR  
JUEGO**

Este botón se puede usar en cualquier momento que el juego haya iniciado. El reloj se detiene. Guarda en el archivo "2048juegoactual.dat" el juego actual (cuadrícula) con su configuración (uso del reloj, etc.) y nombre del jugador. El objetivo es que el jugador pueda en cualquier momento guardar el juego y posteriormente continuarlo. Este archivo solo va a contener la última partida. En caso de que haya una partida en el archivo, se borra y se guarda la del momento. Luego de guardarlo se le pregunta al usuario:

¿ VA A CONTINUAR JUGANDO (SI o NO) ?

Si responde SI sigue jugando la misma partida.

Si responde NO regresa al menú principal.

**CARGAR  
JUEGO**

Este botón se puede usar solamente cuando un juego no se haya iniciado. Trae del archivo "2048juegoactual.dat" 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 PARTIDA.

### B) Configurar

Esta opción es para indicar las condiciones con que se va a empezar a jugar. Contiene el siguiente dato que se va a guardar en el archivo "2048configuración.dat" : (el valor por omisión –o default- está señalado con el círculo en rojo)

1. Reloj: ☒ Si  
☐ No  
☐ Timer

Horas	Minutos	Segundos
0	30	0

Para el timer las horas pueden estar entre 0 y 99, 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.

2. Desplegar mejor jugador: ☐ Si  
☐ No

Esta opción se usa para desplegar o no en la pantalla Jugar los datos del jugador que ha hecho el menor tiempo en completar el juego.

**C) Ayuda**

Esta opción desplegará el manual del usuario.

**D) Acerca de**

Esta opción la usaremos 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.

**E) Salir**

Esta opción se usa para salir del programa.

# Importancia de la documentación de software

## Tipos de documentación e importancia del software

La documentación de software se divide en dos partes:

- **Interna:** Es la documentación que se crea dentro del mismo código, ya sea en forma de comentarios o en archivos dentro de la aplicación.
- **Externa:** Es un tipo de documento que se escribe por fuera del programa.

En la documentación se escribe la definición de los requerimientos del software en general, especificaciones generales del sistema y los planes de mantenimiento.

## Importancia de la documentación

La documentación a la hora de programar es importante porque contiene la información del programa creado. Esta información permite que se reproduzca el software o se quiera adaptar el programa posteriormente. Esta información es de mucho valor para la compañía de software. Esta información es necesaria debido a que los desarrolladores pueden cambiarse de compañía y los próximos desarrolladores no podrían entender el programa.

## Programación por eventos.

La programación orientada a eventos es un paradigma de programación en donde se ejecutan ciertas acciones conforme se van determinando por el usuario o por el propio sistema. La programación orientada a objetos es una manera fácil de programar y se pueden crear aplicaciones sencillas y eficaces. Este paradigma de programación utiliza interfaces gráficas en donde se colocan los controles que permite

manejar el sistema. Las acciones dirigidas por el usuario realiza acciones lógicas para un determinado programa.

## ¿Qué son los eventos?

Cuando un evento se reproduce permite que el usuario realice ciertas acciones que se ha programado en su interior. Algunas acciones que se pueden realizar en un programa son:

- Clic sobre un botón
- Doble clic sobre el nombre de un fichero para abrirlo
- Arrastrar un icono
- Pulsar una tecla o una combinación de teclas
- Elegir una opción de un menú
- Escribir en una caja de texto
- O simplemente mover el ratón

## Uso de interfaz gráfica de usuario (GUI: Graphical User Interface) en el desarrollo de software.

En inglés, *graphical user interface* es una interfaz donde el usuario interactúa con dispositivos electrónicos como ordenadores, dispositivos portátiles y otros dispositivos. De igual forma son intermediarios entre el programa y el usuario. Esta interfaz contiene iconos, menús, botones, imágenes y cualquier representación gráfica, con el objetivo de crear un entorno visual amigable con el usuario y fácil de usar según el sistema operativo. Es ideal que estas relaciones sean directas con el usuario para mejorar la experiencia del programa en conjunto con el sistema operativo.



## Desarrollo de GUI con tkinter

Tkinter es una adaptación de la biblioteca gráfica Tcl/Tk para el lenguaje de programación de Python. Esta librería de python se usa para crear interfaces gráficas (GUI) en python. Además proporciona una amplia gama de usos, como aplicaciones web, de escritorio, redes, y administración. Esta GUI, viene por defecto con la instalación para Microsoft Windows. Al correr `python -m tkinter` desde la consola se abre una ventana que muestra una interfaz simple de TK, demostrando que tkinter está instalado en el sistema. Además muestra que versión de Tcl/Tk está instalada y así se puede leer la documentación específica de esa versión.

## Herramientas de depuración (debugger) en programación.

Las herramientas de depuración es un proceso para verificar los errores en un programa y eliminarlos. Este método puede realizarse tanto manual como con herramientas de depuración para comprobar el correcto funcionamiento de un programa. Estas verificaciones se pueden clasificar en:

- Verificaciones del programador “manuales”: mental, seguimiento escrito, con tablas.
- Verificaciones con ejecución: seguimiento con desarrollo en un lenguaje.
- Verificación con medios de depuración del lenguaje.

Existen diferentes métodos de depuración de acuerdo al lenguaje. Además de verificar los errores manualmente, se puede establecer puntos de interrupción dentro del código al escribir algunas variables que se estén utilizando de forma que se puede ver paso a paso qué es lo que está ocurriendo en nuestro programa.

# Explicación de las funciones del “debugger” del IDE que está usando.

## The Debugger

Es difícil encontrar que está causando un error en el código. Las líneas de código se ejecutan rápido y las variables están en constante cambio. El debugger del IDLE de Python, permiten detener el programa paso a paso para ir evaluando línea por línea.

## Stepping

El stepping permite que se ejecute una instrucción a la vez. Esta opción se encuentra en ventana Debug.

## Globals Area

En la parte donde dice Debug control window es donde todas las variables globales se pueden ver.

## Locals Area

También hay una sección para las variables locales que muestra el alcance que tienen estas variables y sus valores. El área local sólo tendrá en él cuando el programa de ejecución está dentro de la función. Cuando la ejecución está en el alcance global esta área se muestra en blanco.

## The Go and Quit Buttons

Si la persona se cansa de utilizar el botón de **Step** repetidamente y solamente quiere correr el programa normalmente, puede presionar el botón de **Go** en la parte de arriba de la ventana de control del Debug. Lo que hará que el programa corra normalmente sin ir paso a paso. Posteriormente, para terminar el programa completamente solo se presiona el botón **Quit** que se encuentra en la parte inferior de la ventana de control del Debug y el programa se cerrará de inmediato.

## Break Points

. Por ejemplo, si se piensa que se tiene un error en la línea 34 se establece un punto de interrupción en esa línea o antes de llegar a ese punto.

## Comandos de Tkinter utilizados:

### Import \*

La primera línea de tkinter El “import \*” se refiere a que se quiere importar el módulo de tkinter y usar todo (\*) lo que este contenga. La línea que dice Tk() se refiere a que se quiere crear una ventana Top level.

### mainloop():

Este método se usa cuando ya se está listo para que se use la aplicación. Es un loop infinito usado para correr la aplicación. Espera para que ocurra un evento y procesa el evento hasta que la ventana se cierra.

### 1. Widgets

El botón widget implementa varios tipos de botones. Estos pueden contener texto, imágenes y se puede asociar una función de Python o un método con estos botones. Cuando el botón es presionado por el usuario Tkinter llama automáticamente a la función correspondiente

- **Button:** Se usan para poner botones en una ventana de tkinter.
- **Entry:** Se usa para que el usuario ingrese datos en la ventana.
- **Frame:** Un marco es un rectángulo en la pantalla que se usa principalmente para colocar otros widgets como botones sobre él.
- **Label:** Los label se utilizan para insertar imágenes o texto en la pantalla, etc.

## 2. Geometry Management

- **Grid()**: Organiza los widgets en forma de tabla como la matriz para el juego. .
- **Place()**: Se utiliza para posicionar los widget en un lugar específico según x y y.

## 3. Organización de Diseño y Widgets

- **Button**: El botón creado en la ventana puede contener muchos parámetros como: **text**(Value of the Button), **fg**(Color of the text), **bg**(Background color), etc.

## 7. Funciones de Binding

- **Binding**: Mediante las teclas direccionales y a través de programación por eventos, cada vez que se presiona una tecla se llama a una función.

## 8. MessageBox y mensajes de alerta

- **MessageBox**: Se utilizan para hacerle preguntas al usuario y a partir de su respuesta, ejecutar ciertas funciones.

## 9. Diccionarios

Los diccionarios son una estructura de datos y además son un tipo de dato con características particulares que permiten conservar datos de cualquier tipo; enteros, cadenas, listas o incluso otras funciones. Estos elementos se identifican por una llave. También es posible insertar una lista dentro de un diccionario.

## 10: PhotoImage:

PhotoImage se utiliza para desplegar imágenes a color o a escala de grises en labels, buttons y widgets.

## 11. os.system()

El módulo OS provee funciones para interactuar con el sistema operativo. OS, viene con el standard de módulos de python. Este módulo provee una forma portable de usar el operating system dependiendo de la funcionalidad.

## 12. Destroy()



Destroy() destruye la ventana en la cual se encuentra la función.

### **13. Withdraw**

Elimina la ventana de la pantalla.

### **14. Iconify**

Minimiza la ventana o la convierte en un ícono

### **15. deiconify**

El método deiconify() volverá a dibujar la ventana y / o la activará.

### **16. Top level**

El widget de Toplevel se utiliza para desplegar ventanas de aplicación extra, diálogos y otras “pop-up” ventanas.

### **17. Normal**

Normal, habilita un botón para que pueda ser utilizado por el usuario

### **18. Disable**

Deshabilita el botón para que el usuario no lo utilice

### **19. After:**

Llama una función cada cierto tiempo

### **20. Radio Botones:**

Es un elemento de la interfaz que permite al usuario elegir entre diferentes opciones. Estos botones pueden contener imágenes o texto

# Conclusiones

## **Problemas encontrados y soluciones**

Como parte de los problemas encontrados fue a la hora de elegir los colores para el juego. Al principio se quería hacer como un degradado de colores pero a la hora de proyectar todos los números, estos se parecían mucho o no se visualizaba bien el número. Por lo que se optó por utilizar colores diferentes que fueran mejor visualizados. Además en principio al realizar la función de sumar, se pensaba hacer con una matriz transpuesta, sin embargo esta funcionaba para sumar solamente para arriba y para abajo, por lo que se optó por crear diferentes funciones que pudieran sumar para todos lados.

## **Aprendizaje Obtenido**

Al finalizar este proyecto se aprendió cómo funciona la programación orientada a eventos, la importancia de la documentación de software y el uso de interfaces gráficas. Asimismo, se comprendió de una forma más amplia la aplicación de varias funciones para crear un proyecto. Así como al llamar varias funciones, ya sea con for, if o con while se puede realizar una acción específica.

Por otra parte, al realizar el juego se comprendió cómo funciona la interfaz gráfica llamada Tkinter, además de todos sus módulos y comandos utilizados en la elaboración de este proyecto.

Actividad realizada	Horas
Análisis del problema	3
Diseño de algoritmos	14
Investigación de funcionamiento de tkinter y funciones	8
Programación	40
Documentación interna	2
Pruebas	6
Elaboración manual de usuario	2
Elaboración documentación del proyecto	6
<b>Total</b>	<b>81</b>

# Tabla Evaluaciones

Concepto	Puntos	Puntos obtenidos	Avance 100/%/0	Análisis de resultados
Menú principal	1		100	
Desplegar juego inicial (fondo, cuadrícula, reloj, otros datos)	4		100	
Uso de colores por número	4		100	
Uso de teclas de flechas	4		100	
Movimientos (izquierda, derecha, arriba, abajo) y su despliegue	9		100	
Juntar casillas y su despliegue	9		100	
Poner nuevo número en cuadrícula	3		100	
Reloj o timer en tiempo real	10		100	
Terminar juego por falta de casillas libres	5		100	
Terminar juego con 2048	5		100	
Crear Top 10	5			No se realizó por tiempo
Terminar juego por timer	5			
Botón terminar partida	2		100	
Botón terminar juego	2		100	
Botón Top 10	5		20	No se realizó por tiempo
Botón guardar juego	5		100	
Botón Cargar Juego (incluye el despliegue del mismo)	10		100	
Opción configurar	5		80	



Ayuda	5		100	
Acerca de	1		100	
Salir	1		100	
TOTAL	100			
Partes adicionales				

# Referencias

[https://www.python-course.eu/tkinter\\_radiobuttons.php](https://www.python-course.eu/tkinter_radiobuttons.php)

<https://www.datacamp.com/community/tutorials/gui-tkinter-python#tkinter>

<https://unipython.com/tkinter-introduccion/>

<https://desarrolloweb.com/articulos/importancia-documentacion.html>

[https://techlandia.com/importancia-documentacion-software-sobre\\_538552/](https://techlandia.com/importancia-documentacion-software-sobre_538552/)

<http://contenidos.sucerman.com/nivel3/dispositivos/unidad1/leccion2.html>

<https://fp.uoc.fje.edu/blog/que-son-las-interfaces-graficas-o-gui/>

[https://www.aprenderaprogramar.com/index.php?option=com\\_content&view=article&id=352:herramientas-de-depuracion-de-programas-formas-de-verificacion-y-ejecucion-de-tests-testing-cu00244a&catid=36&Itemid=60](https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=352:herramientas-de-depuracion-de-programas-formas-de-verificacion-y-ejecucion-de-tests-testing-cu00244a&catid=36&Itemid=60)

<https://inventwithpython.com/chapter7.html>

[https://www.tcl.tk/man/tcl8.6/TkCmd/ttk\\_widget.htm#M3](https://www.tcl.tk/man/tcl8.6/TkCmd/ttk_widget.htm#M3)

<https://devcode.la/tutoriales/diccionarios-en-python/>

<https://effbot.org/tkinterbook/photoimage.htm>

<https://www.geeksforgeeks.org/python-os-system-method/>

<https://coday.me/es/qa/20190527/767076.html>