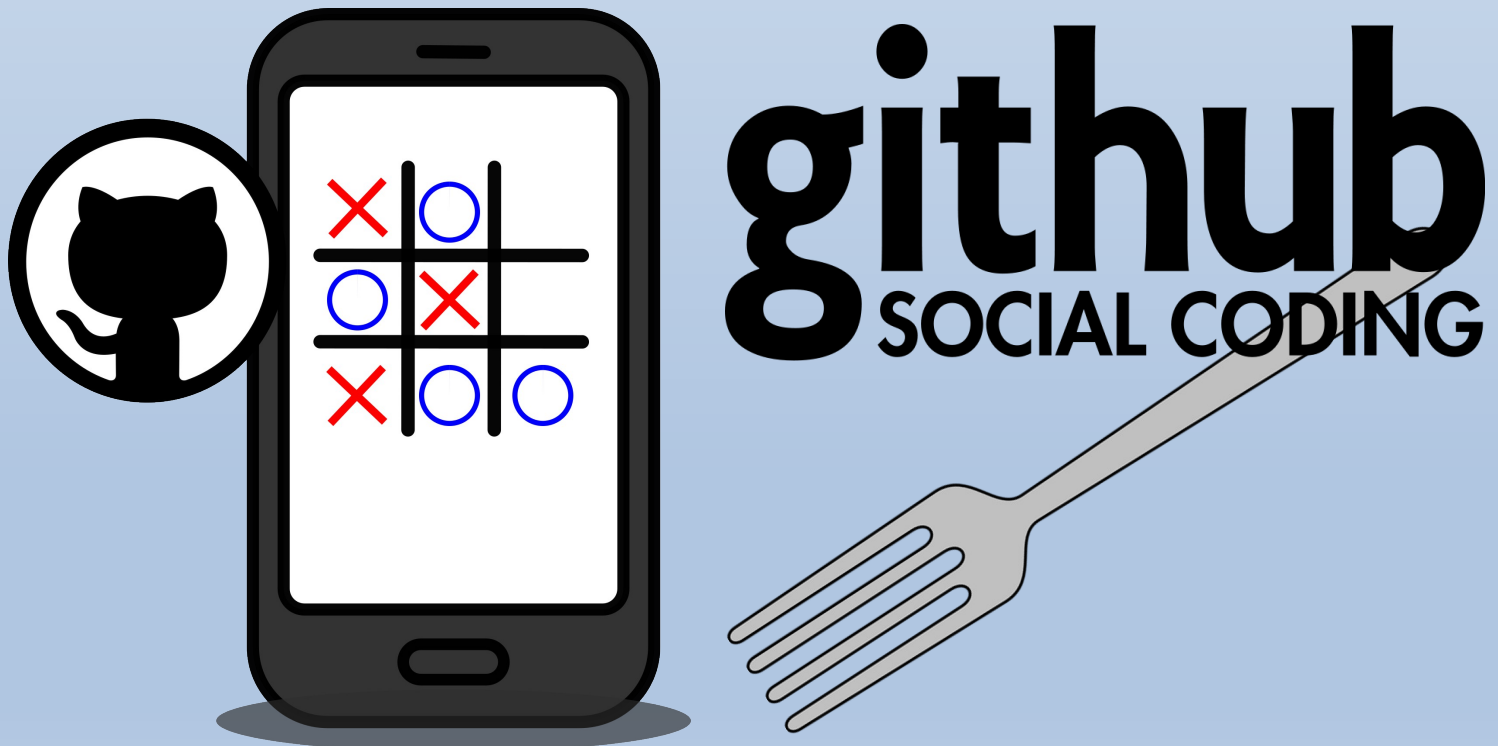


OCTUBRE DE 2025

Por Eden Lattibeaudiere (BA, MSc)

Introducción a la Inteligencia Artificial, Python, GitHub y Copilot

Desarrollar un juego de tres en raya para dos jugadores basado en Python con una mejora opcional de la IA como oponente y distribuirlo a través de GitHub.



<https://github.com/skelphx/thecloud.git>

Tabla de contenido

Capítulo 1 - Introducción	Página 2
Capitulo 2.0 – ¿Qué es Python?	Página 3
Capitulo 2.1 – Conceptos básicos de Python	Página 4
Capitulo 2.2 – ¿Qué es GitHub?	Página 5
Capitulo 3 – Estructura del juego	Página 6
Capitulo 4.0 – clase Tres en raya	Página 7
Capitulo 4.1 – Construcción del programa (PP vs. OOP)	Página 8
Capitulo 5.0 – tictactoe.py - Repositorio de GitHub Crear/Editar archivo	Página 9
Capitulo 5.1 – importar, __init()_, display_board()	Página 10
Capitulo 5.2 – es_movimiento_válido(), hacer_movimiento(), comprobar_ganador(), cambiar_jugador()	Página 11
Capitulo 5.3 – principal(), obtener_movimiento(), jugar_auto()	Página 12
Capitulo 6 – Invocar el intérprete de Python (Windows)	Página 13
Capitulo 7 – Función get_move() - Mejora del oponente de la IA	Página 14
Capitulo 8 – Consejos de estrategia para el tres en raya	Página 15
Capitulo 9.0 – Configuración de la cuenta y el repositorio de GitHub.	Página 16
Capitulo 9.1 – Configuración del repositorio de GitHub / Acceso a Copilot	Página 17
Capitulo 9.2 – GitHub Copilot / Repositorio de GitHub Agregar archivo	Página 18
Capitulo 9.3 – Ventana de diálogo para agregar archivos del repositorio de GitHub	Página 19
Capitulo 9.4 – Distribución del repositorio de GitHub	Página 20
Capitulo 10 – Resumen	Página 21

Capítulo 1

Introducción

Este tutorial ofrece una introducción sencilla al mundo de la programación en Python, la inteligencia artificial, GitHub y Copilot. Es apto tanto para principiantes como para entusiastas de nivel intermedio y profesionales avanzados. ¡Diviértete!

Construirás:

- (i) Un juego completo para dos jugadores.
- (ii) Una versión con un **oponente IA**.
- (iii) Un proyecto que puedes cargar y compartir con el mundo a través de **GitHub**.

Capítulo 2.0

¿Qué es Python?

Introducción a Python

Python es un lenguaje de programación multiplataforma de código abierto que ha ganado popularidad en los últimos diez años. Se lanzó por primera vez en 1991 y la última versión es la 3.13.7. CPython es la implementación de referencia del lenguaje de programación Python escrito en C. CPython es la implementación predeterminada y más utilizada de Python.

Python también es multipropósito (gracias a sus numerosas y versátiles extensiones). Entre los ejemplos de extensiones de Python y su uso se incluyen diversas implementaciones en computación y cálculos científicos, simulaciones, desarrollo web, inteligencia artificial y videojuegos. Python se considera actualmente el lenguaje de programación más popular del mundo.



Capítulo 2.1

Conceptos básicos de Python

Conceptos cubiertos:

(i) Variables.

(ii) Listas y bucles.

(iii) Funciones.

(iv) Clases.

(v) Entrada/Salida.

(vi) Lógica de IA simple..

(vii) Conceptos básicos de GitHub.

Capítulo 2.2

¿Qué es GitHub?

Introducción a GitHub

GitHub es una plataforma en la nube que permite a los desarrolladores almacenar, gestionar y colaborar en código mediante Git, un sistema de control de versiones creado por Linus Torvalds, creador del sistema operativo Linux. GitHub facilita la colaboración en proyectos para equipos de todos los tamaños, proporcionando herramientas para el seguimiento de cambios, la revisión de código, el control de versiones de software y la integración de innumerables herramientas y servicios de desarrollo en la plataforma.

Tanto si trabajas solo como si formas parte de un gran proyecto de código abierto, GitHub te ayuda a mantener tu código organizado y tu flujo de trabajo eficiente. Además, GitHub es gratuito.



Capítulo 3

Estructura del juego

Desarrollarás el juego usando una clase de Python llamada `TicTacToe` y guardarás la clase `TicTacToe` y sus funciones en un script llamado `tictactoe.py`. La extensión de archivo `.py` indica que es un script de Python específicamente diseñado para su uso con el intérprete de Python.

Se encarga de:

- (i) Estado del juego (tablero, turno, ganador).
- (ii) Lógica de visualización.
- (iii) Funciones.
- (iv) Entrada del jugador.
- (v) Comprobación de victorias/empates.

Capítulo 4.0

clase Tres en raya

Al revisar el script de Tres en raya, tenga en cuenta que la palabra clave "class" de Python crea una clase y la palabra clave "def" crea una función.

Una clase de Python tiene una estructura similar a la de un capítulo de un libro, donde el libro en sí es el script. Metafóricamente hablando, una función podría describirse como el vocabulario utilizado para presentar un concepto particular al lector con el fin de ayudarlo a cumplir un propósito o requisito específico como resultado directo de la lectura del libro. Cada función está contenida dentro de un bloque gris para fines de presentación. Las funciones están precedidas por una línea comentada, instanciada con el símbolo "#" y un número que identifica su posición en la clase Tres en raya, seguida de una breve explicación de su sintaxis.

El script del Tres en raya se compone de una sola clase:

clase Tres en raya

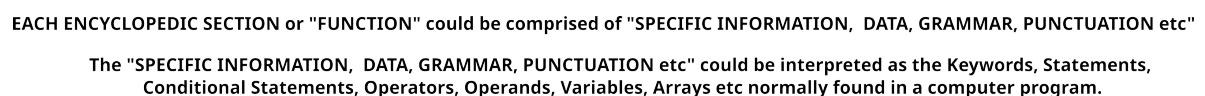
Y nueve funciones:

`__init__()`, `display_board()`, `is_valid_move()`, `make_move()`, `check_winner()`, `switch_player()`, `get_move()`, `play()` y `main()`.

Construcción del programa (PP vs. OOP)

OBJECT REFERENTIAL/ RELATIONAL INTERGRITY CONSTRAINT NOTATION

M:M = many-to-many relationship



A large, stylized comic book speech bubble with a jagged, starburst-like border. Inside the bubble, the word "OOPS!" is written in large, bold, red capital letters with a thick black outline. The background of the bubble is a light beige color with a subtle pattern of small dots. Several small red stars are scattered around the bubble, and black lines radiate from the edges, giving it a dynamic, explosive appearance.

Capítulo 5.0

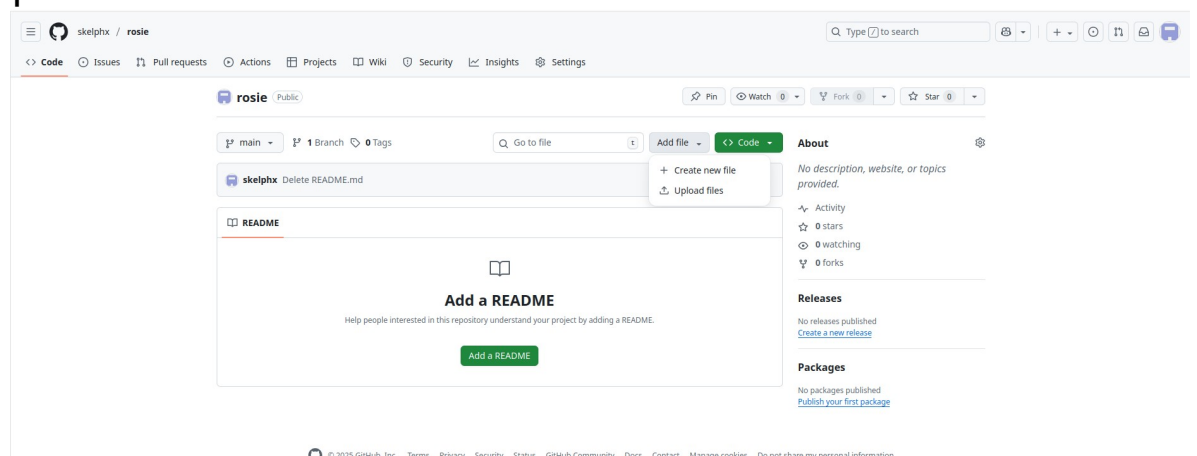
tictactoe.py

Siga cada paso de este tutorial para crear **tictactoe.py**.

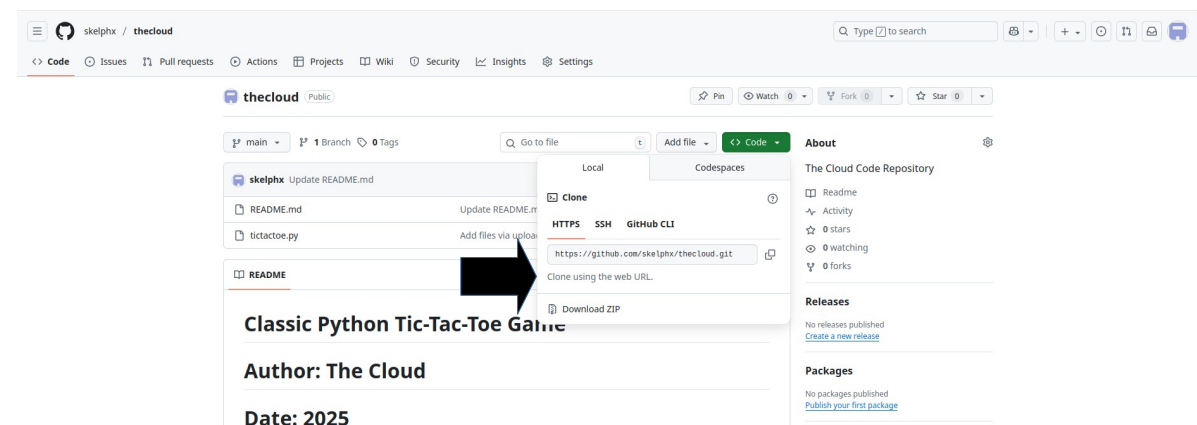
(i) Revise el Capítulo 9 y cree un repositorio de GitHub, etc.

(ii) Haga clic en Agregar archivo y luego en Crear nuevo archivo (ver captura de pantalla a continuación).

(iii) Ingrese los fragmentos de código que aparecen en las siguientes páginas, nombre el archivo tictactoe.py y luego presione Confirmar cambios.



(iv) Alternativamente, puedes descargar el archivo zip completo del tres en raya desde <https://github.com/skelphx/thecloud.git>.



Capítulo 5.1

Tres en raya

Nota: No incluya números de línea; son solo marcadores de posición. Introduzca únicamente el pseudocódigo con el fondo gris y repita el proceso para cada bloque adicional de la secuencia. Esto se conoce como programación procedimental: Bloque 1, Bloque 2, Bloque 3, etc.

Bloque 1

```
1  #Import os and sys Standard Python Library Modules
2
3  import os
4  import sys
5
6  #Initialize class TicTacToe
7
8  class TicTacToe:
9
10     #1 __init__() allocates nine blank spaces to self.board variable, sets current player to X and
11     starts a new game where the winner is nullified.
12
13     def __init__(self):
14         self.board = [' '] * 9
15         self.current_player = 'X'
16         self.game_over = False
17         self.winner = None
```

Bloque 2

```
1  #2 display_board() clears screen, prints title, columns separated by bars, rows separated by hyphenated separators
2  of the same length.
3
4  def display_board(self):
5      os.system('cls' if os.name == 'nt' else 'clear')
6      print("\nTic-Tac-Toe\n")
7      for i in range(3):
8          row = ""
9          for j in range(3):
10             pos = i * 3 + j
11             row += f" {self.board[pos]} "
12             if j < 2:
13                 row += "|"
14             print(row)
15             if i < 2:
16                 print("-----")
17             print()
```

Capítulo 5.2

Bloque 3

```
1 #3 is_valid_move() if position is empty and between >=0 & <=9 it is a valid position.
2
3 def is_valid_move(self, position):
4     try:
5         pos = int(position) - 1
6         return 0 <= pos <= 8 and self.board[pos] == ' '
7     except ValueError:
8         return False
```

Bloque 4

```
1 #4 make_move -1 from positions and play.
2
3 def make_move(self, position):
4     pos = int(position) - 1
5     self.board[pos] = self.current_player
```

Bloque 5

```
1 #5 check_winner() after move check winning combination and that all board spaces are not ' ', if check proves
2 true and there is a winner or a draw, end game is True, if not end game if False.
3
4 def check_winner(self):
5     combos = [
6         [0,1,2],[3,4,5],[6,7,8],
7         [0,3,6],[1,4,7],[2,5,8],
8         [0,4,8],[2,4,6]
9     ]
10    for combo in combos:
11        if self.board[combo[0]] == self.board[combo[1]] == self.board[combo[2]] != ' ':
12            self.winner = self.board[combo[0]]
13            self.game_over = True
14            return True
15        if ' ' not in self.board:
16            self.winner = 'Draw'
17            self.game_over = True
18    return True
19    return False
```

Bloque 6

```
1 #6 switch_player() switches to other player.
2
3 def switch_player(self):
4     self.current_player = 'O' if self.current_player == 'X' else 'X'
```

Capítulo 5.3

Bloque 7

```
1 #7 get_move() allow move to positions 1-9 if 0 positions, else if move is to invalid
2 position display error.
3
4 def get_move(self):
5     while True:
6         move = input(f"Player {self.current_player}, choose (1-9): ")
7         if move.lower() in ['q', 'quit', 'exit']:
8             sys.exit("Thanks for playing!")
9         if self.is_valid_move(move):
10             return move
11         print("Invalid move. Try again.")
```

Bloque 8

```
1 #8 play() starts game then only when playing, call display_board(), get_move(),
2 make_move(), check_winner(), switch_player and display_board() again, if
3 check_winner() indicates a winner, exit, print and concatenate Result: with
4 winning player.
5
6 def play(self):
7     input("Press Enter to begin...")
8     while not self.game_over:
9         self.display_board()
10        move = self.get_move()
11        self.make_move(move)
12        if self.check_winner():
13            break
14        self.switch_player()
15    self.display_board()
16    print(f" Result: {self.winner}!")
```

Bloque 9

```
1 #9 main() if game is won, decide whether or not to play again.
2
3 def main():
4     while True:
5         game = TicTacToe()
6         game.play()
7         again = input("Play again? (y/n): ").lower()
8         if again != 'y':
9             Break
10    if __name__ == "__main__":
11        main()
```

Capítulo 6

Invocar el intérprete de Python

La forma más sencilla de invocar el intérprete de Python en cualquier versión moderna del sistema operativo Windows es hacer clic en Inicio de Windows y, en la barra de búsqueda, escribir CMD (Símbolo del sistema de Windows).

Hacer clic **Run as Administrator**.

Descarga el archivo ZIP (Capítulo 5) desde tu repositorio de GitHub a una carpeta en tu ordenador Windows y extráelo. Escribe CD\directorio (sustituye directorio por el nombre de la carpeta donde se extrajo el archivo).

Nuevamente usando CMD, ejecute el siguiente comando:

python tictactoe.py

El intérprete de Python debería ejecutar el script y mostrar la pantalla del juego Tres en raya, como se muestra a continuación. Si el juego no se inicia y el intérprete devuelve errores, compruebe que Python esté correctamente instalado y que la variable de ruta esté configurada. A continuación, examine el script de Python para detectar errores de sintaxis.



```
Tic-Tac-Toe
| |
| |
| |
| |
| |
| |
| |
| |
| |
Player X, choose (1-9):
```

Capítulo 7

get_move() - Mejora del oponente de la IA

Compare las diferencias entre cada función `get_move()` para comprender mejor las construcciones de programación, la sintaxis, las palabras clave, los operadores y las sentencias. La actualización de `get_move()` que se muestra a continuación es opcional.

Bloque 7

```
1 def get_move(self):
2     if self.current_player == 'X':
3         while True:
4             move = input("Your move (1-9): ")
5             if move.lower() in ['q', 'quit', 'exit']:
6                 sys.exit("Game exited.")
7             if self.is_valid_move(move):
8                 return move
9             print("Invalid. Try again.")
10    else:
11        available = [str(i+1) for i, val in enumerate(self.board) if val == ' ']
12        move = random.choice(available)
13        print(f"AI chooses: {move}")
14    return move
```

Crea un nuevo archivo en tu repositorio de GitHub (consulta el Capítulo 5). Introduce los fragmentos de código de los Capítulos 5.1, 5.2 Y 5.3. Reemplaza la función `get_move()` del Capítulo 5.3, Bloque 7, por la función `get_move()` upgrade, que se muestra arriba en fondo gris. No incluyas los números de línea. Agrega `import random` debajo de `import os` e `import sys` en el encabezado del script. Guarda el archivo como `tictactoeai.py` y continúa con el Capítulo 6. Invoca el intérprete de Python y ejecuta `tictactoeai.py`. Realiza pruebas de errores hasta que `tictactoeai.py` se compile correctamente y se muestre la pantalla del juego de la IA del Tres en Raya. ¿Podrás vencer al oponente de la IA?

Capítulo 8

Consejos de estrategia para el tres en raya

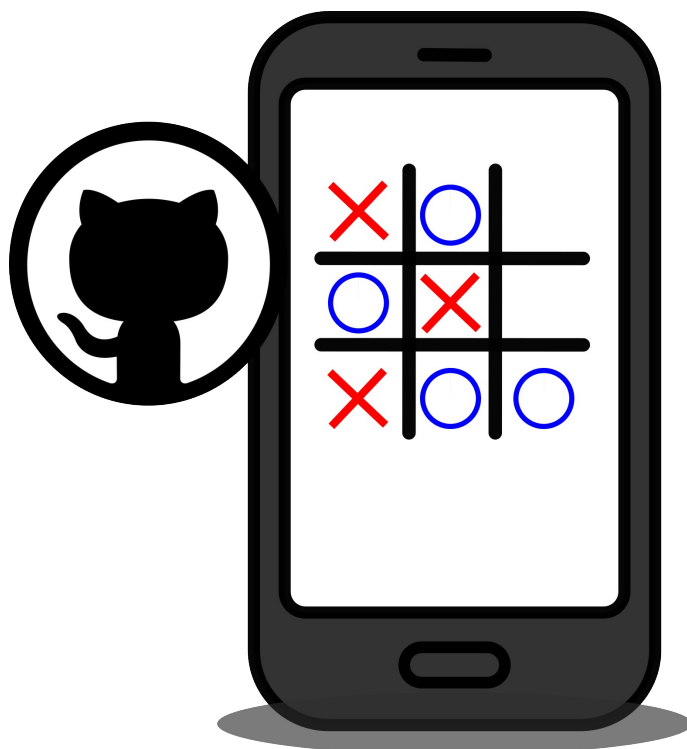
Regla 1: Controla la casilla central.

Regla 2: Bloquea la jugada ganadora del oponente.

Regla 3: Las esquinas son mejores que los bordes.

Regla 4: Evita darle tenedores al oponente.

Regla 5: No hay cuchara (¡Chiste de Matrix, jaja!).

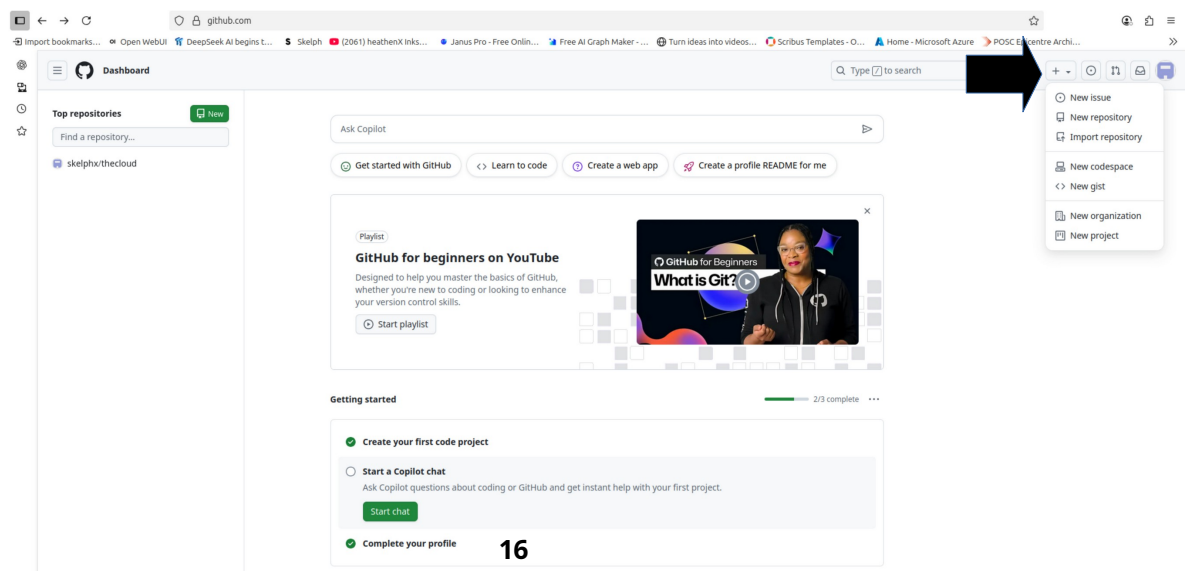


Capítulo 9.0

Configuración de la cuenta y el repositorio de GitHub

(i) Abre tu navegador y escribe `https://github.com/signup` en la barra de direcciones. Introduce tus credenciales en la página "Crear cuenta de GitHub" y haz clic en "Crear cuenta".

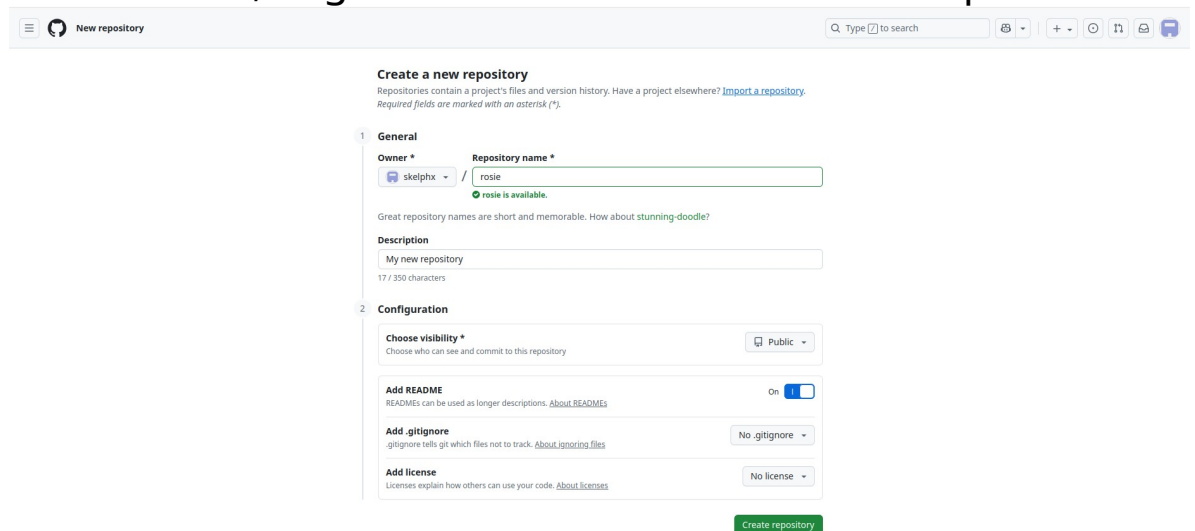
(ii) Haz clic en "+" en el menú principal de GitHub y selecciona "Nuevo repositorio" en el menú desplegable.



Capítulo 9.1

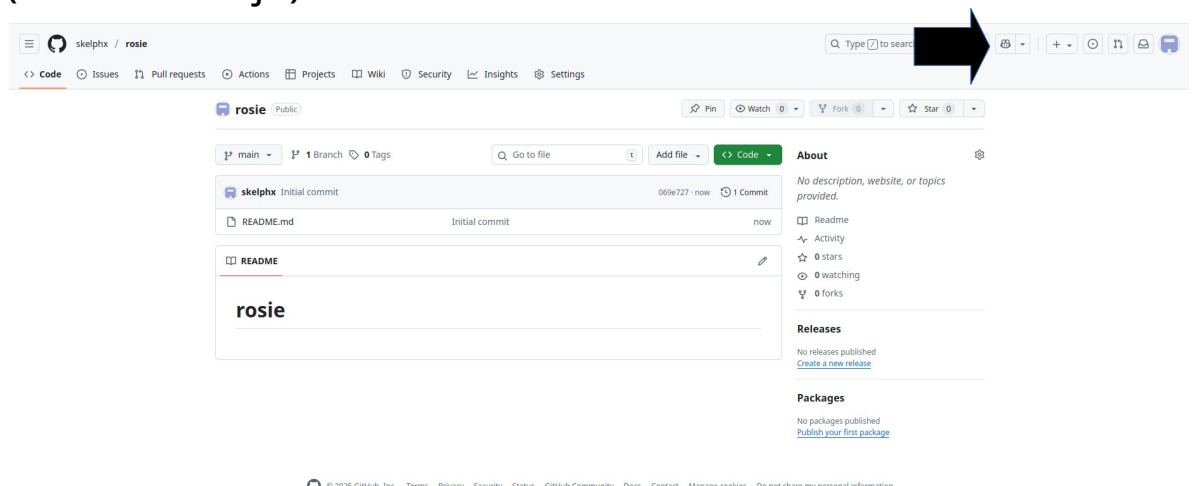
Configuración del repositorio de GitHub

(iii) Seleccione Público (ver ejemplo a continuación), asegúrese de que la opción Agregar README esté activada y, a continuación, haga clic en el botón verde Crear repositorio.



The screenshot shows the 'Create a new repository' page on GitHub. The form is divided into two sections: 'General' and 'Configuration'. In the 'General' section, the 'Owner' is 'skelphx' and the 'Repository name' is 'rosie', with a green checkmark indicating 'rosie is available'. The 'Description' field contains 'My new repository'. In the 'Configuration' section, 'Choose visibility' is set to 'Public', 'Add README' is checked, 'Add .gitignore' is set to 'No .gitignore', and 'Add license' is set to 'No license'. A green 'Create repository' button is at the bottom right.

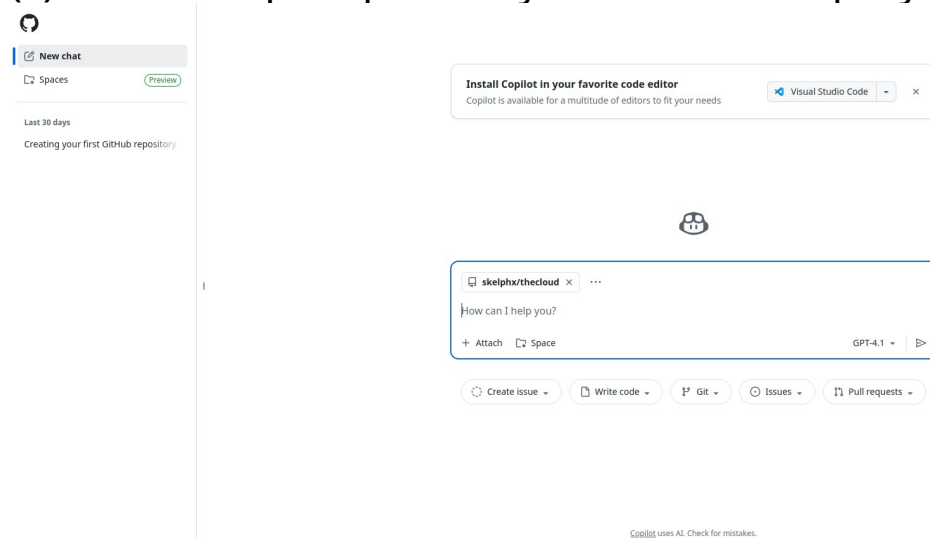
(iv) ¡Excelente trabajo! Has creado correctamente una nueva cuenta de GitHub con un repositorio público donde puedes compartir tus proyectos en internet. Puedes acceder a GitHub Copilot haciendo clic en el icono de GitHub Copilot (ver más abajo).



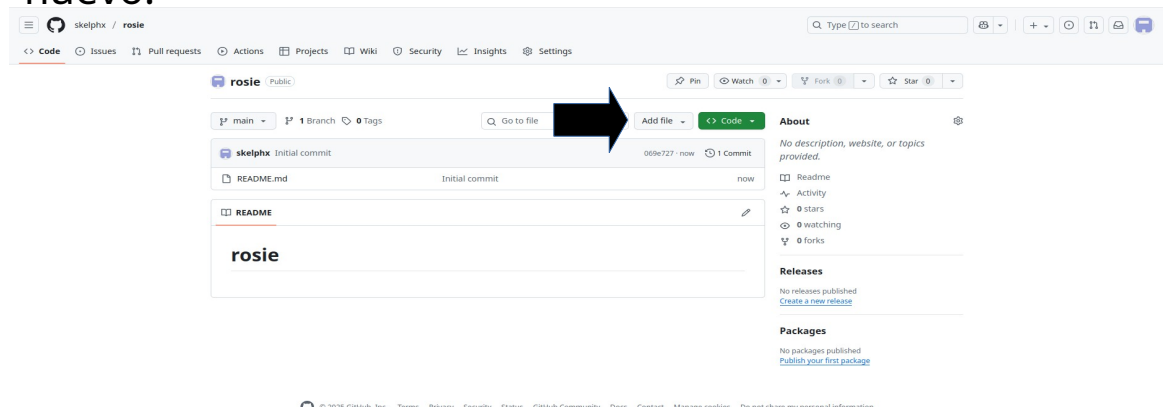
Capítulo 9.2

Agregar archivo al repositorio de CoPilot/GitHub

(v) GitHub Copilot puede ayudarte con tus proyectos de GitHub.



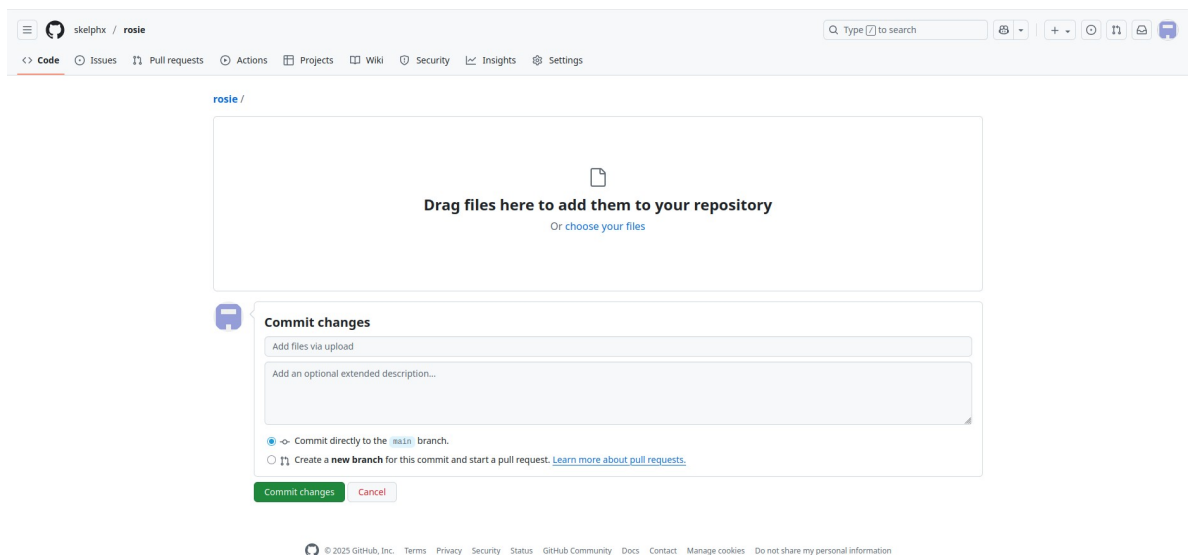
(vi) Haz clic en "Añadir archivo desde tu repositorio" (ver captura de pantalla a continuación) para subir el archivo tictactoe.py que descargaste en el capítulo 5. Deberías ver la ventana "Añadir archivo" en el capítulo 9.3. Los archivos se suben a tu repositorio de GitHub desde aquí. Este paso no es necesario si ya creaste tictactoe.py en el capítulo 5. Para evitar sobrescribir archivos importantes, revisa el contenido de tu repositorio de GitHub antes de subir nada nuevo.



Capítulo 9.3

Ventana para agregar archivos a tu repositorio de Github

(iii) Esta es la página a la que accedes para subir archivos a tu repositorio de GitHub.



Capítulo 9.4

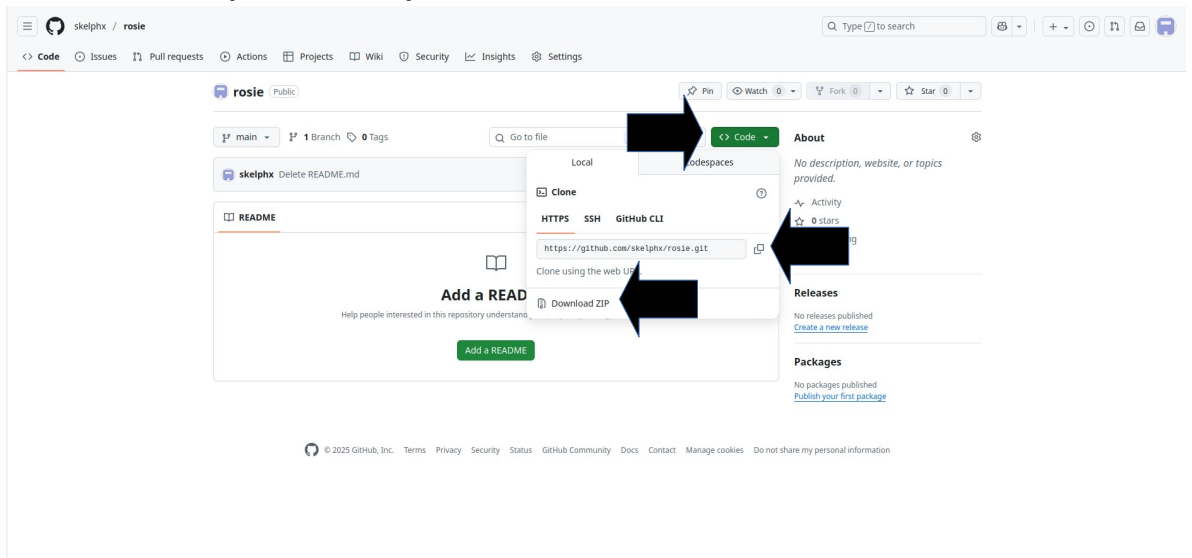
Distribución del repositorio de GitHub

Para distribuir tu juego de Tres en Raya o cualquier otro proyecto en tu repositorio de GitHub, haz clic en el botón verde de código y luego en el botón de copiar a la derecha de la URL que se muestra en la ventana desplegable (ver captura de pantalla).

Tu repositorio tendrá una URL ligeramente diferente a la que se muestra en el ejemplo a continuación.

Pega la URL directamente en correos electrónicos, documentos, mensajes de texto, etc. Al hacer clic en el enlace o copiarlo en la barra de direcciones de tu navegador, se abrirá la página de tu repositorio de GitHub.

El archivo ZIP que contiene los archivos del proyecto en tu repositorio se puede descargar haciendo clic en "Descargar ZIP" (ver captura de pantalla).



Capítulo 10

Resumen

Si completaste el tutorial, ¡bien hecho!

Has cubierto MUCHO.

(i) Sintaxis y terminología básica de programación en Python.

(ii) Desarrolla un juego de tres en raya para 2 jugadores con una mejora de oponente de IA usando el lenguaje de programación Python.

(iii) Cómo invocar el intérprete de Python y ejecutar un script de Python.

(iii) Configurar una cuenta y un repositorio de GitHub.

(iv) Crear y editar un archivo en GitHub.

(v) Subir archivos a un repositorio de GitHub.

(vi) Acceder a GitHub Copilot (Asistente de IA).

(vii) Distribución del repositorio de GitHub.



