

# Manual de Usuario

# Autómatas celulares.

---

**Creado por:**

Christopher Jiménez Gutiérrez

José Manuel Quesada Carvajal

**Profesor:**

Mauricio Avilés Cisneros



## Contenidos

<b>Propósito.</b>	<b>3</b>
<b>Instalación de Python</b>	<b>3</b>
<b>Instalar la biblioteca Pygame</b>	<b>4</b>
<b>Abrir los programas</b>	<b>6</b>
Ejecutarlos abriendolos directamente	7
Abrirlos por medio del Idle de python	8
<b>Cómo utilizar el programa</b>	<b>11</b>
Brian's Brain	12
Autómata celular cíclico:	13
Hormiga de Langton.	14
Modelo de tráfico Biham-Middleton-Levine	15
<b>Tabla resumida</b>	<b>16</b>


## Propósito.

La programación se dedica a resolver muchos de los problemas que se le presentan al ser humano en su vida cotidiana. Por esto, mediante este proyecto se busca estudiar un área muy útil de la programación: la de los autómatas celulares. Los autómatas representan animaciones, y pueden llegar a aplicarse en simular tráfico, vida artificial, diseño de distintas estructuras y explicar varios comportamientos de la naturaleza, de una manera bastante eficaz. En este trabajo, se busca ampliar el conocimiento en este tema, implementando 4 autómatas distintos, reforzando el conocimiento en matrices, manejo de archivos y pygame.

## Instalación de Python

Para poder utilizar los autómatas es necesario instalar Python, ya que es por medio de su shell donde se pueden utilizar las funciones del programa.

Python se descarga en <https://www.python.org/downloads/>. Ya en esa página, se debe presionar el botón amarillo que dice “Download Python”, como se muestra en la siguiente imagen:



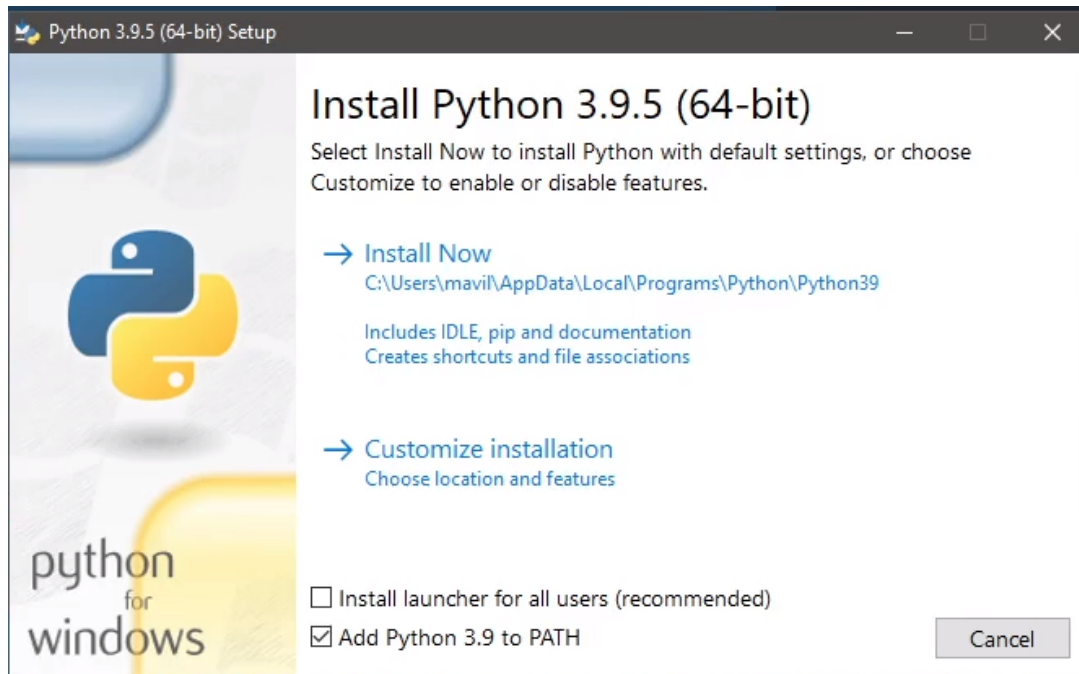
The screenshot shows the Python.org website. The main navigation bar includes links for Python, PSF, Docs, PyPI, Jobs, and Community. Below this is a secondary navigation bar with links for About, Downloads, Documentation, Community, Success Stories, News, and Events. The main content area features a large banner with the text "Download the latest version for Windows" and a prominent yellow button labeled "Download Python 3.9.5". A red circle highlights this button. Below the button, there are links for "Python for Windows", "Linux/UNIX, Mac OS X, Other", "Preleases", and "Docker images". The banner also includes an illustration of two parachutes carrying boxes. Below the banner, there is a section titled "Active Python Releases" with a table listing various Python versions and their support status.

Python version	Maintenance status	First released	End of support	Release schedule
3.9	bugfix	2020-10-05	2025-10	PEP 596
3.8	bugfix	2019-10-14	2024-10	PEP 569
3.7	security	2018-06-27	2023-06-27	PEP 537
3.6	security	2016-12-23	2021-12-23	PEP 494
2.7	end-of-life	2010-07-03	2020-01-01	PEP 373

Posterior a la descarga, se abre el ejecutable como administrador.

Seguido de esto marque la opción “Add Python 3.9 to PATH”, para que sea fácil usar ciertas herramientas de Python.

Por último presione “Install Now”:



Seguidamente se va presionando next hasta que el programa empieza a instalarse. Cuando termine, presione el botón de “close”.

Con esto ya tendría Python en su computador.

## Instalar la biblioteca Pygame

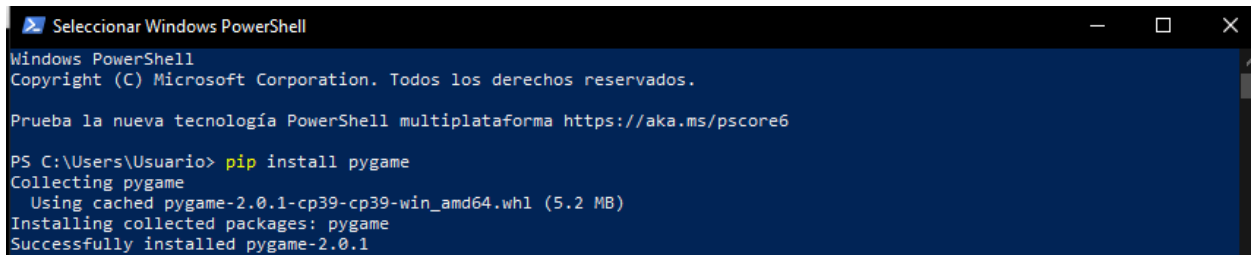
Los programas utilizan la biblioteca gráfica Pygame, por lo que es necesario instalar esta de la siguiente forma:

Primeramente, con Python ya instalado, abra Windows Powershell, escriba python y presione enter. Si sale el siguiente texto, la instalación fue correcta, pues sí se instaló en el PATH.

```
Python 3.9.5 (tags/v3.9.5:0a7dcdb, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

En caso de que salga otra cosa, repetir la instalación de Python de forma correcta, con la casilla “Add Python 3.9 to PATH” marcada.

Seguidamente escriba `pip install pygame`, presione enter:

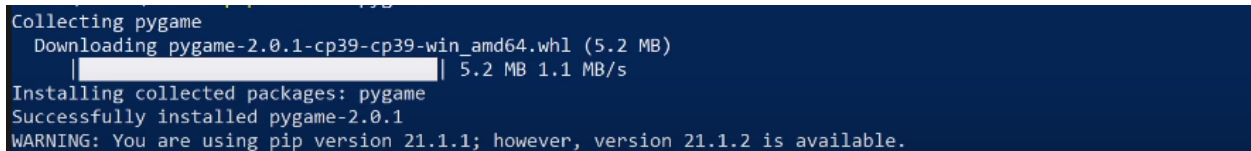


```
Seleccin Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnologa PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\Usuario> pip install pygame
Collecting pygame
  Using cached pygame-2.0.1-cp39-cp39-win_amd64.whl (5.2 MB)
Installing collected packages: pygame
Successfully installed pygame-2.0.1
```

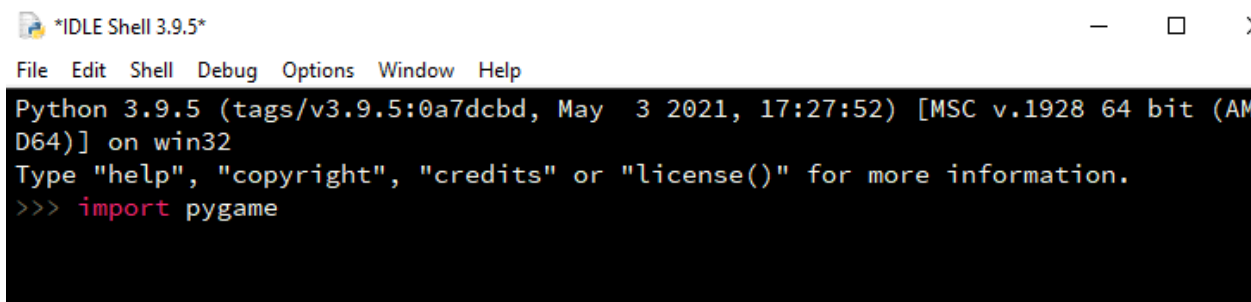
Le debe salir lo siguiente:



```
Collecting pygame
  Downloading pygame-2.0.1-cp39-cp39-win_amd64.whl (5.2 MB)
    | 5.2 MB 1.1 MB/s
Installing collected packages: pygame
Successfully installed pygame-2.0.1
WARNING: You are using pip version 21.1.1; however, version 21.1.2 is available.
```

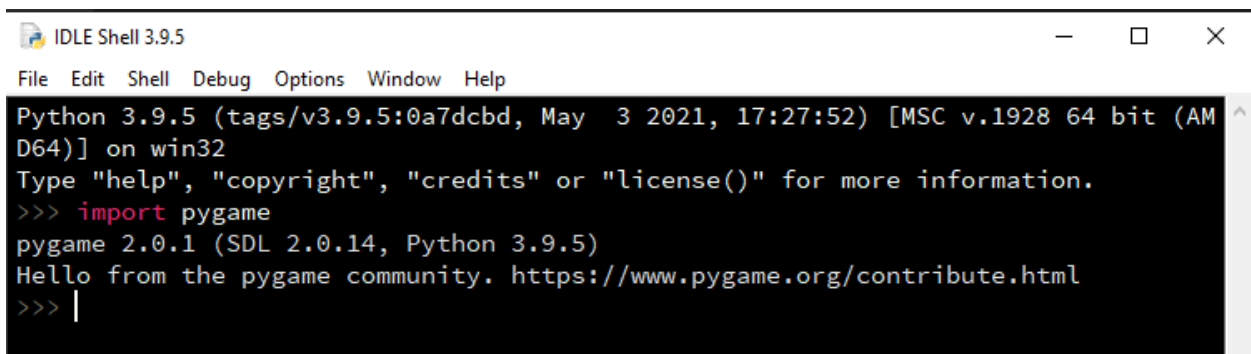
Cuando la instalaci3n termine, cierre Windows PowerShell.

Por 3ltimo, para verificar que todo se instal3 de manera correcta, abra el IDLE shell de Python y escriba lo siguiente y presione enter:



```
IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcdb, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import pygame
```

Si todo sali3 bien, le deber3 salir lo siguiente:

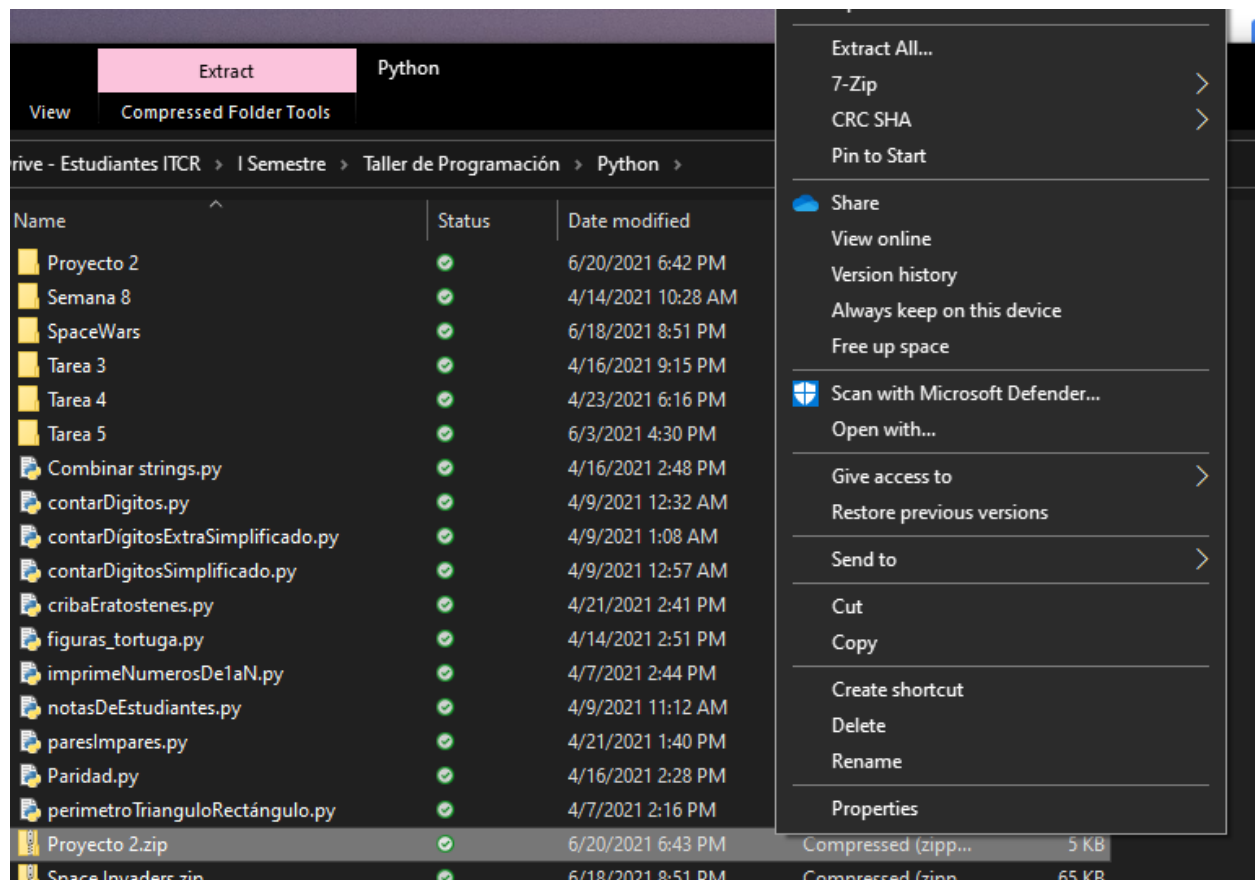


```
IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcdb, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import pygame
pygame 2.0.1 (SDL 2.0.14, Python 3.9.5)
Hello from the pygame community. https://www.pygame.org/contribute.html
>>> |
```

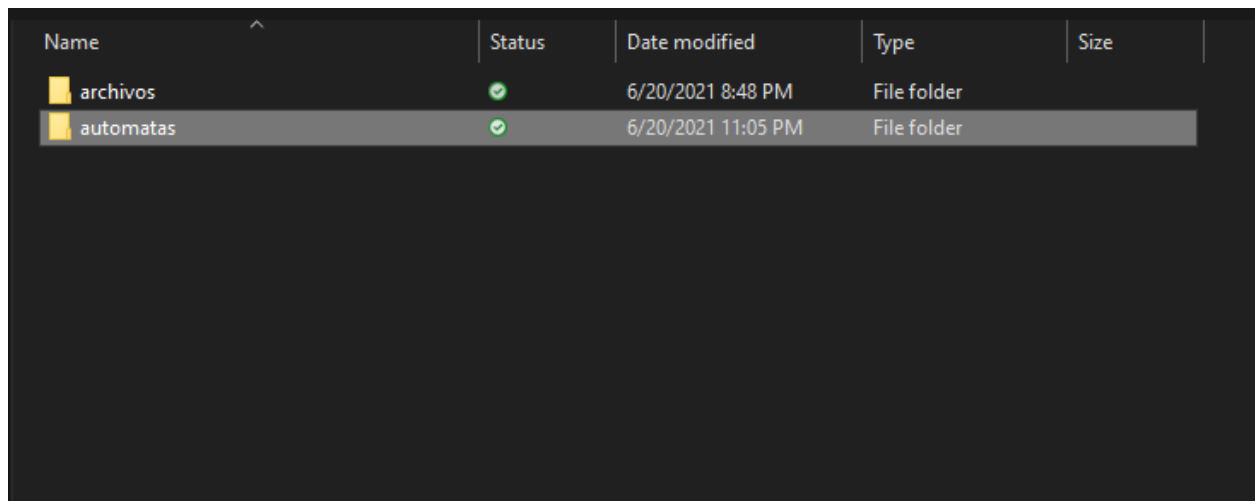
Ya con esto, todo est3 listo para poder correr los programas.

## Abrir los programas

Primeramente descomprima la carpeta “Proyecto 2”. Puede hacerlo dando click derecho en el archivo zip y presionando Extraer todo... (Extract All... en inglés).



Abrala y diríjase a la carpeta autómatas:



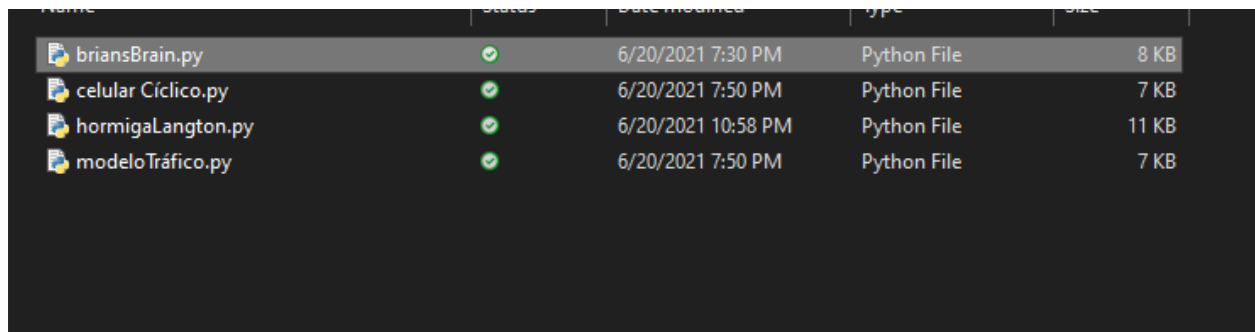
Name	Status	Date modified	Type	Size
archivos	✓	6/20/2021 8:48 PM	File folder	
autómatas	✓	6/20/2021 11:05 PM	File folder	

Ahí verá varios 4 archivos con extensión .py, los cuales son los diferentes autómatas. *Más adelante se explica el funcionamiento de cada uno.*

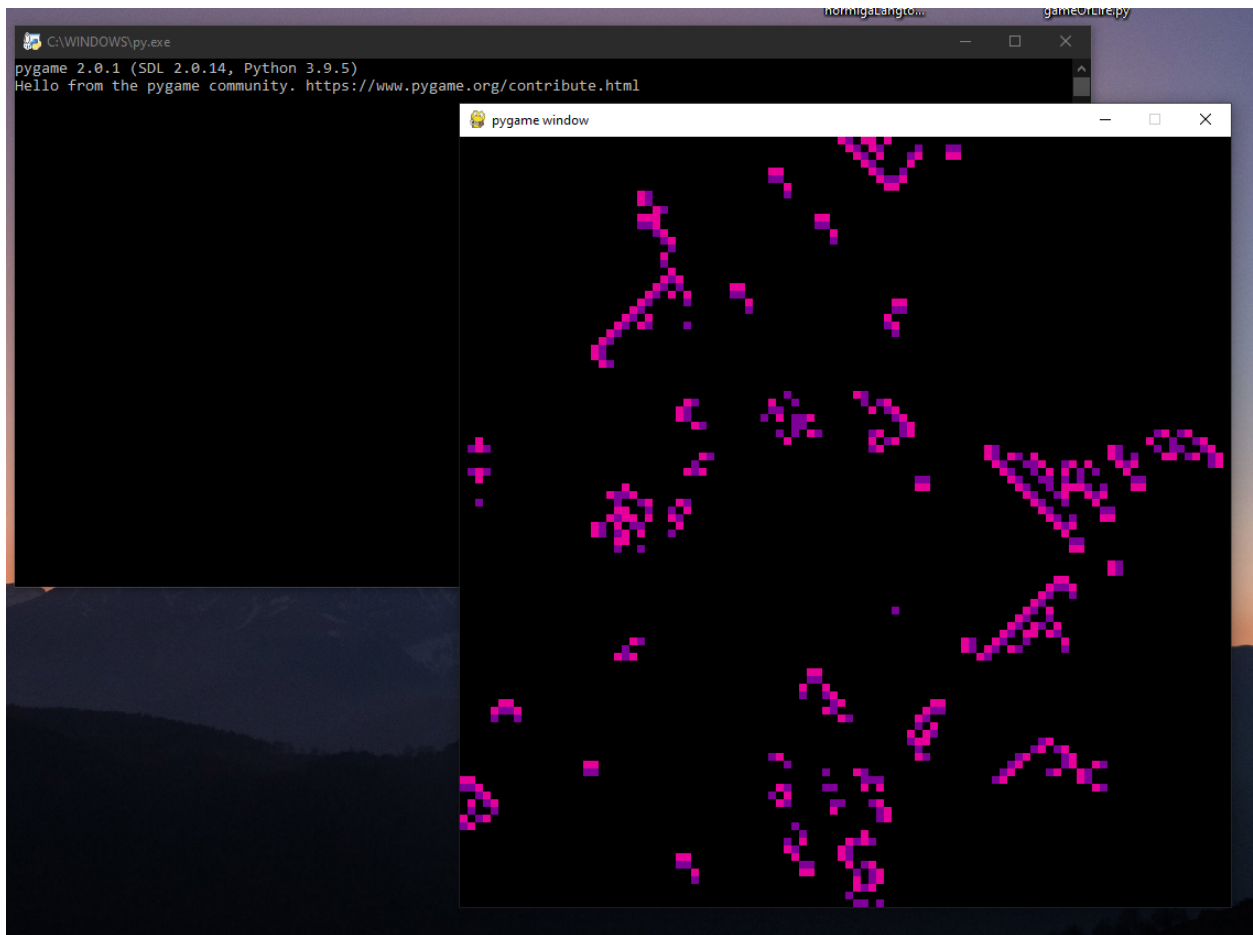
Puede ejecutarlos de las siguientes formas:

- Ejecutarlos abriéndolos directamente

Para comenzar a ver su autómata, puede dar doble click al archivo, e inmediatamente se abrirá una ventana donde su autómata comenzará a funcionar.



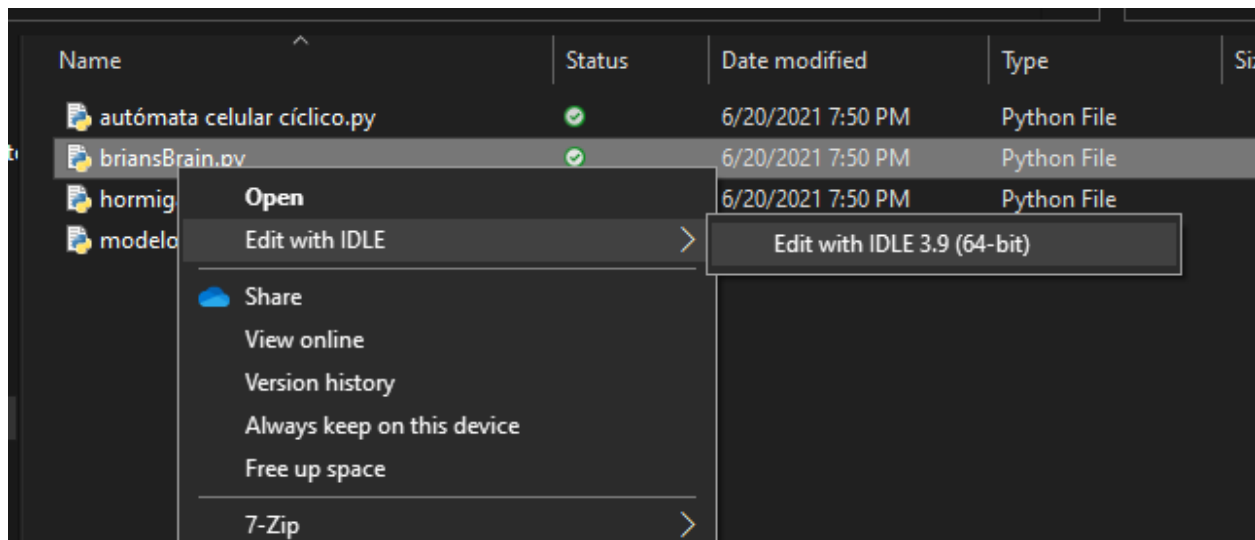
Name	Status	Date modified	Type	Size
briansBrain.py	✓	6/20/2021 7:30 PM	Python File	8 KB
celular Cíclico.py	✓	6/20/2021 7:50 PM	Python File	7 KB
hormigaLangton.py	✓	6/20/2021 10:58 PM	Python File	11 KB
modeloTráfico.py	✓	6/20/2021 7:50 PM	Python File	7 KB



- Abrirlos por medio del Idle de python

Para poder ejecutar los programas también puede abrirlos en Python IDLE. Para esto, presione click derecho sobre el archivo del autómata deseado (todos poseen extensión .py) y elija la opción “Edit with IDLE” y luego presione “Edit with IDLE 3.9” como se muestra a continuación:





Al abrir debería aparecer el código de los programas de esta forma:

```

briansBrain.py - C:\Users\jmque\OneDrive - Estudiantes ITCR\I Semestre\Taller de Programación\Python\Proyect...
File Edit Format Run Options Window Help

#Brian's Brain
#José Manuel Quesada y Christopher Jiménez

##### Imports #####
import pygame as p
import json
from random import randrange
from copy import deepcopy
##### Global Var #####

_ANCHO = 1000
_ALTO = 1000
_TAM = 8

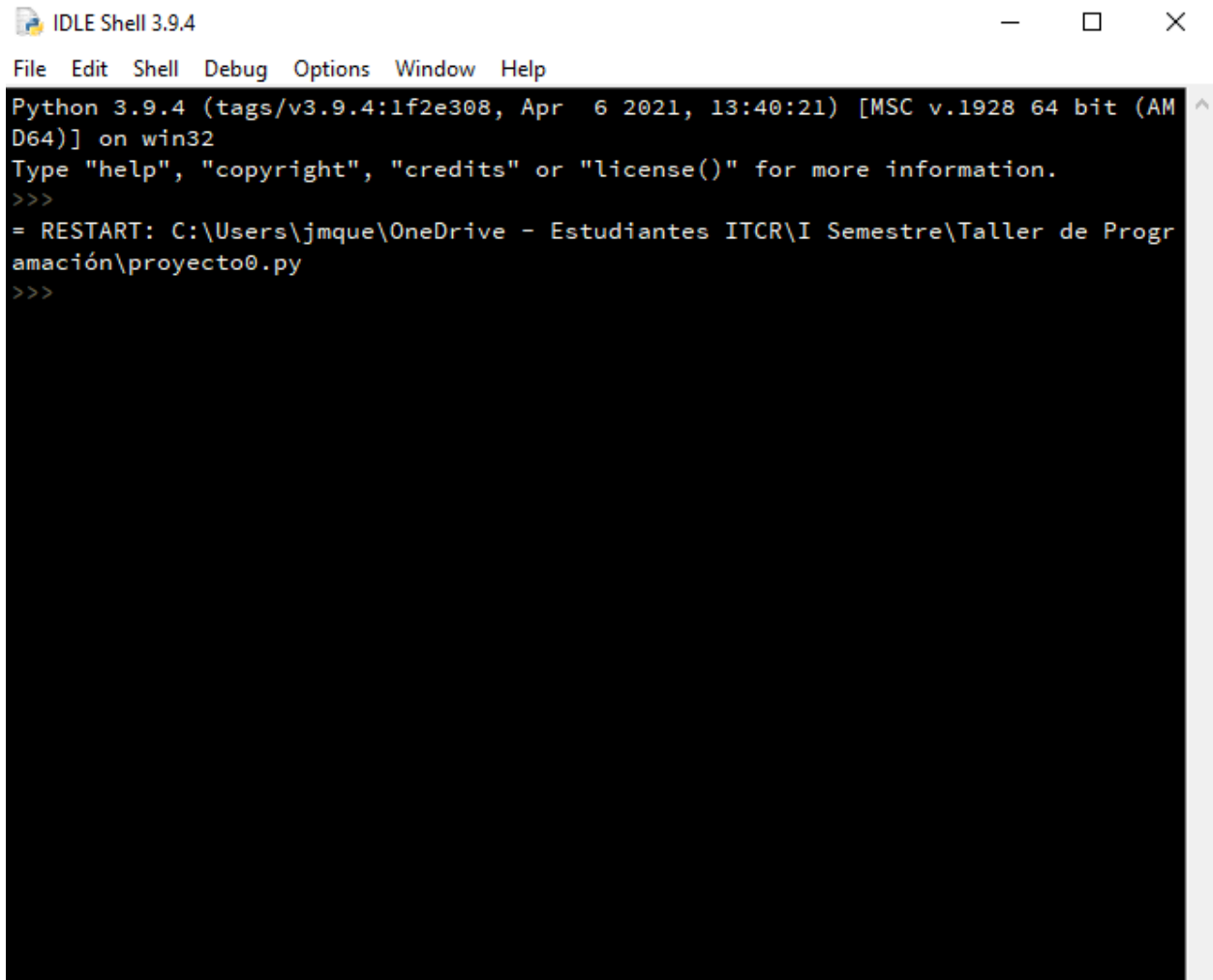
_COLS = _ANCHO // _TAM + 1
_FILAS = _ALTO // _TAM + 1

##### Matrices #####

def crearMatrizRandom(filas, columnas, estados):
    """
    Función que crea una matriz con las dimensiones indicadas
    y con valores aleatorios en el rango de 3 estados (vivo, muriendo, muerto).
    Entradas y restricciones:
    - filas : cantidad de filas, número entero positivo.
    - columnas : cantidad de columnas, numero entero positivo.
    - estados : cantidad de estados, numero entero positivo.
    Salida:
    - Retorna la matriz (en forma de lista) con número de filas y columnas
  
```

Posteriormente, para abrir el shell de Python y poder ejecutar cada programa, es necesario presionar F5 en su teclado, o alternativamente presionar en la parte superior de la ventana del IDLE el botón “Run” y luego “Run module”.

Al realizar este paso de manera correcta, se debe abrir una ventana con el nombre de IDLE Shell, como se muestra a continuación:



```
Python 3.9.4 (tags/v3.9.4:1f2e308, Apr 6 2021, 13:40:21) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\jmque\OneDrive - Estudiantes ITCR\I Semestre\Taller de Programación\proyecto0.py
>>>
```

## Cómo utilizar el programa

El archivo cuenta con un total de 4 programas que implementan un autómata distinto mediante la biblioteca de pygame. A continuación se detalla cómo utilizar cada uno de ello :

Primero que nada, al entrar a cada uno de lo programas, hay una sección de variables globales, la cual luce así:

```
_ANCHO = 1000
_ALTO = 1000
_TAM = 8

_COLS = _ANCHO // _TAM + 1
_FILAS = _ALTO // _TAM + 1
```

En este apartado, se puede cambiar el ancho(\_ANCHO) y el alto(\_ALTO) de la ventana, además del tamaño(\_TAM) de la celda, se puede asignar los valores enteros positivos que se desee.

Nota: no se debe cambiar el valor de las variables \_COLS y \_FILAS.

Además, al abrir cada autómata, se puede interactuar con el mismo mediante varias teclas:

- Espacio: pausa o reanuda el ciclo del autónoma.
- Tecla r: genera una matriz aleatoria nueva .
- Tecla b: reinicia la matriz con el valor neutro(0).
- Tecla g: guarda en un archivo el estado actual del autómata.
- Tecla c: carga el archivo del autómata.
- Click del mouse: cambia al siguiente estado la célula presionada.

A continuación se detalla cómo usar cada autómata.

## 1. Brian's Brain

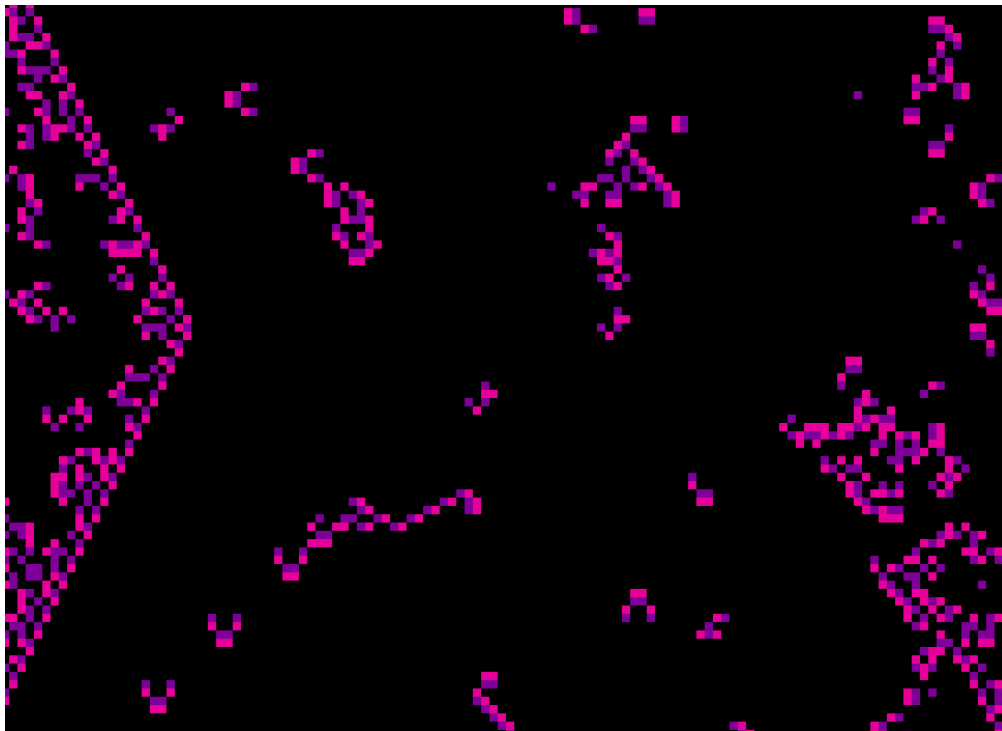
En este autómatas, las celdas representan células que tienen 3 estados: vivas, muriendo o muertas. Cada célula se rige por 3 simples reglas:

1. Si está muerta y tiene 2 vecinos vivos pasa a estar viva.
2. Si está viva, pasa a estar muriendo.
3. Si está muriendo, pasa a estar muerta.

Para obtener este autómatas, se debe abrir el archivo:

`briansBrain.py`

Clickeando dos veces el archivo, la ventana con el autómatas se ejecutará de la siguiente manera:



Igualmente, también se puede seguir el proceso de editar con el IDLE de python, mencionado anteriormente.

## 2. Autómata celular cíclico:

Este autómata presenta 15 estados de la célula, distinguidos por su color.. La regla de funcionamiento es simple: una célula pasa al siguiente estado, si al menos uno de sus vecinos tiene ese mismo estado.

Para obtener este autómata, se debe abrir el siguiente archivo:

`celularCiclico.py`

Clickeando dos veces el archivo, la ventana con el autómata se ejecutará de la siguiente manera:



Igualmente, también se puede seguir el proceso de editar con el IDLE de python, mencionado anteriormente.

### 3. Hormiga de Langton.

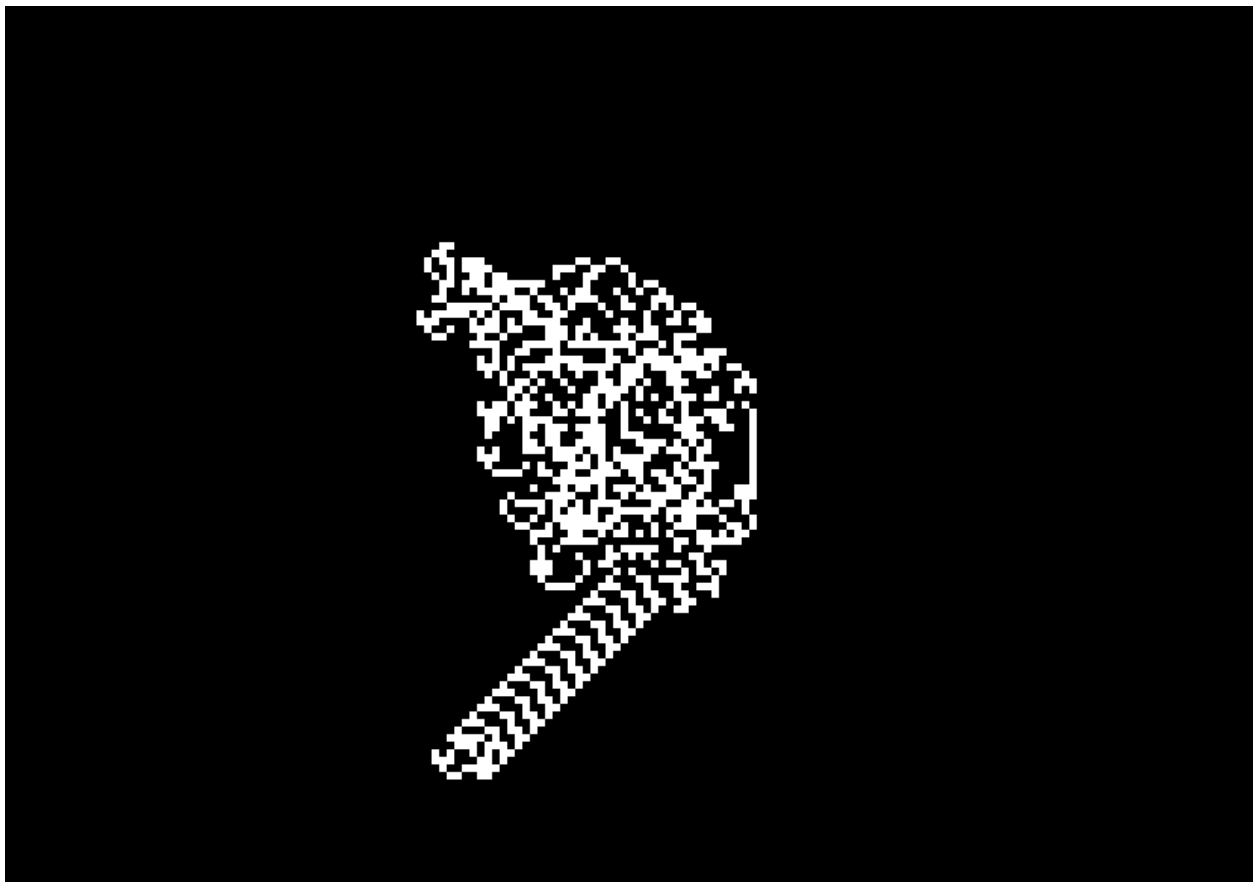
Este autómata utiliza células en dos estados, y una posición que representa una hormiga que tiene dirección. Funciona bajo dos reglas:

1. Si la hormiga está en una célula blanca, gira a la derecha 90 grados, cambia el color de la célula y avanza una celda.
2. Si está en una célula negra. Gira 90 grados a la izquierda, cambia el color de la célula y avanza una celda.

Para obtener este autómata, se debe abrir el siguiente archivo:

`hormigaLangton.py`

Clickeando dos veces el archivo, la ventana con el autómata se ejecutará de la siguiente manera:



Igualmente, también se puede seguir el proceso de editar con el IDLE de python, mencionado anteriormente.

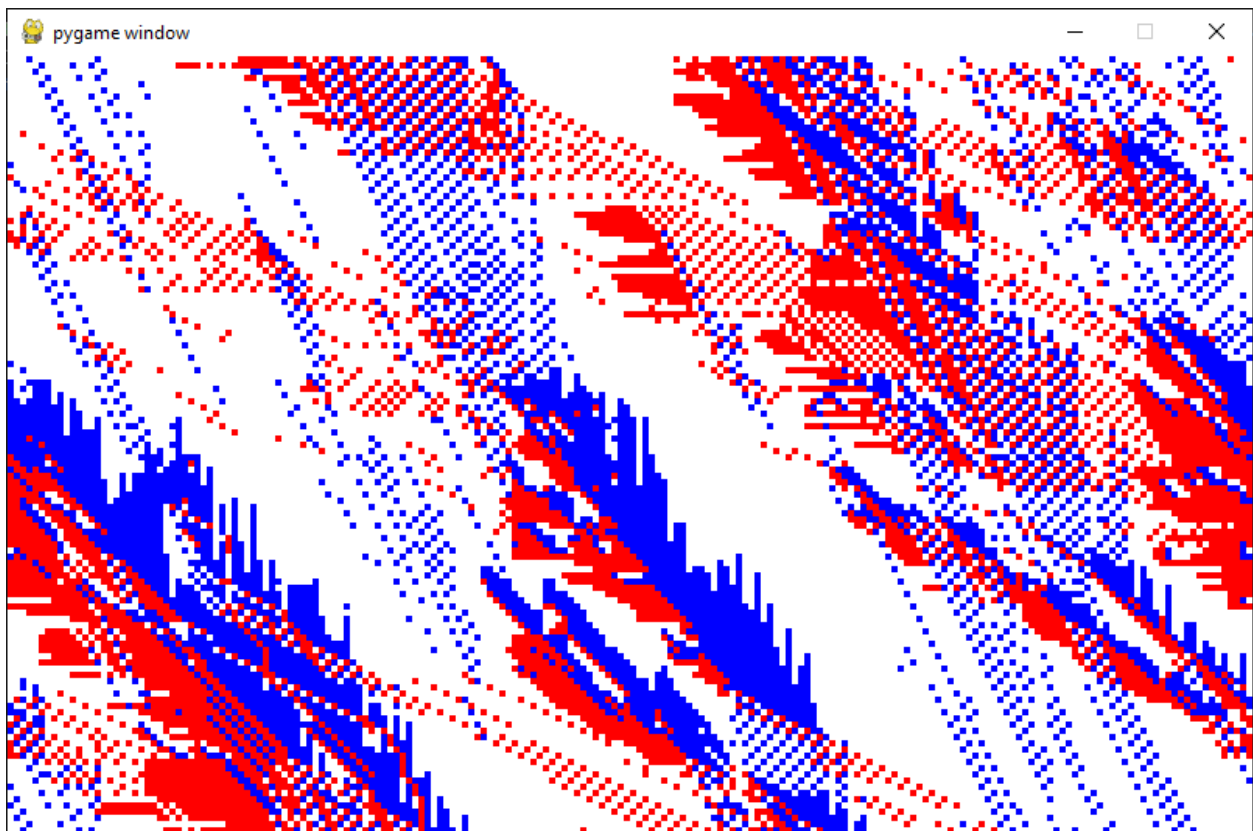
#### 4. Modelo de tráfico Biham-Middleton-Levine

Este autómata simula el tránsito mediante 3 estados de celdas: vacía, con un auto rojo o con un auto azul. Un auto azul se mueve hacia abajo si la celda está vacía, un auto rojo se mueve hacia la derecha si la celda está vacía. De lo contrario, mantienen su posición actual.

Para obtener este autómata, se debe abrir el siguiente archivo:

`modeloTrafico.py`

Clickeando dos veces el archivo, la ventana con el autómata se ejecutará de la siguiente manera:



Igualmente, también se puede seguir el proceso de editar con el IDLE de python, mencionado anteriormente.



## Tabla resumida

Autómata	Archivo
Brian's Brain	briansBrain.py
Autómata celular cíclico	celularCiclico.py
Hormiga de Langton	hormigaLangton.py
Modelo de tráfico Biham-Middleton-Levine	modeloTrafico.py

Programa creado por:

Christopher Jiménez Gutierrez

José Manuel Quesada Carvajal

