**Nombre: Monjaraz Briseño Luis Fernando**

**Materia: Compiladores**

**Actividad: Hands-on 1 Implementación de Autómatas**

**Tema: autómatas**

**Fecha: 09 de marzo de 2025.**

# Índice

[Índice 2](#_Toc192525433)

[Tabla de imágenes 2](#_Toc192525434)

[Implementación de autómatas 3](#_Toc192525435)

[Autómata 1: Validación de cadenas alfabéticas 3](#_Toc192525436)

[Autómata 2: Validación de números reales. 5](#_Toc192525437)

[Autómata 3: Validación de sentencias selectivas (if-else). 7](#_Toc192525438)

[imagen general de evidencia en IDLE Visual Studio Code 10](#_Toc192525439)

[Referencias bibliográficas 11](#_Toc192525440)

# Tabla de imágenes

[Imagen 1 Código Autómata 1 Python – Visual Studio Code 3](#_Toc192525463)

[Imagen 2 Resultado Autómata 1 – Terminal Visual Studio Code 4](#_Toc192525464)

[Imagen 3 Código Autómata 2 Python – Visual Studio Code 5](#_Toc192525465)

[Imagen 4 Resultado Autómata 2 – Terminal Visual Studio Code 6](#_Toc192525466)

[Imagen 5 Código Autómata 3 Python – Visual Studio Code 7](#_Toc192525467)

[Imagen 6 Resultado Autómata 3 – Terminal Visual Studio Code 8](#_Toc192525468)

[Imagen 7 Evidencia en Visual Studio Code 10](#_Toc192525469)

# Implementación de autómatas

## Texto El contenido generado por IA puede ser incorrecto.Autómata 1: Validación de cadenas alfabéticas

Imagen 1 Código Autómata 1 Python – Visual Studio Code

Descripción: Este es el autómata 1: validación de cadenas alfabéticas, como se puede apreciar, realice una versión modificada con un ciclo for, esto para facilitar la ejecución del programa y no tener que correrlo varias veces para cada caso, siendo los casos de uso HelloWorld, Hello, World, HelloWorld123, Hello123, 123HelloWorld, Hello World, Hello123World, Hello123World!, Hello123World@, donde los primeros cinco casos son válidos HelloWorld, Hello, World, HelloWorld123, Hello123 y los últimos cinco son inválidos 123HelloWorld, Hello World, Hello123World, Hello123World!, Hello123World@, se podría decir que es el único cambio y agregue. El código se realizo en el IDLE Visual Studio Code y fue programado en Python.

Texto

El contenido generado por IA puede ser incorrecto.

Imagen 2 Resultado Autómata 1 – Terminal Visual Studio Code

Descripción: Como se puede apreciar los resultados son los esperados. Esto demuestra que se cumple con lo esperado y que el autómata funciona correctamente, siendo el primer print que solo dice “Cadena válida” correspondiente al código original y el resto al código modificado con un ciclo for. A continuación los resultados esperados y que corresponden a los dados.

HelloWorld - Cadena válida.

Hello - Cadena válida.

World - Cadena válida.

HelloWorld123 - Cadena inválida.

Hello123 - Cadena inválida.

123HelloWorld - Cadena inválida.

Hello World - Cadena inválida.

Hello123World - Cadena inválida.

Hello123World! - Cadena inválida.

Hello123World@ - Cadena inválida.

## Texto El contenido generado por IA puede ser incorrecto.Autómata 2: Validación de números reales.

Imagen 3 Código Autómata 2 Python – Visual Studio Code

Descripción: Este es el autómata 2: validación de números reales. Al igual que en el autómata 1, realicé una versión modificada con un ciclo for para facilitar la ejecución del programa y evitar tener que correrlo varias veces para cada caso. Los casos de uso son -123.456, 123.456, 123, -123, 123., .123, 123.456.789, 123.456.789, 123,456, 123,456,789, donde los primeros seis casos son válidos (-123.456, 123.456, 123, -123, 123., .123) y los últimos cuatro son inválidos (123.456.789, 123.456.789, 123,456, 123,456,789). El código se realizó en el IDLE Visual Studio Code y fue programado en Python.

Texto

El contenido generado por IA puede ser incorrecto.

Imagen 4 Resultado Autómata 2 – Terminal Visual Studio Code

Descripción: Como se puede apreciar, los resultados son los esperados. Esto demuestra que se cumple con lo esperado y que el autómata funciona correctamente, siendo el primer print que solo dice “Número válido” correspondiente al código original y el resto al código modificado con un ciclo for. A continuación, los resultados esperados y que corresponden a los dados:

-123.456 - Número válido.

123.456 - Número válido.

123 - Número válido.

-123 - Número válido.

123. - Número válido.

.123 - Número válido.

123.456.789 - Número inválido.

123.456.789 - Número inválido.

123,456 - Número inválido.

123,456,789 - Número inválido.

## Texto El contenido generado por IA puede ser incorrecto.Autómata 3: Validación de sentencias selectivas (if-else).

Imagen 5 Código Autómata 3 Python – Visual Studio Code

Descripción: Este es el autómata 3: validación de sentencias selectivas (if-else). Al igual que en los autómatas anteriores, utilicé un ciclo for para facilitar la ejecución del programa y evitar tener que correrlo varias veces para cada caso. Los casos de uso son if x > 0: pass else: pass, if (x > 0) { } else { }, if x > 0: print(x) else: print('No'), if x > 0: pass elif x == 0: pass else: pass, if x > 0: pass else: pass, if x > 0: pass, else: pass, if x > 0: pass elif x == 0: pass, if x > 0: pass else if x == 0: pass, print('Hello'), donde los primeros cinco casos son válidos (if x > 0: pass else: pass, if (x > 0) { } else { }, if x > 0: print(x) else: print('No'), if x > 0: pass elif x == 0: pass else: pass, if x > 0: pass else: pass) y los últimos cinco son inválidos (if x > 0: pass, else: pass, if x > 0: pass elif x == 0: pass, if x > 0: pass else if x == 0: pass, print('Hello')). El código se realizó en el IDLE Visual Studio Code y fue programado en Python.

Texto

El contenido generado por IA puede ser incorrecto.

Imagen 6 Resultado Autómata 3 – Terminal Visual Studio Code

Descripción: Como se puede apreciar, los resultados son los esperados. Esto demuestra que se cumple con lo esperado y que el autómata funciona correctamente, siendo el primer print que solo dice “Sentencia válida” correspondiente al código original y el resto al código modificado con un ciclo for. A continuación, los resultados esperados y que corresponden a los dados:

if x > 0: pass else: pass - Sentencia válida.

if (x > 0) { } else { } - Sentencia válida.

if x > 0: print(x) else: print('No') - Sentencia válida.

if x > 0: pass elif x == 0: pass else: pass - Sentencia válida.

if x > 0: pass else: pass - Sentencia válida.

if x > 0: pass - Sentencia inválida.

else: pass - Sentencia inválida.

if x > 0: pass elif x == 0: pass - Sentencia inválida.

if x > 0: pass else if x == 0: pass - Sentencia inválida.

print('Hello') - Sentencia inválida.

## Texto El contenido generado por IA puede ser incorrecto.imagen general de evidencia en IDLE Visual Studio Code

Imagen 7 Evidencia en Visual Studio Code

# Referencias bibliográficas

automata-lib. (2025, January 23). Retrieved from <https://pypi.org/project/automata-lib/>

Fernandez, R. (2019, May 28). Autómata Finito en Python - ▷ Cursos de Programación de 0 a Experto © Garantizados. Retrieved from <https://unipython.com/automata-finito-en-python/>

Mateo Ortega. (2022, July 5). *Autómata Finito Determinista Reconocedor implementado en python* [Video file]. Retrieved from <https://www.youtube.com/watch?v=ReR1hOhQKHY>