THE HANGMAN GAME

El Juego del Ahorcado

J. Daniel Figueroa
Kevin Quintero
Kevin Castro

INTRODUCCION

La idea detrás de este proyecto es implementar de manera clara y eficiente el juego del ahorcado. El objetivo es utilizar autómatas para controlar la lógica del juego y generar palabras aleatorias, brindando una experiencia interactiva y entretenida para los jugadores.

DEFINICION FORMAL DEL AUTOMATA

A = (Q, Σ, δ, q0,F)

Donde:

- Q → Estados = {q0 , q1 , q2 , q3 , … , q133 , q134}
- Σ → Alfabeto = {a , b , c , d , e , f , g , h , i , j , k , l , m , n , ñ , o , p , q , r , s , t , u , v , w , x , y , z}
- δ → Transiciones = {δ(q0,a)=q1, δ(q1,m)=q2, δ(q2,a)=q3, δ(q3,r)=q4 }
- q0 → Estado inicial = { q0 }
- F → Estados de aceptación = {q4 , q12, q21 , … , q117 , q125 , q134}

IMPLEMENTACION DEL AUTOMATA

```
from IPython.display import Image, display
```

Verificacion Automata Finito Determinista (DFA)

```
def hangmangame():

    from automata.fa.dfa import DFA
    d = DFA(
        states={'q0', 'q1', 'q2', 'q3', 'q4', 'q5', 'q6', 'q7', 'q8', 'q9', 'q10', 'q11', 'q12', 'q13', 'q14', 'q15', 'q16', 'q17', 'q18', 'q19',
                'q20', 'q21', 'q22', 'q23', 'q24', 'q25', 'q26', 'q27', 'q28', 'q29', 'q30', 'q31','q32','q33','q34','q35','q36','q37','q38',
                'q39','q40','q41','q42','q43','q44','q45','q46','q47','q48','q49','q50','q51','q52','q53','q54','q55','q56','q57','q58','q59'
               ,'q60','q61','q62','q63','q64','q65','q66','q67','q68','q69','q70','q71','q72','q73','q74','q75','q76','q77','q78'
               ,'q79','q80','q81','q82','q83','q84','q85','q86','q87','q88','q89','q90','q91','q92','q93','q94','q95','q96','q97','q98','q99'
               ,'q100','q101','q102','q103','q104','q105','q106','q107','q108','q109','q110','q111','q112','q113','q114','q115','q116','q117','q118',
                'q119','q120','q121','q122','q123','q124','q125','q126','q127','q128','q129','q130','q131','q132','q133','q134' },
        input_symbols={'a', 'b', 'c', 'd', 'e', 'f','g', 'h', 'i','l', 'j', 'k', 'm','n', 'o', 'p', 'q', 'r', 's','t', 'u', 'v', 'w', 'x', 'y','z'},
        transitions={
        'q0': {'a': 'q1','c': 'q32','g': 'q65','l': 'q74','e': 'q82','q': 'q97','s': 'q107','p': 'q126'},
        'q1': {'m': 'q2', 'u': 'q13','l': 'q58'},
        'q2': {'a': 'q3', 'o': 'q5'},
        'q3': {'r': 'q4'},
        'q4': {},
        'q5': {'r': 'q6'},
        'q6': {'t': 'q7'},
        'q7': {'i': 'q8'},
        'q8': {'g': 'q9'},
        'q9': {'u': 'q10'},
        'q10': {'a': 'q11'},
        'q11': {'r': 'q12'},
        'q12': {},
```

IMPLEMENTACION DEL AUTOMATA

```
############################
# VERIFICACION DE PALABRAS #
############################

def verificacion():

    d = hangmangame()
    valid_words = ['amar','amortiguar','automatico','automata','automotriz','autonomo','alfabeto','canguro','cazador',
                   'caramelo','colombia','color','gramatica','sanguijuela','paralisis','estratosfera','estado','quehaceres','litosfera',
                   'lenguaje']

    invalid_words = ['auto','quehacer','caza','cara','colorbia','lengua','amor']

    for i in valid_words:
        if d.accepts_input(i) != True:
            return False
    for j in invalid_words:
        if d.accepts_input(j) != False:
            return False
    return True

if verificacion():
  print('El diseño es correcto')
else:
  print('El diseño es incorrecto')
```

IMPLEMENTACION DEL AUTOMATA

```python
def buscarPalabraAleat(automata):
    ultimaPalabra= '' ''
    estado_actual = automata.initial_state
    palabra = ""

    while estado_actual not in automata.final_states or palabra == ultimaPalabra:
        transiciones_posibles = automata.transitions[estado_actual]
        letra = random.choice(list(transiciones_posibles.keys()))
        palabra += letra
        estado_actual = transiciones_posibles[letra]

    ultimaPalabra = palabra
    return palabra
```

IMPLEMENTACION DEL AUTOMATA

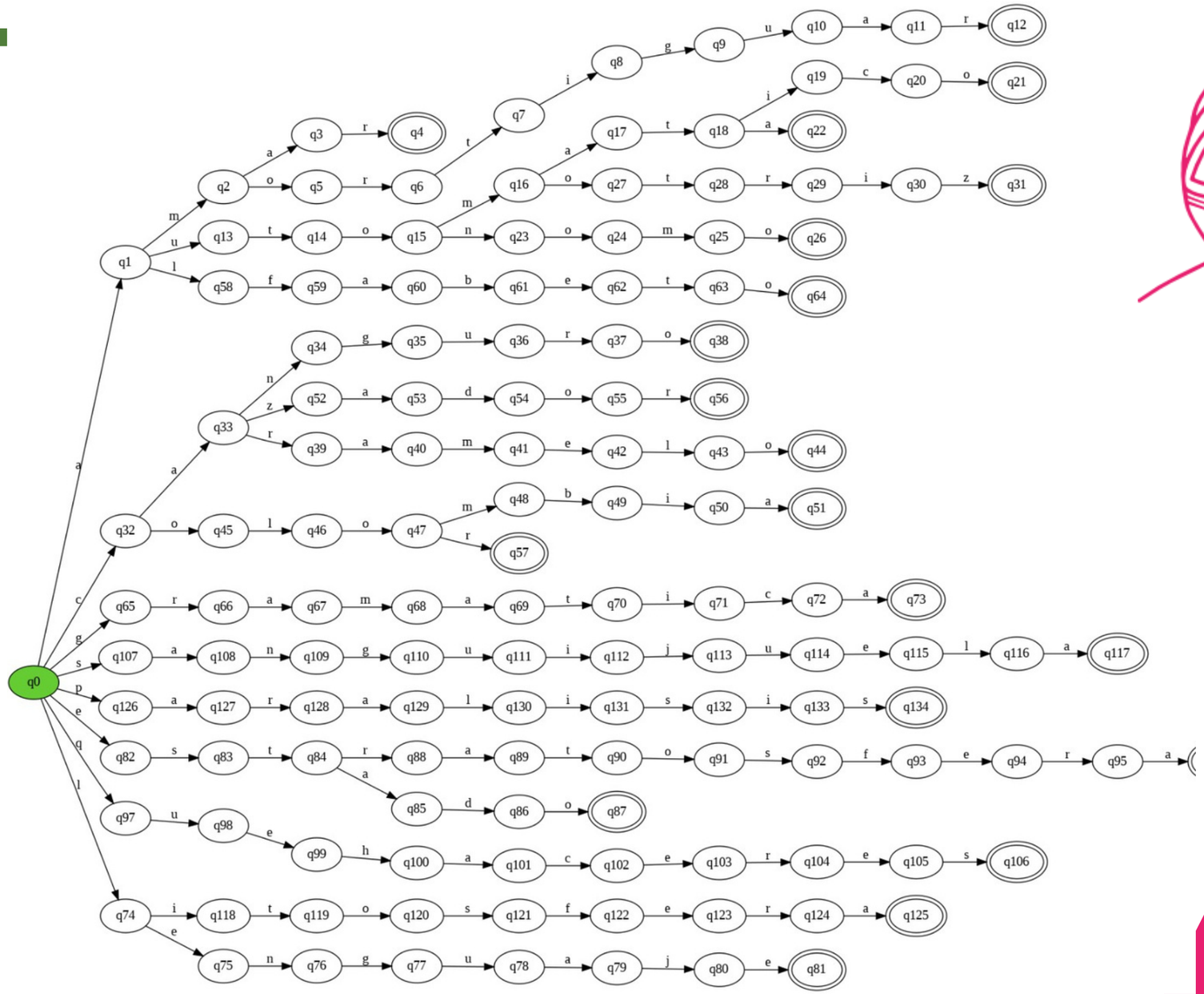
```
def elijeLetra(algunaLetra):
    while True:
        print('Adivina una letra:')
        letra = input()
        letra = letra.lower()
        if len(letra) != 1:
            print('Introduce una sola letra.')
        elif letra in algunaLetra:
            print('Ya has elegido esa letra. ¿Qué tal si pruebas con otra?')
        elif letra not in 'abcdefghijklmnopqrstuvwxyz1234567890':
            print('Elige una letra.')
        else:
            return letra
```

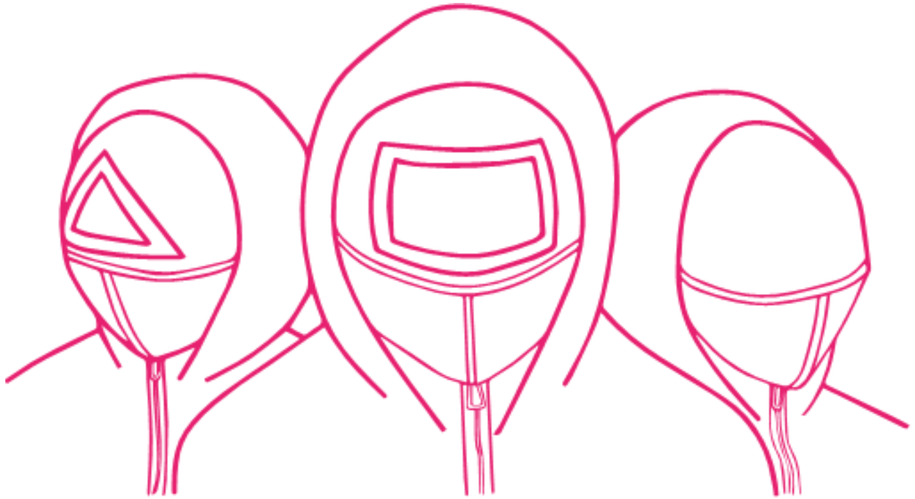
IMPLEMENTACION DEL AUTOMATA

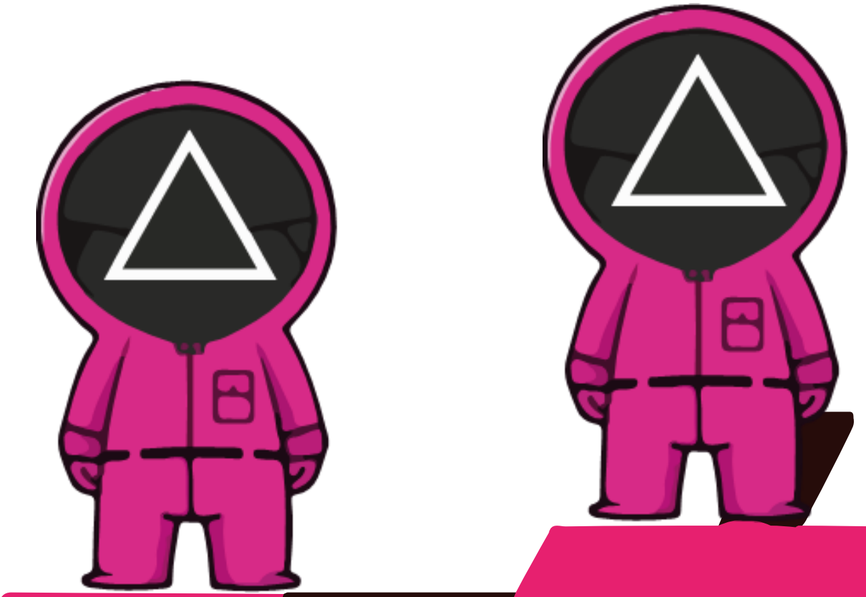
```python
def empezar():
    print('¿Quieres jugar de nuevo? (Si o No)')
    return input().lower().startswith('s')
```

IMPLEMENTACION DEL AUTOMATA

```
# Automata
d = DFA(
        states={'q0', 'q1', 'q2', 'q3', 'q4', 'q5', 'q6', 'q7', 'q8', 'q9', 'q10', 'q11', 'q12', 'q13', 'q14', 'q15', 'q16', 'q17', 'q18', 'q19',
                'q20', 'q21', 'q22', 'q23', 'q24', 'q25', 'q26', 'q27', 'q28', 'q29', 'q30', 'q31','q32','q33','q34','q35','q36','q37','q38',
                'q39','q40','q41','q42','q43','q44','q45','q46','q47','q48','q49','q50','q51','q52','q53','q54','q55','q56','q57','q58','q59'
                ,'q60','q61','q62','q63','q64','q65','q66','q67','q68','q69','q70','q71','q72','q73','q74','q75','q76','q77','q78'
                ,'q79','q80','q81','q82','q83','q84','q85','q86','q87','q88','q89','q90','q91','q92','q93','q94','q95','q96','q97','q98','q99'
                ,'q100','q101','q102','q103','q104','q105','q106','q107','q108','q109','q110','q111','q112','q113','q114','q115','q116','q117','q118',
                'q119','q120','q121','q122','q123','q124','q125','q126','q127','q128','q129','q130','q131','q132','q133','q134' },
        input_symbols={'a', 'b', 'c', 'd', 'e', 'f','g', 'h', 'i','l', 'j', 'k', 'm','n', 'o', 'p', 'q', 'r', 's','t', 'u', 'v', 'w', 'x', 'y','z'},
        transitions={
        'q0': {'a': 'q1','c': 'q32','g': 'q65','l': 'q74','e': 'q82','q': 'q97','s': 'q107','p': 'q126'},
        'q1': {'m': 'q2', 'u': 'q13','l': 'q58'},
        'q2': {'a': 'q3', 'o': 'q5'},
        'q3': {'r': 'q4'},
        'q4': {},
        'q5': {'r': 'q6'},
        'q6': {'t': 'q7'},
        'q7': {'i': 'q8'},
        'q8': {'g': 'q9'},
        'q9': {'u': 'q10'},
        'q10': {'a': 'q11'},
        'q11': {'r': 'q12'},
        'q12': {},
        'q13': {'t': 'q14'},
        'q14': {'o': 'q15'},
        'q15': {'n': 'q23','m': 'q16'},
        'q16': {'o': 'q27','a': 'q17',},
        'q17': {'t': 'q18'},
        'q18': {'a': 'q22','i':'q19'},
        'q19': {'c': 'q20'},
        'q20': {'o': 'q21'},
```


q0
q1
q2
q3
q4
q5
q6
q7
q8
q9
q10
q11
q12
q13
q14
q15
q16
q17
q18
q19
q20
q21
q22
q23
q24
q25
q26
q27
q28
q29
q30
q31
q32
q33
q34
q35
q36
q37
q38
q39
q40
q41
q42
q43
q44
q45
q46
q47
q48
q49
q50
q51
q52
q53
q54
q55
q56
q57
q58
q59
q60
q61
q62
q63
q64
q65
q66
q67
q68
q69
q70
q71
q72
q73
q74
q75
q76
q77
q78
q79
q80
q81
q82
q83
q84
q85
q86
q87
q88
q89
q90
q91
q92
q93
q94
q95
q97
q98
q99
q100
q101
q102
q103
q104
q105
q106
q107
q108
q109
q110
q111
q112
q113
q114
q115
q116
q117
q118
q119
q120
q121
q122
q123
q124
q125
q126
q127
q128
q129
q130
q131
q132
q133
q134





RESULTADOS/CONCLUSIONES

1. Implementación exitosa: Se logro implementar el juego del ahorcado utilizando automatas de manera exitosa.

1. El automata usado en el juego permite generar palabras aleatorias siguiendo las transiciones definidas en el.

1. Se demuestra cómo se pueden utilizar estas estructuras para modelar y controlar el flujo de un juego. Los autómatas son especialmente útiles para juegos basados en reglas y secuencias de eventos.

1. Aunque el juego del ahorcado implementado con autómatas es funcional, siempre hay espacio para mejoras y expansiones. Por ejemplo, se podrían agregar más palabras al autómata, aumentar la complejidad del juego, agregar diferentes niveles de dificultad o implementar características adicionales

GRACIAS!