



# Teoría de la computación

Mtro. Abelardo Gómez Andrade

Proyecto Final

Autómata Finito



## *“UN PERSONAJE EN APUROS”*

### INTEGRANTES DEL EQUIPO:

- Ángel Díaz
- Miguel Ángel Mendoza García
- Manuel Fernando Medina Guerrero

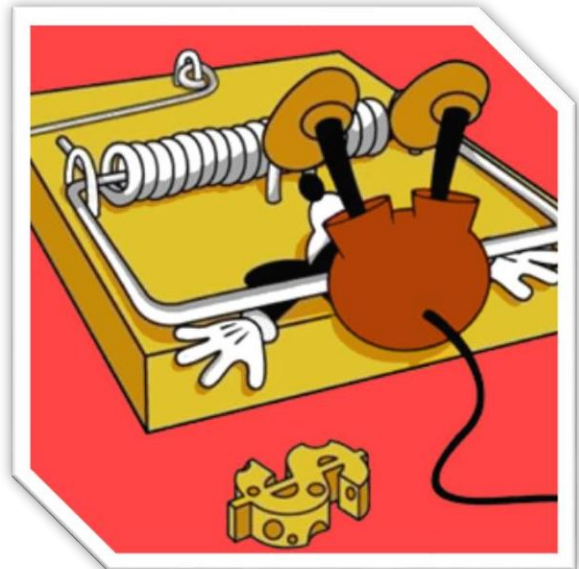
## **Planteamiento del Problema**

El problema radica en un personaje de un videojuego el cual, se encuentra en una situación de riesgo y para ello necesita de varias acciones que permitan completar la misión y no morir en ella, de aquí, el personaje realizará una serie de movimientos para tratar de sobrevivir a todas las trampas y superar esta prueba con vida.

El personaje es aventurero y desempeña distintas actividades a lo largo del videojuego, pero nos enfocamos más en una parte en la que tiene que con volar, avanzar, retroceder, agacharse y estar en constante movimiento; en caso de que se quede parado o quieto el personaje morirá a causa de una trampa, entonces él tiene que pasar diferentes obstáculos con los movimientos antes mencionados, si los lleva a cabo de manera correcta esto hará que el resultado no sea fatal para el personaje.

El propósito principal de nuestro proyecto, básicamente es sobrevivir el mayor tiempo posible hasta morir. Podríamos decir que el personaje tiene el rol más difícil en el videojuego y que al morir hará que el juego terminara.

Mediante un autómata finito representaremos los movimientos que el personaje podrá hacer para la simulación de nuestro juego.



## **Objetivo**



Te has puesto a pensar, ¿Los videojuegos tienen un mecanismo de rechazo o aceptación de instrucciones?

Bueno en este punto explicaremos el objetivo al cual queremos llegar; demostraremos que un autómata finito puede coordinar perfectamente las instrucciones que se le dan a un personaje, del cual se basa nuestro proyecto.

Nuestra hipótesis se basa principalmente en que un autómata rechazará o aceptará símbolos predeterminados por nosotros mismos como variables para llevar a cabo la misión del videojuego, nos daremos cuenta que un autómata puede ser el diagrama perfecto para definir los movimientos adecuados del personaje, lo cual nos lleva a destacar que no solo el autómata es para reconocer cadenas sino también para llevar una simulación muy detallada de lo que se quiere realizar, en nuestro caso, definir con botones del teclado de una computadora las entradas, lo cual nos lleva a acciones que si en la misión son las correctas las aceptará, en caso contrario, el autómata rechazará los movimientos del personaje, lo que quiere decir que se llegó a un estado de no aceptación.

Así queda planteado nuestro objetivo dando a conocer que con modelos de autómatas finitos se pueden llevar a cabo simulaciones y resolver problemáticas de la actualidad.

## **Justificación**

Nuestro proyecto, ayudará a que las personas se den cuenta de la importancia de un mecanismo de simulación, esto con el fin de realizar más proyectos o trabajos que impliquen la implementación de autómatas, que tienen la finalidad de resolver problemáticas a base de entradas que ejecuten acciones.



En muchos campos desconocen de este modelo matemático, lo cual a muchos les ahorraría tiempo y mejoraría la eficiencia de lo que se estuviera llevando a cabo, ya que mediante pruebas se podría demostrar que lo que se está realizando funciona de manera correcta y ahora si implementarlo de una mejor manera, pero con pruebas previas.

Así fue realizado este proyecto, a base de datos, teoría y pruebas, lo cual lo hace interesante, ya que se hizo una investigación a fondo para descubrir si nuestro modelo era válido, y llegamos al punto de vista de que, si un autómata puede describir matemáticamente las acciones de un personaje en un videojuego, de la misma manera se puede satisfacer infinidad de necesidades; solo es cuestión de informarnos más acerca de su funcionamiento y de sus aplicaciones para darle el uso correcto.

Con esta parte concluimos con palabras el Planteamiento, Objetivo y Marco Teórico de nuestro Proyecto, ahora viene la demostración por medio de pruebas en distintos softwares que podrán ver y darse cuenta de que el autómata en este contexto cumple su propósito y puede ser más que un solo modelo matemático para aceptar o rechazar cadenas.

## Marco Teórico

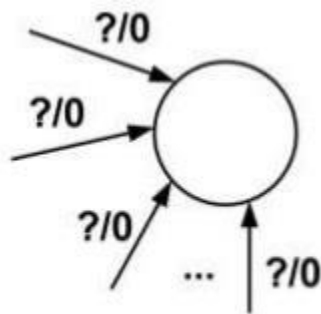
### • TEORIA DE AUTOMATAS

Creo que ya se ha hablado mucho de autómatas y sus modelos, pero ¿Qué es un autómata finito? En resumen, un Autómata es:

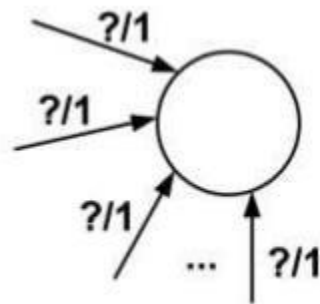
Una maquina (mecanismo) de naturaleza formal que acepta una información de entrada (input), la procesa (la somete a transformaciones simbólicas que pueden adoptar la forma de un cálculo o de una computación), y genera un resultado de salida (output).

Un autómata finito es:

Toda Máquina de Estado Finito en la que el conjunto de símbolos de salida es exclusivamente  $O = \{ 0, 1 \}$  y donde el estado actual determina cuál de ellos fue el último dato de salida. En todo Autómata Finito, representado como A, debe haber cuando menos un estado de aceptación y por sentido común se recomienda que no todos lo sean. En forma gráfica se muestra la forma como se identifican los dos tipos de estado que se pueden presentar en este Autómata. El caracter “?” significa que no importa cuál es el símbolo en la entrada.



**Estado de no aceptación**



**Estado de aceptación**

La segunda característica surge de la necesidad práctica de saber cuál fue el último bit de la cadena de salida, sabiendo en qué estado se termina el recorrido en el diagrama de transición para una cadena de entrada, y sin importar los estados intermedios que se van obteniendo. Si se conoce el estado final, entonces también se deberá conocer cuál fue el último bit de la cadena de salida. En síntesis, es fácil notar que, en muchos casos, lo importante no son las salidas momentáneas del


modelo, sino únicamente el último bit de salida. Lo anterior es muy importante, ya que se presenta un 0 en la salida cuando la cadena de entrada no cumple un enunciado, puesto como especificación de diseño del Autómata; por otro lado, se presenta el 1 en la salida cuando sí lo cumple, tal y como se observa en el reconocedor de cantidades pares de unos que se presentó en el capítulo anterior. Se podría decir que un Autómata Finito es un aceptador o rechazador de cadenas.

Un autómata de estado finito consiste en:

- a) **I** = Un conjunto finito de símbolos de entrada.
- b) **S** = Un conjunto finito de estados.
- c) **F** = Una función transiciones
- d) **A** = Un subconjunto de estados de aceptación.
- e)  $\sigma_0$  = Un estado inicial

Se expresa en notación de conjuntos con la tupla:  $A = (I, S, f, A, \sigma_0)$ .

A ésta se la llamará la descripción formal del Autómata Finito.



$\begin{array}{c} S \\ \backslash \\ I \end{array}$	w	x	y	z
$q_0$	$q_1$	$q_2$	$q_0$	$q_1$
$q_1$	$q_1$	$q_0$	$q_1$	$q_0$
$q_2$	$q_2$	$q_1$	$q_1$	$q_0$

Para el Autómata expresado en la imagen anterior, se tiene:

$I = \{ w, x, y, z \}$        $S = \{ q_0, q_1, q_2 \}$        $A = \{ q_0, q_2 \}$        $\sigma_0 = \{ q_0 \}$   
 $F = \{ f(q_0, w) = q_1, f(q_0, x) = q_2, f(q_0, y) = q_0, f(q_0, z) = q_1, f(q_1, w) = q_1, f(q_1, x) = q_0, f(q_1, y) = q_1, f(q_1, z) = q_0, f(q_2, w) = q_2, f(q_2, x) = q_1, f(q_2, y) = q_1, f(q_2, z) = q_0 \}$ .

- **“UN PERSONAJE EN APUROS”**

En este caso hablaremos acerca de nuestro proyecto, decidimos elegir este tema de un videojuego, ya que, el modelo del autómatas que elaboraremos podría ser una referencia útil para otro proyecto similar, esto debido que muestra las transiciones de estados, entradas y salidas de lo que llevara a cabo el personaje dependiendo de lo que se suscite en la misión encomendada.

Podríamos decir que nuestro proyecto está basado en los movimientos de la vida real en algunos casos, ya que el que personaje de un videojuego puede ir cambiando dependiendo a lo que el desarrollador use en base a su conveniencia, pero los humanos no, entonces es un comportamiento similar al de la vida real donde explicamos a través de las entradas del teclado lo que puede pasar por cada símbolo seleccionado, eso se explicara más adelante en la metodología a realizar.

Basados en esto indicamos el título de esta sección, ya que un autómatas puede sacarnos de apuros, si lo sabemos plantear de una manera correcta y dándole un buen uso a este modelo formal.

Se puede decir también que, aunque el modelo es matemático lo adaptamos a símbolos que es lo que nos facilitará la codificación de código en un lenguaje de programación, adaptando así, las acciones que físicamente hará nuestro personaje en un ambiente computacional que permitirán que esas acciones sean programables.


A continuación, se evidenciará la demostración de su funcionamiento y las pruebas necesarias para describir el comportamiento del personaje en la misión ya mencionada.





## Metodología

- **Autómata Finito**

Tomamos la decisión de que nuestro proyecto constaría de 11 estados, donde 9 de ellos son de aceptación y 2 de rechazo (color rojo), todo esto considerando que nuestro personaje desarrolle un pleno movimiento, estos son: 

Cada uno de estos estados están correlacionados con las entradas y cada una te lleva a un estado diferente, característica de un autómata no determinista, efectuando así, que se puedan generar todos los movimientos posibles hasta caer a un estado de rechazo.

### Estados

0 = Movimiento

1 = Avanzar

2 = Retroceder

3 = Volar

4 = Agachar

5 = Parado

6 = Vuela Avanzando

7 = Vuela retrocediendo

8 = Retrocede agachado

9 = Avanza Agachado

10 = Trampa

Las entradas con las que trabajaremos son las siguientes:

### Entradas

w = volar

a = retroceder

s = agacharse

d = avanzar

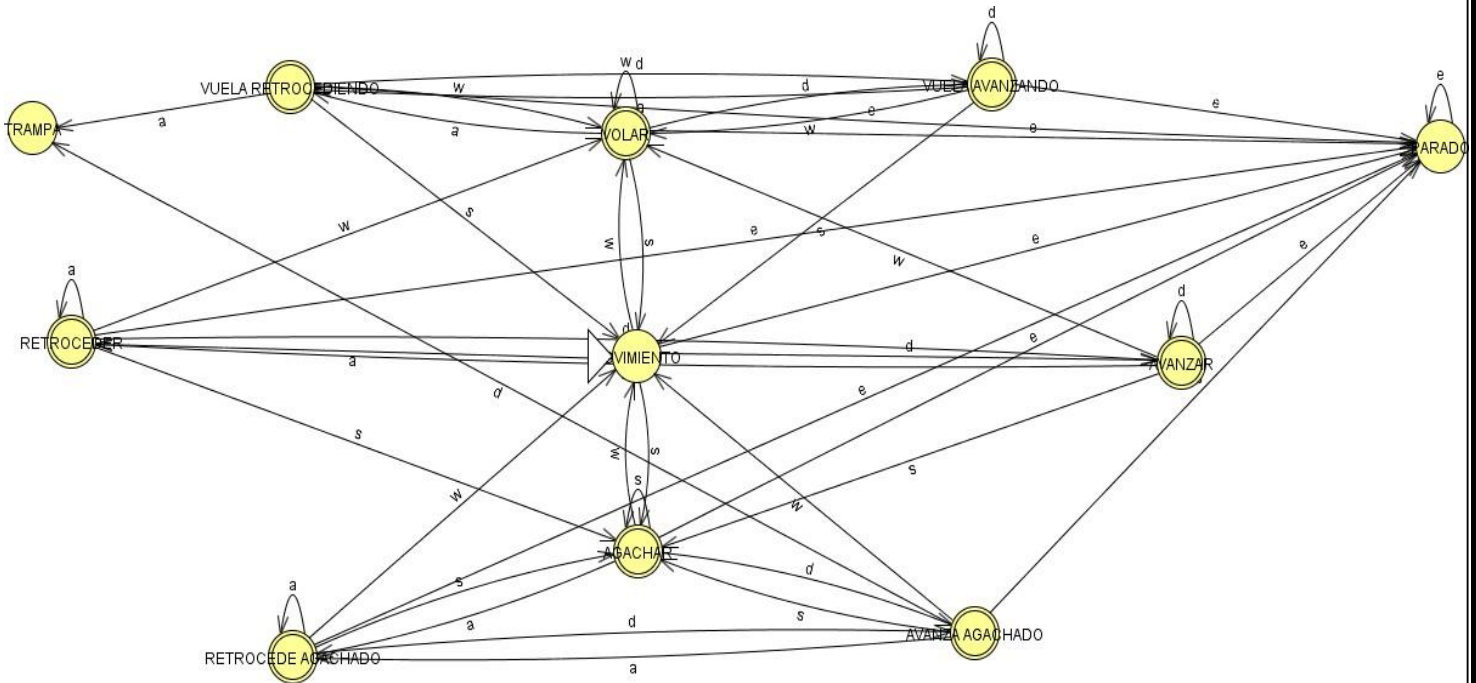
e = parar

x = terminar





Tomamos como estado inicial que el personaje siempre estará en movimiento, y a partir de este puede venir cualquier combinación de estados que el personaje reciba como orden, hasta que lleguemos a un estado de “muerte” que denominamos estado parado o que caiga en una trampa. El diseño de nuestro autómatas se convierte en el siguiente diagrama:



### • Tabla de Transiciones

Con base a nuestro diagrama, se desarrolló la tabla de transiciones, la cual permite saber con exactitud qué cambios de estados surgen a partir de una entrada. La tabla de transiciones resultante es la siguiente:

NOTA: Los estados de aceptación son los circulados

Tabla de transiciones					
s \ l	w	a	s	d	e
Movimiento	Volar	Retroceder	Agachar	Avanzar	Parado
Avanzar	Volar	Retroceder	Agachar	Avanzar	Parado

Retroceder	Volar	Retroceder	Agachar	Avanzar	Parado
Volar	Volar	Vuela retrocediendo	Movimiento	Vuela avanzando	Parado
Agachar	Movimiento	Retrocede agachado	Agachar	Avanza agachado	Parado
Parado	Parado	Parado	Parado	Parado	Parado
Vuela avanzando	Volar	Vuela retrocediendo	Movimiento	Vuela avanzando	Parado
Vuela retrocediendo	Movimiento	Trampa	Movimiento	Vuela avanzando	Parado
Retrocede agachado	Movimiento	Retrocede agachado	Agachar	Avanza agachado	Parado
Avanza agachado	Movimiento	Retrocede agachado	Agachar	Trampa	Parado
Trampa	Trampa	Trampa	Trampa	Trampa	Trampa

- **Estructura de Datos Bidimensional**

Para construir nuestra matriz, fue necesario realizar una asociación entre números y entradas, con el fin de permitirnos acceder a cada elemento de la matriz, dado que nuestra tabla consta de [estados][entradas].

Estados	Entradas
0 = Movimiento	w=0
1 = Avanzar	a=1
2 = Retroceder	s=2
3 = Volar	d=3
4 = Agachar	e=4
5 = Parado	
6 = Vuela Avanzando	
7 = Vuela retrocediendo	
8 = Retrocede agachado	
9 = Avanza Agachado	
10 = Trampa	

A partir de esta tabla de transiciones llegamos a la siguiente estructura de datos bidimensional basada en los posibles recorridos que implementaba nuestro autómatas finitos mostrados anteriormente, siendo las celdas sombreadas en rojo las transiciones que llegaron a un estado de no aceptación o de rechazo:

TABLA DE TRANSICION				
S(0,0) = 3	S(0,1) = 2	S(0,2) = 4	S(0,3) = 1	S(0,4)=5
S(1,0) = 3	S(1,1) = 2	s(1,2) = 4	s(1,3) = 1	S(1,4)=5
S(2,0) = 3	S(2,1) = 2	s(2,2) = 4	s(2,3) = 1	S(2,4)=5
S(3,0) = 3	s(3,1) = 7	s(3,2) = 0	s(3,3)= 6	s(3,4)=5
s(4,0) = 0	s(4,1) = 8	s(4,2) = 4	s(4,3) = 9	s(4,4)=5
s(5,0) =5	s(5,1) =5	s(5,2) = 5	s(5,3) =5	s(5,4)=5
s(6,0) = 3	s(6,1) = 7	s(6,2) = 0	s(6,3) = 6	s(6,4)= 5
s(7,0) = 0	s(7,1) = 10	s(7,2) = 0	s(7,3) = 6	s(7,4)=5
s(8,0) =0	s(8,1) = 8	s(8,2) = 4	s(8,3) = 9	s(8,4) =5
s(9,0) =0	s(9,1) = 8	s(9,2) = 4	s(9,3) =10	s(9,4)= 5
s(10,0)=10	s(10,1)=10	s(10,2)=10	s(10,3)=10	s(10,4) = 10

Siendo cada celda representada de la siguiente manera:

Transición	Estado	Entrada	Estado Siguiente
S	(0,	1)	3

Sin pasar por alto que por medio de los estados y la asociación que hemos realizado de las entradas con los números, podremos acceder al estado siguiente y hacer posible el cambio de transiciones.

- **Pseudocódigo**

```
CADENA nombre
ENTERO input
CARACTER opc
ENTERO estado <- 0

ENTERO tabla[ESTADOS][ENTRADAS]<-{{3, 2, 4, 1, 5},
                                     {3, 2, 4, 1, 5},
                                     {3, 2, 4, 1, 5},
                                     {3, 7, 0, 6, 5},
                                     {0, 8, 4, 9, 5},
                                     {5, 5, 5, 5, 5},
                                     {3, 7, 0, 6, 5},
                                     {0,10, 0, 6, 5},
                                     {0, 8, 4, 9, 5},
                                     {0, 8, 4,10, 5},
                                     {10,10,10,10,10}};

HACER
  INICIO
    ESCRIBIR "Ingresa una tecla: "
    LEER opc
    SELECCION (opc)
    INICIO
      OPCION == 'w':
        input <- 0;
        ROMPER
      OPCION == 'a':
        input <- 1;
        ROMPER
      OPCION == 's':
        input <- 2;
        ROMPER
      OPCION == 'd':
        input <- 3;
        ROMPER
      OPCION == 'e':
        input <- 4;
        ROMPER
      OPCION == 'x':
        ESCRIBIR "El juego ha terminado "
        REGRESA 0;
        ROMPER
```

*OPCION* SiNo:

ESCRIBIR "Opcion no valida"

*ROMPER*

*FIN*

estado <- tabla[estado][input];

*SI*(estado==0)

*INICIO*

ESCRIBIR "personaje esta en movimiento"

ESCRIBIR "Se encuentra en el estado q" estado

*FIN*

*SINO* (estado==1)

*INICIO*

ESCRIBIR "personaje avanza"

ESCRIBIR "Se encuentra en el estado q" estado

*FIN*

*SINO* (estado==2)

*INICIO*

ESCRIBIR "personaje esta en retrocede"

ESCRIBIR "Se encuentra en el estado q" estado

*FIN*

*SINO* (estado==3)

*INICIO*

ESCRIBIR "personaje esta en vuela"

ESCRIBIR "Se encuentra en el estado q" estado

*FIN*

*SINO* (estado==4)

*INICIO*

ESCRIBIR "personaje esta agachado"

ESCRIBIR "Se encuentra en el estado q" estado

*FIN*

*SINO* (estado==6)

*INICIO*

ESCRIBIR "personaje avanza volando"

ESCRIBIR "Se encuentra en el estado q" estado

*FIN*

*SINO* (estado==7)

*INICIO*

ESCRIBIR "personaje retrocede volando"

ESCRIBIR "Se encuentra en el estado q" estado

```

FIN
SINO (estado==8)
INICIO
    ESCRIBIR "personaje retrocede agachado"
    ESCRIBIR "Se encuentra en el estado q" estado
FIN
SINO (estado==9)
INICIO
    ESCRIBIR "personaje avanza agachado"
    ESCRIBIR "Se encuentra en el estado q" estado
FIN
SINO (estado==10)
INICIO
    ESCRIBIR "personaje ha caído en una trampa"
    ESCRIBIR "Cayo en el estado q" estado
    ESCRIBIR "GAME OVER"
FIN
SINO (estado ==5)
INICIO
    ESCRIBIR "personaje se quedó parado y murió a causa de una
trampa"

    ESCRIBIR "Cayo en el estado q" estado
    ESCRIBIR "GAME OVER"
FIN
FIN MIENTRAS (opc != 'x' && estado != 5 && estado != 10 );
REGRESA 0;
FIN

```

## Pruebas y Resultados

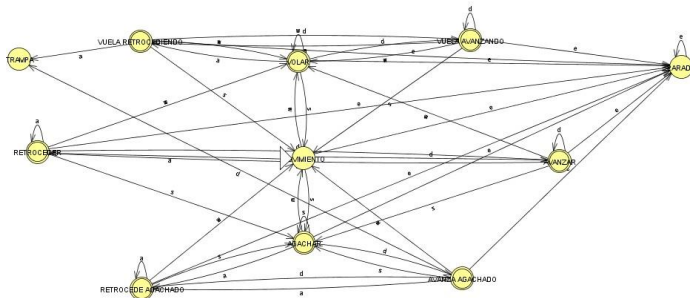


Table Text Size	
Input	Result
adwsdaswd	Accept
awsdse	Reject
awdsd	Reject

### ***Cadena 1:*** adwsdaswd

Esta cadena llega a un estado de aceptación, el recorrido es:

movimiento -> retroceder -> avanzar -> volar -> movimiento -> avanzar -> retroceder  
-> agachar -> movimiento -> avanzar

Esto quiere decir que el personaje sobrevivió a las trampas y no se quedó parado, llegando a un estado de aceptación como antes ya lo habíamos mencionado.

### ***Cadena 2:*** awsdse

Esta cadena llega a un estado de rechazo, el recorrido es:

movimiento -> retroceder -> volar -> movimiento -> avanzar -> agachar -> parado

Esto quiere decir que el personaje se quedó parado y no sobrevivió, llegando así a un estado de rechazo del que no puede salir.

### ***Cadena 3:*** awsdsdd

Esta cadena llega a un estado de rechazo, el recorrido es:

movimiento -> retroceder -> volar -> movimiento -> avanzar -> agachar ->

avanza agachado -> trampa

Esto quiere decir que nuestro personaje cayó en una trampa por permanecer agachado tanto tiempo, llegando así a un estado de rechazo del que no puede salir.

En resumen, nuestro autómata acepta cualquier cadena que no contenga una “e”, de la misma manera, no aceptará las que contengan las secuencias “aa” mientras esta en el estado “volar” (q3) o las secuencias “dd” mientras esta en el estado “avanza agachado” (q9), ya que con estos hacemos referencia a las trampas.

***Link del video ejecutando el código:***

<https://www.loom.com/share/0f064072bbf44fcf8bb236b049cd643f>



## **Conclusiones**

Ya culminando el análisis podemos concluir que el diseño de nuestro autómatas cumplió con las expectativas de la simulación de movimientos virtuales del personaje, demostrando con esto que por medio de un Autómata Finito es posible desarrollar un videojuego con base a una serie de instrucciones que nosotros deseemos, además, de que nuestro modelo es eficiente en cuanto a utilidad y funcionamiento, esto debido a que la manera en que lo implementamos bajo una estructura de datos permitió la fácil comprensión y el ahorro de recursos con nuestro ejemplar.

Con respecto a que si nuestro autómatas se puede mejorar, consideramos que en el futuro podía llegar a ser el pilar para nuevos proyectos en áreas de los videojuegos, dando pie a que se pueda innovar y modificar, como, por ejemplo, agregando nuevas acciones al personaje, nuevos modos de juego o incluso mejorando el prototipo ya elaborado. Lo que quiere decir que el proyecto es un objeto de estudio versátil que está en condiciones de responder ante distintos desafíos y de adaptarse a todo tipo de contexto según las necesidades.

Al principio de nuestro proyecto, creíamos imposible el poder incorporar movimientos a nuestro personaje utilizando un autómatas. Sin embargo, estudiando la interacción con el modelo, haciendo un análisis profundo de los cambios de transición cada que llegaba una entrada y aparte su implementación en un lenguaje de programación pudimos comprobar que todo lo que imaginamos se puede llegar a programar y por supuesto que si cumplió y sobrepasó nuestras expectativas con respecto a los movimientos.

Este proyecto ha permitido evidenciar que los autómatas son más importantes de lo que creemos, pues gracias a estos se han podido diseñar cientos de máquinas y artefactos potentes, productivos y ergonómicos que hacen nuestra vida fácil y todo esto gracias el poder de la mente humana y la lógica con la ayuda de la tecnología.

## **Bibliografía.**

- Gómez Andrade Abelardo. (2014). Introducción a la Teoría de Autómatas y Lenguajes Formales. CUCEI: TRAUCO. Departamento de Lógica.
- Malpica Norberto. (2014). Síntesis de circuitos secuenciales síncronos: Máquinas de estados finitos. 21/06/21, de Universidad Rey Juan Carlos Sitio web: [https://www.cartagena99.com/recursos/alumnos/apuntes/Tema7\\_SED\\_1415%20\(1\).pdf](https://www.cartagena99.com/recursos/alumnos/apuntes/Tema7_SED_1415%20(1).pdf)
- Martínez Gloria. (2005). Apuntes de Teoría de Autómatas y Lenguajes Formales. 21/06/21, de No especificado Sitio web: [http://repositori.uji.es/xmlui/bitstream/handle/10234/5995/parte\\_1.pdf?sequence=1&isAllowed=y](http://repositori.uji.es/xmlui/bitstream/handle/10234/5995/parte_1.pdf?sequence=1&isAllowed=y)
- Ruscitti Hugo. (2005). Autómatas y animaciones. 21/06/21, de LosersJuegos Sitio web: <http://www.losersjuegos.com.ar/referencia/articulos/automatas>
- UNAM. (S/F). Problemas y Algoritmos. 21/06/21, de Unidad de Apoyo para el Aprendizaje Sitio web: [https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/1163/mod\\_resource/content/1/contenido/index.html](https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/1163/mod_resource/content/1/contenido/index.html)
- (S/A). (S/F). Autómatas y Computabilidad. 21/06/21, de UCM Sitio web: <https://webs.ucm.es/info/pslogica/automatas.pdf>
- S/A. (2011). Ventajas / Desventajas de NFA sobre DFA y viceversa. 21/06/21, de La Ingenieria Sitio web: <https://laingenieria.info/questions/14691/ventajas-desventajas-de-nfa-sobre-dfa-y-viceversa>