INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

Teoría Computacional

Proyecto “Máquina de Turing”

Profesora: Luz María Sánchez García

Grupo: 2CM1

Alumnos:

Mendoza Parra Sergio.

Paz Sánchez Brandon.

Macías Castillo Josué.

Aguilar Hernández Irving Arturo.

MEXICO, D.F. a 4 de Junio del 2017

**Introducción:**

Una máquina de Turing es un dispositivo que manipula símbolos sobre una tira de cinta de acuerdo a una tabla de reglas. A pesar de su simplicidad, una máquina de Turing puede ser adaptada para simular la lógica de cualquier [algoritmo](https://es.wikipedia.org/wiki/Algoritmo) de [computador](https://es.wikipedia.org/wiki/Computador) y es particularmente útil en la explicación de las funciones de una [CPU](https://es.wikipedia.org/wiki/Unidad_central_de_procesamiento) dentro de un computador.

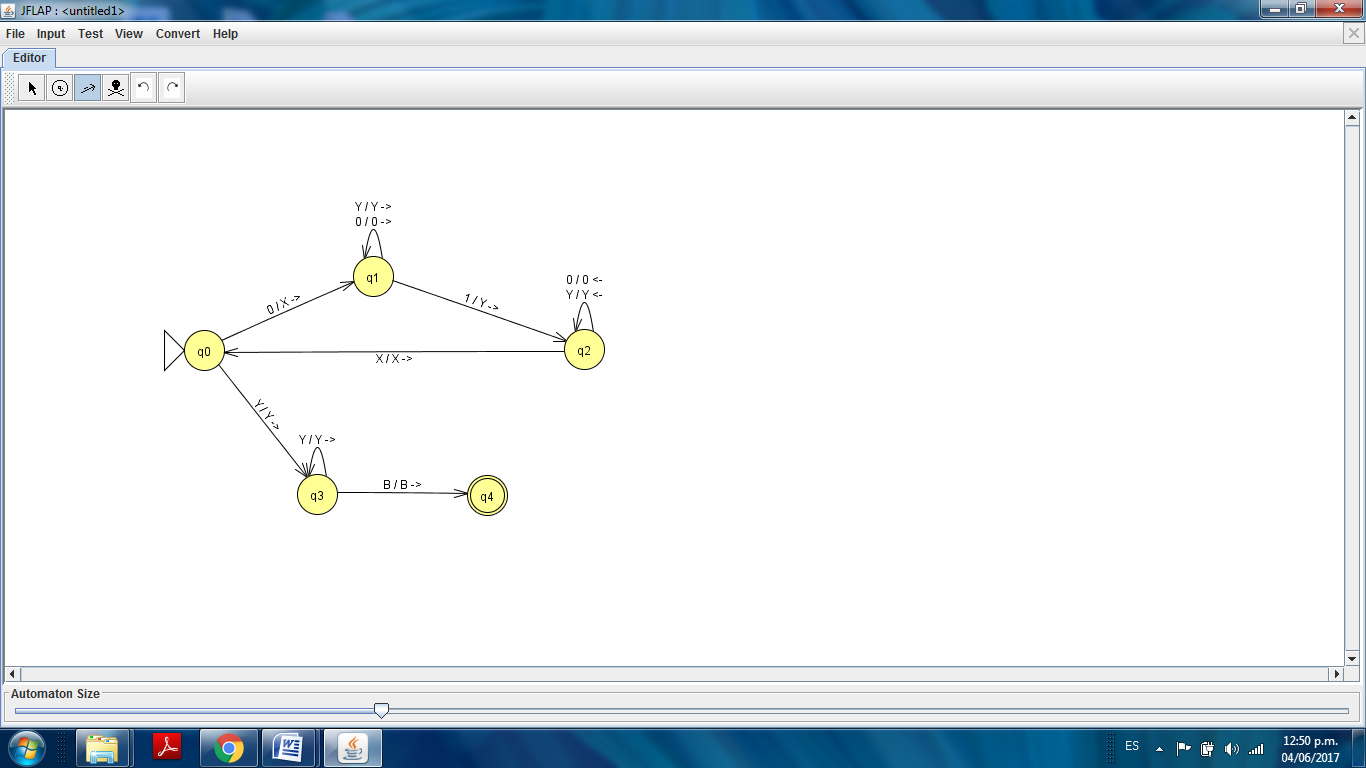
Originalmente fue definida por el matemático inglés [Alan Turing](https://es.wikipedia.org/wiki/Alan_Turing) como una máquina automática en 1936, en la revista *[Proceedings of the London Mathematical Society](https://es.wikipedia.org/wiki/Proceedings_of_the_London_Mathematical_Society" \o "Proceedings of the London Mathematical Society)*. La máquina de Turing no está diseñada como una tecnología de computación práctica, sino como un dispositivo hipotético que representa una [máquina de computación](https://es.wikipedia.org/wiki/Computador). Las máquinas de Turing ayudan a los científicos a entender los límites del cálculo mecánico.

Turing dio una definición sucinta del experimento en su ensayo de 1948, Máquinas inteligentes. Refiriéndose a su publicación de 1936, Turing escribió que la máquina de Turing, aquí llamada una máquina de computación lógica que es capaz de simular cualquier otra máquina por lo cual se le conoce como [máquina universal de Turin](https://es.wikipedia.org/wiki/M%C3%A1quina_universal_de_Turing)g.

Por otro lado en este proyecto se dará a conocer el funcionamiento de una maquina de Turing a partir de la teoría de maquinas de Turing, también daremos algunos ejemplos con sus respectivas explicaciones y también explicaremos cómo funciona el autómata de la máquina de Turing.

**Planteamiento del Problema:**

Escribir un programa que realice la simulación de una máquina de turing a partir del autómata de la máquina de Turing(ver imagen).



Autómata de la Maquina de Turing

El programa tendrá como entrada un lenguaje definido de puros “0”’s y “1”’s, de este modo el lenguaje para las cadenas validas para ésta máquina de Turing es el siguiente:

También hay que tomar en cuenta las cadenas que son y no son validas para este ejemplo.

**Diseño y funcionamiento de la solución:**

Vamos a diseñar una máquina de Turing y a ver cómo se comporta con una entrada típica. La MT que vamos a construir aceptará el lenguaje que ya predefinimos anteriormente. Inicialmente, se proporciona a la cinta una secuencia finita de ceros y unos, procedida y seguida por secuencias infinitas de espacios en blanco. Alternativamente la MT cambiará primero un 0 por X y luego un 1 por una Y, hasta que se hayan cambiado todos los ceros y los unos.

Más detalladamente, comenzando por el extremo izquierdo de la entrada, se cambia sucesivamente un 0 por una X y se mueve hacia la derecha pasando por encima de todos los ceros y las letras Y que se ve, hasta encontrar una X. En esta situación, busca un 0 colocado inmediatamente a la derecha, y si lo encuentra, lo cambia por una X y repite el proceso, cambiando el 1 correspondiente por una Y.

Si la entrada no es de la forma 0\*1\*, entonces la MT terminará no haciendo el siguiente movimiento y se detendrá sin aceptar. Sin embargo, si termina cambiando todos los ceros por X en la misma iteración en la que cambia el último 1 por una Y, entonces determina que la entrada era de la forma y acepta. La especificación formal de la máquina de Turing M es:

Donde β se especifica en la siguiente tabla:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Símbolos | | | | | |
| Estados | 0 | 1 | X | Y | B |
| q0 | (q1, X, R) | - | - | (q3, Y, R) | - |
| q1 | (q1, 0 , R) | (q2, Y, L) | - | (q1, Y, R) | - |
| q2 | (q2, 0, L) | - | (q0, X, R) | (q2, Y, L) | - |
| q3 | - | - | - | (q3, Y, R) | (q4, B, R) |
| q4 | - | - | - | - | - |

Tabla de transición de la Máquina de Turing

Mientras M realiza las operaciones anteriores, la parte de la cinta que ya ha sido recorrida por la cabeza de la misma corresponderá siempre a una secuencia de símbolos descrita por la expresión regular X\*0\*Y\*1\*. Es decir, habrá ceros que han sido sustituidos por X, seguidos de ceros que todavía no han sido sustituidos por X. Luego se encontraran con algunos unos que han sido sustituidos por una Y, y unos que todavía no han sido. A continuación, puede que haya o no algunos ceros y unos.

El estado q0 es el estado inicial y M entra en el estado q0 cada vez que vuelve al cero mas a la izquierda que queda. Si M está en el estado q0 y se señala un 0, la regla de la esquina superior izquierda de la tabla de transiciones indica que M tiene que pasar al estado q1, cambia el 0 por una X y moverse hacia la derecha. Una vez que permaneciendo en el estado q1. Si M ve una X o una B, se detiene. Sin embargo, si M ve un 1estando en el estado q1, cambia dicho 1 por una Y, pasa al estado q2 y comienza a moverse hacia la izquierda.

En el estado q2. M se mueve hacia la izquierda pasando por encima de los ceros y las Y, permaneciendo en el estado q2. Cuando M alcanza la X que esta mas a la derecha, la cual marca el extremo derecho del bloque de ceros que ya han sido cambiados por X, M vuelve al estado q0 y se mueve hacia la derecha. Hay dos casos:

1. Si ahora M ve un 0, entonces repite el ciclo de sustituciones que acabamos de describir.
2. Si M ve una Y, entonces es que ha cambiado todos los ceros por X. Si todos los unos han sido cambiados por Y, entonces la entrada era de la forma , y M acepta. Por tanto, M pasa al estado q3, y comienza a moverse hacia la derecha, pasando por encima de las Y. Si el primer símbolo distinto de una Y que M encuentra un espacio en blanco, entonces existirá el mismo número de ceros que de unos, por lo que M entra en el estado q4 y acepta. Por otro lado, si M encuentra otro 1, entonces la entrada era de la forma errónea y M también se detiene.

Explicación de ejemplos de cadenas:

|  |  |
| --- | --- |
| Cadenas | |
| Validas | No Validas |
| 0011 | 010101 |
| 01 | 1100 |
| 000111 | 0010 |

He aquí un ejemplo de un cálculo de aceptación de M. Su entrada es **0011**. Inicialmente, M se encuentra en el estado q0. Señalando al primer 0, es decir la configuración inicial de M es q00011. La secuencia completa de movimientos de M es:

q00011 => X q1011 => X0 q111 => X q20 Y1 => q2X0Y1 => X q00Y1 => XX q1Y1 =>

XX Y q11 => XX q2YY => X q2XYY => XXq0YY => XXY q3Y => XXYY q3B=>

XXYYB q4B Cadena Valida

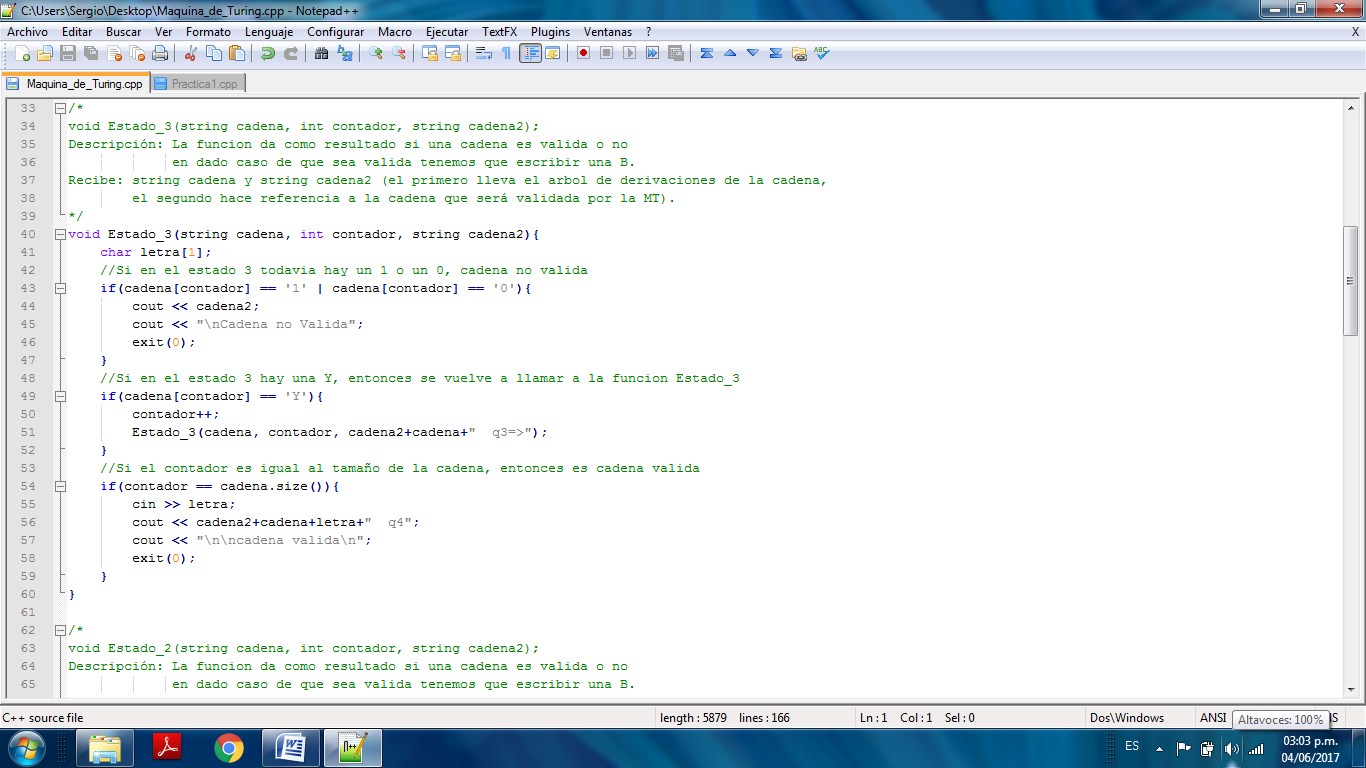
Veamos un ejemplo similar. Consideremos lo que hace M para la entrada 0010, que no pertenece al lenguaje aceptado.

q00010 => X q1010 => X0 q110 => X q20 Y0 => q2X0Y0 => X q00Y0 => XX q1Y0 =>

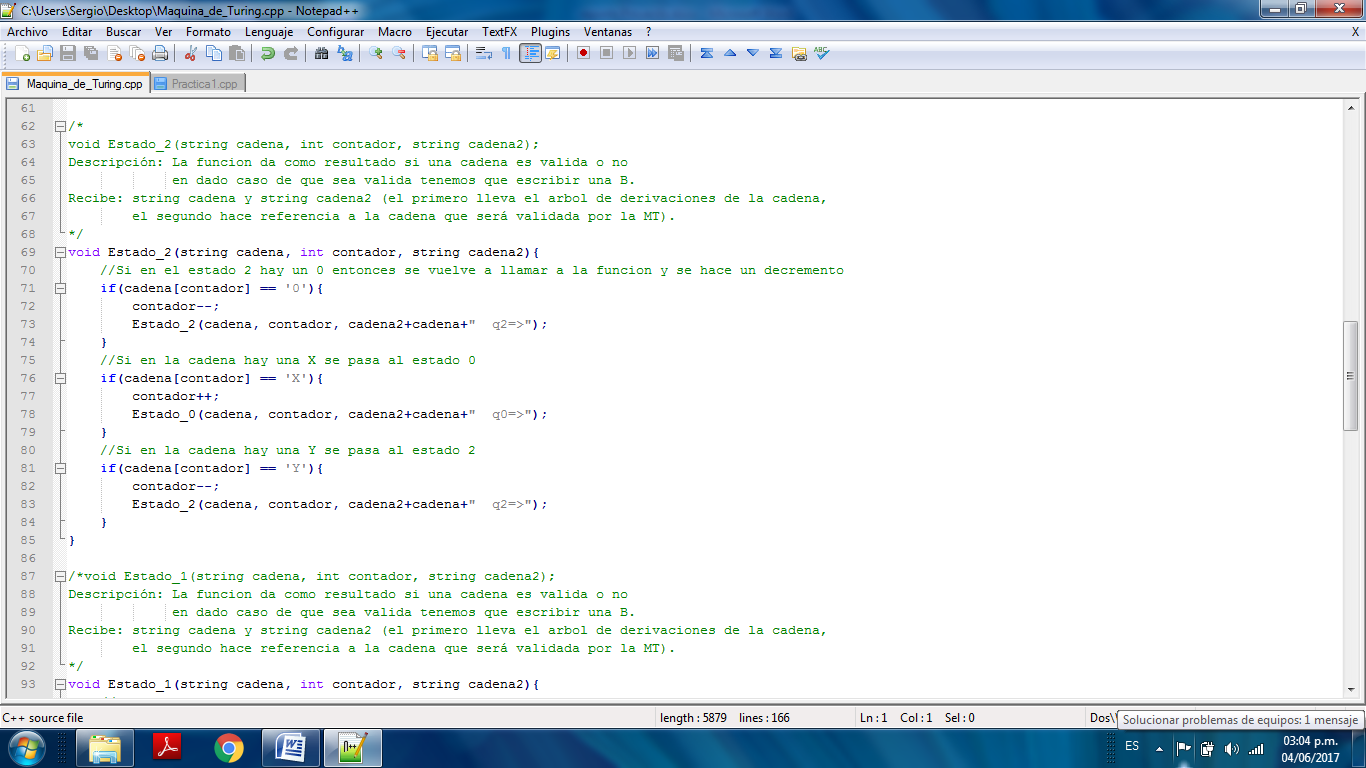
XX Y q10 => XX Y0 q1 B Cadena no Válida

El comportamiento de M para 0010 se parece al comportamiento para 0011, hasta que llega a la configuración XX Y q10 M y señala al 0 final por primera vez. M tiene que moverse hacia la derecha permaneciendo en el estado q1, lo que corresponde a la configuración XX Y0 q1 B. Sin embargo, en el estado q1, M no se realiza ningún movimiento si el símbolo de la entrada al que señala es B; por tanto, M deja de funcionar y no acepta su entrada.

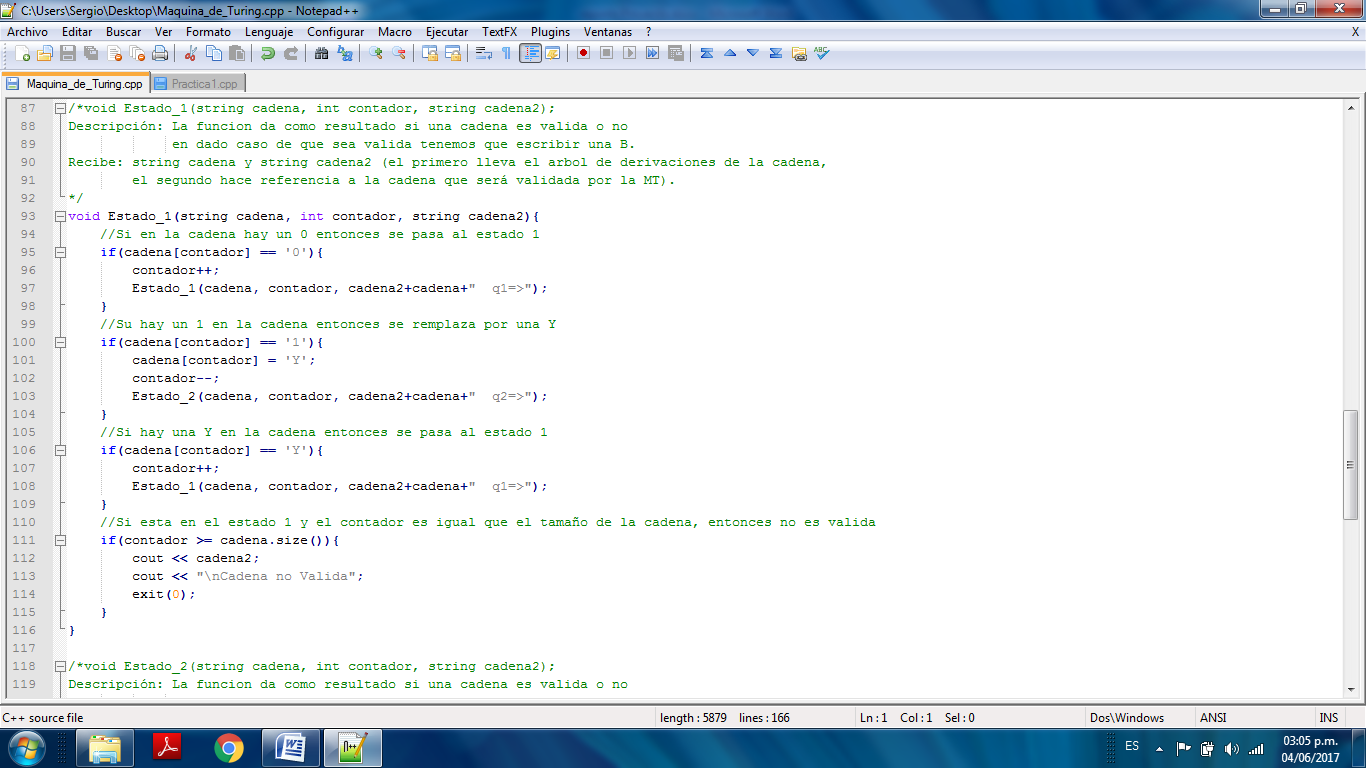
**Implementación de la Solución en Código:**



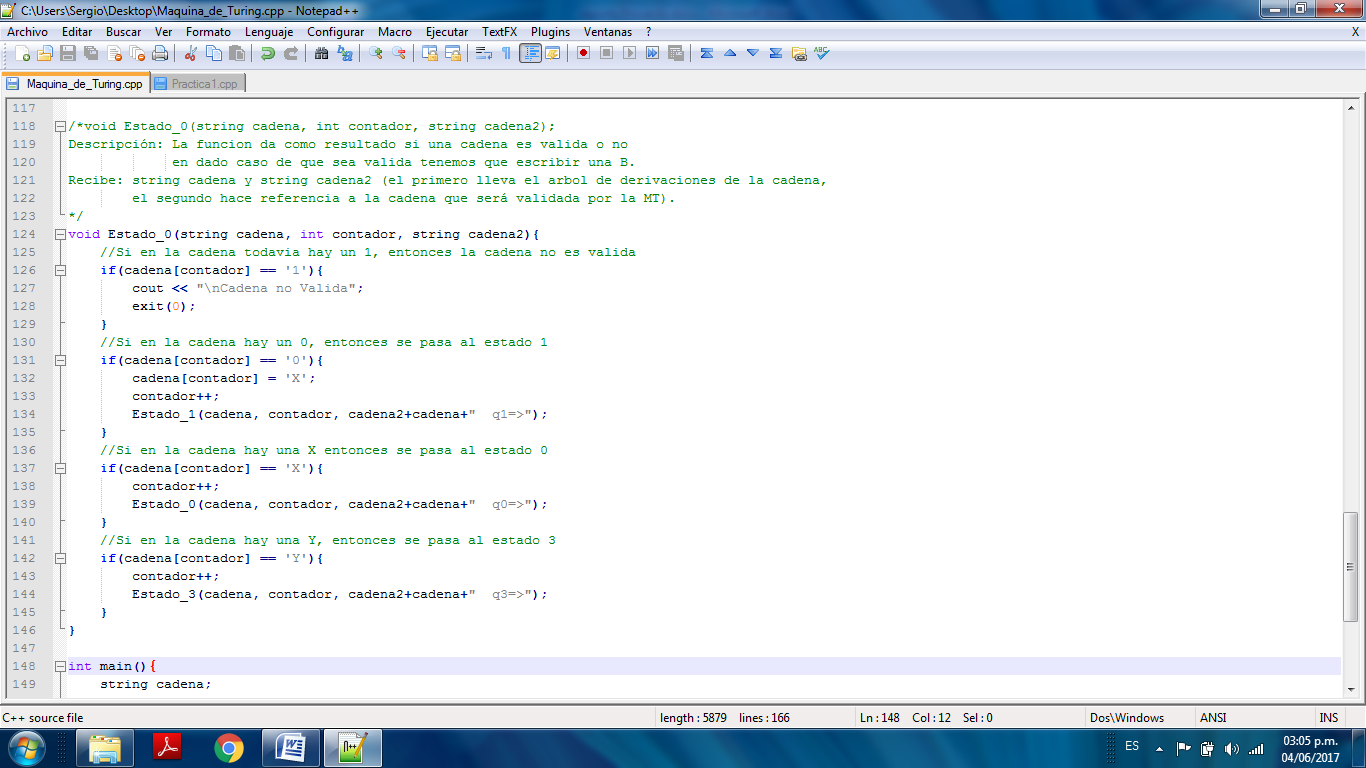
Este bloque lo que hace es validar en el estado 3 si todavía hay números que no han sido evaluados, en este caso si todavía hay “unos” o “ceros” automáticamente es una cadena no valida o si el contador es igual al tamaño de la cadena leemos un carácter para decir que es cadena valida.



Este bloque lo que hace es validar las letras con los caracteres ya que si hay un “cero” en la cadena se hace un decremento para evaluar de nuevo la cadena o de otro modo si hay una “X” o “Y” se mandan a sus respectivos estados para validar .



Este bloque es igual al anterior solo que este tiene un detector de error porque si la cadena pasa por ciertos estados y todavía no es validada como cadena aceptada entonces decimos que es cadena no valida.

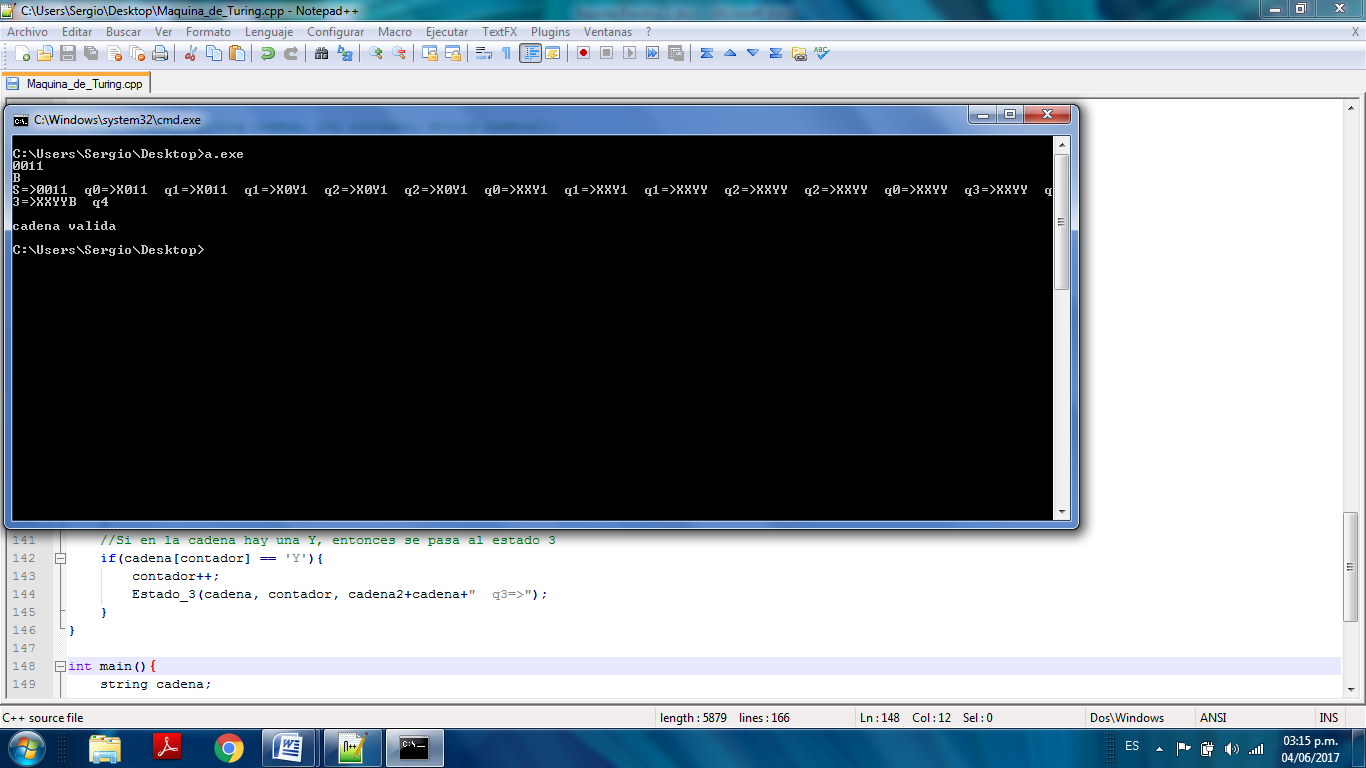


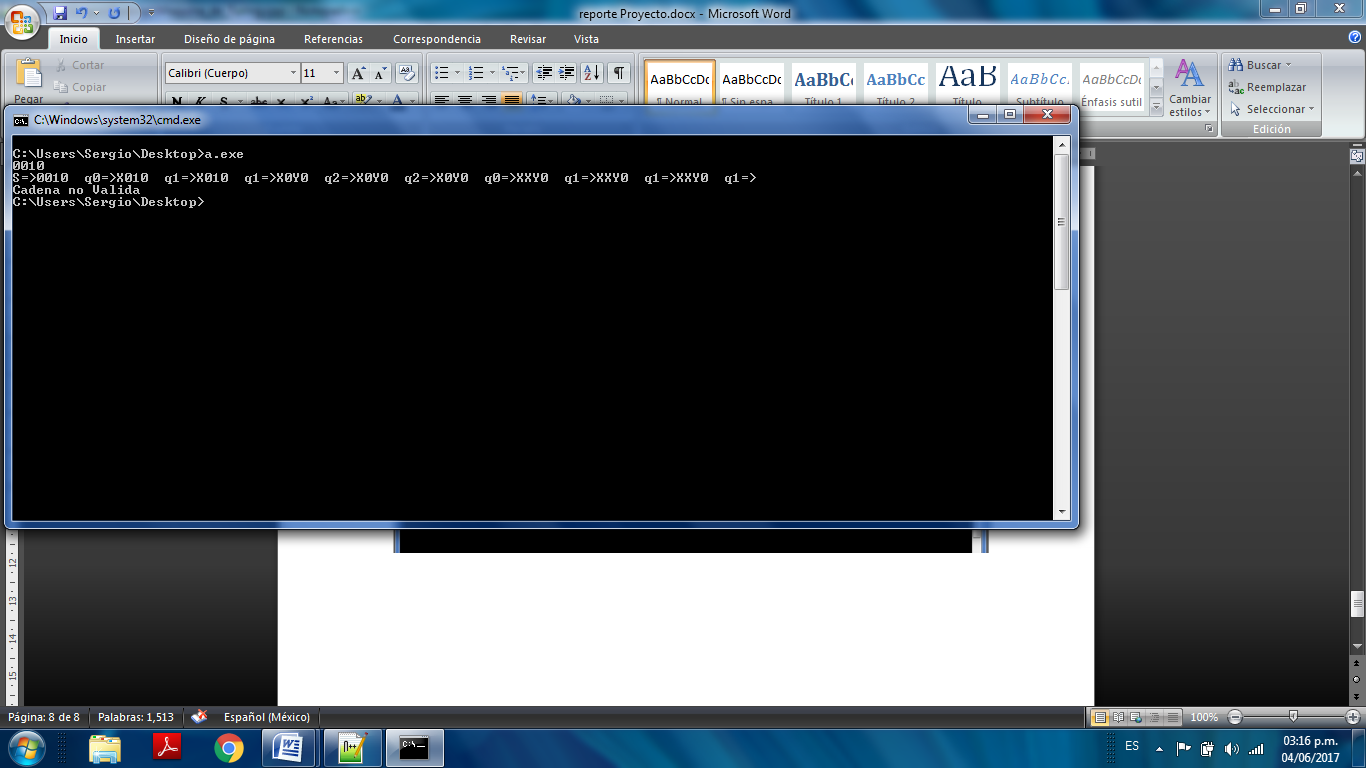
Este último es el estado inicial el cual al igual que los demás, solo evalúa los estados por los que va a pasar la cadena.

**Funcionamiento en terminal**

Una vez que programamos todo lo anterior procedemos a la compilación del programa en el cual vamos a tener una cadena de entrada y por lo tanto se va a validar, como se dijo anteriormente si es válida o no.

Aquí vamos a ingresar los ejemplos que habíamos descrito en anteriormente poniendo la cadena 0011 y 0010, respectivamente:





Aquí podemos ver que ambos ejemplos tienen el mismo árbol de derivaciones como lo habíamos descrito anteriormente.

**Errores detectados**

1. Si se pone una cadena muy larga puede que tenga tiempo límite excedido ya que la en el administrador de tareas con poner una cadena no larga, necesita mucho recurso de la computadora y por lo consiguiente si es una cadena larga puede que no compile.

**Posibles mejoras**

Pues una de las mejoras que se podrían hacer en ingresar los estados por los que pasa la cadena pero insertando en una cadena nueva así como lo descrito en los ejemplos de las cadenas en la sección de “diseño e implementación de la solución”.

**Conclusión**

El proyecto se me hizo interesante ya que las maquinas de Turing son utilizadas para simular procesos de una CPU, de este modo para este caso solo se hizo una maquina que solo cambia algunos números por letras de hecho se podría también hacer un tipo de criptografía dependiendo de cuanto unos y cerros halla para poder mandar un mensaje y al mismo tiempo descencriptarlo de este modo las maquinas de Turing son muy implementadas en la parte de inteligencia artificial o inclusive en la parte de criptografía ya que es una maquina que puede encriptar o descencriptar dependiendo su uso.