

**INSTITUTO POLITÉCNICO NACIONAL**

**ESCUELA SUPERIOR DE CÓMPUTO**

**TEORÍA COMPUTACIONAL**

**2CM4**

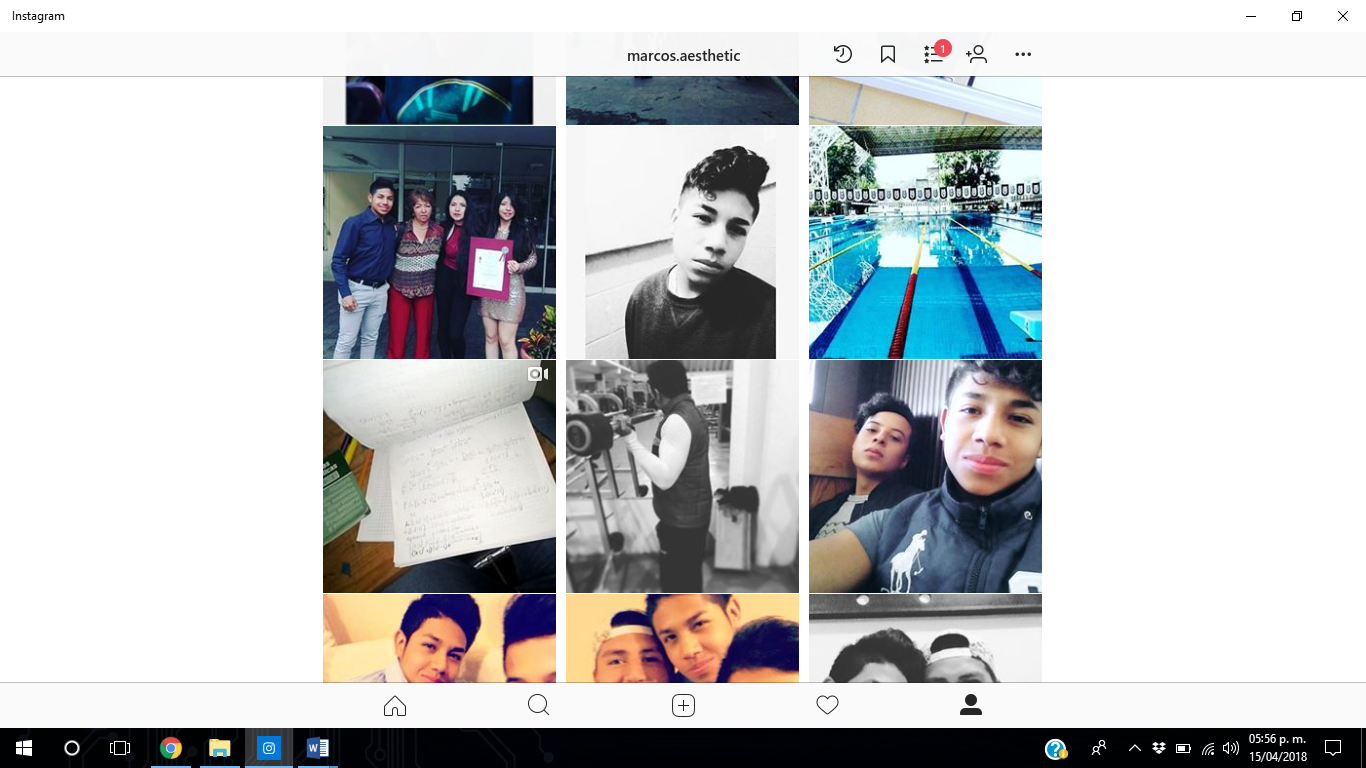
**PROFESOR: LUZ MARÍA SÁNCHEZ GARCÍA**

**PRÁCTICA 6 AUTÓMATA DE PILA**

**INTEGRANTES:**

**VÁZQUEZ MORENO MARCOS OSWALDO 2016601777**

**QUINTANA RUÍZ AJITZI RICARDO 2017631261**

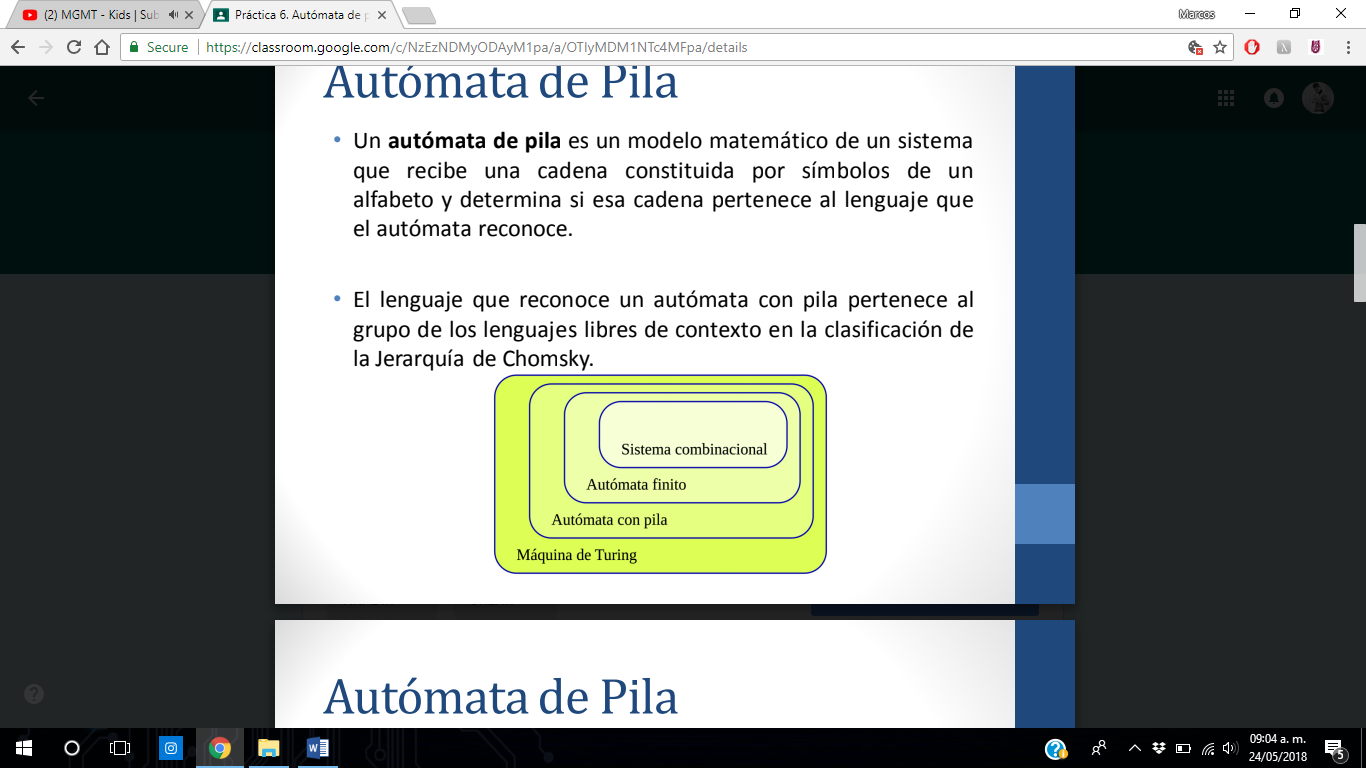


**FECHA DE ENTREGA: 24 DE MAYO DE 2018**

**INTRODUCCIÓN**

En la siguiente práctica se pretende codificar un programa que simule la ejecución de un autómata de pila el cual es un modelo matemático de un sistema que recibe una cadena constituida por símbolos de un alfabeto y determina si esa cadena pertenece al lenguaje que el autómata reconoce.

A su vez, el lenguaje que reconoce un autómata con pila pertenece al grupo de lenguajes libres de contexto en la clasificación de la jerarquía de Chomsky.



**PLANTEAMIENTO DEL PROBLEMA**

Implementar el algoritmo de codificación de un programa el cual permita ingresar una cadena de caracteres con un lenguaje definido por:

**{a nbn donde n>0}**

En donde acepte los siguientes casos:

1.- ab

2.-aabb

3.-aaabbb

Y no acepte los siguientes:

1.-ba 8.-baba

2.-aba 9.-aabbaa

3.-baa 10.-aabaa

4.-abb 11.-baaab

5.-abba 12.-abaaba

6.-aaab

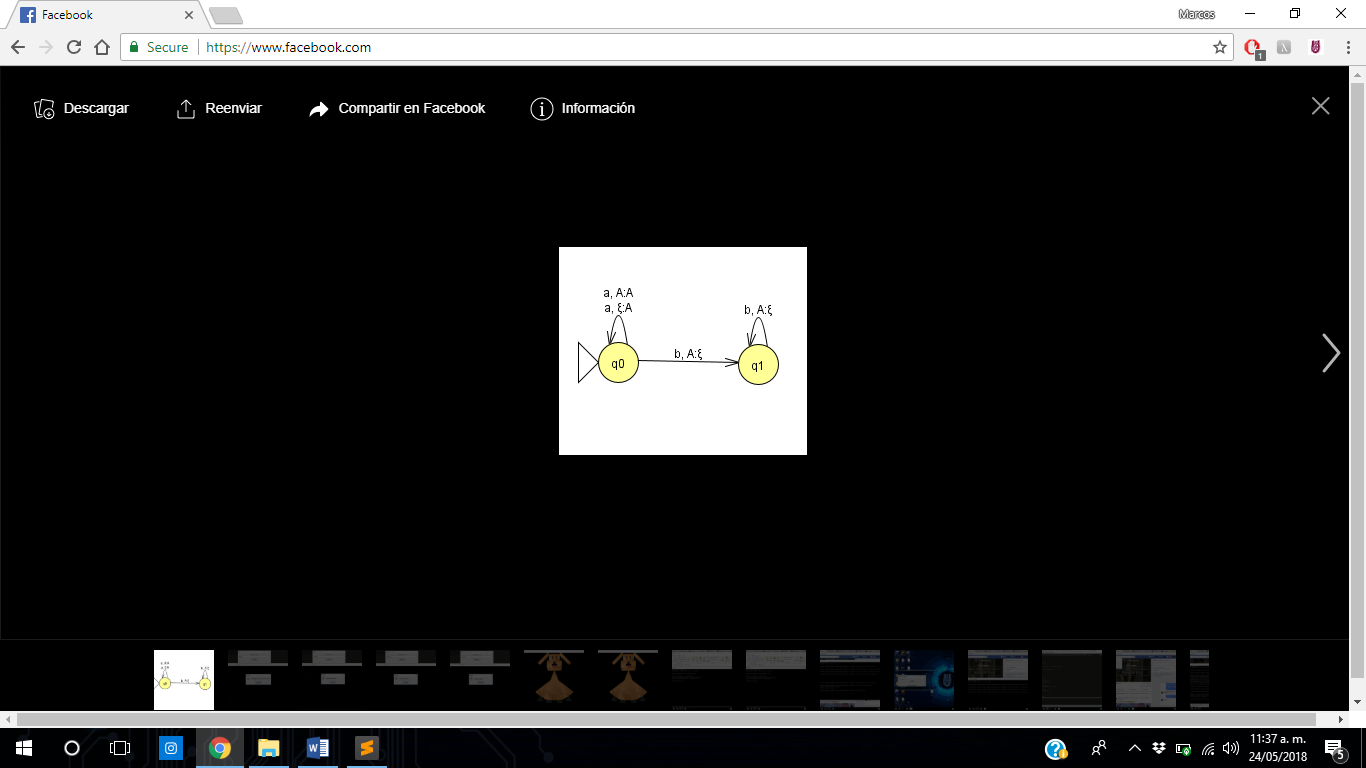
7.-abab

Teniendo de esta manera validación de cadenas con un método distinto, obteniendo en cada inserción una pila nueva, donde ella misma sabrá de acuerdo con nuestro autómata programado si podemos o no aceptar la cadena, si la pila debe de estar vacía, si podemos o no insertar el símbolo de pila correspondiente.

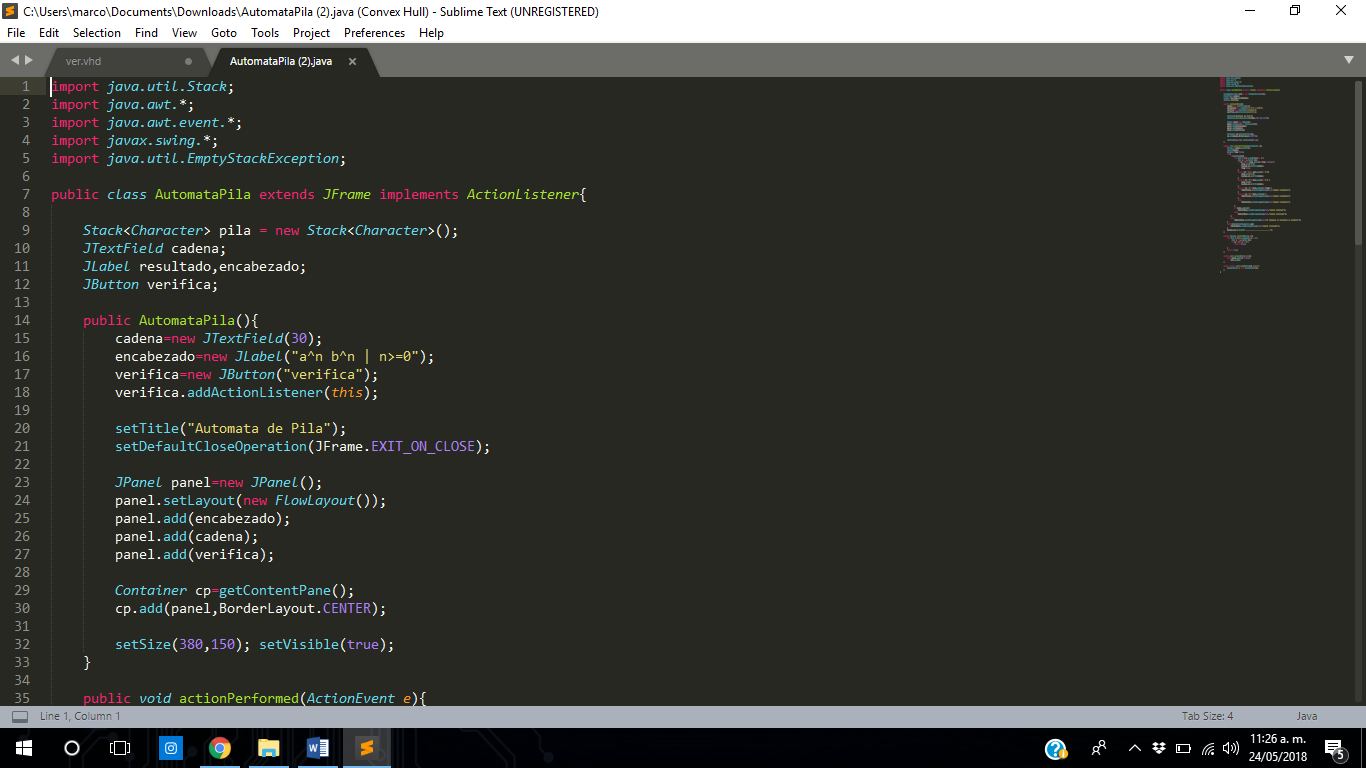
**DISEÑO DE LA SOLUCIÓN**

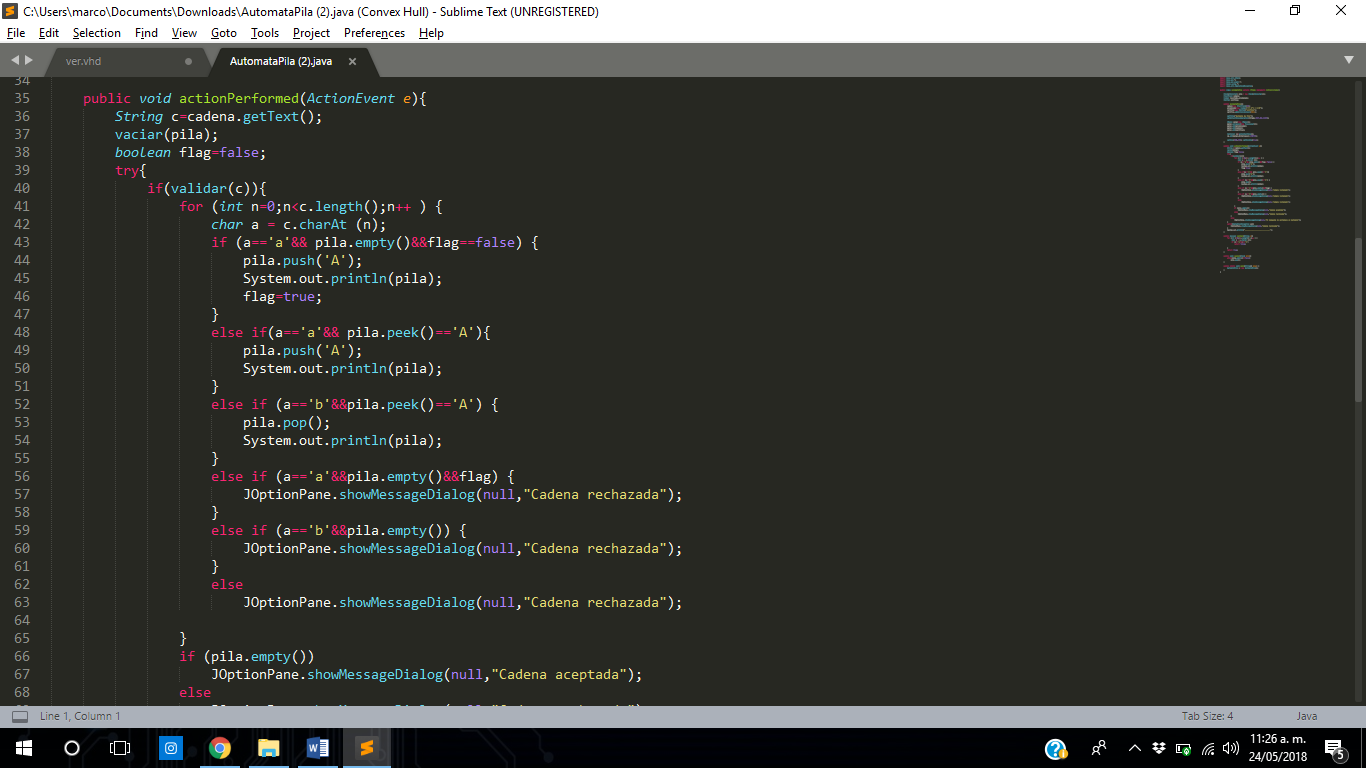
Una vez planteado el problema lo llevamos al lenguaje de programación Java en donde es más fácil crear pilas, en este caso es una pila muy sencilla y sin necesidad de hacer un gran esfuerzo o muchas validaciones, es por eso por lo que se decidió tomar este problema en Java, creando una clase “AutomataPila” que extenderá a un JFrame el cual es nuestra interfaz gráfica en donde podremos ingresar la cadena dentro de un *TextField* creando nuestra validación mediante un botón con el método *ActionListener* el cual al ser presionado entra a diferentes ciclos para lograr la validación tomando en cuenta las condiciones y el lenguaje principalmente, dando así como resultado en un JLabel el mensaje de si la cadena fue aceptada o no, por otra parte, no solo tenemos la validación de la cadena en el entorno gráfico sino que mostramos el contenido de la pila en consola, mostrando inserciones o bien si la cadena queda vacía o tiene algún contenido.

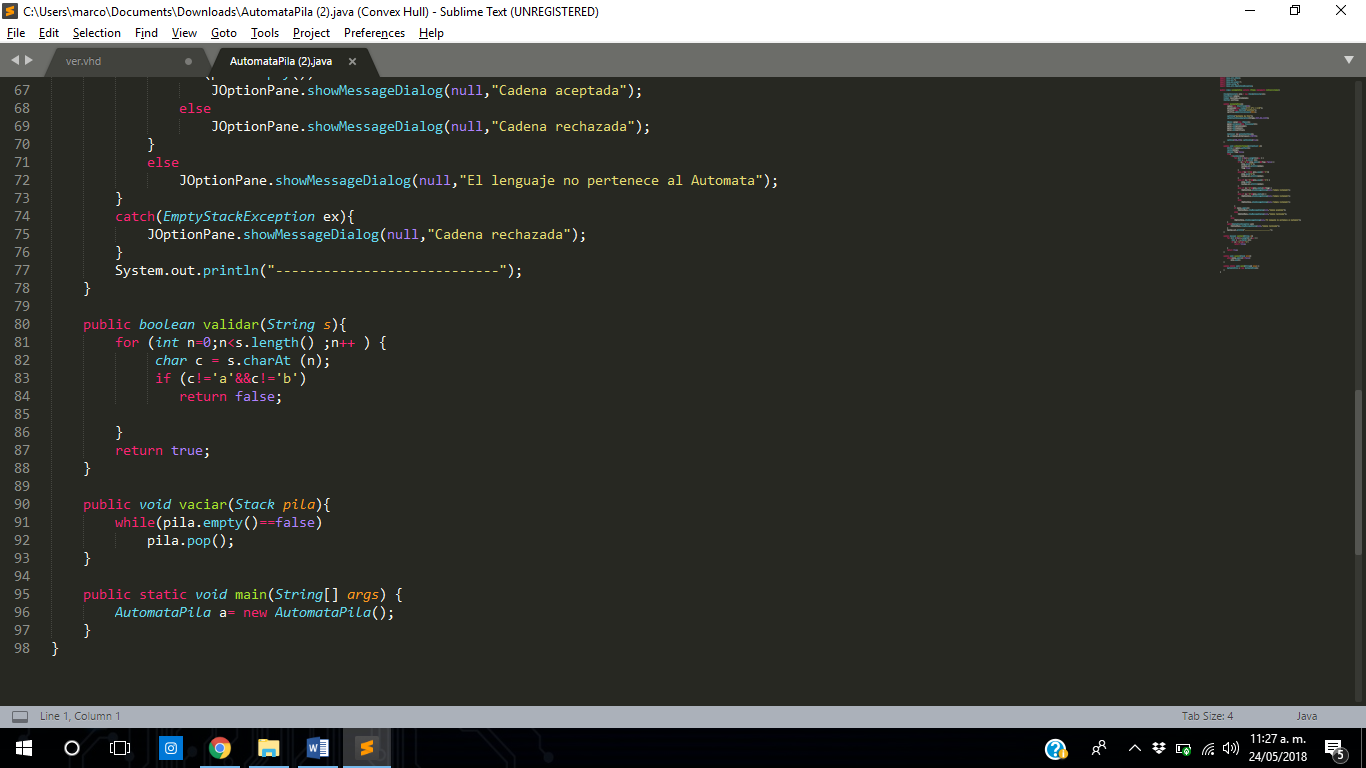
Siendo este nuestro autómata diseñado en JFlap:



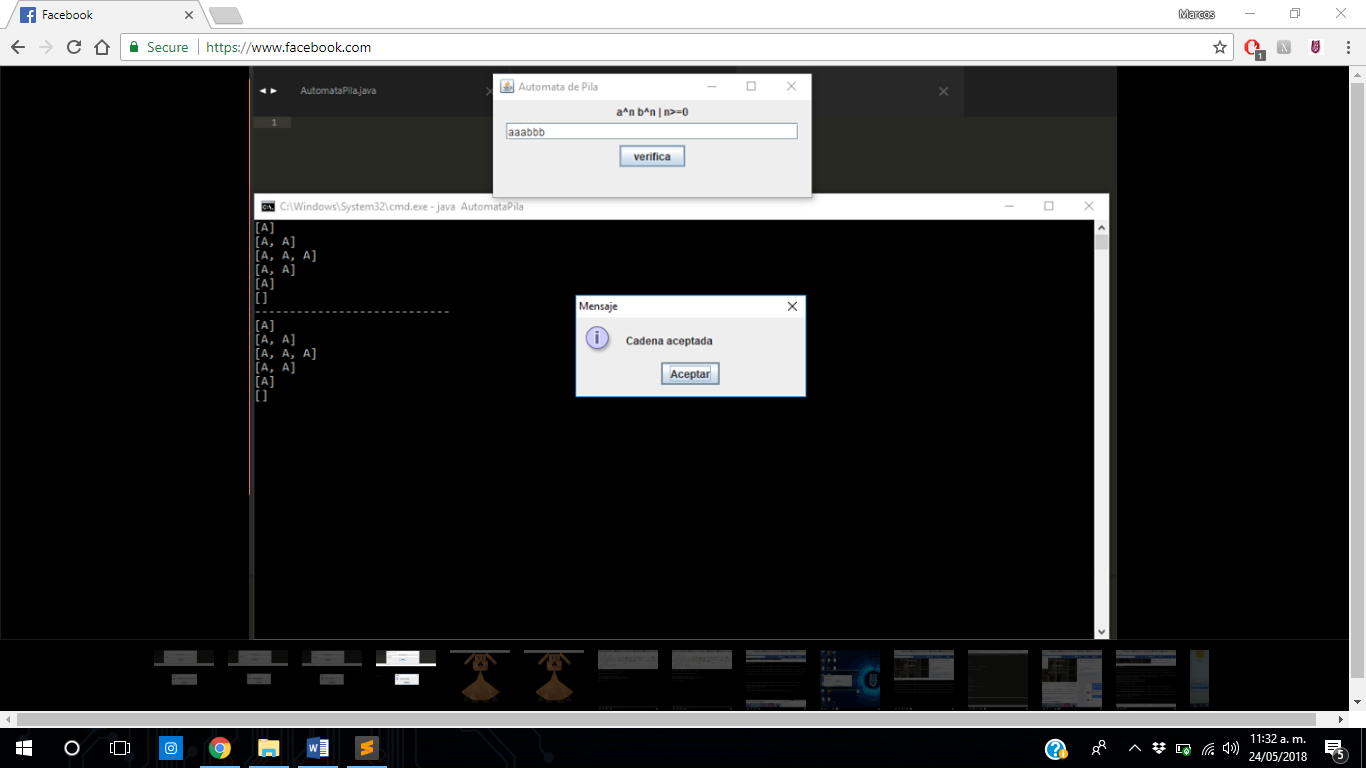
**IMPLEMENTACIÓN DE LA SOLUCIÓN**

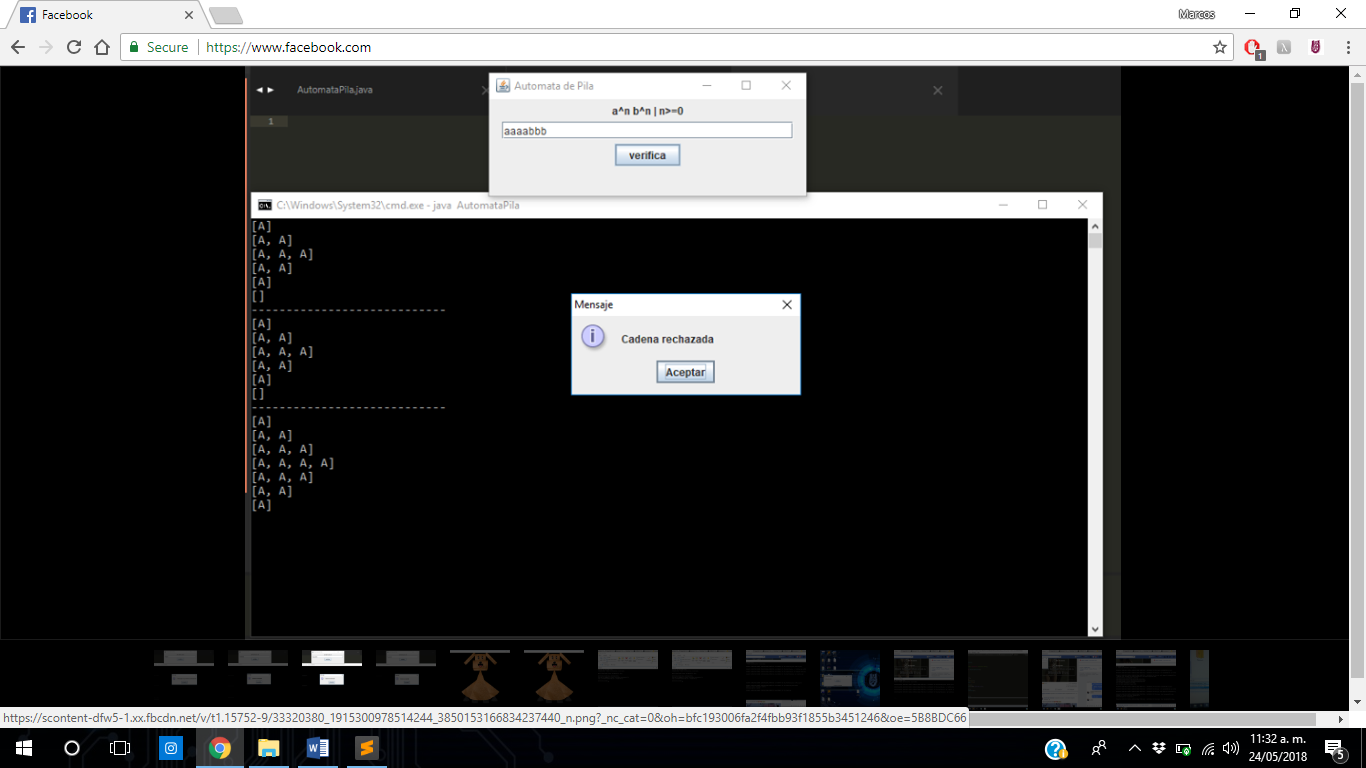


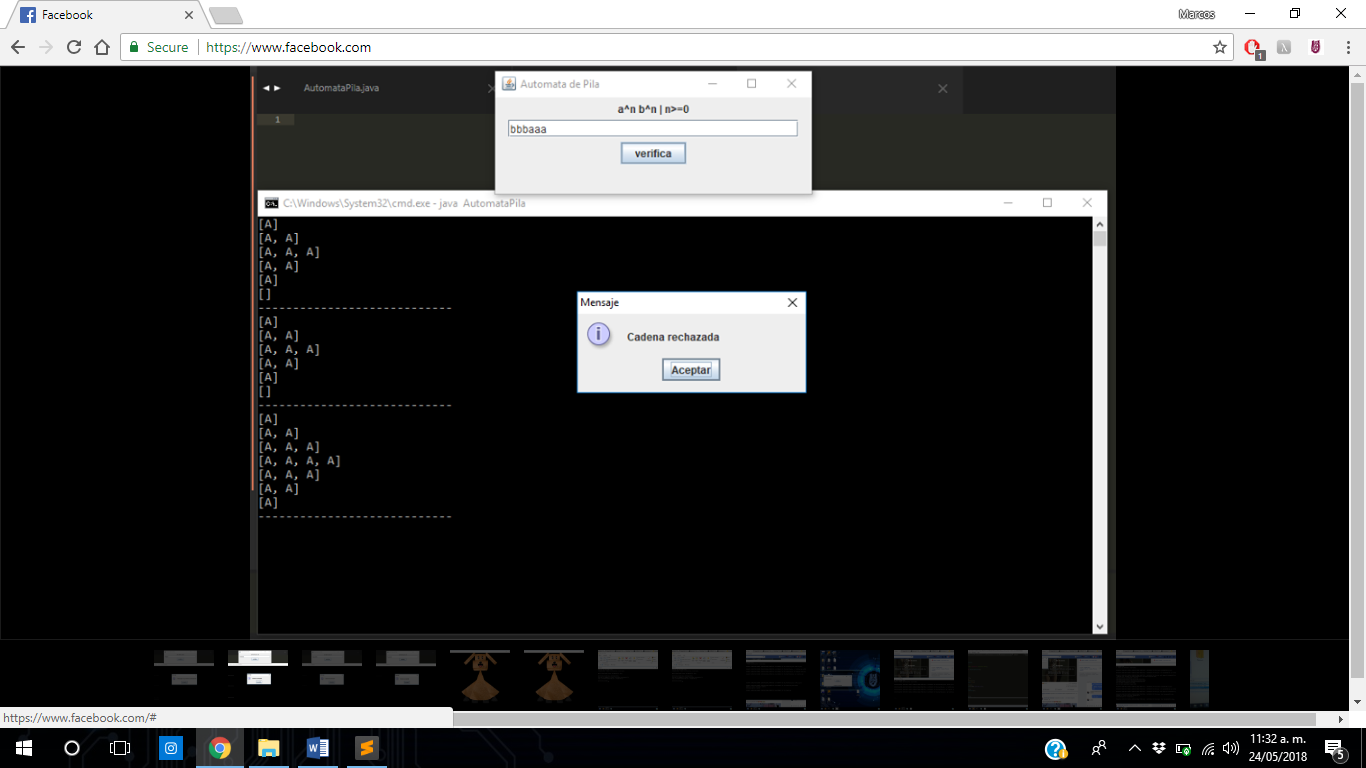


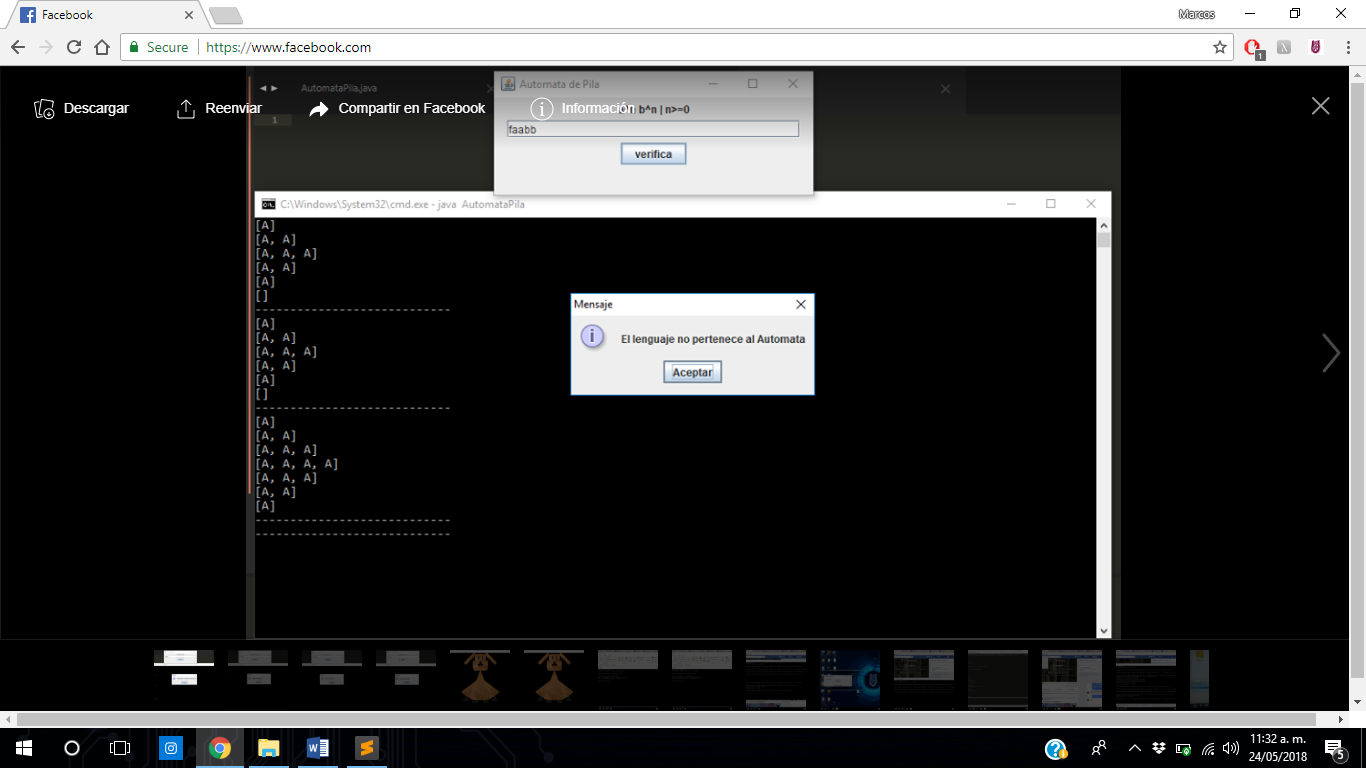


**FUNCIONAMIENTO**









**CONCLUSIONES**

En conclusión, podemos decir que fue una práctica sencilla, con un grado de dificultad de acuerdo a lo que el modelo matemático requiere, sin duda debemos mencionar que el hecho de realizar la práctica en lenguaje Java facilito todo ya que si hubiera sido en lenguaje C hubiéramos tenido que crear la pila por nuestra propia mano, de esta manera concluyo que es de gran utilidad controlar no solo un lenguaje de programación sino dominar varios para este tipo de casos en donde nos damos cuenta que lo más viable es salir de lo cotidiano.

Por otro lado, fue un gran punto de motivación para nosotros que lo aprendido en programación orientada a objetos nos sirviera para llevar una práctica a un entorno gráfico, se vea mejor y sea más entendible, de la misma manera tener más organización en cuanto a información.