



Universidad de San Carlos de Guatemala
Lenguajes Formales de Programación “A+”

Proyecto 2

Hector Josue Ponsoy Ayala
20180720

Clases:

- **AutomataP:** Clase que se encarga de la creación del autómata de pila.
- **EstadoP:** Clase que posee los estados de pila del autómata de pila.
- **TransicionP:** Clase en donde se crean las transiciones de pila del autómata de pila.
- **Gramatica2:** Clase que se encarga de la creación de la gramática del archivo de entrada.
- **Produccion2:** Clase dedicada a la creación de las producciones de los autómatas de pila.

Métodos:

- **Clase AutomataP**
 - **__init__:** Método constructor de la clase.
 - **crearEstados:** Método que se encarga de enviar los estados a la clase EstadoP.
 - **crearTransiciones:** Método que se encarga de crear las transiciones a través de la clase TransicionP.
 - **getPassword:** Método que devuelve la contraseña de un usuario.
 - **getSignIn:** Método para que sirve para iniciar sesión.
- **Clase EstadoP**
 - **__init__:** Método constructor de la clase.
- **Clase TransicionP**
 - **__init__:** Método constructor de la clase.
- **Clase AutomataP**
 - **__init__:** Método constructor de la clase.
 - **crearTerminal:** Método que se encarga de crear el terminal.
 - **crearNoTerminal:** Método que se encarga de introducir el no terminal inicial.
 - **crearProduccion:** Método que se encarga de crear las producciones de las gramáticas.
- **Clase Produccion2**
 - **__init__:** Método constructor de la clase.

Lógica de programación

Paradigma funcional y de objetos:

Los lenguajes puramente funcionales operan solamente a través de funciones. Una función devuelve un solo valor, dada una lista de parámetros. No se permiten asignaciones globales, llamadas efectos colaterales. Un programa es una llamada de función con parámetros que posiblemente llaman a otras funciones para producir valores de parámetro real.

Las funciones mismas son valores de primera clase que pueden ser pasados a otras funciones, y devueltos como valores funcionales. Así, la programación funcional proporciona la capacidad para que un programa se modifique a sí mismo.

En este caso se decidió usar este paradigma ya que los conocimientos de Python eran muy escasos por lo que se tornaría un poco complicado con respecto al tiempo de tratar de aprender su estructura más detallada para poder tomar otro paradigma, con lo cual con los conocimientos ya tenidos este era el paradigma que se adecuaba mejor.

Un objeto es un ente computacional que exhibe comportamiento (podemos comunicarnos con él mediante mensajes) y puede (o no) tener un estado interno (referencias a otros objetos).

Un programa basado en este paradigma es un conjunto de objetos que interactúan entre sí en un ambiente mandándose mensajes para lograr un objetivo determinado.



