Proyecto Lenguajes de Programación

*Compilador de Listas en Java*

AVANCE FINAL

*GRUPO 6*

*Integrantes:*

*\* Isabela Vinces*

*\* Fanny Valverde*

*\* Pedro Mendoza*

**Introducción.**

**Java.**

Java es un lenguaje orientado a objetos, eso implica que su concepción es muy próxima a la forma de pensar humana.

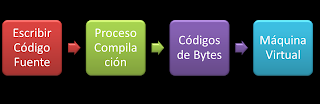
* Es un lenguaje que es compilado, generando ficheros de clases compilados, que en realidad interpretadas por la máquina virtual java. Siendo la máquina virtual de java la que mantiene el control sobre las clases que se estén ejecutando.
* Es un lenguaje multiplataforma: El mismo código java que funciona en un sistema operativo funcionará en cualquier otro sistema operativo que tenga instalada la máquina virtual java.
* Es un lenguaje seguro: La máquina virtual realiza comprobaciones de seguridad, además el propio lenguaje carece de características inseguras, como por ejemplo los punteros.

Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, o "write once, run anywhere"), lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Java es, a partir de 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de web, con unos diez millones de usuarios reportados.

**Compiladores en Java**

La función de un compilador es convertir un programa escrito un en lenguaje como Java a un lenguaje que la computadora pueda entender. Los programas en Java se convierten en códigos de bytes.

Al compilar el programa, el compilador verifica que éste obedezca las reglas de programación en Java, y si algo sale mal, muestra los mensajes de error apropiados. El compilador crea un archivo con el sufijo .class. Este contiene el código de bytes equivalente a su programa en Java.



Pero el archivo .class que produce el compilador no está listo todavía para ejecutarse ya que requiere vincularse con las bibliotecas de clases Java proceso que se realizará cuando el programa se ejecute.

**Objetivos**

* Desarrollar un compilador en Python con la librería PLY.
* Lograr que el compilador reconozca las estructuras de listas de Java funcional.
* Utilizar expresiones regulares para el reconocimiento de tokens que permitan el desarrollo adecuado del compilador.
* Crear una interfaz gráfica interactiva para que el usuario pueda comprobar la apropiada declaración de las estructuras.

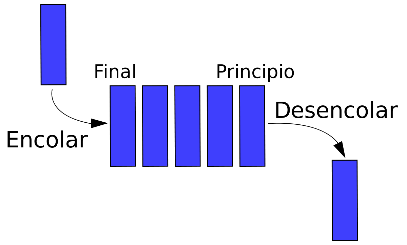
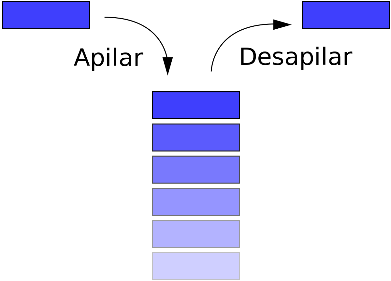
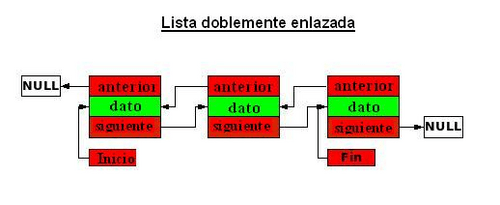
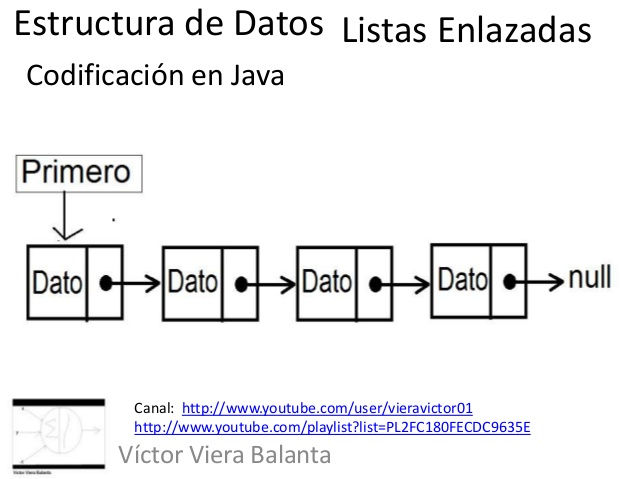
**Alcance**

Se manejará una aplicación que tendrá una interfaz gráfica donde el usuario podrá ingresar las declaraciones de las siguientes estructuras de datos:

* ArrayList.
* LinkedList.
* DoubleLinkedList.
* Stack.
* Queue.

El usuario además deberá definir, mediante la declaración, el tipo de dato que contendrá estas listas, es decir si va a trabajar con Integers, Floats, Strings, etc. El compilador reconocerá los tipos de datos más básicos y verá si el tipo que escriba el usuario existen y es correcto para la declaración de la estructura.

**Metodología.**



Para realizar la declaración de las palabras reservadas que serán las claves en la verificación de las estructuras, utilizaremos tokens definidos como PLY lo requiera. Para los tokens se usan expresiones regulares. Estas expresiones regulares serán escritas de tal manera que siga un patrón adecuado según como tendría que seguir una declaración de las 5 estructuras previamente mencionadas.

Luego sigue la parte del yacc, en la cual se analizará y aplicarán las reglas gramaticales del lenguaje, que en este caso es Java. (etapa de parsing).

Una vez haya sido todo analizado correctamente por las herramientas, pasaremos a la creación de una interfaz básica donde el usuario escribirá su declaración de la estructura, y se dará a conocer si la estructura fue correctamente creada o si tiene errores en la sintaxis.

Por ejemplo, si quiere crear un arraylist, deberá escribir: “ArrayList<TipoDato> nombreVariable= new ArrayList<TipoDato>();” Si no es así, presentará un mensaje de error y dará a conocer cómo se declara la estructura.

**Tema**

El compilador se basa solo en la declaración de 5 listas básicas de Java. Estas listas pueden tener distintos modificadores de acceso, declaraciones internas, palabras reservadas necesarias para declarar bien las listas, entre otras cosas.

Estas listas son estructuras de datos muy conocidas, importantes y manejadas por muchos programadores para el correcto manejo de la información.

En Java, estas listas pueden contener diversos tipos de datos, desde primitivos hasta Objetos declarados por nosotros mismos. Sin embargo, por el alcance del proyecto, solo se manejarán los tipos de datos más básicos.

La declaración de las listas va de la forma: Tipo de lista con su tipo de dato, el acceso y el nombre de la variable del lado izquierdo del “igual” (=), mientras que en la derecha irá “new TipoDeLista” sin el tipo de dato. Esto por convención a la declaración en Java donde es más recomendable no escribir el tipo de dato dentro de los “<>”.

**Ejemplos:**

* **public ArrayList<String> lista= new ArrayList<>();**
* **private LinkedList<int> n= new LinkedList< >();**
* **Queue<int> cola=new LinkedList<>();**
* **Stack<float> pila = new Stack<>();**
* **ArrayList<float> array= new ArrayList<>();**
* **DoubleLinkedList<String> doble = new DoubleLinkedList<>();**
* **public final LinkedList<String> lst= new LinkedList<>();**
* **Queue<Object> cola2=new LinkedList<>();**
* **Stack<int> struct = new Stack<int>();**
* **private DoubleLinkedList<float> lst= new DoubleLinkedList<>();**

**Herramientas.**

* **Librería PLY (Lex-Yacc) en Python.**

PLY (Python Lex & yacc) es una herramienta de analizador léxico implementada para python. PLY consta de dos módulos separados; lex.py y yacc.py , ambos de los cuales se encuentran en un paquete de Python llamada capa. El módulo lex.py se utiliza para romper texto de entrada en una colección de fichas especificadas por una colección de reglas de expresiones regulares. yacc.py se utiliza para reconocer la sintaxis del lenguaje que se ha especificado en el formulario de una gramática libre de contexto. Las dos herramientas tienen el propósito de trabajar juntos.

* **Github**

Github es una plataforma creada para facilitar el desarrollo colaborativo de software, que permite alojar proyectos en la web gratuitamente, por lo general de forma pública, aunque si se desea alojar los proyectos de modo privado, se debe pagar una suscripción.

Github ofrece al desarrollador toda la potencia y agilidad del sistema de control de versiones Git, más un interesante set de herramientas añadidas:

* + Wiki
  + Sistema de seguimiento de incidencias
  + Interfaz gráfica para revisión/comparación de código
  + Visor de ramas de desarrollo
* **Tkinter**

Tkinter es un binding de la biblioteca Tcl/Tk que está también disponible para otros lenguajes como Perl y Rubí. Tkinter es fácil de usar, es multiplataforma y, además, viene incluido con Python en su versión para Windows, para Mac y para la mayoría de las distribuciones GNU/Linux. Se considera un estándar para la interfaz gráfica de usuario (GUI) para Python.

# Referencias

Miró, A. (17 de 08 de 2017). *Deusto Formación*. Obtenido de https://www.deustoformacion.com/blog/programacion-diseno-web/que-es-para-que-sirve-github

*Mundo Java*. (s.f.). Obtenido de http://www.mundojava.net/caracteristicas-del-lenguaje.html?Pg=java\_inicial\_4\_1.html

Oramas, R. (s.f.). *Profesor Java*. Obtenido de http://profejavaoramas.blogspot.com/2010/04/compilador-java.html

Pherkad. (12 de 2015). *Python Para Impacientes*. Obtenido de https://python-para-impacientes.blogspot.com/2015/12/tkinter-interfaces-graficas-en-python-i.html