

07/11/2013

Proyecto Programado 2

Tabla Dinámica y Estática de SML

Yaxiri Azofeifa García
Carlos Campos Fuentes
Fabricio Soto Mejías
Michelle García Campos
Daniela Valerio Cerdas

El objetivo consiste en aumentar el conocimiento en cuanto al funcionamiento de la tabla de tipado de SML en cuanto al ambiente dinámico y estático.
Y poder realizar el intercambio de mensajes por medio de sockets.



Tabla de contenido

Tabla de contenido	2
Métodos Estudiados:	3
Diseño del programa:	3
Decisión de diseño	3
Algoritmos usados (Funciones principales del programa):	4
Librerías de Java utilizadas.....	6
Análisis de resultados	6
Objetivos alcanzados	6
•Se logró con éxito la identificación de los valores de almacenamiento en ambiente dinámico y es- tático con respecto a tuplas, listas y operaciones.	6
Manual de usuario	6
Conclusión Grupal	6

Descripción del programa:

Al querer implementar un sistema de información al nivel computacional, es de vital importancia el manejo de los datos, métodos y estructuras para cubrir con las necesidades que requiere la implementación.

El desarrollo del proyecto se establecerá bajo las reglas del lenguaje funcional SML, sin embargo, por los objetivos de desarrollo de la tarea se tomó la decisión de emplear el lenguaje de programación Java.

Las especificaciones de la tarea nos llevarán a desarrollar una serie de métodos a partir de los cuales se llevará a cabo el análisis para poder identificar el ambiente correspondiente al tipado, ya sea el dinámico o estático; para así obtener las tablas necesarias para representar la información.

En cuanto a las tablas, estas tendrán el objetivo de representar datos específicamente:

Ambiente Dinámico: Representa el valor almacenado en una variable.

Ambiente Estático: Representa el tipo (String, int, tupla, 'a list...') del dato almacenado en la variable.

Métodos Estudiados:

Durante el desarrollo de este proyecto no fue necesario el estudio de métodos en específico, ya que las diferentes necesidades que se presentaron durante el desarrollo, fueron saciadas mediante la creación de métodos y implementación de librerías.

Diseño del programa:

A continuación se especifica el diseño usado en el programa, además de la manera en que se maneja las diferentes funciones implementadas.

Decisión de diseño

- Se desarrolló en el lenguaje de Programación Java, debido a que era un lenguaje con el que se encontraba familiarizado el equipo de trabajo.
- El diseño del programa fue bajo la base de la modularidad, ya que fue mediante el uso de múltiples métodos (Detallados en el resto del documento), que se logró la identificación correcta de las variables, lets, en sus estados dinámicos y estáticos.
- En cuanto almacenamiento de los datos de las tablas, se seleccionó el uso de una lista doble, con nodos, que contaban con 3 espacios, de manera que se almacena en el primer espacio el identificador (nombre de la variable), en otro el valor, y en otro el tipo.

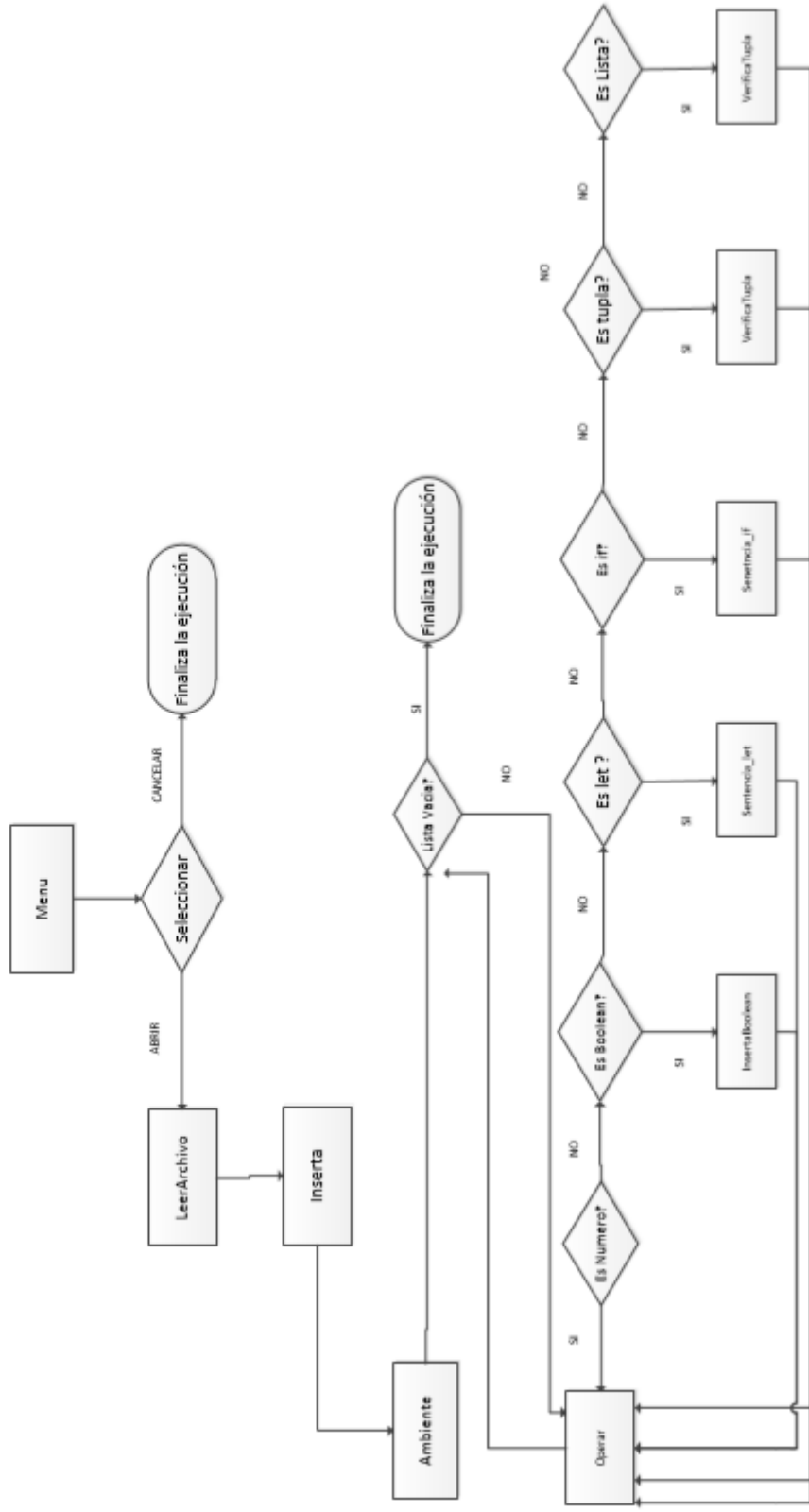
- El uso de la recursividad fue imprescindible para la creación de métodos para la identificación del tipo y el valor de expresiones en cuanto a tuplas, listas, if, y lets.

Algoritmos usados (Funciones principales del programa):

Nombre de Función	Funcionamiento	Valor de retorno
Inserta()	Posee la funcionalidad de verificar el valor de la tabla de ambiente dinámico y estático a partir de una instrucción, misma que se encuentra almacenada en una lista, específicamente tuplas, listas y operaciones.	No existe valor de retorno, El resultado son datos insertados en la lista que almacena el valor de las variables en ambos ambientes.
Sentencia_if()	Verifica si una serie de instrucciones posee un “if”, y sigue las instrucciones necesarias para almacenarlas en la tabla de estado dinámico y estático.	No posee un valor de retorno, lo que realiza, es la inserción correspondiente a la información en el ambiente, dinámico y estático, de acuerdo a la instrucción que recibe.
Sentencia_let()	Verifica si una serie de instrucciones posee un “if”, y sigue las instrucciones necesarias para almacenarlas en la tabla de estado dinámico y estático.	No posee un valor de retorno, lo que realiza, es la inserción correspondiente a la información en el ambiente, dinámico y estático, de acuerdo a un let.
VerificaTupla()	Le corresponde la verificación del tipo de las tuplas.	El valor de retorno que posee es un String, correspondiente al tipo de dato que se encuentra almacena en la tupla verificada.
VerificaLista()	Le corresponde la verificación del tipo de las listas.	El valor de retorno que posee es un String, correspondiente al tipo de dato que se encuentra almacena en la lista verificada.
RevisaTabla ()	Posee la función de buscar un elemento en la lista que tiene almacenado el valor de una variable correspondiente al ambiente dinámico.	El valor de retorno corresponde a un String con el valor de la variable que fue consultada en el ambiente dinámico.
LeerArchivo ()	Tiene la funcionalidad de leer de abrir un archivo .txt, y leer	El valor de retorno consiste en una lista, mediante la cual

	los datos que contiene almacenado.	se pondrán en funcionamiento los métodos necesarios para completar las tablas de ambiente dinámico y estático.
Operar ()	Le corresponde realizar a cabo las operaciones a partir de una cadena de Strings.	El valor de retorno consiste en una cadena de strings, como resultado de la operación de entrada,

Diagrama Lógico



Librerías de Java utilizadas

Swin y AWP: Consisten en la biblioteca de java que permite el desarrollo de interfaz gráfica para aplicaciones, a partir de estas se desarrollaron las tablas para presentar los resultados obtenidos de la lectura de instrucciones del documento.txt.

Script: Consiste en un método que permite llevar a cabo evaluaciones a partir de una cadena de strings mediante el motor de operaciones llamado ScriptEngine.

Análisis de resultados

Objetivos alcanzados

- Se logró con éxito la identificación de los valores de almacenamiento en ambiente dinámico y estático con respecto a tuplas, listas y operaciones.
- Se logró el análisis de ambientes correspondientes a expresiones que involucran, if, then, else.
- Se logró con éxito representar en las tablas de ambiente dinámico y estático los valores y tipos de las expresiones let.
- Se logró obtener los resultados correspondientes a la combinación de expresiones mencionadas anteriormente.
- Se logró el desarrollo de la representación de las tablas de los ambientes dinámicos y estáticos.

Manual de usuario

Se encuentra adjunto con el nombre Manual de Usuario.pdf.

Conclusión Grupal

Con este trabajo se desarrollaron conocimientos sobre el manejo de datos en el ambiente estático y dinámico que utiliza el lenguaje SML. Entre las funcionalidades del programa que presentaron un problema de dificultad un poco mayor, fueron la implementación de tuplas, debido a que las tuplas pueden contener cualquier tipo de dato almacenado; se debe mencionar también la implementación del if, por el comportamiento de las instrucciones, and, or y let, este último presentó un nivel de complejidad mayor. El motivo es que el let solo funciona dentro de un alcance específico. En general con la implementación del programa adquirimos mucho valor agregado en cuanto a SML, específicamente, el manejo de ambientes de este lenguaje. En conclusión, se alcanzaron con éxito todas las funcionalidades requeridas en la especificación del proyecto.