

**INSTITUTO TECNOLOGICO DE CHIHUAHUA II**

Ing. Celaya Chavira Lorena

Lenguajes y Autómatas

PROYECTO “FRUTILENGUAJE”

Ingeniería en Sistemas Computacionales

Integrantes:

Meraz Zapata Daisy Guadalupe 12550509

Aragonez Olivas Manuel Eduardo 12550434

García Medina Carmen Jazmín 12550465

Aguayo Barrón José Fernando



JUSTIFICACIÓN:

Desarrollaremos y pondremos en práctica los conocimientos obtenidos durante el curso creando un nuevo lenguaje de programación basado en el tema “Frutilenguaje “ fue elegido así para que sea más fácil su aprendizaje de una manera didácticas, todo fue creado de acuerdo al lenguaje ya conocido “JAVA”

DISEÑO:

Palabras reservadas:

|  |  |
| --- | --- |
| Lenguaje Java | “Frutilenguaje” |
| boolean | **zarzamora** |
| True | **manzana** |
| false | **durazno** |
| Class | **sandia** |
| Else | **melon** |
| float | **toronja** |
| For | **naranja** |
| If | **Uva** |
| int | **Frambuesa** |
| null | **Lima** |
| private | **Granada** |
| protected | **Higo** |
| public | **Guayaba** |
| static | **fresa** |
| void | **Ciruela** |
| while | **Tamarindo** |
| String | **Datil** |

**IDENIFICADORES:**

|  |  |  |
| --- | --- | --- |
| Tipo | JAVA | Frutilenguaje |
| Letras | **a-z**  **A-Z** | **a-z**  **A-Z** |
| Números | **0-9** | **0-9** |
| Operadores aritméticos | **+**  **-**  **\***  **/**  **++**  **--** | **+**  **-**  **\***  **/**  **+1**  **-1** |
| Operadores comparativos | **<**  **>**  **<=**  **>=**  **!=** | **(**  **)**  **(=**  **)=**  **?** |
| Operadores delimitadores | **(**  **)**  **,**  **{**  **}**  **;**  **“ ”**  **.** | **#-**  **-#**  **.**  **¡**  **!**  **&**  **‘’**  **.** |
| Operador de asignación | **=** | **=** |
| Operador de igualación | **==** | **==** |

Autómata Finito Determinista (AFD)

**L= Letras**

**N = Números**

**O = Otro**

**ε**

**ε - D**

**L, D**

**\***

**O**

**L**

**ε - D**

**D**

**D**

**ε-D**

**\***

**D**

**.**

**D**

**ε**

**ε**

**+**

**-**

**#**

**q0**

**ε- #**

**¡**

**.**

**\**

**\***

**q0**

**ε**

**¡**

**&**

**ε**

**q0**

**?**

**)**

**(**

**=**

**ε**

**=**

**q0**

**ε - (= I ( )**

**°I**

**q0**

**=**

**(**

**ε**

**ε - =**

**-**

**q0**

**#**

**ε- (-)**

**EJEMPLO**

El siguiente programa consiste en preguntar edades, buscar las edades mayores y sacar suma y promedio de las mismas, así al igual lo hace con las edades menores, busca una edad deseada y te menciona cuantas personas son mayores de edad y cuantas menores.

PROGRAMA EN “FRUTILENGIAJE”

guayaba sandia Edades¡

guayaba guanabana ciruela main -#Datil‘ ’args#- ¡

frambuesa nElementos = Frambuesaeger.parseFrambuesa-#JOptionPane.showInputDialog-#-Número de elementos del arreglo: -#-#-,

edadesmayor = 0, edadesmenor = 0, promedadmayor = 0, promedadmenor = 0, sumaedadmayor = 0, sumaedadmenor = 0, elementbus = 0, encontrado = 0&

frambuesa’’ edades = new frambuesa’nElementos’&

frambuesa i = 0&

JOptionPane.showMessageDialog-#null, -Frambuesaroduce las edades:-#-&

naranja -#i = 0& i ( nElementos& i +1 #- ¡

edades’i’ = Frambuesaeger.parseFrambuesa-#-#JOptionPane.showInputDialog-#-Edad’- + -#i + 1#- + - = -#-#-#-&

uva -#edades’i’ )= 18#- ¡

edadesmayor+1&

sumaedadmayor += edades’i’&

promedadmayor = sumaedadmayor / edadesmayor&

! else ¡

uva -#edades’i’ (= 18#- ¡

edadesmenor+1&

}

sumaedadmenor += edades’i’&

promedadmenor = sumaedadmenor / edadesmenor&

!

!

naranja -#i = 0& i ( nElementos& i+1#- ¡

JOptionPane.showMessageDialog-#null, -Edad’- + -#i + 1#- + -’ = - + edades’i’#-&

!

elementbus = Frambuesaeger.parseFrambuesa-#JOptionPane.showInputDialog-#-cual es la edad que deseas buscar: -#-#-&

naranja -#i = 0& i ( nElementos& i+1#- ¡

uva -#elementbus == edades’i’#- ¡

encontrado+1&

!

!

JOptionPane.showMessageDialog-#null, -las veces que se encontro la edad - + elementbus + - = - + encontrado + - veces-#-&

JOptionPane.showMessageDialog-#null, -el numero de personas mayores de edad son: - + edadesmayor

+ -\n el total de personas mayores de edad es: - + edadesmenor

+ -\n el promedio de las personas mayores de edad es: - + promedadmayor

+ -\n el promedio de las personas menores de edad es: - + promedadmenor#-& !

!

PROGRAMA ORIGINAL EN “JAVA”

**public class Edades{**

**public static void main(String[] args) {**

int nElementos = Integer.parseInt(JOptionPane.showInputDialog("Número de elementos del arreglo: ")),

edadesmayor = 0, edadesmenor = 0, promedadmayor = 0, promedadmenor = 0, sumaedadmayor = 0, sumaedadmenor = 0, elementbus = 0, encontrado = 0;

int[] edades = new int[nElementos];

int i = 0;

JOptionPane.showMessageDialog(null, "Introduce las edades:");

for (i = 0; i < nElementos; i++) {

edades[i] = Integer.parseInt((JOptionPane.showInputDialog("Edad[" + (i + 1) + " = ")));

if (edades[i] >= 18) {

edadesmayor++;

sumaedadmayor += edades[i];

promedadmayor = sumaedadmayor / edadesmayor;

} else {

if (edades[i] <= 18) {

edadesmenor++;

}

sumaedadmenor += edades[i];

promedadmenor = sumaedadmenor / edadesmenor;

}

}

for (i = 0; i < nElementos; i++) {

JOptionPane.showMessageDialog(null, "Edad[" + (i + 1) + "] = " + edades[i]);

}

elementbus = Integer.parseInt(JOptionPane.showInputDialog("cual es la edad que deseas buscar: "));

for (i = 0; i < nElementos; i++) {

if (elementbus == edades[i]) {

encontrado++;

}

}

JOptionPane.showMessageDialog(null, "las veces que se encontro la edad " + elementbus + " = " + encontrado + " veces");

JOptionPane.showMessageDialog(null, "el numero de personas mayores de edad son: " + edadesmayor

+ "\n el total de personas mayores de edad es: " + edadesmenor

+ "\n el promedio de las personas mayores de edad es: " + promedadmayor

+ "\n el promedio de las personas menores de dad es: " + promedadmenor);

**}**

**}**

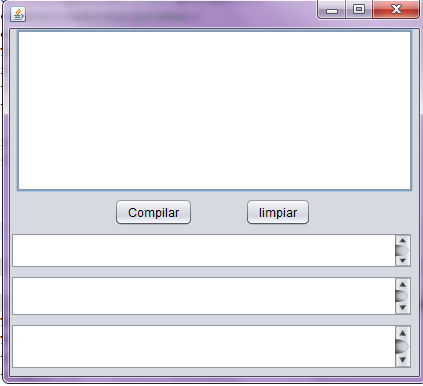
**Reglas semánticas**

* **E** → E+B {si (E.va l= ; si = Frambuesa, E.tipo = Frambuesa) si no (E.tipo = Toronja}
* **E** → E-B { si (E.va l= ; si = Frambuesa, E.tipo = Frambuesa) si no (E.tipo = Toronja }
* **E** → B {E.tipo = B.tipo; E.val = B.val}
* **B** → B\*F { si (B.val = ; si = Frambuesa, B.tipo = Frambuesa) si no (B.tipo = Toronja }
* **B**→ B/F {B.tipo = Toronja}
* **B** → F { B.tipo = F.tipo; B.val = F.val }
* **F** → #-E-# {F.tipo = E.tipo; F.val = E.val }
* **F** → id {F.tipo = buscar(id.ent); F.val = buscar(id.ent)}
* **F** → num {F.tipo = buscartipo(num.lex); F.val = num.val}

**Eliminación de recursividad**

* **E** → B {E´.h = B.tipo; E´. = B.val} E´ {E.tipo =E’.S; E.val = E´.}
* **E** → +B {. = E’.+ B.val; si (.h y B.tipo = Frambuesa, E’.h = Frambuesa) sino (E’.h = Toronja)} {E’.S=.S; E’.=. }
* **E** → -B {. = E’.- B.val; si (.h y B.tipo = Frambuesa, E’.h = Frambuesa) sino (E’.h = Toronja)} {E’.S=.S; E’.=. }
* **E** → ε {E’.S = E’.h; .= .}
* **B** → F {B´.h = F.tipo; B´. = F.val} B´ {B.tipo =B’.S; B.val = B´.}
* **B** → \*F { . = B’.\* F.val; si (.h y F.tipo = Frambuesa, B’.h = Frambuesa) sino (B’.h = Toronja)} {B’.S=.S; B’.=. }
* **B**→ /F {. = B’./ F.val; .h = Toronja; }
* **B** → ε {B’.S = B’.h; .= .}
* **F** → #-E-# {F.tipo = E.tipo; F.val = E.val }
* **F** → id {F.tipo = buscar(id.ent); F.val = buscar(id.ent)}
* **F** → num {F.tipo = buscartipo(num.lex); F.val = num.val}

Manual

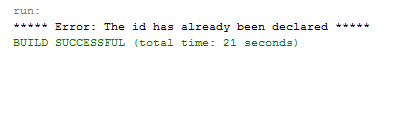


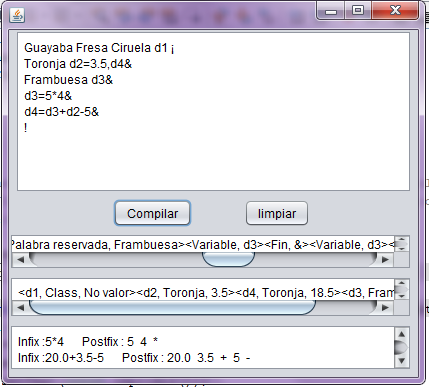
Pantalla Principal

Para empezar a escribir un programa debemos siempre colocar el siguiente encabezado Guayaba Fresa Ciruela [variable] ¡ , sabiendo que nuestras variable se inicializan de la forma las siguientes vocales c, d y e seguidas de un número, que pueden seguir de alguna letra del abecedario. Después que nosotros colocamos nuestro encabezado sigue la parte de Declarar nuestras variables ahí viendo la tabla de palabras reservadas podemos conocer el tipo de datos que podemos manejar en nuestro lenguaje. Finalmente Tenemos nuestro cuerpo es donde podemos inicializar variables hacer operaciones, agregar condiciones lógicas.

Cabe mencionar que todas las palabras reservadas llevan su primera letra Mayúscula.

Si se llegase a equivocar al escribir nos arrojaría un error léxico si nos llegamos a equivocar en la forma que se escribe el programa nos arrojaría un error sintáctico y por ultimo si lo que escribimos no tiene sentido sería un error semántico. Que en la parte de abajo del programa. Como se puede apreciar en la siguiente imagen:





Programa corriendo