

**Diseño de Compiladores**

**Profesora Elda Gudalupe Quiroga**

**NOMBRE DEL PROYECTO**

**Integrantes:**

**Alberto Bermea A01196516**

**Esthephany Ayala A00818207**

**PROPUESTA INICIAL**

**28 de Septiembre del 2020**

**2.Your language main objective, including category (area).**

El lenguaje a desarrollar será un lenguaje funcional que contiene

La estructura general de un programa será de la siguiente manera:

( define ( nombreFuncion1 **param )**

<BLOQUE 1>

<BLOQUE 2>

<BLOQUE n> // los bloques serán utilizados para definir el comportamiento de dicha función

)

( define ( nombreFuncion2 **param )**

<BLOQUE 1>

<BLOQUE 2>

<BLOQUE n> // los bloques serán utilizados para definir el comportamiento de dicha función

(funcion2 param)

)

**3.Language requirements (depending on specific purpose):**

Para la declaración de variables:

* Se tendrá una propia función para la declaración de variables.
* Por cada variable que se desee declarar, se realizara una nueva función.
* Deberá utilizarse la palabra <<*declare>>* para la declaración de variables.
* No será necesario declarar el tipo de dato.
* Las declaraciones de variables no podrán ir dentro de una función.

La estructura para la función será de la siguiente manera:

( declare tipoDeDato valor )

Para la declaración de funciones:

* Debera utilizarse la palabra <<define>> para crear una nueva función
* Si se desea referenciar otra función dentro de la nueva función creada, la función que se quiera utilizar deberá estar creada antes que la función que se esta creando en ese momento.

**3.1.Basic elements (Tokens) like keywords, id’s, etc.**

|  |  |
| --- | --- |
| Token | Nombre |
| ( | **OPEN\_PAREN** |
| ) | **CLOSE\_PAREN** |
| print | **PRINT** |
| declare | **DECLARE** |
| id | **ID** |
| + | **SUM** |
| - | **RESTA** |
| \* | **MULT** |
| / | **DIV** |
| > | **MAYOR** |
| < | **MENOR** |
| != | **DIF** |
| if | **IF** |
| define | **DEFINE** |
| int | **INT** |
| float | **FLOAT** |
| char | **CHAR** |
| lambda | **LAMBDA** |
| cte\_int | **CTE\_INT** |
| cte\_float | **CTE\_FLOAT** |
| cte\_char | **CTE\_CHAR** |

**3.2.Syntax Diagrams for all the structures in your language.**

* **GRAMATICA**

**<ESCRITURA>**

S -> ( print S' )

S' -> cte\_string | TERMINO

**<DEFINIR>** // Para declarar variables

S -> ( declare id S' )

S' -> cte\_int | cte\_float | cte\_string

**<VARCTE>**

S -> id

S -> cte\_int

S -> cte\_float

**<FACTOR>**

S -> EXPRESION

S -> ( + VARCTE )

S -> ( - VARCTE )

S -> VARCTE

**<TERMINO>** // Para multiplicaciones y divisiones

S -> ( \* FACTOR FACTOR )

S -> ( / FACTOR FACTOR )

S -> FACTOR

**<EXP>** //para sumas y restas

S -> ( + TERMINO TERMINO )

S -> ( - TERMINO TERMINO )

S -> TERMINO

**<EXPRESION>** // para comparativas o lo demas

S -> EXP

S -> ( S' EXP EXP )

S' -> > | < | != (este ultimo es diferente a racket)

**<CONDICION>**

S -> ( if EXPRESION BLOQUE BLOQUE )

**<BLOQUE>**

S -> CONDICION | ESCRITURA | EXPRESION | LLAMADA | LAMBDA

**<TIPO>**

S -> int

S -> float

S -> char

**<PARAM>**

S -> id S | epsilon

**<FUNCION>**

S -> ( define ( id PARAM ) BLOQUE )

**<LLAMADA>** //llamada de funcion

S -> ( id S' )

S' -> EXP S' | epsilon

**<LAMBDA>**

( ( lambda (PARAM) BLOQUE ) EXP )

* **DIAGRAMAS**

**Diagram

Description automatically generated**

**Diagram

Description automatically generated**

**Diagram

Description automatically generated**

**Diagram

Description automatically generated**

**Diagram

Description automatically generated**

**Diagram

Description automatically generated**

**3.3.Main Semantic characteristics.**

**3.4.Brief description of every special functions as well as rarely used instructions in your language (most of them are related tothe area of your language).**

Lambda

Declare : se utiliza apra la declaración de variabes

Define: se utiliza para la declaración de funciones

Fst (lo mismo que car pero con mejor nombre): regresa el primer elemento de una lista

Rest(lo mismo que cdr): regresa lo que sobra de la lsta

List: crea a lista con los argumentos que se proporciones

Lambda

Filter :

Map:

List?

Number?

Null?

Int?

Float?

**3.5.Data types.**

Integer

Float

Char

Lists

**4.Language and OS that will be used fordevelopment.**

Python

PLY

**5.Bibliography..**