



Proyecto final - Fundamentos de Lenguajes Programación

Robinson Duque, Ph.D
robinson.duque@correounivalle.edu.co

Julio 23 de 2017

1. Introducción

El presente proyecto tiene por objeto enfrentar a los estudiantes del curso:

- a la comprensión de todos los conceptos vistos en clase
- a la implementación de tres versiones de interpretadores de un lenguaje de programación
- al análisis de estructuras sintácticas, de datos y de control de un lenguaje de programación para la implementación de los interpretadores

2. (50pts) Lenguaje BASICO-I

Se propone para este proyecto que usted implemente un lenguaje de programación interpretado para lo cual usted es libre de proponer la sintaxis e implementar el interpretador para darle la semántica que se describe a continuación:

2.1. Valores:

- **Valores denotados:** enteros, flotantes, hexadecimales, octales, cadenas de caracteres, booleanos (true, false), procedimientos.
- **Valores expresados:** enteros, flotantes, hexadecimales, octales, cadenas de caracteres, booleanos (true, false), procedimientos.

Sugerencia: trabaje los valores enteros, flotantes, hexadecimales y octales desde la especificación léxica. Implemente cadenas de caracteres, booleanos (true, false) y procedimientos desde la especificación gramatical.

2.2. Características del Lenguaje BASICO-I

El lenguaje debe permitir utilizar:

- **Identificadores:** Son secuencias de caracteres alfanuméricos que comienzan con una letra
- **Definiciones:** Permiten crear (o ligar) variables a valores

- **Condicionales:** Son estructuras para controlar el flujo de un programa
- **Expresiones:** las estructuras sintácticas son una expresión y todas las expresiones producen un valor
- **Primitivas booleanas:** $<$, $>$, $<=$, $>=$, $==$, $!=$, $, ==$, *and*, *or*, *not*. Estas primitivas son binarias y permiten evaluar expresiones para generar un valor booleano
- **Primitivas aritméticas para enteros:** $+$, $-$, $*$, $\%$, $/$, *add1*, *sub1*
- **Primitivas aritméticas para hexadecimales y octales:** $+$, $-$, $*$, *add1*, *sub1*
- **Primitivas sobre cadenas:** *length*, *concat*
- **Definición/invocación de procedimientos:** el lenguaje debe permitir la creación/invocación de procedimientos que retornan un valor al ser invocados
- **Definición/invocación de procedimientos recursivos:** el lenguaje debe permitir la creación/invocación de procedimientos que pueden invocarse recursivamente
- **Iteración:** el lenguaje debe permitir la definición de una estructura de repetición tipo **for**

3. (30pts) BASICO-II

Extienda el lenguaje BASICO-I y añada:

- **Asignación:** los valores de las variables pueden mutar sus valores durante la ejecución de un programa
- **Secuenciación:** el lenguaje debe permitir expresiones para la creación de bloques de instrucciones
- **Tipos de datos:** el lenguaje debe permitir manejo de tipos¹

Nota: Los pasos de valores a procedimientos deben ser por valor.

¹Pendiente por definir el alcance...

4. (20pts) BASICO-III

Extienda el lenguaje BASICO-II y añada:

- **Creación de objetos:** el lenguaje debe permitir la creación de objetos
- **Invocación de métodos y selección de campos:** el lenguaje debe permitir invocar métodos asociados a objetos y obtener los valores asociados a los campos
- **Actualización de campos:** el lenguaje debe permitir actualizar los campos asociados a un objeto

5. Evaluación BASICO-I, II, III

El proyecto podrá ser realizado en grupos de hasta 3 personas utilizando la librería SLLGEN de Dr Racket. Este debe sustentado y cada persona del grupo obtendrá una nota entre 0 y 1 (por sustentación), la cual se multiplicará por la nota obtenida en el proyecto.

Cada especificación léxica y gramatical de los lenguajes BASICO-I, II, y III debe ser socializada y aprobada por el profesor.