



Primer examen opcional práctico 50 % - Fundamentos de lenguajes de programación

Duración: 2 horas

Carlos Andres Delgado S, Ing *

25 de Marzo 2019

Nombre: _____

Código: _____

Importante: Entregue los interpretadores modificados en el campus virtual.

1. Interpretador simple

Estos puntos debe trabajarlos en el interpretador simple visto en clase.

1. (5 puntos) Agregar las expresiones True y False al interpretador estas corresponden al tipo booleano

```
--> True
#t
--> False
#t
```

2. (10 puntos) Agregar las primitivas and, or y not al interpretador

```
--> and(True, False)
#f
--> or(True, False)
#t
--> not(True)
#f
```

3. (10 puntos) Agregar las primitivas de comparación <, >, <=, >=.

4. (10 puntos) Modificar if-exp para que test-exp sea un tiempo booleano. La siguiente expresión debe funcionar:

```
let
  p = proc(x) if >(x,3) then 2 else 3
in
  (p 4)
;;La expresión retorna 2
```

2. Interpretador de procedimientos recursivos

Estos puntos debe trabajarlos en el interpretador de procedimientos recursivos vistos en clase.

5. (15 puntos) Agregar al interpretador las listas, las cuales son una construcción recursiva

`<expresion> ::= cons "[" <expresion> <expresion> "]"`

Donde la primera expresión es un valor y el segundo una lista. Así mismo construya los siguiente:

- Las primitivas **first** y **rest**. First retorna el primer elemento de la lista y rest el segundo que es una lista
- La expresión **empty** que es una lista vacía
- La primitiva **empty?** que retorna verdadero si recibe una lista vacía y falso en otro caso

El siguiente código debería funcionar:

```
letrec
  suma(1) = if empty?(1) then 0
            else +(first(1), (suma rest(1)))
in
  let
    lst = cons[ 1 cons [2 cons [3 empty]]]
    in
      (suma lst)
;;La expresión retorna 6
```

*carlos.andres.delgado@correounivalle.edu.co