



Primer examen parcial - Fundamentos de lenguajes de programación - Duración: 1 hora

Carlos Andres Delgado S, Msc *

12 de Agosto 2019

Reglas

Leer con atención:

- Los siguientes ejercicios debe trabajarlos en el interpretador de procedimientos recursivos, es fundamental que comente en el sitio del código donde realiza las modificaciones y explique porque las realiza, una solución sin estos comentarios no será tenida en cuenta.
- Entregue un sólo archivo con ambas modificaciones.
- Se le solicita **no usar comentarios como caja** ni pegue imágenes, ya que dificultan la revisión.
- Debe dejar comentados 3 ejemplos no triviales del funcionamiento. Si no realiza este paso se le quitará la mitad del punto que no cumpla.

Nota:

Enunciado

1. (30 puntos) **Registros:** Un registro es una estructura de datos que permite almacenar información de forma similar a las listas, sin embargo, se diferencia en que en lugar de indexar por números lo hace por cualquier nombre que le demos a cada elemento. Ejemplo:

```
let
  a = {gato:11 loro:12 perro:10}
  ...
```

Este registro tiene tres elementos, que son los números 11,12 y 10, pero en lugar de estar indexados por números que serían posición 1,2 y

3 respectivamente son con gato, loro y perro. Para hacer esto, en el interpretador de procedimientos recursivos (letrec) vamos a agregar la siguiente regla gramatical:

```
<expression> ::= "{" (identifier ":" expression)* "}"
```

Para esto debe:

- a) (5 puntos) Agregar la expresión gramatical correspondiente a los registros y su caso en eval-expression
- b) (8 puntos) Crear la primitiva record->list que toma los elementos del registro y retorna la lista equivalente, ejemplo:

```
let
  a = {gato:11 loro:12 perro:10}
  in
    record->list(a)
```

Retorna [11 12 10]

- c) (8 puntos) Crear la primitiva label que toma un registro y retorna las etiquetas asociadas, ejemplo

```
let
  a = {gato:11 loro:12 perro:10}
  in
    record->list(a)
```

Retorna [gato loro perro]

- d) (9 puntos) Crear la primitiva nth, que recibe un registro y un símbolo, este retorna el elemento de la posición símbolo, Ejemplo:

```
let
  a = {gato:11 loro:12 perro:10}
  in
    nth(a, loro)
```

Retorna 12.

¡Exitos!

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