



# Fundamentos de compilación e interpretación de lenguajes de programación

## Primer exámen parcial, Duración 2 horas

Carlos Andres Delgado S, Msc

carlos.andres.delgado@correounivalle.edu.co

03 de Abril de 2025

1. ¿Que se debe tener en cuenta al intentar implementar procedimientos recursivos con let?
  - a) Se requiere un nuevo tipo de expresión, ya que podríamos tener definiciones recursivas de ligaduras
  - b) let sólo sirve para ligar valores que no sean procedimientos
  - c) Se debe cambiar la forma de construir de clausura para tener el ambiente actual y no el anterior
  - d) Al extender del ambiente anterior, los procedimientos no se conocen a sí mismos
  - e) Las clausuras no contienen el cuerpo del procedimiento, por lo cual no puede conocerlo
2. En un TAD las funciones que contamos son constructores, predicados y extractores que permiten:
  - a) Extractores: Permiten conocer que variante pertenece un tipo de dato
  - b) Los predicados y extractores se denominan procedimientos observadores
  - c) Constructor: Permite crear un elemento perteneciente al TAD
  - d) Predicado: Permite insertar un elemento nuevo al TAD
  - e) Un programador puede usar los predicados para conocer los valores internos de un TAD
3. Indique cuales de las siguientes opciones son correctas con respecto a la receta de diseño de tipos de datos recursivos
  - a) Incluir una función de validación de datos, esto para asegurar que los constructores trabajan apropiadamente
  - b) Incluir un extractor para cada pieza de dato
  - c) Incluir un constructor para cada clase de dato
  - d) Incluir una representación de listas o procedimientos para manejar el dato en Racket fácilmente
  - e) Incluir un predicado para cada clase de dato en el tipo de dato
4. ¿Que es sintaxis concreta o representación externa?
  - a) Es el código binario generado por un compilador de un lenguaje.
  - b) Es código en un lenguaje de bajo nivel.
  - c) Es código en un lenguaje de alto nivel.
  - d) Es la representación de un tipo de dato a partir de su gramática.
  - e) Es una estrategia para transformar código en un lenguaje de bajo nivel a uno de alto nivel.
5. Un parser es un procedimiento que:
  - a) Determina si una variable esta ligada o no
  - b) Permite buscar los valores de ligaduras en los ambientes
  - c) Transforma sintaxis abstracta a concreta
  - d) Extrae las unidades significativas de la sintaxis concreta
  - e) Transforma sintaxis concreta a sintaxis abstracta
6. Dada la siguiente definición de TAD

```
(define-datatype dato dato?
  (datoA (x number?))
  (datoB (y symbol?) (z (list-of symbol?)
                        (w dato?))))
)
```

¿Que declaraciones son correctas?

  - a) (datoB 'x '(a b c) (datoA 5))
  - b) (datoA 'x)
  - c) (datoB 'z '(t u v) (datoB 'y '(a b c) (datoA 3)))
  - d) (datoB 'p (datoB 'x '(a b c) (datoA 3)))
  - e) (datoB 'z '(a b c))



¿Que limitación tiene SLLGEN con respecto a la gramáticas?

7.
  - a) No se puede tener variables mutables
  - b) Sólo puede reconocer números enteros como datos
  - c) No se puede diferenciar casos que inician igual por la izquierda, ejemplo var x con var x = 10
  - d) Se deben hacer todos los datatypes con define-datatype para que el interprete funcione
  - e) No se pueden tener expresiones de más de una línea

```
let
  a = let x = +(x,z) in (f x y)
  b = let x = +(x,y) in (f x z)
  c = let x = +(x,x) in (f y z)
in
letrec
  g(x,y) = if >(y,0) then +(*(2, x),(g +(x,1)
    -(y,1))) else let a = +(a,x) in (f a b)
in
  (proc (n,m) (g n m) b c)
```

8. Indique que es cierto con respecto a un tipo abstracto de dato (TAD):

- a) Si la implementación interna cambia, el programador debe ajustar sus funciones
  - b) Tienen implementación e interfaz
  - c) Si la implementación interna cambia, el programador no nota la diferencia
  - d) Tienen Representación interna y predicados
  - e) Tienen dos casos, TAD vacío que es el base y TAD extendido que permite extender a partir de una definición anterior

9. Cuando cambiamos la representación interna de un tipo de dato (listas, procedimientos, bits, etc) ¿Que sucede con las funciones externas que usan la interfaz?

- a) Cambian, es necesario reestructurar las funciones
  - b) Cambian levemente, es necesario ajustar los condicionales de las funciones
  - c) No cambian, pero el cambio de la representación puede implicar nuevas funciones auxiliares
  - d) No cambian, el cambio de representación no las afecta
  - e) Cambian, pero solo es necesario cambiar los condicionales y los llamados a los extractores

10. Un ambiente es un tipo de dato recursivo que:

- a) El caso de ambiente vacío tiene una lista de identificadores y una lista de valores
  - b) Permite representar sintaxis abstracta desde sintaxis concreta
  - c) Tiene dos casos ambiente vacío y ambiente extendido
  - d) Tiene tres casos: ambiente vacío, ambiente extendido y ambiente sobreextendido
  - e) El caso de ambiente extendido contiene lista de identificadores, lista de valores y ambiente que extiende

Conteste las preguntas de 11 a 15 de acuerdo al siguiente código:

Considere el ambiente inicial env0 (x,y,z,f), (4,2,1, closure '(a,b) +(2,+(a,b)) empty-env) Dibuje los ambientes para la expresión:

11. ¿Cuantos ambientes del procedimiento f se generan?

- a) 2
- b) 3
- c) 4
- d) 5
- e) 6

12. Para los valores de a, b, c es cierto que:

- a) b = 6
- b) a = 3
- c) b = 5
- d) a = 6
- e) a = 7

13. El valor final de la expresión es:

- a) 159
- b) 144
- c) 129
- d) 122
- e) 153

14. a = let x = +(x,z) in .... ¿Donde extiende el ambiente que contiene x = +(x,y)?

- a) Ambiente del procedimiento de f que se invoca
- b) Ambiente inicial env0
- c) Ambiente vacio
- d) Ambiente que contiene a,b,c
- e) Ambiente recursivo de g(x,y)

15. Para los valores de a, b, c es cierto que:

- a) c = 6
- b) a = 6
- c) c = 5
- d) a = 9
- e) a = 7