



# Fundamentos de compilación e interpretación de lenguajes de programación

## Primer examen parcial, Duración 2 horas

Carlos Andres Delgado S, Msc

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

03 de Abril de 2025

1. El ambiente extendido recursivo recibe:
  - a) procnames: lista de simbolos
  - b) old-env: ambiente
  - c) condicionales: lista de símbolos
  - d) cuerpos: lista de expresiones
  - e) lidss: lista de lista símbolos (argumentos)
2. Un tipo abstracto de dato (TAD) tiene:
  - a) Constructores e interfaz
  - b) Implementación e interfaz
  - c) Predicados e implementación
  - d) Representación interna y predicados
  - e) Representación interna e implementación
3. ¿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.
4. Dada la siguiente especificación inductiva:
$$(0, 1) \in S \frac{(i,j) \in S}{(i+1,3j) \in S}$$
Marque los elementos que pertenecen a esta especificación:
  - a) (4, 81)
  - b) (2, 9)
  - c) (2, 12)
  - d) (3, 0)
  - e) (1, 3)
5. 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
6. ¿Que sucede cuando cambiamos la implementación de un tipo abstracto de dato desde el punto de vista del programador (quien usa el lenguaje)?
  - a) El programador debe cambiar las funciones de la interfaz para adaptarlas al nuevo tipo de implementación
  - b) Las funciones de la interfaz no sirven y se hace necesario una reconstrucción del tipo de dato
  - c) No hay efecto, ya que el programador usa estas funciones y no trabaja directamente con la implementación
  - d) Se debe reestructurar todo el código, ya que la implementación exige cambiar las funciones ya realizadas por el programador.
7. ¿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

```
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. 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

9. 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

10. ¿Que es un árbol de sintaxis abstracta?

- a) Es el código binario generado por un compilador de un lenguaje.
- b) Es el código fuente de un lenguaje de programación.
- c) Es una representación de cada parte del programa de acuerdo a las reglas gramaticales
- d) Es una representación que se muestra al programador para que entienda los resultados del programa.
- e) Es una estructura de datos que se utiliza para representar las estructuras de datos.

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:

```
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
```

11. Que valores son validos en los llamados recursivos en `g(x,y)` para `x`

- a) 8
- b) 9
- c) 12
- d) 14
- e) 18

12. 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`

13. En el procedimiento recursivo `g(x,y)` de que ambiente extiende el ambiente que contiene `a` en la expresión `let a = +(a,x) in ...?`

- a) Ambiente del llamado de `(f a b)`
- b) Ambiente inicial `env0`
- c) Ambiente que extiende del recursivo donde `y = 1`
- d) Ambiente que extiende del recursivo donde `y = 0`
- e) Ambiente recursivo inicial de `g(x,y)`

14. En el tercer llamado recursivo de `g(x,y)` ¿Cual es el valor de la suma acumulada?

- a) 45
- b) 89
- c) 23
- d) 69
- e) 72

15. El valor de `a` en `let a = +(a,x)` en el momento que es invocado es:

- a) 18
- b) 14
- c) 26
- d) 23
- e) 31



# 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

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:

```
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)
```

11. ¿Cuántos 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 vacío  
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



# 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. 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
2. ¿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.
3. ¿Porque se debe agregar la expresión letrec?
  - a) Para poder declarar procedimientos, dado que en el let no podemos.
  - b) Dar más poder expresivo a las clausuras, al agregar más información.
  - c) Debido a que en el let las definiciones no pueden conocerse así mismas.
  - d) Porque el let no permite declarar procedimientos en su área de definiciones.
4. 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
5. Dada la siguiente especificación inductiva:
$$(0, 1) \in S \frac{(i,j) \in S}{(i+1,3j) \in S}$$
Marque los elementos que pertenecen a esta especificación:
  - a) (4, 81)
  - b) (2, 9)
  - c) (2, 12)
  - d) (3, 0)
  - e) (1, 3)
6. 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
7. ¿Que es un árbol de sintaxis abstracta?
  - a) Es el código binario generado por un compilador de un lenguaje.
  - b) Es el código fuente de un lenguaje de programación.



- c) Es una representación de cada parte del programa de acuerdo a las reglas gramaticales
  - d) Es una representación que se muestra al programador para que entienda los resultados del programa.
  - e) Es una estructura de datos que se utiliza para representar las estructuras de datos.
- b)  $a = 6$
  - c)  $c = 5$
  - d)  $a = 9$
  - e)  $a = 7$

8. 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

9. ¿Que limitación tiene SLLGEN con respecto a la gramáticas?

- 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  $\text{var } x$  con  $\text{var } x = 10$
- 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

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

- a) Representa sintaxis abstracta en concreta
- b) Relaciona variables con sus valores
- c) Implementa gramáticas
- d) Representa sintaxis concreta en sintaxis abstracta
- e) Representar variables libres y ligadas

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

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

```
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)
```

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

- a)  $c = 6$

12. El valor de a en  $\text{let } a = +(a, x)$  en el momento que es invocado es:

- a) 18
- b) 14
- c) 26
- d) 23
- e) 31

13.  $a = \text{let } x = +(x, z) \text{ in } \dots$  ¿Donde extiende el ambiente que contiene  $x = +(x, y)$ ?

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

14. En el tercer llamado recursivo de  $g(x, y)$  ¿Cual es el valor de la suma acumulada?

- a) 45
- b) 89
- c) 23
- d) 69
- e) 72

15. El valor final de la expresión es:

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



# 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. 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
2. ¿Que es una función predicado?
  - a) Es una función que construye un valor del tipo de dato.
  - b) Permite extraer campos del tipo de dato
  - c) Permite comparar dos datos del mismo tipo.
  - d) Permite destruir un dato.
  - e) Permite preguntar si un dato pertenece a un tipo de dato específico.
3. 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
4. ¿Porque se debe agregar la expresión letrec?
  - a) Para poder declarar procedimientos, dado que en el let no podemos.
  - b) Dar más poder expresivo a las clausuras, al agregar más información.
  - c) Debido a que en el let las definiciones no pueden conocerse así mismas.
  - d) Porque el let no permite declarar procedimientos en su área de definiciones.
5. Un ambiente es un tipo de dato recursivo que:
  - a) Representa sintaxis abstracta en concreta
  - b) Relaciona variables con sus valores
  - c) Implementa gramáticas
  - d) Representa sintaxis concreta en sintaxis abstracta
  - e) Representar variables libres y ligadas
6. 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
7. 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

8. Dada la siguiente especificación inductiva:

$$(0, 1) \in S \quad \frac{(i,j) \in S}{(i+1,3j) \in S}$$

Marque los elementos que pertenecen a esta especificación:

- a) (4, 81)
- b) (2, 9)
- c) (2, 12)
- d) (3, 0)
- e) (1, 3)
9. ¿Que sucede cuando cambiamos la implementación de un tipo abstracto de dato desde el punto de vista del programador (quien usa el lenguaje)?

- a) El programador debe cambiar las funciones de la interfaz para adaptarlas al nuevo tipo de implementación
- b) Las funciones de la interfaz no sirven y se hace necesario una reconstrucción del tipo de dato
- c) No hay efecto, ya que el programador usa estas funciones y no trabaja directamente con la implementación
- d) Se debe reestructurar todo el código, ya que la implementación exige cambiar las funciones ya realizadas por el programador.

10. 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))

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:

```
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)
```

11. El valor de a en let a = +(a,x) en el momento que es invocado es:

- a) 18
- b) 14
- c) 26
- d) 23
- e) 31

12. 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

13. ¿Cuántos ambientes del procedimiento f se generan?

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

14. 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

15. En el procedimiento recursivo g(x,y) de que ambiente extiende el ambiente que contiene a en la expresión let a = +(a,x) in ...?

- a) Ambiente del llamado de (f a b)
- b) Ambiente inicial env0
- c) Ambiente que extiende del recursivo donde y = 1
- d) Ambiente que extiende del recursivo donde y = 0
- e) Ambiente recursivo inicial de g(x,y)