



# Fundamentos de lenguajes de programación

Duración 2 horas

Carlos Andres Delgado S, Msc

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

13 de Noviembre de 2021

## Reglas

- Debe entregar un archivo en formato PDF con la solución de su examen adjunto al enlace de entrega, el cual **sólo acepta un archivo en este formato**.
- Usted puede tomar fotos a los procedimientos realizados a mano o bien usar el editor de ecuaciones y así estructurar su documento.
- Integre estas capturas en el archivo de entrega, **no se aceptan bajo ninguna circunstancia enlaces externos**.
- Las capturas de los puntos deben estar en buena calidad, **si alguna no se entiende no se le valdrá**.
- Sea ordenado en las capturas de sus puntos, **no se valdrán puntos desorganizados y que no sea fácil entender su respuesta**.
- Debe explicar el procedimiento realizado en cada uno de los puntos, no se considera válido únicamente escribir la respuesta. **Es decir, toda respuesta debe ir argumentada y sustentada** de acuerdo a los conceptos vistos en el curso.
- El examen puede ser realizado en parejas, hacer **un sólo envío por pareja**. Si los dos envían, calificaré únicamente el primer envío que reciba.
- El examen debe ser entregado en el formulario de google especificado por el docente vía correo electrónico. **No se aceptarán entregas por otro medio**.
- En el formulario de google debe colocar los nombres y códigos de los estudiantes. **No se aceptarán reclamos por omisión de esta información**.
- El examen tiene una duración de 2 horas, es decir que va hasta las 4:00:00pm, de allí se dan 30 minutos de gracia para entregarlo, es decir se recibe sin penalización hasta las 4:30:00pm.
- Usted puede entregar a partir de las 4:30:01pm pero cada 5 minutos de retraso o fracción le descontaré 0.3 en la nota. Por ejemplo, si entrega a partir de las 4:35:01 pm la penalización es 0.6. **Se debe entender que es un tiempo extra para la organización de la entrega**.

## Recomendaciones

- Si tiene una cámara que toma fotos en alta calidad, se recomienda trabajar en 720p para que el archivo no resulte de gran tamaño
- Puede utilizar aplicaciones como CamScanner, las cuales permiten funciones premium con su correo institucional. Estas aplicaciones optimizan el tamaño y calidad de las imágenes.
- En caso de trabajar en parejas puede usar una herramienta colaborativa como Google Docs para integrar su trabajo, estas permiten integrar fácilmente su trabajo y exportar directamente en PDF.
- No se recomienda que comparta sus fotos en herramientas de mensajería, ya que estas le reducen la calidad en gran medida
- En caso de que su archivo PDF es grande, puede usar la impresora PDF como estrategia para reducir su tamaño.
- Tenga en cuenta que los 30 minutos que doy son para organizar su entrega teniendo en cuenta los posibles dificultades de conexión. Esta regla depende de su situación en particular y usted debe ser consciente de usar correctamente este tiempo.

# 1. Enunciado

1. (40 puntos) En el interpretador de procedimientos recursivos agregar los arreglos, bajo las siguientes condiciones

- Los arreglos se declaran de dos formas:  $[X_1, X_2, \dots, X_N]$ , donde  $X_i$  es un valor expresado y con la primitiva `list` seguida de un número  $n$ , esta última creará un arreglo de ceros de tamaño  $n$ .

```
let
  a = [1,2,3,4]
  b = list(4)
in
...
```

- Los arreglos se acceden mediante la sentencia `access`, la cual recibe un identificador y una expresión que indica la posición del arreglo que se desea acceder.

`<expression> ::= "access" <identifier> "." <expression>`

Ejemplo:

```
let
  a = [1,2,3,4]
  b = list(4)
in
access a.2  %Retorna 3
```

Ejemplo:

```
let
  a = [1,2,3,4]
  b = list(4)
in
access b.0  %Retorna 0
```

Ejemplo:

```
let
  a = [1,2,3,4]
  b = list(4)
in
access b.4 %Retorna un mensaje de error
```

Esta retorna el elemento de la posición indicada tomando como posición inicial 0, en caso de introducir un número negativo o un valor que es igual o mayor que el tamaño del arreglo se debe retornar un mensaje de error.

- Se tiene la primitiva length la cual retorna el número de elementos que tiene un arreglo.

```
let
  a = [1,2,3,4]
  b = list(4)
in
  length(a)  %Retorna 4
```

- (10 puntos) Escriba las sentencias que se agrega en la especificación gramatical
- (15 puntos) Implemente la creación de arreglos de las dos formas especificadas
- (15 puntos) Implemente el acceso y la primitiva length.

- (30 puntos) Considerando el ambiente inicial (x,y,z,f), (3,6,9, closure '(a,b) \*(2,+(a,b)) empty-env) Dibuje los ambientes para la expresión:

```
let
  m = (f (f x y) (f x y))
  n = let t = (f x y) z = (f y z) in +(t,z)
  q = proc(x,y) proc(u,v) let w = +(u,v,x,y) in proc(s,k) +(*(s,w)
    ,*(k,w))
in
  let
    u = (q m n)
    v = (q x z)
  in
    let
      t = (u m n)
      q = (v (f m n) (f n m))
    in
      +((t m n) ,(q m n))
```

El resultado es 87840

3. (30 puntos) Considerando el ambiente inicial  $(x, y, z, f)$ ,  $(3, 6, 9, \text{closure } '(a, b) * (2, + (a, b)) \text{ empty-env})$  Dibuje los ambientes para la expresión:

```
let
  a = (f x y)
  b = (f x y)
in
  letrec
    g(x, y) = if let t = -(x, 3) in >(t, 0)
                  then (k -(x, 1) y)
                  else let t = proc(a) *(a, 3) in (t +(a, b))
    k(x, y) = if >(x, 0)
                  then (g -(x, 1) +(y, 2))
                  else (f (f a b) (f a b))
  in
    let
      m = (g y z)
      n = (k x z)
    in
      +(m, n)
```

El resultado es 216