



Programación Declarativa
Ingeniería Informática
Cuarto curso. Primer cuatrimestre



Escuela Politécnica Superior de Córdoba
Universidad de Córdoba

Curso académico: 2019 - 2020

Práctica número 1.- Introducción al lenguaje Scheme

• Observaciones:

- Sólo se han de presentar los ejercicios marcados con un asterisco (*), que deberán estar contenidos en un mismo fichero.
 - **IMPORTANTE:** todas las funciones deberán tener un comentario de cabecera
 - Nombre de la función
 - Objetivo
 - Descripción de la solución (salvo que se deduzca de forma inmediata)
 - Significado de los parámetros de entrada.
 - Significado del resultado que devuelve.
 - Funciones auxiliares a las que llama.
1. **Constantes y literales:** teclea las siguiente constantes y literales (creados con la forma especial **quote** o con la comilla simple) y comprueba el resultado devuelto por el intérprete de *Scheme*:

; Los comentarios comienzan con el símbolo de “punto y coma”

<code>#t</code>	<code>;; constante lógica de verdadero</code>
<code>3</code>	<code>;; número entero</code>
<code>20.5</code>	<code>;; número real</code>
<code>"ejemplo de cadena"</code>	<code>;; se utilizan comillas dobles para delimitar las cadenas</code>
<code>'dato</code>	<code>;; no debes olvidar las comillas de cierre</code>
<code>'dato</code>	<code>;; se utiliza la comilla simple para crear un literal</code>
<code>(quote dato)</code>	<code>;; también se puede utilizar quote para crear un literal</code>
<code>dato</code>	<code>;; la variable dato no es un literal</code>
	<code>;; y producirá un error porque posee no todavía un valor</code>
<code>#t</code>	<code>;; las constantes lógicas también son literales</code>
<code>(quote #t)</code>	
<code>'3</code>	<code>;; los números también son literales</code>
<code>(quote 3)</code>	
<code>'20.5</code>	
<code>(quote 20.5)</code>	
<code>(quote "ejemplo de cadena")</code>	<code>;; una cadena también es un literal</code>
<code>(+ 2 3)</code>	<code>;; expresión aritmética con notación prefija</code>
<code>'(+ 2 3)</code>	<code>;; la expresión aritmética se convierte en un literal y "no" se evalúa</code>

(quote (+ 2 3)) ;; la expresión aritmética se convierte en un literal y “no” se evalúa
 '(a b c) ;; lista de literales
 (quote '(a b c)) ;; otra forma de crear una lista de literales
 '(Ana Luis Juan) ;; lista de literales
 (quote (Ana Luis Juan)) ;; otra forma de crear una lista de literales

2. Teclea las siguientes expresiones aritméticas y comprueba los resultados.

; Siempre se debe separar el operador de los argumentos
 (+ 2 3)

; Si no se separa el operador del argumento, se producirá un error
 (+2 3)

(+ 0.1)	(+ 0.001)	(+ 0.00000001)	(+ 3)
(+ 3 4)	(+ 3 4 5)	(+ 3 4.)	(+ 3 4.0)
(+)			
(- 2)	(- 10 2)	(- 10 3 1)	(- 10 3. 1)
(* 2)	(* 2 3 4)	(* 2.0 3 4)	(*)
(/ 5)	(/ 5.)	(/ 10 2)	(/ 8 3)
(/ 8. 3)	(/ 8 3.0)		

; Aproximación racional al número π
 (/ 355 113)

; Aproximación al número π con seis decimales exactos.
 (/ 355.0 113)

; Se divide el primer argumento por el producto de los demás
 (/ 60 3 5 4)

; Combinación de operadores
 (/ (* 9 4 3) (+ 3 2))

; Expresión “sangrada” con tabuladores: más legible
 (/

 (* 9 4 3)

 (+ 3 2)

)

3. Escribe las siguientes expresiones aritméticas con notación prefija:

a. $2 * 3 + 4$

b. $- 2 * 3 + 4 * (5 - 2)$

c. $\frac{(5 + 2) (5 - 2)}{5 * 4 - 3 * 6 + 1}$

4. Utiliza la forma especial **define** para declarar las siguientes variables y asignarles los valores que se indican:

Variable	Valor
iva	18
mayor-edad	18
meses	12
x	2.5
y	-12.3
z	$2x + y^3$
partido1	36.5
partido2	30.75
blanco	2.55
nulo	0.34
;; comprueba si el intérprete admite variables acentuadas	
abstención	100 - partido1 - partido2 - blanco - nulo
celsius	19.5
fahrenheit	$32.0 + (9.0/5.0) \text{ celsius}$

5. ¿Qué ocurre si se aplica **set!** sobre una variable no definida previamente?
Por ejemplo:

(set! votantes 23732)

6. Define las siguientes variables y escribe en *Scheme* las expresiones asociadas a las funciones matemáticas predefinidas que se indican:

Variable	Valor
a	1
b	2
c	-3
pi	(acos -1.0)

Función	Significado	Ejemplo	Scheme
<i>(abs x)</i>	Valor absoluto de x	$abs(a^2 - b^2)$	
<i>(sqrt x)</i>	Raíz cuadrada de x	$\sqrt{b^2 - 4ac}$	
<i>(square x)</i>	Cuadrado de x	$(3a-2b+c)^2$	No existe
<i>(exp x)</i>	Exponencial de x	e^{2a}	
<i>(log x)</i>	Logaritmo neperiano de x	$log(e^a)$	
<i>(expt x y)</i>	Potencia: x^y	$(2a-b)^c$	
<i>(sin x)</i>	Seno de x	$sin(2 pi)$	
<i>(cos x)</i>	Coseno de x	$cos(pi/2)$	
<i>(tan x)</i>	Tangente de x	$tan(2 pi)$	
<i>(asin x)</i>	Arco seno de x	$asin(-0.5)$	
<i>(acos x)</i>	Arco coseno de x	$acos(0.5)$	
<i>(atan x)</i>	Arco tangente de x	$atan(1.0)$	
<i>(atan x y)</i>	Arco tangente de x/y	$atan(a/b)$	
<i>(max x₁ x₂ ...)</i>	Máximo de $x_1 x_2 \dots$	$max(a,b,c)$	
<i>(min x₁ x₂ ...)</i>	Mínimo de $x_1 x_2 \dots$	$min(2a,3b,4c)$	
<i>(gcd x₁ x₂ ...)</i>	Máximo común divisor	$gcd(12,15,-18)$	
<i>(lcm x₁ x₂ ...)</i>	Mínimo común múltiplo	$lcm(12,15,-18)$	

Función	Significado	Ejemplo	Scheme
<i>(floor x)</i>	Mayor entero no más grande que x	<i>floor(-2.7)</i> <i>floor(7.5)</i>	
<i>(ceiling x)</i>	Menor entero no más pequeño que x	<i>ceiling(-2.7)</i> <i>ceiling(7.5)</i>	
<i>(truncate x)</i>	Entero más próximo a x cuyo valor absoluto no es más grande que el valor absoluto de x	<i>truncate(-2.7)</i> <i>truncate(7.5)</i>	
<i>(round x)</i>	Entero más próximo a x; redondeando a un número par si x está justo entre dos enteros.	<i>round(-2.5)</i> <i>Round(7.5)</i>	
<i>(modulo x y)</i>	Resto de la división entera (Signo del divisor)	<i>modulo (12, 5)</i> <i>modulo(12, -5)</i> <i>modulo(-12, 5)</i>	
<i>(quotient x y)</i>	Cociente de la división entera	<i>quotient(12,5)</i>	
<i>(remainder x y)</i>	Resto de la división entera (Signo del dividendo)	<i>remainder(12, 5)</i> <i>remainder(12,-5)</i> <i>remainder(-12,5)</i>	

7. (*) Codifica funciones que permitan calcular el valor del término general de las siguientes sucesiones numéricas:
- $a_n = \left(1 + \frac{1}{n}\right)^n$
 - Esta sucesión numérica converge al número $e = 2.718281828459045$
 - $a_n = C \left(1 + \frac{i}{100}\right)^n$
 - Esta sucesión numérica permite calcular la cantidad que se obtiene al depositar una cantidad C durante n años con un interés del $i\%$.
8. (*) Codifica funciones de conversión entre las siguientes unidades de medida:
- Julios a calorías
 - Calorías a julios
 - Grados Celsius a grados Fahrenheit.
 - Ejemplos: $0^\circ\text{C} \rightarrow 32^\circ\text{F}$, $100^\circ\text{C} \rightarrow 212^\circ\text{F}$
 - Grados Fahrenheit en grados Celsius.
9. (*) Codifica las siguientes funciones que calculan áreas de figuras geométricas del plano:
- areaTriangulo**
 - Calcula el área del triángulo a partir de sus lados.
 - Utiliza la **fórmula de Herón**.
 - areaRombo**
 - Calcula el área del rombo a partir de sus diagonales.
 - areaTrapezio**
 - Calcula el área del trapecio a partir de sus bases y altura.
10. (*) Codifica las siguientes funciones de distancias entre puntos del plano:
- D2: distancia euclidiana o distancia L_2** entre dos puntos $P_1 = (x_1, y_1)$ y $P_2 = (x_2, y_2)$.

$$D2(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

- b. **D1: distancia de Manhattan, distancia de la ciudad de los bloques o distancia L_1** entre dos puntos $P_1 = (x_1, y_1)$ y $P_2 = (x_2, y_2)$.

$$D1(P_1, P_2) = |x_2 - x_1| + |y_2 - y_1|$$

- c. **Dmax: distancia de ajedrez, distancia de Chebyshev o distancia L_∞** entre dos puntos $P_1 = (x_1, y_1)$ y $P_2 = (x_2, y_2)$.

$$Dmax(P_1, P_2) = \max(|x_2 - x_1|, |y_2 - y_1|)$$

11. (*) Codifica las siguientes funciones que calculan áreas de figuras geométricas del plano a partir de sus vértices:

a. **areaTrianguloVertices**

b. **areaRomboVertices**

- **Observaciones:**

- Las funciones deben recibir como argumentos a las coordenadas de los vértices de cada figura geométrica.
- Utiliza los comentarios para indicar en qué **orden** se han de introducir las coordenadas de los vértices.
- Utiliza la función auxiliar **D2** para calcular la distancia entre los vértices: lados y diagonales (en su caso).
- Utiliza como funciones auxiliares a las funciones **areaTriangulo** y **areaRombo** del ejercicio nº 9.

12. (*) Codifica las siguientes funciones:

a. Función denominada **distanciaPuntoRecta**

- Ha de calcular la distancia de un punto $P = (x_0, y_0)$ a una recta $r \equiv a x + b y + c = 0$ mediante la siguiente fórmula

$$d(P, r) = \frac{|ax_0 + by_0 + c|}{\sqrt{a^2 + b^2}}$$

b. Utiliza la forma especial **let** para codificar la función denominada **distanciaPuntoRecta2**

- La función ha de calcular la distancia de un punto $P = (x_0, y_0)$ a la recta que pasa por otros dos puntos $P_1 = (x_1, y_1)$ y $P_2 = (x_2, y_2)$.
- **Sugerencia:**
 - En primer lugar, determina los coeficientes de la recta $r \equiv a x + b y + c = 0$ que pasa por los puntos P_1 y P_2 .
 - A continuación, utiliza la función del apartado “a”.

13. (*) Utiliza la forma especial **let** para codificar una función que calcule el área de un trapecio a partir de sus vértices

- **areaTrapecioLetVertices**

- La función recibirá como argumentos las coordenadas de los vértices del trapecio.
- Utiliza los comentarios para indicar en qué “orden relativo” se han de introducir las coordenadas de los puntos del trapecio para formar las bases.
- Utiliza las siguientes funciones auxiliares:
 - **D2:** distancia euclidiana entre dos puntos o vértices.
 - **distanciaPuntoRecta2:** distancia de un punto a una recta definida por dos puntos.
 - **areaTrapecio:** área del trapecio conocidas las bases y la altura.