

# Lenguajes de Programación

## Práctica 1

Karla Ramírez Pulido

Manuel Soto Romero

Alejandro Hernández Mora

Alma Rocío Sánchez Salgado

Silvia Díaz Gómez

Semestre 2021-1

Facultad de Ciencias, UNAM

**Fecha de inicio: 30 de septiembre de 2020**

**Fecha de entrega: 7 de octubre de 2020**

## 1. Objetivos

Utilizar los tipos de datos primitivos en *Racket*, para familiarizarse con sintaxis, *IDE*, estructuras de control y llamadas recursivas.

## 2. Ejercicios

Programa las siguientes funciones en *Racket*. Eres libre de utilizar cualquier IDE o cualquier editor de texto, pero se sugiere fuertemente utilizar *Dr. Racket*, el cuál estaremos utilizando en el laboratorio. Está prohibido utilizar funciones primitivas del lenguaje que resuelvan directamente los ejercicios. **La entrega de esta práctica es individual.**

A continuación vienen las firmas de las funciones y un ejemplo para que te des una idea del resultado a obtener.

1. Un predicado que reciba un número natural  $n$ , que devuelva verdadero si  $n$  es par y falso en otro caso.

---

```
;; esPar?: number -> boolean
(define (esPar? n) ...)

> (esPar? -2)
#t
> (esPar? 0)
#t
```

---

2. Una función que reciba un número natural  $n$  y devuelva la lista ordenada de menor a mayor, de los números menores o iguales a  $n$ , desde el cero.

---

```
;; menores: number -> (listof number)
(define (menores n) ...)

> (menores 13)
'(0 1 2 3 4 5 6 7 8 9 10 11 12 13)
```

---

3. Una función que reciba un número natural  $n$  y devuelva la lista de números pares desde 0 hasta  $n$ .

---

```
;; pares: number -> (listof number)
(define (pares n) ...)

> (pares 10)
'(2 4 6 8 10)
```

---

4. Una función que reciba un número natural  $n$  y calcule la suma de los primeros  $n$  números naturales al cuadrado. La formula es:  $\frac{n(n+1)(2n+1)}{6}$ .

---

```
;; suma-cuadrados: number -> number
(define (suma-cuadrados n) ...)

> (suma-cuadrados 10)
385
```

---

5. Una función **recursiva** que reciba un número natural  $n$  y calcule la suma de los primeros  $n$  números naturales al cuadrado. En este ejercicio no se permite usar la fórmula dada en el ejercicio 4.

---

```
;; suma-cuadradosR: number -> number
(define (suma-cuadrados n) ...)

> (suma-cuadrados 10)
385
```

---

Considera un polinomio de segundo grado de una variable. Recordemos que este polinomio representa una parábola y que sea de segundo grado, significa que el exponente más grande de cualquier término del polinomio es 2. También significa que pueden existir a lo más dos puntos que corten al eje de las  $X$ , a estos puntos les llamamos raíces del polinomio. Decimos que un polinomio tiene:

- Dos raíces reales, si existen dos puntos que corten al eje de las  $X$ .
- Una raíz real, si existe un punto que corta al eje de las  $X$ .
- Ninguna raíz real si no existe un punto de la parábola que corte al eje de las  $X$ .

Recordemos que podemos encontrar las raíces reales de un polinomio de segundo grado utilizando la fórmula general, que es:

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Utilizando esta ecuación podemos encontrar el valor de  $x_1$  y  $x_2$ , que son las raíces reales del polinomio (si es que éstas existen), donde:

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

Un método eficaz para saber si el polinomio tiene 0, 1 ó 2 raíces reales es calcular el discriminante. El discriminante está dado por la parte dentro de la raíz cuadrada de la fórmula general, es decir que:

$$\text{discriminante} = b^2 - 4ac$$

- Si el discriminante es menor que 0, entonces el polinomio no tiene raíces reales.
- Si el discriminante es exactamente 0, entonces el polinomio tiene una raíz real.
- Si el discriminante es mayor que 0, entonces el polinomio tiene dos raíces reales.

Considera la información anterior para resolver los ejercicios 7, 8 y 9.

6. Un predicado que reciba tres números reales  $a$ ,  $b$  y  $c$ , de una expresión cuadrática y decida si la expresión tiene raíces reales. La función verifica si el discriminante es mayor o igual a cero

---

```
;; raicesReales? : number number number -> boolean
(define (raicesReales? a b c) ... )

> (raicesReales? 1 2 3)
#f
> (raicesReales? 1 5 3)
#t
```

---

7. Una función que reciba tres números reales  $a$ ,  $b$  y  $c$  y devuelva la primer raíz de la fórmula general (sumando la raíz cuadrada del discriminante). La función debe lanzar un error si el polinomio no tiene raíces reales.

---

```
;; general1: number number number -> number
(define (general1 a b c) ...)

> (general1 1 2 3)
Error general1 El polinomio no tiene raíces reales
> (general1 1 5 3)
-0.6972243622680054
```

---

8. Una función que reciba tres números reales  $a$ ,  $b$  y  $c$  y devuelva la segunda raíz de la fórmula general (restando la raíz cuadrada del discriminante). La función debe lanzar un error si el polinomio no tiene raíces reales.

---

```
;; general2: number number number -> number
(define (general2 a b c) ...)

> (general2 1 2 3)
Error general2 El polinomio no tiene raíces reales
> (general2 1 5 3)
-4.302775637731995
```

---

9. Una función que reciba una lista como parámetro y nos devuelva una lista con el orden de sus elementos invertidos.

---

```
;; reversa-lista: (listof a) -> (listof a)
(define (reversa-lista lista) ... )

> (reversa-lista (list 1 2 3 4))
'(4 3 2 1)
```

---

10. Un predicado que verifique si una lista (no necesariamente homogénea<sup>1</sup>) es un palíndromo.

---

```
;; palindromo-lista?: (listof a) -> Boolean
(define (palindromo? lista) ... )

> (palindromo? '(1 "hola" 1))
#t
```

---

Deberás respetar todas las definiciones de las funciones, es decir que esta prohibido cambiar el tipo que tengan, ya sea el de algún parámetro o el del resultado.

---

<sup>1</sup>Es decir que puede tener elementos de diferentes tipos.

### 3. Requerimientos

Recuerda que la entrega de esta práctica es individual.

Deberás entregar tu práctica a través de la plataforma **Google classroom** antes de las 23:59 del día de entrega indicado, tal y como lo indican los lineamientos de entrega. Deberás adjuntar tu código en un archivo llamado ***Practica1.rkt***. Revisa bien mayúsculas, minúsculas y espacios para el nombre del archivo que se pide.

El orden en el que aparezcan las funciones en el archivo solicitado, debe ser el orden especificado en este PDF, de lo contrario podrán penalizarse algunos puntos. Puedes utilizar funciones auxiliares, se recomienda definirla después de la función que la va a utilizar.

No dudes en aclarar tus dudas en la sesión de laboratorio y vía correo electrónico.

¡Que tengas éxito en tu primera práctica!