

EJERCITACIÓN EXTRA Nº 1 :

PROPOSICIONES Y PREDICADOS EN RACKET

Cátedra Redictado Programación I

Septiembre 2021

EJERCICIO 1. Traducciones

Traducir las siguientes expresiones dadas en lenguaje matemático o coloquial al lenguaje de programación Racket.

1. La frase “*una imagen vale más que mil palabras*” es la concatenación de las palabras *una*, con *imagen*, *vale más que mil palabras*
2. La raíz cuadrada de 25 es mayor que raíz cuadrada de 0
3. La palabra *queso* tiene exactamente 5 letras.
4. La palabra *Nemo* tiene menos de 7 letras.
5. El año 2000 es bisiesto pero el año 1900 no lo es.
6. La palabra *ananá* comienza y termina con la letra a.

EJERCICIO 2. Evaluar Expresiones

Evaluar las siguiente expresiones usando lápiz y papel, y luego verificarlas usando Racket en el modo el *Paso a Paso*. Además realice una traducción al lenguaje coloquial.

1. (zero? 1)
2. (>(string-length “azul”) 4)
3. (>(string-length “verde”) (string-length “azul”))
4. (= (+ (string-length “azul”) (string-length “anaranjado”)) (string-length “paralelepipedo”)
5. (not (= 1 2))
6. (= (sqrt -1) 0+1i)

EJERCICIO 3. Predicados

Escriba los siguientes enunciados de predicados (funciones que nos devuelven un valor del tipo **Boolean**) en Racket. Recuerde el formato de definición para esta práctica, el cuál deberá incluir el *propósito* de la función. Como por ejemplo:

```
1 ;primer-cuadrante?:Number Number -> Boolean
2 ;-----
3 ;Dada las coordenadas de un punto en el plano, determina si las mismas ↔
   corresponden al primer cuadrante, es decir, si ambas son no negativas.
4 ;-----
5 (define (primer-cuadrante? x y)
6     (and (>= x 0) (>= y 0))
7 ;-----
```

```

8 (primer-cuadrante? 1 0) ; ve #t
9 (primer-cuadrante? 0 -1) ; ve #f
10 (primer-cuadrante? -1 -1) ; ve #f
11 (primer-cuadrante? 1 1) ; ve #t

```

(A) Escribir un predicado que:

- Determine si una función es par o no. (No usar las funciones **even?** y/u **odd?** de Racket).
- Determine si un número n es múltiplo de 3.
- Determine si un número n es múltiplo de otro k .
- Observar que la función anterior es *más general* que las dos primeras. Reescriba entonces los primeros dos predicados en función de ésta última.

(B) Escribir un predicado que determine si carácter (considerarlo como string de longitud 1) es una vocal.

(C) Escribir un predicado que determine si un ángulo es agudo.

(D) Escribir un predicado que determine si una persona puede votar o no, considerando la edad, y el tipo de elección: Nacional o Provincial.

(E) Escribir un predicado que determine si una persona puede manejar un auto o no, considerando la edad, y el país en el que se encuentra. Por ejemplo: En Argentina se debe tener 18 años para manejar, pero en los EEUU, basta con tener 16 años. Considere los siguientes países: Argentina, EEUU, Brasil, Francia, Reino Unido, Alemania, Japón, Canadá, México, Nueva Zelanda, Israel, Rusia, y Grecia. (Para más datos puede consultar: [https://es.wikipedia.org/wiki/Conducción_\(vehículo\)](https://es.wikipedia.org/wiki/Conducción_(vehículo)))

(F) Escribir un predicado que:

- Determine si un valor dado es un cero (raíz) del siguiente polinomio:

$$x^2 + 2x + 1$$

- Determine si un valor dado es un cero del siguiente polinomio:

$$-x^2 + 7x - 5$$

- ¿Cómo podríamos generalizar este proceso, para que me sirva para cualquier polinomio de grado dos, es decir,

$$ax^2 + bx + c?$$

Pensar de qué forma podemos identificar a un polinomio cualquiera. *Ayuda:* Mirar bien qué es lo que cambia y lo que permanece igual en cada polinomio anterior. Probar luego nuestro predicado genérico para los siguientes casos:

- $x^2 + x + 1$
- $-x^2 - x - 2$
- $x^2 - 25$

EJERCICIO 4. Evaluación de condicionales if

Dada la siguiente definición en Racket:

```
1
2 (define PUNIC 60) ; precio unitario cuadernos
3 (define PDESC 54) ; precio descuento cuadernos (-%10)
4
5 ;precio:Number -> Number
6 ;En base a la cantidad de cuadernos vendidos calcula el
7 ;costo total de la compra.
8 ;-----
9 (define (precio cant)
10     (if (< cant 4)
11         (* PUNIC cant) ; #t caso cantidad < 4
12         (* PDESC cant) ) ) ; #f cantidad x >= 4
13 ;-----
14 (precio 1) ; ve: 60
15 (precio 10) ; ve: 540
```

y siguiendo las **Reglas de Reducción del if** vistas en clase evalúe las expresiones que se indican a continuación, y luego verifique el resultado usando el *Paso a Paso de Racket*.

1. (precio 2)
2. (precio 6)
3. (precio 0)
4. (precio -1)
5. (precio 1.5)
6. (precio 4.5)

Las últimas evaluaciones nos trajeron un problema! ¿Se puede resolver? Si usted así lo cree, modifique la función para que contemple esta situación.