

**Universidad Nacional Autónoma de México**  
**Facultad de Ciencias**  
**Lenguajes de Programación**

Actividad de laboratorio 3

**Karla Ramírez Pulido**  
karla@ciencias.unam.mx

**J. Ricardo Rodríguez Abreu**  
ricardo\_rodab@ciencias.unam.mx

**Manuel Soto Romero**  
manu@ciencias.unam.mx

16 de febrero de 2018  
Semestre 2018-2

## Objetivo

Que los estudiantes:

- Aprendan a usar las estructuras de datos básicas de Racket.
- Recuerden el concepto de recursividad y definan funciones de este estilo.
- Usen y entiendan el funcionamiento de las funciones de orden superior, *lambdas* y la primitiva `letrec`.

## Repositorio

El material necesario para completar esta actividad de laboratorio se encuentra en el repositorio de *GitHub Classroom* del curso: <https://classroom.github.com/g/jWS1ADS9>.

## Actividades

En clase se implementó una pequeña biblioteca para trabajar con vectores con elementos en  $\mathbb{R}^2$ . La actividad consiste en extender esta biblioteca para que trabaje con vectores cuyos elementos se encuentran en  $\mathbb{R}^n$ .

Resolver los siguientes ejercicios de forma limpia y en **orden**. No se permite definir funciones auxiliares, en caso de requerirlas usar funciones anónimas (*lambdas*).

**Actividad 3.1** Dar una definición para vectores (mvector x y z ...). Para definir una función que recibe varios parámetros se usa la sintaxis:

```
(define (<nombre> . <parámetros>)
  <cuerpo>)
```

Los parámetros se tratan como lista. Por ejemplo, la siguiente función suma los números pasados como parámetro.

```
;; Función suma que recibe varios parámetros.
;; suma-multiparam: number ... -> number
(define (suma-multiparam . numeros)
  (foldr + 0 numeros))

(test (suma-multiparam 1 2 3) 6)
```

```
> (mvector 1 2 3)
'(1 2 3)
```

### Actividad 3.2

1. Definir una función (suma u v) que regrese la suma de dos vectores.

```
> (suma (mvector 1 2 3 4) (mvector 5 6 7 8))
'(6 8 10 12)
```

2. Definir una función (punto u v) que regrese el producto punto de dos vectores.

```
> (punto (mvector 1 2 3 4 5) (mvector 6 7 8 9 10))
130
```

3. Definir una función (prod k u) que regrese el producto por escalar (k es el escalar). Este ejercicio debe implementarse apoyándose de map.

```
> (prod 2 (mvector 1 2 3 4 5 6))
'(2 4 6 8 10 12)
```

4. Definir una función (norma u) que regrese la norma del vector. Este ejercicio debe implementarse apoyándose de foldr o foldl.

```
> (norma (mvector 1 2 3 4 5 6 7))
11.83
```

5. Definir una función (ángulo u v) que regresa el ángulo entre dos vectores.

$$\theta = \cos^{-1} \left( \frac{u \cdot v}{|u||v|} \right)$$

```
> (angulo (mvector 1 2) (mvector 3 4))  
~0.17
```

## Punto extra

Los primeros dos equipos que escriban el resultado de **alguna** de las siguientes llamadas a función en el pizarrón y lo explique correctamente obtendrán un punto extra en la práctica 1<sup>1</sup>.

1. `> (foldl 'above (mvector 1 2) (mvector 3 4))`

2. `> (foldr 'above (mvector 1 2) (mvector 3 4))`

## Entrega

- Subir la solución al repositorio de la actividad.
- Si el ejercicio se termina en clase, levantar la mano para que el ayudante de laboratorio califique presencialmente.
- Si por cuestiones de tiempo el ejercicio no se puede calificar presencialmente, se revisará directamente desde la rama master de la actividad.

**Esta actividad sólo será tomada en cuenta a estudiantes que aparezcan en la lista de asistencia de la sesión de laboratorio y equivale a 0.33 puntos extra sobre la calificación final de la Práctica 1.**

<sup>1</sup>Al entregar este punto extra, no se tomarán en cuenta los puntos extra individuales especificados en el archivo PDF de la práctica 1.