

Teoría de Lenguajes de Programación**Proyecto 02:** Programación funcional**Formato:** Equipo / Individual**LCC/LIS****Fecha entrega:****Profesor:** M.C. Luis R. Basto

Descripción: Realizar un Jupyter Notebook utilizando el kernel iRacket con los siguientes ejercicios.

1. Defina la función B que calcule los coeficientes binomiales $B(n,k) = n!/((n-k)!*k!)$ para $n \geq 0$, y $0 \leq k \leq n$
2. Defina una función para la evaluación del número combinatorio $C(n,k)$, que utiliza la definición recursiva.
$$C(n, 0) = 1, C(n, n) = 1$$
$$C(n, k) = C(n - 1, k - 1) + C(n - 1, k), \quad n \geq 0, k \geq 0, n \geq k$$
3. Defina una función recursiva para calcular el Máximo Común Divisor de dos enteros negativos a y b con $a < b$ usando el hecho de que $MCD(a, b) = MCD(a, b-a)$.
4. Definir una función que devuelva, en una lista, todos los números primos desde un número inicial hasta un número final, ejemplo: (primos 3 10) este ejemplo devolverá '(5 7).
5. Realizar una función para hallar el valor de e definiéndola como sigue:

$$e = \frac{1}{0!} + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots$$

- a. Ejemplo de entrada: (funcionE 3)
 - b. Ejemplo de salida: 2.666
6. Realizar una función que indique la longitud de una lista
 - a. Ejemplo de entrada: (len '(2 4 5))
 - b. Ejemplo de salida: 3
 7. Realizar una función para buscar un elemento en una lista, regresar #t si lo encontró y #f si no lo encontró.
 - a. Ejemplo de entrada: (busca 4 '(2 4 5))
 - b. Ejemplo de salida: #t
 8. Realizar una función recursiva en Lisp que invierta una lista.
 - a. Ejemplo de entrada: (invierte '(2 4 5))
 - b. Ejemplo de salida: (5 4 2)
 9. Realizar una función que elimine un elemento de una lista
 - a. Ejemplo de entrada: (elimina 4 '(2 4 5))
 - b. Ejemplo de salida: (2 5)

10. Defina una función para calcular la desviación estándar de un conjunto de datos (lista). La fórmula es la siguiente:

$$s = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2}$$

NOTAS:

- Realice un manual de usuario (portada, índice, paginación, etc) que describa lo siguiente:
 - Instalación y configuración del notebook
 - Uso y ejecución del notebook
 - Ejecución de los ejercicios
- En el notebook:
 - Para cada programa, describa en qué consiste, cuáles son las entradas que recibe y cómo será la salida (en formato markdown)
 - Se debe permitir que el usuario interactúe con el código (proporcionar diferentes valores de entradas) y observe el resultado.
- Enviar a la plataforma el manual del usuario y el notebook en un sólo archivo comprimido `Proy02_En.zip`, donde n es el número de equipo.