



**ACTIVIDAD
FUNCIONES**

Aprendizaje esperado

Estructurar un algoritmo lógico utilizando estructuras de control y expresiones para dar solución a un problema de baja complejidad acorde al lenguaje java.

Actividad

Contexto:

En un sistema de monitoreo climático, se requiere transformar temperaturas entre distintas escalas: Celsius, Fahrenheit y Kelvin. Para lograrlo de manera modular y reutilizable, se solicita crear una función que realice esta conversión a Celsius, recibiendo un valor de temperatura y el tipo de unidad de origen.

Este ejercicio permite reforzar el uso de funciones con parámetros, estructuras condicionales y retorno de valores, aplicando lógica de selección múltiple.

Objetivo:

Crear una función que reciba una temperatura y su unidad original, y la convierta a grados Celsius, retornando el valor convertido. Luego, usar esta función en un programa principal que solicite datos al usuario, invoque la función y muestre el resultado en pantalla.

Instrucciones:

1. Crear una función llamada `ConversionTemperatura(temperatura, unidadOrigen)` que reciba:
 - `temperatura`: un valor real.
 - `unidadOrigen`: una cadena de texto con uno de los siguientes valores: "FAHRENHEIT", "KELVIN" o "CELSIUS".
2. Según el valor de `unidadOrigen`, realizar la conversión a grados Celsius utilizando las siguientes fórmulas:
 - De Fahrenheit a Celsius: $(temperatura - 32) * 5 / 9$
 - De Kelvin a Celsius: $temperatura - 273.15$
 - Si ya está en Celsius, retornar el mismo valor.
3. La función debe retornar la temperatura convertida a Celsius.
4. En el programa principal:

- Solicitar al usuario un valor de temperatura y su unidad de origen.
- Llamar a la función `ConversionTemperatura` con los datos ingresados.
- Mostrar el resultado convertido en pantalla.

Entregable:

- Un archivo con extensión `.psc` que contenga:
 - La definición completa de la función `ConversionTemperatura` en pseudocódigo usando `PSelnt`.
 - El programa principal que interactúe con el usuario e invoque la función.
 - Comentarios explicativos en cada sección del código (entrada, función, llamada y salida).
 - Código estructurado de forma ordenada e indentada.
 - El nombre del estudiante al inicio del archivo como comentario.