#Resolucion – Funciones de Orden Superior

Ejercicio 1:

object Ejemplo1 {  
 def main(args: Array[String]): Unit = {  
 def duplicarLista(lista: List[Int]): List[Int] = {  
 lista.map(x => x \* 2)  
 }  
  
 val numeros = List(1, 2, 3)  
 *println*(duplicarLista(numeros)) // Salida: List(2, 4, 6)  
 }  
}

Ejercicio 2:

object Ejemplo1 {  
 def main(args: Array[String]): Unit = {  
 // Definición de la función doble  
 val doble: Int => Int = (x: Int) => x \* 2  
  
 // Función que aplica otra función a una lista  
 def aplicarALista(funcion: Int => Int, lista: List[Int]): List[Int] = {  
 lista.map(funcion)  
 }  
  
 // Lista de números  
 val numeros = List(1, 2, 3, 4)  
  
 // Aplicar la función doble a la lista  
 val resultado = aplicarALista(doble, numeros)  
  
 // Imprimir el resultado  
 *println*(resultado) // Salida: List(2, 4, 6, 8)  
 }  
}

Ejercicio 3

object Ejemplo1 {  
 def main(args: Array[String]): Unit = {  
 def filtrarPares(lista: List[Int]): List[Int] = {  
 lista.filter(x => x % 2 == 0)  
 }  
  
 val numeros = List(1, 2, 3, 4, 5, 6)  
 *println*(filtrarPares(numeros)) // Salida: List(2, 4, 6)  
 }  
}

#Resolucion - Ejercicios PM

Ejercicio 1:

object Ejemplo1 {  
 def main(args: Array[String]): Unit = {  
 def parImpar(num: Int): String = {  
 num match {  
 case 0 => "Cero"  
 case n if n % 2 == 0 => "Par"  
 case \_ => "Impar"  
 }  
 }  
  
 *println*(parImpar(5)) // Salida: Impar  
 *println*(parImpar(4)) // Salida: Par  
 *println*(parImpar(0)) // Salida: Cero  
 }  
}

Ejercicio 2:

object Ejemplo1 {  
 def main(args: Array[String]): Unit = {  
 def describirTipo(dato: Any): String = {  
 dato match {  
 case s: String => s"**$**s es un string"  
 case i: Int => s"**$**i es un número entero"  
 case \_ => "Tipo desconocido"  
 }  
 }  
  
 *println*(describirTipo(10)) // Salida: 10 es un número entero  
 *println*(describirTipo("Hola")) // Salida: Hola es un string

*println*(describirTipo(Boolean))   
 }  
}

Ejercicio 3:

object Ejemplo1 {  
 def main(args: Array[String]): Unit = {  
 def contarElementos(lista: List[Int]): String = {  
 lista match {  
 case Nil => "Lista vacía"  
 case \_ => s"La lista tiene **$**{lista.length} elementos"  
 }  
 }  
 *println*(contarElementos(List(1, 2, 3))) // Salida: La lista tiene 3 elementos  
 *println*(contarElementos(Nil)) // Salida: Lista vacía  
 }  
}