

**PROGRAMACION FUNCIONAL**

**RODRIGUEZ MORENO MARCO ANTONIO**

**Aarón Hernández García**

**Actividad: 2.1**

**18/01/2024**

**PÉREZ JUÁR HERNANDEZ GARCIA**

**1.2 APLICACIONES DE LA DISTRIBUCIÓN NORMAL**

**1.3 Tipos de Datos en Programación Funcional**

En programación funcional, los tipos de datos son fundamentales para definir la naturaleza y comportamiento de los datos que se manipulan. Estos tipos ayudan a garantizar la seguridad y claridad en el código. A continuación, se describen los principales tipos de datos utilizados:

1. **Tipos de Datos Primitivos:**
   * **Enteros:** Números sin parte decimal.
   * **Flotantes:** Números con parte decimal.
   * **Caracteres:** Símbolos individuales como letras o dígitos.
   * **Booleanos:** Valores de verdad (True o False).
2. **Listas:**
   * Estructuras que almacenan secuencias ordenadas de elementos del mismo tipo.
   * Ejemplo en Haskell:



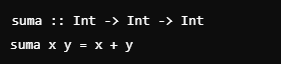
1. **Tuplas:**
   * Estructuras que agrupan un número fijo de elementos, que pueden ser de diferentes tipos.
   * Ejemplo en Haskell:



1. **Tipos de Datos Algebraicos:**
   * Permiten la creación de tipos complejos mediante la combinación de otros tipos.
   * Ejemplo en Haskell:



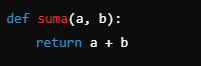
1. **Funciones como Tipos de Datos:**
   * Las funciones son tratadas como ciudadanos de primera clase, lo que significa que pueden ser asignadas a variables, pasadas como argumentos y devueltas como resultados.
   * Ejemplo en Haskell:

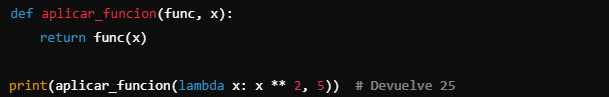


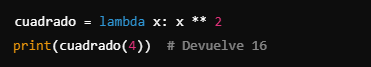
**1.4 Funciones en la Programación Funcional**

Las funciones son el núcleo de la programación funcional. Se utilizan para abstraer y encapsular comportamientos, facilitando la modularidad y la reutilización del código. A continuación, se detallan las características y tipos de funciones más relevantes:

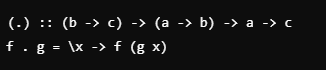
1. **Funciones Puras:**
   * Siempre producen el mismo resultado para los mismos argumentos y no tienen efectos secundarios.
   * Ejemplo en Python:



1. **Funciones de Orden Superior:**
   * Funciones que pueden recibir otras funciones como argumentos o devolverlas como resultados.
   * Ejemplo en Python:
2. **Funciones Anónimas (Lambdas):**
   * Funciones definidas sin un nombre explícito, generalmente utilizadas para operaciones simples y de corta duración.
   * Ejemplo en Python:



1. **Composición de Funciones:**
   * Proceso de combinar dos o más funciones para producir una nueva función.
   * Ejemplo en Haskell:



1. **Recursión:**
   * Técnica donde una función se llama a sí misma para resolver problemas que pueden ser descompuestos en subproblemas similares.
   * Ejemplo en Haskell:

