Tipo de Dato

Conjunto de valores y conjunto de operaciones asociadas a esos valores

- **Tipos primitivos** → Tipos de datos cuyos valores son atómicos.
  - **Ej:** Char, Boolean, Integer, enums...
- **Tipos compuestos** → Tipos de datos cuyos valores no son atómicos y pueden ser descompuestos en varias partes.
  - **Ej:** String, vectores, clases...

Tipo de Dato Predefinido (Built-in)

Tipos que ya se encuentran implementados en el lenguaje. Reflejan y abstraen el comportamiento del hardware subyacente. También pueden ser pensados como un mecanismo para clasificar los datos manipulados en un programa, y como una forma de protejer los datos de operaciones sin sentido o ilegales (Ghezzi, p. 137)

Tipo de Dato Definido por el Usuario

Son los tipos de datos creados por el usuario (valga la redundacia) por medio de un constructor de tipos provisto por el lenguaje. También pueden ser simples (*enums*) o compuestos

Constructores de tipos

Son mecanismos que permiten construir tipos de datos (valga la redundancia x2). Hay lenguajes como Pascal que permiten declarar los tipos (type mes = 1 .. 12) aparte, y hay otros lenguajes que, en general, definen la estructura del tipo al crear una instancia de dicho tipo (C, Python, Javascript)

Producto Cartesiano

Tipo de constructor cuyos tipos creados son productos cruzados de N conjutos, donde cada elemento del conjunto del producto cartesiano (lo que sería el tipo en

sí) tiene un elemento de cada uno de los conjuntos que lo componen (Sebesta, p. 309)

Correspondencia Finita

Tipo de constructor que se asemeja a una función matemática. Ghezzí lo define así “una correspondencia finita es una función que tiene como dominio un conjunto finito de valores del tipo DT (tipo de dominio), y como codominio o **rango**, un conjunto de valores de tipo RT (tipo del rango)”

- Estos constructores se pueden definir de manera **intencional**, por medio de rutinas de tipo función, o por **extensión**, por medio de un vector por ejemplo

Union

Tipo de constructor que construye los tipos como disyunciones entre dos tipos. Los capos de estos tipos uniones son mutuamente excluyentes, es decir, el objeto o variable en un determinado momento tendrá un tipo de valor u otro, pero no ambos.

- El problema con las instancias de estos tipos es que se tiene que controlar de alguna forma qué tipo de valor va a tener en un momento dado, para así no realizar operaciones ilegales o que no tienen sentido
- Una solución a este problema podría ser crear un tipo de dato, mediante producto cartesiano, que tuviera un campo que fuera del tipo unión, y otro campo que sirviera para desambiguar o discriminar el tipo de valor que tiene el objeto. Este campo se debería actualizar cada vez que se realice una asignación a cada campo correspondiente. Estos tipos de uniones se llaman **unión discriminada.** No son un constructor de tipo propiamente dicho

Recursión

Estructura que puede contener componentes del mismo tipo, por medio de un campo de tipo puntero, que le permite a la estructura crecer indefinidamente de tamaño.

Tipo de Dato Abstracto

Tipo de dato al que podemos definirle operaciones para manipular instancias, mientras la estructura de datos permanece oculta al programador. Tiene dos características:

- *Principio de encapsulamiento*
- *Ocultamiento de información*