

### Definición de Tipo Abstracto de Datos (TAD):

Existen muchas definiciones sobre lo que se entiende como **TIPO ABSTRACTO DE DATOS** o simplemente **TAD**. Se mencionan a continuación un par de definiciones generalmente aceptadas:

*“Un tipo abstracto de datos es un modelo matemático compuesto por una colección de operaciones definidas sobre el modelo y un nombre que lo identifica.”*

Aunque desde un punto de vista más práctico puede resultar más fácil de entender la definición provista por John Guttag, donde se define a los TAD simplemente como :

*“Un tipo de dato abstracto es aquél definido por el programador que puede ser manipulado de forma similar a los definidos por el sistema.”*

Muchas veces el programador necesitará hacer uso de tipos de datos no implementados en la mayoría de los lenguajes modernos. En otras palabras, muchas veces los tipos de datos simples como ser datos enteros, reales, caracteres, etc, pueden no resultar adecuados para la implementación de una solución a un problema en particular. En aquellos casos, resultará necesario en muchos casos definir nuevos tipos de datos.

### Elementos que componen un TAD:

Al igual que los datos nativos definidos por el lenguaje, un **TAD** contiene un **nombre** que corresponde a un conjunto de **valores** lícitos y de **operaciones** asociadas a los mismos.

### Especificación e Implementación de un TAD

Los valores y operaciones de un TAD que se definen mediante una **especificación** que es **independiente de la implementación** de dichos datos. De esta manera, a la hora de diseñar un TAD, se establecen claramente las diferencias entre la **especificación del TAD** y su **implementación**.

Durante la especificación simplemente se definen: el nombre del TAD, sus valores posibles y el conjunto de operaciones asociadas. Este conjunto de operaciones muchas veces es llamado **interface** del TAD. Mientras que durante la implementación se utiliza algún **lenguaje de programación** (puede ser pseudo-código) para detallar la estrategia elegida para representar el tipo de dato. Esta representación puede ser un tipo de dato nativo o también un TAD previamente definido.

También durante la implementación se detallan de manera programática o algorítmica todos los pasos necesarios para llevar cabo las operaciones previamente definidas. Un resumen en forma gráfica de esto último puede verse en la Figura 1.

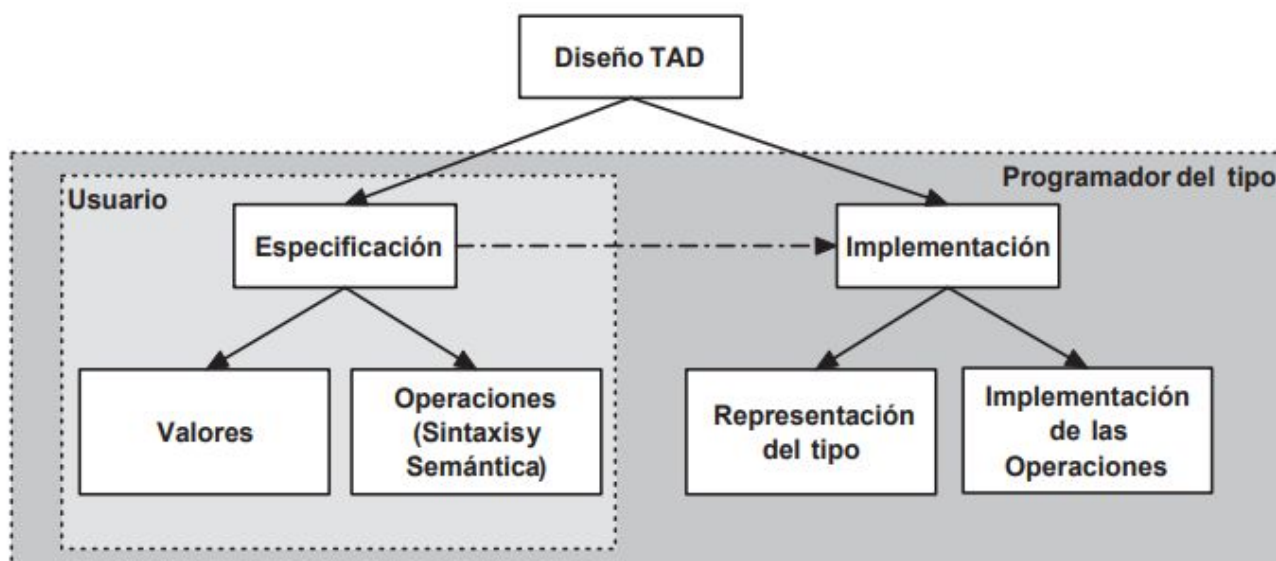


Figura 1: Especificación e implementación de un TAD

### Especificación informal de un TAD

Normalmente a la hora especificar un TAD de manera formal se hace uso de la lógica de primer orden. Sin embargo existe una manera de **especificar los TAD de manera informal**. Para realizar una especificación informal basta con definir mediante un criterio, que puede llegar a incluir el lenguaje natural, los diferentes elementos de un TAD. De esta manera puede definir

**TAD** Nombre del tipo

**VALORES:** valores que pueden tomar los datos del tipo

**OPERACIONES:** nombre de las operaciones que los manipulan

A su vez para cada una de las **OPERACIONES** hay que definir su firma (también llamada signatura). Los elementos que componen la firma de una operación son los siguientes:

**Nombre de operación (tipo de argumentos) → tipo de resultado**

Además de la firma de una operación, existe otra información relativa a la operación que puede detallarse utilizando lenguaje el lenguaje natural.

```
{
    Objetivo: Descripción de la operación
    Entrada: Descripción de los datos de entrada
    Salida: Indica qué es lo que retorna la operación al invocarla
    Precondiciones: Posibles excepciones. Características que tendrán que reunir los datos de entrada para que se realice bien la tarea.
    Poscondiciones: Indica un efecto lateral en la invocación a una función. Afirmaciones que podemos hacer sobre los datos después de que se ejecute la operación y que complete la información del objetivo y las salidas
}
```

A continuación ejemplo de especificación informal utilizando mediante la utilización del lenguaje natural para el TAD números racionales

**NOMBRE TAD:** Racional

**VALORES:** Concepto matemático de números racionales, es decir, un par de números enteros tal que el primero es el numerador y el segundo el denominador.

**OPERACIONES:**

◦ **Crear Racional (n,d:entero) → Racional**

{

**Objetivo:** Crear un número racional

**Entrada:** n: numerador del nuevo racional d: denominador del nuevo racional

**Salida:** El número racional creado

}

◦ **Suma (r1,r2:Racional) → Racional**

{

**Objetivo:** Calcula la suma de dos numero racionales

**Entrada:** r1, r2: números racionales a sumar

**Salida:** Un nuevo racional que contiene la suma de los números a la entrada

}

◦ **Numerador (Racional) → Entero**

{

**Objetivo:** Obtener el numerador de un número racional

**Entrada:** Número racional del que obtener el numerador

**Salida:** numerador del número a la entrada

}

◦ **Denominador (Racional) → Entero**

{

**Objetivo:** Obtener el denominador de un número racional

**Entrada:** Número racional del que obtener el denominador

**Salida:** denominador del número a la entrada

}