



# Tipos de Datos

## Números racionales

[Página inicial](#)[TAD Racional](#)[AbstractRacional](#)[RacionalNoMod](#)[RacionalMod](#)[Documentación](#)[Descargar](#)

## TAD Racional

El tipo abstracto de dato (TAD) `Racional` se proporciona mediante una [interfaz Java](#), especificando cada una de sus operaciones por medio de Javadoc. La documentación se ha generado para el ámbito *package*; es decir, desde el punto de vista del desarrollador.

Por analogía con los tipos numéricos predefinidos en Java, un tipo de dato para números racionales seguramente debería ser inmutable, en cuyo caso, el TAD `Racional` no debería incluir operaciones básicas modificadoras. Sin embargo, se ha preferido incluir una operación de este tipo, *acumula*, para que sea posible implementar tipos con distintas características: *inmutables* y *mutables*.

```
default void acumula (Racional r) {  
    throw new UnsupportedOperationException();  
}
```

*Código 1. Implementación de una operación no soportada*

Para que la misma interfaz se pueda utilizar para realizar implementaciones de tipos de datos mutables y de tipos de datos inmutables, todas las operaciones modificadoras se implementan, por defecto, como operaciones no soportadas, tal como se indica en el cuadro adjunto.

La interfaz incluye las operaciones siguientes:

Operación	Descripción	Coste temporal
<code>default void acumula (int dato)</code>	Modifica este racional sumándole el entero especificado (opcional)	Constante
<code>default void acumula(Racional r)</code>	Modifica este racional sumándole el racional especificado (opcional)	
<code>int denominador()</code>	El denominador de este racional	
<code>int numerador()</code>	El numerador de este racional	
<code>Racional reduce()</code>	Un nuevo racional equivalente a éste e irreducible	
<code>Racional suma(Racional r)</code>	Retorna el racional suma de este racional y el racional especificado	

*Tabla 1. Operaciones de la interfaz*

## Tipo RacionalNoMod

La primera implementación del TAD Racional es un tipo de dato no modificable (o inmutable) dado por la clase `RacionalNoMod`.

Para almacenar la información de los valores racionales del tipo de dato, se utiliza un campo estático, el `array datos`. Los racionales se almacenan en posiciones consecutivas del `array`, por lo que cada componente de éste es otro `array` de dos enteros, uno para el numerador y otro para el denominador de un racional. El número de racionales guardados en el `array datos` se mantiene el campo estático `numRacionales`.

Para representar un racional se utiliza un campo de tipo entero, `rep`. El valor de este campo es el número de componente del `array datos` en el que se almacena la información (numerador y denominador) de éste.

En la Figura 1 adjunta se muestra gráficamente cómo se almacenan los números racionales del tipo `RacionalNoMod` y cómo se representan éstos.

## Tipo RacionalMod

En este caso se va a realizar una implementación del TAD `Racional`, como un tipo de dato modificable (o mutable), el tipo de dato `RacionalMod`. Para simplificar la implementación se extenderá la clase `RacionalNoMod`.

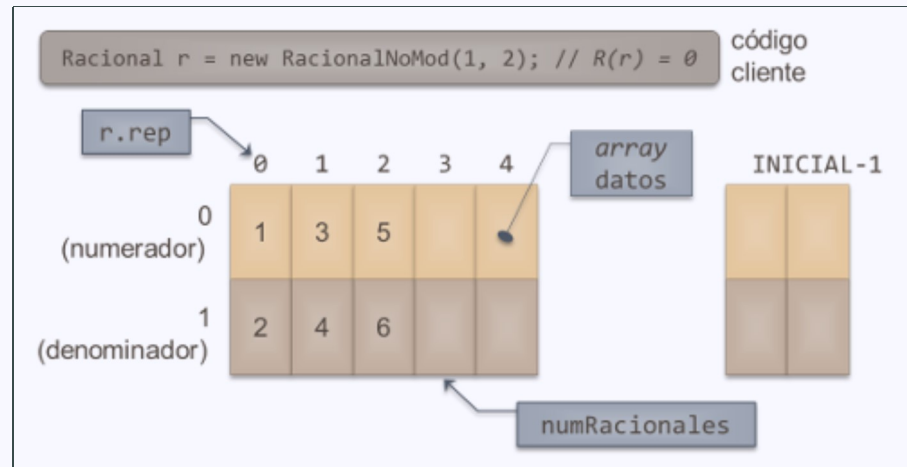


Fig. 1. Almacenamiento para el tipo de dato `RacionalNoMod`