

Ventajas del uso de un Tipo Abstracto de Datos (TAD)

1. Recogen mejor la semántica de los tipos. Al agrupar la representación junto a las operaciones que definen su comportamiento, y forzar a utilizar el TAD a través de estas operaciones se evitan errores en el manejo del tipo de datos (Por ejemplo, la división por 0 en el caso del TAD *Racional*).
2. Abstracción: Separa la especificación (**qué hace**) de la implementación (**cómo lo hace**). Los usuarios de un TAD no necesitan conocer sus detalles de implementación. Como consecuencia:
 - a) Se favorece la extensibilidad del código: Es posible *modificar* y *mejorar* la implementación del TAD sin afectar a los demás módulos que lo utilizan.
 - b) Aumenta la facilidad de uso.
 - c) Aumenta la legibilidad del código que usa el TAD.
3. Produce código reutilizable.
4. Favorece la ausencia de errores, al reutilizar código ya probado y forzar a utilizar la estructura de datos correctamente.