

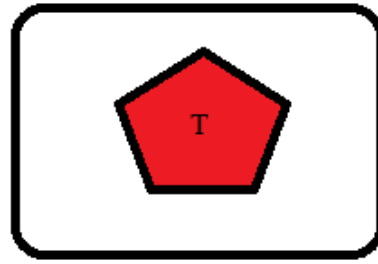
1) Optional<T>

- **Librería:** *Java Collection Framework*
- **Desarrollador:** *Sun Microsystems*
- **Archivo:** *Incluido en el JDK*
- **Paquete:** *java.util*

Esta clase representa un envoltorio que puede estar vacío, o tener dentro un dato de tipo T. O sea, un objeto de esta clase puede tener dos estados posibles:



vacio



guarda un objeto de tipo T

Sus métodos estáticos permiten obtener un Optional<T>, mientras que los métodos de instancia permiten ver si está vacío y recuperar su valor.

Optional<T>
+ static Optional<T> empty() + static Optional<T> of(T objeto) + static Optional<T> ofNullable(T objeto) + boolean isPresent() + T get() + T orElse(T alternativo)

- **empty:** Devuelve un Optional<T> que no tiene nada dentro (está vacío).
- **of:** Devuelve un Optional<T> que guarda el objeto de tipo T pasado como parámetro. Si el objeto que se quiere guardar es null se lanza una NullPointerException.
- **ofNullable:** Es como el método of, pero si el objeto que se quiere guardar es null, devuelve un Optional<T> vacío.
- **isPresent:** Devuelve true si el objeto no está vacío (y guarda un objeto de tipo T)
- **get:** Devuelve el objeto que hay guardado en el Optional, pero si está vacío se lanza una NoSuchElementException.
- **orElse:** Devuelve el objeto que hay guardado en el Optional, y si está vacío se devuelve el objeto alternativo que se pasa como parámetro.

2) OptionalInt

- **Librería:** *Java Collection Framework*
- **Desarrollador:** *Sun Microsystems*
- **Archivo:** *Incluido en el JDK*
- **Paquete:** *java.util*

Esta clase representa un objeto que puede o no, contener un número entero de tipo int. Al igual que tenemos esta clase, también están la clase **OptionalLong** y **OptionalDouble**, que son prácticamente iguales.

OptionalInt
+ static OptionalInt empty() + static OptionalInt of(T objeto) + boolean isPresent() + int getAsInt() +int orElse(int alternativo)

- **empty:** Devuelve un OptionalInt que no tiene nada dentro (está vacío).
- **of:** Devuelve un OptionalInt que guarda el número pasado como parámetro.
- **isPresent:** Devuelve true si el objeto no está vacío (y guarda un número entero)
- **getAsInt:** Devuelve el número que hay guardado en el OptionalInt, pero si está vacío se lanza una NoSuchElementException.
- **orElse:** Devuelve el número que hay guardado en el Optional, y si está vacío se devuelve el número alternativo que se pasa como parámetro.

Las clases **OptionalLong** y **OptionalDouble** funcionan de forma similar a esta clase.