CLASE OPTIONAL

Disponible desde Java 8. Está fuertemente vinculado a la programación funcional.

Es una clase genérica, por lo que puede contener objetos de cualquier clase.

El principal uso de esta clase es reducir los errores NullPointerException, es decir, aquellos errores que surgen cuando invocamos un método de un elemento no inicializado.

En esencia, esta es una clase contenedora que puede contener un valor opcional, que significa que puede contener un objeto o simplemente puede ser vacío. de este modo evitamos crear código preventivo para evitar este tipo de excepciones.

Métodos que dispones esta clase **Optional <T>**

- isPresent: Devuelve true si hay valor.
- get: devuelve valor si lo hay, o NoSuchElementException si no existiera. Es del tipo de dato que indique con antelación.
- ifPresent(Consumer): si hay valor, invoca al conumer que se le pasa como argumento.
- orElse(T other): si hay valor, lo devuelve: sino, devuelve other.
- orElseGet(Supplier): si hay valor, lo devuelve: sino, lo toma del Supplier
- orElseThrow(Supplier): si hay valor, lo devuelve: sino, lanza una excepción

Como obtener un **Optional <T> (Métodos estáticos)**

- Optional.empty(): Devuelve un optional vacío.
- Optional.of(….): Devuelve un optional de un valor determinado.
- Optional.ofNullable(): Devuelve un optional con un valor si es NO nulo; en otro caso devuelve empty().