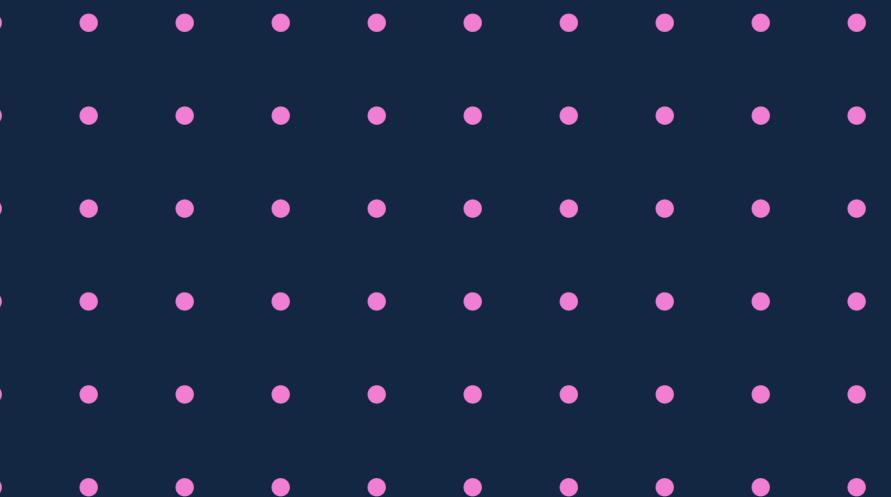




Optional

Por: Gabriel Chaldú

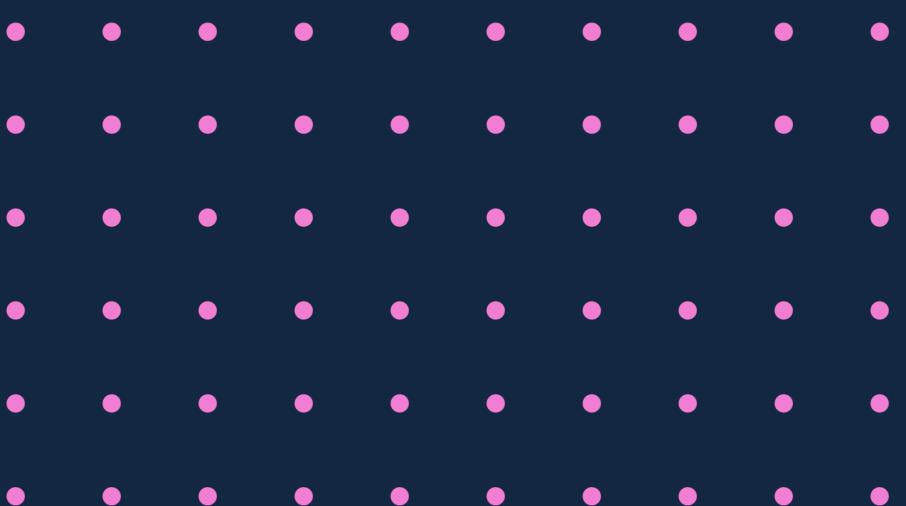


Metáfora del Optional

Una caja ***transparente***, puede tener algo adentro...
o no.

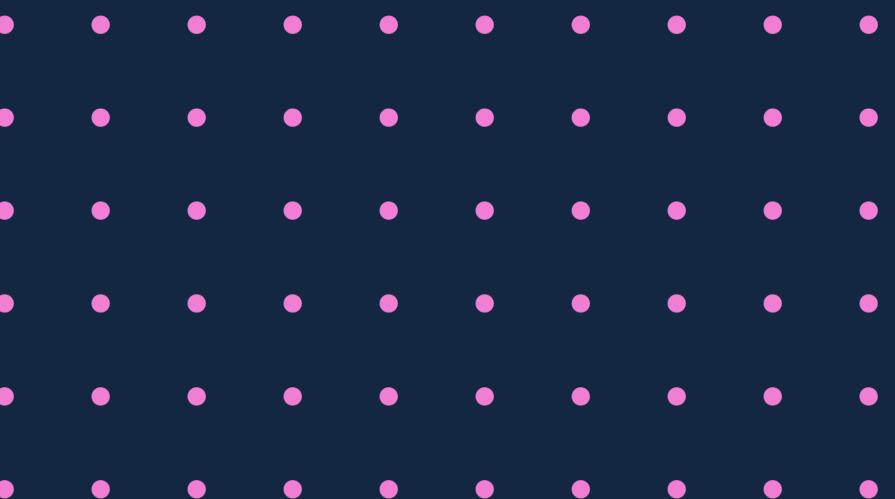


- Si la caja tiene un objeto, lo ves y lo podés usar.
- Si la caja está vacía, lo sabés de antemano.



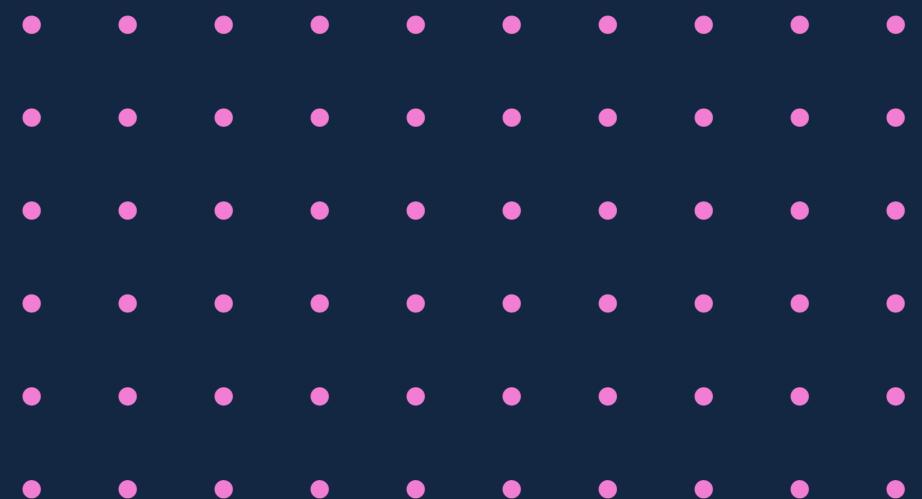
¿Qué es un Optional?

- Clase contenedora que puede o no contener un valor.
- Diseñada para evitar el uso de null.
- En lugar de retornar un null retorna un Optional vacío.



Sin Optional

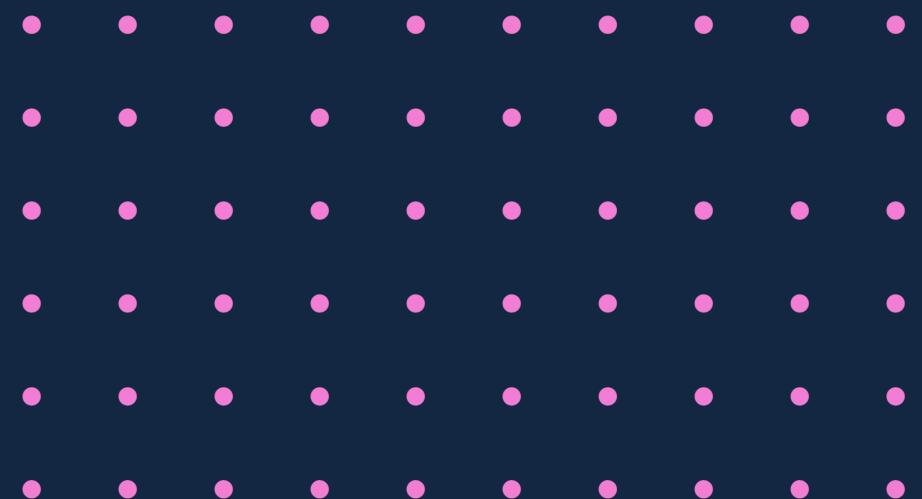
```
● ● ●  
if (user != null) {  
    // usar user  
}
```



Con Optional



```
userOptional.ifPresent(user -> {  
    // usar user  
});
```



Creación de un Optional



```
//Crea un Optional vacío.  
Optional<String> optionalVacio = Optional.empty();  
  
//Crea un Optional con un valor no nulo.  
Optional<String> optionalConValor = Optional.of("Hola Mundo");  
  
//Crea un Optional que puede contener un valor o estar vacío. Se utiliza cuando el valor puede ser null.  
Optional<String> optionalConValor = Optional.ofNullable("Hola Mundo");  
Optional<String> optionalVacio = Optional.ofNullable(null);
```

isPresent & ifPresent



```
//Verifica si el Optional tiene un valor presente.  
Optional<String> optional = Optional.of("Hola");  
if (optional.isPresent()) {  
    System.out.println(optional.get()); // Accede al valor  
}
```

//Ejecuta una acción si el valor está presente. Evita el uso de isPresent() y proporciona una forma más limpia de realizar la acción.

```
Optional<String> optional = Optional.of("Hola");  
optional.ifPresent(System.out::println); // Imprime el valor si está presente
```

get()



```
//Obtiene el valor presente de un Optional. Lanza una NoSuchElementException si  
no hay valor presente.  
Optional<String> optional = Optional.of("Hola");  
String valor = optional.get(); // Devuelve "Hola"
```

orElse()



```
//Devuelve el valor presente o un valor por defecto si el Optional está vacío.  
Optional<String> optional = Optional.ofNullable(null);  
String valor = optional.orElse("Valor por defecto");  
System.out.println(valor); // Imprime "Valor por defecto"
```

orElseThrow()



```
Optional<String> optional = Optional.ofNullable(null);
String valor = optional.orElseThrow(() -> new IllegalArgumentException("Valor no encontrado"));
```

map()



//Transforma el valor presente si está presente, de lo contrario,
devuelve un Optional vacío.

```
Optional<String> optional = Optional.of("Hola Mundo");  
Optional<String> transformado = optional.map(s -> s.toUpperCase());  
System.out.println(transformado.get()); // Imprime "HOLA MUNDO"
```

filter()



```
//Filtra el valor presente en el Optional según una condición. Si la  
//condición no se cumple, devuelve un Optional vacío.  
Optional<String> optional = Optional.of("Hola Mundo");  
Optional<String> filtrado = optional.filter(s -> s.length() > 5);  
System.out.println(filtrado.get()); // Imprime "Hola Mundo"
```



**¡A seguir
mejorando!**

