TIPOS DE LOS LAMBDA

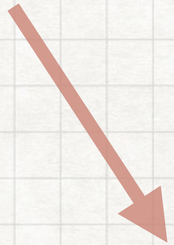
```
Comparator<String> comparadorLongitud =
        (o1, o2) -> o1.length() - o2.length();
```

```
(o1, o2) -> o1.length() - o2.length()  ?
```

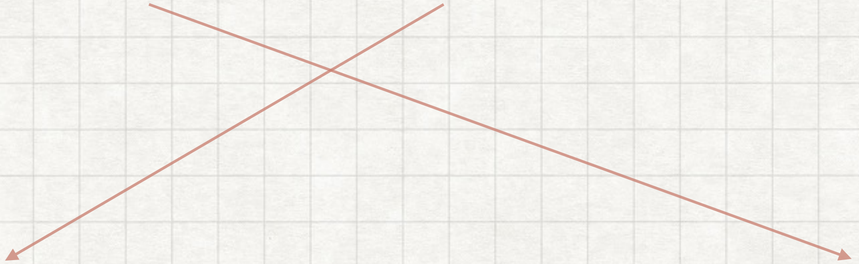
¿Como han encajado los lambdas en Java?

```
(o1, o2) -> o1.length() - o2.length()
```

lambda expression



functional interface instance



es decir: un objeto de un determinado tipo

```
Comparator<String> comparadorLongitud =
        (o1, o2) -> o1.length() - o2.length();
```

lambda expresssion

```
@FunctionalInterface
public interface Comparator<T>
```

implementación del único mètodo abstract

```
@FunctionalInterface
public interface Comparator<T>
```

```
int compare(T o1, T o2);
```

```
(o1, o2) -> o1.length() - o2.length()
```

```
Comparator<String> comparadorLongitud =
        (o1, o2) -> o1.length() - o2.length();
```

```
class XX
   implements Comparator<String>
```