

## Ejercicio Kotlin Imperativo JS Java

### Preguntas: conteste en Kotlin usando lenguaje imperativo

Considere el código en directorios `work/js` y `work/javaKotlin`. En `js` se implementa en `eif400.transpose` un algoritmo usual de transpuesta de una matriz. El objetivo de este ejercicio es mover (reescribir) ese algoritmo a Kotlin imperativo. Se incluye en `js` un módulo `eif400.matrix` con utilitarios de matrices. Estos ya fueron movidos a Java en el directorio `javaKotlin`. Ud. solo debe implementar el método `transpose` en `kotlin/transpose.kt` sin modificar el resto del código.

Corrida y Salida actual en JS (sólo es para su referencia). En `work/js`:

```
PP:node eif400.test.js
>>> Testing Loop <<<
>>> Matrix <<<
[ [ 1, 2, 3 ], [ 4, 5, 6 ] ]
>>> Transposed Matrix <<<
[ [ 1, 4 ], [ 2, 5 ], [ 3, 6 ] ]
PP:
```

Compilación, corrida y Salida sin implementar `transpose.kt` aún. En `work/javaKotlin`:

**NOTA:** Asumimos acá que Kotlin está instalado en `C:\Kotlin` (chequear y ajustar según su máquina si es necesario)

### Compilación

C:\Windows\System32\cmd.exe

```
PP:kotlinc -d classes kotlin\transpose.kt
PP:javac -d classes -cp classes;%classpath% java\*.java
PP:
```

Setear en el `classpath` el runtime de Kotlin

```
PP:set classpath=C:\Kotlin\kotlinc\lib\*;classes;%classpath%
PP:
```

### Correr en Java

```
PP:java -cp %classpath% eif400.test.Main
>>> Testing Loop <<<
>>> Matrix <<<
[[1.0, 2.0, 3.0], [4.0, 5.0, 6.0]]
>>> Transposed Matrix <<<
[[1.0, 2.0, 3.0], [4.0, 5.0, 6.0]]
PP:
```

Note que la salida no es correcta aún. Ud. debe implementarla. Sólo recompila con Kotlin su versión de `transpose.kt` y corra este paso nada más para lograr su solución. Ignore la impresión del punto decimal con relación a JS; eso no es relevante.