

CLEAN CODE

CODE SMELLS EN FUNCIONES

Daniel Blanco Calviño

F1: DEMASIADOS ARGUMENTOS

- Lo mejor es que una función no reciba argumentos, seguido por uno, dos y tres argumentos. **Se deben evitar las funciones con > 3 argumentos.**
- Encapsular argumentos en una clase.

F2: ARGUMENTOS DE SALIDA

```
1  #include <stdio.h>
2
3  void suma (int input_1, int input_2, int* result)
4  {
5      *result = input_1 + input_2;
6  }
7
8  int main() {
9      int result;
10     suma(1, 2, &result);
11     printf("Result: %d\n", result);
12
13     return 0;
14 }
```

/tmp/JJtLkL4SKp.o

Result: 3

E3: PASAR FLAGS COMO ARGUMENTO

Un flag como argumento suele indicar que la función hace más de una cosa

```
private static final Double PREMIUM_DISCOUNT_FACTOR = 0.2;
private static final Double REGULAR_DISCOUNT_FACTOR = 0.1;

public Double calculateDiscount(Item item, boolean isPremium) {
    if(isPremium) {
        return item.getPrice() * PREMIUM_DISCOUNT_FACTOR;
    }

    return item.getPrice() * REGULAR_DISCOUNT_FACTOR;
}
```

E4: FUNCIONES MUERTAS

Las funciones que no se llaman nunca se deben eliminar. Si la necesitamos en un futuro, la podremos encontrar en el software de control de versiones