

## REGLAS DE LA UPM

Todas estas reglas están basadas en que tu código se desarrolla dentro de una función, al igual que ocurrirá en el examen final de Programación 1, voy a separarlos por los temas de los PDF, más que reglas de la UPM, esto serán cosas con las que tener cuidado a la hora del examen:

### Variables

-A la hora de crear variables **NO** pongáis static delante

### Funciones y Procedimientos

-Si es un procedimiento será devolverá un dato del tipo *void* (vacío), si es una función devolverá cualquiera de los otros datos: *int*, *doublé*, *char*, etc.

-Recordad esta estructura →

```
public static TIPO_DE_DATO_A_DEVOLVER NOMBRE_FUNCION (PARÁMETROS)
```

En los parámetros recordad **especificar el tipo de dato**, por ejemplo:

```
mayor (int a,b) : boolean → public static boolean mayor (int a, int b) { //CÓDIGO }
```

### CONDICIONALES

-Los *returns* pueden dentro de cualquier condicional (*if*, *else*, *switch*) o fuera de cualquier estructura.

-El *switch* es el **único** que puede usar *breaks*;

-Si usamos muchos *ifs* y se pueden anidar, anídadlos.

### BUCLES

-Los bucles (*while*, *do while*, *for*) no pueden tener ningún *return* dentro.

-Si queremos usar varias condiciones (por medio de **&&** o **||**) debemos usar un bucle *while* o *do while*, no podemos poner varias condiciones en un bucle *for*, salvo por lo mencionado en el apartado de arrays/matrices de este mismo PDF.

-Poned la menor cantidad de bucles posibles, es mejor poner más condicionales.

### ARRAYS / MATRICES

-Supongamos que tenemos un algoritmo de búsqueda, y queremos ver si se encuentra el número *n* en nuestro array/matriz, acordaos de que, si se encuentra, debemos de salirnos del bucle poniendo una condición extra en el mismo, y es en éste único caso donde podemos poner **2** condiciones en un bucle *for*, más información en el PDF de “arrays” al final.

### STRINGS

-Repasad los métodos que están en el PDF.