

# Memoria de la práctica - Fundamentos de Informática

Borja

12 Agosto 2020

---

En esta memoria paso a comentar algunas breves consideraciones sobre las decisiones que he tomado y por qué he hecho algunas cosas como las he hecho.: Los detalles necesarios sobre cómo funciona el código están bien documentados en `JavaDOC`

---

## Etapa 1

- Las unidades de medida de los distintos tipos de medición las he dejado como `static final`, son un atributo propio de cada clase y creo que es la mejor forma de modelar este hecho sin usar un tipo de dato enumerado como haremos más adelante
- 

## Etapa 2

- Para los instrumentos de medición, he usado los mismos nombres de atributos. Por ejemplo, en los tres instrumentos he usado `valorMaximo` en vez de tener tres atributos con nombres prácticamente idénticos: `valorMaximoTemperatura`, `valorMaximoPrecipitaciones` y `valorMaximoConcentracionN2O`.
  - Además, esto va en línea de lo que se va a hacer más adelante en la etapa 3, factorizando el código para evitar su repetición
  - En la estación meteorológica, se pueden añadir valores a un instrumento de la estación y no añadirlos a la lista de la estación meteorológica. He tomado esta decisión por la forma en la que se usa la clase en el código del `Lanzador`
-

## **Etapas 3**

- Las clases hijas de la super clase `Sensor` apenas tienen código. Lo único que tienen de interesante es el valor distinto de `porcentajeMaximo`. El código del constructor no introduce cambios nuevos
- Esto es gracias a lo dicho para la etapa 2, usar los mismos nombres hace que la refactorización sea mucho más sencilla
- Por tanto, podría ser interesante usar el mismo enfoque que para las mediciones, usar una única clase y a partir de unos atributos adicionales modelar los distintos tipos de sensores (atributo `unidadMedida` y atributo `porcentajeMaximo`, pasados por el constructor)
- Como se dice que la clase `Sensor` es no instanciable, la declaramos como clase abstracta