

## Objetivo:

- Repasar las formas de herencia en programación orientada objetos.
- Coger soltura con la codificación.
- Presentarlo bonito.

## Los niveles:

Como en la clase hay alumnos de diferente experiencia. Se han puesto varios niveles también dentro del ejercicio. Desde el mínimo y que consideraría un fracaso no saber hacer sería el nivel A, a partir de allí se va complicando un poco todo, hasta llegar al nivel D.

Aunque en el tiempo que os dé, sólo lleguéis al nivel A, continuar cuando tengáis tiempo. Es un ejercicio didácticamente muy interesante y me ha costado bastante el planteamiento y los materiales para su realización.

## Entrega y corrección:

El Ejercicio lo pondréis en un zip o rar con vuestro nombre, seguido por el nivel. Por ejemplo: JacintoAisaA.Zip o JacintoAisaD.Rar. El archivo deberéis de dejarlo en esta ruta.

[Ejercicio01 \(Herencia\)](#)

Proporcionada el lunes, 19 de abril de 2023. La entrega tiene límite de tiempo hasta el miércoles que es el último día de esta semana.

## Enunciado:

Obdulio trabaja en una tienda de ordenadores. Pero tiene la inquietud de hacerse un programa para poder calcular el precio de los ordenadores, calcular calores, precios y otras cosas raras que le gustan a Obdulio. A recibido unas sesiones se POO en Java en un curso del SEPE por parte de un maño y quiere ver si ha entendido lo que le han estado explicando.

Es una tienda para ordenadores a medida. No sólo tiene que construir los ordenadores, sino también modificar un ordenador existente, ya sea por avería o por mejora del propio ordenador. Obdulio decide crear un programa orientado a objetos para poder crear computadoras y que estas puedan ser modificadas posteriormente.

- Un **ordenador** lo podemos entender como una colección de componentes.
- Un **componente** tiene un **número de serie** y un **precio**, independientemente de la funcionalidad que ocupe. Dentro de los componentes, tendremos:
- Procesadores.
- Memorizadores.
- Guardadores.

Obdulio decide crear tres interfaces con el fin de poder crear su jerarquía de clases. Ha escuchado a su profesor indicar que es muy flexible esa forma de hacer las cosas, mediante interfaces y composición. Sabe que los componentes se van cambiando muy rápidamente y sabe que va a tener que hacer clases de componentes nuevos, procesadores nuevos, de tipo de memoria y de almacenamiento. Las interfaces que se van a crear tienen nombres como estos.

- IProcesable (Indica que es capaz de procesar) ◦ getCalor() ◦ getCores();
- IMemorizable ◦ getCalor(); ◦ getAlmacenamiento();
- IGuardable, ◦ getCalor(); ◦ getAlmacenamiento();

Para empezar el desarrollo Obdulio tiene los siguientes componentes en su tienda, que paso a poner en un **Anexo I (Componentes Existentes)**.

### Que pedimos:

Nivel A.

- Construir varios Ordenadores, relatados en el **Anexo III (Orden De Trabajo A)**
- Calcular cual es el precio para cada ordenador y que cantidad de calor genera.

Nivel B.

- Uppps: Se han creado dos nuevos componentes, se reflejan en el **Anexo II (Nuevos componentes)**.
- Modificar el Nivel A y añadir un ordenador que tenga la capacidad de tener un sistema de almacenamiento primario y una colección de dispositivos de almacenamiento.
- Construir dos ordenadores de este tipo, relatados en el **Anexo IV (Orden de Trabajo B)**
- Calcular cual es el precio para cada ordenador y que cantidad de calor genera.

Nivel C.

- Crear una nueva clase denominada Pedido que será una colección de Ordenadores.
- Crearemos un objeto denominado PedidoA, reflejando los ordenadores del Anexo III (Orden de Trabajo A).
- Crearemos un objeto denominado PedidoB, reflejando los ordenadores del Anexo IV (Orden de Trabajo B)
- Calcular cual es el precio para el PedidoB y que cantidad de calor genera.

Nivel D:

- Crear una clase denominada Almacén, que será una colección de Componentes , para poder hacer este ejercicio tenemos que añadir la propiedad Cantidad en la clase Componente. Adjunto en el **Anexo V (Cantidades en Stock)**
- Crear una clase denominada Factura que será una colección de Pedidos.
- Crear un objeto denominado FacturaA que contendrá el PedidoA,
- Crear un objeto denominado FacturaB que contendrá el Pedido B,
- Crear un objeto denominado FacturaC que contendrá el PedidoA y el Pedido B.
- Calcular el total de la facturación y si con las cantidades que disponemos es posible servirlo.