



Alumno/a:

Programación III - 2º Parcial

Un cliente, requiere un sistema modelado e implementado observando los principios de la POO para registrar y administrar componentes informáticos en un taller de reparación de equipos de computadora. Los datos pertinentes a dicho objeto *Componente* son:

- *Nº de Serie*: Número entero del intervalo [0 a 999.999.999.999]. Ejemplo: 435.733.615.993.
- *Detalle*: Cadena de caracteres descriptiva del componente. Ejemplo: "Placa de Video NVidia 6800".
- *Costo de Componente*: Número real (punto flotante de precisión simple) que representa el importe a cobrar por el componente de reemplazo. Ejemplo: \$765,50.
- *Costo de Mano de Obra*: Número real (punto flotante de precisión simple) que representa el importe a cobrar en concepto de mano de obra en el reemplazo del componente. Ejemplo: \$150,0.

Se pide:

A) Código fuente del archivo de clase *CComponente.cs* conteniendo:

- 1) Adecuada declaración de las variables miembro *numSerie*, *detalle*, *costoC* y *costoMO*.
- 2) Un método constructor por defecto, que asigne "A Definir" a la variable miembro *detalle*.
- 3) Los métodos setters y getters correspondientes a todas las variables miembro.
- 4) Un método *darPrecio()*, que sin argumento alguno, devuelva como número real (punto flotante de precisión simple), el *Precio* a abonar por el servicio de reparación, producto de sumarle al *Costo del Componente*, el *Costo de Mano de Obra*.
- 5) Un método *darDatos()*, que sin argumento alguno, devuelva una cadena de caracteres que concatene los valores presentes en las variables miembro *numSerie*, *detalle*, *costoC*, *CostoMO*, más el valor calculado de *Precio* a abonar por el componente (emplear método anterior).
- 6) Debe implementar la interfaz *IEquatable* para poder buscar por número de serie.

B)

C) Se pide código fuente del archivo de clase *CPlacaDeVideo* que debe tener los siguientes atributos extras:

- 1) RAM dedicada: Entero sin signo.
- 2) Frecuencia: Flotante sin signo.
- 3) MarcaPlaca: String que debe validarse al setearse que sea ATI o Nvidia
- 4) Se debe sobrescribir el método *darDatos()* para tener en cuenta estos datos.

D) Se pide código fuente del archivo de clase *CMicroProcesador* que debe tener los siguientes atributos extra:

- 1) FrecuenciaReloj: Flotante sin signo.
- 2) CantidadDeNucleos: Entero sin signo.
- 3) MarcaProcesador: String que debe validarse al setearse que sea Intel O AMD.

E) Se pide el código fuente de una clase Interfaz que debe ser estática y debe encargarse del ingreso y el egreso de datos.

F) Se pide el código fuente de una clase controladora que tenga como atributo una lista de Componentes y maneje la lógica del programa (buscar en la lista, agregar placa a la lista, agregar CPU a la lista, quitar de la lista, modificar un micro, modificar una placa de video, mostrar datos de la lista, cargar de archivo y guardar en archivo).

G) Se pide una clase principal que se encargue de mostrar la siguiente lista y realizar las acciones pertinentes para cada opción:

Aplicacion Componentes

Menu:

- 1 - Ingresar una placa de video
- 2 - Ingresar una CPU
- 3 - Mostar datos de un componente
- 4 - Mostrar datos de todos los componentes
- 5 - Eliminar un componente
- 6 - Editar un componente
- 7 - Cargar placas de video desde archivo
- 8 - Cargar CPUs desde archivo
- 0 - Salir

Su opcion: