

## 1. Clase Hora

Crea una clase Hora con atributos para las horas, los minutos y los segundos de la hora Incluye, al menos, los siguientes métodos:

Constructor predeterminado con el 00:00:00 como hora por defecto. En el constructor se podrán indicar horas, minutos y segundos.

leer(): pedirá al usuario las horas, los minutos y los segundos.

valida(): comprobará si la hora es correcta; si no lo es la ajustará. Será un método auxiliar (privado) que se llamará en el constructor parametrizado y en leer().

a\_segundos(): devolverá el número de segundos transcurridos desde la medianoche.

de\_segundos(int): hará que la hora sea la correspondiente a haber transcurrido desde la medianoche los segundos que se indiquen.

segundos\_desde(Hora): devolverá el número de segundos entre la hora y la proporcionada.

siguiente(): pasará al segundo siguiente.

anterior(): pasará al segundo anterior.

copia(): devolverá un clon de la hora.

Además (métodos especiales):

print: mostrará la hora (07:03:21).

igual\_que(Hora): indica si la hora es la misma que la proporcionada.

menor\_que(Hora): indica si la hora es anterior a la proporcionada.

mayor\_que(Hora): indica si la hora es posterior a la proporcionada.

Crea los tests correspondientes para demostrar que el programa funciona bien.

---

## 2. Clase Canción

Desarrolla una clase Canción con los siguientes atributos:

\_titulo: una variable String que guarda el título de la canción.

\_autor: una variable String que guarda el autor de la canción.

\_duracion: tiempo en segundos de la canción.

y los siguientes métodos:

El constructor que recibe como parámetros el título y el autor de la canción (por este orden)

dame\_titulo(): devuelve el título de la canción.

dame\_autor(): devuelve el autor de la canción.

pon\_titulo(String): establece el título de la canción.

pon\_autor(String): establece el autor de la canción.

- 
3. Haz una clase llamada **Persona** que siga las siguientes condiciones: Sus atributos son: **nombre, edad, DNI, sexo** (H hombre, M mujer), **peso y altura**. No queremos que se accedan directamente a ellos. Piensa que modificador de acceso es el más adecuado, también su tipo. Si quieres añadir algún atributo puedes hacerlo. Por defecto, todos los atributos menos el DNI serán valores por defecto según su tipo (0 números, cadena vacía para String, etc.). Sexo será hombre por defecto.

Se implantaran varios constructores:

- Un constructor por defecto.
- Un constructor con el nombre, edad y sexo, el resto por defecto.
- Un constructor con todos los atributos como parámetro.

Los métodos que se implementaran son:

- **calcularIMC()**: calculara si la persona esta en su peso ideal (peso en  $\text{kg}/(\text{altura}^2 \text{ en m})$ ), devuelve un -1 si esta por debajo de su peso ideal, un 0 si esta en su peso ideal y un 1 si tiene sobrepeso .Te recomiendo que uses constantes para devolver estos valores.
- **esMayorDeEdad()**: indica si es mayor de edad, devuelve un booleano.
- **comprobarSexo(char sexo)**: comprueba que el sexo introducido es correcto. Si no es correcto, sera H. No será visible al exterior.
- **toString()**: devuelve toda la información del objeto.
- **generaDNI()**: genera un numero aleatorio de 8 cifras, genera a partir de este su número su letra correspondiente. Este método será invocado cuando se construya el objeto. No será visible al exterior.
- Métodos set de cada parámetro, excepto de DNI.

Ahora, crea una clase ejecutable que haga lo siguiente:

Pide por teclado el nombre, la edad, sexo, peso y altura.

Crea 3 objetos de la clase anterior, el primer objeto obtendrá las anteriores variables pedidas por teclado, el segundo objeto obtendrá todos los anteriores menos el peso y la altura y el último por defecto, para este último utiliza los métodos set para darle a los atributos un valor.

Para cada objeto, deberá comprobar si esta en su peso ideal, tiene sobrepeso o por debajo de su peso ideal con un mensaje.

Indicar para cada objeto si es mayor de edad.

Por último, mostrar la información de cada objeto.

- 
4. Haz una clase llamada **Password** que siga las siguientes condiciones: Que tenga los atributos **longitud** y **contraseña** . Por defecto, la longitud será de 8.

Los constructores serán los siguiente:

- Un constructor por defecto.
- Un constructor con la longitud que nosotros le pasemos. Generara una contraseña aleatoria con esa longitud.

Los métodos que implementa serán:

- **esFuerte()**: devuelve un booleano si es fuerte o no, para que sea fuerte debe tener mas de 2 mayúsculas, mas de 1 minúscula y mas de 5 números.
- **generarPassword()**: genera la contraseña del objeto con la longitud que tenga.
- Método get para contraseña y longitud.
- Método set para longitud.

Ahora, crea una clase ejecutable:

Crea un array de Passwords con el tamaño que tu le indiques por teclado.

Crea un bucle que cree un objeto para cada posición del array.

Indica también por teclado la longitud de los Passwords (antes de bucle).

Crea otro array de booleanos donde se almacene si el password del array de Password es o no fuerte (usa el bucle anterior).

Al final, muestra la contraseña y si es o no fuerte (usa el bucle anterior). Usa este simple formato:

contraseña1 valor\_booleano1

contraseña2 valor\_bololeano2

---

5. Crearemos una supeclase llamada **Electrodomestico** con las siguientes características:

Sus atributos son **precio base**, **color**, **consumo energético** (letras entre A y F) y **peso**.

Por defecto, el color sera blanco, el consumo energético sera F, el precioBase es de 100 € y el peso de 5 kg. Usa constantes para ello.

Los colores disponibles son blanco, negro, rojo, azul y gris. No importa si el nombre esta en mayúsculas o en minúsculas.

Los constructores que se implementaran serán

- Un constructor por defecto.
- Un constructor con el precio y peso. El resto por defecto.
- Un constructor con todos los atributos.

Los métodos que implementara serán:

- Métodos get de todos los atributos.
- **comprobarConsumoEnergetico(char letra)**: comprueba que la letra es correcta, sino es correcta usara la letra por defecto. Se invocara al crear el objeto y no será visible.
- **comprobarColor(String color)**: comprueba que el color es correcto, sino lo es usa el color por defecto. Se invocara al crear el objeto y no será visible.
- **precioFinal()**: según el consumo energético, aumentara su precio, y según su tamaño, también. Esta es la lista de precios:

Letra	Precio
A	100 €
B	80 €
C	60 €

Letra	Precio
D	50 €
E	30 €
F	10 €

  

Tamaño	Precio
Entre 0 y 19 kg	10 €
Entre 20 y 49 kg	50 €
Entre 50 y 79 kg	80 €
Mayor que 80 kg	100 €

Crearemos una subclase llamada **Lavadora** con las siguientes características:

Su atributo es **carga**, además de los atributos heredados.

Por defecto, la carga es de 5 kg. Usa una constante para ello.

Los constructores que se implementaran serán:

- Un constructor por defecto.
- Un constructor con el precio y peso. El resto por defecto.
- Un constructor con la carga y el resto de atributos heredados. Recuerda que debes llamar al constructor de la clase padre.

Los métodos que se implementarán serán:

- Método get de carga.
- **precioFinal()**: Si tiene una carga mayor de 30 kg, aumentara el precio 50 €, sino es así no se incrementara el precio. Llama al método padre y añade el código necesario. Recuerda que las condiciones que hemos visto en la clase Electrodomestico también deben afectar al precio.

Crearemos una subclase llamada **Televisión** con las siguientes características:

Sus atributos son **resolución** (en pulgadas) y **sintonizador TDT** (booleano), además de los atributos heredados.

Por defecto, la resolución será de 20 pulgadas y el sintonizador será false.

Los constructores que se implementaran serán:

- Un constructor por defecto.
- Un constructor con el precio y peso. El resto por defecto.

- Un constructor con la resolución, sintonizador TDT y el resto de atributos heredados. Recuerda que debes llamar al constructor de la clase padre.

Los métodos que se implementara serán:

- Método get de resolución y sintonizador TDT.
- **precioFinal()**: si tiene una resolución mayor de 40 pulgadas, se incrementara el precio un 30% y si tiene un sintonizador TDT incorporado, aumentara 50 €. Recuerda que las condiciones que hemos visto en la clase Electrodomestico también deben afectar al precio.

Ahora crea una clase ejecutable que realice lo siguiente:

Crea un array de Electrodomésticos de 10 posiciones.

Asigna a cada posición un objeto de las clases anteriores con los valores que desees.

Ahora, recorre este array y ejecuta el método precioFinal().

Deberás mostrar el precio de cada clase, es decir, el precio de todas las televisiones por un lado, el de las lavadoras por otro y la suma de los Electrodomésticos.

Por ejemplo, si tenemos un Electrodoméstico con un precio final de 300, una lavadora de 200 y una televisión de 500, el resultado final será de 1000 (300+200+500) para electrodomésticos, 200 para lavadora y 500 para televisión.

6. Crearemos una clase llamada **Serie** con las siguientes características: Sus atributos son **titulo, numero de temporadas, entregado, genero y creador**.

Por defecto, el numero de temporadas es de 3 temporadas y entregado **false**. El resto de atributos serán valores por defecto según el tipo del atributo.

Los constructores que se implementaran serán:

- Un constructor por defecto.
- Un constructor con el titulo y creador. El resto por defecto.
- Un constructor con todos los atributos, excepto de entregado.

Los métodos que se implementara serán:

- Métodos get de todos los atributos, excepto de entregado.
- Métodos set de todos los atributos, excepto de entregado.

Crearemos una clase **Videojuego** con las siguientes características:

Sus atributos son **titulo, horas estimadas, entregado, genero y compañía**.

Por defecto, las horas estimadas serán de 10 horas y entregado false. El resto de atributos serán valores por defecto según el tipo del atributo.

Los constructores que se implementaran serán:

- Un constructor por defecto.
- Un constructor con el titulo y horas estimadas. El resto por defecto.
- Un constructor con todos los atributos, excepto de entregado.

Los métodos que se implementara serán:

- Métodos get de todos los atributos, excepto de entregado.
- Métodos set de todos los atributos, excepto de entregado.
- **entregar()**: cambia el atributo prestado a true.
- **devolver()**: cambia el atributo prestado a false.
- **isEntregado()**: devuelve el estado del atributo prestado.

Ahora crea una aplicación ejecutable y realiza lo siguiente:

Crea dos arrays, uno de **Series** y otro de **Videojuegos**, de 5 posiciones cada uno.

Crea un objeto en cada posición del array, con los valores que desees, puedes usar distintos constructores.

Entrega algunos **Videojuegos** y **Series** con el método **entregar()**.

Cuenta cuantos **Series** y **Videojuegos** hay entregados. Al contarlos, devuélvelos.

Por último, indica el **Videojuego** tiene más horas estimadas y la serie con mas temporadas. Muéstralos en pantalla con toda su información.

- 
7. El sistema es para una microempresa que fabrica y vende ropa, esta empresa requiere tener una relación completa de los productos que tiene en el almacén, y que cuando se realice una venta además de que se imprima una nota y se pueda registrar el efectivo recibido para que el sistema entregue cuanto cambio se debe dar, las existencias del almacén se actualicen de acuerdo a la venta realizada, por otro lado necesita que al terminar un día, una semana y un mes, el sistema entregue un reporte de las ventas realizadas así como el dinero obtenido, también requiere que cuando queden menos de 8 productos de un mismo tipo el sistema de un aviso de que el

producto se está terminando. Además de que se puedan agregar y dar de baja productos, también se debe tener la opción de ingresar las inversiones que se han hecho para que se puedan visualizar las ganancias obtenidas.

- 
8. Crear una clase Persona, con atributos nombre, apellido y teléfono. Controlar que el teléfono solo acepte 9 dígitos. Crear una clase Cuenta, con atributos numeroCuenta, saldo y propietario. El propietario es un objeto tipo Persona. Crear un constructor con parámetros y otro sin parámetros, métodos accesorios y toString para estas clases. Controlar que el saldo de la cuenta no debe ser menor que 0. Crear un método llamado transacción que ingrese como parámetros cantidad y tipoTransaccion; tipoTransaccion es “retiro” o “deposito”. Si es un retiro, la cantidad se resta del saldo, y si es deposito la cantidad se incrementa al saldo. El método transacción debe imprimir el tipo de transacción y el nuevo saldo. Crear en una clase llamada Principal, dos cuentas pertenecientes a dos personas distintas y hacer un deposito y un retiro en cada cuenta. Imprimir los valores de las personas, propietarios y transacciones.

- 
9. Una empresa grande paga a sus vendedores mediante comisiones. Los vendedores reciben \$200 por semana, mas el 9% de sus ventas brutas en esa semana. Por ejemplo, un vendedor que vende \$5000 de mercancías en 1 semana recibe \$200 mas el 9% de \$5000, es decir recibe un total de \$650.

Hacer una aplicación que reciba como entrada los artículos vendidos por el vendedor durante la ultima semana, y que calcule y muestre los ingresos de ese vendedor. No hay limite en cuanto al numero de artículos que un vendedor puede vender.

- 
10. Se desea desarrollar un sistema para una concesionaria de vehículos. Los vehículos se clasifican en: autos, camionetas y motocicletas. Todos los vehículos tienen un código, una marca, tipo (auto, camioneta, etc.), un modelo (año), una patente y un kilometraje. Además, la concesionaria lleva un registro de las ventas realizadas, cada registro consiste de la siguiente información: monto de la venta, vehículo vendido (código), apellido, nombre y ci del comprador.

- 
11. Realizar un sistema de Control de Alquiler de Vehículos:

El sistema debe tener un menú:

Ingreso de Vehiculos

Placa del Vehiculo

Nombre de la Persona que Alquila



Km actual del vehículo (INTEGER)  
Buscar los vehículos cuyo KM sea mayor a 20 y menor a 100  
Listar todo los vehículos Ingresados

NOTA: Trabajar con un ingreso de 10 Vehículos. Los reportes de  
Buscar y  
Listar deben ser en columnas, Ej.:  
Placa KM Persona  
GHJ456 12  
Julio Pérez

---

12. Escribe un programa en que realice lo siguiente:  
Que muestre el siguiente menú en pantalla:

Menú principal  
1.-Escribir el archivo de datos  
2.-Leer y procesar los datos  
3.- Terminar

La opción 1 debe pedir la siguiente información por el teclado de n  
alumnos, nombre, sexo, edad, estatura y peso y escribirla en el  
archivo.

La opción 2 debe abrir el archivo para leer los datos y calcular el  
promedio de edad de las personas cuyo sexo sea masculino, así como  
imprimir el total de las personas (hombres y mujeres) que miden mas  
de 1.65 metros.

Nota: Se debe de usar un método para mostrar el menú, un metodo  
para la opción 1 y un método para la opción 2.  
Así como otros métodos que sean necesarios e instanciar las clases  
para ejecutar estos métodos.

- 
13. La tienda doña chachi desea que usted le ayude con un programa  
para automatizar sus ventas. Para esto se piensa crear un archivo  
que contenga los siguientes datos: Fecha, Nombre del producto,  
cantidad, p/u. Este archivo debe ser llenado cada vez que se hace  
una venta en la tienda, con cada uno de los datos especificados. La  
forma de llenado es la siguiente: cada dato está separado por una “\_”.  
Ejemplo: 26/06/2006\_Coca cola de 2 Lts.\_3\_6 .

Donde la fecha es: 26/06/2006, El producto es: Coca Cola de 2 Lts, La  
cantidad es: 3, El precio unitario es: 6.

Cada venta debe ser ingresada en una línea del archivo exactamente de la forma especificada. Por otro lado se quiere que dada una fecha el programa me muestre todas las ventas de esa fecha en una lista, además del monto total que debería haber ingresado a la tienda por ventas ese día. (Para esto deberán encontrar el total de ventas de cada línea que es:  $(p/u * cantidad)$ ). También se debe poder hacer búsquedas por nombre del producto y mostrar una lista de todas las ventas realizadas con ese nombre.

- 
14. Se desea manejar los antecedentes de un curso. Cada curso cuenta con un nombre y con una nómina de alumnos. Por su parte, cada alumno cuenta con un ID, un nombre y un promedio final. Algunos de los métodos asociados al curso son los que se indican a continuación:

<b>MÉTODO</b>	<b>DESCRIPCIÓN</b>
matricular(Alumno)	Permite AGREGAR al alumno 'a' al curso.
remover(Alumno)	Permite REMOVE al alumno 'a' del curso.
estaInscrito(Alumno)	Permite saber si el alumno 'a' se encuentra presente en el curso.

En el curso no se pueden tener alumnos repetidos, es decir, NO SERÍA POSIBLE tratar de matricular a un mismo alumno dos veces.

Se pide implementar las clases que permitan almacenar los datos de un curso y sus alumnos.

Se pide implementa el método main que permita cumplir con la siguiente funcionalidad:

Crear a tres alumnos con los siguientes datos:

- > ID = 1; NOMBRE = "ALFA"; PROMEDIO = 6.5
- > ID = 2; NOMBRE = "BETA"; PROMEDIO = 4.5
- > ID = 3; NOMBRE = "GAMMA"; PROMEDIO = 5.5

Crear el curso llamado PEF2501

Imprimir la nómina de alumnos del curso

Matricular a ALFA y BETA en el curso PEF2501

Imprimir en la salida estándar, utilizando el método estaInscrito(), si el alumno llamado BETA se encuentra inscrito en el curso.

Imprimir en la salida estándar, utilizando el método estaInscrito(), si el alumno llamado GAMMA se encuentra inscrito en el curso.

Remover del curso al alumno llamado BETA

Imprimir la nómina del curso

---

15. Este ejercicio trata de modelar un banco.

El banco consiste de una lista de cuentas y una lista de clientes. De los clientes tenemos su DNI, nombre, apellidos, dirección y teléfono. Sobre un banco podemos efectuar una revisión de todas las cuentas del mismo. En cada cuenta podemos ingresar dinero, sacar dinero, consultar saldo, cambiar el cliente, efectuar una revisión mensual.

La revisión mensual de la cuenta hace lo siguiente:

$\text{Saldo} = \text{saldo} + \text{interes\_cuenta} - \text{comisión}$

Todas las cuentas tienen una comisión de 0,6€ mensuales que puede ser modificada.

Hay tres tipos de cuenta: CuentaCorriente (CC), CuentaVivienda (CV) y FondoInversion (FI).

La CC tiene un interés del 0.1, la CV del 0.2 y el FI de un 0.34. De la CC no se puede sacar mas dinero del que hay, del FI se pueden sacar hasta 500€ de números rojos (si se supera la cuenta se bloquea) y de la CV no se puede sacar dinero.

La revisión mensual de la CV no paga comisión bancaria. La cuenta de puntos, cada vez que se ingresan 6€ se suma un punto. Se pueden ver cuantos puntos lleva la cuenta.

Hay que implementar el modelo del banco, y crear una clase de prueba que compruebe la funcionalidad de las clases.

En este apartado, no es necesaria la creación de interfaz textual o gráfica.

1-Diseñar el modelo de clases y objetos que se va a usar, con sus atributos y operaciones.

2-Implementar cada una de las clases.

3-Crear un método main, que genere varios objetos, y pruebe el comportamiento de las clases mostrando el resultado por pantalla, en la misma consola.

---

16. Una empresa familiar de zapatos, fabrica 3 modelos de zapatos para caballero con los siguientes precios de venta:

1.-Modelo Ejecutivo.....345.50

2.-Modelo Premier.....298.70

### 3.-Modelo Emperador.....246.00

Los 3 modelos son fabricados únicamente en talla 8. La empresa planea fabricar los tres modelos también en tallas 9 y 10, y desea un programa que calcule los precios de venta para estas nuevas tallas. Se sabe que por cada número que se aumente en la talla del zapato, su precio de venta deberá incrementarse 10 pesos, esto significa que si el modelo Emperador cuesta 246.00 pesos en medida número 8, su costo en número 9 será 256.00 y en 10 es 266.00. Escribe un programa que pida el modelo del zapato (a, b ó c) y la medida (8, 9 ó 10) e imprima en pantalla su precio de venta correspondiente.

---

17. Un importante local comercial de la provincia solicita la confección de un sistema para el seguimiento de sus ventas. Para ello solicita la confección de un programa que realice lo siguiente:

1. Ingrese los datos de los productos a la venta que dispone dicho local. De cada producto se conoce: código (un valor numérico), descripción y precio unitario.
2. Ingrese los datos de las ventas registradas. De cada venta se conoce: Fecha, Nro de Vendedor (un valor entre 0 y 9), código del producto vendido, cantidad vendida y forma de pago (0 - Efectivo, 1 - Débito, 2 - Tarjeta)
3. Determinar cual fue el vendedor que realizó la mayor cantidad de ventas.
4. Generar un listado, ordenado por número de vendedor, de todas las ventas realizadas por débito. El listado debe incluir el precio final de cada venta (precio unitario x cantidad)
5. Determinar si el vendedor x realizó una venta del producto y. En caso de que dicha venta exista mostrar todos sus datos (incluido el precio final de venta), en caso que no exista, informar con un mensaje.
6. Determinar el monto total de ventas realizadas.
7. Mostrar los datos de la venta de mayor importe abonada con tarjeta de crédito.

---

18. Implementar la clase Asignatura que represente el nombre de una asignatura y la nota correspondiente obtenida. Las operaciones son:

- > Constructor que acepte como parámetros el nombre de la asignatura y la nota obtenida.
- > Métodos para modificar la nota (setNota) y para consultar la nota (getNota).
- > Método que nos devuelva "Aprobado" si la nota es mayor o igual a 60 o "Reprobado" si la nota es menor que 60.
- > Método para consultar el nombre de la asignatura.

Implementar la clase Alumno que incluya una colección de Asignaturas a las que el alumno ha asistido. Además de incluir los atributos nombre y edad. Las operaciones disponibles sobre el alumno son:

- Constructor que acepte como parámetro el nombre del alumno y edad.
- Métodos para modificar el nombre (setNombre) y para consultarlo (getNombre).
- Métodos para modificar y consultar la edad.
- Método que nos devuelva el promedio del alumno.
- Método para agregar una Asignatura a su plan de estudio; verificar que la asignatura no exista previamente en el arreglo de este Alumno.

Implementar la clase Aplicación para hacer uso de las clases Alumno y Asignatura.

1.Crear 3 alumnos (Tres instancias de la clase Alumno) con sus respectivos nombre y edad.

2.Para cada alumno establecer sus asignaturas y la nota obtenida.

3.Imprimir en pantalla:

1.Nombre del alumno.

2.Edad.

3.Asignaturas que cursó:

Nombre de la asignatura.

Nota obtenida.

Si es una asignatura aprobada o no.

4.Promedio del alumno

---

#### 19. Crear una aplicación para gestionar un videoclub.

El videoclub cuenta con varios tipos de producto

-Todos los productos tienen: Referencia (Titulo, tipo(película/videojuego), precio alquiler, plazo alquiler (días), alquilado (si/no)

-Película: Genero (acción, fantástica, drama, aventuras, puzzle, infantil), año, director, interpretes.

-Videojuego: Estilo (acción, deportes, aventuras, puzzle, infantil), plataforma (Xbox, playstation, wii).

Se mantiene un listado de clientes: N° cliente, nombre, dirección, teléfono, productos alquilados

Se guarda un listado de registros de alquiler: Cliente, producto, fecha alquiler, fecha devolución, importe.

Crear una aplicación de consola con el siguiente menú:

-Lista productos

-Añadir producto

- Ficha producto
- Lista clientes
- Añadir cliente
- Ficha cliente
- Alquiler producto

---

20. Crear una clase Equipo que contenga los atributos Partidos\_Ganados, Puntos\_a\_Favor y Puntos\_En\_Contra, crear 5 equipos y darle valores a sus atributos, ordenar por partidos ganados de mayor a menor y en el caso de que hubiese equipos con los mismos partidos ganados se mirara el criterio de Puntos\_a\_favor para saber cual esta primero y si en los puntos a favor también están empatados se miraran los puntos en contra.

---

21. Un Empresario ha realizado una reunión para recortar gastos en la empresa, por lo tanto ha tomado la decisión de despedir los empleados con menor tasa de éxito, a los empleados se les tomara en cuenta 3 factores. Ganancia Generadas, Horas de Trabajo Diario y Sueldo, en base a esto el Empresario a decidido que despedirá a los empleados que sean menos efectivos que otro. El Empresario medirá esto de la siguiente manera: Si un empleado A Genera mas ganancia, trabaja mas tiempo diario y tiene un menor sueldo que un empleado B, el empleado B será despedido.

- >La cantidad de empleados será entre 1 y 1,000
- >Las Ganancias no sobrepasaran 1,000,000.
- >Las horas diarias trabajadas no sobrepasaran las 24 horas.
- >El sueldo no será menor o igual a 0 ni mayor a 1,000,000.
- >Solo se permitirán números enteros.

=>Valores de entrada: cantidad de empleados, luego cada valor (Ganancia, Horas de Trabajo, Sueldo).

=>Valor de Salida, la cantidad de empleados despedidos.

Ejemplo:

Cantidad de Empleados: 5

Empleado 1: 500 12 300

Empleado 2: 400 14 400

Empleado 3: 100 10 500 #Este será Despedido.

Empleado 4: 450 11 1000000

Empleado 5: 50 1 10000 #Este también será Despedido.

Cantidad de Empleados Despedidos: 2