# RETO 4 – PROGRAMACIÓN BÁSICA

## CONTEXTO

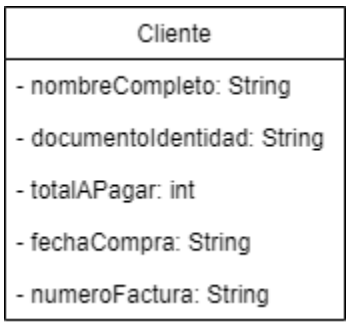
Usted ha sido contratado por la tienda de su barrio para hacer una solución que le facilitará llevar al tendero estadísticas de una compra realizada por un cliente.

Su solución le permitirá conocer al tendero la siguiente información:

- El promedio del total pagado por todos los clientes.
- El nombre del cliente que menos pagó.
- El total pagado más bajo.
- El nombre del cliente que más pagó.
- El total pagado más alto.

## TAREAS

Realizar un método en Java (Llamado reportes) que le permita al tendero conocer unas estadísticas básicas sobre sus clientes, teniendo en cuenta las siguientes especificaciones:
Los clientes que frecuentan la tienda de barrio que lo contrató se van a representar como un `ArrayList` de clientes, donde cada elemento es un objeto de la clase `Cliente` de la cual el equipo de ingeniería de software le hace entrega del siguiente diagrama de clases:




Recuerde que los métodos relacionados al constructor, getters y setters son obviados en el diagrama de clases, pero deberán ser incluidos en el código;

estos métodos deben ser creados con el estándar camel case, por ejemplo, si el atributo se llama `nombreCompleto`, sus métodos correspondientes a get y set serían `getNombreCompleto` y `setNombreCompleto`.

A partir de este `ArrayList`, usted deberá calcular el promedio del total pagado por cada cliente (La suma de todos los totales a pagar dividida por la cantidad de clientes), calcular cuál es el total pagado más bajo y alto y los nombres de los clientes que hicieron esos pagos.

## ENTRADAS

Su método recibirá como parámetro un `ArrayList` de instancias de la clase `Cliente` que representa los clientes que frecuentan la tienda de barrio. NO CREE LOS DATOS DE ENTRADA, DEBE USAR LOS QUE SE RECIBEN COMO PARÁMETRO.

## SALIDAS

Su método debe retornar un `Array` de `Object` donde:

- En la primera posición se guardará el promedio del total pagado por todos los clientes.
- En la segunda posición irá el nombre del cliente que menos pagó.
- En la tercera posición irá el total pagado más bajo.
- En la cuarta posición irá el nombre del cliente que más pagó.
- En la quinta posición irá el total pagado más alto.

**Nota:** Suponga que todos los clientes del `ArrayList` pagan diferentes montos (En este ejercicio no habrán dos clientes o más que paguen la misma cantidad de dinero).

## EJEMPLO

Suponga que se le entrega el siguiente Array:

```
//Cliente 3
ArrayList<Cliente> t3 = new ArrayList<>();
t3.add(new Cliente("Valeria Di", "10367876345", 9653, "03/07/2022", "0004"));
t3.add(new Cliente("Johan Doe", "1037645345", 3918, "03/07/2022", "0005"));
t3.add(new Cliente("Maurice Doe", "98765234", 6048, "03/07/2022", "0006"));
t3.add(new Cliente("Matthew Doe", "1036789453", 5840, "03/07/2022", "0007"));
t3.add(new Cliente("Agustina Doe", "10003456", 3940, "03/07/2022", "0008"));
t3.add(new Cliente("Agustina Doe", "10003456", 3840, "03/07/2022", "0009"));
t3.add(new Cliente("Milena Doe", "20003456", 3696, "03/07/2022", "0010"));
t3.add(new Cliente("Carla Di", "103789762", 2432, "03/07/2022", "0011"));
```

Se espera que su algoritmo retorne un Array de tipo Object con los siguientes datos:

```
run:
[4920.875, Carla Di, 2432, Valeria Di, 9653]
BUILD SUCCESSFUL (total time: 0 seconds)
```

## NOTA ACLARATORIA

Usted podrá desarrollar la prueba en Netbeans. Al final debe copiar y pegar el código en la herramienta VPL, pero **NO** deberá subir archivos, es decir:

**Modo incorrecto:**


**Modo correcto:**


**¡TRIPULANTE MUCHOS ÉXITOS EN EL DESARROLLO DEL RETO!**

## ACLARACIÓN DE PLAGIO

El objetivo es que los tripulantes cuenten con una oportunidad de aprendizaje relacionada con la programación. La colaboración académica es buena mientras no se lleve a un engaño académico, ya que el engaño académico inflige las buenas conductas del saber y del aprendizaje. El engaño académico hace referencia al plagio o envío de ideas que no son propias.

Colaborar implica compartir ideas, explicar a alguien cómo podría hacer su trabajo (más no hacer el trabajo por el otro) y ayudar al otro si tienes problemas a la hora de ejecutar o encontrar errores en el código.

En aras de evitar el plagio se recomienda colaborar pero no compartir su código o proyecto, no compartir sus soluciones, no usar un código encontrado en internet u otras fuentes que las propias. (Mason, Gavrilovska, y Joyner, 2019)

Los ejercicios enviados a verificación deben cumplir con la política antiplagio. Es decir, cualquier envío que sea una copia textual de otro trabajo puede ser suspendido o no aprobado por parte del equipo evaluador. El acto de copiar material de otro estudiante es un comportamiento inaceptable para el desarrollo de las competencias individuales y su progreso en este curso.

Referencia.

Mason, T., Gavrilovska, A., y Joyner, D.A. (2019). *Collaboration vesus cheating. 50th ACM Technical Symposium on Computer Science Education SIGCSE 2019*, Mineapolis, MN. DOI: 10.1145/3287324.3287443