

**Instrucciones:** Escriba el código en Python 3 que funcione como describe cada enunciado. Entregue el examen en un cuaderno de examen u hojas engrapadas. Firme al entregar el examen. No se permite el uso de dispositivos electrónicos o material que contenga información relacionada al curso, a excepción de una hoja con materia del curso. Si realiza el examen en lápiz, pierde el derecho a reclamos.

**Puntaje:** 100 puntos

## Enunciado

El Banco Nacional de Sangre ha hecho un llamado a la población costarricense por la escasez de donantes. Tras varias campañas de incentivo, la población ha aumentado considerablemente su interés de donar sangre. De hecho, ha sido tanto el éxito que el banco está teniendo problemas con el análisis y clasificación de cada una de las muestras. Por esto, han optado por automatizar este proceso, y han solicitado a la UCR que les ayude a contar la cantidad de sangre de cada tipo.

Un archivo en formato CSV, titulado `sangre.csv`, contiene la información correspondiente a todas las muestras de sangre donadas en una campaña. Cada fila del archivo describe una muestra de sangre, que contempla información sobre la presencia de antígenos de tipo A, antígenos de tipo B, antígenos Rh, y la cantidad de sangre donada. En el laboratorio clínico, a cada muestra se le realizan pruebas para determinar la presencia de Antígenos de varios tipos, que funcionan para obtener el tipo de sangre. Los antígenos de tipo A y B se obtienen de una prueba, y están registrados en el archivo como 1 si están presentes, o como 0 si están ausentes. Los antígenos de tipo Rh son en realidad una familia de 3 Antígenos: D, C y E. Este dato en el archivo es registrado como un texto de tamaño 3, en el formato "DCE". Las 3 letras de este dato pueden ser mayúsculas o minúsculas. Una letra en mayúscula denota presencia del antígeno, mientras que una letra en minúscula denota su ausencia. Por ejemplo, el dato "Dce" denota presencia del antígeno D, pero ausencia de los antígenos C y E. Finalmente, la cantidad de sangre donada se refiere al tamaño de la muestra, y viene en mililitros (mL). El siguiente es un ejemplo de cómo se vería este archivo:

---

AntiA;AntiB;AntiRh;Cantidad	1;0;DCE;482.316	0;1;DCE;450.966
0;0;DcE;493.140	1;0;Dce;491.428	0;0;Dce;491.341
0;0;DCE;483.840	1;0;DCE;462.968	0;0;Dce;507.811
0;0;dce;475.260	0;0;DCE;511.381	
0;1;DCE;458.248	0;0;DcE;480.700	

---

La clasificación del tipo de sangre tiene 2 características: el tipo ABO, y el tipo Rh. El tipo ABO está relacionado con la presencia de antígenos de tipo A y B en la sangre, y es considerado su tipo principal. La sangre es de tipo A si contiene solo antígenos de tipo A; es de tipo B si contiene solo antígenos de tipo B; es tipo AB si contiene ambos antígenos; o es de tipo O si no contiene ninguno. El tipo Rh está relacionado con los antígenos Rh, particularmente con el antígeno D. Se dice que la sangre es positiva (+) cuando hay presencia de antígenos D, y es negativa (-) cuando hay ausencia. Combinando estas dos clasificaciones se obtiene el tipo de sangre, por ejemplo AB+ si contiene los 3 antígenos A, B y D. En total, existen 8 tipos de sangre: A+, B+, AB+, O+, A-, B-, AB-, y O-.

El Banco Nacional de sangre quiere entonces saber los totales de sangre de cada tipo recolectados al final de la campaña. Cuando es ejecutado, el programa deberá leer los datos del archivo, y con base en estos mostrar, para cada tipo de sangre, el total de cantidad de sangre recolectada en **Litros (L)**. El siguiente es un ejemplo de cómo debería ser la salida del programa:

---

Sangre recolectada  
A+: 1.44 L  
B+: 0.91 L  
AB+: 0.00 L  
O+: 2.97 L

A-: 0.00 L  
B-: 0.00 L  
AB-: 0.00 L  
O-: 0.48 L

---

Finalmente, el programa debería estar programado en el paradigma orientado a objetos. Se utilizará un diseño de 2 clases: *Muestra* y *Banco*. *Muestra* estaría encargada de almacenar la información correspondiente a los antígenos y la cantidad de sangre de cada muestra, y a partir de estas calcular su tipo de sangre. *Banco* estaría encargada de almacenar las instancias de *Muestra*, leer la información del archivo, y mostrar los resultados.

## Evaluación

1. (40%) Implementa la clase *Muestra*
  - a. (10%) La muestra tiene atributos que le permiten almacenar la información sobre presencia de antígenos A, B, Rh; y la cantidad de sangre.
  - b. (30%) La muestra implementa los siguientes métodos:
    - i. (5%) Un constructor para inicializar los atributos de la clase.
    - ii. (10%) *tipo\_ABO(t)* que calcula y retorna el tipo ABO de la muestra (ej: B).
    - iii. (10%) *tipo\_Rh(t)* que calcula y retorna el tipo Rh de la muestra (ej: +).
    - iv. (5%) *tipo\_sangre()* que calcula y retorna el tipo de sangre de la muestra (ej: B+).
2. (60%) Implementa la clase *Banco*
  - a. (5%) El banco tiene un atributo que le permite representar las varias muestras
  - b. (55%) El banco implementa los siguientes métodos:
    - i. (5%) Un constructor para inicializar los atributos de la clase.
    - ii. (30%) *leer\_datos()* que abre el archivo sangre.csv, lee la información de este, crea instancias de muestra y las almacena.
    - iii. (20%) *mostrar()* que muestra, para cada tipo de sangre, la cantidad de sangre total recolectada, en Litros.

**Nota:** En cada punto se evaluará la correctitud, completitud, y buenas prácticas de programación. Estas incluyen, pero no se limitan a identificadores significativos para variables e indentación adecuada.

**Nota:** El diseño mostrado es una sugerencia. Puede utilizar otro diseño de clases, métodos y atributos si lo considera adecuado; siempre y cuando implemente la funcionalidad descrita.