



RETO 4 – PROGRAMACIÓN BÁSICA

VARIANTE 1

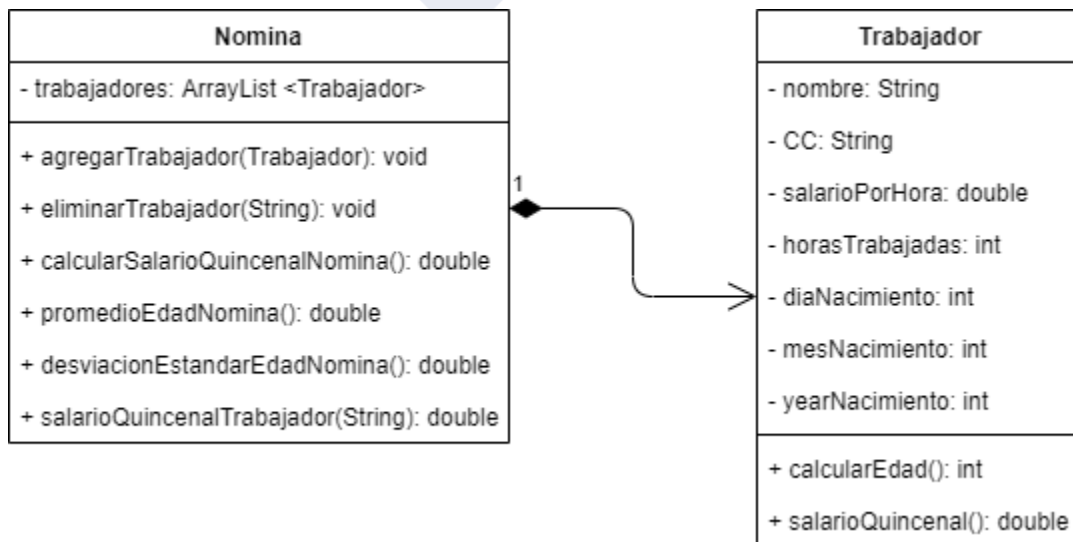
La Universidad de Antioquia es una institución reconocida por sus altos estándares de calidad en la educación e investigación en Colombia.

Para lograr su misión, la universidad dispone de personal preparado con gran sentido de pertenencia y calidad humana para desempeñar (Cada uno en su especialidad) diferentes labores que permiten alcanzar las metas que tiene trazadas; sin embargo ha habido un problema en el sistema de nóminas y se necesita desarrollar de carácter urgente un sistema secundario que permita sustituir temporalmente el sistema averiado, ya que según el arquitecto de software, su puesta en funcionamiento puede llevar varias semanas en lograrse, y los pagos al personal no dan espera.

Usted, como tripulante de Misión TIC 2022, ha mostrado un buen desempeño en la realización de los diferentes retos del Ciclo 1 y Ciclo 2 de formación, por lo que ha sido contratado como desarrollador junior, con el motivo de que vaya adquiriendo experiencia en el área de la tecnología.

Usted está encargado de implementar las siguientes clases que ha detallado el área de ingeniería de software de la universidad, para facilitar la codificación de estas clases, el equipo de Ingeniería de software le hace entrega del diagrama de clases (Recuerde que los métodos relacionados a los *getters* y *setters*, así como también el método constructor son obviados en el diagrama de clases, pero deberán ser incluidos en el código, estos métodos deberán ser creados con el estándar camel case).





Además del diagrama, el área de Ingeniería le hace entrega de esta documentación para comprender mejor los elementos del diagrama:

Clase Trabajador

Atributos

NOMBRE	TIPO DATO	CONCEPTO	INICIALIZACIÓN
nombre	String	Guarda el nombre del trabajador	En el método constructor
CC	String	Guarda la cédula del trabajador	En el método constructor
salarioPorHora	double	Guarda el salario que gana el trabajador por hora	En el método constructor
horasTrabajadas	int	Guarda las horas trabajadas por el trabajador en una quincena	En el método constructor
diaNacimiento	int	Guarda el día de nacimiento del trabajador (Es un número entre 0 y 31)	En el método constructor





mesNacimiento	int	Guarda el mes de nacimiento del trabajador (Es un número entre 1 y 12)	En el método constructor
yearsNacimiento	int	Guarda el mes de nacimiento del trabajador (Es un número entre 1900 y 3000)	En el método constructor

Métodos

NOMBRE	TIPO RETORNO	PARÁMETROS	CONCEPTO
calcularEdad	int	No recibe	Retorna la edad del trabajador (Este método ya se le entrega implementado)
salarioQuincenal	double	No recibe	Retorna el salario quincenal del trabajador (Es el producto entre salarioPorHora y horasTrabajadas)

Clase Nomina

Atributos

NOMBRE	TIPO DATO	CONCEPTO	INICIALIZACIÓN
trabajadores	ArrayList <Trabajador>	Guarda la lista de trabajadores que hacen parte de la nómina de la universidad	Vacío





Métodos

NOMBRE	TIPO RETORNO	PARÁMETROS	CONCEPTO
agregarTrabajador	void	Trabajador t: Es el trabajador por agregar a la lista trabajadores	Agrega el trabajador recibido como parámetro a la lista trabajadores si este NO está en la lista (Dos trabajadores son diferentes si sus documentos de identidad son diferentes)
eliminarTrabajador	void	String ID: Documento de identidad del trabajador a eliminar de la lista trabajadores	Elimina el trabajador con documento de identidad igual al recibido por parámetro (ID)
calcularSalarioQuincenalNomina	double	No recibe	Retorna la suma del salario quincenal de todos los trabajadores
promedioEdadNomina	double	No recibe	Retorna el promedio de edad de todas las personas que hacen parte de la nómina
desviacionEstandarEdadNomina	double	No recibe	Retorna la desviación estándar de la edad de todas las personas que hacen parte de la nómina
salarioQuincenalTrabajador	double	String ID: Documento de identidad del trabajador del que se desea conocer su	Retorna el salario quincenal del trabajador cuyo documento de identidad es igual a ID (Si este no se encuentra, retorna -1)





		salario quincenal	
--	--	----------------------	--

PRECISIONES

1. No hay métodos estáticos.
2. Deben existir *getters* y *setters* de todos los atributos de cada clase, estos deben ser escritos en la forma estándar camel case.
3. La fórmula de la desviación estándar a usar es: $\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2}$, donde n es el número de muestras a considerar en el cálculo (Número de trabajadores que hay en el ArrayList), x_i es la edad de cada trabajador y μ es el promedio de edad del conjunto de trabajadores que hay en el ArrayList.
4. El calificador asumirá un error del 1% por aproximaciones que su código pueda hacer en el cálculo del promedio y la desviación estándar.

TAREAS

- En el archivo preconstruído en la plataforma Moodle, codificar las clases especificadas en el diagrama de clases, teniendo en cuenta las precisiones dadas por el equipo de Ingeniería de software.
- Los nombres de los métodos y atributos **DEBEN** ser nombrados tal y como aparecen en el diagrama de clases.
- Usted **NO** debe solicitar datos por teclado, ni programar un método `main`, tampoco use `Java Source Package`, usted está solamente encargado de la construcción de la clase.





NOTA ACLARATORIA

Usted podrá desarrollar la clase requerida en un IDE como NetBeans, y al final copiar y pegar el código en la herramienta VPL, pero **NO** deberá subir archivos, es decir:

Modo incorrecto:

La imagen muestra la interfaz de entrega en VPL. En la parte superior, hay una barra de navegación con los logos de 'Misión TIC 2022', 'Ingeniería' y 'UNIVERSIDAD DE ANTIOQUIA'. Debajo, se indica el curso: 'SEM-Programacion basica ciclo 2'. El título principal es 'SEM-Programacion básica ciclo 2'. A la derecha, un texto rojo grande dice 'NO SUBIR NINGÚN ARCHIVO'. En el centro, hay un campo de texto con el texto 'Entrega' y un botón 'Entrega'. Debajo de este, hay un campo de comentarios. En la parte inferior, hay un botón 'Seleccione un archivo...' y un mensaje 'Tamaño máximo para archivos nuevos: 5MB'. Hay un área de arrastrar y soltar archivos con el texto 'Puede arrastrar y soltar archivos aquí para añadirlos'. En la parte inferior, hay botones 'Enviar' y 'Cancelar'. En la esquina inferior derecha, hay un pequeño texto 'VPL'. Hay una flecha verde que apunta al campo de texto 'Entrega' con el texto 'COPIA TU CÓDIGO ACÁ (Como en el Ciclo 1)'.

Modo correcto:

La imagen muestra la interfaz de entrega en VPL. En la parte superior, hay una barra de navegación con los logos de 'Misión TIC 2022', 'Ingeniería' y 'UNIVERSIDAD DE ANTIOQUIA'. Debajo, se indica el curso: 'SEM-Programacion basica ciclo 2'. El título principal es 'SEM-Programacion básica ciclo 2'. En el centro, hay un campo de texto con el texto 'Entrega' y un botón 'Entrega'. Debajo de este, hay un campo de comentarios. En la parte inferior, hay un botón 'Seleccione un archivo...' y un mensaje 'Tamaño máximo para archivos nuevos: 5MB'. Hay un área de arrastrar y soltar archivos con el texto 'Puede arrastrar y soltar archivos aquí para añadirlos'. En la parte inferior, hay botones 'Enviar' y 'Cancelar'. En la esquina inferior derecha, hay un pequeño texto 'VPL'. Hay una flecha verde que apunta al campo de texto 'Entrega' con el texto 'LUGAR CORRECTO'. Hay un recuadro rojo con el texto 'CLIC AQUÍ PARA ACCEDER AL ENUNCIADO'.