



RETO 3 – PROGRAMACIÓN BÁSICA

CONTEXTO

La casa desarrolladora UdeASoft desea desarrollar una nueva plataforma Moodle para uso exclusivo de la Universidad de Antioquia.

Los usuarios de esta plataforma estarán diferenciados inicialmente en dos (Formador y tripulante) dado que tienen permisos diferentes, por ejemplo:

- El tripulante podrá entregar tareas en la plataforma (Entregar retos).
- El formador podrá calificar con 0 o 5 la tarea entregada por el tripulante (Para este caso se asume que la tarea entregada por el tripulante está correcta).

A pesar de tener diferentes permisos en la plataforma comparten algunas características (Ambos roles poseen estos datos) como: nombre, documento de identidad, contraseña, correo electrónico y descripción del perfil, y tienen la opción de editar el perfil.

Usted ha sido contratado como Desarrollador Senior en Java, porque ha logrado demostrar habilidades de desarrollo en este lenguaje de programación y se le ha concedido implementar las clases correspondientes a `UsuarioMoodle` (Super clase), `Formador` y `Tripulante`.

Para facilitar la implementación de estas tres clases, el equipo de Ingeniería de software le hace entrega del diagrama de clases (Figura 1).



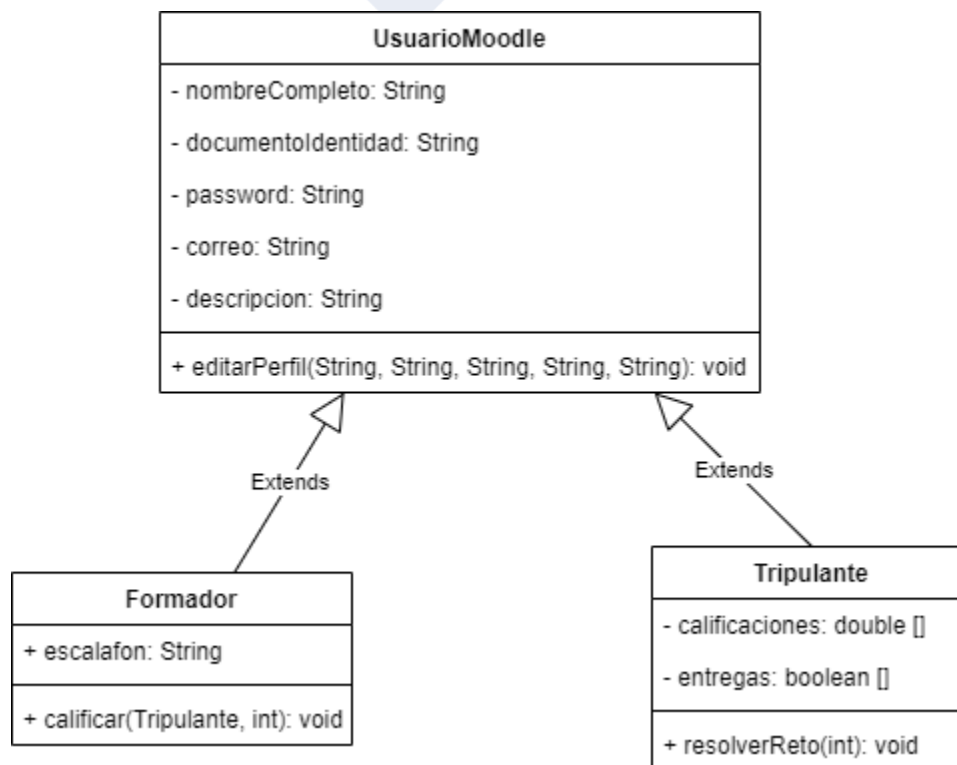


Figura 1

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`, para el caso de los atributos de tipo boolean, el get se cambia por un `is`, por ejemplo, si el atributo se llama `pausado` y es de tipo boolean, su getter será `isPausado`.

Nota: Recuerde que desde NetBeans puede generar automáticamente los getter y setters con la opción `Insert Code...` (Figura 2)

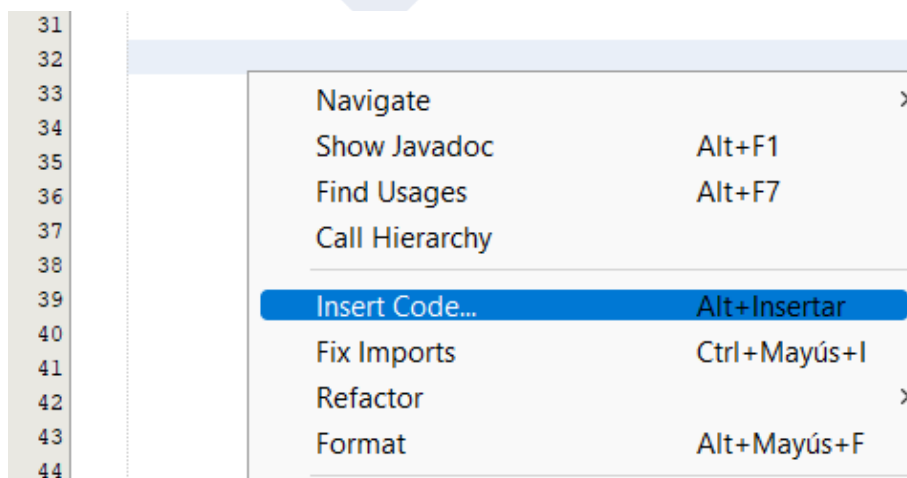


Figura 2

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

Clase UsuarioMoodle

Atributos (El tipo de dato de TODOS los atributos de esta clase es String y además TODOS se inicializan en el constructor)

NOMBRE	CONCEPTO
nombreCompleto	Contiene el nombre del usuario
documentoIdentidad	Contiene el número de documento de identidad del usuario
password	Contiene la contraseña de la cuenta del usuario
correo	Contiene el correo electrónico del usuario
descripcion	Contiene una breve descripción del usuario

Nota: Los parámetros del método constructor deben ponerse en el orden que aparecieron en la tabla anterior de arriba hacia abajo.





Métodos (Este método no retorna datos, es decir, el tipo de retorno es void)

NOMBRE	PARÁMETROS	CONCEPTO
editarPerfil	<ul style="list-style-type: none">String nombreCompletoString documentoIdentidadString passwordString correoString descripcion	Actualiza toda la información del usuario con los nuevos datos

Clase Tripulante (Hija - Hereda de UsuarioMoodle)

Atributos

NOMBRE	TIPO DATO	CONCEPTO	INICIALIZACIÓN
calificaciones	double []	Guarda en la posición i la calificación obtenida en la entrega i (0 si no entregó el reto y 5 si entregó el reto)	Se inicializa con una longitud de n y con 0 en todas las posiciones
entregas	boolean []	Guarda en la posición i <code>true</code> si el tripulante ha hecho alguna entrega y <code>false</code> en caso contrario	Se inicializa con una longitud de n y <code>false</code> en todas las posiciones

Nota: El método constructor recibirá en este orden los parámetros:

1. nombreCompleto
2. documentoIdentidad
3. password
4. correo
5. descripcion
6. n

n es la cantidad de tareas que tendrá el tripulante para entregar a lo largo del curso.

Métodos (Este método no retorna datos, es decir, el tipo de retorno es void)

NOMBRE	PARÁMETROS	CONCEPTO
--------	------------	----------





<code>resolverReto</code>	<code>int retoAEntregar: Asigna true a la posición retoAEntregar del array entregas</code>	Le permite al tripulante realizar una entrega del reto especificado
---------------------------	--	---

Clase Formador (Hija - Hereda de UsuarioMoodle)

NOMBRE	TIPO DATO	CONCEPTO	INICIALIZACIÓN
<code>escalafon</code>	<code>String</code>	Contiene el escalafón en el que está ubicado el formador	En el método constructor

Métodos (Este método no retorna datos, es decir, el tipo de retorno es void)

NOMBRE	PARÁMETROS	CONCEPTO
<code>calificar</code>	<ul style="list-style-type: none"> • <code>Tripulante</code> <code>tripulanteACalificar</code>: Es el tripulante al que le asignará 0 o 5 en la entrega en la posición k • <code>int k</code>: Reto a calificar 	Permite al formador calificar una actividad realizada por un tripulante

Notas:

- El constructor tendrá los parámetros en el siguiente orden:
 1. `escalafon`
 2. `nombreCompleto`
 3. `documentoidentidad`
 4. `password`
 5. `correo`
 6. `descripcion`
- El formador asigna 0 en la posición k del arreglo `calificaciones` si `entregas[k]` es igual a `false`.
- El formador asigna 5 en la posición k del arreglo `calificaciones` si `entregas[k]` es igual a `true`.





TAREAS

- En los archivos preconstruidos en la plataforma Moodle, implementar 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 debe especificar el paquete al que pertenece la clase, usted está solamente encargado de la construcción de la clase.

EJEMPLO

El calificador hará veces de usuario y será quien evalúe el funcionamiento de los roles que usted desarrolló:

1. El calificador le crea una cuenta al estudiante John, tripulante de Misión TIC 2022 y una cuenta a la formadora Jane, profesora de Misión TIC 2022. Al tripulante se le calificarán 5 retos:

```
Tripulante John = new Tripulante(  
    "John Doe",  
    "1037654354",  
    "123456789",  
    "john.doe@udea.edu.co",  
    "Soy una persona con muchas ganas de aprender",  
    5  
);  
  
Formador Jane = new Formador(  
    "3D",  
    "Jane Di",  
    "1038654355",  
    "987654321",  
    "jane.di@udea.edu.co",  
    "¡Me apasiona enseñar!"  
);
```

2. John resuelve los retos 2, 3 y 4:

```
John.resolverReto(1);  
John.resolverReto(2);  
John.resolverReto(3);
```

3. Jane le califica los 5 retos a John:





```
for(int i = 0; i < 5; i++)  
    Jane.calificar(John, i);
```

4. John y Jane actualizan su perfil:

```
John.editarPerfil("John Doe", "1037654354", "123456789", "john.doe@udea.edu.co", "¡Estoy muy feliz en este programa!");  
Jane.editarPerfil("Jane de María Di", "1038654355", "987654321", "jane.di@udea.edu.co", "¡Enamorada de la vida!");
```

5. Al final los perfiles de ambos quedan así:

Nombre completo: John Doe

Documento de identidad: 1037654354

Contraseña: 123456789

Correo electrónico: john.doe@udea.edu.co

Descripción: ¡Estoy muy feliz en este programa!

Entregas: [false, true, true, true, false]

Calificaciones: [0.0, 5.0, 5.0, 5.0, 0.0]

Nombre completo: Jane de María Di

Documento de identidad: 1038654355

Contraseña: 987654321

Correo electrónico: jane.di@udea.edu.co

Descripción: ¡Enamorada de la vida!

Escalafón: 3D





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

Modo correcto:

La imagen muestra la interfaz de entrega de una tarea en VPL, pero en el modo correcto. El título de la tarea es 'SEM-Programación básica ciclo 2'. A la derecha, un texto verde grande dice 'LUGAR CORRECTO'. En el centro, hay un editor de código con el archivo 'Personaje.java' abierto. El código muestra una clase 'Personaje' con comentarios para insertar atributos, constructor, métodos y setters/getters. A la derecha del editor, hay un panel de descripción con el texto 'CLIC AQUÍ PARA ACCEDER AL ENUNCIADO'.

¡MUCHOS ÉXITOS EN EL DESARROLLO DEL RETO 3 TRIPULANTE!





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

