*Taller 1 Patron*

*Link:*

[*https://onlinegdb.com/dt7Nbq3E8*](https://onlinegdb.com/dt7Nbq3E8)

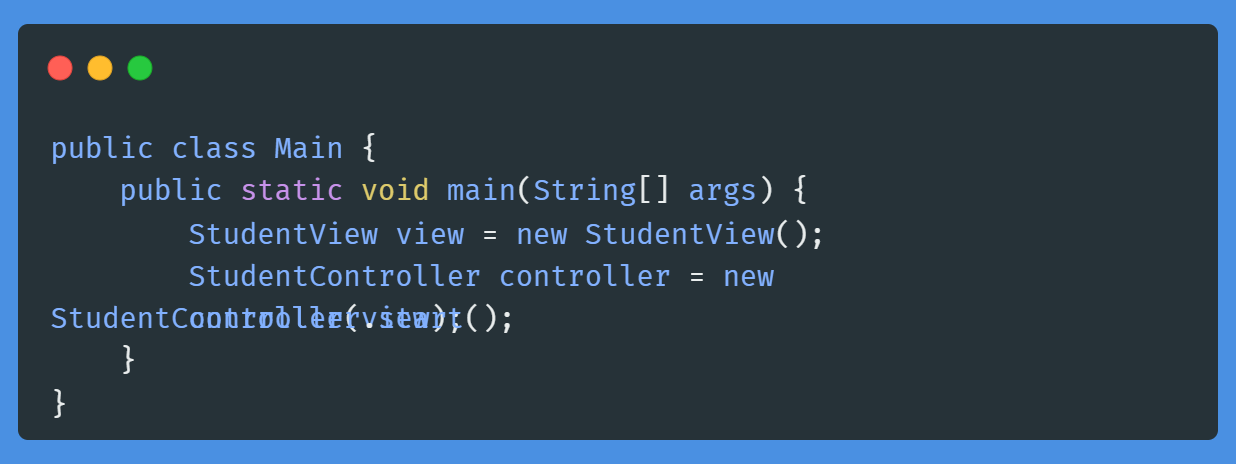
| *RÚBRICA PARA EVALUACIÒN* | | |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
| *Preguntas* | *Puntos* | *Calificación* | *Observación* |
| 1. La clase main, llama a la vista (View), y al controlador (Controler) | 1 | 1 | El main llama a las dos capas responsables de la actividad |
| 1. Elaborar el modelo en base de datos (BD) al dado, únicamente se adaptando, añadiendo tres nuevos Constructores | 1 | 0.75 | El tercer constructor no asigna correctamente el atributo *name*. |
| 1. Para la Vista view crear un método de inserción, el cual simula cómo sería la inserción tradicional | 1 | 1 | Se debería crear el método no simulado para probar la entrada de datos |
| 1. El controlador se cambió en su mayoría, haciendo uso únicamente de los métodos que se requieren para hacer un intermediario entre el modelo, y la vista. | 1 | 1 | El controlador utiliza sólo los métodos necesarios para que se trabaje la lógica, los datos y la vista. |
| 1. Finalmente, crear una clase que simula una base de datos, la cual brinda apoyo en la administración de los datos quemados. (05 estudiantes). | 1 | 1 | La clase *StudentDatabase* simula la base de datos mediante una lista de objetos del tipo *Student.* |
| EJECUCIÓN |  |  | El programa se ejecuta con éxito. |
| TOTAL | 5 | 4.75 / 5 |  |

REQUISITOS:

Para cada RF realizar la revisión de código y explicar a través de la ejecución el funcionamiento de MVC

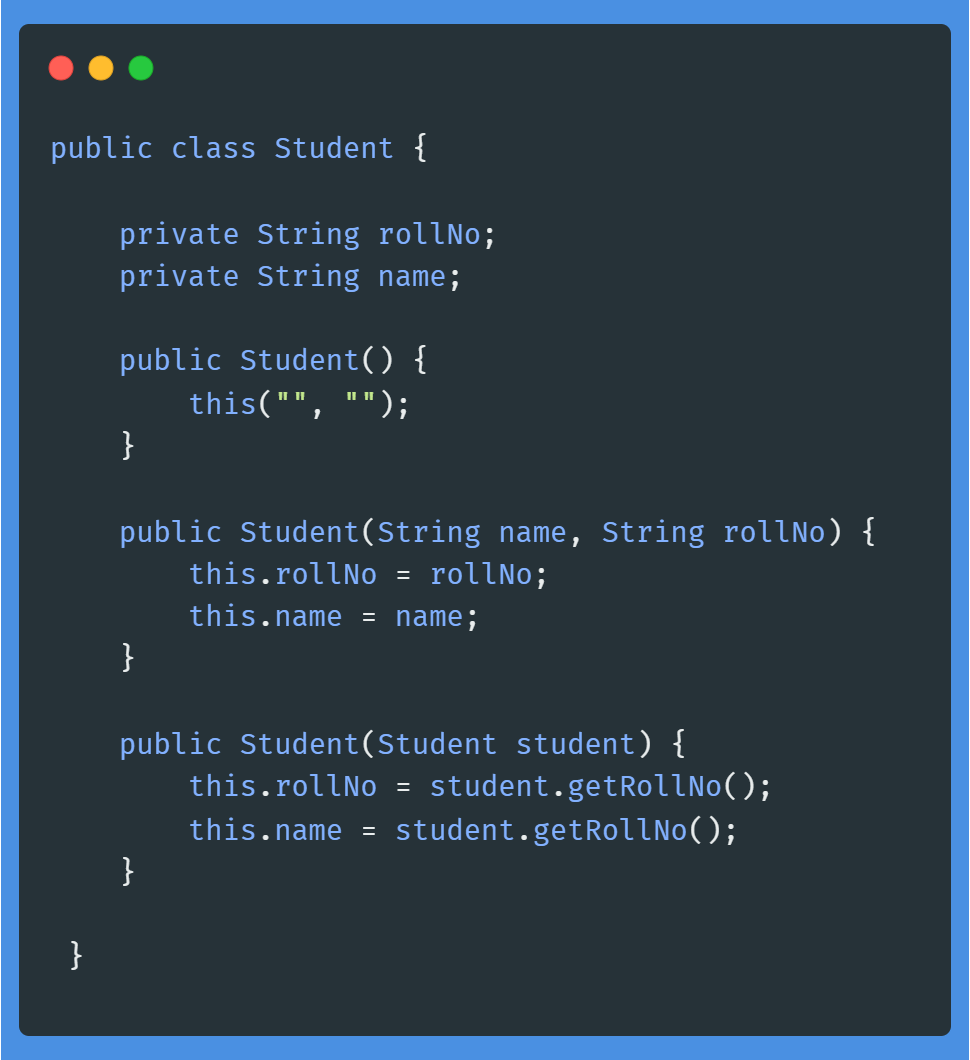
1. La clase main, llama al View, y al controlador

Sí, en el código la clase Main crea instancias tanto de la vista como del controlador. El controlador recibe la vista como argumento en su constructor, lo que permite interactuar con la vista y gestionar las operaciones entre el modelo y la vista, la ejecución de estos métodos se evidencia en la Figura 1, Figura 2 y Figura 3.



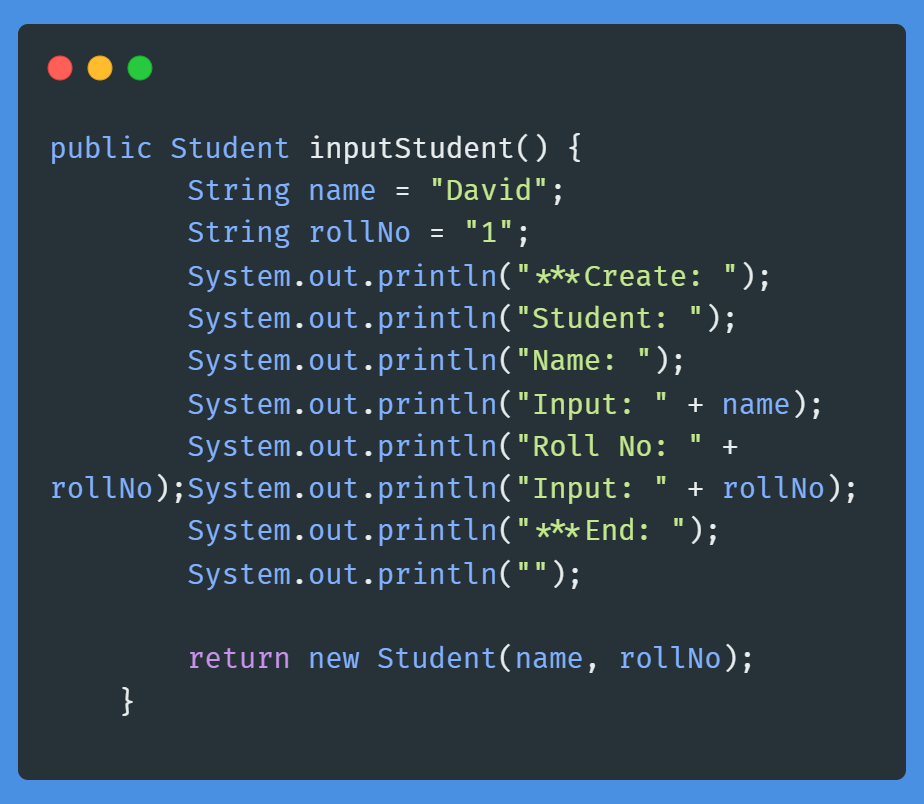
1. Se hizo el modelo en base al dado, únicamente se adaptando, añadiendo tres nuevos Constructores

Sí, el modelo Student incluye tres constructores, sin embargo el tercero asigna incorrectamente el atributo *name*. La aplicación del segundo de ellos se puede observar en la Figura 2.



1. Para el view se creó un método de inserción, el cual simula cómo sería la inserción tradicional

Sí, el método *inputStudent()* simula la inserción de un estudiante con datos predeterminados. Debido a que este método utiliza el segundo constructor de la clase *Student.java*, y es llamado en la creación del *StudentController.java*, la ejecución de este se puede observar en la Figura 2.



1. El controlador se cambió en su mayoría, haciendo uso únicamente de los métodos que se requieren para hacer un intermediario entre el modelo, y la vista

Si el controlador maneja las interacciones entre la vista y el modelo. Su función principal en este caso es invocar métodos de la vista para simular la entrada de datos , interactúa con el modelo para obtener y modificar los datos y luego actualizarlos , esto lo realiza mediante los métodos fetchStudents(), createStudent(), y updateStudent()



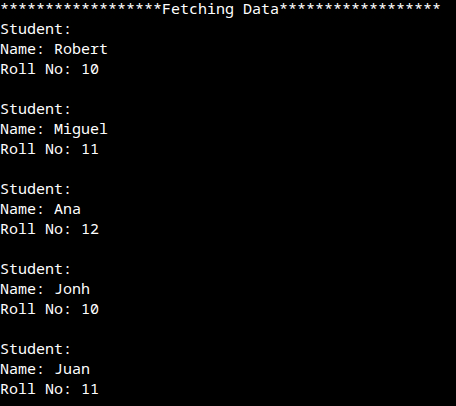
1. Se creó una clase que simula una base de datos, la cual brinda apoyo en la administración de los datos quemados.

Si está creada la clase que simula la base de datos, con 5 registros predefinidos, un método post y un delete para poder simular la persistencia de los datos con una Lista dentro del programa. En la Figura 1. se evidencia cómo esa información quemada de los 5 estudiantes si se encuentra cargada y en la Figura 3. que si se actualiza al insertar.



**Ejecucion del codigo/ Pantallas**

En la captura se evidencia como la capa de la vista a través de la consola muestra a los usuarios la información que el controlador obtiene de la base de datos

  
Figura 1. Captura de ejecución del método fetchStudents.

Aquí se puede observar la creación de un estudiante mediante el segundo constructor definido en *Student.java*.

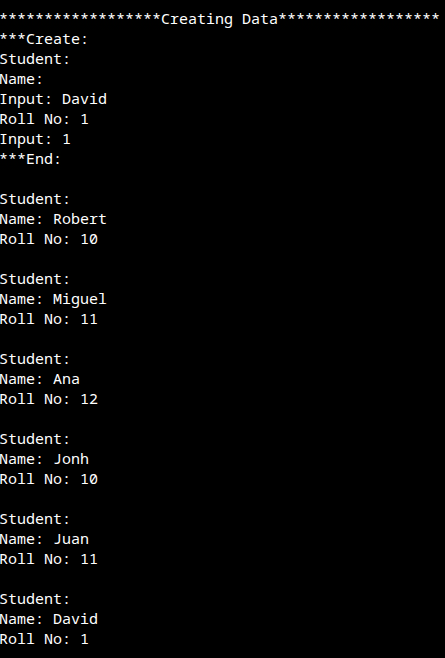
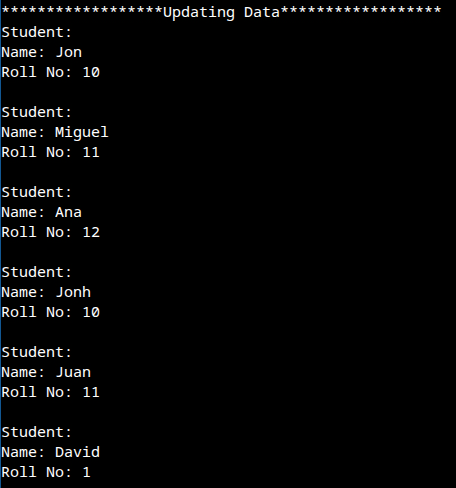


Figura 2. Creación de un estudiante.

Y luego se evidencia la inserción de esa información a la Base de Datos Simulada

  
Figura 3. Actualización de la información en la base de datos simulada.