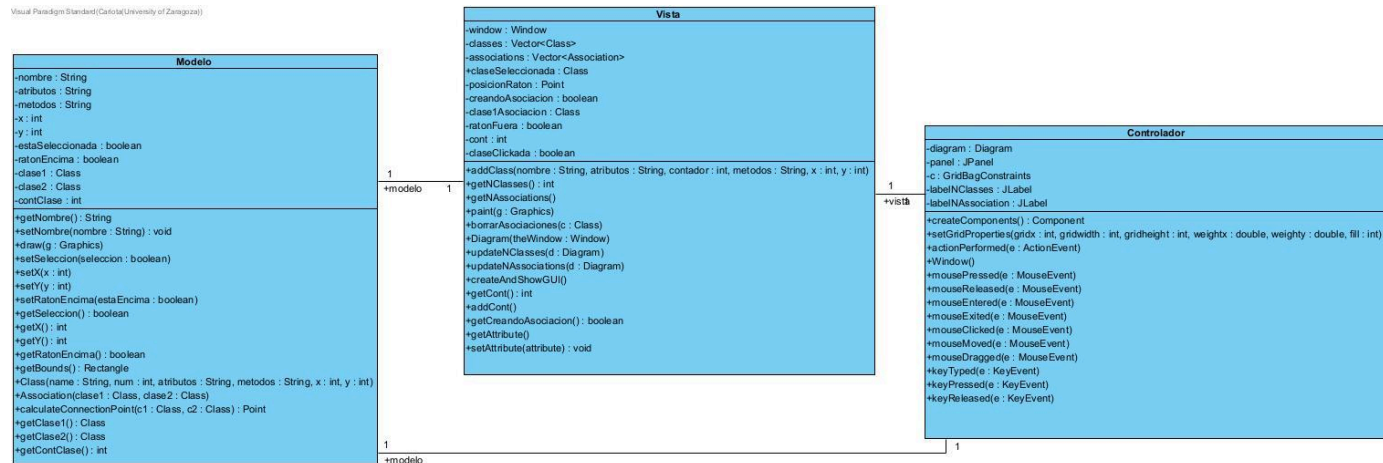


## PRÁCTICA 2 - MODELO VISTA CONTROLADOR

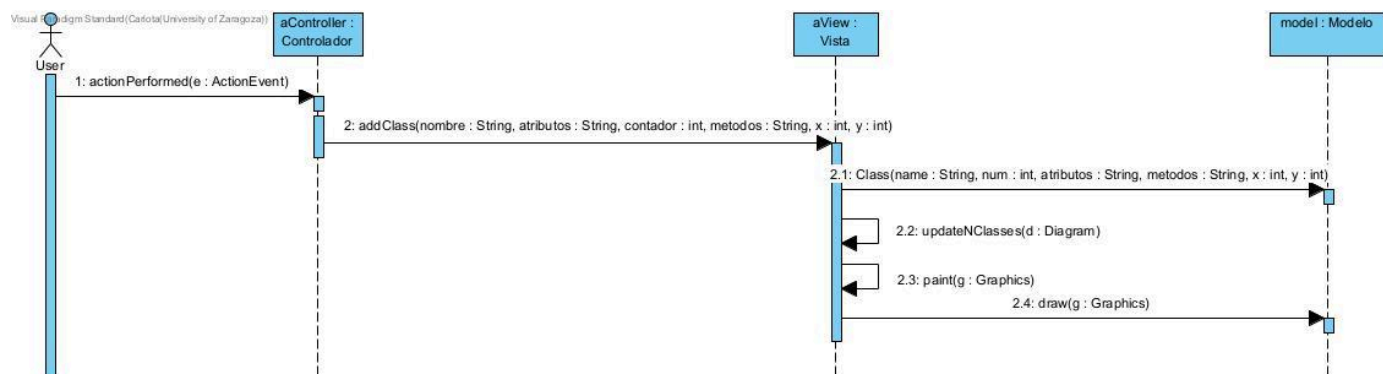
### Diagrama de clases



La clase Diagram actúa como Controlador y Vista, mientras que las clases (clase Class) y las asociaciones (clase Associations) definidas corresponden a Modelo.

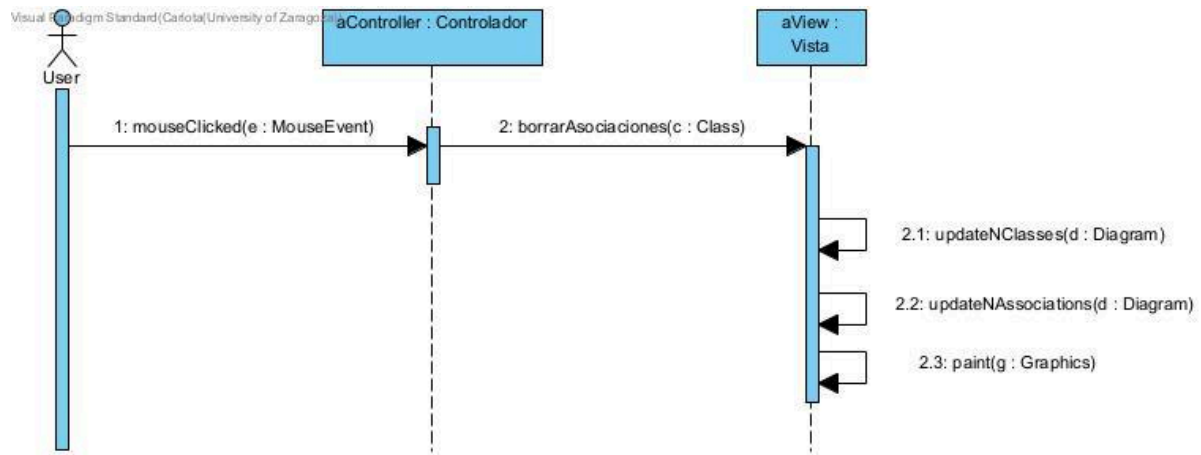
### Diagramas de secuencia

#### Creación de una clase nueva

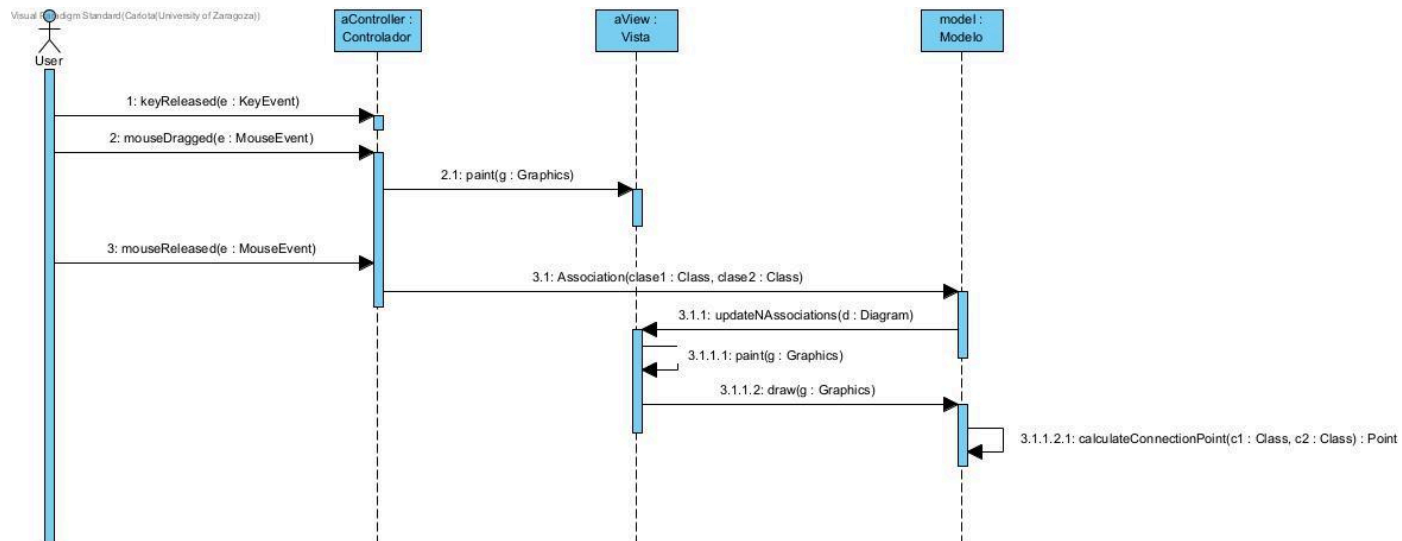


Carlota Moncasi Gosá 839841  
Paula Soriano Sánchez 843710

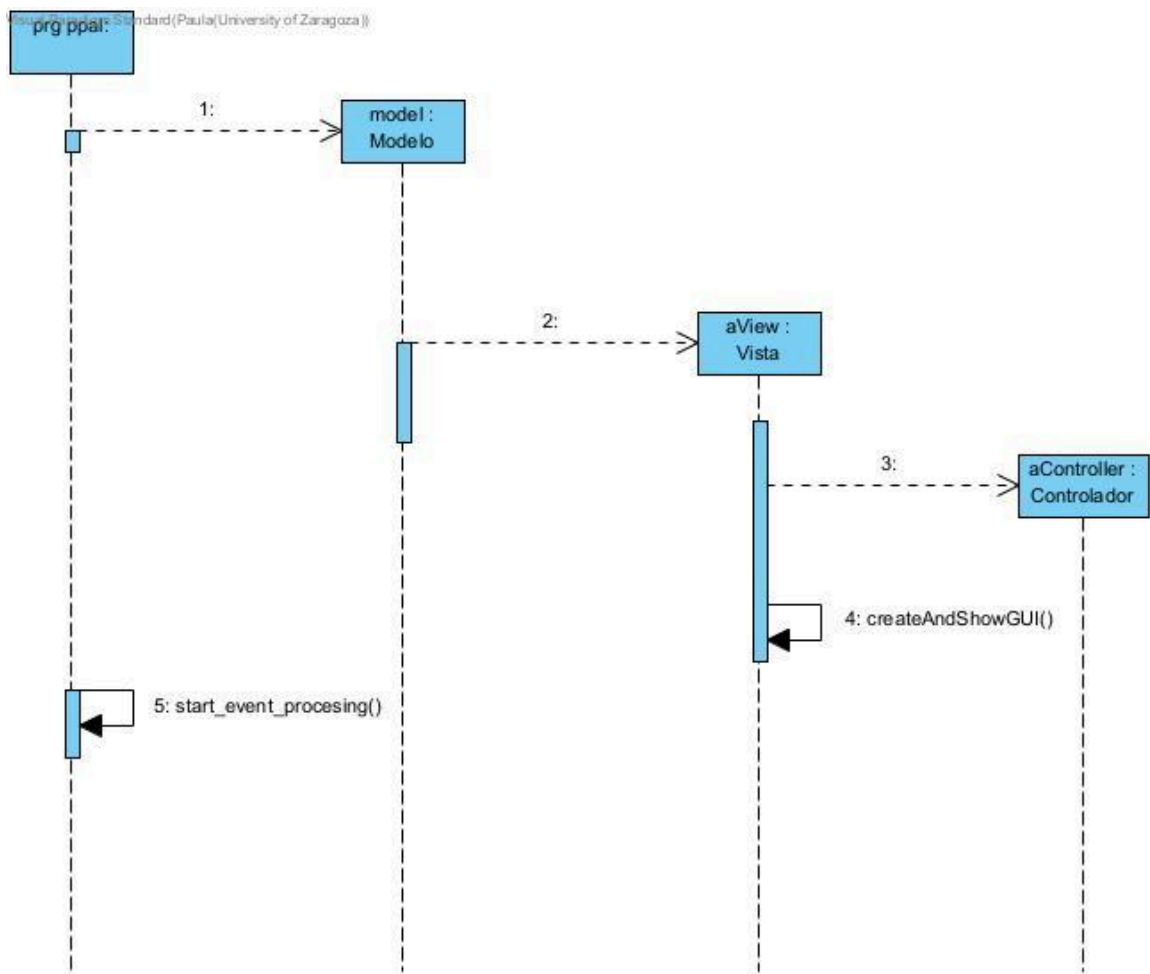
## Eliminación de una clase y sus asociaciones



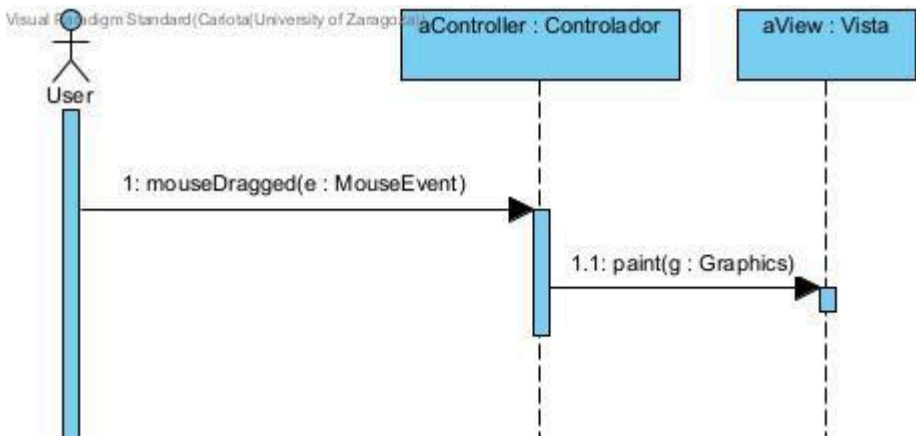
## Creación de una asociación



Inicialización del programa



Desplazamiento de una clase



## Implementación del patrón MVC en el código

El código que tenemos en java difiere del patrón MVC en algunos aspectos. Este patrón cuenta con una clase Modelo y dos clases que heredan de la clase Observer, que se denominan Controlador y Vista.

La clase Controlador se refiere a todos aquellos métodos que reciben una notificación cada vez que el usuario realiza una acción que afecta a la visualización de ciertas opciones que el usuario puede o no hacer tras la que ha realizado. En nuestro código, la clase Controlador se refiere a todos aquellos métodos que se corresponden con los métodos implementados por los *Listeners*, implementados en la clase Diagram.

Por otro lado, la clase Vista se corresponde con aquellos métodos que afectan a la visualización del sistema, en nuestro código implementados en la clase Diagram. Por lo tanto, la clase Diagram es al mismo tiempo Controlador y Vista.

Estas dos clases (Controlador y Vista) son observadores de la clase Modelo. Esta clase contiene toda la información que se muestra y modifica. En nuestro código en java, esta clase queda representada por los atributos y métodos correspondientes a las clases Class y Association. Dado que en el código solamente hay un observador de la clase Modelo (la clase Diagram), no es necesario incluir una clase Observer, por lo que no se ha representado en el diagrama de clases ni en el código, y en consecuencia tampoco se incluye ninguno de sus métodos *attach()*, *detach()* y *notify()*.