

Memoria DS

Reparto de tareas

Integrantes:

- Javier Serrano Lucas
- Andrés Merlo Trujillo
- Sergio Hervás Cobo
- Ricardo Molina Rodríguez

El reparto de tareas se ha realizado de la siguiente manera:

En la sesión 1:

- **Módulo 1:** Javier y Sergio
- **Módulo 2:** Andrés y Ricardo

En la sesión 2:

- **Módulo 1:** Andrés y Ricardo
- **Módulo 2:** Javier y Sergio

En la sesión 3 hemos trabajado entre todos; sin embargo, Ricardo sufrió un accidente que le ha impedido realizar esta parte. Por tanto, realmente hemos sido 3 personas las que hemos realizado esta sesión.

Si bien este ha sido el reparto de tareas "oficial", los dos grupos se han estado retroalimentando y participando muy activamente en el código de los otros. Por tanto, realmente no solo ha sido un trabajo en parejas.

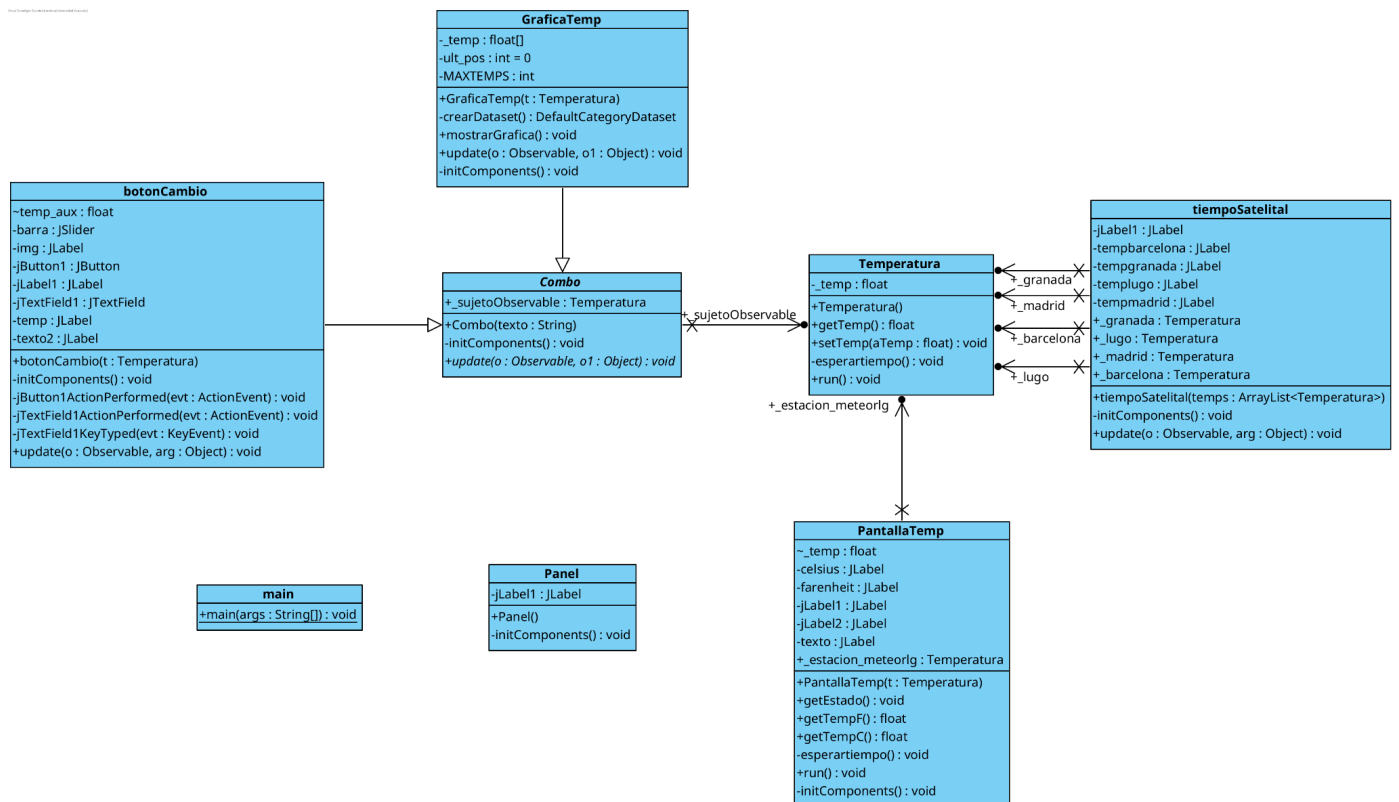
Los diagramas (tanto la imagen como el proyecto de visual paradigm) han sido incluidos en la carpeta "VPP" de la entrega.

Sesión 1:

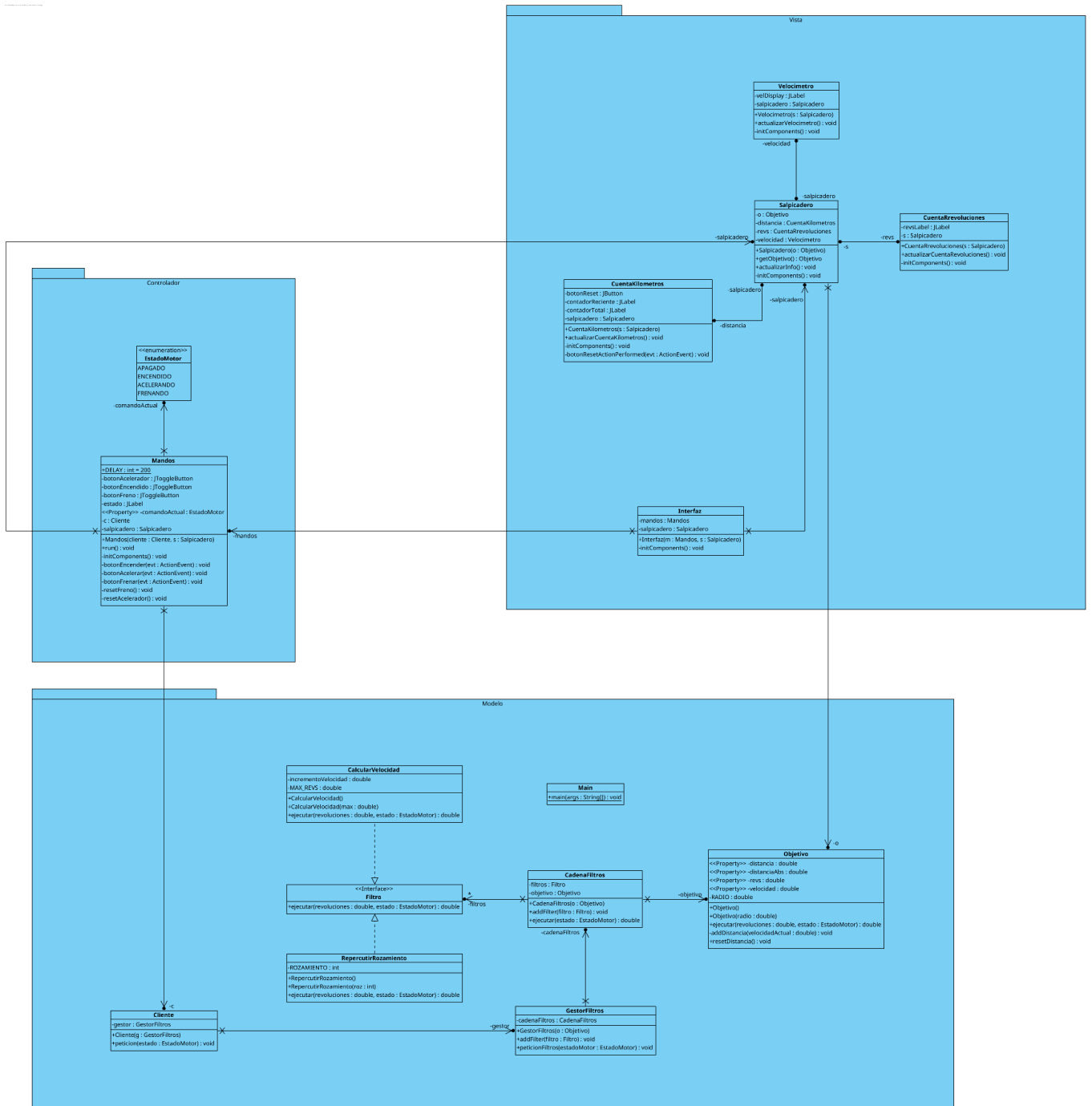
Módulo 1:

En este módulo hemos elegido como tercer observador el “botonCambio”, el cual tiene dos formas de cambiar la temperatura: por texto o por un deslizador. Además, finalmente hicimos voluntariamente los otros dos observadores propuestos.

Somos conscientes del número de ventanas que hay, pero hemos seguido el guión y tal y como dice debe ser una ventana por observador. En todas las ventanas se monitoriza y utiliza el tiempo de “Granada” del observador tiempoSatelital.



Módulo 2:

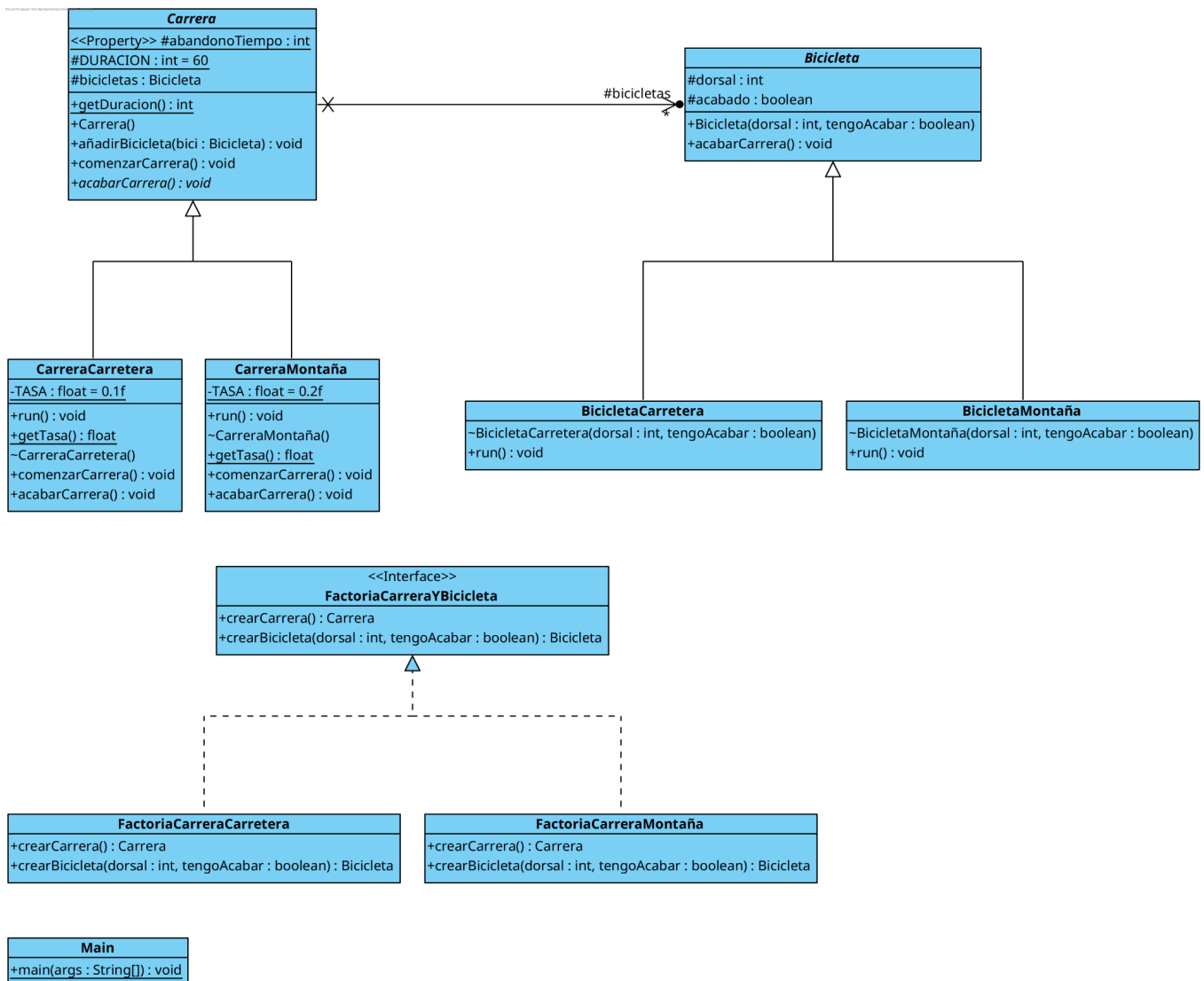


Sesión 2:

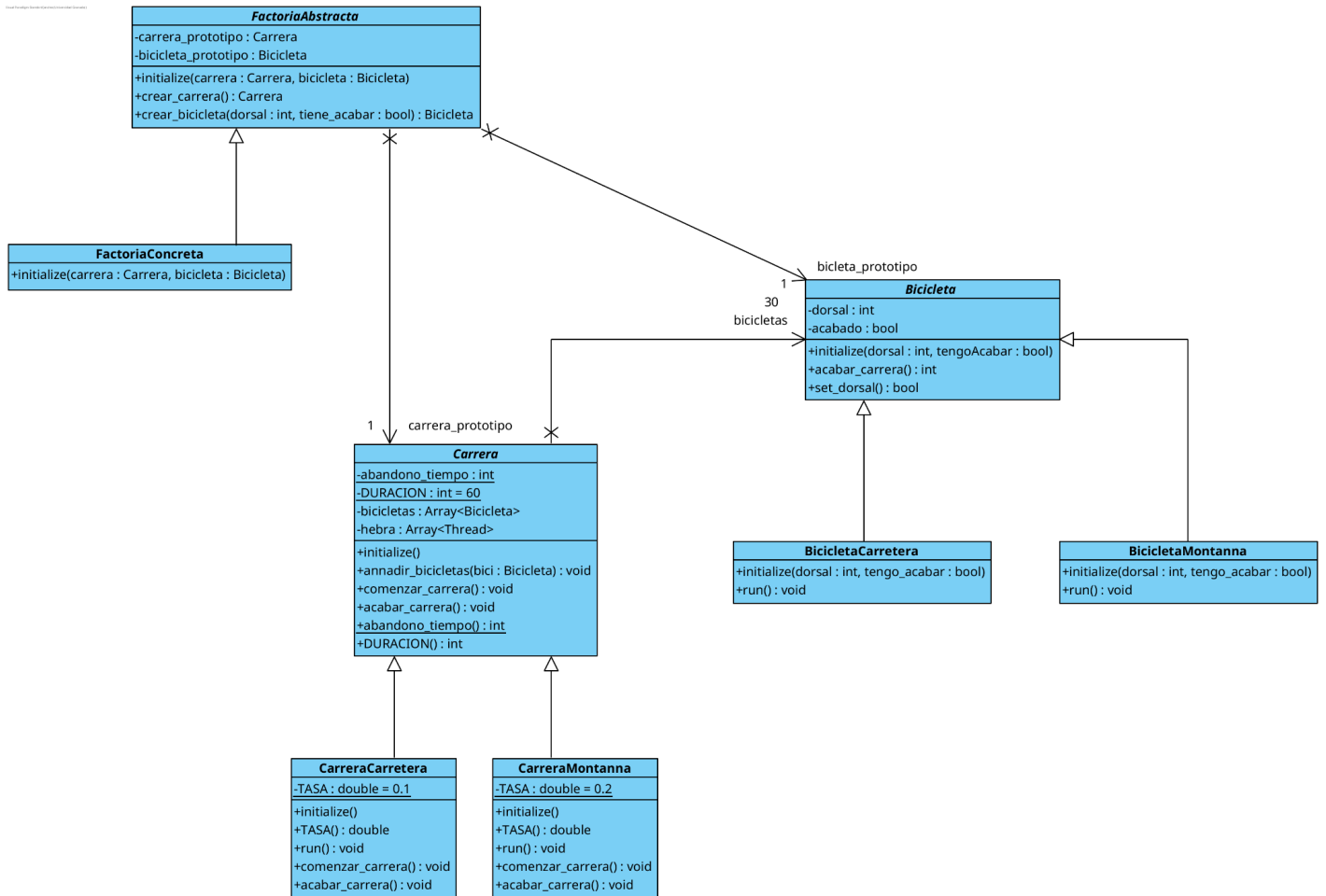
Pese al reparto del trabajo inicial, debemos remarcar que sobre todo en esta sesión colaboramos activamente entre todos para hacer cada módulo debido a su mayor complejidad. Así pudimos trabajar de forma más eficaz.

Módulo 1:

- Java:

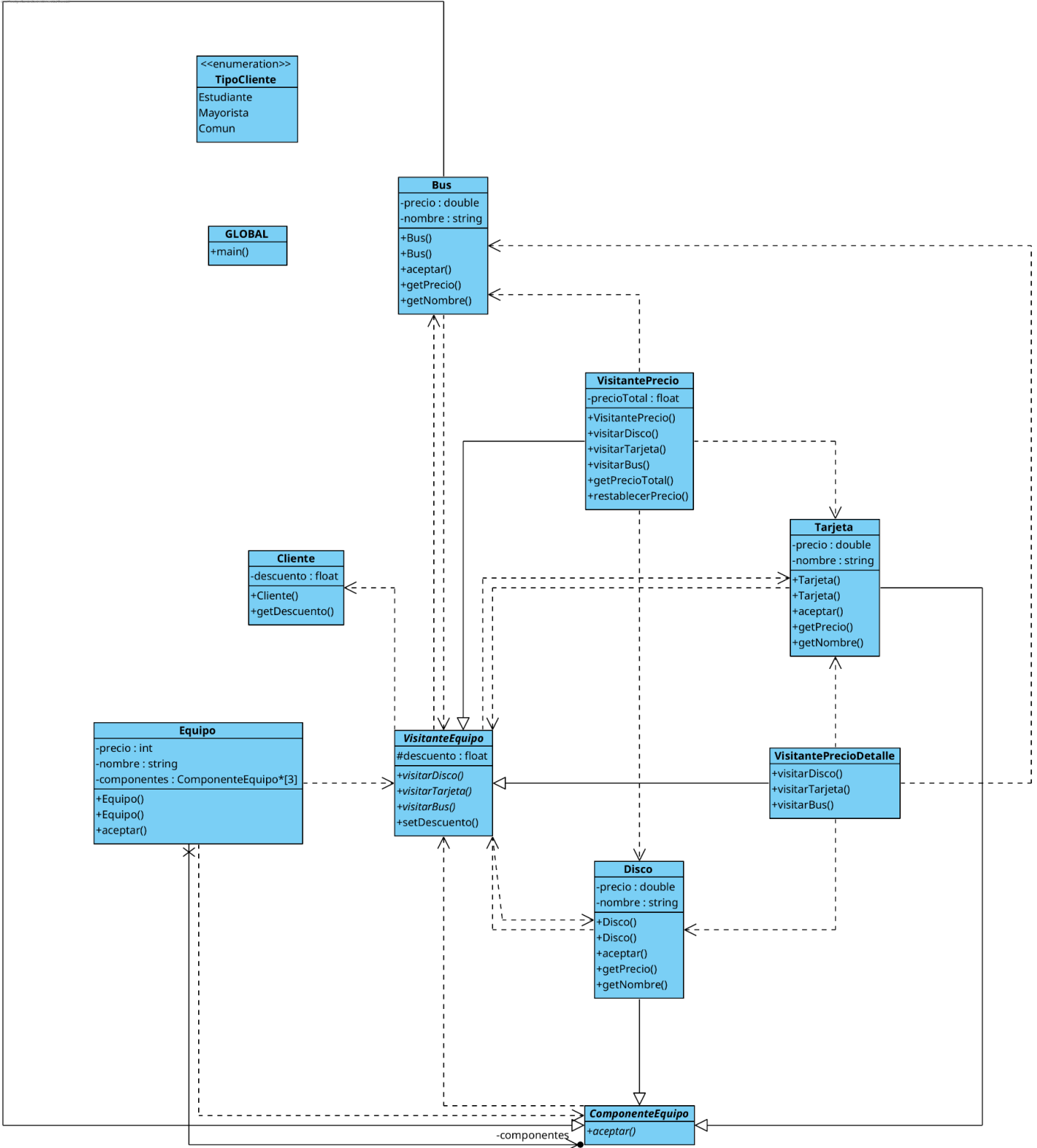


- Ruby:



Módulo 2:

Para la parte extra, simplemente creamos una clase nueva que eligiese el descuento en función del tipo del cliente que fuese (mediante un enumerado). Como cambio en el patrón, lo único que se implementa nuevo en el patrón es la modificación de los métodos que calculan los distintos precios en cada visitante y la incorporación del atributo “descuento” para tener en cuenta dicha rebaja de precio. El descuento se aplica desde el main, con un nuevo método en la clase “VisitanteEquipo” llamado “setDescuento()”, al que se le pasa el cliente para almacenar su descuento



Sesión 3:

Patrón elegido:

Para desarrollar el SCACV y mejorar la GUI hemos optado por seguir el patrón MVC (Modelo-Vista-Controlador) para desacoplar la parte visual (la vista) de la parte de la lógica (el modelo) de la que intercambia información con estas dos (el controlador).

Debido a que hemos sido 3 como hemos mencionado anteriormente, nos ha faltado algo de tiempo y no hemos podido ser capaces de pulir el código y la estructura como hubiésemos deseado.

