



Departamento de Informática
Universidad Técnica Federico Santa María



Entregable III

Análisis y Diseño de Software / Fundamentos de Ingeniería de Software

Integrantes:

Nombre	Email	Teléfono
Nicolás Alarcón	nicolas.alarcon.14@sansano.usm.cl	(+569)65745692
Rodrigo Elicer	rodrigo.elicer.14@sansano.usm.cl	(+569)62402412
Gabriel Valenzuela	gabriel.valenzuel.14@sansano.usm.cl	(+569)84459457

1- Listado de requerimientos

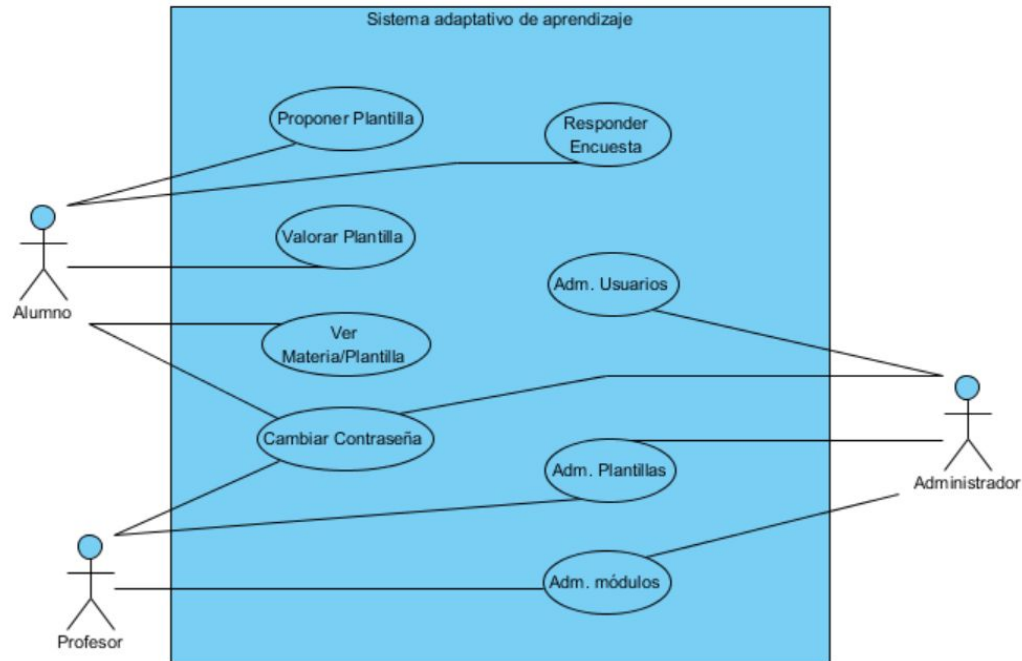
Requisitos funcionales:

ID	Requerimiento	Obligatoriedad
FR1	Hacer Login al Sistema	Obl.
FR2	Ver si encuesta está hecha	Obl.
FR3	Clasificar alumno según Encuesta	Obl.
FR4	Mostrar plantilla	Obl.
FR5	Responder Encuesta	Obl.
FR6	Proponer plantilla	Obl.
FR7	Valorar plantilla	Obl.
FR8	Administrar módulos (CRUD)	Obl.
FR9	Administrar usuarios(CRUD)	Obl.
FR10	Administrar plantilla (CRUD)	Obl.

Requisitos no-funcionales:

ID	Requerimiento	Obligatoriedad
NFR1	Ocultar información personal	Obl.
NFR2	Que sea escalable	Deseable
NFR3	Que el sistema sea capaz de soportar a todos los alumnos inscritos simultáneamente	Deseable
NFR4	El sistema debe poseer una interfaz gráfica ad-hoc	Deseable

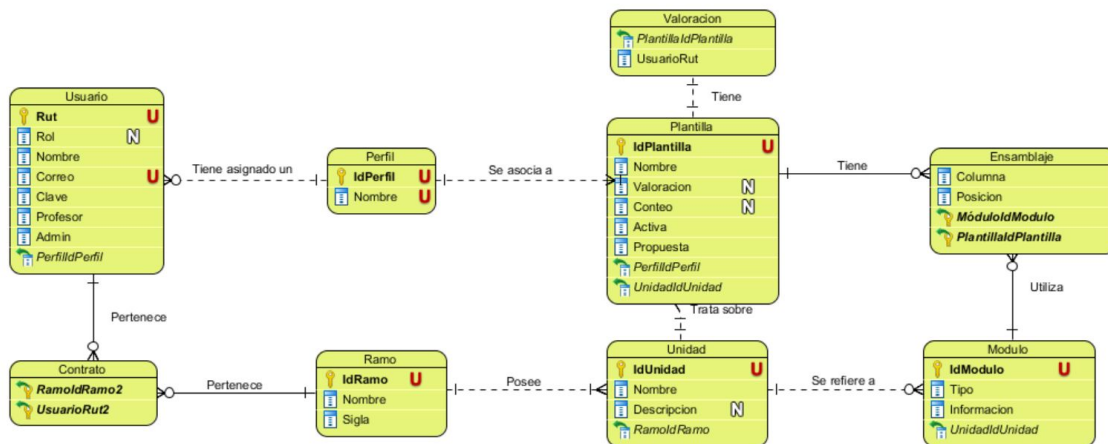
2- Casos de uso



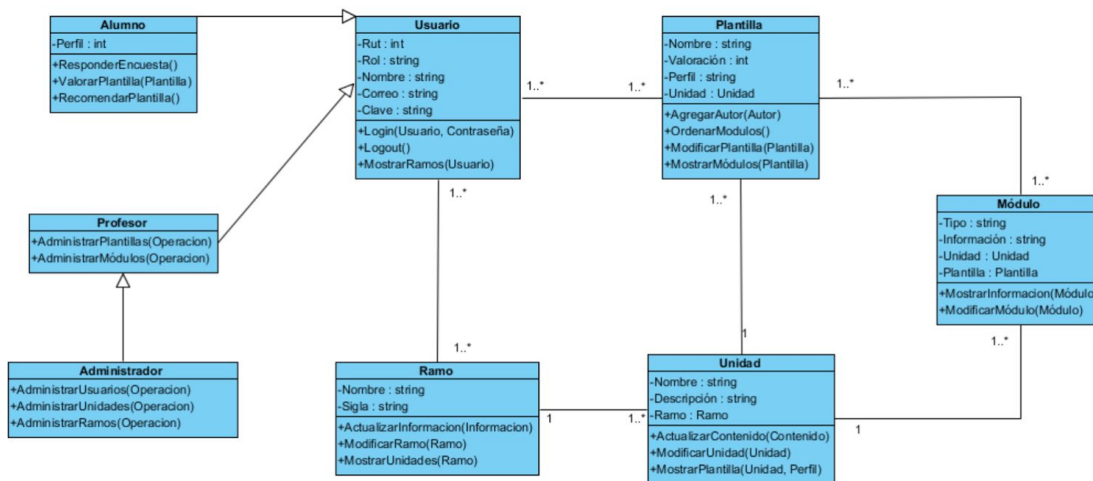
3- Diagramas de secuencia del sistema

Debido a la gran cantidad de diagramas, se presentarán sólo en el archivo .vpp

4- Modelo relacional de la base de datos



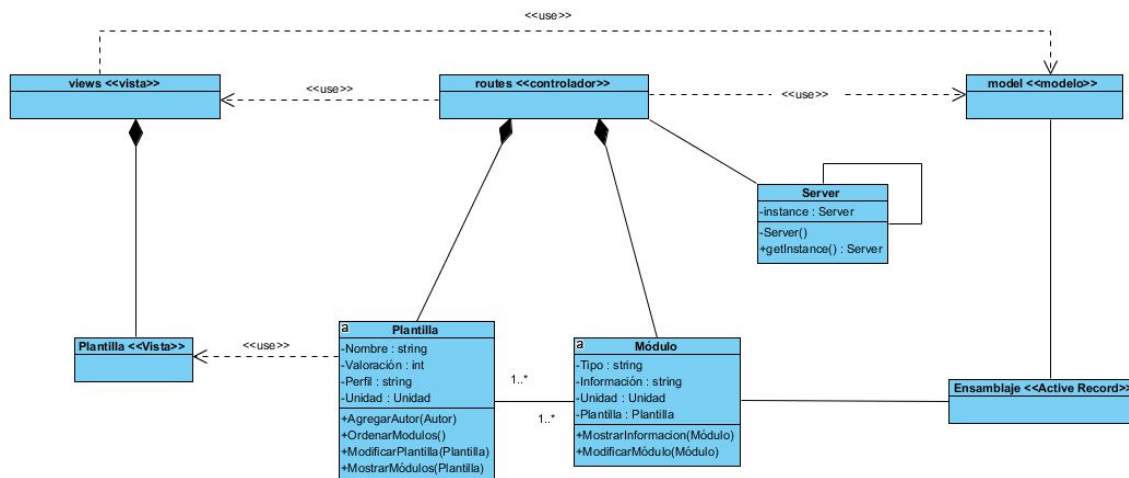
5- Modelo de clases



6- Diagrama de secuencia de componentes del sistema

Similar al punto 3.-, los Diagramas se encuentran en el archivo .vpp para no saturar de imágenes el informe. Los requerimientos críticos elegidos fueron **Ver Materia/Plantilla**, **Responder Encuesta** y **Proponer Plantilla**

7- Bosquejo MVC



La organización MVC utilizada fue la de *Page Controller* dado que para cada vista se hizo uno o más controladores con las cuales se interactúa, habiendo mínimo proceso lógico dentro de estas. Además, se incluye en una de las interacciones el patrón MVC *Active Record*, ya que la escena dada por plantilla y módulo utiliza de forma activa parte de la base de datos, por lo que se prefirió tener tal sección del modelo en un registro como una fuente accesible y modificable en cualquier momento de su utilización. Finalmente, se implementó el patrón de diseño *Singleton* para disminuir la cantidad de instanciaciones de distintos elementos utilizados a lo largo de la plataforma. Estos solo se instancian en el servidor y se utilizan constantemente enviados como parámetro hacia el controlador *routes*, el cual lo envía a todos los controladores que los fuesen a utilizar.

8- Lecciones aprendidas

Realizando el trabajo nos dimos cuenta como grupo que un proyecto de esta magnitud puede variar de diversas formas, por lo que es conveniente tener un código adaptable y flexible. También sirvió tener constante comunicación con los interesados en el proyecto (en este caso el profesor, los ayudantes, entre otros), además de la comunicación dentro del mismo grupo.

Debido a lo anterior es que GitHub nos fue muy útil al momento de escribir código en distintas partes sin tener que preocuparse de qué hizo el otro exactamente. Por otro lado, encontramos que la plataforma Trello no era demasiado necesaria, ya que GitHub tiene un sistema parecido para anotar tareas y asignarlas llamada "Projects".

Entregable 3

Esto hubiese sido bastante útil ya que GitHub es una plataforma que está creciendo cada vez más por lo que tener conocimiento de dicha funcionalidad hubiese sido un mayor aporte que Trello.

Uno de los mayores problemas que tuvimos fue el comienzo, en donde al no saber ninguno sobre cómo hacer una página web, nos costó mucho comunicar bien lo que íbamos haciendo, por lo que algunos hicieron cosas que otros no supieron hasta tiempo después. Sin embargo una vez ya teniendo las bases, nos resultó mucho más fácil el poder trabajar en el proyecto.