



Departamento de Informática
Universidad Técnica Federico Santa María



Requisitos de Software

Proyecto: Control Parental Web (CPW)

Integrantes:

Nombres y Apellidos	Email	ROL USM
Alejandra López B.	alejandra.lopez.13@sansano.usm.cl	201373601-7
Paula Marianjel A.	paula.marianjel.13@sansano.usm.cl	201373501-0
Marcom Moyano Ó.	marcom.moyano.13@sansano.usm.cl	201373582-7

Índice

Desarrollo del Prototipo	4
Patrones de Diseño	5
Diagrama de Clases	6
Diagramas de Secuencia	7
Análisis de Trade-off	7

Objetivo

El objetivo de este entregable es mostrar el avance realizado en relación al entregable anterior e incluir el feedback del Cliente y los ayudantes.

Contenido del Informe a Entregar

En este entregable señalaremos cómo se lleva a cabo la realización del software, las herramientas utilizadas y los modelos aprendidos en clases, que son necesarios para el correcto desarrollo de este.

Desarrollo del Prototipo

El primer prototipo funcional del proyecto se encuentra en el repositorio de GitHub. Para este se desarrolló la funcionalidad de mayor prioridad dentro del proyecto, la cual está marcada con rojo en el modelo de casos de uso.

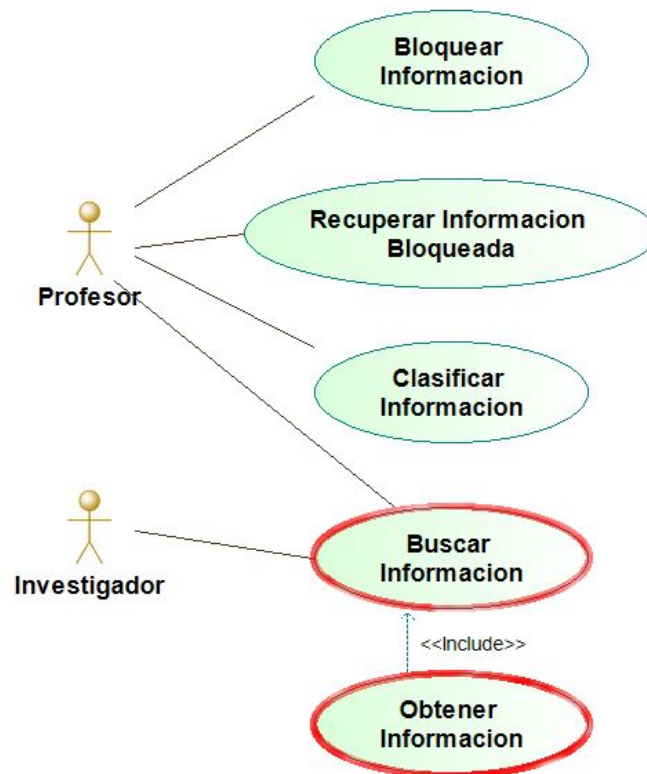


Figura 1: Modelo de Casos de Uso 1.2, el cual tiene marcadas las funcionalidades realizadas en el prototipo.

La capacidad de buscar información, la cual incluye obtener la información de Internet, es la funcionalidad que se realizó primero ya que esta es la base de todo el proyecto y la cual se debía terminar lo antes posible para poder continuar con los otros requisitos del cliente.

Para realizar este primer paso, se decidió utilizar el framework Rails (versión 4.2.6) que utiliza el lenguaje de programación Ruby (versión 2.3.4).

Patrones de Diseño

Los patrones de diseño utilizados para realizar el software son Active Record y MVC, patrones que vienen incluidos en el framework escogido Rails.

Intención	Patrón de Diseño	Razonamiento
Facilitar la forma de visualización de los componentes del software, haciendo más fácil el desarrollo y también encontrar los errores que puedan surgir.	MVC	Al desarrollar el software entre más de 1 persona y no estar siempre juntos, es necesario tener un orden al desarrollar. Esto lo entrega el patrón MVC.
Facilitar la creación de de modelos y manejo de los registros existentes al clasificar los contenidos encontrados con el buscador.	Active Record	Para utilizar el patrón de diseño MVC, uno de los mejores patrones para el correcto manejo de la base de datos (M en MVC) es Active Record.
Se quiere dar aviso de que un resultado de la búsqueda fue clasificado o bloqueado.	Observer	Cómo se desea notificar que se realizó alguna acción sobre un resultado (clasificar o bloquear) y además agregarlo a su correspondiente modelo, el patrón que más se adecua es Observer.

Diagrama de Clases

El diagrama de clases obtenido y refinado por el grupo es el siguiente:

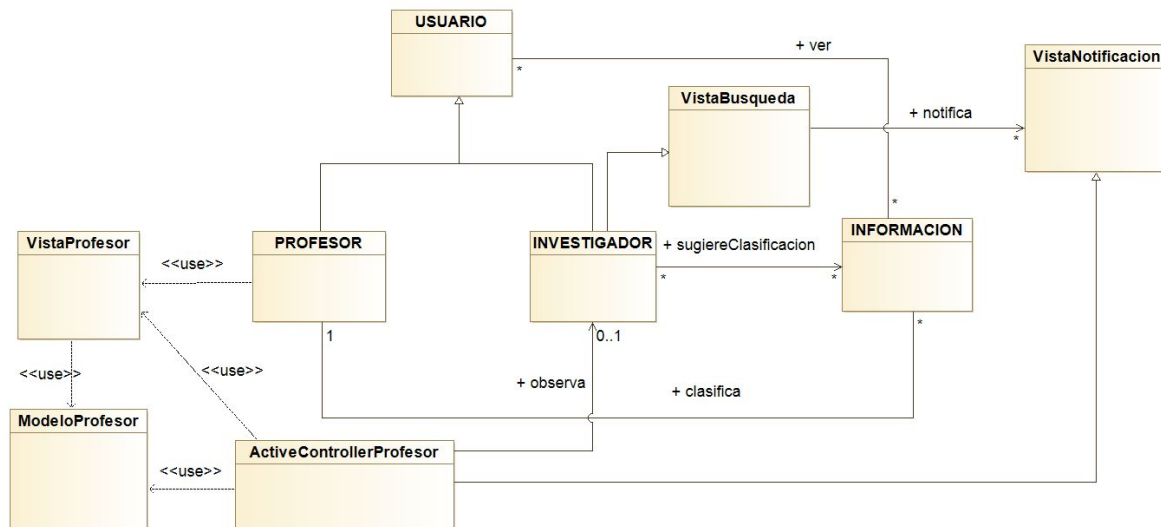


Figura 2: Diagrama de clases inicial incluyendo los patrones de diseño.

En este se pueden observar las clases principales, que fueron heredadas del modelo de dominio, de Profesor, Investigador e Información. Lo principal que se quiere mostrar son las facultades de cada actor, por ejemplo: se quiere mostrar que el profesor puede clasificar resultados de la búsqueda, mientras que el investigador es un usuario que sólo puede ver información y, en caso de querer clasificarla, debe sugerirle al Profesor para que este decida si incluirla en la clasificación o no. Además, cabe destacar que el patrón MVC incluye al patrón Active Record para tener un mejor manejo de la base de datos, por lo que también se ve reflejado en el Diagrama de Clases.

Por otro lado, el patrón de diseño observer se ve reflejado en la asociación entre las clases ActionControllerProfesor e Investigador. Este observador es el que notifica al Profesor cuando se ha realizado una sugerencia, así este puede clasificarla de manera más sencilla.

Diagramas de Secuencia

Los Casos de Uso relevantes seleccionados fueron *buscar información* y *clasificar información*.

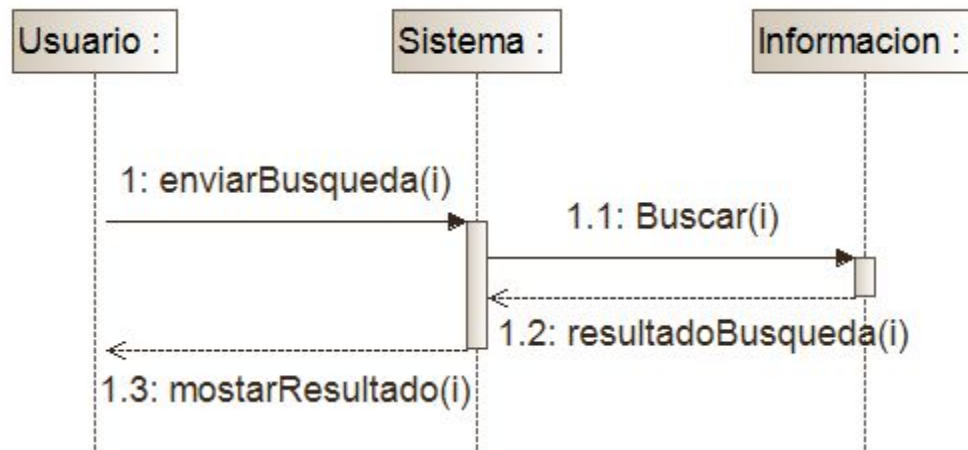


Figura 3: Diagrama de secuencia del caso de uso buscar información.

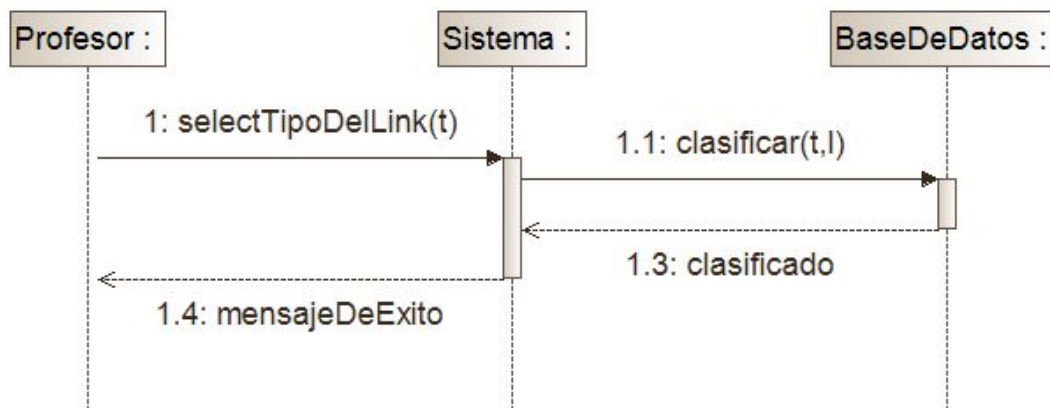


Figura 4: Diagrama de secuencia del caso de uso clasificar información.

Análisis de Trade-off

Luego del feedback realizado por el cliente se identifica una nueva funcionalidad, la cual es “Agregar una forma de que los Usuarios puedan recomendar páginas al Profesor”. De lo anterior:

1. Se obtiene la siguiente pregunta (Question) *¿Qué cambios se deben realizar en el Sistema para agregar una forma para que los usuarios puedan recomendar páginas?*
2. A continuación, se plantean las opciones (Options)
 - O1: Agregar nueva vista donde se puedan sugerir páginas.
 - O2: Agregar botón de “sugerir” en cada resultado de la búsqueda.
3. Ahora se define el criterio (Criteria) con el cual se juzgarán las opciones
 - C1: Confiabilidad
 - C2: Mantenibilidad
 - C3: Escalabilidad
 - C4: Rendimiento
4. Finalmente, el análisis (Trade-Off) para asignar una correlación a cada criterio a partir de las opciones. El puntaje está definido de la siguiente manera: ++ (Muy fuerte)/+ (Fuerte)/0 (No disponible)/-(Baja)/--(Muy baja). En análisis quedaría de esta forma:

Criterio\Opciones	O1	O2
C1	+	+
C2	+	++
C3	-	+
C4	-	++

Como resultado final, se detalla el resultado obtenido de la tabla. Los resultados serían:

- *La opción 1 (O1) no tiene grandes impactos con respecto a Confiabilidad y Mantenibilidad. No obstante, impacta negativamente la Escalabilidad del Sistema y el Rendimiento.*
- *La opción 2 (O2) favorece la en la mayoría de los criterios, siendo más fuerte en la Mantenibilidad y el Rendimiento, siendo este último uno de los más*

importantes mencionados por el cliente.

En conclusión, la opción 2 es la opción seleccionada por el equipo, ya que, se favorece bastante la Mantenibilidad y Rendimiento (entre otros criterios), que son justamente atributos de calidad que el cliente pide con mayor énfasis en este proyecto, debido a que no desea largos periodos de espera al estar realizando una búsqueda ni tardar tanto en encontrar lo que necesita.