



Departamento de Informática
Universidad Técnica Federico Santa María



Requisitos de Software

Proyecto: The Adaptive Learning Wiki

Integrantes:

Nombres y Apellidos	Email	ROL USM
Yerko Cuzmar Ortiz	yerko.cuzmar.14@sansano.usm.cl	201473594-4
Felipe Montero Díaz	felipe.montero.13@sansano.usm.cl	201356565-4
Cristian Vallejos Pérez	cristian.vallejos.14@sansano.usm.cl	201473551-0

Desarrollo del Prototipo

El prototipo desarrollado se encuentra alojado en la siguiente dirección web:

<https://github.com/cristian-vallejos/TheAdaptiveLearningWiki/tree/master>

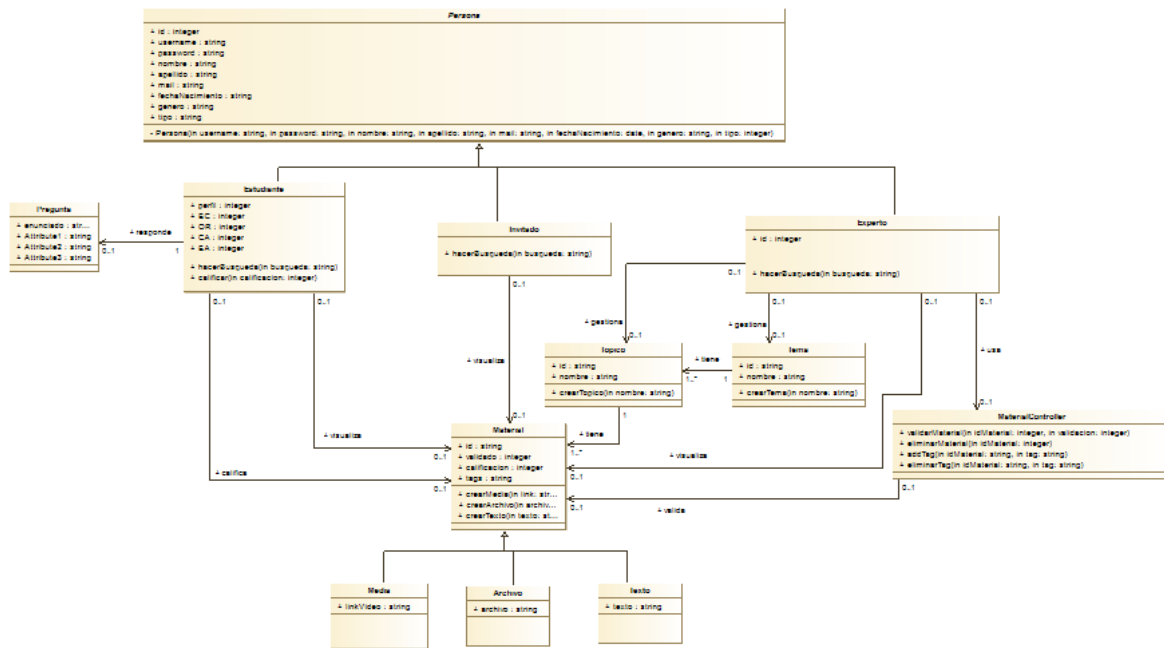
Cabe destacar que dicha dirección no es la misma en donde se alojan los informes y modelos de cada entregable. Esta metodología de orden y estructuración se consideró como la más apropiada en el propósito de brindar mayor autonomía tanto al diseño como al desarrollo del proyecto del presente semestre.

Selección de Patrones de Diseño

Intención	Patrón de Diseño	Razonamiento
Se desea mostrar en el diagrama de clases la interacción entre los componentes del sistema, mediante el uso de controladores que comuniquen la vista final y el modelo.	MVC	Las clases del dominio deben poder comunicarse entre sí de manera adecuada, al mismo tiempo que el usuario activo debe poder buscar/visualizar Contenido y acceder a los distintos recursos de la plataforma. MVC es práctico en este sentido, pues permite acceder fácilmente a los distintos objetos de la aplicación.
Se busca esconder las interacciones complejas del sistema detrás de métodos que simplifiquen dicha interacción	Facade	La búsqueda de información (<i>por parte del sistema</i>) según los parámetros brindados por el experto, debe verse simple para este último, de modo que la verdadera complejidad del sistema de búsqueda -y el sub-sistema de filtrado- debe ser imperceptible para quien gestiona la validación de contenidos en la aplicación web.

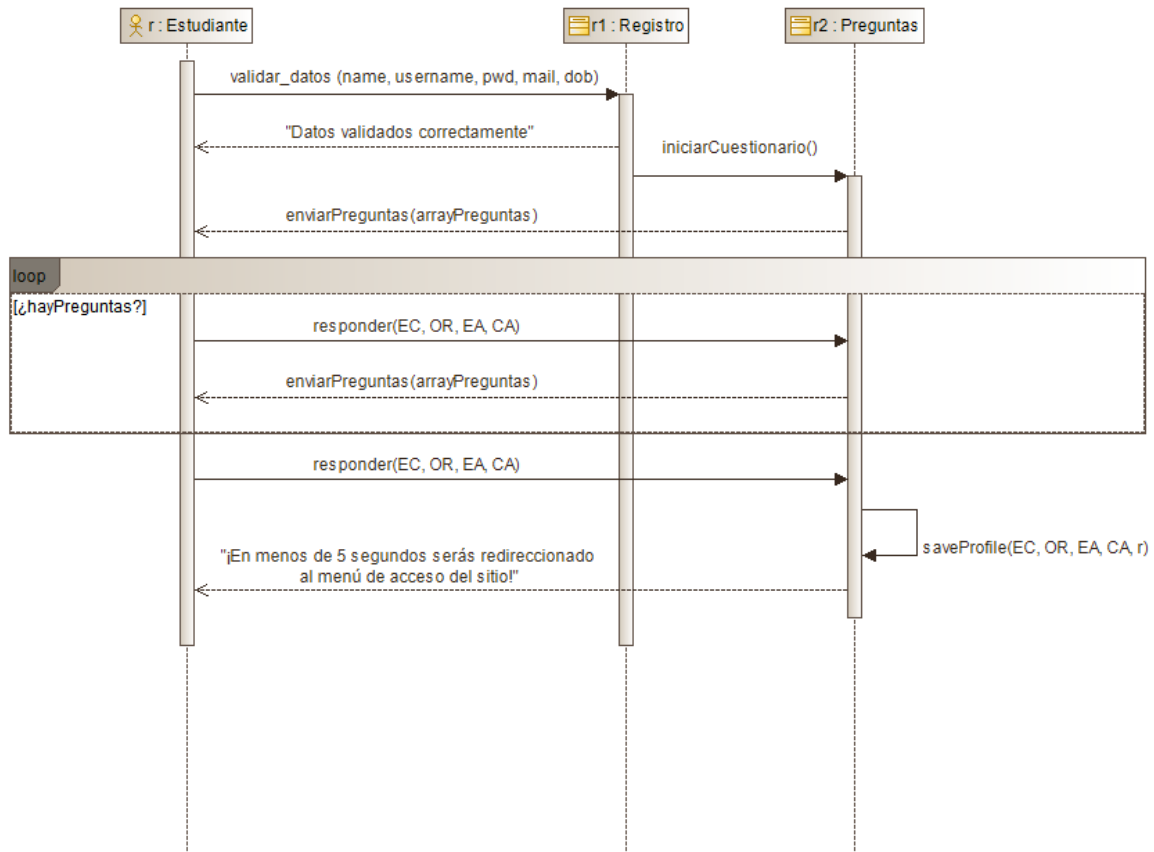
Se busca mostrar en el diagrama de clases el manejo de los requests de autenticación de usuarios en el sistema	Front Controller	La necesidad de resolver las peticiones de autenticación de usuarios es clave en la plataforma. Además, es fundamental poder delegar ciertas solicitudes que deben ser resueltas por otros controladores, lo cual se ve facilitado por la inclusión de este patrón de diseño.
--	------------------	---

Creación de Diagrama de Clases

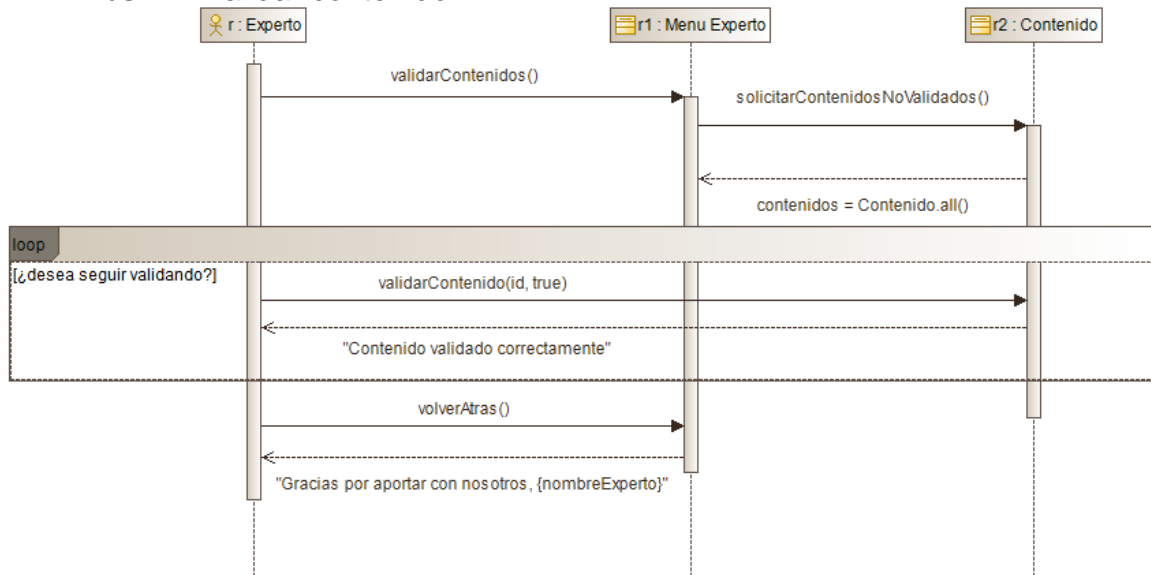


Diagramas de Secuencia

- DdS1 → Responder test de Kolb:



- Dds2 → Validar contenido:



Análisis de Trade-off

Dado que no fue posible recibir feedback del cliente para este entregable, nos es imposible realizar un análisis de trade-off basado en hechos reales. Es por esto que para esta sección consideraremos una funcionalidad ficticia a la cual realizaremos el análisis de trade-off correspondiente.

La nueva funcionalidad mencionada por el cliente y que no fue considerada con anterioridad consiste en que el sistema debe ser capaz de recopilar los resultados de búsqueda para cada usuario, y realizar recomendaciones de contenido en relación al tiempo y cantidad de búsquedas recientes.

De esto nace la pregunta, ¿Qué cambios es necesario realizar para implementar esta funcionalidad?

Debido al volumen de información necesaria a guardar, las opciones requeridas si o si deberán implementar algún tipo de servidor dedicado. Debido a esto, se pensó en lo siguiente:

O1: Implementar servidores propios para almacenar la información necesaria.

O2: Delegar esta información a un Cloud Service.

Los criterios escogidos fueron los siguientes:

C1: Confiabilidad

C2: Mantenibilidad

C3: Rendimiento

C4: Costo de Implementación

Finalmente, realizamos el análisis de trade-off:

Criterios/Opciones	O1	O2
C1	+	-
C2	+	0
C3	++	--
C4	--	+

De lo anterior, es posible concluir:

- La opción 1 es la más confiable, debido a que no depende de un servicio externo conectado a la red.
- Así mismo, es la que ofrece un mejor rendimiento, debido a las mismas razones.
- La opción 2 no influye en nada en la mantenibilidad del sistema. Sin embargo, la opción 1 si afecta y positivamente, debido a las ventajas de poder acceder de forma local al hardware y datos que contiene el sistema.
- Finalmente, la opción 1 es la que posee mayor plus. Sin embargo, debido al costo de implementar un servidor que maneje los datos, la opción 2 es la más viable económicamente