



**Departamento de Informática**  
Universidad Técnica Federico Santa María



## Entregable IV

Proyecto: Scientific Adaptative Searching Platform (SASP)

Integrantes:

Nombres y Apellidos	Email	ROL USM
Pablo Alberto Aguirre Moreau	pablo.aguirre.14@sansano.usm.cl	201473555-3
Sebastian Ignacio Muñoz Godoy	sebastian.munoz.14@sansano.usm.cl	201473503-0
Jorge Ignacio Aliste Ahumada	jorge.aliste.14@sansano.usm.cl	201473538-3

## Post-Mortem Metodológico



### 1. Fortalezas

Los ámbitos en los que más fortaleza tiene el equipo es en las técnicas de desarrollo , equipo de trabajo y la dinámica del equipo. Esto ocurre porque como equipo dimos más énfasis a estos aspectos ya que el poco tiempo para desarrollar durante el semestre hizo necesaria plantearnos cómo desarrollar una buena dinámica de trabajo en equipo y descubrir las habilidades de cada integrante del equipo junto con la comunicación entre todos. Por otro lado, como el cliente era un cliente más bien ausente y con requisitos poco claros y cambiantes se le dio importancia a las técnicas de desarrollo.

### 2. Debilidades

Las mayores debilidades se encontraron en liderazgo, espacio de trabajo y cliente. Como equipo asociamos a las pocos tiempos disponibles de cada integrante estas deficiencias, ya que no hubo un líder claro en el grupo que facilitara la comunicación ni la organización, más bien cada uno realizaba un avance en lo que podía bajo sus propios tiempos disponibles. Además, no hubieron espacios claros de trabajo, se tendió a trabajar cada uno por su cuenta en el sitio donde tuviera disponible. Como grupo realizamos pocas reuniones para realizar acuerdos, y estas eran regularmente cercanas a las entregas. Finalmente, nuestro cliente dificultó ponernos de acuerdo o tener contacto con él, por lo que como grupo terminamos pensando más en que le

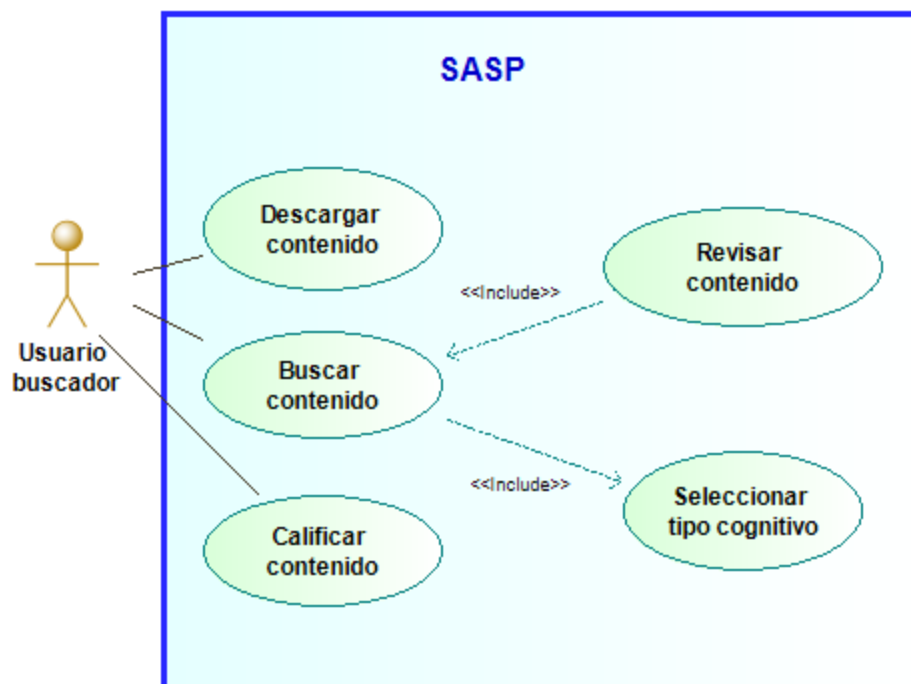
gustaría o que querría sin darle mayor importancia a tener reuniones periódicas con él , ya que su predisposición para atención era muy baja.

### 3. Propuesta de mejoras

Si se continúa el desarrollo el proyecto , proponemos las siguientes propuestas de mejoras:

- **Espacio de trabajos y Reuniones:** Se declararán desde el comienzo espacios para trabajo en conjunto al menos cada dos semana, los cuales comenzarán con una reunión para discutir avances , defectos y tareas pendientes.
- **Producto y Proceso:** Se evaluaría la forma de implementar un sistema de automatización de testing, para probar integralmente el software y reconocer defectos a medida que se cambien o incorporen cosas. Además, se haría un cronograma con fechas estimadas de realización de tareas juntos con la especificación de su prioridad.

## Diagrama de Casos de Uso (final)



El diagramas de casos de uso se mantuvo a lo largo del desarrollo del sistema SASP debido a que captó los requisitos que el cliente quería para el sistema. Los casos de uso finales satisfacen los siguientes requisitos funcionales especificados y/o extraídos

de la información que brindó el cliente:

Req. funcional	Descripción y medición
Seleccionar tipo cognitivo	Permitir elegir el tipo cognitivo de la búsqueda.
Filtrar Contenido	Mostrar sólo los contenidos adecuados al tipo cognitivo del usuario.
Buscar Contenido	Mostrar enlaces a los contenidos buscados.
Restablecer calificación.	El administrador puede restablecer la calificación de un enlace si esta no es adecuada.
Calificar contenido	El usuario podrá calificar el contenido.
Ordenar lista enlaces	El usuario podrá ordenar la lista de enlaces según calificación.

Al final del proyecto se cumple el objetivo ya que se tienen implementadas todas las funcionalidades especificadas por el cliente en los requisitos funcionales.

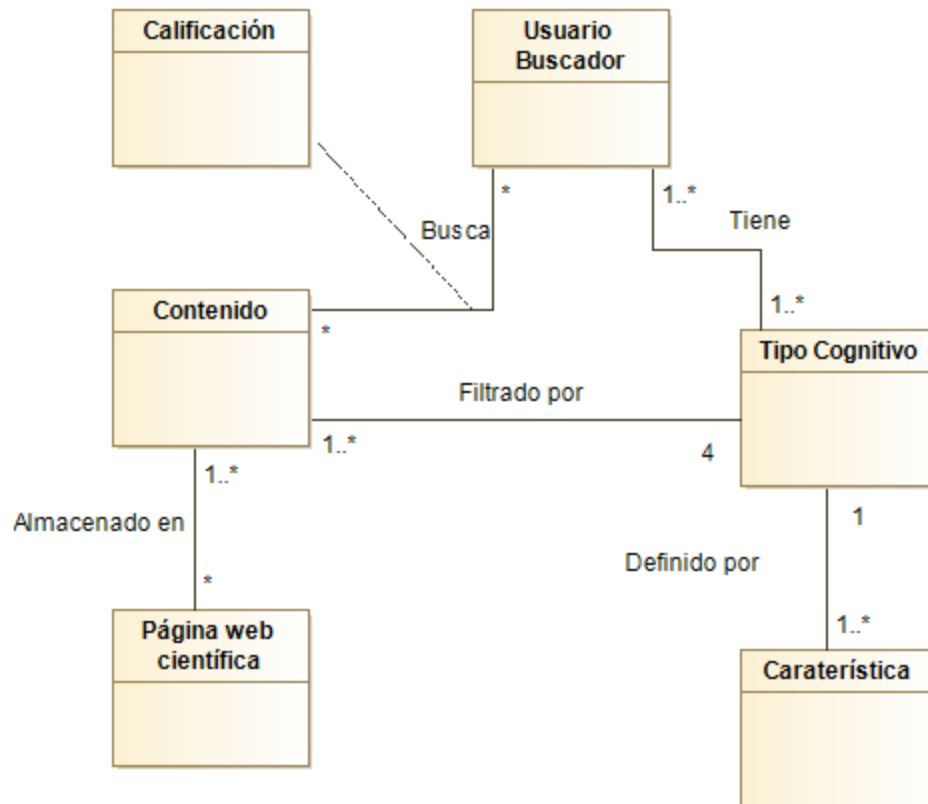
## Patrones de diseño y Frameworks (final)

Intención	Patrón de Diseño	Razonamiento
En el Diagrama de Clases se desea mostrar el proceso de una búsqueda realizada exitosamente y posterior evaluación del enlace visitado, además de cómo interactúan las distintas partes del sistema.	MVC	<p>Este patrón de diseño nos permite administrar de manera más expedita nuestro código, a la vez que las funciones que ejecuta cada parte de este. Utilizando MVC podemos realizar separaciones visibles, dividiendo el Modelo, parte del sistema que interactúa con la base de datos que almacena y recupera las calificaciones y enlaces, las vistas, código HTML que genera la parte visual con la que interactúa el usuario del portal web, y el controlador, la capa de que sirve de enlace entre ambas, envía comandos al modelo para actualizar su estado, o sea, agregar y/o modificar opiniones y a la vista que corresponda para cambiar su presentación, o sea, reaccionar a las interacciones con usuarios, por ejemplo mostrar la lista de resultados tras una búsqueda, o abrir un enlace al ser seleccionado, no obstante no es el encargado de manipular datos o generar una salida.</p> <p>En resumen, la utilización de este patrón de diseño en nuestro caso se justifica con una mejor</p>

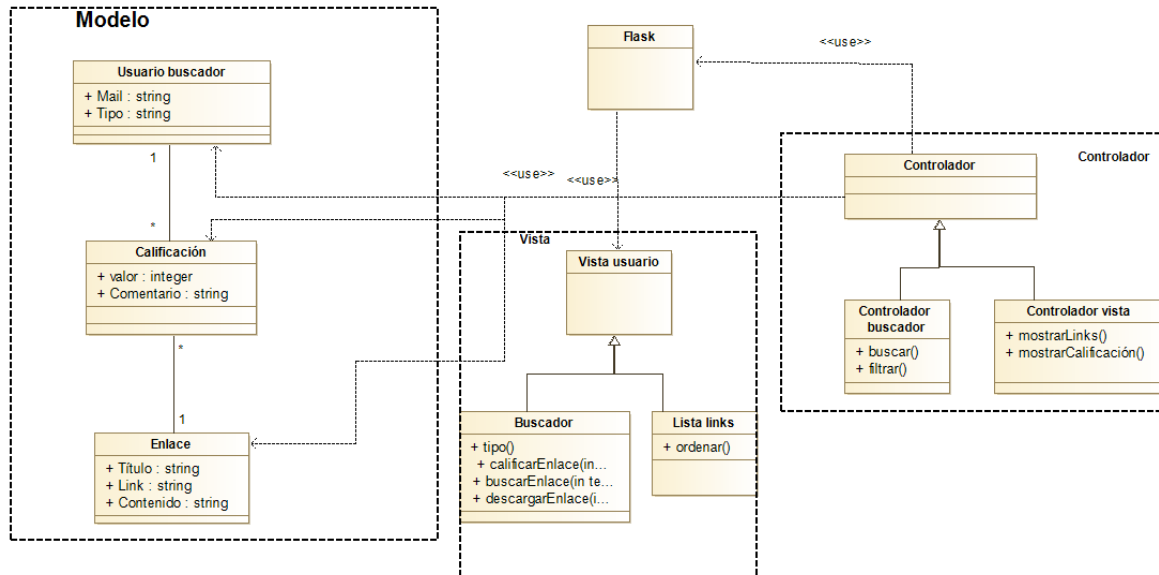
		modularización del trabajo, separar las responsabilidades mejorando las prácticas de desarrollo, abriendo posibilidades a escalabilidad y facilitar el trabajo de los desarrolladores.
Se desea desarrollar una plataforma web que se integre bien con el motor de búsqueda elegido y que cumpla con las expectativas del producto.	Flask	Este framework nos permite una mejor integración con la plataforma de Google ya que está basado en Python y al mismo tiempo nos permite tener un desarrollo más expedito debido al manejo que el equipo de trabajo posee con dicho lenguaje de programación. Por otra parte, contiene características orientadas al desarrollo de plataformas web, que es justamente el enfoque de nuestro producto.

## Modelo de Dominio y Diagrama de Clases (final)

El modelo de dominio se conserva el realizado en entregas anteriores debido a que a juicio de los desarrolladores representa el sistema que quiere el cliente.



Por otro lado, el diagrama de clases si cambió desde la entrega 1, debido a que hay patrones de diseño que no fueron utilizados. Esto ocurre por la modificación del software inicial que se pretende usar, al comienzo se esperaba utilizar Elastic Shearch, pero a medida que avanzó el proceso de desarrollo se comprendió que utilizar la plataforma de Google como base era más conveniente teniendo en cuenta los tiempos de trabajo disponibles.



## Pruebas de Software (actualización)

Defecto encontrado	Mitigación	Resultado obtenido	Observaciones
Buscar contenido no relacionado con la ciencia.	Para lograr que esto no ocurriera, se agregó un filtro para las búsquedas realizadas por la api de Google, agregando sólo páginas que permiten encontrar material de ciencia.	Efectivamente se obtienen resultados que solo tienen que ver con las páginas que están admitidas por la api de Google.	Con esta mitigación podemos dejar fuera a varias páginas que podrían ser una buena fuente de material de ciencia pero que no conocemos.
Descargar contenido y no la aceptar descarga.	El usuario debe acceder al enlace si es que quiere descargar el contenido.	Simplemente el usuario decide cuándo descargar lo que busca mediante el propio navegador por lo que es efectivo para descargar archivos.	No se pudo realizar una implementación directa en el sistema de las descargas (como un botón de descarga) pero se puede aprovechar que los navegadores actuales permiten

			la descarga de archivos como PDF.
Calificación errónea.	El método de calificación se ha modificado. Ahora para realizar una calificación el usuario debe indicar el enlace, su mail y la calificación (del 1 al 5) del respectivo enlace a calificar.	Ahora el usuario puede calificar cualquier enlace que se encuentre en la plataforma de una manera más simple.	El usuario debe tener presente que debe ingresar los datos correctamente.