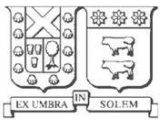

Requisitos de Software

Proyecto: FisFinder

Integrantes :

Nombres y Apellidos	Email	ROL USM
Erika Riveros Díaz	erika.riveros.14@sansano.usm.cl	201404533-6
Felipe Monsalve Cantín	felipe.monsalve.14@sansano.usm.cl	201473512-k
Francisco Vásquez Pinto	francisco.vasquez.14@sansano.usm.cl	201473568-5

1. Desarrollo del Prototipo

El desarrollo del prototipo se ha logrado en 4 etapas:

1. Planificación
2. Desarrollo del componente de búsqueda de contenidos
3. Desarrollo de la vista del usuario
4. Integración del sistema

1.1. Planificación

Durante la planificación se abordó el problema bajo diversas perspectivas dado que el sistema debe entregar información extraída de la WEB, razón por la cual existen diferentes formas de lograr este objetivo, pues se puede definir una selección específica de sitios, buscar sólo en uno de ellos o buscar en todo el espectro posible dicha información. Por lo anterior, se considera un factor de riesgo el cómo se obtendrá la información, dado que esta parte del sistema presenta una dificultad intrínseca en la búsqueda correcta de lo que el cliente necesita. De acuerdo con el método de desarrollo iterativo incremental, se consideró construir un prototipo en que se aprecie cómo se realizará la búsqueda, considerando únicamente un sitio donde buscar; y elaborar una interfaz de usuario tipo, con la cual el cliente pueda comprender la disposición en que se entregarán los resultados.

1.2. Desarrollo del componente de búsqueda

El desarrollo del componente que permite obtener los resultados de la búsqueda se logró mediante el uso de **arXiv** una base de datos de documentos académicos. En detalle, arXiv almacena una gran cantidad de papers científicos de áreas como física y matemática, existiendo documentos publicados desde 1996 hasta la fecha actual.

El funcionamiento del componente de búsqueda se ilustra a continuación, donde en primera instancia se ingresan las palabras de lo que se quiere buscar y se define la cantidad de resultados que se desean visualizar, esto último debido a que al contener la base de datos una cantidad enorme de datos se debe limitar el número de elementos extraídos para lograr una presentación eficiente de los mismos. Cabe destacar que, a pesar de lo anterior, al realizar dos o más búsquedas consecutivas ingresando las mismas palabras se obtienen idénticos resultados.

Se ingresan los parámetros de búsqueda:

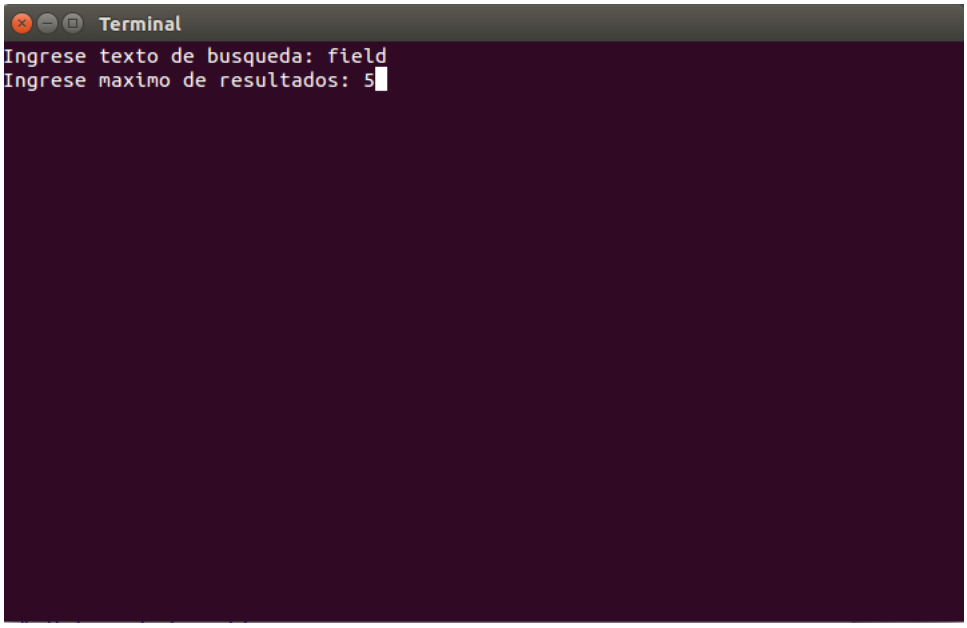

Figura 1: *Componente de búsqueda del prototipo: Ingreso de parámetros.*

Se obtienen los resultados de la búsqueda:

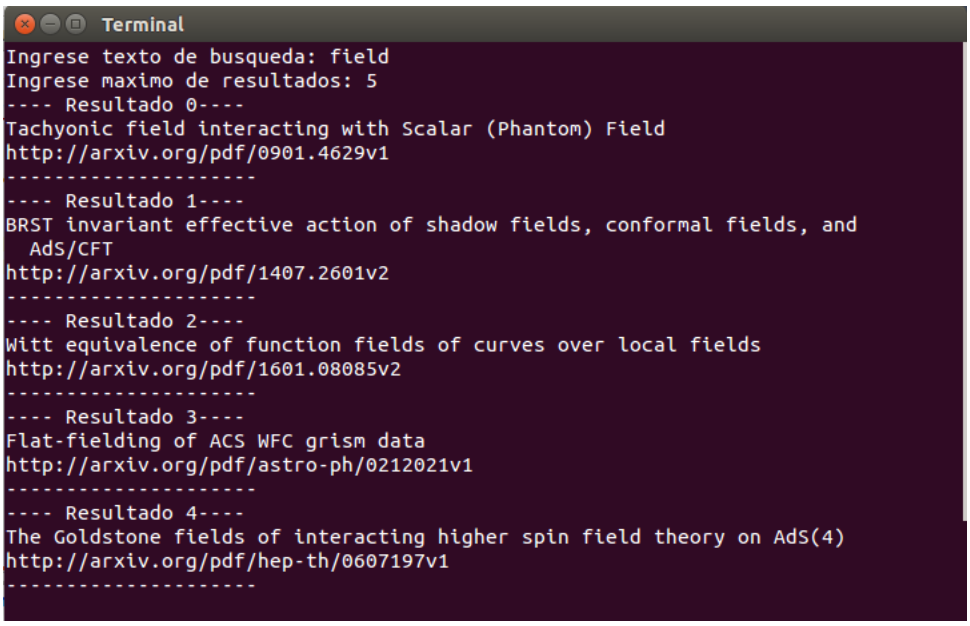

Figura 2: *Se visualizan los n resultados definidos al ingresar los parámetros.*

1.3. Desarrollo de la Interfaz del Usuario

Para la creación de la interfaz que utilizará el Usuario para realizar la búsqueda de contenidos se consideró un entorno sencillo en el cual pueda utilizar las herramientas necesarias para cumplir con su propósito. Cabe destacar que para la elaboración del prototipo se considera una etapa inicial de las características del sistema, razón por la cual la interfaz permite acceso limitado únicamente a la herramienta de búsqueda por palabras clave, sin posibilidad de seleccionar perfiles cognitivos para los cuales filtrar los resultados u otras herramientas.

Interfaz de Usuario:

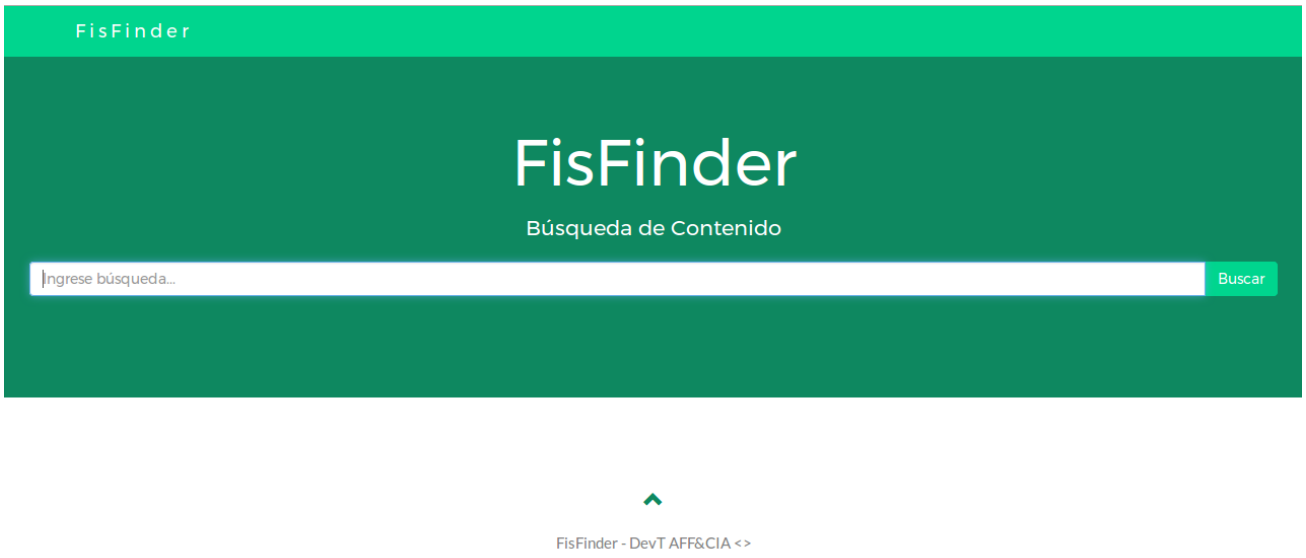

Figura 3: *Visualización en un Navegador WEB de la interfaz que se dispone para el uso del sistema por un Usuario.*

1.4. Integración del Sistema

Para la demostración del prototipo se integra el componente de búsqueda con la interfaz del Usuario con el fin de ilustrar el funcionamiento prelimianar del sistema. A continuación se muestra una secuencia de los pasos que se ejecutan para buscar un contenido y visualizarlo en un navegador WEB. Cabe destacar que para la realización de la demostración se fijó el número de resultados (a diez de estos), razón por la cual no se ingresa este parámetro.

Se ingresa en la barra de búsqueda lo que el Usuario necesita:

Figura 4: *Visualización en un Navegador WEB de la interfaz de Usuario, una vez ingresado lo que se desea buscar.*

Tras presionar la tecla enter o el botón Search de la interfaz se visualizan los resultados de la búsqueda:

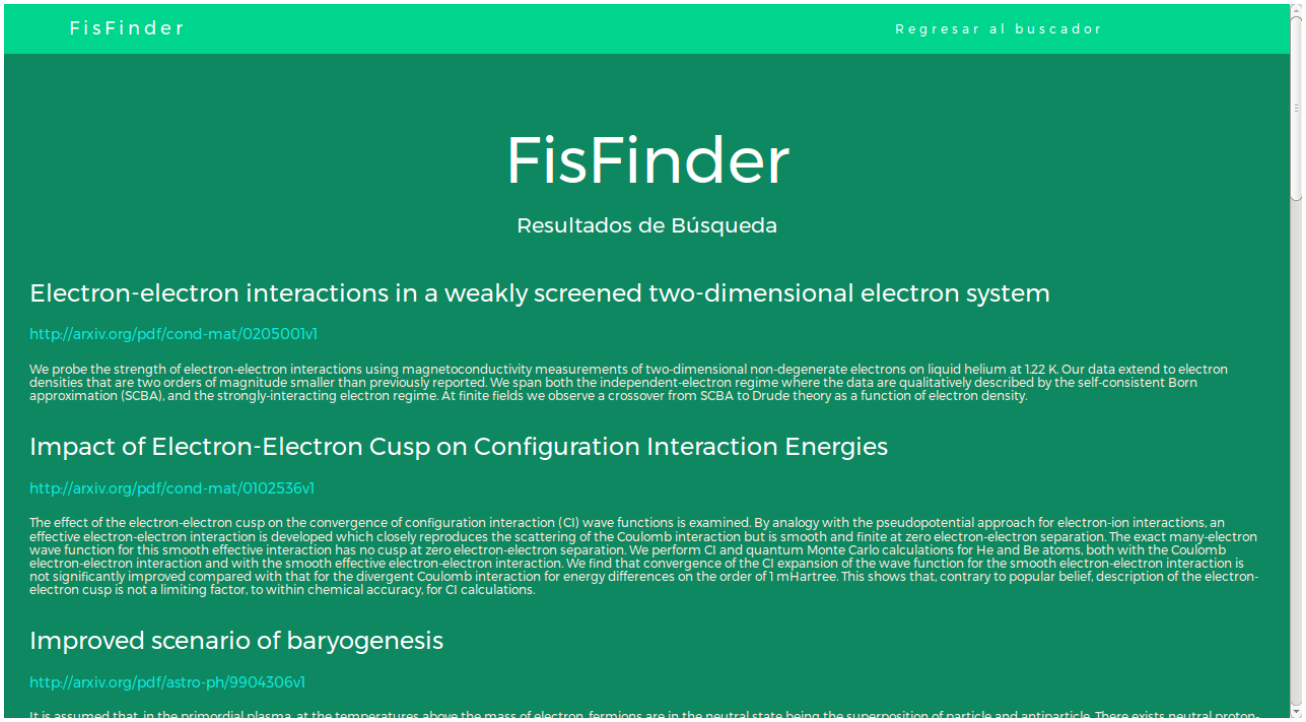

Figura 5: *Visualización en un Navegador WEB de la interfaz de Usuario una vez que se ha cargado la búsqueda. Se muestran los resultados hallados.*

Al posar el puntero sobre el título de un resultado se visualiza la opción de ver el contenido completo en formato PDF:

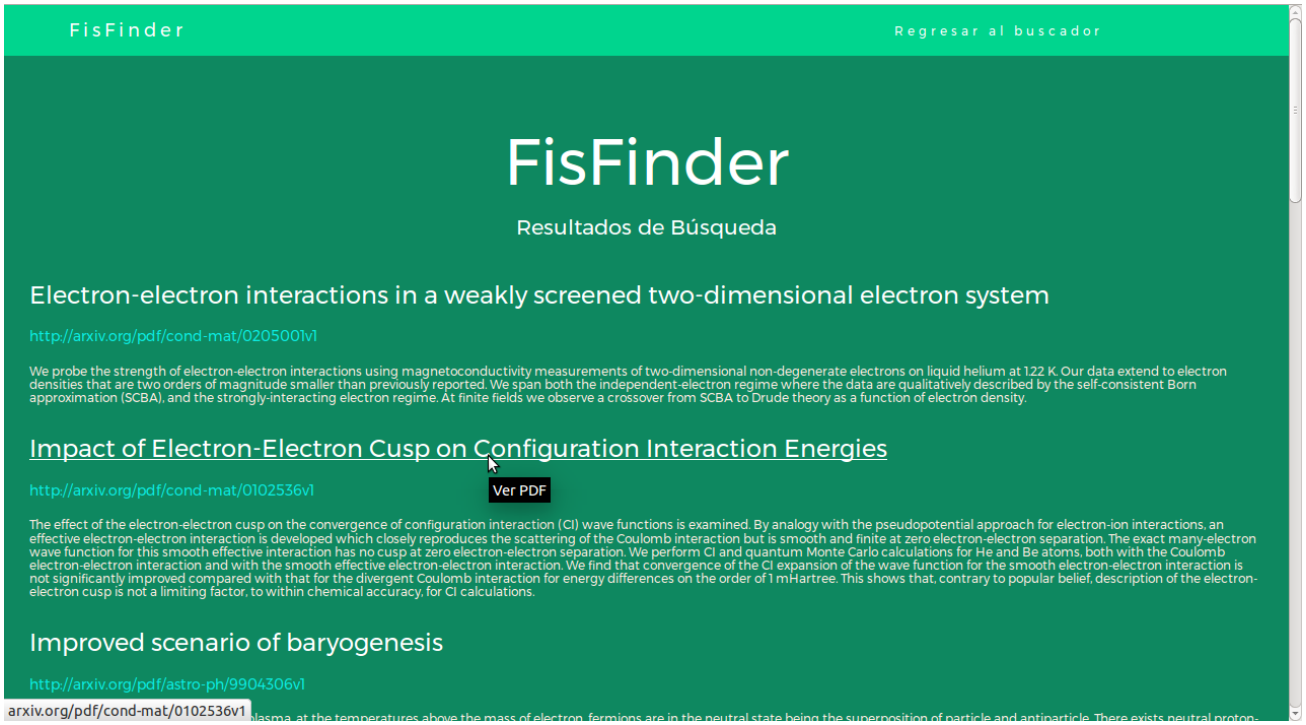

Figura 6: *Visualización en un Navegador WEB de la interfaz de Usuario, donde potencialmente se seleccionará un elemento para visualizarlo en PDF.*

Tras seleccionar un resultado, para ser visualizado en PDF, se redirige al usuario al visor de documentos, mostrando aquel que ha sido seleccionado:

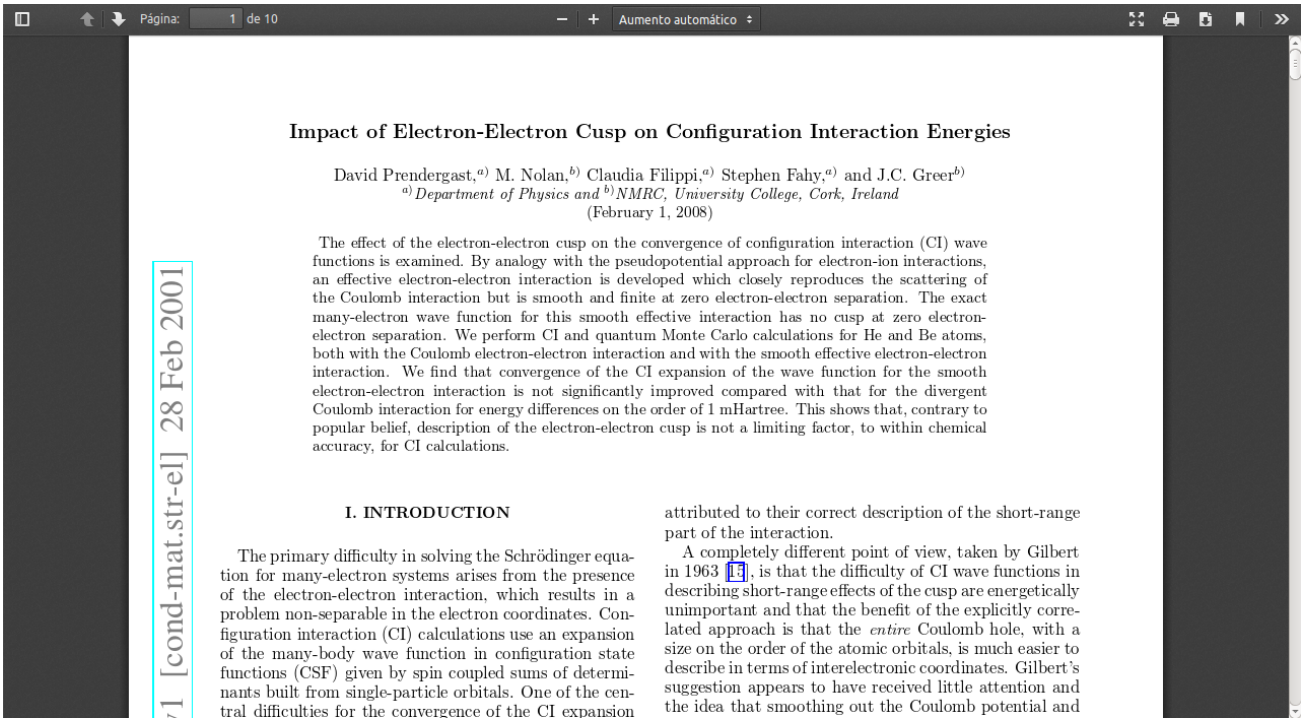

Figura 7: *Visualización en un Navegador WEB de un documento que ha sido seleccionado entre los resultados de una búsqueda realizada por el Usuario.*

2. Selección de Patrones de Diseño

Intención	Patrón de Diseño	Razonamiento
Se desea mostrar en el Diagrama de Clases cómo la vista y el modelo se comunican entre sí mediante un controlador, el cual accede a los datos del modelo.	Modelo MVC	La clase Interfaz se deberá comunicar con la clase Controlador mediante las funciones que éste le provea (funciones para el usuario), a su vez el Controlador deberá acceder a los datos y funcionalidades del modelo para entregar una respuesta a la Vista de modo que ésta pueda entregar resultados por pantalla, es decir, cuando se realice una búsqueda la Vista le enviará al Controlador el contenido que el usuario desea buscar y los estilos de aprendizaje para los que desea realizar la búsqueda, para lo cual el controlador utilizará estos datos para extraer la información del modelo y enviársela a la vista para que esta la muestre en la interfaz. El patrón de diseño MVC separa los componentes del sistema de tal forma que la vista no se comunica ni maneja directamente el modelo, por lo cual sirve para los motivos mencionados.
Se desea mostrar en el Diagrama de Clases cómo la interfaz utiliza componentes del sistema para mostrar contenido en cada vista.	Facade	La interfaz debe interactuar con los componentes del sistema y las clases del modelo sin que el usuario tenga conocimiento de estos ni menos aún posibilidad de manipularlos. Por esta razón el patrón de diseño Facade es adecuado, pues permite que la interfaz interactúe con las clases del modelo mediante el uso de sus componentes, sin manejarlos directamente.
Se desea mostrar en el Diagrama de Clases la notificación que emite el sistema cuando un usuario realiza un feedback de algún contenido	Observer	La clase Feedback debe ser capaz de avisar al Técnico sobre el ingreso de un nuevo feedback. Así, el Técnico podrá modificar los parámetros de búsqueda según el feedback recibido(positivo o negativo). Como es necesario enviar una notificación a la vista de Técnico cuando se agreguen datos nuevos de feedback se escoge el patrón Observer, ya que este satisface esta necesidad debido a que permite mantener notificaciones en tiempo real a otras clases en función de una en particular.

3. Creación de Diagrama de Clases

A continuación se muestra el diagrama de clases realizado para cumplir con los objetivos del proyecto, con la incorporación de los patrones de diseño, MVC, Facade y Observer.

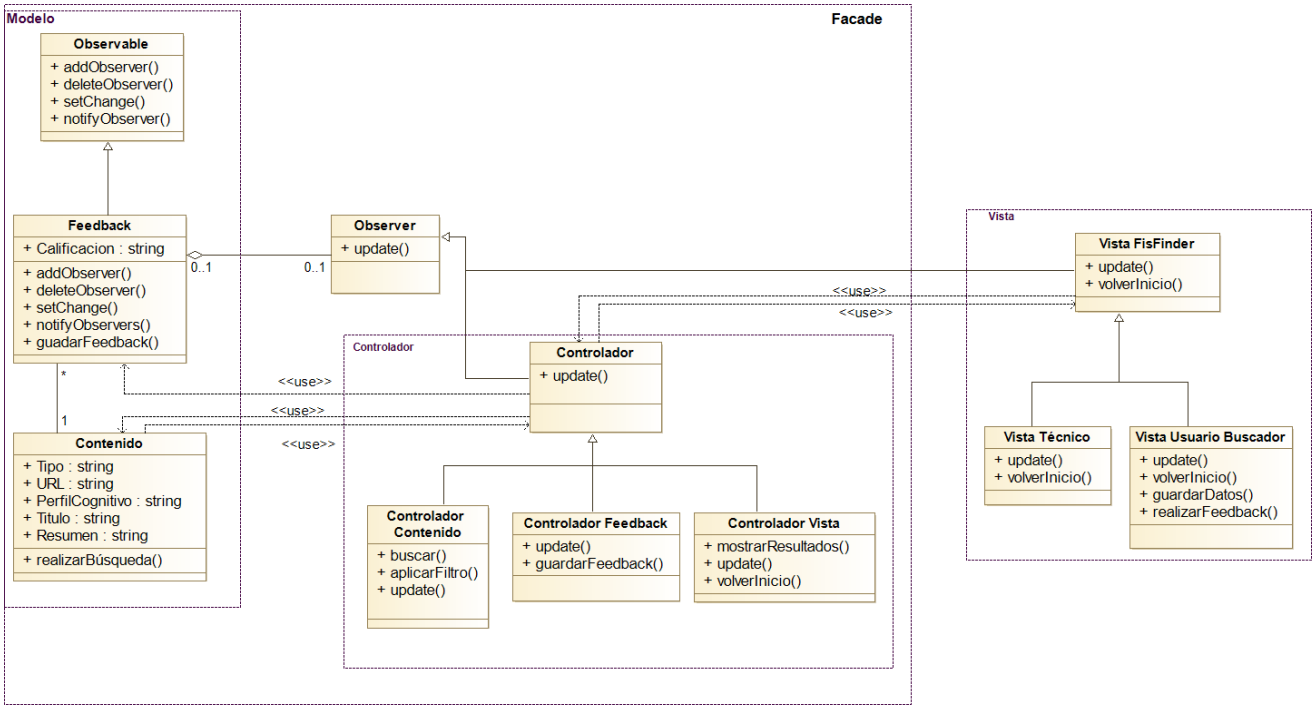

Figura 8: *Diagrama de clases con patrones de diseño MVC, Facade y Observer.*

Nota: En la documentación de Github se adjuntan el diagrama de clases en caso de algún problema con la visualización

4. Diagramas de Secuencia

Se escogieron los casos de uso **Realizar búsqueda** y **Realizar Feedback**. Realizar búsqueda se consideró porque es el objetivo principal del programa y además porque en este caso de uso se aprecia claramente el funcionamiento del modelo MVC para la obtención de información y visualización por pantalla. Por otro lado, Realizar Feedback se escogió porque este proceso será la guía principal para ver si el programa está funcionando correctamente, además en el se puede ver como funcionará el patrón de diseño Observer, el cual cumple un rol esencial en la notificación de nuevos feedback para que el técnico pueda cambiar los parámetros de búsqueda, si es que el feedback así lo dice, con el fin de mejorar el buscador.

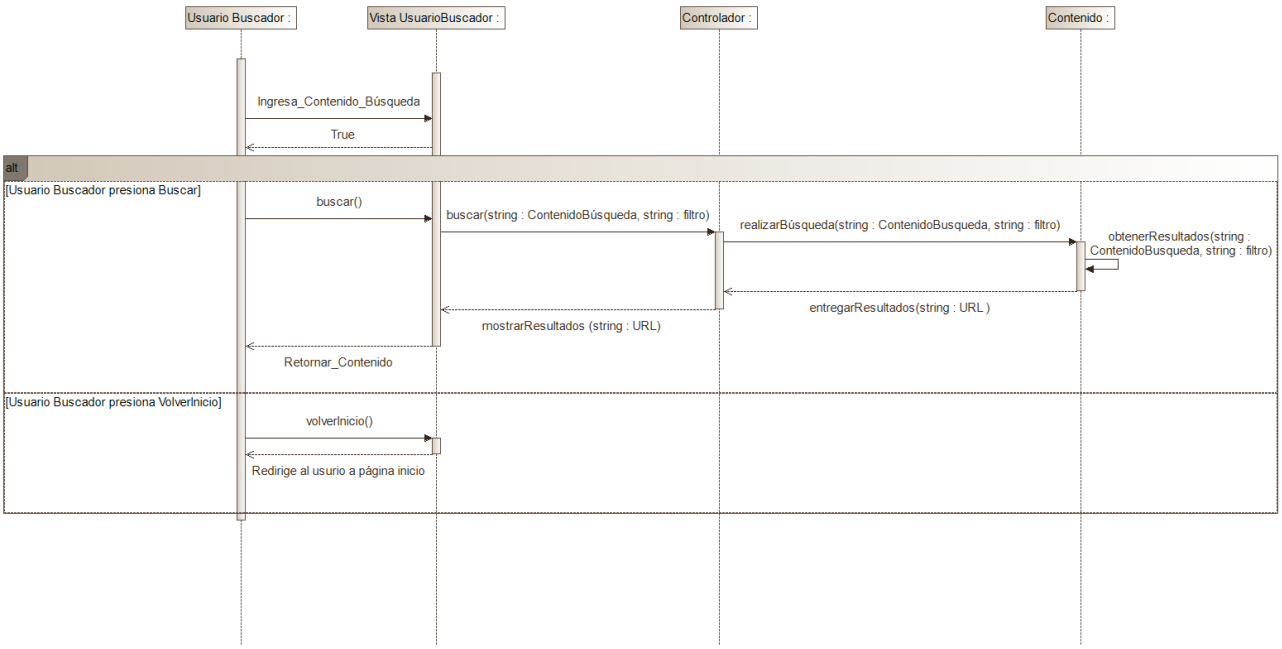

Figura 9: *Diagrama de secuencia del caso de uso Realizar búsqueda.*

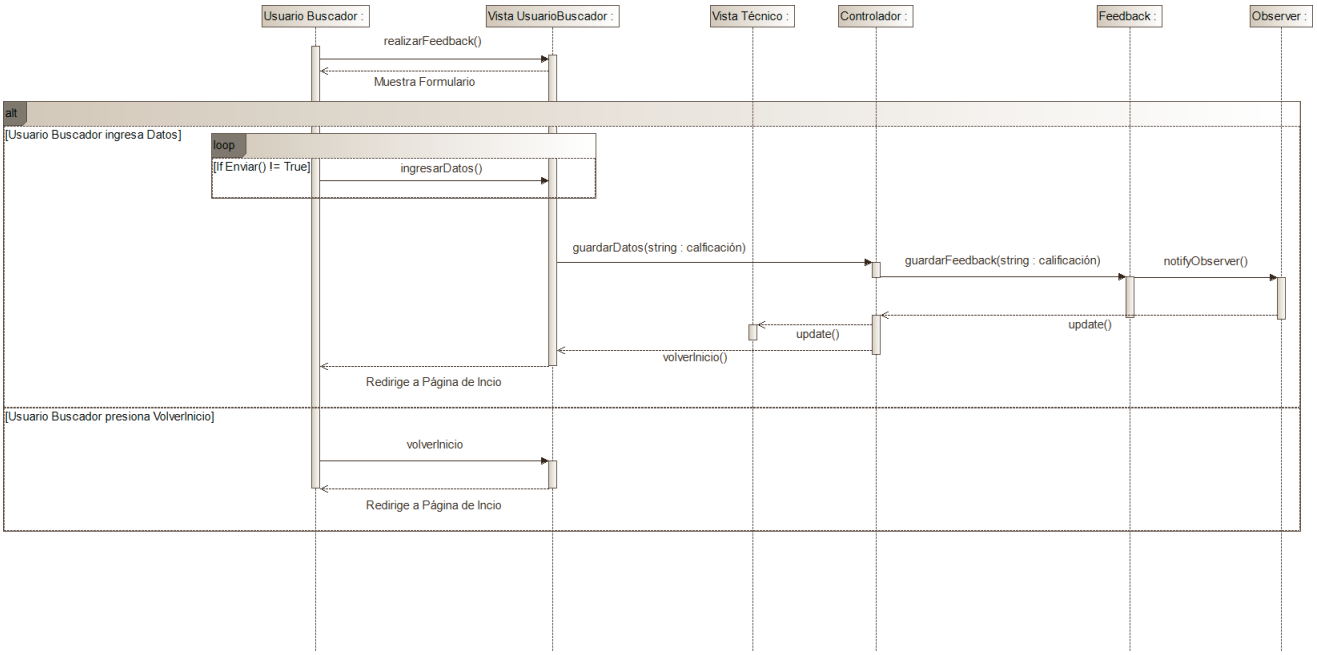

Figura 10: *Diagrama de secuencia del caso de uso Realizar feedback.*

Nota: En la documentación de Github se adjuntan ambos diagramas en caso de algún problema con la

visualización

5. Análisis de Trade-Off

La nueva funcionalidad identificada corresponde a la validación de contenido por parte de un experto. Las posibles soluciones halladas consisten en:

1) Agregar función de validación al perfil de usuario Buscador.

2) Agregar un nuevo actor "Experto" en el sistema.

3) Utilizar únicamente fuentes con material validado.

4) Contar con validación a partir de un comité encargado de ello.

Respecto a la posibilidad de implementar una de estas posibles soluciones, se consideran los requisitos del sistema en cuánto a cómo se desenvuelve una vez que esté funcionando. Dichos requisitos son:

I - Obtención de resultados bien logrados (coherentes con la búsqueda) en mayor parte.

II - Respuesta rápida del sistema.

III - Posibilidades de Mejora.

IV - Disponibilidad del Sistema.

Teniendo en cuenta las posibles soluciones y los requisitos que se deben cumplir, se establece la siguiente tabla de análisis, que considera el peso de una solución respecto a la satisfacción o desmedro de un requisito:

Requisito/Solución	Buscador que Valide	Nuevo actor Experto	Solo usar fuentes específicas	Comité Validador
Calidad	++	++	+	++
Rendimiento	0	0	+	0
Modificable	+	+	-	+
Disponibilidad	0	0	–	0

La solución 1 cumple con mejorar la obtención de resultados acordes con lo que se espera para una búsqueda. Por otra parte, no influye en disponibilidad del sistema ni en la rapidez de respuesta. Tiene ventaja en las posibilidades de mejora dado que el método de validación puede perfeccionarse.

Se tiene que la solución 2 es bastante similar, en cuanto a resultados, a la solución 1. Las diferencia está en quién realizará el proceso de validación, lo cual no afecta los requisitos que se tienen en consideración.

Por su parte, la solución 3 a pesar de significar una mejora en la obtención de resultados coherentes, está limitada a no poder mejorar el método de validación o modificarlo según las necesidades del cliente,

puesto que está fuera del alcance del desarrollo. Por otra parte, este método significa una mejora en los tiempos de respuesta, puesto que al usar sólo contenido validado se reduce el espectro de objetos entre los cuales buscar. Aunque, como punto en contra se tiene que la disponibilidad del sistema queda sujeta a la disponibilidad de la única fuente de contenido.

La solución 4 provee una mejora en la calidad de los resultados, sin afectar el rendimiento ni la disponibilidad del sistema. Por otra parte, es una solución que puede perfeccionarse en el tiempo.

Se considera como mejor opción a la primera presentada, puesto que significa un beneficio en la calidad de los resultados que puede ir mejorando aún más progresivamente y que por otra parte se traduce en que el buscador necesariamente sea quien se encargue de validar el contenido, por lo cual puede intervenir directamente cuando lo necesite. Lo anterior se entiende como un beneficio propio del contexto, dado que quien busca sabe que es lo que tenía en mente y puede distinguir de inmediato si lo halla o no.