



Departamento de Informática
Universidad Técnica Federico Santa María



Requisitos de Software

Proyecto: Bentobox

Grupo: Kryptonite

Integrantes:

Nombres y Apellidos	Email	ROL USM
Carlos Andrade Cabello	carlos.andrade.13@sansano.usm.cl	201373508-8
Camilo Saldías González	camilo.saldias.12@sansano.usm.cl	201223512-K

Desarrollo del Prototipo

Para este entregable, se implementó una función para buscar contenido en base a tags y al perfil de aprendizaje del usuario, clasificando los resultados por orden de relevancia en base a un sistema de puntuación.

Cada archivo, al momento de ser ingresado en la base de datos, es acompañado de una serie de tags (o *keywords*) que la definen, junto con una serie de datos que ayudan a caracterizar el documento, tales como nombre, descripción y URL de origen. Además, se incluye un factor de utilidad para cada uno de los 4 perfiles de aprendizaje, que indica cuán relevante es el documento para cada perfil de aprendizaje.

Para cada búsqueda realizada por el usuario, se realiza el siguiente cálculo para cada documento dentro de la base de datos:

$$\text{file_score} = (\text{match_keywords} / \text{num_keywords}) * \text{ponderacion_perfil}$$

donde:

- *match_keywords*: corresponde al número de *keywords* de la búsqueda del usuario que se encuentran en el archivo
- *num_keywords*: corresponde al número de *keywords* que contiene la búsqueda del usuario
- *ponderacion_perfil*: corresponde al factor de utilidad del archivo para el perfil de aprendizaje del usuario que realiza la búsqueda.

Esta función produce un número entre 0 y 1 para cada uno de los documentos que componen la base de datos, pudiendo ordenar los resultados por relevancia para el usuario y, de esta forma, mostrar primero los resultados que son relevantes para la búsqueda del usuario y el perfil de aprendizaje del mismo.

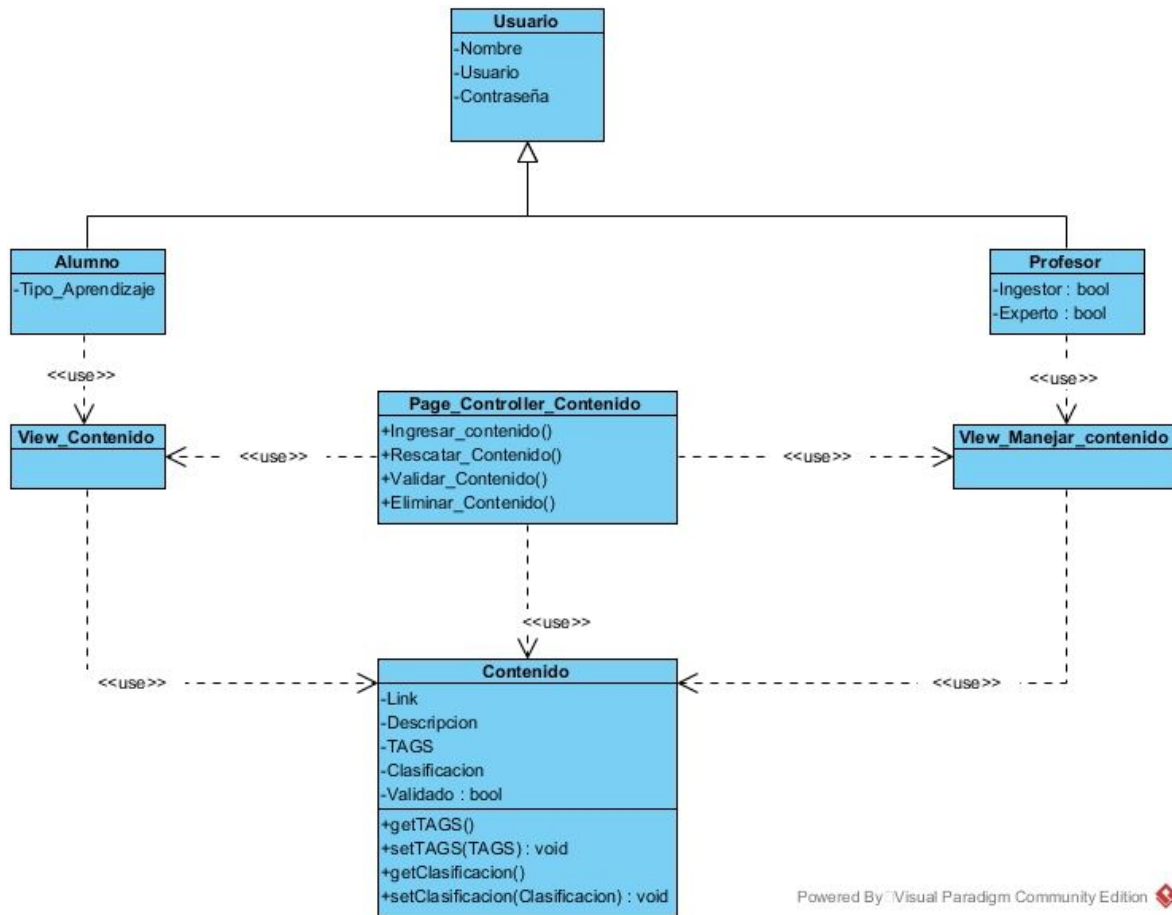
Una versión de prototipo de esta funcionalidad se puede apreciar en el código que acompaña este entregable, en el programa *searchrank_test.py*, la cual hace uso del paquete *searchRank* que implementa esta funcionalidad. Para esta versión, la base de datos de documentos consiste en un archivo de texto plano, aunque esto será cambiado a una base de datos en el futuro. Además, la función permite que la acción de ingresar nueva información a la base de datos sea parcial o totalmente automatizada en el futuro.

Selección de Patrones de Diseño

Intención	Patrón de Diseño	Razonamiento
Se busca una estructura bien definida del proyecto, que diferencie entre vistas y funcionalidades	Modelo-Vista-Controlador	Se debe desarrollar de manera separada todo lo relacionado con páginas o vistas, de forma aparte a los archivos controladores, que realizan las funciones del sistema, y a la vez separado de todo lo que se refiere al modelo del sistema.
Se desea que el controlador posea una gran importancia a nivel de sistema, esto debido a que las funcionalidades que posee el sitio son de gran importancia.	Page Controller	Los resultados de búsqueda se deben presentar de forma dinámica según el usuario, siendo este orden calculado a través de la funcionalidad PageRank, la cual a través de TAGS, además de la categoría del usuario, calculará cierto puntaje para cada resultado, para así poder organizar la presentación.
Debido al constante acceso a la información de un registro a la base de datos, a la vez que objetos del sistema tienen gran similitud a los registros de una tabla de la base, es conveniente encapsular el registro en un objeto.	Active Record	Al momento de realizar la búsqueda, el registro queda asociado a un objeto, el cual manejará tanto sus TAGS como su valoración para el usuario, luego de esto, se procede a calcular a través de otro sistema su puntuación para el usuario, seguido se procederá a ordenar la lista de resultados según este valor, finalmente se obtiene de cada objeto la información necesaria para presentar los

		resultados.
--	--	-------------

Creación de Diagrama de Clases



Diagramas de Secuencia

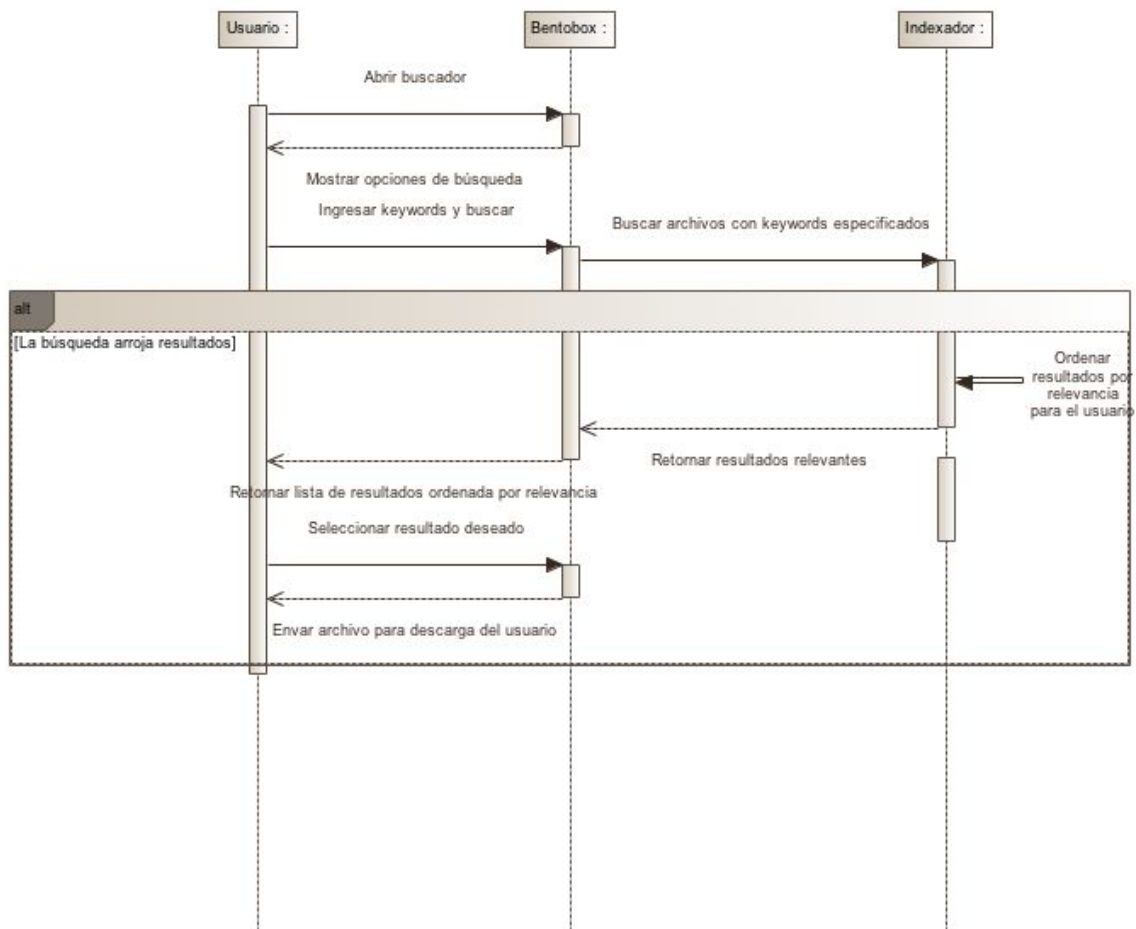


Diagrama de Secuencias para el requisito FR1

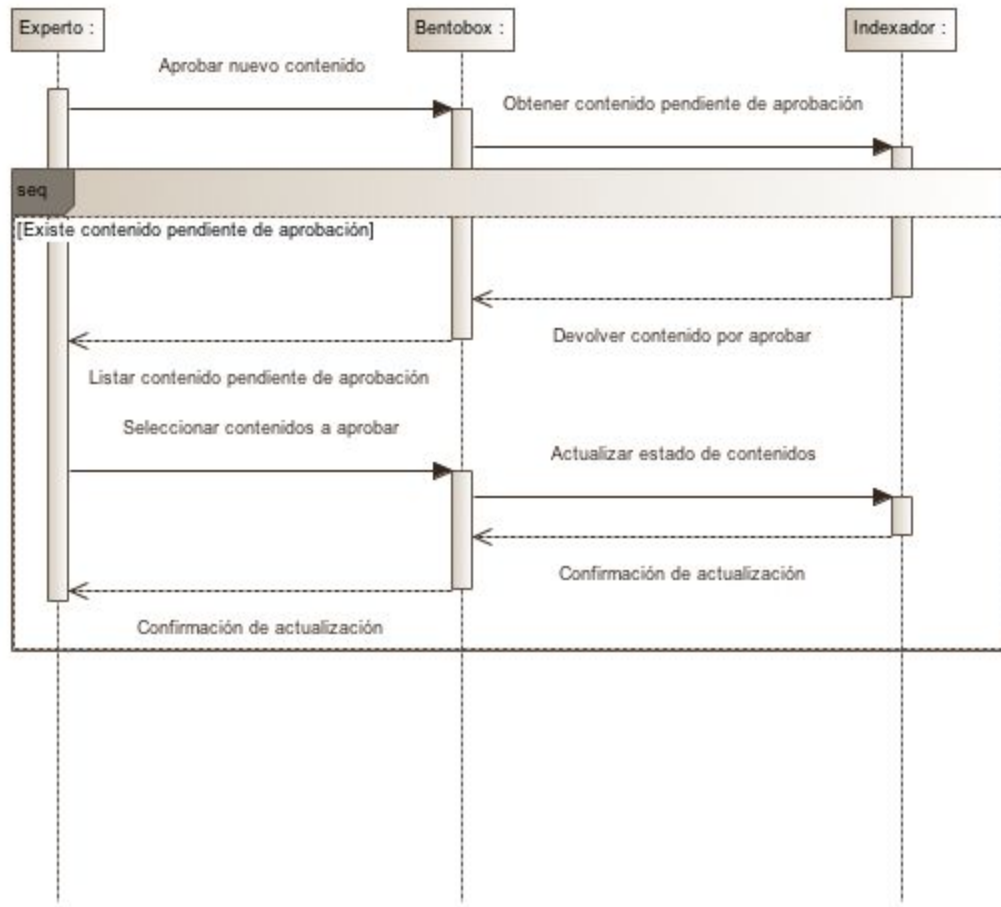


Diagrama de Secuencias para el requisito FR3

Análisis de Trade-off

1. Pregunta: *¿qué técnica usarán para realizar lo dicho, algoritmos propios para la búsqueda?*
2. Opciones:
 - O1: Desarrollar un buscador Web desde sus bases.
 - O2: Utilizar de manera directa un buscador ya desarrollado (EJ: Google)
 - O3: Desarrollar sobre un buscador ya existente y adaptarlo para cumplir nuevos objetivos.
3. Criterios de evaluación
 - C1: Rendimiento
 - C2: Escalabilidad
 - C3: Concurrencia

4. Análisis (Trade-Off):

Puntaje: ++ (Muy fuerte)/+ (Fuerte)/0 (No disponible)/-(Baja)/--(Muy baja).

Criterio\Opciones	O1	O2	O3
C1	0	++	++
C2	++	0	+
C3	0	++	++

- *La opción 1 (O1) La opción uno al ser un desarrollo completo no se puede asegurar el rendimiento ni la concurrencia, pero al ser construido especialmente para el sistema, es posible desarrollarlo de la forma más escalable posible.*
- *La opción 2 (O2) Ofrece un excelente rendimiento, puesto que los sistemas de búsqueda disponibles están en constante crecimiento, en respecto a la concurrencia sucede lo mismo, pero no es posible obtener información en respecto a la escalabilidad.*
- *La opción 3 (O3) La tercera opción presenta excelentes calificaciones frente a rendimiento y concurrencia, a la vez que posee buena presentación en escalabilidad.*

En conclusión, la opción 3 es la opción seleccionada por el equipo, debido a que es la opción que presenta una mejor presentación en todos los criterios. Además se observa que, al ser un desarrollo sobre soluciones ya desarrolladas, es una solución intermedia entre las otras dos, por lo que se considera la solución más apta.