



Departamento de Informática
Universidad Técnica Federico Santa María



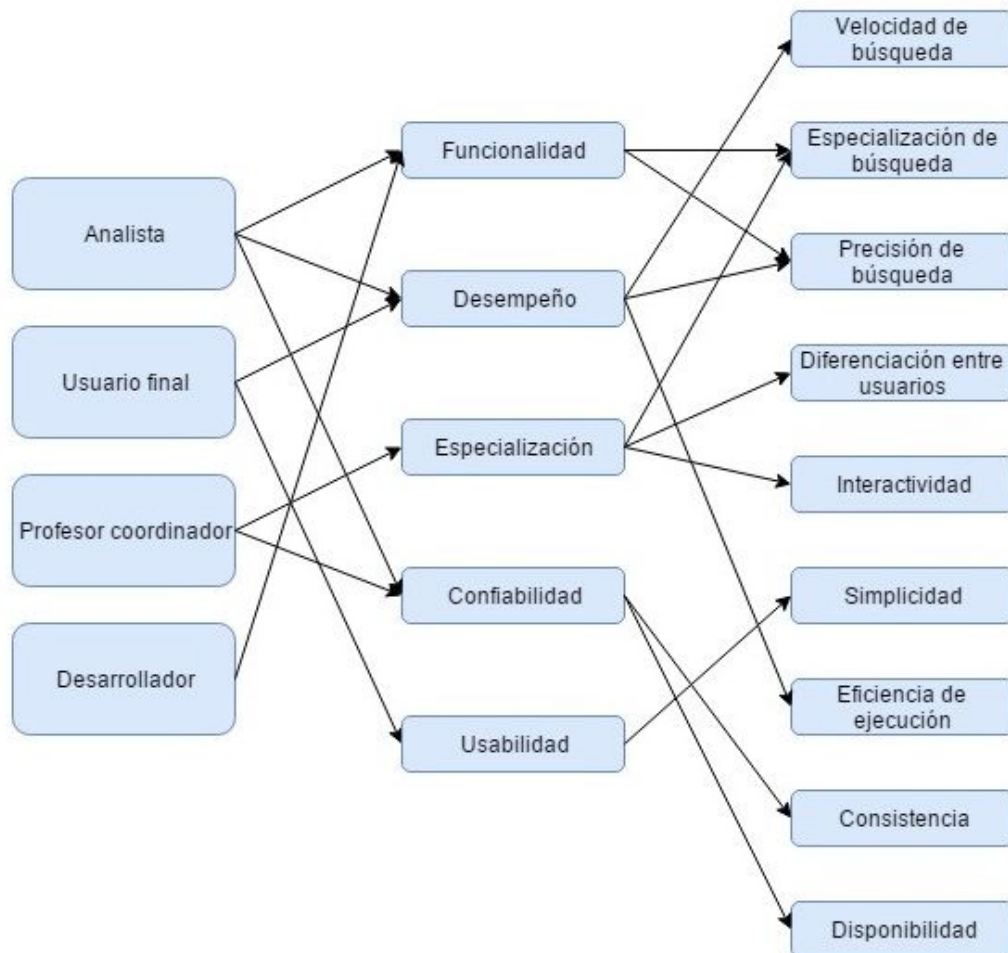
Entregable III

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

Modelo de Calidad



En nuestro modelo de calidad, los stakeholders se seleccionaron a partir de identificar qué actores toman interés en el desarrollo de nuestro proyecto: El analista es quien desarrolla el diseño de sistema, y el desarrollador quien crea el código que le da forma a este, ambos toman, por lógica, interés en el desarrollo del proyecto. Profesor coordinador es en este caso nuestro cliente, y futuro “dueño” de la plataforma, por ende también se considera un ente relevante. Finalmente, el usuario final es quien utilizará la plataforma para procesar búsquedas de material científico adaptado a su estilo de aprendizaje, dándole un sentido al desarrollo de este portal web.

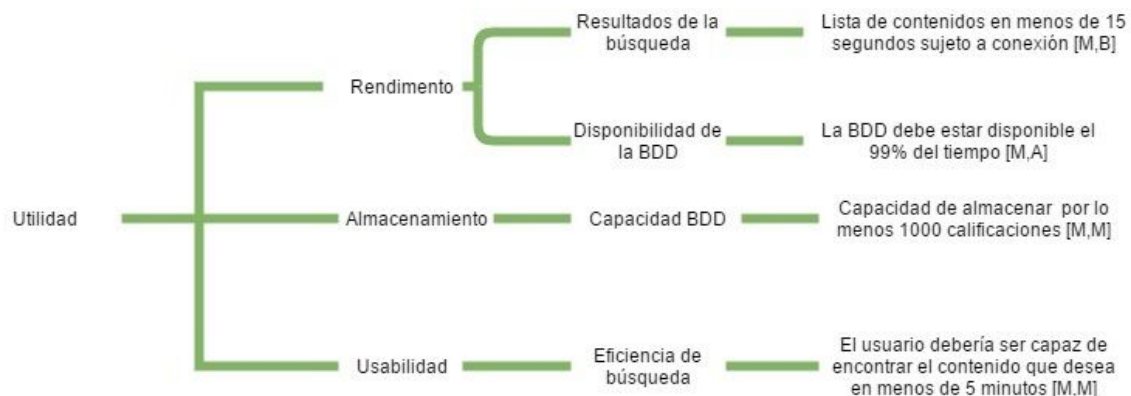
Sobre los atributos de calidad de alto nivel, se eligieron principalmente basados en los softgoals detallados en el entregable I, aprobado previamente por el cliente, aunque es necesario remarcar que no son los softgoals en sí, sino abstracciones de estos con nombres más ad-hoc, que fueron seleccionados dada su previa aprobación por el

cliente, además de que engloban características a grandes rasgos que desean los stakeholders seleccionados sobre el sistema. Funcionalidad hace referencia al conjunto de características funcionales deseadas en el sistema, por ello, tanto el analista como el desarrollador presentan interés en este atributo. Desempeño agrupa los sub-atributos que hacen referencia al rendimiento general del sistema, por ello el analista centra su vista en el, pero además lo hace el usuario final, cuya experiencia al utilizar la plataforma depende del desempeño de esta. Especialización es el atributo de alto nivel que desea con mayor prioridad el cliente, puesto que la idea general es crear una plataforma de búsqueda especializada; suena lógico entonces que el principal interesado en este atributo sea el profesor coordinador, nuestro cliente. Respecto a la confiabilidad, es en general, un atributo bastante grande que abarca distintos aspectos del sistema, no obstante, nosotros nos centramos en dos en particular, de los que se hablará posteriormente; quienes están interesados en esto son tanto el analista, puesto que es su rol elaborar esta característica, y el profesor coordinador, dado que los sub-atributos que describe son deseables por el cliente. Por último, el atributo de usabilidad es el que engloba las características que hacen “amigable” y utilizable el sistema por el usuario final, por ende es este quien marca su interés sobre este tópico.

En último lugar están los sub-atributos, aquellas características del sistema de bajo nivel que si bien son relevantes para el proyecto, son específicas y por tanto caen dentro de la categoría de otro atributo más general, o de alto nivel, su elección pasa al igual que los atributos de calidad de alto nivel, por un análisis cualitativo del sistema y sus necesidades por parte del equipo desarrollador, considerando además las demandas del cliente, y las futuras necesidades del usuario final. Pasaremos por una breve descripción de cada uno a continuación: Velocidad de búsqueda tiene un nombre bastante auto-descriptivo, cae en la categoría desempeño. Especialización de búsqueda hace referencia la función de filtro de resultados personalizada que el cliente requiere del sistema, perteneciendo tanto a funcionalidad como a desempeño. Precisión de búsqueda significa que tan acertados son los resultados arrojados por la búsqueda realizada en la plataforma, al determinar el producto final entregado, no solo califica como funcionalidad, sino también como desempeño. Diferenciación de usuarios es un atributo de especialización que denota el grado en que el sistema se adapta a cada tipo de usuario, no solo a través de la búsqueda, sino también en las sugerencias recibidas, por ejemplo. Interactividad, pertenece a especialización al igual que el anterior atributo, y es un atributo que se refiere a cómo el usuario influye en el contenido que ve más allá de escribir lo que desea encontrar, creando un ambiente de búsqueda dinámico. Simplicidad, es el atributo que describe informalmente lo “amigable” del sistema con el usuario, o sea, que tan fácil es para nuevos usuarios adaptarse a este, y por tanto es encasillado por el atributo de nivel superior usabilidad. Eficiencia de ejecución es un atributo de desempeño y hace referencia a que tan veloz es el sistema en general al momento de ejecutarse, como las partes del sistema responden a las acciones del usuario más allá de solo la búsqueda. Consistencia es un atributo de confiabilidad que evalúa qué tan íntegra es la información que almacena este, para el caso de SASP, mide que las calificaciones entregadas como “sugeridas” no sean explotables para generar información falsa o

errónea. Para finalizar, disponibilidad, otro atributo de confiabilidad, se refiere al uptime del sistema, o si este se encuentra disponible para los usuarios en todo o casi todo momento, este último solo puede ser medido una vez esté en línea.

Árbol de utilidad (actualización)



Como se puede apreciar a simple vista, los cambios que sufrió nuestro árbol de utilidad son bastante menores y sutiles, esto dado que tras realizar un análisis en profundidad del sistema en desarrollo, determinamos que las 3 propiedades principales del sistema estimables por mediciones objetivas son rendimiento, almacenamiento y usabilidad.

Sobre el rendimiento del sistema, refinamos las expectativas sobre la velocidad de búsqueda a 15 segundos dado que si bien, los 5 segundos especificados en el árbol de calidad anterior son, en general, suficientes con una buena conexión, esta última fluctúa y varía junto con el contexto en que la plataforma es utilizada, por ende una ponderación de 15 segundos sujeta a conexión nos parece más apropiado y garantizable en la mayoría de los casos. La disponibilidad de la base de datos se mantiene, dado que aún tenemos como objetivo que esta esté operativa tiempo completo para un funcionamiento óptimo del sistema, salvo unas pocas excepciones, como comportamientos imprevistos, tiempo de reparación de estos y/o mantenimiento del sistema.

En cuanto al almacenamiento tuvimos cambios mayores que en los otros aspectos: utilizando la idea de Lean Thinking cambiamos la forma en que modelamos las calificaciones de los enlaces, de un sistema de registro con cuentas que solo servían para validar la no repetición de valoraciones a un enlace, a una validación simple utilizando un correo electrónico. Esto pensando en la idea de que SASP aspira a ser una plataforma de uso didáctico y científico, para estudiantes, por tanto exploits de sobre valoración de enlaces no tienen cabida puesto que no sirven a ningún propósito, por ende reformulamos la verificación para ser solo una herramienta que prevenga a

cualquier estudiante de accidentalmente valorar 2 veces un mismo enlace, eliminando así una funcionalidad que no entrega valor al usuario final, o como describe el Lean Thinking, basura.

Finalmente, la usabilidad tras proyecciones sobre la funcionalidad actual entregada por la plataforma se decidió mantener puesto que se considera una buena medida 5 minutos desde que inicia una búsqueda con la generación de la necesidad de esta, hasta que culmina con el usuario final encontrando el material deseado, esto considerando la mayoría de los escenarios adversos posibles, como búsquedas mal tipeadas, tópicos de búsqueda erróneos, búsqueda reiteradamente infructuosa, material de poco aporte o por el contrario, información abundante repartida en muchas fuentes.

Pruebas de Software

Pruebas realizadas:

- Prueba 1: Buscar contenido no relacionado con la ciencia.
Corresponde al testing de input de búsqueda en que se pide al sistema búsquedas de cualquier tipo, para el caso del testing, contenido no científico, con el fin de corroborar la propiedad del sistema detallada como “búsqueda de material de estudio científico”. La búsqueda no debiera arrojar resultados.
La salida de esta prueba es un fallo dado que nuestro grupo aún no implementa el filtro de páginas para asegurar resultados enteramente científico, no obstante es un outcome esperado, dado que no es un fallo de implementación, sino que aún no se alcanza esa etapa del desarrollo del sistema. Queda pendiente de implementación.
- Prueba 2: Descargar contenido y no aceptar la descarga.
La idea es probar la reacción del sistema antes una solicitud de descarga, seguida de una negación en la confirmación, a fin de observar el comportamiento de la plataforma en escenarios alternativos.
No hubo salida de este test dado que el botón de descarga no ha sido implementado con el algoritmo detrás que la ejecute, también está pendiente de desarrollar.
- Prueba 3: Búsqueda errónea
Se probará la respuesta del sistema cuando se ingresa en el buscador secuencias de letras que no tienen sentido, o sea, búsqueda mal escritas, para testear la sensibilidad a error del sistema.
La respuesta del sistema fue exitosa como era esperado, dado que corrige las palabras a un acercamiento a estas, y ante un input mal escrito, retorna un output basado en la búsqueda bien escrita, mostrando que el sistema es capaz de detectar errores menores de escritura sin problemas.

- Prueba 4: Calificación errónea.
Se probará la respuesta del sistema cuando se ingresa otro carácter dentro del formulario para probar la consistencia de este, característica esencial para el correcto almacenamiento de las calificaciones en el sistema.
El outcome es un fallo dado que las calificaciones tampoco se encuentra implementadas al momento del testing. No obstante, si se tiene el código que permite la ejecución de esta funcionalidad, solo no está unido a su parte gráfica en la interfaz, y se prevé esté funcional dentro del corto plazo.
- Prueba 5: Búsqueda nula.
Se pone a prueba la respuesta del sistema ante una búsqueda nula, que no solo no debiese entregar resultados, sino que no debería ejecutarse en absoluto puesto que no se trata de ningún tipo de búsqueda real.
El resultado es exitoso, el sistema entrega una advertencia de búsqueda vacía y reanuda su estado anterior sin ejecutar ninguna acción adicional, la plataforma fue construida con resguardo contra este tipo de irregularidades.

Plan de mejoras

Para el caso de nuestro proyecto, las fallas a la hora de testear fueron completamente dadas por la falta de implementación de componentes que no pudieron ser apropiadamente testeados dado que no había algo que testear.

Si bien esto nos limita mucho de opciones para ejecutar un plan orientado a mitigar los defectos encontrados, en vista de que los componentes que sí pudieron ser testeados arrojaron el resultado esperado para todos los casos, podemos aún reconocer la falta de un trabajo planificado como la razón causante de la falta de componentes implementados, por lo que decidimos optar como plan de mejora una mejor coordinación y optimización de tiempos de trabajo para lograr el objetivo de alcanzar una mayor funcionalidad del sistema en la entrega.

En razón de todo lo anterior, consideramos que la tabla comparativa no aplica, por no haber resultados erróneos que corregir y contrastar con resultados nuevos, las implementaciones de nuevas funcionalidades que podrían contrastar con los resultados de “no ejecutado” anteriores no podrían estar a tiempo para alterar el contenido del informe, por ende la tabla se omite.

Sobre el QA

Como grupo nos encontramos de acuerdo en que ambos consultores QA fueron de gran aporte en el desarrollo de nuestro proyecto, dado que si bien, no identificaron “defectos” que pudieran ser corregidos en el corto plazo con acciones puntuales, si nos ayudaron a dar cuenta de varias funcionalidades completas que no estaban siendo implementadas en tiempo, tales como calificación, descargas y filtro de contenido no científico. Esto nos permitió elaborar un mejor plan de trabajo que apunte a la

implementación de estas características en el corto a mediano plazo.

Esto es obviamente gracias a que las pruebas sobre el sistema fueron apropiadamente elegidas y ejecutadas acorde a los criterios de los consultores QA. Sin pruebas su labor no hubiera sido realizable y por tanto no hubiéramos notado la necesidad de planificar mejor los avances del proyecto, por ende consideramos que las pruebas fueron un aporte valioso al desarrollo de nuestro proyecto, permitiéndonos detectar ciertas falencias en un tiempo apropiado.

En cuanto a la calificación de nuestros consultores QA, consideramos que la labor de ambos califica con un “5: Muy Bueno”.

Esto dado principalmente a que en primera instancia entregaron sus respectivos informes en un plazo prudente para la realización del proyecto e implementar las modificaciones necesarias para corregir errores menores en el sistema e incluirlos en el informe a tiempo, aunque nosotros no hayamos presentado errores corregibles en el corto plazo, dieron la ventana de tiempo apropiada para corregirlos a tiempo. Por otra parte se mostraron con una gran disponibilidad para atender dudas e inquietudes sobre su labor de QA, el testing y cualquier otra necesidad, utilizaron canales de comunicación apropiados y variados (contactaron por una red social distinta cada uno, además del correo) asegurándose del rápido flujo de feedback y uno de los dos incluso mantuvo el flujo de comunicación incluso durante el proceso, por tanto, en presencia de una labor claramente bien ejecutada y sin reparos que hacer, se les otorga a ambos esa calificación.

Formulación de pruebas de software de requisitos no funcionales

- Caso de prueba 1: Buscar contenido en tiempo menor a 5 segundos.
Se debe probar que el sistema se demore 5 segundos o menos en realizar una búsqueda ingresada por un usuario.
- Caso de prueba 2: Almacenamiento.
Se probará la capacidad de almacenamiento máximo de calificaciones del sistema.

Para el desarrollo de nuestra plataforma, y tras repasar el sistema en desarrollo varias veces con el fin de indagar más allá, hemos determinado que no existen mas NFR que puedan ser testeados en este momento. Si bien existen otros requisitos no funcionales detallados en nuestros entregables, no son testeables, como el uptime de la BD, y otros. Por dicha razón, el caso de prueba 1 fue testeado por los QA en más de un ítem.

Listado de Pruebas/Requisitos

Paso 1:

ID	Nombre del Requerimiento	Roles asociados	Descripción
RF1	Buscar contenido	Usuario final.	Mostrar enlaces a los contenidos buscados.
RF2	Calificar contenido	Usuario final	El usuario podrá calificar el contenido.
RNF1	Tiempo de búsqueda	Usuario final.	Podrá buscar contenido en menos de 10 segundos sujeto a la velocidad de conexión..
RNF2	Almacenamiento	Administrador del sistema.	Base de datos con capacidad mínima de almacenar 1000 recomendaciones.

Paso 2:

- RT1: Búsqueda debe estar mal escrita,
- RT2: Calificación debe contener caracteres inapropiados.
- RT3: La búsqueda debe estar vacía.
- RT4: La búsqueda de contenido no debe tener tópico científico.

Paso 3:

#Caso de Prueba	#RT	Caso de Prueba	Pasos del caso	Datos utilizados	Resultados obtenidos
CP1	RT1	Búsqueda errónea	1: Ingresar al sector de búsqueda. 2: Escribir algún tema a buscar con faltas de ortografía. 3: Presionar "Buscar"	Input = "ingeniería de software".	Ingreso exitoso. Se muestran resultados de búsqueda sobre la palabra correctamente escrita.
CP2	RT2	Calificación errónea.	1. Ingresar al sector de búsqueda. 2. Escribir algún	Input "h".	Sin ejecución. La función no se

			tema a buscar. 3. Presionar "Buscar". 4. Seleccionar el enlace. 5. Seleccionar "calificar un enlace". 6. Ingresar un carácter (input: "h"). 7. Enviar.		encuentra implementada, por lo que se asume un caso fallido.
CP3	RT3	Búsqueda nula.	1. Ingresar al sector de búsqueda. 2. Dejar en blanco el cuadro de texto. 3. Presionar "Buscar".	Input: ""	Éxito. Se muestra un mensaje de error de que se debe completar el campo.
CP4	RT4	Buscar contenido no relacionado con la ciencia.	1. El usuario se posiciona en el campo de texto para realizar una búsqueda. 2. El usuario ingresa la siguiente entrada: "Comprar pasajes para Brasil". 3. El usuario presiona el botón para realizar la búsqueda	Input: "Comprar pasajes para Brasil"	Fallido. El sistema arroja respuestas encontradas en la web. El sistema arroja links de páginas no-científicas.
CP5	N.A.	Descargar contenido y no aceptar la descarga,	1. El usuario presiona el botón "Descargar contenido", el sistema mostrará la información del contenido a descargar. 2. El usuario presiona el botón "Rechazar descarga".	N.A.	No ejecutado. Falta implementación.

CP6	N.A.	Test de tiempo de búsqueda	1. Ingresar al sector de búsqueda. 2. Escribir algún tema a buscar (input: "ingeniería de software"). 3. Presionar "Buscar".	Input cualquiera no nulo.	El resultado que se espera es una búsqueda realizada en un total de tiempo menor o igual a 10 segundos con una conexión a internet apropiada.
CP7	N.A.	Capacidad máxima de almacenamiento.	1. Creación de usuarios para sobrepasar la capacidad.	N.A.	Se espera que la BD soporte la creación de al menos 1000 calificaciones sobre enlaces mostrados por el sistema.

Paso 4:

#Caso de Prueba	#RF	#RT	Caso de Prueba	Pasos del caso	Datos utilizados	Resultados obtenidos
CP1	RF1	RT1	Búsqueda errónea	1: Ingresar al sector de búsqueda. 2: Escribir algún tema a buscar con faltas de ortografía. 3: Presionar "Buscar"	Input = "ingeniería de software".	Ingreso exitoso. Se muestran resultados de búsqueda sobre la palabra correctamente escrita.

CP2	RF2	RT2	Calificación errónea.	<ol style="list-style-type: none"> 1. Ingresar al sector de búsqueda. 2. Escribir algún tema a buscar. 3. Presionar "Buscar". 4. Seleccionar el enlace. 5. Seleccionar "calificar un enlace". 6. Ingresar un carácter (input: "h"). 7. Enviar. 	Input "h".	Sin ejecución. La función no se encuentra implementada, por lo que se asume un caso fallido.
CP3	RF1	RT3	Búsqueda nula.	<ol style="list-style-type: none"> 1. Ingresar al sector de búsqueda. 2. Dejar en blanco el cuadro de texto. 3. Presionar "Buscar". 	Input: ""	Éxito. Se muestra un mensaje de error de que se debe completar el campo.
CP4	RF1	RT4	Buscar contenido no relacionado con la ciencia.	<ol style="list-style-type: none"> 1. El usuario se posiciona en el campo de texto para realizar una búsqueda. 2. El usuario ingresa la siguiente entrada: "Comprar pasajes para Brasil". 3. El usuario presiona el botón para realizar la búsqueda 	Input: "Comprar pasajes para Brasil"	Fallido. El sistema arroja respuestas encontradas en la web. El sistema arroja links de páginas no-científicas.
CP5		N.A.	Descargar	1. El usuario	N.A.	No ejecutado.

			contenido y no aceptar la descarga,	presiona el botón "Descargar contenido", el sistema mostrará la información del contenido a descargar. 2. El usuario presiona el botón "Rechazar descarga".		Falta implementación.
CP6	N.A.	N.A.	Test de tiempo de búsqueda	1. Ingresar al sector de búsqueda. 2. Escribir algún tema a buscar (input: "ingeniería de software"). 3. Presionar "Buscar".	Input cualquier a no nulo.	El resultado que se espera es una búsqueda realizada en un total de tiempo menor o igual a 10 segundos con una conexión a internet apropiada.
CP7	N.A.	N.A.	Capacidad máxima de almacenamiento.	1. Creación de usuarios para sobrepasar la capacidad.	N.A.	Se espera que la BD soporte la creación de al menos 1000 calificaciones sobre enlaces mostrados por el sistema.