

Universidad Técnica Federico Santa María

Ingeniería de Software

Entregable 3

Buscador de Herramientas de Aprendizaje

Autores:

Quantum Taco

Francisco Olivares

francisco.olivars.14@sansano.usm.cl

201473575-8

Felipe Vega

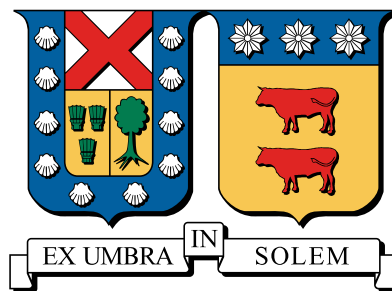
felipe.vega.14@sansano.usm.cl

201473511-1

Profesor:

Hernán Astudillo

Primer Semestre 2017



Índice

1. Modelo de calidad [1]	1
2. Árbol de utilidad	3
3. Pruebas de Software	4
3.1. Trabajo conjunto con consultores QA	4
4. Formulación de pruebas de requisitos no funcionales	4
5. Listado de Pruebas/Requisitos	5
6. Referencias	6

1. Modelo de calidad [1]

Stakeholders

- Cliente y Usuario final, Profesor Maximiliano Rivera.
- Equipo Quantum Taco

Obviamente el principal stakeholder es el cliente Maximiliano Rivera, ya que lo que busca el proyecto es generar un software que ayude a resolver el problema (necesidad) planteado por él. El equipo Quantum Taco también es parte de los stakeholders ya que son los principales desarrolladores de la solución y están altamente interesados en el resultado del proyecto.

Atributos de calidad de alto nivel y Sub atributos

- **Adecuación funcional**, es la capacidad del software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas, cuando el producto se usa en las condiciones especificadas. Se selecciono porque el software debiese tener las funcionalidades necesarias para solucionar el problema abordado (ej: Buscar en la web). **Complejidad funcional**, **Corrección funcional** y **Pertinencia funcional** son sub-atributos que apuntan a que las funciones del software sean las que resuelvan el problema de manera correcta y completa.
- **Eficiencia de desempeño**, característica que representa el desempeño desde el punto de vista de los recursos del sistema bajo determinadas condiciones. Se seleccionó porque es relevante los tiempos que se demoren en realizar las búsquedas. **Comportamiento temporal** y **Capacidad** son sub-atributos que ayudan al desempeño. Comportamiento temporal va relacionado con el tiempo de espera al realizar las búsquedas. Capacidad en este proyecto tiene que ver con el número máximo de sitios web que se están ocupando al realizar las búsquedas los cuales claramente afectan la eficiencia.
- **Usabilidad**, entendida como la capacidad del software de ser entendido, aprendido, usado y atractivo para el usuario final. Se selecciono porque el software será usado por una persona no informática, entonces es necesario que pueda ser capaz de usarlo y que el proceso de adaptación a este no sea demasiado. **Inteligibilidad**, **Aprendizaje** y **Operabilidad** son sub-atributos que ayudan a ver que tan usable es el sistema. Inteligibilidad se refiere a la capacidad del software que permite al usuario entender si este es adecuado para sus necesidades. Aprendizaje es la capacidad del software que permite al usuario aprender como usar la aplicación. Finalmente Operabilidad se refiere a la capacidad del sistema que permite al usuario operarlo y controlarlo con facilidad.
- **Mantenibilidad**, referida a la capacidad del sistema de ser modificado de manera efectiva y eficientemente debido a diferentes necesidades. Se selecciono porque al cliente le interesa trabajar con varias paginas de la web, entonces es relevante la capacidad del software para poder trabajar sobre nuevas paginas. **Reusabilidad**, **Capacidad de modificación** y **Capacidad de prueba** son sub-atributos que ayudan a aumentar la mantenibilidad del sistema. Reusabilidad es la capacidad de volver a usar componentes de software en otros productos. La capacidad de modificación es que tan fácil es modificar, de forma efectiva y eficiente, el sistema sin inducir a defectos. Por último, la capacidad de prueba se refiere a la facilidad con la que se pueden crear criterios de prueba y determinar si se cumplen estos mediante casos de prueba.
- **Portabilidad**, representa que tan fácil es poder llevar el software de un ambiente a otro (distinto hardware, SO, etc). Se selecciono porque en el proyecto se necesita poder ejecutar el software que se esta desarrollando en Linux en el sistema que ocupa el profesor MacOS. **Adaptabilidad** va relacionado con la capacidad de adaptar el software a otro entorno, es decir correr en el computador del cliente (distinto SO y hardware al nuestro). **Capacidad para ser instalado** es que tan fácil es ser instalado/desinstalado de otro entorno, es necesario que se pueda instalar en el computador del cliente y también en el de los evaluadores del proyecto.

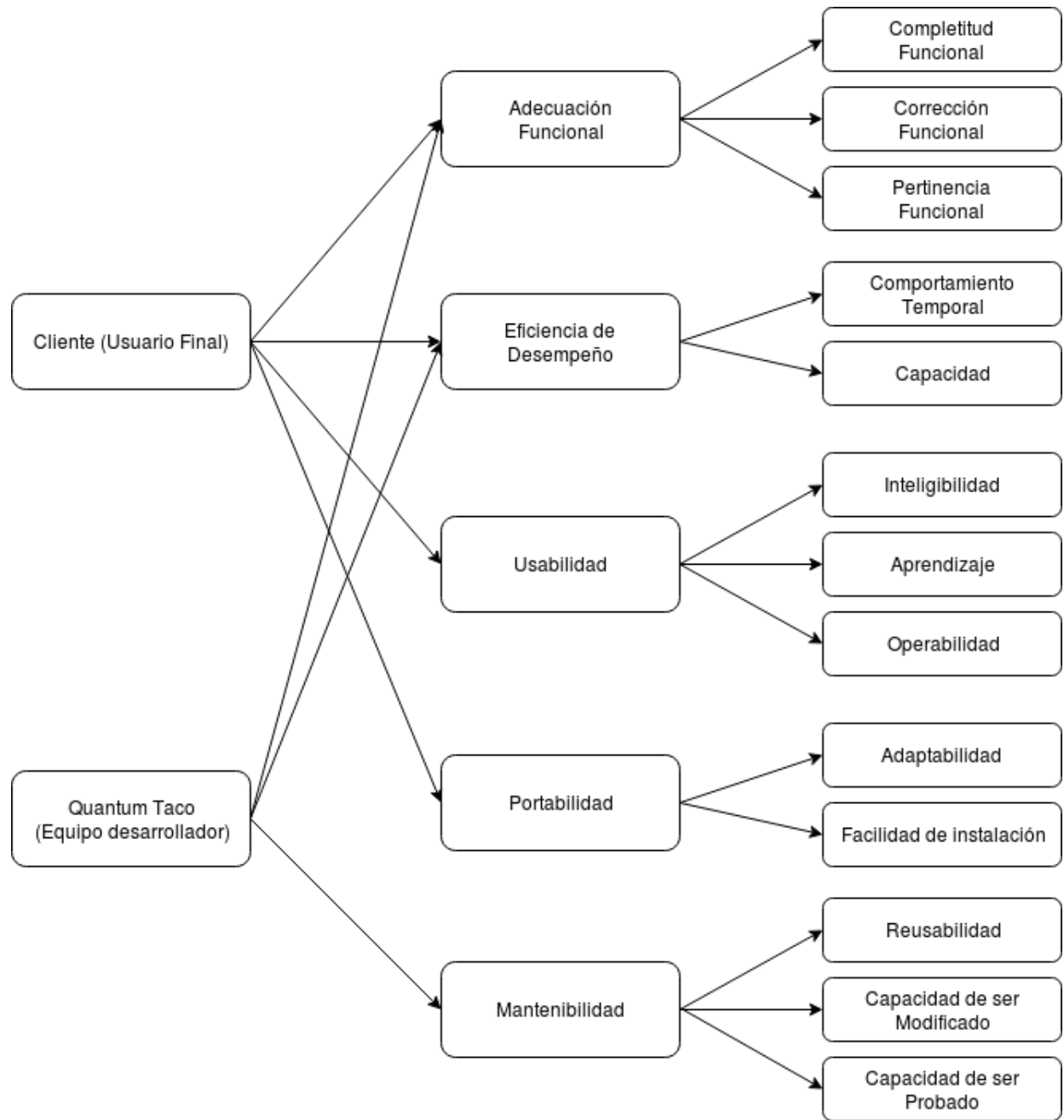


Figura 1: Modelo de calidad de Software

2. Árbol de utilidad

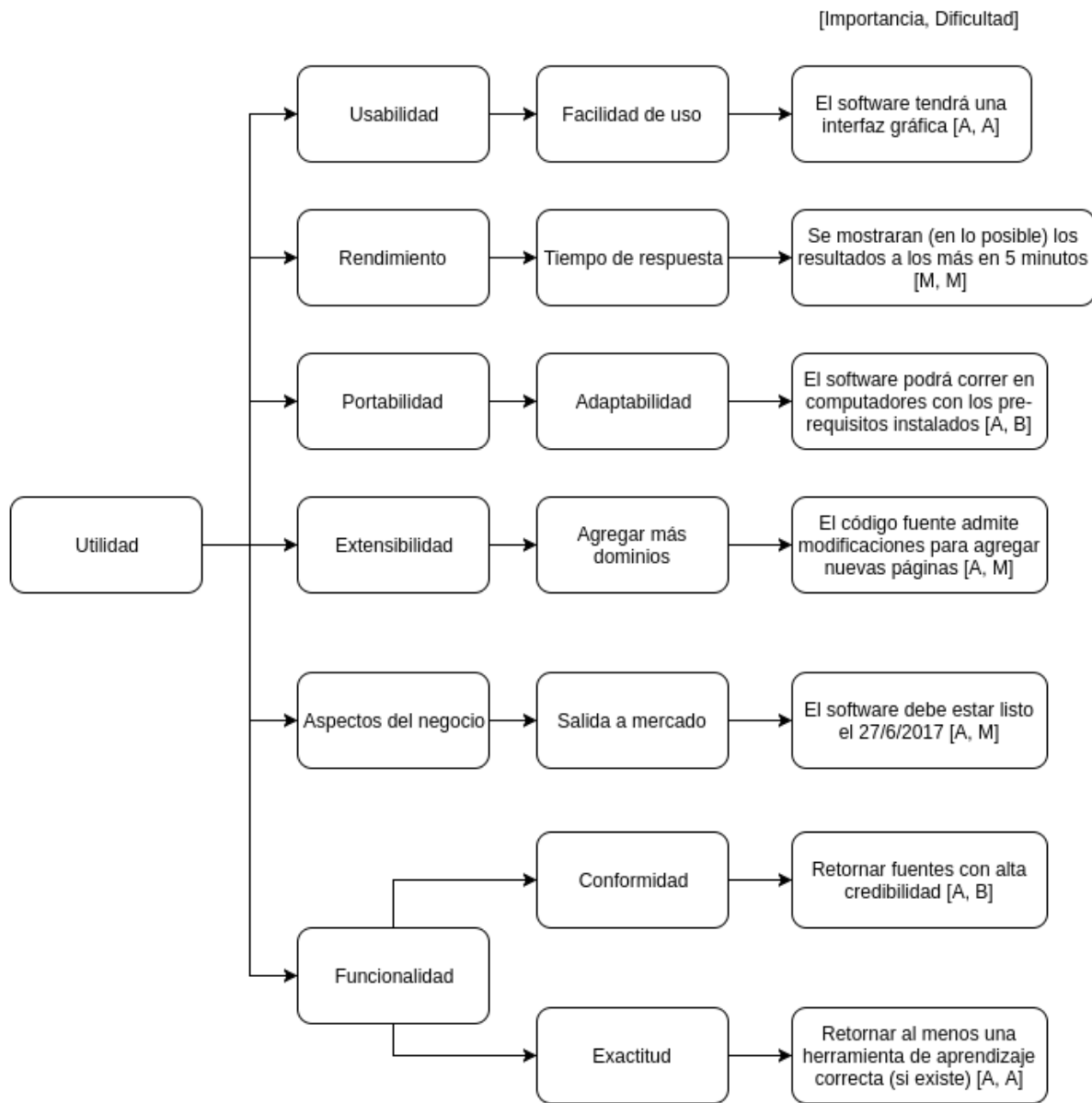


Figura 2: Árbol de utilidad actualizado

Al inicio del proyecto no detectamos requisitos extra-funcionales, pero con el paso del tiempo y al ver el resto de las presentaciones concordamos que hay requisitos que son necesarios para el desarrollo del proyecto (ej: tiempo de respuesta), por lo tanto se agregaron nuevas ramas al árbol. El desarrollo del proyecto no cambió drásticamente ya que para cuando nos dimos cuenta de las nuevas necesidades la cantidad de desarrollo era ínfimo, debido a esto no hubo que re-escribir código.

IMPORTANTE! Nótese que en la hoja “Se mostrarán (en lo posible) los resultados a lo más en 5 minutos” está sostenido a 2 condiciones, la velocidad de conexión de internet desde donde se corra la aplicación y la cantidad de imágenes que tenga la página. En el caso particular de las imágenes ocurre que el algoritmo baja la imagen y la decodifica lo cual es costoso en tiempo, entonces si la página tiene una cantidad grande de imágenes (ej: 100) es posible que no se pueda cumplir con la cuota de tiempo mínimo.

3. Pruebas de Software

Por parte de nuestro QA Eduardo Borgoño se realizaron las siguientes pruebas de software

- **Búsqueda vacía**, Esta prueba consiste en que al momento de ingresar al menú principal de la aplicación se debería verificar si el campo de búsqueda es no nulo, los resultados obtenidos de esta prueba fueron **fallidos** ya que al no ingresar un string en la búsqueda se genera un error y en la consola se muestra **IndexError: tuple out of range**.
- **Sin resultados**, Al seleccionar una tópicos y buscar una herramienta de aprendizaje se puede dar el caso que no hayan resultados de la búsqueda, de darse el caso la aplicación debería informar al usuario que no hay resultados de la búsqueda y permanecer en el menú principal. Los resultados de esta prueba son **fallidos** ya que al no encontrar resultado de la búsqueda la aplicación muestra la vista en donde se visualizan los resultados de la herramienta de aprendizaje deseada, pero no muestra nada en estas vistas y no entrega ningún tipo de feedback sobre que no hay resultados.
- **Propiedad intelectual**, Se espera que cada objeto de aprendizaje encontrado tenga una referencia a su fuente. El resultado de esta prueba es **exitoso** ya que al momento de mostrar las herramientas de aprendizaje (materia o imágenes) aparece la referencia a la fuente de donde proviene la herramienta.

Después de realizar las mejoras al software se logró tener éxito en ambas pruebas, para la **Búsqueda vacía** si es que el usuario no ingresa un tópico, la aplicación se queda en el menú principal y le muestra un mensaje en rojo que dice “Error, por favor ingrese un tópico para la búsqueda”. En el caso de **Sin resultados** la aplicación también se queda en el menú y muestra un mensaje en rojo que dice “No se encontraron resultados para <tópico ingresado>”.

3.1. Trabajo conjunto con consultores QA

Hasta el momento contamos con la participación de uno de los dos consultores QA asignados a nuestro equipo (Eduardo Borgoño), el cual se muestra abierto a recibir y responder consultas por parte del equipo desarrollador, misma disposición a realizar las pruebas de software, sobretodo al momento de volver a realizar dichas pruebas una vez corregidos los defectos encontrados. Por parte del segundo consultor QA (Erick Valenzuela) no hemos recibido material debido a la carga académica y el número de equipos que tiene asignados.

Las pruebas realizadas hasta el momento han sido de gran aporte para detectar situaciones no consideradas al momento de programar las funcionalidades solicitadas por el cliente. También fueron útiles al momento de recordar verificaciones que, si bien se tenían en cuenta, debido al tiempo fueron postergadas y hasta olvidadas.

Finalmente se muestra la calificación de ambos consultores QA en la tabla 1.

Consultor	Calificación (escala de 1 a 5)
Eduardo Borgoño	5
Erick Valenzuela	–

Tabla 1: Tabla de calificaciones para consultores QA, se omite calificación para Erick Valenzuela debido a que aún no se recibe feedback por su parte.

4. Formulación de pruebas de requisitos no funcionales

- Requisito: La aplicación debe devolver herramientas de aprendizaje con su respectiva fuente de origen. La prueba sería lo mismo que entrego el QA Eduardo.
- Requisito: La aplicación ser fácil de usar. Prueba: El usuario al querer buscar sobre una herramienta de cierto tópico a lo más deberá hacer 5 clicks.
- Requisito: La aplicación mostrará los resultados (en lo posible) en a lo más 5 minutos (Ver final de la sección 2 para más información). Prueba: Correr la aplicación en un computador con conexión estable y medir el tiempo al momento de realizar búsqueda por un tópico y una herramienta de aprendizaje seleccionada.
- Requisito: La aplicación debe poder correr en el sistema operativo del cliente (MacOS). Prueba: Instalar el proyecto en un computador con sistema operativo MacOS y ver si puede ser ejecutada.
- Requisito: La aplicación debe mostrar los resultados con una distinción clara entre cada uno. Prueba: Se debe realizar una búsqueda exitosa de más de 1 objeto de aprendizaje y ver si es que hay algún método de separación entre cada resultado.

5. Listado de Pruebas/Requisitos

ID	Nombre del requerimiento	Roles	Descripción
RF1	Búsqueda de herramientas	1)Usuario (Cliente Maximiliano Rivera)	Usuario: El usuario debe ingresar un tópico, seleccionar una herramienta de aprendizaje y apretar el botón iniciar búsqueda. Una vez encontrado los resultados, estos se mostraran en la aplicación.
RNF1	Referencia de herramientas	No aplica	La aplicación al entregar los resultados de cada búsqueda deberá mostrar una referencia al origen (url) desde dónde proviene ese objeto.

Tabla 2: Requisitos probados en los casos de prueba.

Requisitos técnicos

- RT1: El campo de búsqueda no debe estar vacío para buscar.
- RT2: Se debe seleccionar una herramienta de búsqueda para buscar.
- RT3: Sólo se refleja un cambio en la interfaz si la palabra ingresada generó resultados.
- RT4: Cada resultado debe estar en conjunto con el enlace del que se obtuvo la materia.

N° prueba	RF RNF	RT	Caso de prueba	Pasos del caso	Datos utilizados	Resultados obtenidos
CP1	RF1	RT1 RT2	Búsqueda vacía	<ul style="list-style-type: none"> ■ Usuario ingresa texto a buscar ■ Usuario selecciona una herramienta de búsqueda ■ Usuario presiona el botón "Iniciar búsqueda" ■ Se muestra mensaje en pantalla. 	Se ingresa texto vacío "" y se selecciona cualquier herramienta.	Se muestra mensaje pidiendo que ingrese un tópico.
CP2	RF1	RT3	Sin resultados	<ul style="list-style-type: none"> ■ Usuario ingresa texto a buscar. ■ Usuario selecciona una herramienta de búsqueda. ■ Usuario presiona botón "Iniciar Búsqueda". ■ Se muestra mensaje en pantalla. 	Se ingresa el texto "asd", se selecciona la herramienta "materia" y luego se inicia la búsqueda	Se muestra un mensaje indicando que no se encontraron resultados para "asd"
CP3	RNF1	RT4	Propiedad intelectual	<ul style="list-style-type: none"> ■ Usuario ingresa texto a buscar. ■ Usuario selecciona una herramienta para la búsqueda. ■ Usuario presiona botón para iniciar búsqueda ■ Se muestran resultados en pantalla. 	Se ingresa el texto "Ondas", la herramienta es "Materia".	Cada objeto encontrado tiene asociado el url fuente de origen.

Tabla 3: Listado de pruebas y requisitos.

6. Referencias

[1] Norma ISO/IEC 25010. <http://iso25000.com/index.php/normas-iso-25000/iso-25010>.