

Título

Uso de tecnologías abiertas (Processing) en el desarrollo de interfaces avanzadas de visualización y acceso de datos.

Estudio del caso Organic.Edunet

Autor

Francisco de Andrés Cerdá

Abstract

En los últimos años, el desarrollo de la web semántica y de las interfaces avanzadas para la visualización de datos ha experimentado un gran auge, en paralelo al aumento de información que de forma exponencial ha ido poblando la web.

Cada día es más necesario poder ordenar toda esa información, o poder visualizar, por ejemplo, como reacciona una determinada red social ante un acontecimiento importante. Así, a cada momento, aparecen aplicaciones o interfaces que nos ayudan a buscar, encontrar, visualizar, procesar y comprender esa información.

Sin embargo, el uso de tecnologías cerradas en este proceso impide una evolución paralela al ritmo con el que se presentan las necesidades.

El artículo mostrará, a través del estudio del caso del [buscador semántico](#) del portal [Organic.Edunet](#), cómo el uso de tecnologías cerradas en la creación de este tipo de interfaces impide su desarrollo y evolución. En el artículo se mostrará también cómo, combinado con técnicas para la medición y valoración de la usabilidad de las aplicaciones, el uso de tecnologías abiertas permite detectar los problemas del *interface*, proponer soluciones o alternativas, e implementarlas rápidamente.

1 Introducción

La forma de desarrollar software, tanto si se trata de desarrollos cerrados como si se trata de Software Libre, ha ido evolucionando a lo largo del tiempo en una búsqueda de paradigmas que permitan un desarrollo mucho más rápido y sin errores.

Este tipo de modelos se basan en desarrollos cortos y rápidos, que se retroalimentan para evolucionar, y que requieren que pueda revisarse el trabajo realizado hasta el momento de forma sencilla, de manera que puedan solucionarse los problemas detectados e implementar nuevas funciones.

Este tipo de desarrollo permite además el uso de herramientas complementarias al desarrollo de software en sí, pudiendo integrarlas de forma sencilla y a un bajo coste en cualquier etapa del desarrollo.

Por otro lado, la aparición de la web, especialmente desde el auge de la denominada web 2.0, ha provocado un aumento de la cantidad de información generada que precisa ser procesada de algún modo para poder ser entendida y resultar útil para los usuarios. Este tipo de información puede ir desde la presentación de los datos resultantes de un estudio realizado por una institución académica, empresa o medio de comunicación, hasta la información generada directamente por los usuarios en las redes sociales.

Para el procesamiento y presentación de estas cantidades ingentes de información no se puede enfocar el problema desde una sola perspectiva, sino que son necesarias diferentes aproximaciones para poder dar una solución válida.

Así, el uso de las herramientas semánticas nos permite dotar a la información de sentido para que pueda ser entendida y relacionada por las máquinas que lo procesan. Y el uso de interfaces gráficas avanzadas hace posible presentar dicha información de forma que tenga sentido para el usuario.

Sin embargo, estas técnicas y herramientas están todavía en una fase que hacen que sea necesario un modelo de desarrollo que permita iteraciones rápidas para poder probar y modificar los cambios con el mínimo esfuerzo posible.

En este escenario se hace fundamental también el uso de una herramienta que nos permita detectar los problemas e implementar de forma rápida las necesidades de nuestro desarrollo: los Test de Usabilidad.

Sin embargo, esta forma de desarrollo se encuentra con problemas cuando una de las partes se encuentra desarrollada bajo código cerrado, lo que crea un tapón que impide que los avances puedan ser implementados de forma rápida.

Así ocurre en el caso del portal Organic.Edunet que se utilizará en este artículo como sujeto de estudio.

Organic.Edunet es un portal creado como "una federación online multilingüe de repositorios de aprendizaje con contenido de calidad para la sensibilización y educación de la juventud europea sobre Agricultura Ecológica y Agroecología". A través del mismo se ofrece enlaces a los contenidos de los repositorios federados. Existen varios métodos de búsqueda para los contenidos en el portal, pero sin duda el más atractivo y el que en principio tiene mayor capacidad para sacar el máximo provecho de las características de web semántica que tiene el portal es el denominado Buscador Inteligente, una interface gráfica para la búsqueda que nos permite visualizar las relaciones entre los distintos conceptos.

El problema es que dicha interface se desarrolló como una aplicación Flash cerrada, de la que no se dispone del código, por lo que tal y como se verá en el artículo, la implementación de nuevas funciones, así como la solución de problemas, resulta en la práctica casi inviable.

Para demostrar las ventajas de los modelos y tecnologías libres en este tipo de desarrollos, a lo largo del artículo se realizará un estudio de usabilidad del actual Buscador Semántico, con el fin de detectar fallos, problemas en el uso y necesidades no satisfechas.

A partir de los datos obtenidos se propondrán ejemplos de soluciones para el rediseño del buscador y se hará una demostración práctica de la sencillez para implementar mejoras basándose en lenguajes de programación abiertos (Processing /Java).

Así, los objetivos del presente artículo son:

- Describir el estado del arte de las tecnologías implicadas en el desarrollo de los interfaces avanzados.
- Demostrar la utilidad de los Test de Usabilidad a la hora de identificar y corregir errores y nuevas necesidades.
- Plantear posibles soluciones para un rediseño del Buscador Semántico de Organic.edunet basadas en los datos obtenidos en los Test de Usabilidad.
- Mostrar la ventaja del uso de métodos y tecnologías libres en la aplicación a nuestro desarrollo de los resultados encontrados con los Test de Usabilidad.

El presente artículo se estructura de la siguiente forma: en el apartado 2 se describirá el estado del arte de las tecnologías implicadas en el desarrollo de interfaces avanzados; El apartado 3 contendrá el desarrollo del proyecto y se dividirá en los siguientes puntos: en el punto 3.1 se expondrán los resultados del Test de Usabilidad realizado sobre el Buscador Inteligente del portal Organic.Edunet, en el apartado 3.2 se plantearán posibles soluciones para el rediseño del Buscador Inteligente basadas en los datos obtenidos en los Test de Usabilidad, en el apartado 3.3 se comentarán las ventajas del modelo de Software Libre, en el apartado 3.4 se mostrará la facilidad de implementación de los resultados obtenidos en los Test de Usabilidad mediante el uso de tecnologías libres; finalmente los apartados 4 y 5 presentarán un resumen y conclusiones del trabajo realizado, así como las referencias usadas en el desarrollo del mismo. También se adjuntará como documentación separada del artículo dos anexos conteniendo la documentación utilizada en los Test de Usabilidad del apartado 3.1 y el código en [Processing](#) usado en el apartado 3.4.

El artículo presentado a continuación forma parte del Proyecto Fin de Máster del Máster de Software Libre de la Universidad Oberta de Catalunya.

2 Estado del Arte

El estudio del Estado del Arte de la investigación se centrará en:

- Web Semántica
- interfaces avanzados de visualización
- Usabilidad y Test de Usabilidad

Aunque el concepto de Web Semántica, en sus fundamentos teóricos y técnicos, queda fuera del objetivo de este proyecto, dado que se trata de un campo relativamente nuevo y en muchos casos no demasiado conocido, se realizará un breve estudio del estado del arte en este campo, con el objetivo de mostrar la base sobre la que se desarrollará el proyecto.

A continuación se realizará un estudio sobre el estado de los *interfaces* avanzadas de visualización.

Finalmente se realizará el estado del arte de la usabilidad y especialmente de los Test de Usabilidad, parte fundamental del proyecto.

Web Semántica

Actualmente, el método más frecuente para la búsqueda de información en Internet es los motores de búsqueda, tipo Google. Estos motores relacionan los términos introducidos en la consulta, mediante algoritmos más o menos certeros, con los documentos de la web que tienen indexados, y nos muestran los documentos **donde**

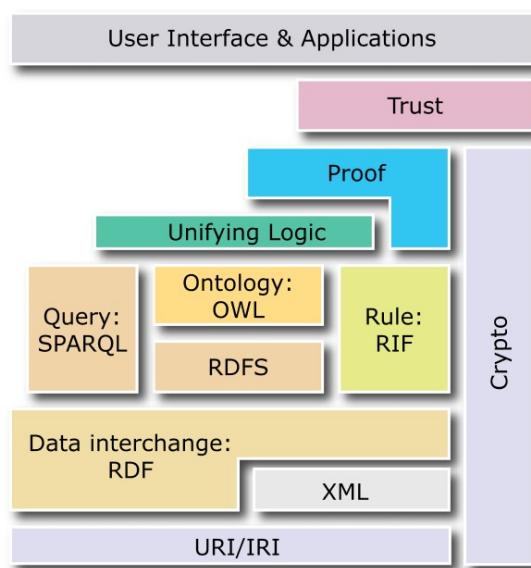
se encuentra la información que hemos solicitado, **no la información en sí** (Búsqueda Sintáctica).

Como contraposición a esto se encuentra la Web Semántica, concepto creado por Tim Berners-Lee, padre de la World Wide Web.

Se basa en crear una web que sea comprensible de manera formal, además de por los humanos, por las máquinas. Esto se lograría mediante la adición de metadatos a los contenidos de los documentos (páginas web, recursos, etc), y su objetivo sería la mejora en la búsqueda, compartición y combinación de la información disponible.

Para lograr este objetivo existen varias aproximaciones, dado que es un campo en el que hay una intensa actividad investigadora desde hace varios años. Para el desarrollo de este estado del arte, y teniendo en cuenta las limitaciones de espacio de la presente investigación, vamos a intentar ceñirnos lo máximo posible a las directrices del W3C.

El funcionamiento de la Web Semántica puede observarse en el siguiente gráfico:



Dicho funcionamiento podría resumirse en tres fases: Anotación, Ontologías y Razonamiento [1].

Anotación:

Se trataría del nivel más bajo, y consistiría en agregar metadatos semánticos a los contenidos, mediante la anotación semántica. Dicha anotación se realizaría mediante elementos conocidos como identificadores (*Uniform Resource Identifier -URIs-*), que se aplicarían sobre los diferentes elementos de las páginas web: palabras, cadenas de texto, imágenes, etc, y que lo relaciona con un concepto.

Para que esto se realice de forma estandarizada y resulte así más útil, existen una serie de tecnologías aprobadas por el W3C: RDF (*Resource Description Framework*), RDFa, GRDDL (*Gleaning Resource Descriptions from Dialects of Languages*) y POWDER (*Protocol for Web Description Resources*).

Ontologías:

Dado que las anotaciones pueden ser realizadas sin ningún tipo de restricción, es necesario establecer una serie de reglas que hagan que el vocabulario utilizado en las anotaciones sea común para un determinado grupo, de forma que dichas anotaciones resulten útiles y que el sistema pueda establecer relaciones entre ellas.

Esto es lo que se conoce como Ontologías: "La especificación formal y explícita de una conceptualización compartida"[2]

Las tecnologías aprobadas por el W3C que dan soporte a las Ontologías serían: OWL (*Ontology Web Language*) y SKOS (*Simple Knowledge Organization System*).

Razonamiento:

Se trata del desarrollo de motores de inferencia que pongan en relación las anotaciones realizadas en los contenidos con las ontologías, de forma que la máquina pueda utilizarlas para el razonamiento, y llegar a conclusiones lógicas, aprovechando así las posibilidades de la Web Semántica.

A su vez pueden dividirse en dos el razonamiento.

- Query: En el contexto de la Web Semántica, se refiere a tecnologías y protocolos usados para la obtención de información de una web estructurada de forma semántica.

Para esta tarea se utiliza SPARQL que permite realizar consultas a los tripletes RDF para obtener información de ellos.

- Inference: Se trata de la parte del proceso en el que el sistema es capaz de establecer nuevas relaciones entre los datos, basándose en los propios datos y en algunas informaciones adicionales, como ontologías. Para facilitar esta tarea se utiliza RIF (Rule Interchange Format) para establecer relaciones entre los diferentes vocabularios u ontologías.

interfaces avanzadas de visualización

Una interface avanzada de visualización es aquella en que la información es mostrada de forma gráfica, pudiendo el usuario interactuar con ella, adaptándola a sus necesidades u objetivos.

Desde el principio, uno de los principales usos de la web ha sido mostrar datos e informaciones a los usuarios. En un principio esta información se presentaba fundamentalmente en forma de texto e imágenes, pero de forma estática, de tal manera que el usuario no podía explorarla ni modificarla de forma libre.

Sin embargo la aparición de nuevas tecnologías ha posibilitado la creación de interfaces que permiten una interacción con los datos que muestran, posibilitando al usuario una comprensión mayor de la información[3].

Para el desarrollo de este tipo de interfaces fue fundamental la aparición de la tecnología Flash de Adobe, que permitió la realización de animaciones interactivas que dotaban a la web de una riqueza visual y una capacidad de interacción desconocida hasta ese momento.

Aunque durante muchos años ha sido considerado un *estándar de facto*, el hecho de ser una tecnología cerrada, así como problemas de seguridad, compatibilidad y uso de recursos, ha hecho que poco a poco vaya perdiendo relevancia.

La aparición de tecnologías de servidor, donde el proceso se realiza en la parte del servidor mostrando al usuario el resultado de dicho proceso, también ha influido notablemente en el desarrollo de este tipo de interfaces. Este tipo de tecnologías dotan a las interfaces de una interactividad y flexibilidad que las hacen atractivas y sobre todo útiles al usuario.

Así, lenguajes como Java, PHP, Python... facilitan la creación de aplicaciones dotadas de una interface gráfica rica e interactiva. Todos estos lenguajes son utilizados de forma intensiva en el desarrollo de esta clase de aplicaciones.

Mención aparte merece el lenguaje de programación Processing, un lenguaje y entorno de desarrollo basado en Java y desarrollado por Ben Fry y Casey Reas, que se distribuye bajo licencia GNU/GPL[4].

Este lenguaje está desarrollado para ser usado por personas con una fuerte formación estética, pero sin formación técnica en programación (arquitectos, diseñadores...). Este enfoque ha permitido a una gran cantidad de gente con

habilidades de diseño disponer de una herramienta de programación fácil de aprender, gratuita y que se ajusta a sus necesidades.

Como resultado de esto, han surgido infinidad de proyectos realizados con este lenguaje, de los que un alto porcentaje están orientados a la visualización de datos, siendo una de las aplicaciones de dicha visualización la realización de interfaces para buscadores, que permiten mostrar los resultados y las relaciones entre ellos de forma gráfica e interactiva.

En cuanto a lo que se espera a corto plazo en el campo de los interfaces avanzados de visualización, caben destacar dos tecnologías.

Por un lado la implementación del estándar HTML5, que permitirá aplicaciones e interfaces mucho más ricas e integradas en el navegador. Gran parte de esto se conseguirá gracias a la integración de nuevas APIs y su mejor funcionamiento con Javascript, que permitirá funcionalidades como arrastrar y soltar, trabajo off-line, geoposicionamiento, almacenamiento persistente local, integración de forma nativa de audio y vídeo... lo que producirá interfaces con capacidades muy superiores a las actuales. Además HTML5 incluye etiquetas específicas para la web semántica.

Por otro lado, todos los desarrollos que se están produciendo en el campo del Cloud Computing están teniendo su repercusión en el campo de las interfaces avanzadas, ya que por un lado permiten "externalizar" el procesado de los datos con servicios como los ofrecidos por [Amazon EC2](#), algo fundamental a la hora de manejar enormes cantidades de datos.

Sin salir del Cloud Computing existen también proyectos como [Many Eyes de IBM](#) y [Fusion Tables de Google](#), que están acercando la creación de este tipo de interfaces para la visualización de datos a todos los usuarios, dotándolos además de la dimensión de "comunidad", donde el resto de los usuarios se aprovechan de lo realizado por otros.

Usabilidad y Test de Usabilidad

La ISO/IEC 9241 define la usabilidad como "la eficacia, eficiencia y satisfacción con la que un producto permite alcanzar objetivos específicos a usuarios específicos en un contexto de uso específico".

Según esta definición los atributos de la usabilidad serían:

- **Eficiencia:** La rapidez con la que el usuario es capaz de alcanzar sus metas.
- **Eficacia:** La facilidad con la que el usuario es capaz de alcanzar sus metas.
- **Satisfacción:** Es la impresión subjetiva que el usuario obtiene del sistema.

Tanto la ISO/IEC 9126, que se centra en la usabilidad del software definiéndola como "la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso.", como las 5Es (Effective-Efficient-Engaging-Error tolerant-Easy to learn)[5] de la experta en usabilidad Whitney Quesenbery añaden nuevas dimensiones a la definición de usabilidad:

- **Tolerante a los errores:** La prevención de errores y la facilidad para recuperarse de los ocurridos.
- **Atractivo:** Cómo la interface resulta agradable para el usuario y le anima a usar el software.
- **Fácil de aprender y recordar:** La facilidad de aprender a usar el software y el recuerdo en el tiempo.

Para la medición de la usabilidad de un software y como herramienta para la mejora de la misma se utilizan los Test de Usabilidad, que pueden definirse como "La actividad que se centra en la observación de usuarios trabajando con un producto y realizando tareas reales y coherentes para ellos".

Hasta principios de los años 90, los Test de Usabilidad eran realizados por grandes empresas en laboratorios específicamente creados para tal labor y con una cantidad de participantes que oscilaba entre 30 y los 50 participantes por estudio.

A principio de los 90, varios investigadores, especialmente Jakob Nielsen y Tom Landauer realizaron una serie de experimentos con el objetivo de encontrar una forma más barata y rápida de realizar los Test de Usabilidad.

Según dichos experimentos, el mejor índice entre coste y beneficio se obtiene cuando se realiza el estudio sobre un cantidad de entre tres y cinco participantes.

Con esta cantidad de participantes se consigue encontrar el 85% de los problemas, y a partir del quinto participante, los problemas detectados vuelven a repetirse con los siguientes participantes, de forma que el coste para hallar el 15% de problemas restantes se dispara[6].

Este descubrimiento, junto a otros similares que se produjeron en la misma época, hicieron que se modificara el panorama de las pruebas de usabilidad, pasando a ser utilizadas por muchas más personas, instituciones y empresas.

Los efectos más inmediatos de esta forma de realizar los test fueron que podían ser incorporados al desarrollo del producto con un bajo coste, sin prácticamente afectar a los tiempos de desarrollo, de forma que podían realizarse desde el principio del mismo y repetirse varias veces a lo largo de este.

En función del momento de desarrollo en el que realicemos los test se pueden distinguir dos tipos:

Formativos: Su objetivo es diagnosticar y solucionar problemas. Suelen ser estudios pequeños que se repiten a lo largo del desarrollo.

Sumativos: Se realizan una vez finalizado el desarrollo del producto y se utilizan para comprobar que el producto ha alcanzado los objetivos establecidos. Suelen ser estudios de mayor tamaño que los formativos, ya que precisan de gran cantidad de datos para validar de forma estadística la consecución de los objetivos.

Dado que la cantidad de participantes en los test va a ser reducida, se suelen realizar perfiles de usuarios, de forma que los participantes pertenezcan todos a un mismo perfil, haciendo de esta forma los resultados más coherentes que si no hubiera un nexo entre los participantes.

A través de la realización de tareas, que se desarrollan en escenarios que las ponen en contexto, se podrá observar a los participantes relacionarse con el software, pudiendo extraer mediante diversas técnicas la información necesaria para mejorar la usabilidad del software.

La principal técnica para obtener información que se utiliza en este tipo de test consiste en la denominada "Think-aloud" en la cual el participante va comentando en voz alta todos sus pensamientos, de forma que nos comunica lo que está haciendo, lo que pretende hacer, cuales son los problemas y dudas que está teniendo, etc.[7]

Aunque es una técnica que requiere que el participante supere cierta vergüenza y falta de costumbre a expresar en voz alta lo que piensa, si se consigue realizar de forma correcta probablemente sea la más valiosa fuente de información. A través de este proceso podremos obtener de forma directa la información sobre los mapas mentales del participante, y ver donde nuestro desarrollo falla y donde tiene éxito.

Durante la realización de los Test de Usabilidad es también frecuente apoyarse en cuestionarios y entrevistas, que pueden llevarse a cabo antes de los test, después de cada tarea, y al final del test.

Dichos cuestionarios pueden ser creados por nosotros mismos, de forma que se adapten totalmente al test que se está realizando, o pueden utilizarse cuestionarios estándar, así como una mezcla de ambos.

La ventaja de los cuestionarios estándar es que han sido probados en infinidad de ocasiones, por lo que su efectividad es muy alta. Dos de los más usados son:

- *SUS (System Usability Scale)* desarrollado por John Brooke. Se trata de diez afirmaciones, en las que se van alternando las positivas con las negativas, y con las que el participante tiene que decir en qué medida se encuentra de acuerdo en una escala del uno al cinco. En función de las respuestas dadas por los participantes se obtiene un índice de usabilidad del software, que varía entre 0 y 100.
- *CSUQ (Computer System Usability Questionnaire)* desarrollado por James Lewis. Se trata de 19 afirmaciones con una escala del 1 al 7 con las que el participante muestra en qué medida está de acuerdo con dichas afirmaciones. Este cuestionario nos permite obtener la percepción de la usabilidad del participante en varios aspectos del software.

Aunque el método de *Think Aloud*, los cuestionarios, entrevistas, y principalmente la observación del participante durante la realización de las tareas a las que se le somete durante el Test de Usabilidad proporcionan la principal fuente de información a la hora de mejorar la usabilidad de un producto, existen muchas otras herramientas que son usadas junto a las anteriores de forma habitual, en mayor o menor medida.

A continuación realizaré un resumen de las más habituales, centrándome más en las que son más usadas por los investigadores:

Evaluación Heurística. Junto con los Test de Usabilidad es la herramienta más utilizada a la hora de realizar un estudio de usabilidad. Consiste en el examen del objeto del estudio por uno o más expertos. Este estudio se realiza la mayoría de veces según protocolos definidos, como las 10 reglas de usabilidad de Nielsen u otras similares, si bien también puede realizarse de forma más abierta.

Al tratarse de evaluaciones que son realizadas por expertos, aparecen de forma rápida problemas de usabilidad que pasarían desapercibidos a usuarios comunes.

La ventaja de este tipo de evaluaciones es que permiten detectar y solucionar una gran cantidad de problemas, de forma que al realizar los Test de Usabilidad con nuestros participantes habremos eliminado una gran cantidad de fallos y además tendremos una valiosa información a la hora de diseñar los escenarios y tareas, de forma que los Test serán mucho más efectivos.

Paso-a-paso cognitivo. Similar al anterior, un miembro del equipo de evaluación se pone en el papel del usuario para realizar una prueba del producto con el objetivo de identificar problemas en la facilidad de aprendizaje.

Diseño participativo. Se trata de involucrar al usuario en el proceso de desarrollo, pidiéndole que revise el producto en desarrollo y dé su opinión, o que ayude en la generación de conceptos de diseño.

Las anteriores herramientas son usadas en la fase de desarrollo del proyecto, si bien también existen otras diseñadas para etapas anteriores o posteriores, que pueden ayudar a mejorar la usabilidad del software, tales como encuestas, estudios de mercado, información interna de la empresa (llamadas al servicio de asistencia técnica, *feedback* de los comerciales...), etc.

Para finalizar con el Estado del Arte de la usabilidad y las pruebas de usabilidad, comentar que tanto el desarrollo de nuevos dispositivos (smartphones, tablets...) como de nuevas herramientas, han hecho que, en los últimos años, por un lado se hallan ampliado los dispositivos sobre los que realizar las pruebas de usabilidad y, por otro, estas puedan realizarse con mejores medios técnicos, tales como aparatos de seguimiento de la mirada, equipos de grabación mucho más potentes y asequibles, software de apoyo para las sesiones de test...

Igualmente ya no es estrictamente necesario que los test se realicen de forma presencial, permitiéndonos las nuevas aplicaciones e Internet poder realizarlos a distancia (dirigidos o no), por videoconferencia, mediante aplicaciones web, etc.

3 Desarrollo

3.1 Test de Usabilidad

Como se ha visto en el apartado dedicado al Estado del Arte de la usabilidad, esta es una herramienta extremadamente eficaz a la hora de localizar errores e identificar necesidades.

La metodología para la realización de los test está basada en los Discount Test descritos por Nielsen y Landauer que se desarrollaron según la metodología descrita por Carol M. Barnum en su libro Usability Testing Essentials.

En primer lugar se realizó una Evaluación Heurística, siguiendo el guión de Nielsen y Molich, en el que se dan 10 directrices para evaluación de un *interface*, dividiéndolo en apartados.

Los resultados obtenidos se exponen a continuación:

Resultados de la Evaluación Heurística

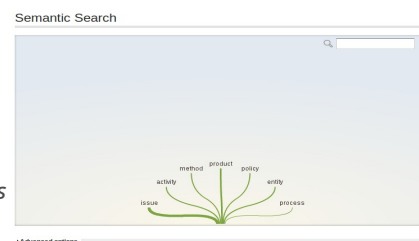
1) Visibilidad del estado del sistema

La información que el sistema ofrece al usuario acerca de sí mismo.

a) Al entrar en el buscador inteligente no hay nada que nos indique qué es lo que debemos hacer o cómo comenzar a realizar la búsqueda. Tampoco recibimos ningún tipo de indicación sobre la utilidad de cada uno de los elementos que nos encontramos.

Recomendación:

Al iniciar, mostrar antes de hacer el primer click carteles semitransparentes ofreciendo indicaciones o ayuda.



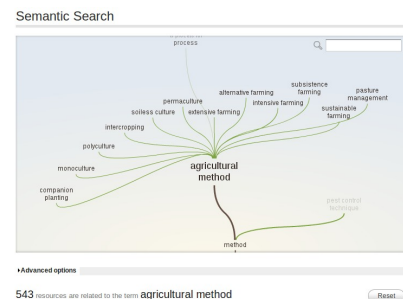
2) Relación entre el sistema y el mundo real

La comunicación con el usuario debe hacerse en un lenguaje y con unas convenciones que le resulten familiares.

a) La estructura en árbol es correcta, ya que es una forma de representación habitual a la hora de mostrar elementos relacionados entre sí de alguna forma (árboles genealógicos, por ejemplo), pero cuando se sube a un cierto nivel, puede llegar a perderse el hilo de la búsqueda, a pesar de poder navegar mediante el *scroll*.

Recomendación:

Utilización de breadcrumbs para seguir el hilo; Utilización de navegación tipo Pan para desplazarse por el interface.



3) Control y libertad del usuario

Cómo el usuario puede acceder a funciones por error, y cómo puede retroceder en dichas elecciones: opciones de hacer y deshacer.

a) En principio, la forma de navegar es correcta ya que permite ir hacia atrás en el árbol. Igualmente resulta correcto el botón de 'Reset', que permite devolver el buscador semántico a su estado inicial.

b) En cualquier caso no permite la navegación tipo *pan* que permite moverse por el *interface* del buscador en las cuatro direcciones usando el ratón, permitiendo solo subir y bajar con el *scroll*, por lo que muchas veces partes del árbol quedan fuera de la pantalla y la única forma de verlos es haciendo click en un nivel inferior del árbol para que se recoja, con lo cual se modifica la búsqueda y por tanto los resultados.

Recomendación:

Implementar la navegación por la ventana de forma que se pueda ver el árbol y las relaciones sin modificar la búsqueda.

c) El botón 'Más resultados' que se encuentra al final de la página una vez que se muestran los resultados, nos lleva a la página de *búsqueda por texto* con los elementos buscados en el Buscador Semántico.

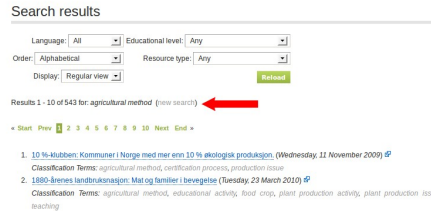
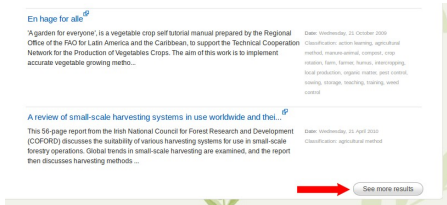
Esto provoca dos problemas:

Por un lado al pasar al buscador por texto no se muestran las relaciones semánticas, que son la base del *buscador inteligente*, mostrando solo los términos de búsqueda.

Por otro, la única forma de volver al buscador semántico es mediante el *link* 'Nueva búsqueda', que: [1] está mal situado, [2] no se percibe como link, [3] no indica claramente su función, [4] produce confusión en relación al botón, situado sobre él y mucho más visible, de 'Recargar', y [5] devuelve a un buscador semántico reseteado, que no conserva la búsqueda realizada hasta pulsar el botón de 'Más resultados'.

Recomendación:

Hay que repensar la opción 'Más resultados': integrarla en la misma página, incluir los botones de refinamiento en búsqueda avanzada...



4) Consistencia y estándares

Es la forma en la que se siguen convenciones ampliamente aceptadas, de forma que el usuario sepa de forma intuitiva realizar tareas comunes.

a) Al tratarse de una interface novedosa no se rige por estándares que sean comúnmente aceptados. A pesar de que algunos sistemas de búsqueda de información visual empiezan a ser comunes, todavía ninguno de ellos es conocido para el gran público.

Pese a todo ello, si que respeta algunas convenciones básicas tales como la caja de búsqueda en la esquina superior derecha, la forma en que los resultados son mostrados al estilo de los buscadores web y la opción de acceder a más resultados situada al final de todos los resultados.

Recomendación:

Intentar usar estándares cuando sea posible.

b) Se podría enmarcar dentro de este epígrafe el círculo que se muestra en la esquina superior izquierda del *interface* y en la zona de resultados mientras se cargan los resultados, al estilo de los círculos que aparecen mientras se cargan los videos en servicios tipo *Youtube*, y que son reconocidos como un indicador del proceso de carga, además de servir para indicarnos si ha habido algún error o bloqueo durante la misma.

5) Prevención de errores

Un sistema que no prevenga los errores o trate de minimizarlos pidiendo confirmación antes de llevar a cabo acciones que puedan provocarlos.

a) El botón 'Reset' está colocado en un lugar, junto a las opciones avanzadas, que puede provocar su pulsación por error, o asumir que su acción solo afectará a los cambios realizados en el cuadro de búsqueda avanzada.

Recomendación:

Recolocar el botón para que no sea pulsado por equivocación y diseñarle de forma que explique de forma más explícita su función.



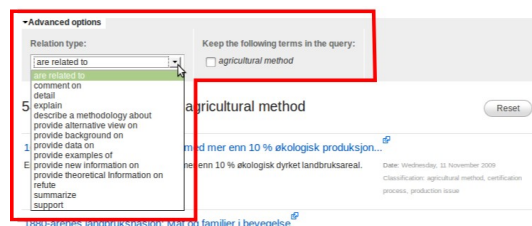
6) Reconocer mejor que recordar

Minimizar el uso de la memoria haciendo que los objetos, acciones, instrucciones, información, etc. sea reconocible mediante un diseño adecuado, en lugar de forzar al usuario a recordar como realizar las tareas necesarias para alcanzar su objetivo.

Las opciones de búsqueda avanzada en las que se pueden establecer las relaciones semánticas que son la base del buscador aparecen por defecto minimizadas, y su uso no es claro, ya que no se ven de forma clara e intuitiva las opciones y posibilidades de este tipo de búsqueda, que son la base del buscador.

Recomendación:

Dar relevancia a las opciones de búsqueda avanzada haciendo también que su uso resulte más intuitivo.



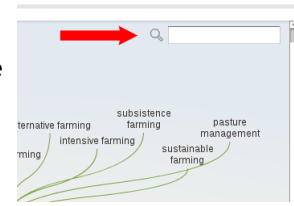
7) Flexibilidad y eficiencia de uso

Atajos de teclado y aceleradores, así como la posibilidad de personalizar las tareas más comunes, que permiten al usuario ser más eficiente en la realización de su tarea.

Existe en el buscador, en la parte superior derecha, un recuadro de búsqueda que permite una búsqueda más directa de los términos, pero exige que los términos introducidos sean exactos. Esto requiere un conocimiento por parte del usuario bastante aproximado de lo que está buscando a pesar de que ofrezca sugerencias.

Recomendación:

Dotar a la caja de búsqueda de mayor flexibilidad, de forma que realmente suponga un atajo.



8) Estética y diseño minimalista

Ausencia de toda aquella información que no resulte relevante para el usuario, y que por lo tanto entorpezca su uso.

a) Al navegar por la estructura de árbol, hay algunos conceptos que se abren en muchas ramificaciones. Esto provoca que se superpongan los conceptos, lo cual hace que tanto la lectura como la navegación se complique.

Recomendación:

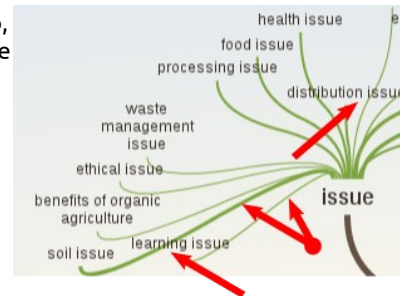
Ordenar la forma en la que se muestran los conceptos de forma que sea clara y que estos no se mezclen.

b) Para mostrar la cantidad de recursos que existe en cada concepto, se ha utilizado como recurso gráfico el grosor de la línea que los une (las ramas de árbol) de forma que cuanto más gruesa es, más recursos van a encontrarse.

Este sistema no es eficiente, ya que en absoluto es intuitivo por no apreciarse muchas veces la diferencia de grosor entre unas y otras ramas del árbol, y tampoco ofrece una información visual clara y rápida como debería ser su objetivo.

Recomendación:

Habría que buscar una forma más visual de mostrar esto, utilizando otros recursos gráficos.



9) Ayuda para reconocer, diagnosticar y recuperarse de errores

Los mensajes de error deben expresarse en lenguaje sencillo, y sugerir una solución.

Cuando se realiza una búsqueda en la caja de 'Búsqueda por Texto' y el término buscado no existe o no se encuentra, no devuelve ningún mensaje de error ni realiza ninguna acción.

Recomendación:

Se debe mostrar algún mensaje de error cuando no se encuentre el término, e incluso dar opciones de búsqueda, de forma que se guíe al usuario.

10) Ayuda y documentación

Aunque sea mejor el uso de un sistema sin necesidad de documentación, en caso de ser necesaria esta debe de ser fácil de encontrar, centrada en el problema y que muestre de forma clara los pasos a seguir.

Una vez en el buscador, no existe ningún tipo de ayuda o de *link* a algún tipo de tutorial.

Lo único que existe en este sentido es un botón en la portada de la web que nos lleva a una explicación de cada uno de los sistemas con un pequeño vídeo sin sonido donde se hace una insuficiente demostración de una búsqueda.

Recomendación:

Habría que introducir links a documentos de ayuda -wikis-, ventanas emergentes con explicaciones, etiquetas flotantes al dejar el ratón sobre los elementos del buscador, etc.

La realización del Análisis Heurístico nos proporciona, por un lado, una aproximación a la usabilidad del *interface* desde el punto de vista del experto que nos permitirá detectar errores y necesidades para poder ser subsanados e implementados en fases posteriores. Por otro lado, nos proporciona la base sobre la que crear los Test de Usabilidad con usuarios, de forma que podamos realizar las pruebas y cuestionarios más ajustados a los requisitos del estudio.

El estudio de usabilidad sobre el Buscador Inteligente de Organic.Edunet se realizó en noviembre de 2011, por el autor del artículo, en los diferentes institutos de la provincia de Burgos donde los participantes en el mismo ejercían la docencia.

El objetivo del estudio era determinar el grado de usabilidad de dicho buscador, y establecer los problemas que se encontraban los usuarios, así como detectar nuevas necesidades y mejoras para aumentar la usabilidad del mismo, con el objetivo de poder implementarlas mediante el uso de modelos y tecnologías abiertas.

Se seleccionaron cuatro participantes siguiendo los siguientes criterios:

- Los participantes tenían que ser profesores de educación secundaria.
- Su área de conocimiento en la que impartían clases tenía que estar relacionada con la agricultura, biología, conocimiento del medio...
- Los participantes tenían que tener conocimientos medios en el uso de Internet y herramientas informáticas.
- No debían conocer de antemano el buscador.

Basándose en los resultados obtenidos en el Análisis Heurístico y en los requerimientos de los participantes se desarrollo un usuario tipo y un escenario base, a partir de los cuales se crearon las tareas que conformaron los Test de Usabilidad.

Dichos test consistieron en cinco escenarios en los que se les solicitaba que realizaran diferentes tareas, expresando en voz alta sus intenciones, dudas, problemas, etc. de forma que estos pudieran ser registrados para su posterior análisis.

Al final de cada escenario se les realizaba una encuesta en la que se les solicitaba valorar en una escala del 1 al 5 (Completamente en desacuerdo – Completamente de acuerdo) una serie de cuestiones relacionadas con las tareas que acababan de realizar, incluyendo también preguntas abiertas.

El Test concluía con la realización de una encuesta SUS, a fin de valorar el grado de usabilidad del buscador.

A continuación se muestra un resumen de los hallazgos más importantes encontrados en cada uno de los escenarios durante la realización de los test.

ESCENARIO 1: Se le muestra el interface del buscador y se le deja que la observe durante un rato. A continuación se le realizan las siguientes preguntas: ¿Qué tipo de aplicación es?; ¿Que cree que se puede hacer aquí?; Observe el interface y diga, ¿qué crees que ocurrirá si haces click en los diferentes elementos que hay?; ¿Qué sería lo primero que harías?; ¿Hay algún elemento que no entienda?; ¿Conocía alguna aplicación así?; ¿Cual es su primera impresión general? El objetivo de este escenario es conocer la primera impresión (look&feel) del buscador, y tratar de conocer cuales son los elementos más intuitivos del mismo y cuales no.

| Usuarios afectados | Hallazgos | Gravedad |
|--------------------|---|----------|
| 1/4 | Lo consideran limpio y organizado. | POSITIVO |
| 3/4 | No identifican la función del buscador. | GRAVE |
| 3/4 | Consideran el <i>interface</i> pobre y sin explicaciones. | GRAVE |
| 1/4 | Consideran el <i>interface</i> demasiado pequeña. | MEDIO |
| 1/4 | Comenzarían haciendo una búsqueda en la caja. | MEDIO |
| 3/4 | Comenzarían pinchando en el concepto "Asunto" por ser el más destacado. | BAJO |

ESCENARIO 2: Precisa buscar recursos para dar una clase sobre agricultura ecológica, haciéndolo a través del Buscador Inteligente. El documento deberá estar preferentemente en español y ser un documento de texto.

Con este escenario se pretende conocer la usabilidad general del buscador, con tareas de búsqueda básicas sobre temas de los que existen abundantes recursos y con unos parámetros de refinamiento básicos, como el idioma y el tipo de recurso.

| Usuarios afectados | Hallazgos | Gravedad |
|--------------------|--|----------|
| 4/4 | No percibe la existencia del botón 'Más Resultados'. | GRAVE |
| 2/4 | Uso de la caja de búsqueda introduciendo términos libremente. No | GRAVE |

| | | |
|-----|--|-------|
| | perciben que no da resultados. | |
| 1/4 | No se da cuenta de que aparecen los resultados en la parte inferior de la página. | GRAVE |
| 4/4 | Uso del botón 'Reset' de forma errónea, perdiendo los resultados de la búsqueda realizada. | MEDIO |
| 2/4 | Problemas para la navegación por el árbol. | MEDIO |
| 2/4 | No aparecen resultados ni ningún mensaje al utilizar la caja de búsqueda. | MEDIO |
| 1/4 | Superposición de términos en el árbol. | MEDIO |

Además, mediante los cuestionarios Post-Tarea también descubrimos que:

- A medida que lo van utilizando les va pareciendo más útil.
- Que la cantidad de información a veces es demasiada y confusa.
- Que es más interesante para temas concretos que otro tipo de buscadores.

Igualmente, mediante los comentarios realizados en alto (*Think Aloud*) mientras realizaban las tareas, se desprende que el movimiento del árbol al abrirse les resulta atractivo a todos ellos.

ESCENARIO 3: Comenzando de nuevo en la pantalla principal del buscador inteligente, buscará ahora recursos sobre turismo ecológico o relacionados con la enseñanza para relacionarlos con la agricultura ecológica (por ejemplo para preparar una excursión didáctica). Para ello necesitará acotar la búsqueda a recursos que expliquen o den información sobre ecoturismo o enseñanza. En este caso no será imprescindible que los recursos sean en español.

En este escenario lo que se pretende es ver el manejo de las opciones avanzadas de búsqueda, donde se pueden establecer las relaciones ontológicas entre conceptos.

| Usuarios afectados | Hallazgos | Gravedad |
|--------------------|---|----------|
| 4/4 | No percibe la existencia del botón 'Opciones Avanzadas'. | GRAVE |
| 4/4 | No comprende el uso de las opciones semánticas de las 'Opciones Avanzadas'. | GRAVE |
| 1/4 | No aparecen las opciones avanzadas hasta que no se selecciona un término del árbol. | MEDIO |

Mediante los cuestionarios Post-Tarea y los comentarios realizados por los participantes mientras realizaban las tareas, vemos que las opciones avanzadas, necesarias para obtener las características de web semántica del buscador, son difíciles de localizar, y una vez localizadas no son entendidas, y por lo tanto no utilizadas, por lo que se desaprovecha uno de los puntos fuertes del buscador.

ESCENARIO 4: Una vez que ya se tiene información sobre todos los temas anteriores, ahora se deberá buscar recursos multimedia (imágenes, audio, video...) relacionados con la agricultura ecológica, el ecoturismo y la enseñanza para ilustrar las clases.

Con este escenario se pretende ver la facilidad par eliminar el tipo de recurso más común (documentos) y refinar las búsquedas a los recursos multimedia.

| Usuarios afectados | Hallazgos | Gravedad |
|--------------------|--|----------|
| 3/4 | Una vez se encuentran los campos para refinar la búsqueda, no presenta problemas. | POSITIVO |
| 4/4 | Dificultad para llegar a las opciones de refinamiento de la búsqueda a través del botón de 'Más Resultados'. | GRAVE |
| 2/4 | Problemas para volver al buscador inteligente. | MEDIO |
| 2/4 | No ve el desplegable de 'Tipo de Recurso'. | MEDIO |

Un comentario bastante interesante, resultado del Think Aloud de uno de los participantes, fue la sugerencia de la inclusión de una miniatura junto a los resultados a la hora de buscar recursos multimedia.

ESCENARIO 5: Se realizarán las búsquedas anteriores utilizando el resto de sistemas de búsqueda que ofrece la página, comentando las ventajas e inconvenientes respecto del buscador inteligente. El objetivo es ver las diferencias entre unos sistemas y otros, y observar cuales son las preferencias de los usuarios.

Los resultados de este escenario, comparando el Buscador Inteligente con el resto de buscadores disponibles en el portal Organic.Edunet arrojaron los siguientes resultados:

A continuación se mostrará también el resultado de las encuestas SUS (System Usability Scale) Post-Test, con el objetivo de tener una visión global de la usabilidad de la aplicación.

| | Total desacuerdo | | | Totalmente de acuerdo | |
|---|------------------|---------|-----|-----------------------|-----|
| | 1 | 2 | 3 | 4 | 5 |
| 1. Creo que me gustaría usar este sistema frecuentemente | | | 4 | 3 | 1,2 |
| 2. Encuentro el sistema innecesariamente complejo | | 1,2,4 | 3 | | |
| 3. Creo que el sistema fue fácil de usar | | 1 | | 2,3,4 | |
| 4. <i>Creo que necesitaría asistencia técnica para poder utilizar el sistema</i> | 1,2 | 3 | 4 | | |
| 5. <i>He encontrado varias funciones en el sistema que estaban bien integradas</i> | | | 4 | 3 | 1,2 |
| 6. <i>Creo que había demasiada inconsistencia en el sistema</i> | 2 | 1,3 | 4 | | |
| 7. <i>Creo que la mayoría de la gente aprenderá rápidamente a utilizar el sistema</i> | | | 2,3 | 1,4 | |
| 8. <i>Encuentro que el sistema es demasiado incómodo de usar</i> | | 1,2,3,4 | | | |
| 9. <i>Me siento muy cómodo usando este sistema</i> | | | | 1,2,3,4 | |
| 10. <i>Necesitaría aprender varias cosas antes de hacerme con el sistema</i> | 1,2 | 3,4 | | | |

Las celdas amarillas indican el punto de la escala escogido por mayor número de participantes.

La valoración global de cada una de las cuestiones, arroja en todos los casos un balance positivo.

La valoración media de usabilidad obtenida de los cuatro participantes en la encuesta SUS es de 75 en una escala de 100 puntos.

Conclusiones de los Test de Usabilidad

Tanto el Análisis Heurístico -el punto de vista del experto-, como los Test de Usabilidad -el punto de vista de los usuarios-, se complementan como herramientas que son de un concepto más grande que es la Usabilidad, y gracias a ellos han ido apareciendo los diferentes problemas, de diferentes ámbitos, y con diferentes grados de severidad, que actualmente tiene el Buscador Inteligente.

En última instancia estas herramientas generan una serie de recomendaciones para ser implementadas en la siguiente iteración del buscador. El grado de prioridad no solo hace referencia a la urgencia de implementarlas cuanto antes, sino a la complejidad que conlleva cada una de ellas. Así, aunque un rediseño estético completo resultaría deseable, la complejidad de la tarea recomienda que se solucionen antes otras cuestiones, abordando esta con más planificación.

Pero este tipo de test no solo sacan a la luz los problemas y carencias de una aplicación, sino que a través de ellos hemos podido ver todo lo que es útil, gusta y funciona. Así se puede decir que los usuarios encontraron la aplicación atractiva y útil, gustándoles la forma de funcionar y el aspecto novedoso y atractivo de la misma.

Por último, recordar que este tipo de test no se deben hacer únicamente al final del desarrollo, sino que deben repetirse a lo largo del mismo y durante la vida de la aplicación, de forma que una vez encontradas y subsanadas las carencias, vuelvan a hacerse con el objetivo de una mejora y evolución continua de la aplicación. Y es en este punto, como se verá más adelante, donde el uso de tecnologías y modelos abiertos pueden marcar la diferencia frente a tecnologías cerradas.

3.2 Mockups

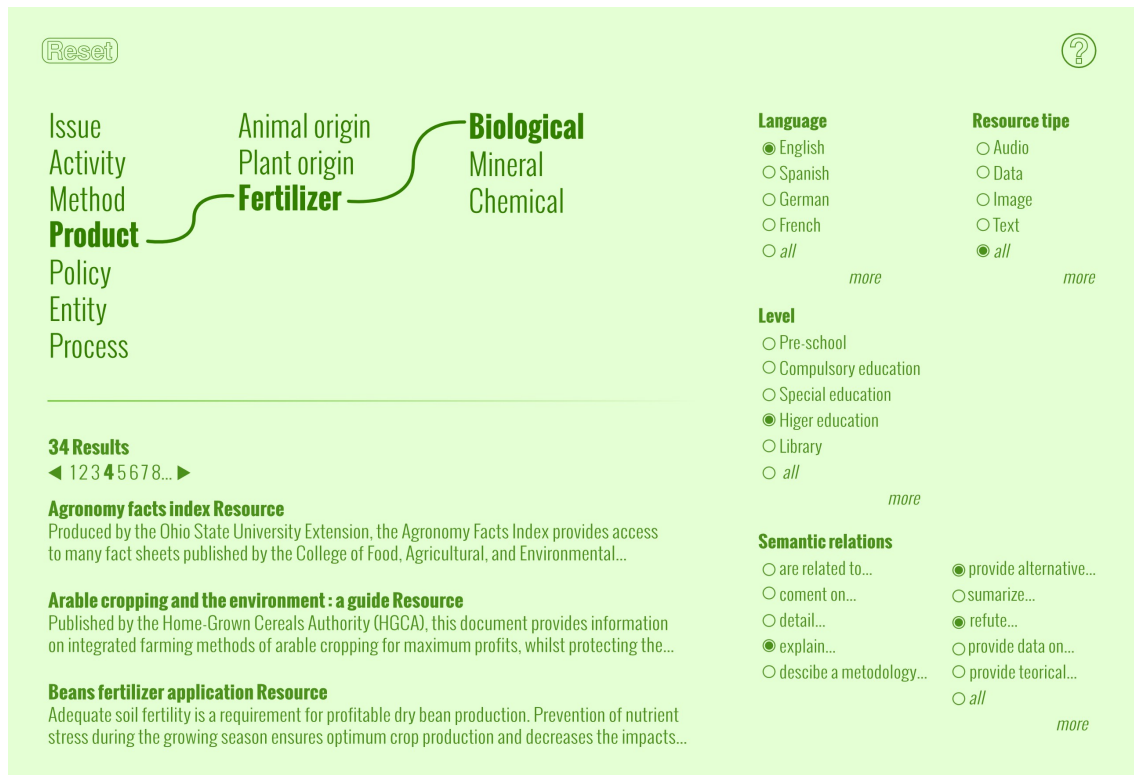
Una vez realizados los Test de Usabilidad, analizados los resultados y obtenidas las recomendaciones, poseemos una valiosa información que podemos aplicar en la mejora y evolución de nuestra aplicación.

Utilizando todos estos datos se han diseñado un conjunto de *Mockups* que nos sirven para aplicar y comprobar, de forma rápida, cómo podrían ser implementados e integrados los hallazgos encontrados en los test.

Mockup: es un modelo a escala o a tamaño real de un aparato o diseño utilizado para la enseñanza, demostración, evaluación del diseño, promoción u otros propósitos. Se diferencia del prototipo en que este último aporta, al menos, alguna funcionalidad del sistema.

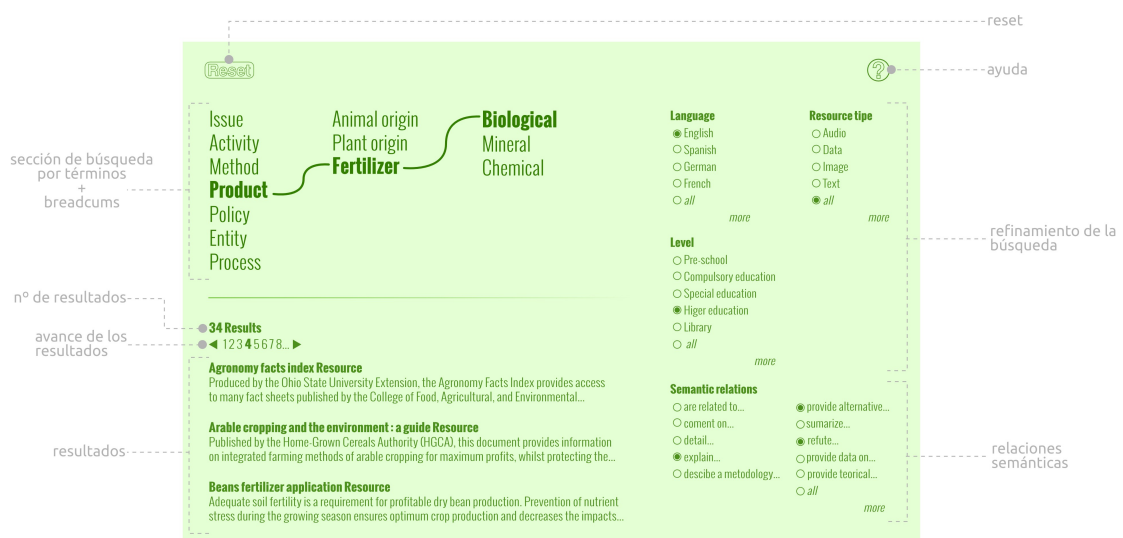
A continuación se presentan y comentan una serie de *mockups* con propuestas para la mejora y rediseño de la aplicación.

MOCKUP 1.0



Este primer *mockup* es una reinterpretación de Buscador Inteligente que existe actualmente en el portal Organic.Edunet.

Se ha mantenido la sencillez y la limpieza del diseño. Sin embargo, se han implementado muchos de los requisitos que se vieron necesarios en los Test de Usuario.

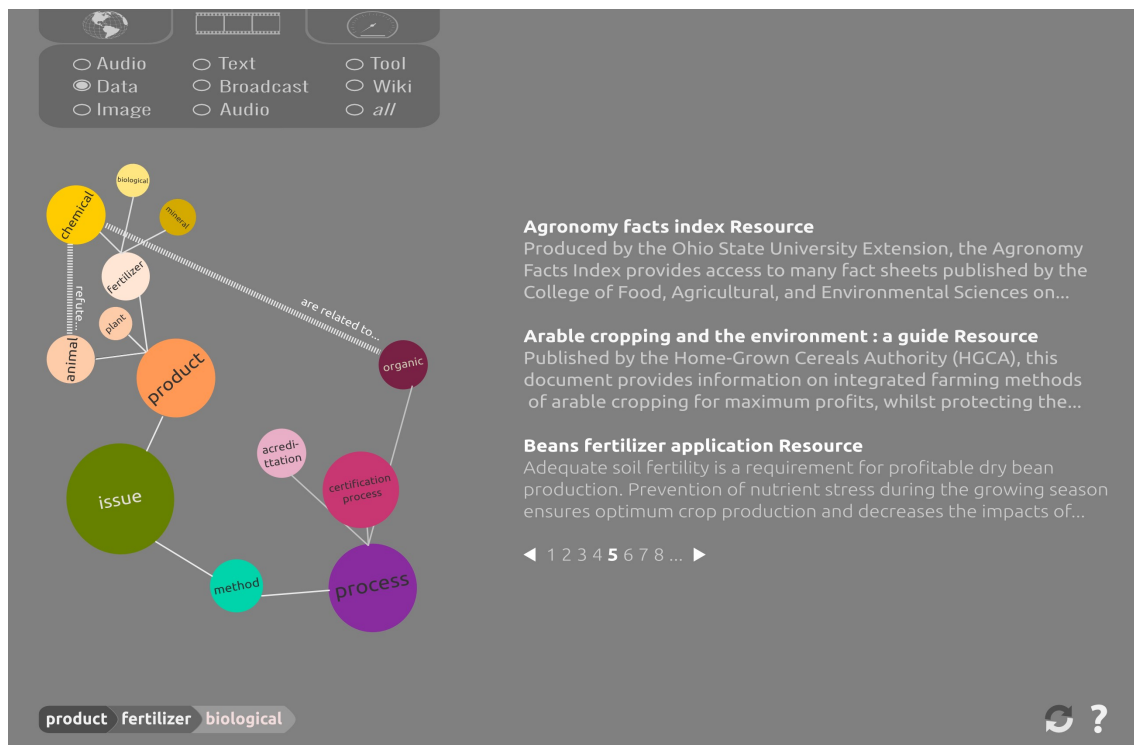


Como puede verse en la imagen, ahora es mucho más fácil acceder a las opciones de refinamiento de la búsqueda, las relaciones semánticas adquieren importancia y la navegación a través de los conceptos gana en claridad.

Una versión simplificada de este diseño es la que se utilizará en el apartado 3.4 Implementación para mostrar de forma práctica la implementación de requisitos usando tecnologías abiertas.

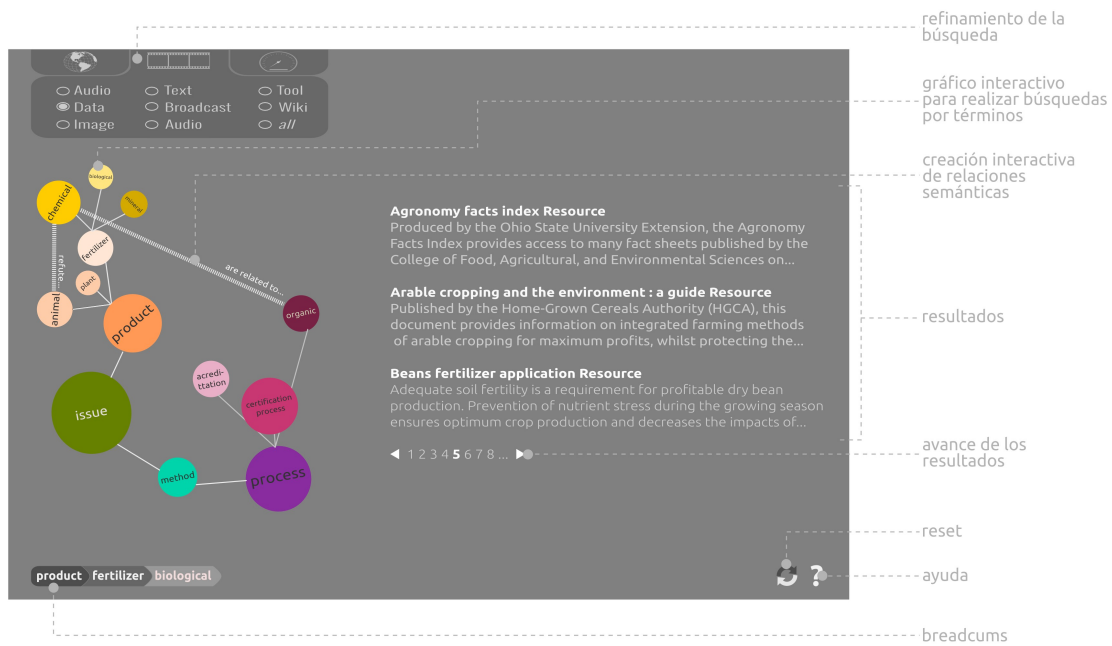
MOCKUP 2.0

Para este *mockup* se ha realizado un rediseño completo del buscador, no solo estético sino también en el concepto de las búsquedas.



Con un diseño mucho más actual, la búsqueda se realiza en este caso a través de un *interface* animado, donde el usuario de forma interactiva puede ir viendo los conceptos, la importancia, los recursos disponibles de cada uno, la familia a la que pertenecen, etc.

Además las relaciones semánticas se crearían también de forma interactiva uniendo los distintos círculos, de forma que se simplifica la tarea de explicar el cómo y el porqué de esta función, ya que pasa a realizarse de una manera natural.



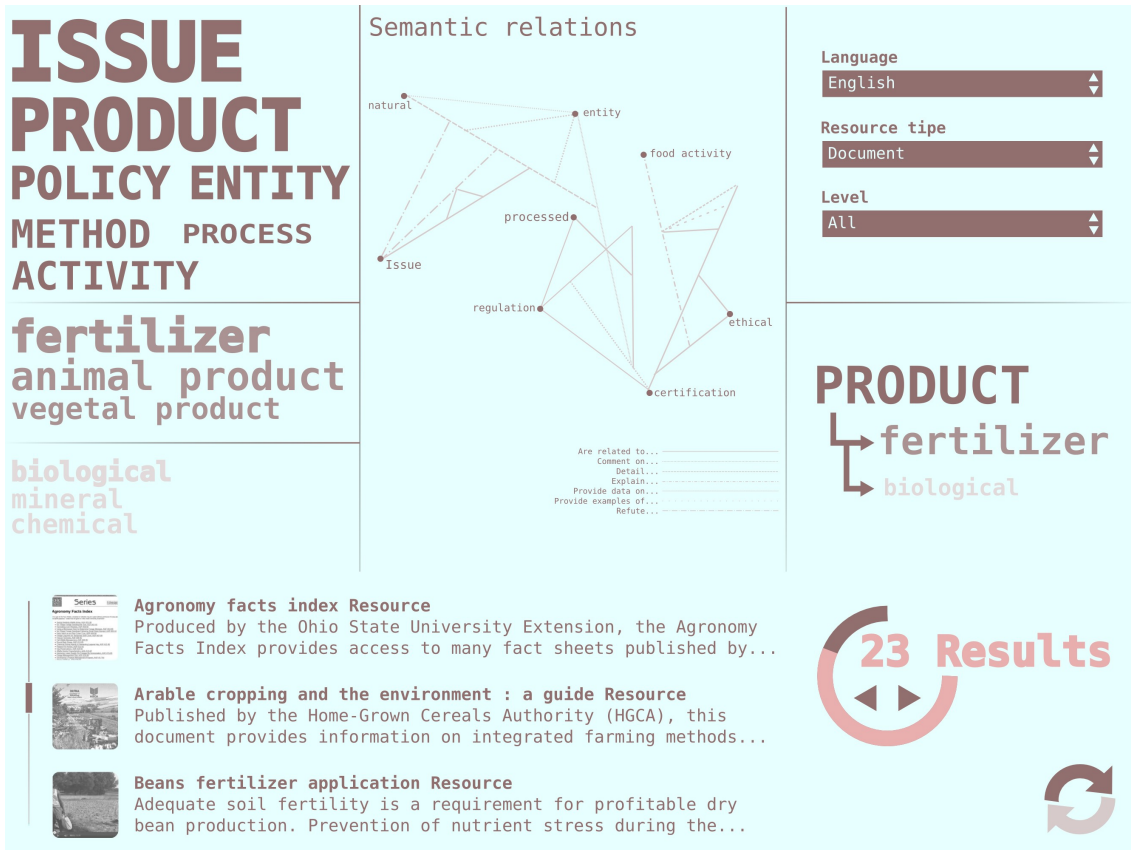
Al igual que en el caso anterior, se han implementado las recomendaciones del Estudio de Usabilidad. En este caso las opciones de refinamiento se han agrupado en pestañas, de forma que aun siendo accesibles desde el propio buscador, no molesten si no están siendo usadas.

En este ejemplo se ha pretendido mostrar las posibilidades estéticas y funcionales que nos ofrece el campo de la visualización de datos, permitiéndonos salir de representaciones más ortodoxas o clásicas y experimentar con nuevas formas de presentar los datos y de interactividad que hacen que podamos disfrutar de experiencias más atractivas, interesantes, intuitivas y útiles.

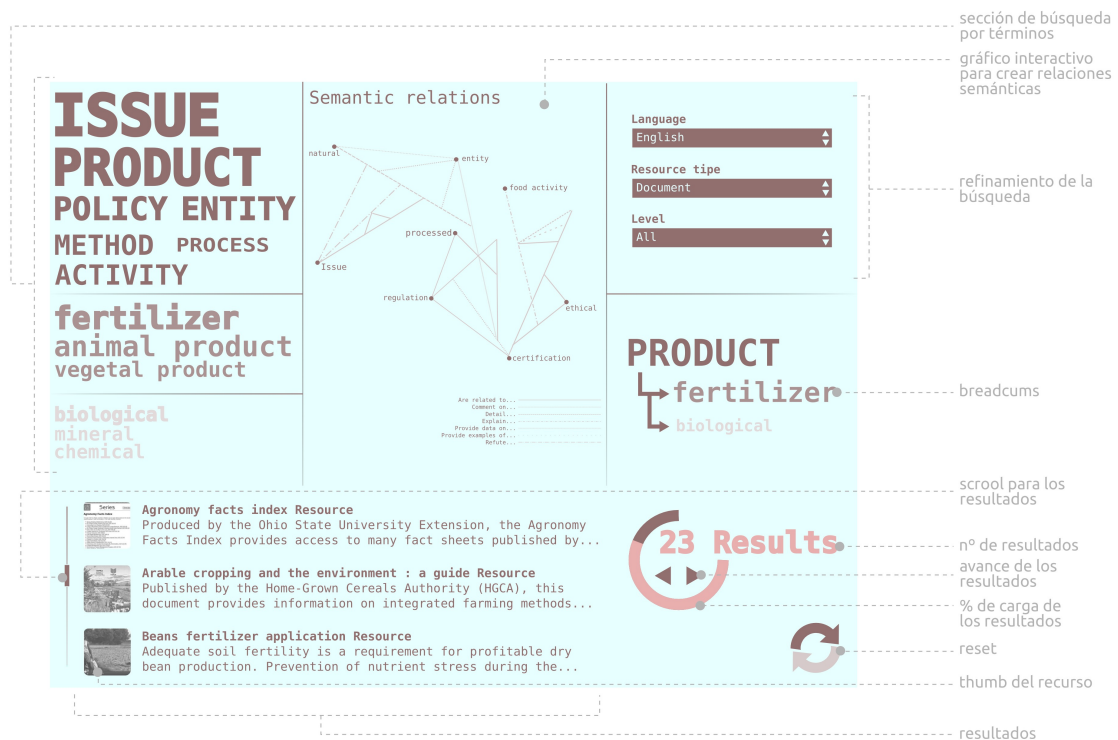
Para un rediseño de estas características una herramienta como el lenguaje de programación Processing sería probablemente la alternativa más interesante.

MOCKUP 3.0

Por último, y en la línea del *mockup* anterior, se presenta otra variante de lo que podría ser el *interface* del Buscador Inteligente en el que se sigue explorando las posibilidades gráficas y de concepto.



En este caso se ha optado por realizar una interface donde la tipografía adquiere especial relevancia estética, y es utilizada también como medio para ofrecer información relativa a los términos buscados.



El resto de elementos de la interface destacan por la sencillez y contundencia de sus formas, ofreciendo un aspecto limpio, sencillo y elegante, que muestra la información de manera clara y directa.

Entre los requerimientos que no se encuentran en otras propuestas están las miniaturas junto a la descripción del recurso y un indicador de carga de resultados.

Con el área de relaciones semánticas se pretende centrar la importancia del buscador en las posibilidades de este tipo de tecnología, y que lo convierten en un buscador especial en comparación con otros. A través de dicha área se podrían establecer relaciones semánticas complejas entre los diferentes conceptos de forma interactiva, intuitiva y visual.

3.3 Justificación del uso de Software Libre

El caso concreto del Buscador Inteligente de Organic.Edunet es un perfecto ejemplo de cuándo y porqué el uso de tecnologías y modelos de desarrollo libres debe ser prioritario en determinados casos.

Al intentar justificar la preferencia por el Software Libre convendría separar los motivos filosóficos de los técnicos, ya que a veces la mezcla de ambos provoca confusión a la hora de entender el porqué de esta decisión.

En primer lugar, el proyecto Organic.Edunet es un proyecto público, financiado con fondos de la Unión Europea, por lo que los desarrollos realizados en el marco de dicho proyecto deberían poder ser usados, estudiados, modificados, y poder distribuir las mejoras (4 libertades del Software Libre).

Se trata también de un proyecto desarrollado en un entorno académico, que es llevado a cabo por diferentes universidades de la Unión Europea. Este entorno facilita enormemente la colaboración, así como la reutilización de los recursos creados en el desarrollo del proyecto. El no aprovechar esta ventaja mediante el uso de Software Libre supondría un desperdicio de recursos.

Por otro lado, si nos fijamos en el fin último del portal, veremos un caso claro de lo que Bruce Perens llama *los elementos diferenciadores y la ventaja competitiva*[10].

El portal Organic.Edunet pone a disposición de los usuarios *“recursos de calidad sobre Agricultura ecológica y Agroecología”*. En este contexto, el Buscador Inteligente supondría una forma más cómoda, visual o intuitiva de encontrar la información deseada, pero el elemento diferenciador del portal frente a otros portales similares o frente al propio Google reside en la calidad de los contenidos ofrecidos, y no en el buscador.

Por este motivo carece de sentido el invertir una gran cantidad de recursos en el desarrollo de las diferentes interfaces de búsqueda, cuando dichos recursos deberían centrarse en la calidad de los contenidos.

Si se utiliza el modelo de Software Libre, una vez liberada la interface, esta podrá ser aprovechada por el resto de la comunidad. Esto incluye a otras universidades, iniciativa privada, particulares... que adaptarán el *interface* a sus necesidades corrigiendo errores y añadiendo funcionalidades, por lo que se libera, al menos en gran parte, al desarrollador original del mantenimiento de la aplicación, pudiendo aprovechar lo realizado fuera.

Tratándose además, como es el caso, de una aplicación novedosa en varios aspectos (web semántica, visualización de datos...) para la que no existen opciones que puedan ser utilizadas libremente, su liberación supondría una aportación importante que contribuiría al desarrollo de dichas áreas.

En lo referente al aspecto técnico, las ventajas del Modelo Libre son también evidentes.

El actual desarrollo del Buscador Inteligente fue realizado por encargo y entregado como binario en formato Flash, por lo que no se dispone del código, y cualquier tipo de modificación o mejora tendría que realizarse a través del autor original con el uso de recursos económicos y de tiempo que ello conlleva.

Como se ha visto en los puntos anteriores del artículo, este tipo de aplicaciones está en constante evolución. El uso de las herramientas de Usabilidad se hace imprescindible en este tipo de desarrollos, y este tipo de herramientas provocan que tengan que realizarse constantes cambios y mejoras en la aplicación. Por ello, tener que recurrir a recursos externos no parece una opción viable.

Pero aunque el desarrollo se hubiera realizado dentro del proyecto y dispusiéramos del código de la aplicación, pudiéndola modificar por nuestros propios medios, tampoco es una opción recomendable.

El estar atado a tecnologías propietarias supone un lastre en desarrollos de este tipo. En primer lugar está el tema de las licencias, que por un lado añaden un coste económico no siempre asumible, y por otro pueden -y suelen- imponer una serie de condiciones y restricciones que no siempre serán compatibles con el desarrollo del proyecto.

Resulta particularmente interesante en este caso la confrontación de la tecnología utilizada para el Buscador Inteligente actual, realizado en Flash de Adobe, con la tecnología abierta propuesta: Processing.

Flash es una tecnología propietaria de Adobe. Esto supone, en primer lugar, el tener que descargar, instalar y aceptar las condiciones de la licencia, el *pluging* necesario para poder visualizar en nuestro navegador el buscador. Además para el desarrollo de aplicaciones en Flash es necesario el programa del mismo nombre distribuido por Adobe y que tiene un coste mínimo por licencia de 223€. Incluso si nos pudiésemos permitir el desembolso de dicha cantidad, debemos recordar que este precio incluiría tan solo una licencia, y que se trata de un proyecto colaborativo, por lo que probablemente muchas de las entidades que participan en el proyecto no podrían afrontar el gasto.

¿Qué hacer en ese caso? ¿Centralizar todo el trabajo de desarrollo en uno o dos de los centros? ¿Que una sola persona se ocupe del buscador? Como hemos visto eso sería ir contra las necesidades naturales de un proyecto de estas características.

Por otro lado el confiar en tecnologías propietarias nos deja a merced de lo que el desarrollador -o sus competidores- pretendan hacer con ellas. *Flash* es un caso paradigmático de esto último. Aunque sigue siendo una de las tecnologías más usadas en Internet, su uso ha caído notablemente, y se prevé que desaparezca en los próximos años en favor de tecnologías y estándares como HTML5, Java, JavaScript... De hecho el sistema operativo iOS, presente en los teléfonos y *Tablets* de Apple no soporta de forma nativa Flash.

Frente a esto, se propone como alternativa el uso de Processing, un lenguaje abierto basado en Java.

Su IDE puede ser descargado de la página web de los desarrolladores sin coste alguno y puede ser instalado en tantos ordenadores como se desee.

Es un lenguaje creado específicamente para artistas, diseñadores y arquitectos, por lo resulta perfecto para ser aprendido por personas que no están acostumbradas a la programación. Sin embargo, al estar basado en Java, también puede ser utilizado por programadores con experiencia, lo cual resulta especialmente útil a la hora de revisar y depurar código, de implementar características complejas, o en la formación de equipos mixtos diseñador-programador.

La cantidad de información y recursos existentes tanto para Processing como para Java es impresionante, y muchos de ellos se encuentran a su vez disponibles de forma libre. Estos recursos van desde tutoriales, cursos, ejemplos, foros, etc., hasta código disponible para ser reutilizado, de forma que aprovechemos el trabajo de otros para poder así liberar nuestros recursos.

Finalmente decir que la tecnología que está por detrás es la misma hacia la que se dirige Internet en un futuro inmediato, siendo compatible e integrándose con HTML5.

3.4 Implementación

Nota sobre la implementación:

Tal y como se ha comentado, la implementación se ha realizado utilizando el lenguaje de programación [Processing](#).

Por razones de seguridad y eficiencia en la programación las consultas del buscador se realizan contra el servidor de Organi.Edunet lo que hace que la respuesta sea más lenta. En una hipotética implantación real las consultas se realizarían contra la base de datos, lo que mejoraría notablemente el rendimiento. Este aspecto también ha condicionado la implementación de algunas características, como el uso de recursos gráficos para mostrar la cantidad de recursos de cada término de búsqueda, ya que aunque la parte gráfica y estética no era el objetivo de la implementación, resultaba sencillo añadir algún recurso de este tipo, si bien hubiese demorado demasiado la ejecución del buscador por el número de consultas necesarias al servidor.

La implementación realizada del servidor puede encontrarse en:

<http://finife.es/PFM/>

La versión online tiene una pequeña modificación sobre la original, ya que las consultas se realizan a través de un pequeño programa php para salvar la restricción de seguridad de Java que no permite pedir recursos a un servidor externo al que está alojado. Al igual que anteriormente esto no resultaría un problema en el caso de una implantación real.

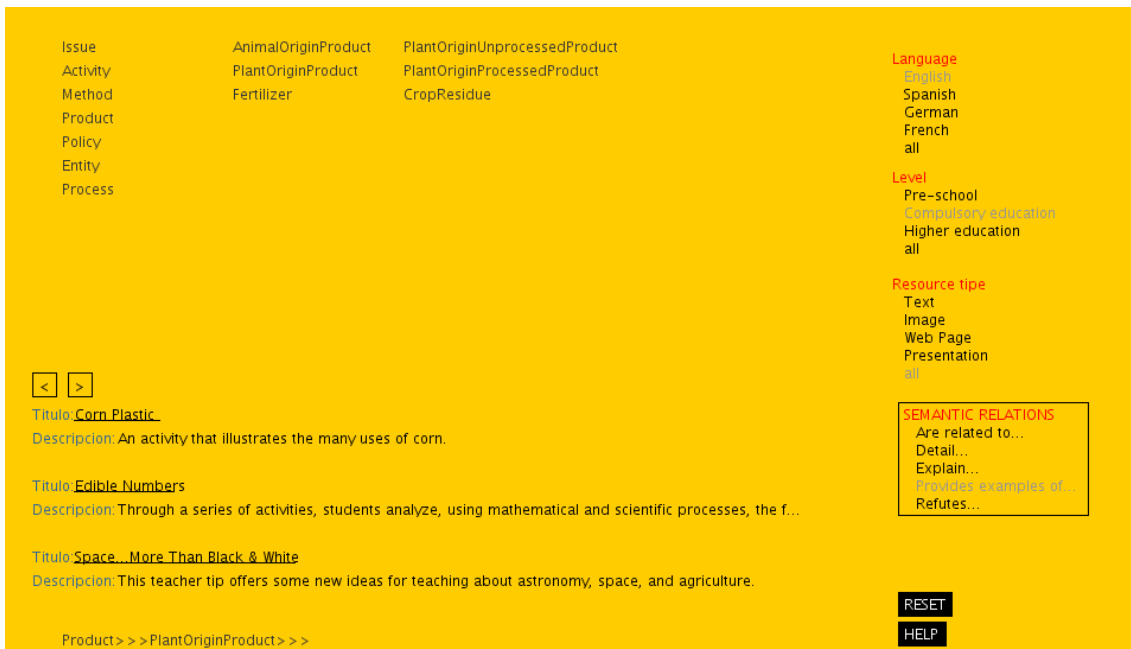
El código fuente puede consultarse en el Anexo A que acompaña al trabajo, o bien descargarse desde [Sourceforge](#).

Además también pueden bajarse ejecutables del buscador para ser corridos directamente en [Linux](#), [Mac](#), o [Windows](#).

Para demostrar lo anteriormente expuesto, se ha realizado una implementación de gran parte de las recomendaciones obtenidas mediante los Test de Usabilidad utilizando el lenguaje de programación Processing, del que ya se habló en el Estado del Arte.

Debido a que un rediseño estético completo del Buscador Inteligente excede con mucho el alcance del presente artículo, se ha decidido dejar esta parte de lado, no entrando en la parte más visual, animaciones, métodos de representación, etc. de los que se presentó una aproximación en el apartado 3.2 *Mockups*, dejando abierta la posibilidad para su desarrollo en futuros trabajos.

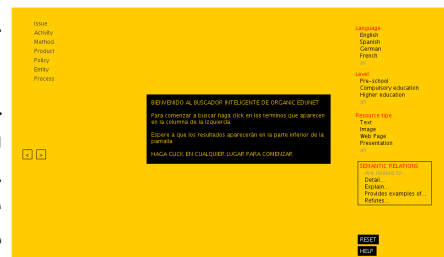
En su lugar se ha pretendido demostrar la conveniencia del uso de sistemas y lenguajes abiertos, tal y como se justificó en el punto 3.3, mostrando cómo de forma rápida y con pocas líneas de código se pueden implementar las recomendaciones que se vayan encontrando en cada uno de los test realizados en las distintas iteraciones que tienen lugar a lo largo de la vida de una aplicación, sin que la falta de acceso al código o el uso de tecnologías propietarias suponga un obstáculo para ello.



Como se puede ver en la imagen, se ha optado por un *interface* sencillo y limpio, de forma que el esfuerzo de desarrollo se centra en este caso en la implementación de los requerimientos obtenidos de los participantes en el Test de Usabilidad.

La organización de los elementos es similar al diseño realizado para el Mockup 1.0.

La primera vez que se accede al buscador, el usuario es recibido con una pantalla de bienvenida en la que se dan unas pequeñas instrucciones al usuario de cómo comenzar la búsqueda, algo que se vio que era necesario durante la realización de los test.



En la siguiente imagen puede verse como se han organizado los elementos en la *interface*, y cómo se han introducido características que facilitan la localización y filtrado de los recursos obtenidos.



Tal y como puede observarse, ahora los términos de la búsqueda están visibles en todo momento. Además podemos saber en qué lugar nos encontramos gracias a los *breadcrumbs* que se encuentran en la parte inferior de la página.

Una vez obtenidos los resultados podemos navegar por ellos mediante los botones situados sobre ellos sin salir del *interface*, evitando ir a una nueva página como ocurría en el caso de Organic.Edunet. Además los títulos nos llevarán directamente al recurso al hacer *click* sobre ellos.

Las opciones de filtrado de resultados se encuentra ahora totalmente a la vista, y su uso resulta natural por parte de los usuarios pudiendo seleccionar el idioma, nivel y tipo de recurso con un solo *click*, recargandose los resultados según nuestras preferencias.

Igualmente sucede con las posibilidades semánticas del buscador, que una vez que se hacen accesibles de una forma más natural, se hace mucho más fácil su uso y menor el número de explicaciones necesarias para que los usuarios sepan utilizarlas y puedan sacar el rendimiento que ofrecen.

También se ha colocado el 'Botón de Reset' de forma que no induzca a error, y se ha añadido un 'Botón de Ayuda' que nos lleva a la página de Organic.Edunet donde se muestra con videos como realizar las búsquedas en los distintos buscadores, pero que también podría llevarnos a una página de ayuda específica del buscador.

Las ventajas de haber realizado la programación del buscador en un entorno abierto frente a uno propietario son tanto inmediatas como a medio plazo.

Como puede verse en el Anexo A, o descargando el código, se han necesitado pocas líneas de código para desarrollar el buscador. Además durante el proceso de desarrollo fue enorme la cantidad de información encontrada *online*, la altísima actividad existente en foros y comunidades de Processing así como la cantidad de recursos de Processing o Java que podían ser estudiados o directamente reutilizados, adaptándolos a nuestras necesidades.

Además, al tratarse de un lenguaje basado en Java, la integración con navegadores y servidores no ofrece ningún tipo de complicación.

A medio plazo, una vez que ya tenemos la estructura del buscador, podrían volver a repetirse los Test de Usabilidad para ir mejorándolo y puliéndolo. Igualmente, una vez completada esta fase, se podrían empezar a añadir las características de diseño, gráficas, de animación, hasta llegar a propuestas como las mostradas en los *mockups*.

Dado que es un lenguaje orientado a diseñadores y artistas, abordar esta etapa utilizando Processing simplificaría el proceso, pudiendo obtener grandes resultados y facilitando el trabajo interdisciplinar. Si además se combina también en esta etapa con nuevos Test de Usabilidad, se podría conseguir un buscador de gran atractivo visual y una usabilidad contrastada, conteniendo los recursos empleados para su desarrollo.

Por último, comentar un aspecto de la programación en Processing que es paradigmático del Software Libre. Dado que se trata de un lenguaje que posee una licencia libre, permite que sea adaptado a las necesidades de quienes lo utilizan. Así ha surgido el proyecto [Processing.js](#), que permite convertir el código de Processing de forma que pueda ejecutarse en cualquier navegador actual (compatible con HTML5) sin necesidad de descargar ni ejecutar ningún tipo de Plug-in, lo que muestra el compromiso de los proyectos desarrollados bajo la filosofía del Software Libre con los estándares y la facilidad para el usuario, frente a las estrategias seguidas muchas veces por los desarrollos cerrados.

4 Conclusiones

En la primera parte del artículo se ha presentado el Estado del Arte de las diferentes áreas que intervienen en el diseño de una aplicación como la analizada en este trabajo.

A través de esta visión del Estado del Arte se puede constatar como se trata de desarrollos multidisciplinares, donde intervienen conocimiento y habilidades de áreas tan diferentes como el diseño, la Inteligencia Artificial y la programación.

Igualmente nos da una visión de cómo muchas de estas áreas se encuentran en una fase relativamente temprana de desarrollo, con lo que ello supone de continuas revisiones y cambios que permiten evolucionar a los proyectos.

En una segunda parte se ha realizado un estudio de usabilidad sobre un proyecto existente, el Buscador Inteligente de Organic.Edunet. El objetivo de dicho estudio es doble. Por un lado poner en valor la utilidad de las herramientas que nos proporciona la Usabilidad para la mejora de los proyectos, y por otro lado identificar los puntos que necesitan ser mejorados o añadidos dentro del desarrollo de una aplicación y se han podido presentar propuestas que incorporen las recomendaciones obtenidas.

Con toda esta información se ha podido valorar las ventajas, tanto filosóficas como técnicas, del modelo de Software Libre frente a modelos propietarios.

Finalmente, a modo de demostración, se han implementado gran parte de las recomendaciones obtenidas mediante los Test de Usabilidad en un prototipo cuyo objetivo era mostrar cómo mediante lenguajes abiertos y el uso de estándares se pueden realizar desarrollos e implementar mejoras con pocas líneas de código y lenguajes asequibles para usuarios no acostumbrados a la programación.

Parece claro que en el desarrollo de este tipo de aplicaciones, donde las mejoras y cambios son constantes, donde intervienen grupos con distintos objetivos y recursos, y donde las personas dedicadas al desarrollo del mismo provienen de áreas de conocimiento diversas, el uso de un modelo basado en el uso de Software Libre resulta mucho más eficiente que las opciones privativas utilizadas hasta la fecha.

Para futuros trabajos queda abierta la posibilidad de un rediseño completo del Buscador Inteligente, volviendo a programarlo en Processing, y aprovechando el cambio de tecnología para implementar el total de las recomendaciones halladas en los Test de Usabilidad y especialmente para replantear la estética del buscador, explorando las posibilidades visuales a la hora de presentar información.

5 Referencias

1. Luis Sánchez Fernández, Norberto Fernández García. *La Web Semántica: fundamentos y breve "estado del arte"*. Novática, nº178, pags. 06-11. Noviembre-Diciembre 2005.
2. R. Studer, V.R. Benjamins, D.Fensel. "Knowledge Engineering: Principles and Methods". *IEEE Transactions on Data and Knowledge Engineering* 25(1-2):161-197, 1998.
3. Ben, F. (2007). *Visualizing Data*. O' Reilly, USA.
4. Terzidis, K. (2009). *Algorithms for Visual Design Using the Processing Language*. Wiley Publishing, Indianapolis (USA)
5. Quesenbery, Whitney (2001). *What does usability mean: looking beyond "Ease of use"*. [en línea]. <http://www.wqusability.com/articles/more-than-ease-of-use.html> [Consulta: 6 de octubre 2011]
6. Nielsen, Jacob (2000). *Why you only need to test with 5 users*. [en línea]. <http://www.useit.com/alertbox/20000319.html> [Consulta: 29 de septiembre 2011]
7. Barnum, Carol M.(2011) *Usability Testing Essentials: ready, set... test!*. Morgan Kaufman, Burlington (USA)
8. Paulheim, H. and Probst (2010). *Ontology-Enhanced User Interfaces: A Survey*. *Journal on Semantic Web and Information Systems (IJSWIS)*
9. Tufte, E. (2001). *The Visual Display of Quantitative Information, 2nd edition*. Graphics Press, Cheshire (Connecticut)
10. Perens, Bruce (2005). *The emerging economic paradigm of open source*. [en línea]. <http://perens.com/works/articles/Economic.html> [Consulta: 3 de noviembre de 2011]