

**COMPONENTE PARA LA VISUALIZACIÓN DE RESULTADOS DE BÚSQUEDA  
MULTIDOMINIO UTILIZANDO LÍNEAS DE TIEMPO EN JAVASCRIPT Y HTML5  
PARA EL PROYECTO SECO**



**LUZ MERLEN BOHORQUEZ BARRERA  
ADRIÁN GUILLERMO BERNAL SANDOVAL**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER - UIS  
FACULTAD DE INGENIERIAS FISICOMECAÑICAS  
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA  
BUCARAMANGA  
2011**

**COMPONENTE PARA LA VISUALIZACIÓN DE RESULTADOS DE BÚSQUEDA  
MULTIDOMINIO UTILIZANDO LÍNEAS DE TIEMPO EN JAVASCRIPT Y HTML5  
PARA EL PROYECTO SECO**



**AUTORES**

**LUZ MERLEN BOHORQUEZ BARRERA  
ADRIÁN GUILLERMO BERNAL SANDOVAL**

**TRABAJO DE GRADO PARA OPTAR AL TÍTULO DE:  
INGENIERO(A) DE SISTEMAS**

**DIRECTOR**

**FERNANDO ROJAS MORALES**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER - UIS  
FACULTAD DE INGENIERIAS FISICOMECAÑICAS  
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA  
BUCARAMANGA**

**2011**

## **AGRADECIMIENTOS**

Damos gracias a Dios por darnos la vida y la oportunidad de rodearnos de seres maravillosos que nos sirvieron de ejemplo y guía para querer cumplir nuestros sueños.

A nuestros padres quienes con su esfuerzo, cariño, confianza y apoyo nos dieron lo mejor sus vidas.

A nuestros hermanos por ser ese aliento en tantos tropiezos y momentos difíciles, gracias por sus palabras, sobre todo por el ánimo de cada día, por los besos y los abrazos en esos momentos de lágrimas.

A nuestro profesor Fernando Rojas que a nuestro parecer es un gran docente y un maravilloso ser humano, para él, queremos desearle miles y miles de bendiciones.

Al profesor Piero Fraternali quien nos dio la oportunidad de trabajar con ellos en el proyecto SeCo y a Alessandro Bozzon quien nos brindó apoyo con sus consejos e indicaciones en el desarrollo del proyecto.

## TABLA DE CONTENIDO

INTRODUCCIÓN.....	21
Generalidades.....	22
1.1. OBJETIVO GENERAL.....	22
1.1.1. Objetivos específicos.....	22
1.2. JUSTIFICACIÓN.....	23
1.3. DESCRIPCIÓN DEL PROBLEMA.....	24
1.4. DESCRIPCIÓN DE LA SOLUCIÓN.....	24
Marco teórico.....	26
2.1. CONTEXTO DEL PROYECTO.....	26
2.1.1. Search Computing (SeCo) Project.....	26
2.1.2. Liquid Query.....	28
2.2. HUMAN-COMPUTER INTERACTION (HCI).....	29
2.3. VISUALIZACIÓN.....	30
2.4. LÍNEAS DE TIEMPO.....	32
2.5. TECNOLOGÍAS.....	32
2.5.1. jQuery.....	32
2.5.2. AJAX (Asynchronous JavaScript And XML).....	34
2.5.3. Hojas de estilo en cascada (CSS).....	34
2.5.4. HTML5.....	35
2.5.5. JavaScript.....	35
2.6. MEDOLOGÍA.....	36
2.7. DSBC.....	37
2.7.1 La interfaz del componente.....	41
2.7.2 Frameworks y modelos de componentes.....	41
Requerimientos.....	43
3.1. REQUERIMIENTOS.....	43
3.1.1. Requerimientos funcionales.....	43

3.1.2. Requerimientos no funcionales.....	44
3.2. INTERACCIÓN CON EL USUARIO.....	45
3.3. DIAGRAMAS UML.....	46
3.3.1 Caso de Uso.....	46
3.3.2 Diagramas de clases .....	49
3.3.3 Diagrama de Secuencia .....	50
Diseño.....	51
4.1. PRINCIPIOS DE DISEÑO.....	51
4.2. DISEÑO DE LA INTERFAZ DE USUARIO.....	53
4.3. HERRAMIENTAS DE APLICACIÓN.....	55
4.3.1. Paper prototyping .....	55
4.3.2. StoryBoard.....	55
4.3.3. Mockups .....	60
4.3.3.1 Mockup basado en diagramas de <i>Venn</i> .....	61
4.3.3.2. Descripción del mockup inicial del visualizador <i>Timeline</i> .....	65
4.3.3.3 Mockup última versión del componente de visualización .....	68
4.4. SIMULACIÓN CON DATOS.....	71
4.5. PRÁCTICA CON LIBRERÍAS JAVASCRIPT.....	72
Y CANVAS.....	72
Implementación.....	76
5.1. FUNCIONAMIENTO DEL COMPONENTE.....	76
5.2. FRAMEWORK Y LIBRERIAS UTILIZADAS.....	77
5.3. DISEÑO Y FUNCIONALIDAD.....	78
Evaluación .....	80
6.1. EVALUACIÓN HEURÍSTICA.....	81
6.2. RESULTADO DE LA EVALUACIÓN.....	85
7. CONCLUSIONES.....	88
BIBLIOGRAFÍAS.....	89

## LISTA DE TABLAS

	Pág.
Tabla. 1. Documentación caso de uso del componente.....	46
Tabla. 2. Principios de diseño.....	52
Tabla. 3. Criterios de .usabilidad.....	54
Tabla. 4. StoryBoard del componente de visualización.....	56
Tabla. 5. Descripción diseño de componente en diagramas de ven.....	61
Tabla. 6. Descripción de la primera versión del componente.....	66
Tabla. 7. Descripción componente final.....	69
Tabla. 8. Descripción de las prácticas realizadas con el componente.....	73

## LISTA DE FIGURAS

	Pág.
Figura 1. Vista general de Liquid Query y ciclo de vida del resultado.....	29
Figura 2. Diseño de una interfaz de usuario y esquemas de visualización dentro de la arquitectura de SeCo.....	31
Figura 3. Fases de trabajo.....	37
Figura 4. Caso de Uso Principal. Describe la interacción del componente de visualización con SeCo, para visualizar la consulta multidominio de un usuario SeCo.....	46
Figura 5. Diagrama de clases: para describir la estructura del componente, las clases con sus atributos para el análisis del componente.....	49
Figura 6. Diagrama de Secuencia: muestra la interacción entre el usuario y el componente de visualización.....	50
Figura 7. El panel de consulta SeCo.....	56
Figura 8. Después de ordenados los datos se pueden visualizar en un componente vista atómica.....	57
Figura 9. El usuario da clic en la pestaña Líneas de Tiempo.....	57
Figura 10. Área de visualización de las combinaciones.....	57

Figura 11. El área de visualización se muestra las mejores combinaciones.....	58
Figura 12. Al pasar el mouse por un evento se resalta la fecha del evento o combinación.....	58
Figura 13. Al dar click se despliega una ventana con la información de la combinación.....	59
Figura 14. Al dar click en uno de los días se despliega una ventana con las combinaciones en las horas del día.....	59
Figura 15. Combinaciones para que el usuario escoja.....	59
Figura 16. Pestañas donde el usuario escoge el método de visualización en la que quiere ver sus datos.....	60
Figura 17. Imagen inicial hecha a lápiz y en papel, de un visualizador de esferas por horas.....	61
Figura 18. Vista panorámica del segundo diseño de componente de visualización.....	61
Figura 19. Imagen que muestra las combinaciones.....	63
Figura 20. Vista detallada de la acción sobre cada conjunto de combinaciones.....	64
Figura 21. Vista de tabla después de realizarle click en una de las combinaciones.....	64
Figura 22. Imágenes de las herramientas que ofrece el componente de visualización.....	65
Figura 23. Primer diseño en papel y lápiz del componente de visualización en barras.....	65

Figura 24. Visualizador de barras mockup realizado en Visio.....	66
Figura 25. Escala de tiempo por horas. Vista zoom de un intervalo de un día.....	67
Figura 26. Las combinaciones en los días.....	67
Figura 27. Rollover con la información de la combinación.....	68
Figura 28. Herramientas del visualizador.....	68
Figura 29. Componente de visualización Time Line.....	69
Figura 30. Rollover con la fecha del evento de la combinación.....	69
Figura 31. Se despliega una ventana con la información de la combinación.....	70
Figura 32. Herramienta para dar zoom y pasar fechas.....	70
Figura 33. Ventana con la información de la combinación, se cierra en la X.....	70
Figura 34. Herramienta para escoger otras combinaciones.....	71
Figura 35. Primera interacción con Canvas, días de la semana con imágenes que representaran las posibles combinaciones.....	72
Figura 36. Insertando imágenes en el Canvas, describe como aparecen las imágenes prediseñadas en el Canvas.....	73
Figura 37. Distribución del espacio del componente de visualización.....	73

Figura 38. Utilizando jQueryUI.....	74
Figura 39. El componente de visualización en horas de cierto día.....	74
Figura 40. Diseño final del componente de visualización utilizando la librería timeglider.....	74
Figura 41. El visualizador con sus combinaciones.....	75
Figura 42. Combinaciones en la línea de tiempo con escala por horas.....	79
Figura 43. Combinaciones (eventos) en la línea de tiempo.....	80

## G L O S A R I O

**SeCo:** Abreviatura de Search Computing project.

**JavaScript:** Es un lenguaje interpretado orientado a las páginas web, con una sintaxis semejante a la del lenguaje Java.

**HTML5:** Es la quinta revisión de HTML (Hyper Text Markup Language).

**Diagramas UML:** Es un conjunto de herramientas, que permite modelar (analizar y diseñar) sistemas orientados a objetos.

**Mockups:** Herramienta utilizada por los diseñadores y son los primeros prototipos realizados en el diseño del componente.

**Storyboard:** Herramienta útil para la elaboración de guiones, es un guión que narra la historia del componente de visualización.

**CSS(Cascade Style Sheet):** Hojas de estilo en cascada, describe la apariencia de la página.

**Meta-search engines:** Es una herramienta de búsqueda que envía peticiones de usuario a otros motores de búsqueda o bases de datos y agrega los resultados en una sola lista o los muestra en función de su origen.

**Kosmix:** Era una compañía privada americana in Mountain view, California, es un buscador semántico cuyo objetivo es proporcionar información sobre un tema.

**Google:** Es un motor de búsqueda de contenido en la internet.

**Yahoo:** Es un servicio global de internet para consumidores y negocios.

**Day-Life:** Fuente de noticias independientes en la red.

**YouTube:** Es un sitio electrónico en el cual los usuarios pueden subir y compartir vídeos.

**Twitter:** Es una red social basada en el *microblogging*.

**Liquid Query:** Es una herramienta flexible para la búsqueda de información, basada en la exploración progresiva del espacio de búsqueda.

**Back-end:** Parte que procesa la entrada desde el front-end.

**Front-end:** Es la parte del software que interactúa con el o los usuarios.

**HCI (Human–Computer Interaction):** Disciplina que estudia el intercambio de información entre las personas y los computadores.

**Feedback:** Realimentación, desde el punto de vista social y psicológico, es el proceso de compartir observaciones, preocupaciones y sugerencias. Con la intención de recabar información, a nivel individual o colectivo.

**UCD (Diseño Centrado en el Usuario):** Es una filosofía de diseño que tiene por objetivo la creación de productos que resuelvan las necesidades concretas del usuario final.

**jQuery:** Es una biblioteca o framework de Javascript, que permite interactuar con los documentos HTML, manipular el árbol de DOM, entre otras.

**jQueryUI:** Es una biblioteca de componentes para el framework jQuery que le añaden un conjunto de *plug-ins*, *widgets* y efectos visuales para la creación de aplicaciones web.

**AJAX (Asynchronous JavaScript And XML):** Es una técnica de desarrollo web para crear aplicaciones enriquecidas de internet (*Rich Internet Applications*).

**TimeLine:** Las líneas de tiempo permiten crear gráficas de líneas de tiempos a partir de un lenguaje de marcado especial.

**Drag&drog:** Característica de un elemento para ser arrastrado y soltado en la interfaz gráfica de un usuario.

## RESUMEN

### TÍTULO:

COMPONENTE PARA LA VISUALIZACIÓN DE RESULTADOS DE BÚSQUEDA MULTIDOMINIO UTILIZANDO LÍNEAS DE TIEMPO EN JAVASCRIPT Y HTML5 PARA EL PROYECTO SECO<sup>1</sup>.

### AUTORES<sup>2</sup>:

LUZ MERLEN BOHORQUEZ BARRERA  
ADRIÁN GUILLERMO BERNAL SANDOVAL

### PALABRAS CLAVE:

Visualización de la información, SeCo (Search Computing Project), Storyboard, Paper Prototyping, Liquid Query, Búsquedas multidominio, líneas de tiempo.

Mostramos la visualización de datos a través del enfoque de visualización de línea de tiempo en un componente software donde se demuestra la presentación y el control de datos permitiendo amplificar la cognición como elemento central de la disciplina de la visualización de la información. La visualización consiste en la transformación de los datos por medio de los procesos mentales hasta llegar a ser un elemento que permita tomar decisiones y convertirse en conocimiento. Este trabajo de grado se enfoca en la visualización de resultados de búsqueda multidominio centrada en datos orientados en el tiempo, es decir, con atributos que son conectados con el tiempo. Se muestra el desarrollo desde la fase de análisis y requerimientos, se enfoca ampliamente en el diseño y se da relevancia a la creación de mockups y storyboards utilizando la técnica paper prototyping para observar la interacción y obtener realimentación. El proceso de desarrollo se basa en el diseño centrado en el usuario. El resultado final es un componente de visualización de resultados de consultas multidominio propuesto por Search Computing (SeCo). SeCo es un proyecto fundado por el Consejo Europeo de Investigación (ERC) y pretende construir una plataforma multidominio para realizar consultas basadas en el paradigma Liquid Query.

---

<sup>1</sup> Trabajo de grado en la modalidad de investigación

<sup>2</sup> Facultad de Ingenierías Fisicomecánicas. Escuela de Ingeniería de Sistemas e Informática.

Director: Fernando Rojas Morales

## ABSTRACT

### TITLE:

COMPONENT FOR THE VISUALIZATION OF MULTIDOMAIN SEARCH RESULTS USING TIMELINES IN JAVASCRIPT AND HTML5<sup>3</sup>.

### AUTHORS<sup>4</sup>:

LUZ MERLEN BOHORQUEZ BARRERA  
ADRIÁN GUILLERMO BERNAL SANDOVAL

### KEYWORDS:

Information Visualization, SeCo (Search Computing Project), Storyboard, Paper Prototyping, Liquid Query, Multi-domain search results, timeline.

We show the data visualization through timeline approach visualization in a software component where it is demonstrated the presentation and data control, allowing the cognition amplification as a central element of the discipline of the visualization of the information. The visualization consists on data transformation by means of mental processes until it becomes an element that allows to make decisions and become knowledge.

This graduation work is focused on visualization of multi-domain search results based on time oriented data, i.e. with attributes that are connected with time. It shows the development from the analysis and requirements, it widely focuses on the design, and relevance is given to the creation of mockups and storyboards using the paper prototyping technique to observe the interaction and get feedback. The development process is based on the user centered design. The final result is a component for the visualization of multi-domain search results proposed by Search Computing (SeCo).

SeCo is a project founded by the European Research Council (ERC) and it aims to build a platform for multidomain queries based on the Liquid Query paradigm.

---

<sup>3</sup> Graduate work in the mode of research

<sup>4</sup> Faculty of physico-mechanical engineerings. School of systems and informatics engineering.

Director: Fernando Rojas

## PALABRAS CLAVE

Information Visualization (visualización de la información), SeCo (Search Computing Project), Storyboard, *paper prototyping*, *Liquid Query*, Búsquedas multidominio, ordenamiento de datos, visualización, líneas de tiempo (*timelines*), HTML5, JavaScript, Ajax, jQuery.

## INTRODUCCIÓN

El fundamento teórico y experimental de este trabajo de grado es el proyecto SeCo, dirigido por investigadores y académicos del Politécnico de Milán. SeCo es fundado por el Consejo Europeo de Investigación.

SeCo pretende llegar a realizar búsquedas complejas sobre diversas fuentes de datos. Construir respuestas a dichas búsquedas y presentar como resultado la mejor combinación de datos multidominio también incluye el problema de la visualización.

Se aborda el problema de la visualización mediante técnicas y métodos para la representación gráfica de la información, particularmente, utilizando líneas de tiempo para construir un componente de visualización de resultados de búsquedas multidominio con una interfaz gráfica flexible como aplicación *front-end*.

Para el desarrollo del componente, se utilizará tecnologías Web tales como *HTML5*, *AJAX* y *Javascript* utilizando el *framework jQuery*. El componente será implementado e incorporado en la arquitectura de SeCo.

## Capítulo 1. Generalidades

### 1.1. OBJETIVO GENERAL

Desarrollar un visualizador para resultados de búsqueda multidominio utilizando líneas de tiempo, que será implementado e incorporado en la arquitectura de SeCo, utilizando tecnologías Web como JavaScript y HTML5.

#### 1.1.1. Objetivos específicos

1. Analizar los requerimientos y construir modelos apropiados utilizando diagramas *UML* y prototipos mediante *Mockups* y *storyboards*.
2. Diseñar el Componente de Visualización a partir de una experiencia de usuario flexible y agradable, utilizando como base la metodología de *Diseño Centrado en el Usuario*.
3. Utilizar tecnologías de desarrollo web tales como *Javascript*, *CSS* y *HTML5*, orientadas a la creación de *Aplicaciones Enriquecidas de Internet (RIA)*, para el desarrollo de un Componente de Visualización para SeCo.
4. Realizar pruebas con usuarios para evaluar la usabilidad del componente y corregir errores de diseño e implementación.
5. Realizar un registro del proceso seguido en el desarrollo del Componente de Visualización.

## 1.2. JUSTIFICACIÓN

El tema elegido para este trabajo de grado obedece a una necesidad de mostrar datos temporales en un visualizador agradable, usable, flexible y perceptible para expresar consultas multidominio de una manera fácil, presentando resultados efectivos y permitiendo interactividad con el usuario. Este trabajo se enfoca en diseñar un componente de visualización para un usuario final para el proyecto SeCo con particular atención en aspectos de la visualización de la información para datos multidominio. Se tiene en cuenta el creciente interés de las personas en realizar búsquedas en la web, y para ello se requiere interfaces que sigan un proceso de diseño centrado en el usuario.

El componente de visualización se adaptará al motor de búsqueda del proyecto SeCo, desarrollado en el Politécnico de Milán (Italia), llegando a pertenecer a la arquitectura SeCo, en la parte *front-end* o del lado del cliente. Por tanto, el componente es importante para el proyecto SeCo, pues es el que va a estar en continua interacción con el usuario.

Para desarrollar el componente se ha requerido un estudio previo del proyecto SeCo, y entender su propósito y así adoptar las tecnologías que ellos proponen para el desarrollo del proyecto, las cuales están a la vanguardia de nuevas propuestas de investigación.

El uso de nuevas tecnologías teniendo poca experiencia en ellas es un reto, porque puede llegar a ser un problema en un proyecto de software, por el poco conocimiento que se tiene de éstas y por el tiempo que hay que dedicar para aprenderlas; hay que resaltar que utilizar nuevas tecnologías es innovador, y lo innovador requiere de entusiasmo y dedicación.

### **1.3. DESCRIPCIÓN DEL PROBLEMA**

Si pensamos en la cantidad de datos que hay a nuestro alrededor, observamos que son tantos que no los percibimos bien, por esto, la información que puede brindar es confusa y difícil de entender para cualquier persona. Desde el inicio de la historia, el hombre ha recibido la información a través de la percepción visual, ya que es una de las principales capacidades sensoriales del hombre y de los animales. Y es este sentido el que ha llegado a convertirse, tal vez, en la forma más común de interacción del ser humano con el ordenador.

La tecnología ha cambiado mucho, desde la prehistoria nos hemos comunicado con jeroglíficos e imágenes pintadas en las piedras y hasta comunicarnos con nubes de humo o llamas de un extremo a otro para prevenir ataques o posibles invasiones del enemigo, utilizando atracción visual, pero a medida que la tecnología ha evolucionado se ha mantenido fiel al principio de aportar una forma clara, creativa, cómoda y realista de visualización de la información.

El problema general de la visualización de la información consiste en escoger entre la mejor asignación entre el conjunto de resultado de datos y el espacio de visualización. Particularmente, este trabajo aborda la necesidad de representar y mostrar objetos multidominio combinados según ciertos criterios de tal forma que representen información utilizando datos orientados en el tiempo.

### **1.4. DESCRIPCIÓN DE LA SOLUCIÓN**

Hay muchas formas de visualizar la información, pero en este caso es visualizar datos de resultados multidominio, por ello se quiere diseñar y elaborar un componente de visualización utilizando líneas de tiempo, ya que es una forma de medir un acontecimiento o evento según su duración. Para la mente humana es muy difícil imaginar la temporalidad porque implica un alto grado de abstracción, la forma más sencilla y clara de entender el tiempo es viéndolo, si plasmamos una imagen, se puede adquirir mayor conciencia del transcurso temporal del

acontecimiento o evento, las líneas de tiempo precisamente sirven para entender a través de la visualización el tiempo histórico, Para ello, el componente de visualización quiere mostrar un hecho o proceso temporal en el que tres objetos se encuentren simultáneamente abiertos al público.

## Capítulo 2. Marco teórico

### 2.1. CONTEXTO DEL PROYECTO

#### 2.1.1. Search Computing (SeCo) Project

*Search computing* (SeCo)[1] es un proyecto fundado por el Consejo Europeo de Investigación (ERC), en respuesta a una convocatoria de 2008 de IDEAS *Advance Grants*, un programa dedicado al apoyo de proyectos excelentes de investigación en las fronteras del conocimiento.

El proyecto SeCo busca permitir consultas complejas de búsqueda sobre fuentes de datos, envueltas y registradas como servicios de búsqueda [2]. SeCo tiene por objetivo la construcción de una plataforma para consultas multidominio a través de servicios de búsqueda, mientras aborda varios problemas de investigación, incluyendo la ingeniería y registro de los servicios de búsqueda, métodos de combinación de servicios de búsqueda eficientes y motores de ejecución de consultas flexibles [3].

SeCo es una nueva disciplina cuyo objetivo es responder a complejas consultas multidominio. Estas consultas suelen requerir en el dominio de sus resultados, la combinación de conocimientos extraídos de múltiples recursos *web*, por lo tanto, las técnicas convencionales de rastreo e indexación que se observan en páginas web individuales no son adecuadas para ellas.

*Search Computing* es un nuevo paradigma para la composición de servicios de búsqueda. Mientras el estado del arte evidencia que los sistemas de búsqueda responden a consultas genéricas o de dominio específico, *Search Computing* permite responder a las consultas a través de una constelación de servicios de búsqueda cooperantes seleccionados dinámicamente, que son correlacionados

por medio de operaciones de combinación. La idea es simple, pero penetrante. Un nuevo lenguaje y la descripción de paradigmas son necesarios para expresar consultas y para conectar servicios. Nuevas interfaces de usuario y protocolos ayudan a capturar preferencias de ordenamiento y a permitir su refinamiento.

El procesamiento de consultas en múltiples motores de búsqueda no es nuevo, los *meta-search engines* [4], [6], son capaces de encaminar la misma consulta de múltiples motores de búsqueda y luego presentar resultados globales. Un ejemplo de esto es *Kosmix*, un *meta-search engine* de nueva generación que está conectado a más de mil fuentes mediante el uso de los servicios Web. En *Kosmix*, las fuentes de datos pertinentes para una consulta se determinan haciendo coincidir las palabras claves de los usuarios con un gran concepto privado de taxonomía (de aproximadamente un millón de nodos), después etiqueta manualmente las fuentes de datos con los mismos conceptos de taxonomía. Posteriormente, *Kosmix* encamina las consultas hacia las fuentes de información sin intentar su integración. Los resultados son recogidos de *Google*, *Yahoo*, *Flicker*, *YouTube*, *Twitter* y así sucesivamente, y presentados a los usuarios; las fuentes suelen incluir motores de búsqueda muy populares, pero algunas veces también fuentes de dominio específico tales como las de *Day-Life* o *Slate* (en el dominio de las noticias). Mientras que *Kosmix* tiene la habilidad de mostrar resultados individuales de diferentes fuentes de datos, no integra dominios múltiples y por consiguiente no puede responder a consultas complejas, sin embargo, puede responder a consultas sencillas mediante la recuperación de datos de una variedad de fuentes. A pesar de todo, *Kosmix* demuestra que los servicios *Web* son métodos viables para obtener información de fuentes remotas.

Los motores de búsqueda vertical están enfocados hacia un solo dominio, por ejemplo, hoteles (reservas) o vuelos (*Tuifly*), que son bien entendidos en términos de calidad de datos y criterios de clasificación. Por ejemplo, para hoteles y vuelos la clasificación depende del precio, más otros criterios de dominio específico, tales

como la localización y las estrellas del hotel, o la duración del vuelo y el número de paradas intermedias. Por lo tanto, los sistemas de motores de búsqueda vertical realizan una clasificación de los resultados usando una sola función de puntuación y calculan el mejor resultado global de su dominio. *Kosmix*, en cambio, colecciona resultados de acuerdo con una fuente relevante de datos, sin entremezclar elementos de las diversas fuentes [7].

### **2.1.2. Liquid Query**

Las consultas líquidas son una herramienta flexible para la búsqueda de información, basada en la exploración progresiva del espacio de búsqueda; producen resultados “líquidos” que se adaptan dinámicamente a la forma de la consulta, así como un líquido se adapta a su recipiente. La búsqueda exploratoria se define como la situación en la que el usuario empieza con una necesidad de información no muy bien definida y progresivamente, va descubriendo más, en la información disponible para abordarla, sobre su necesidad, con una mezcla de búsqueda, navegación, análisis y exploración.

*Liquid Query* es un nuevo paradigma que permite a los usuarios formular y obtener respuestas a consultas multidominio a través del enfoque de búsqueda exploratoria de información, basado en las fuentes de información estructurada, expuestas como servicios de *software*. El enfoque de *Liquid Query* se puede apreciar gráficamente en la figura 1 y más detalladamente en [3].

El paradigma de *Liquid Queries* se basa en el conjunto de servicios de *SeCo* y en los conceptos de ejecución de consultas multidominio: un usuario experto [22] selecciona a priori, manualmente, el conjunto de servicios relevantes para las tareas de búsqueda de información y las conexiones necesarias para unirlos, y publica dicha definición en *el back-end* de *SeCo*. La interfaz del lado del cliente de

*Liquid Query* utiliza la definición creada por el experto y genera dinámicamente una interfaz de consulta para el usuario final.

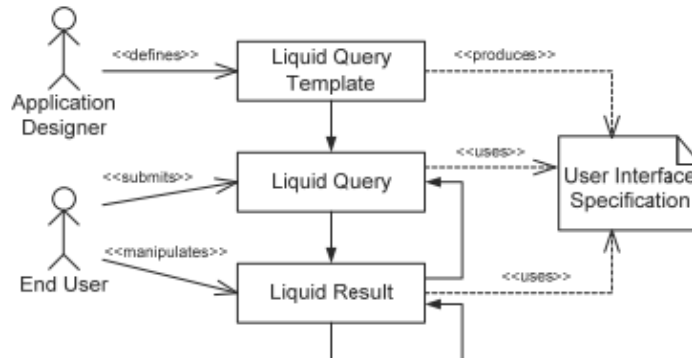


Figura 1. Vista general de *Liquid Query* y ciclo de vida del resultado. Fuente: [3]

## 2.2. HUMAN-COMPUTER INTERACTION (HCI)

Es una disciplina que estudia el diseño, la evaluación y la implementación de sistemas informáticos para uso humano, junto con el estudio de los principales fenómenos que les rodea [26]. El objetivo es que el intercambio persona-ordenador sea más eficiente mediante la minimización de errores, el incremento de la satisfacción y la disminución de la frustración, logrando que sean más productivas las tareas entre personas y computadores. El enfoque científico de la HCI incluye una variedad de herramientas y técnicas que ayudan a desarrollar mejores interfaces de usuario.

HCI tiene una marcada influencia de la psicología. Es esta ciencia la que estudia la memoria, la percepción, la adquisición de habilidades y el aprendizaje, la resolución de problemas, el movimiento, las tareas de juicio, de búsqueda o procesamiento de la información y la comunicación, es decir, procesos cuyo conocimiento se requiere para el adecuado diseño de mecanismos de interacción del usuario.

La interacción se trata de comunicación. Por lo tanto, la interfaz debe estar diseñada pensando en las necesidades del usuario y para esto se han creado modelos que se centran en que haya un *feedback* entre usuarios, diseñadores e ingenieros. Es así como ha surgido la metodología de Diseño Centrado en el Usuario (*UCD*) en la cual se basa este proyecto.

### 2.3. VISUALIZACIÓN

La visualización está muy relacionada con los procesos cognitivos [10]. Existe una estrecha relación entre los procesos internos mentales, la percepción externa (visual) y su manipulación, la cual es resaltada por la evidencia de cómo se logra expandir la inteligencia y amplificar la cognición, mediante el uso de cualquier tipo de invención gráfica [11].

El objetivo principal de crear representaciones visuales para un conjunto de datos es amplificar la cognición mediante el uso de la percepción visual. Utilizar herramientas gráficas permite ver cosas que anteriormente estaban ocultas como estructuras, patrones, tendencias, excepciones, relaciones y datos ocultos por otros [10].

Según [11], la *visualización* es la representación visual interactiva de datos para amplificar la cognición.

Las dos principales características de una herramienta de visualización de información son la *presentación* y *el control*. Es decir, que una herramienta de visualización de la información es contraria a una representación gráfica de datos estáticos, dado que el elemento clave es la interactividad.

Para SeCo, el enfoque de la visualización, amplía las prácticas actuales de varias maneras: los datos a visualizar son combinaciones de objetos n-dimensionales, con criterios asociados de ordenamiento para objetos individuales y para conjuntos de combinaciones; las propiedades de los objetos pueden ser

clasificadas en varios tipos, lo que optimiza el uso de las familias de visualización, que son preferidas según los datos (líneas de tiempo para datos relacionados con el tiempo, mapas para información geo-localizada); las combinaciones pueden presentar cualquier número de propiedades relevantes para ser mostradas, que necesitan ajustarse a un espacio de presentación bidimensional, enfatizando los atributos más importantes y atenuando o escondiendo los de menor importancia. Por lo tanto, el problema de la visualización consiste en decidir la mejor asignación entre el conjunto de resultados de datos y el espacio de visualización.

El componente de visualización estará orientado hacia la visualización de resultados de búsqueda multidominio. Este trabajo de grado se enfoca en el diseño de un artefacto de interfaz de usuario (componente de visualización), aplicación *front-end* para el proyecto SeCo. En la *figura 2* se presenta una gráfica de la arquitectura de SeCo, enfocando la sección de interfaz de usuario.

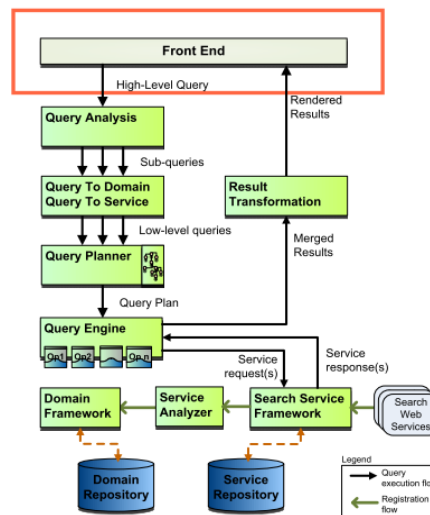


Figura 2. Diseño de una interfaz de usuario y esquemas de visualización dentro de la arquitectura de SeCo. Fuente: [2].

## 2.4. LÍNEAS DE TIEMPO

Las líneas de tiempo son un medio eficaz para relacionar objetos que se ubican en el tiempo [9]. Permiten organizar gráficamente una secuencia de eventos ordenados a lo largo de un período de tiempo.

Como dice [17], la visualización del tiempo no es un tema muy habitual y tampoco existen muchas metáforas visuales asociadas a este tema. Las metáforas visuales que representan el tiempo, proporcionan un marco de referencia para mostrar los eventos ligados en un cierto momento.

Habitualmente, con el tiempo se asocia una metáfora lineal:

*Línea de tiempo:* Es un gráfico unidimensional en que el devenir temporal se representa por una línea, en la cual se resaltan o se señalan los eventos mediante marcadores, y cuya posición sobre la línea de tiempo es proporcional al tiempo transcurrido desde un punto de referencia.

*Banda temporal:* Es un gráfico bidimensional. En una dimensión se representa el paso del tiempo, mientras que en la otra se muestra una magnitud asociada a los sucesos que se representan. Ejemplo de esto es el *diagrama de Gantt*.

*Espiral:* Es una línea (o banda) que se recoge sobre sí misma. Aprovecha mejor el espacio y puede llegar a brindar más información a partir de su origen. Ejemplos en [18], [19].

*Hélice:* Una versión 3-D de la espiral. Un ejemplo en [20].

## 2.5. TECNOLOGÍAS

### 2.5.1. jQuery

*jQuery* es una biblioteca o *framework* de *Javascript*, creada por *Jhon Resig* [31]. Permite simplificar la manera de interactuar con los documentos *HTML*, manipular

el árbol *DOM* (*Document Object Model*), manejar eventos, desarrollar animaciones y agregar interacción apoyada en la técnica *Ajax* de desarrollo web.

Las aplicaciones en internet son cada vez más complejas, entre otras cosas, porque incorporan animaciones, funcionalidades y efectos visuales, los cuales resultan muy complejos si se desarrollan desde cero, es por esta razón que el uso de librerías o *frameworks* facilita la implementación y reduce las líneas de código, simplificando con el código que ya posee para realizar operaciones comunes. Otras de las ventajas de *jQuery* es que integra funcionalidades para trabajar con *AJAX* y las aplicaciones son compatibles con los principales navegadores.

*jQuery* es software libre y de código abierto, posee Licencia *MIT* y Licencia Pública General de GNU, permitiendo su uso en proyectos libres y privativos [25]. Para descargar y ver su documentación en [24].

Se usa este framework porque fue sugerida por los investigadores de SeCo, en especial Alessandro Bozzon is a post-doc at DEI. Sus intereses de investigación actuales en los paradigmas de interacción con el usuario, visualización de datos y técnicas de resultado la diversificación en múltiples dominios de búsqueda en las colecciones de datos heterogéneos, además experto en *jQuery*, sugirió la siguiente librería:

- *jQueryUI*[24]: es una librería de componentes para el framework *jQuery* que provee un conjunto de *plugins*, efectos visuales y *widgets*. *jQuery* proporciona abstracciones de bajo nivel de interacción y animación, efectos avanzados y de alto nivel, *widgets* temable, se encuentra en la parte superior de la biblioteca *jQuery JavaScript*, se utiliza para la construir aplicaciones web altamente interactivas.

1. Componentes de interacción: *drag & drop, sorting, resizing*.

2. Efectos: *show, hide, toggle, color animation, class manipulation*.
3. Widgets: *accordion, date picker, dialog, slider y tabs*.

### **2.5.2. AJAX (Asynchronous JavaScript And XML)**

Es una técnica de desarrollo web para crear aplicaciones enriquecidas de internet (*Rich Internet Applications*). La idea central es que las aplicaciones se ejecutan en el cliente (*browser*), mientras se mantiene una comunicación asíncrona con el servidor en segundo plano. De esta manera se logra realizar cambios en las páginas sin necesidad de recargarlas, lo cual se traduce en aumento de la interactividad, la velocidad y la usabilidad de las aplicaciones.

Las tecnologías incluidas en *Ajax* son: *HTML*; Hojas de estilo en cascada (*CSS*); *DOM*, accedido por un lenguaje de *scripting* como *Javascript*; el objeto *XMLHttpRequest* para intercambio de datos de forma asíncrona con el servidor y *XML* que es el formato usado para la transferencia de datos.

### **2.5.3. Hojas de estilo en cascada (CSS)**

*CSS* son las siglas de *Cascade Style Sheet* que traducido significa hojas de estilo en cascada.

Las hojas de estilo es una tecnología que nos permite controlar la apariencia de una página web. En un principio, los sitios web se concentraban más en su contenido que en su presentación.

*HTML* no pone mucha atención en la apariencia del documento. *CSS* describe como los elementos dispuestos en la página son presentados al usuario. *CSS* es un gran avance que complementa el *HTML* y la Web en general.

Con *CSS* podemos especificar estilos como el tamaño, fuentes, color, espaciado entre textos y recuadros así como el lugar donde disponer texto e imágenes en la página.

El lenguaje de las Hojas de Estilo está definido en la Especificaciones CSS1 y CSS2 del *World Wide Web Consortium (W3C)*, es un estándar aceptado por toda la industria relacionada con la Web, o por lo menos, gran parte de navegadores. Podemos visitar W3C [36].

#### **2.5.4. HTML5**

HTML5 (*HyperText Markup Language, versión 5*) es la quinta revisión más importante que se hace al lenguaje *HTML*. En esta versión, se introducen nuevas características para ayudar a los autores de aplicaciones Web, y se ha prestado especial atención a la definición de claros criterios de conformidad para los agentes de usuario (navegadores) en un esfuerzo por mejorar la interoperabilidad [36].

Se usa esta tecnología porque fue la primera sugerida por los investigadores de SeCo, y en el transcurso del proyecto se realizaron pruebas con esta tecnología y con una etiqueta *CANVAS* que permite dibujar en dicha área mediante *jQuery* o *JavaScript*.

#### **2.5.5. JavaScript**

*Javascript* es un lenguaje con muchas posibilidades, utilizado para crear pequeños programas que luego son insertados en una página web y en programas más grandes, orientados a objetos mucho más complejos. Con *Javascript* podemos crear diferentes efectos e interactuar con nuestros usuarios.

Este lenguaje posee varias características, entre ellas podemos mencionar que es un lenguaje basado en acciones que posee menos restricciones. Además, es un lenguaje que utiliza Windows y sistemas X-Windows, gran parte de la programación en este lenguaje está centrada en describir objetos, escribir

funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas entre otros.

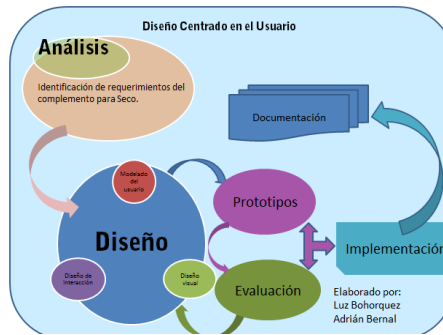
Es necesario resaltar que hay dos tipos de JavaScript: por un lado está el que se ejecuta en el *cliente*, este es el *Javascript* propiamente dicho, aunque técnicamente se denomina *Navigator JavaScript*. Pero también existe un *Javascript* que se ejecuta en el servidor, es más reciente y se denomina *LiveWire Javascript*.

## 2.6. METODOLOGÍA

Para cumplir con los objetivos trazados en este proyecto, se plantea la ejecución del marco metodológico conocido como Diseño Centrado en el Usuario o *User Centered Design* [5]. La importancia del diseño del componente radica en que modela la interacción entre el usuario y el componente, y por tanto, posibilita o no la consecución de los objetivos perseguidos por el usuario (encontrar información, comprar, reservar, aprender o guiarse en un viaje o destino en una ciudad, etc.).

En el Diseño Centrado en el Usuario, se debe tener en cuenta la usabilidad y la accesibilidad. Cuando se dice que un componente es usable, independientemente de quién y cómo lo use, corresponde con una visión o enfoque universal de usabilidad, con una visión realista y práctica. Esto se debe a que todo componente o aplicación se diseña con la intención de satisfacer las necesidades de una audiencia concreta y determinada, por lo que será más usable cuando más adaptado esté el diseño a la audiencia específica.

Un concepto íntimamente ligado al de usabilidad es el de accesibilidad. Este último hace referencia a la posibilidad de acceso. En concreto a que el diseño, como prerequisite imprescindible para ser usable, posibilite el acceso a todos sus potenciales usuarios, sin excluir aquellos con limitaciones individuales. En la figura 3 se observa un gráfico con las fases de trabajo:



*Figura 3. Fases de trabajo*

## 2.7. DSBC

### INGENIERÍA DEL SOFTWARE BASADA EN COMPONENTES

La reutilización de componentes de software es un proceso inspirado en la manera en que se producen y ensamblan componentes en la ingeniería de sistemas físicos. La aplicación de este concepto al desarrollo de software no es nueva. Las librerías de subrutinas especializadas, comúnmente utilizadas desde la década de los setenta, representan uno de los primeros intentos por reutilizar software.

Reutilización de Software. Algunas de estas definiciones son las siguientes:

- Según Somerville [39], la reutilización es un enfoque de desarrollo [de software] que trata de maximizar el uso recurrente de componentes de software existentes.
- Según Sametinger [40], la reutilización de software es el proceso de crear sistemas de software a partir de software existente, en lugar de desarrollarlo desde el comienzo.

- Según Sodhi et al. [41], la reutilización de software es el proceso de implementar o actualizar sistemas de software usando activos de software existentes.

Estas tres definiciones consideran la reutilización como un enfoque o proceso de desarrollo de software. Su principal diferencia radica en el objeto de reutilización (componente, software y activo). Tal como lo plantean Sodhi et al. [41], la reutilización de software va más allá de la reutilización de piezas de software. Ella involucra el uso de otros elementos de software, tales como algoritmos, diseños, arquitecturas de software, documentación y especificaciones de requerimientos.

Con base en el análisis de estas definiciones podemos establecer nuestra propia definición en los siguientes términos:

*La reutilización de software es un proceso de la Ingeniería de Software que conlleva al uso recurrente de activos de software en la especificación, análisis, diseño, implementación y pruebas de una aplicación o sistema de software.*

Varios estudios han demostrado la efectividad de la reutilización del software. Sametinger [40], en particular, describe algunos de estos indicadores:

- Entre el 40 y 60% del código fuente de una aplicación es reutilizable en otra similar.
- Aproximadamente el 60% del diseño y del código de aplicaciones administrativas es reutilizable.
- Aproximadamente el 75% de las funciones son comunes a más de un programa.
- Sólo el 15% del código encontrado en muchos sistemas es único y novedoso a una aplicación específica. El rango general de uso recurrente potencial está entre el 15% y el 85%.

Partiendo de estos indicadores es fácil deducir el impacto y la importancia que tiene la reutilización de componentes en el proceso de desarrollo de software;

particularmente en tres variables más importantes de la Ingeniería del Software: el costo, tiempo y esfuerzo requerido para desarrollar un producto software.

Uno de los logros más importantes de la revolución industrial fue el desarrollo por componente. Los desarrolladores tradicionales de aplicaciones incurren en altos costos y en una inversión de tiempo extensa. Otra de las ventajas es que un usuario puede integrar lo mejor de las tecnologías a su proyecto sin incurrir en gasto u costos innecesarios como licenciamiento o soporte y actualización de las grandes soluciones, ya que muchas de estas tecnologías son gratis y existen bajo la premisa de Freeware y GNU (General Public License).

El DSBC, pertenece al paradigma de programación de sistemas abiertos, los cuales son extensibles y tienen una interacción con componentes heterogéneos que ingresan o abandonan el sistema de forma dinámica, es decir que los componentes pueden ser reemplazados, por otros independientemente de su arquitectura y desarrollo [42].

Lo más importante que se debe tener en cuenta al definir el concepto de DSBC, es el de diferenciar las características existentes entre el DSBC y el paradigma de objetos, la Programación Orientada a Objetos (POO) es cuando el software puede ser desarrollado de acuerdo a un modelo mental de un objeto real o imaginado, mientras que el DSBC dice que el software debe ser desarrollado a partir de componentes prefabricados.

Existen varias definiciones de componentes realizadas por expertos que han sido los encargados del desarrollo de esta metodología, ellos han tomado como base la metodología de la programación orientada objetos y el modelado a través de UML:

- Un componente es una parte no trivial, casi independiente, y reemplazable de un sistema que llena claramente una funcionalidad dentro de un contexto en una arquitectura bien definida. Un componente se conforma y provee la realización física por medio de un conjunto de interfaces [43].
- Un componente de software en tiempo de ejecución es un paquete dinámicamente vinculado con uno o varios programas manejados como una

unidad y que son accedidos mediante interfaces bien documentadas que pueden ser descubiertos en tiempo de ejecución [44].

- Un componente de software es una unidad de composición con interfaces contractualmente especificadas y explícitas sólo con dependencias dentro de un contexto. Un componente de software puede ser desplegado independientemente y es sujeto a la composición de terceros [45].
- Un Componente de Negocio representa la implementación de software del concepto de un negocio “autónomo” o un proceso de negocio.

Que consiste de artefactos de software necesarios para expresar, implementar y poner en marcha el concepto de elemento reusable de un sistema más grande de negocios [46].

Estas definiciones no son mutuamente excluyentes, por el contrario, se complementan y construyen el significado no sólo de componente, también el significado del desarrollo basado en componentes. Sin embargo más allá de su definición existen algunas características claves para que un elemento pueda ser catalogado como componente:

- **Identificable:** Debe tener una identificación que permita acceder fácilmente a sus servicios y que permita su clasificación.
- **Auto contenido:** Un componente no debe requerir de la utilización de otros para finalizar la función para la cual fue diseñado.
- **Puede ser remplazado por otro componente:** Se puede remplazar por nuevas versiones u otro componente que lo remplace y mejore.
- **Con acceso solamente a través de su interfaz:** Debe asegurar que estas no cambian a lo largo de su implementación.
- **Sus servicios no varían:** Las funcionalidades ofrecidas en su interfaz no deben variar, pero su implementación sí.

### **2.7.1 La interfaz del componente**

Una interfaz define el conjunto de operaciones que un componente puede realizar; estas operaciones son llamadas también servicios o responsabilidades. Las interfaces proveen un mecanismo para interconectar componentes y controlar las dependencias entre ellos.

La naturaleza de la interfaz varía dependiendo del lenguaje programación empleado para implementar el componente.

### **2.7.2 Frameworks y modelos de componentes**

Existe cierta confusión en la literatura referente a la terminología de modelos y frameworks de componentes. Sin embargo, hay consenso acerca de que los sistemas basados en componentes confían en estándares y convenciones bien definidas (modelo de componentes) y en una infraestructura de soporte (framework de componentes).

Los modelos de componentes especifican las reglas de diseño que deben obedecer los componentes. Estas reglas de diseño mejoran la composición, aseguran que las calidades de servicio se logren en todo el sistema, y que los componentes se puedan desplegar fácilmente tanto en entornos de desarrollo como de producción.

Los modelos de componentes imponen estándares y convenciones sobre:

- **Tipos de Componentes:** Un tipo de componente puede ser definido en términos de las interfaces que implementa. Los tipos diferentes de componentes pueden desempeñar diferentes roles en el sistema, y participar en distintos tipos de esquemas de interacción.
- **Esquemas de Interacción:** especifican cómo los componentes son localizados, cuáles protocolos de comunicación son utilizados, y cómo se satisfacen las calidades de servicio (e.g. seguridad, transacciones, alta disponibilidad). El modelo de componentes puede describir cómo interactúan

los componentes entre sí y cómo interactúan dichos componentes con el framework.

- Asociación (bindings) de recursos: El proceso de composición de componentes es simplemente enlazar un componente a uno o más recursos. Un recurso es un servicio ofrecido por un framework o por otro componente desplegado en ese framework.

## Capítulo 3. Requerimientos

### 3.1. REQUERIMIENTOS

El proyecto de grado tiene por objetivo general responder a una propuesta del proyecto *Search Computing* (SeCo), para realizar un componente de visualización *front-end* para datos multidominio, utilizando datos orientados en el tiempo.

El componente de visualización mostrará los resultados de respuestas compuestas de SeCo, obteniendo información por agregación a través de la unión de varios dominios. El visualizador es una interfaz de usuario que puede ser elegida de acuerdo a una consulta específica, es decir, que los resultados se asocien coherentemente con el método de visualización representado. La consulta ha sido refinada con las interfaces de búsqueda que hacen uso de *Liquid Query*, al desglosar la información agregando o eliminando atributos en el transcurso de la consulta.

En el caso de este visualizador se trata de tres objetos (para el caso: Hoteles, Restaurantes y Museos, pero puede ser cualquier combinación que se corresponda con el criterio de búsquedas multidominio para la visualización con líneas de tiempo), que según su consulta, se ordenan de acuerdo a la menor distancia promedio con respecto a un lugar específico.

#### 3.1.1. Requerimientos funcionales

1. Graficar en la línea de tiempo datos de una consulta, que son el resultado de una búsqueda multidominio.

2. El componente tiene varias escalas para la línea de tiempo: semana, días y horas, en las cuales simultáneamente se encuentran relacionados los tres objetos.
3. El componente de visualización facilita al usuario escoger de un conjunto de combinaciones cual quiere que se grafique en la línea de tiempo.
4. El usuario tiene la opción de arrastrar el cursor (*drag*) por los días y horas del visualizador hasta llegar a la combinación o evento del cual quiere tener información con más detalle.
5. El usuario al colocar el cursor o puntero sobre una de las combinaciones o eventos mostrados en el visualizador despliega un efecto de *rollover* con la fecha en el cual se encuentra el evento.
6. Si el usuario quiere tener información detallada de la combinación o evento simplemente hace *click* sobre la combinación o evento y aparece un cuadro con la descripción que contiene todos los atributos de los objetos contenidos en la combinación. Este cuadro tiene efecto *drag&drop*.
7. El usuario puede realizar *zoom* de los eventos del componente de visualización.
8. Efectuar el *zoomin* y el *zoomout* es tan sencillo como mover la rueda del mouse.

### 3.1.2. Requerimientos no funcionales

**Rendimiento:** Al ser un componente de visualización para SeCo este realiza la tarea de representar la información gráfica de forma rápida y eficiente, utilizando la arquitectura de SeCo.

**Portabilidad:** Funciona con todos los navegadores. Debido a que funciona en un entorno web es multiplataforma.

El componente de visualización estará centrado en el navegador.

Del lado del cliente se usan tecnologías tales como *Html5*, *JavaScript*, con el *framework* principal *jQuery* y su librería *jQueryUI* y *TimeGlider*, librerías sugeridas por los desarrolladores de SeCo.

### **3.2. INTERACCIÓN CON EL USUARIO**

Después de haber refinado la consulta en SeCo, los datos que el usuario obtiene se pueden visualizar de diferentes maneras tales como tablas, mapas (geolocalización), diagramas de dispersión y hasta una vista atómica de los objetos en consulta, dependiendo del tipo de los mismos. En este caso se propone la visualización por medio de líneas de tiempo, que resulte en un componente de visualización simple, de fácil comprensión para que el usuario interactúe, juegue con los eventos que allí se presentan como deslizarse o arrastrarse en la pantalla para observar los eventos ya sea en días o en horas. La interacción con el usuario se basa en un modelo de sistema, en el que el usuario aprende rápidamente del paradigma objeto-acción.

### 3.3. DIAGRAMAS UML

#### 3.3.1 Caso de Uso

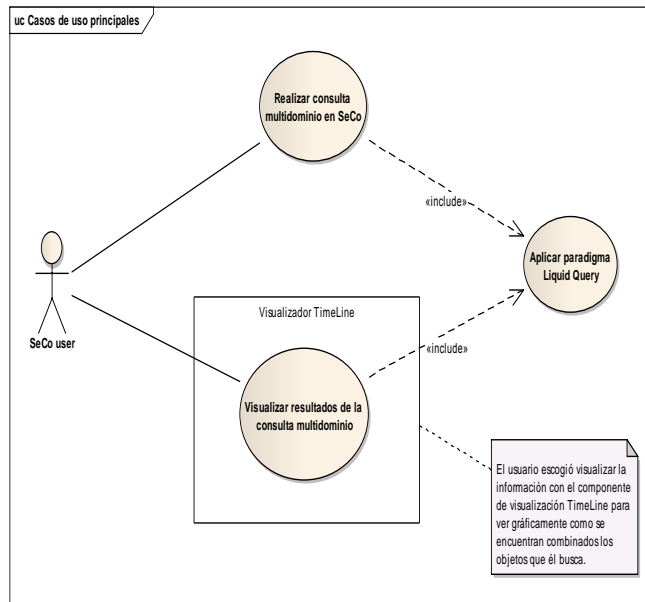


Figura 4. Caso de Uso Principal. Describe la interacción del componente de visualización con SeCo, para visualizar la consulta multidominio de un usuario SeCo.

Tabla. 1. Documentación caso de uso del componente

ACTOR(ES)	
<b>Actor</b>	SeCo User
<b>Casos de Uso</b>	Realizar consulta multidominio, Visualizar datos de resultados de una consulta multidominio.
<b>Tipo</b>	Básico
<b>Descripción</b>	Este es el actor principal de la interfaz <i>front-end</i> de SeCo, inicia todas las consultas hasta recibir el resultado.  El usuario SeCo puede ser: Proveedores de contenido, quienes desean organizar su contenido (ahora en formato de colecciones de datos, bases de datos, páginas <i>Web</i> ), para

	<p>acceso por búsqueda de terceras partes.</p> <p>Desarrolladores de aplicaciones, quienes quieren ofrecer nuevos servicios construidos por composición de contenido de dominio específico versus motores de búsqueda existente de propósito general tales como <i>Google</i> y otros.</p>	
<b>CASO DE USO</b>		
<b>Caso de Uso</b>	Visualizar datos de una consulta multidominio	
<b>Actor(es)</b>	<i>SeCo User</i>	
<b>Propósito</b>	Mostrar gráficamente la mejor combinación entre tres objetos	
<b>Descripción</b>	Después de realizada y refinada la consulta por el usuario en <i>SeCo</i> este escoge la forma en que quiere visualizar la información. En este caso es un componente de visualización basado en líneas de tiempo.	
<b>Precondición(es)</b>	El usuario inicia una consulta multidominio en la que elige los (tres) objetos y unos criterios de búsqueda. Después debe escoger la forma en que quiere visualizar los resultados. En este caso escoge el componente de visualización <i>timeLine</i> .	
<b>Flujo Principal</b>	<b>Acciones de Actor(es)</b>	<b>Respuestas del Sistema</b>
	<p>1. El usuario después de haber refinado los atributos de los objetos en la consulta en <i>SeCo</i>, da <i>click en la pestaña componente de visualización TimeLine</i>.</p> <p>3. El usuario pasa el mouse</p>	<p>2. Despliega la pantalla con las mejores combinaciones que se obtuvieron de los objetos seleccionados (Hoteles, restaurantes y museos).</p>

	<p>sobre los resultados en el área de visualización.</p> <p>5. El usuario da <i>click</i> en un evento (combinación).</p> <p>7. El usuario puede hacer <i>drag&amp;drog</i> con este pequeño cuadro en el área de visualización. Da <i>click</i> en <i>cerrar</i>.</p> <p>9. El usuario puede desplazarse en la línea de tiempo arrastrando el cursor.</p> <p>11. Cuando el usuario quiere ver los objetos en otro visualizador da <i>click</i> en la pestaña correspondiente de SeCo.</p>	<p>4. El sistema despliega el efecto <i>rollover</i> con información de la fecha en la cual está ocurriendo el evento.</p> <p>6. El sistema despliega la información en un cuadro con todos los atributos de la combinación (nombres, direcciones, categorías, distancia a).</p> <p>8. El sistema cierra el cuadro.</p> <p>10. El componente permite <i>drag</i> sobre la línea de tiempo.</p> <p>12. Cambia la vista por el componente de visualización seleccionado.</p>
<b>Sub-flujos</b>	Haber iniciado la consulta, y seguir el proceso de agregar o	

	eliminar atributos para llegar a un resultado óptimo.
<b>Poscondición(es)</b>	Que pueda pasar a otro componente de visualización al dar click.

### 3.3.2 Diagramas de clases

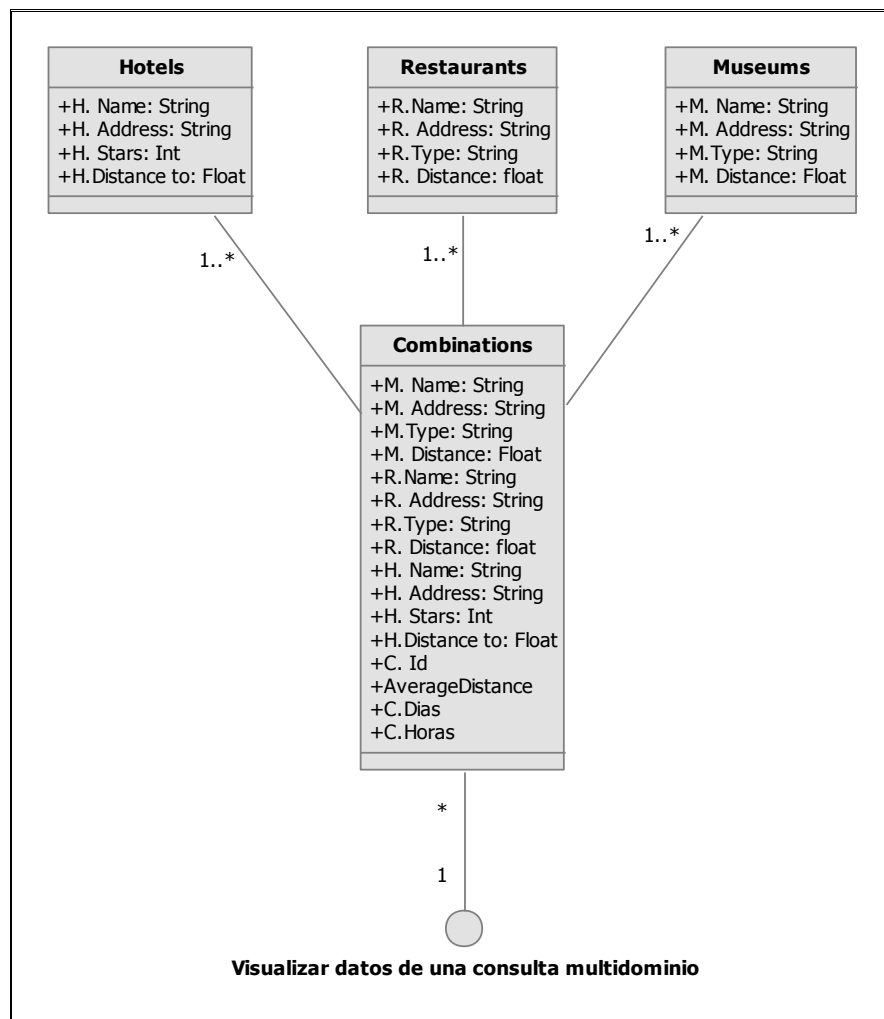


Figura 5. Diagrama de clases: para describir la estructura del componente, las clases con sus atributos para el análisis del componente.

### 3.3.3 Diagrama de Secuencia

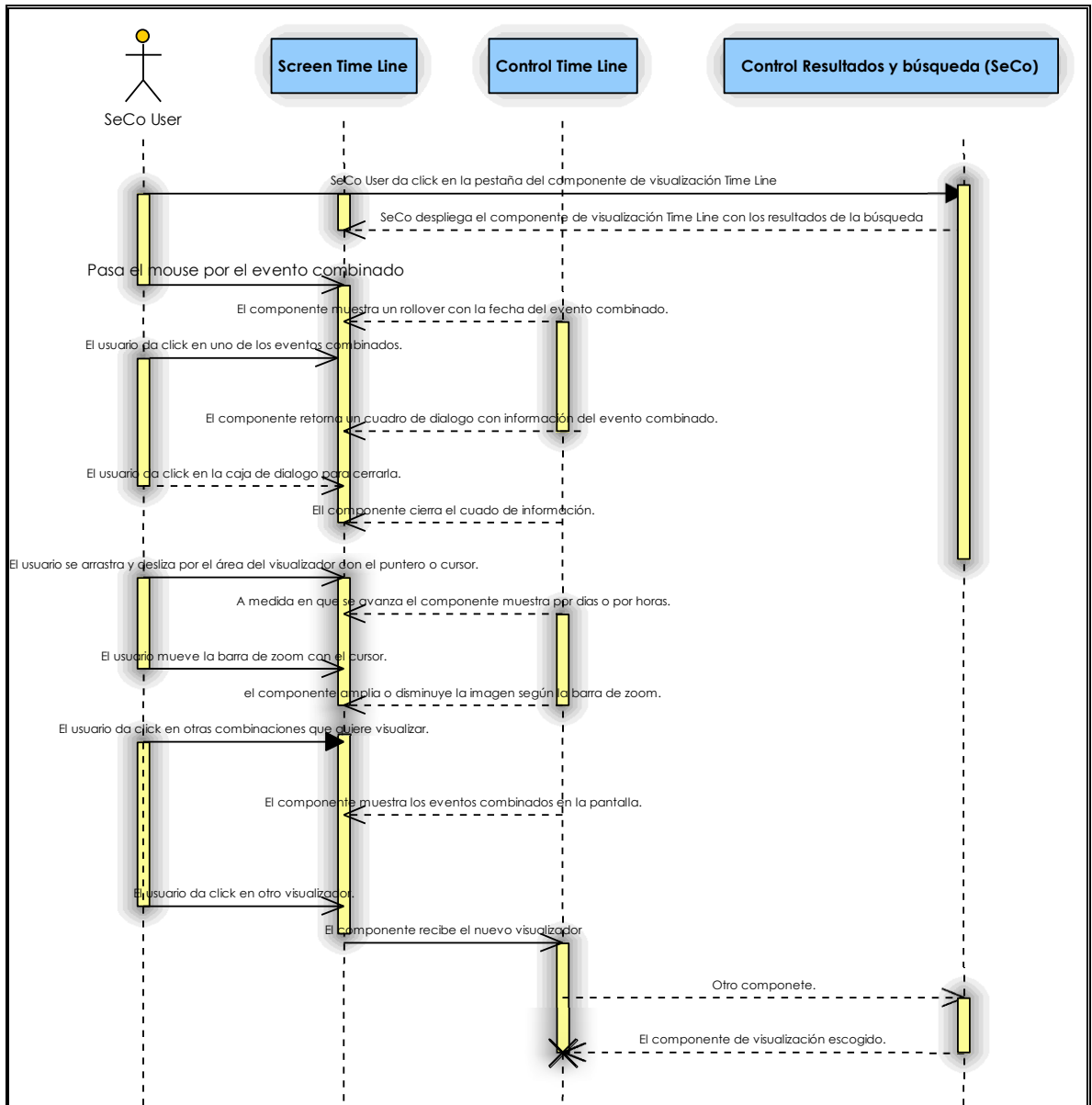


Figura 6. Diagrama de Secuencia: muestra la interacción entre el usuario y el componente de visualización.

## Capítulo 4. Diseño

### 4.1. PRINCIPIOS DE DISEÑO

Según [10], cuando se diseña una nueva aplicación, el primer paso fundamental en la creación, es definir el estado del producto (*product definition statement*), que describe que es lo que hace la aplicación, y la identificación del público objetivo.

Muchas de las aplicaciones web reflejan una falta de entendimiento de muchos de los siguientes principios. Estos principios no cambian aunque se trate de una aplicación web; es más, aplicar estos principios se vuelve más importante.

Las interfaces efectivas son visualmente comprensibles y permiten errores por parte del usuario sin dejar de operar, dándole una sensación de control. Los usuarios ven rápidamente el alcance de las opciones y comprenden como alcanzar sus metas y realizar su trabajo.

Las interfaces efectivas ocultan al usuario el funcionamiento interno del sistema. El trabajo se guarda continuamente y tiene la opción de deshacer en todo momento cualquier paso que se haya dado.

Las aplicaciones y servicios efectivos realizan el máximo trabajo requiriendo la mínima información del usuario [32].

La siguiente tabla adaptada de [10], enuncia algunos principios de diseño aplicados específicamente en la creación de interfaces gráficas de usuario (*GUI*):

Tabla. 2. Principios de diseño.

<b>Principio de diseño</b>	<b>Acciones</b>
<b>Metáforas</b>	<p>Modelar una aplicación para usar objetos y soportar acciones y controles con analogías en el mundo real es sabido que ayuda a los usuarios nuevos.</p> <p>El diseñador de la aplicación debería:</p> <p>Ser consciente de las metáforas que existen en el sistema operativo y por tanto, no redefinirlos en la aplicación.</p> <p>Identificar las metáforas naturales para las tareas de la aplicación.</p> <p>Usar controles estándar cuando no existan metáforas naturales o con las seleccionadas es improbable que el usuario promedio las reconozca.</p>
<b>Manipulación de objetos</b>	<p>El usuario tiene una fuerte afinidad con los objetos de la <i>GUI</i> cuando no hay dispositivos intermedios para manipulación.</p> <p>Para mejorar la manipulación directa, el diseñador de la aplicación debe asegurar que:</p> <p>Los objetos en pantalla permanecen visibles mientras el usuario realiza en ellos alguna acción.</p> <p>El resultado de la acción del usuario es evidente inmediatamente.</p>
<b>Solo lo que se ve</b>	<p>Siempre que sea posible se deben presentar las opciones en forma de lista al usuario. Minimizar las entradas <i>open-ended</i> reduce el riesgo de errores, al tiempo que mejora la experiencia de usuario.</p>
<b>Realimentación</b>	<p>Los usuarios deben recibir una respuesta inmediata cuando operan los controles e informes de estado durante</p>

	operaciones prolongadas: resaltar objetos seleccionados, usar animaciones y mostrar indicadores de actividad.
<b>Control del usuario</b>	Los usuarios deben iniciar y controlar las acciones y no las aplicaciones. La aplicación también puede permitir cancelar acciones antes de empezar y con “elegancia” detener operaciones en curso y pedir confirmación cuando se inicia una acción potencialmente destructiva.
<b>Integridad estética</b>	La integridad estética no es una medida de la belleza de la aplicación sino una medida de qué tan bien se integra la apariencia con la funcionalidad.
<b>Legibilidad</b>	<p>Utiliza texto con alto contraste. Procura utilizar negro sobre blanco o amarillo pálido. Evita fondos grises cuando haya texto.</p> <p>Utiliza tamaños de letra que se lean bien en los monitores más comunes. Da mayor importancia a los datos e información que quieres presentar, más que a instrucciones y etiquetas.</p> <p>Ten en cuenta a los mayores, cuya visión suele ser peor que la de los jóvenes.</p>

## 4.2. DISEÑO DE LA INTERFAZ DE USUARIO

El diseño de la interfaz se ha convertido de vital importancia hasta el punto de que un nuevo campo de estudio ha tenido que ser explorado *Human-Computer Interaction* (HCI), es una disciplina que estudia el diseño de la interfaz y cómo las personas interactúan con la información y la tecnología. Una técnica de diseño llamado *User-Centered Design* ha sido desarrollada para dar soluciones útiles, tal técnica estructura el diseño de la interfaz de usuario dentro de las siguientes fases:

- Identificación de objetivos y tareas del usuario destino.
- La creación de prototipos, diseño, evaluación y rediseño de estos.
- Obtención de la reacciones del público potencial, evaluación y revisión de los diseños con base en las reacciones del usuario.

El diseño de la interfaz, de hecho, es más una práctica que una ciencia, por lo tanto es importante disponer de un procedimiento iterativo para buscar un “óptimo local” entre el dominio de buenas soluciones dentro del espacio de diseño.

A continuación se presenta una tabla donde se muestran algunos criterios que hacen referencia a la usabilidad.

Tabla. 3. Criterios de usabilidad

<b>Usabilidad [31]</b>	<b>Descripción</b>
<b>Facilidad de aprendizaje</b>	¿Cuán fácil debe ser para un usuario sin experiencia realizar tareas básicas?
<b>Eficiencia</b>	¿Qué tan rápido puede el usuario realizar sus tareas una vez que ha aprendido a usar la interfaz?
<b>Es más fácil reconocer que recordar</b>	Después de un periodo de inactividad, ¿Cuánto tiempo necesita un usuario para restablecer la habilidad de realizar tareas en la interfaz?
<b>Errores</b>	¿Cuántos errores cometen los usuarios, cuántos de gravedad, qué tan fácil es recuperarse de estos errores?
<b>Satisfacción</b>	¿Cuán agradable y satisfactorio es para el usuario utilizar la interfaz?

## 4.3. HERRAMIENTAS DE APLICACIÓN

### 4.3.1. Paper prototyping

“Los prototipos en papel son una variación de las pruebas de usabilidad en las cuales los usuarios representativos realizan tareas realistas mediante la interacción con una versión en papel de la interfaz, la cual es manipulada por una persona que *representa* la acción del computador, quien además, no explica cómo debería funcionar la interfaz.” [38].

La creación de prototipos de papel es un método ampliamente usado en el proceso de diseño centrado en el usuario ya que permite a los desarrolladores crear software que satisfaga las expectativas y las necesidades del usuario.

Se utiliza especialmente para el diseño y prueba de interfaces de usuario ya que es útil para generar ideas de diseño y realizar revisiones internas de la interfaz. El objetivo es conseguir una rápida realimentación de los usuarios mientras el diseño continua, literalmente, en la “mesa de dibujo” [38].

### 4.3.2. StoryBoard

El *storyboard* es una herramienta útil para la elaboración de guiones, consiste en una serie de pequeños dibujos ordenados en secuencia de las acciones que se van a mostrar, de manera que la acción de cada escena se presenta en términos visuales.

El storyboard ayuda a visualizar las ideas del diseñador y es muy utilizada para el diseño visual. En el storyboard, cada dibujo va acompañado de un comentario descriptivo de la acción, con viñetas individuales que presentan las imágenes importantes del desarrollo de la historia.

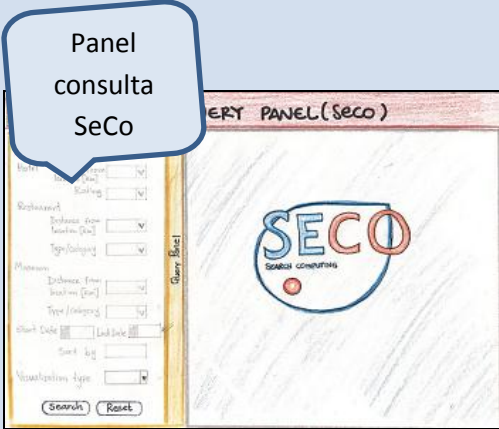
El nivel de complejidad de la storyboard va desde los dibujos más rudimentarios a los más elaborados, el objetivo es visualizar una secuencia de acciones en las imágenes.

Para realizar un seguimiento visual del diseño, se realizó una storyboard, creando y editando un video, con unos de los diseños alcanzados. Puede verse en [33].

#### 4.3.2.1. Descripción del *Storyboard* del componente de visualización

##### *Timeline*

Tabla. 4. StoryBoard del componente de visualización

<p><i>Storyboard:</i></p>	
 <p><b>Figura 7.</b> El panel de consulta SeCo</p>	<p>El usuario en el buscador de SeCo, escoge la pestaña <i>Panel de consulta</i>, en ella el usuario especifica los parámetros de consulta.</p>
<p>SeCo realiza la consulta y la primera vista de los datos la representa en</p>	

tablas, donde cada fila representa una combinación, posteriormente el usuario refina la consulta con la interfaz de *LiquidQuery*, hasta obtener los mejores resultados. Aquí se muestra la forma de visualizar las mejores combinaciones con un componente llamado Vista Atómica.

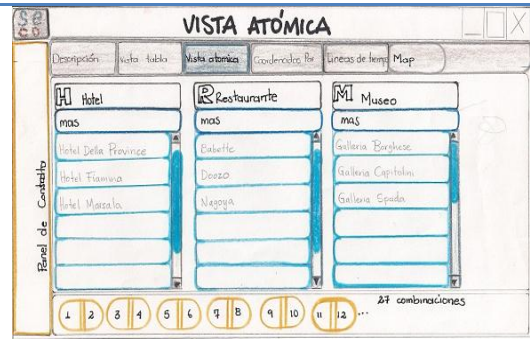


Figura 8. Después de ordenados los datos se pueden visualizar en un componente vista atómica.



Figura 9. El usuario da clic en la pestaña Líneas de Tiempo

El usuario después de refinar los parámetros de la consulta tiene la opción de escoger entre varias formas de visualizar la información, en este caso, escoge el componente de visualización de líneas de tiempo.

Aparece el componente de visualización: en la parte izquierda están los parámetros con los que se iniciaron la consulta, por tanto el componente no da la opción de modificarlo, esta área se encontrará inhabilitada para el usuario. A la derecha, se observa el área de

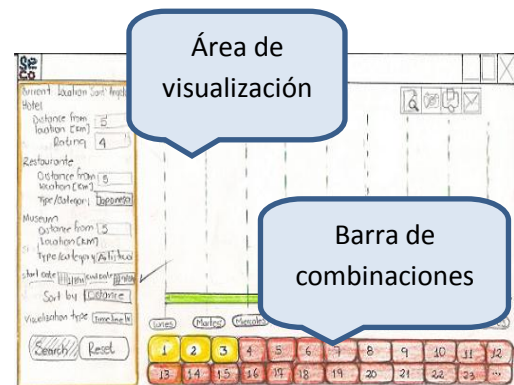
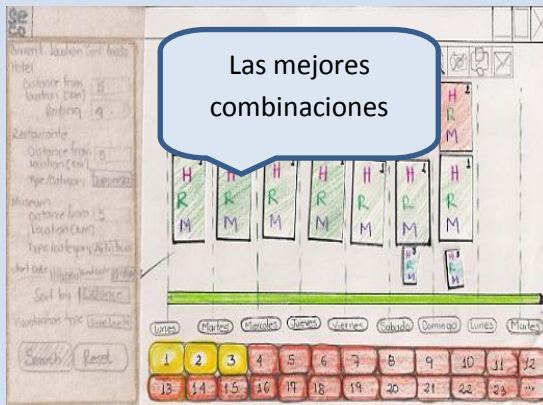


Figura 10. Área de visualización de las

visualización y en la parte inferior otras posibles combinaciones a visualizar, si así lo dispone el usuario.

combinaciones.



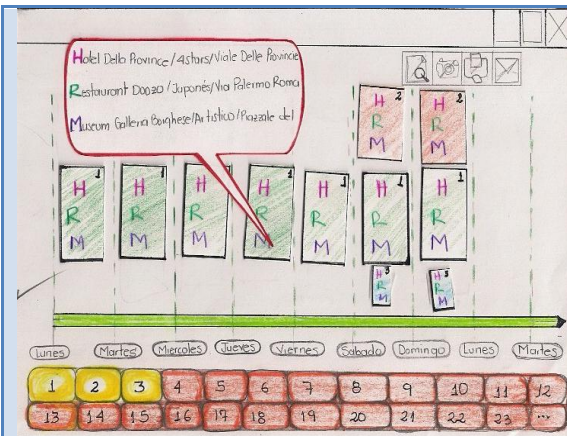
**Figura 11.** El área de visualización se muestra las mejores combinaciones.

El componente de visualización muestra las tres mejores combinaciones de los tres objetos que el usuario había escogido en los parámetros iniciales, según el horario de atención en que los tres objetos están combinados, y están ordenados de menor a mayor distancia con respecto de un lugar específico, indicando el número de combinación a la cual pertenece cada una.

Al pasar el mouse por cualquiera de las combinaciones o eventos, se realiza un rollover y se ve la fecha en la cual se encuentran combinados los tres objetos.



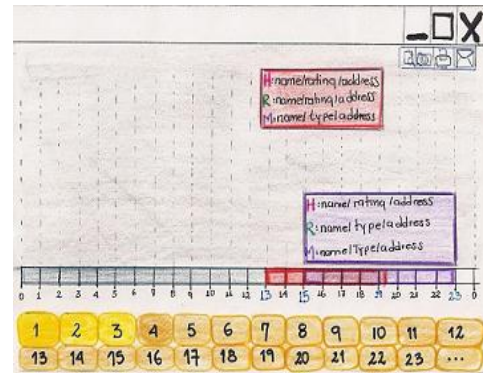
**Figura 12.** Al pasar el mouse por un evento se resalta la fecha del evento o combinación.



**Figura 13.** Al dar click se despliega una ventana con la información de la combinación.

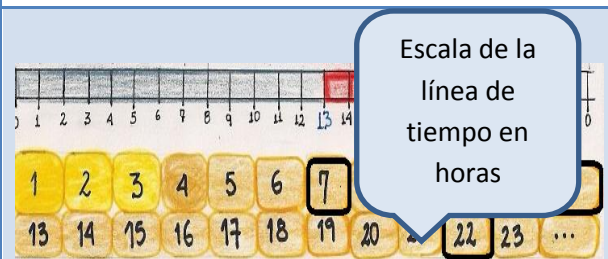
Cuando el usuario da *click* en el evento o combinación se despliega la información detallada de cada objeto con sus combinaciones; esta ventana puede ser arrastrada por cualquier parte del área de visualización. El usuario cierra la ventana con *click* en un ícono de cerrar.

El usuario puede realizar *drag & drop* y *zoom* en el visualizador y pasar de días a horas para ver en más detalle en que horarios se encuentran abiertos y atendiendo los tres objetos (hoteles, restaurantes y museos) a la vez, al igual cada combinación se encuentra rotulada con el número de combinación que representa, los eventos pueden verse más grandes o más pequeños dependiendo del *zoom*.

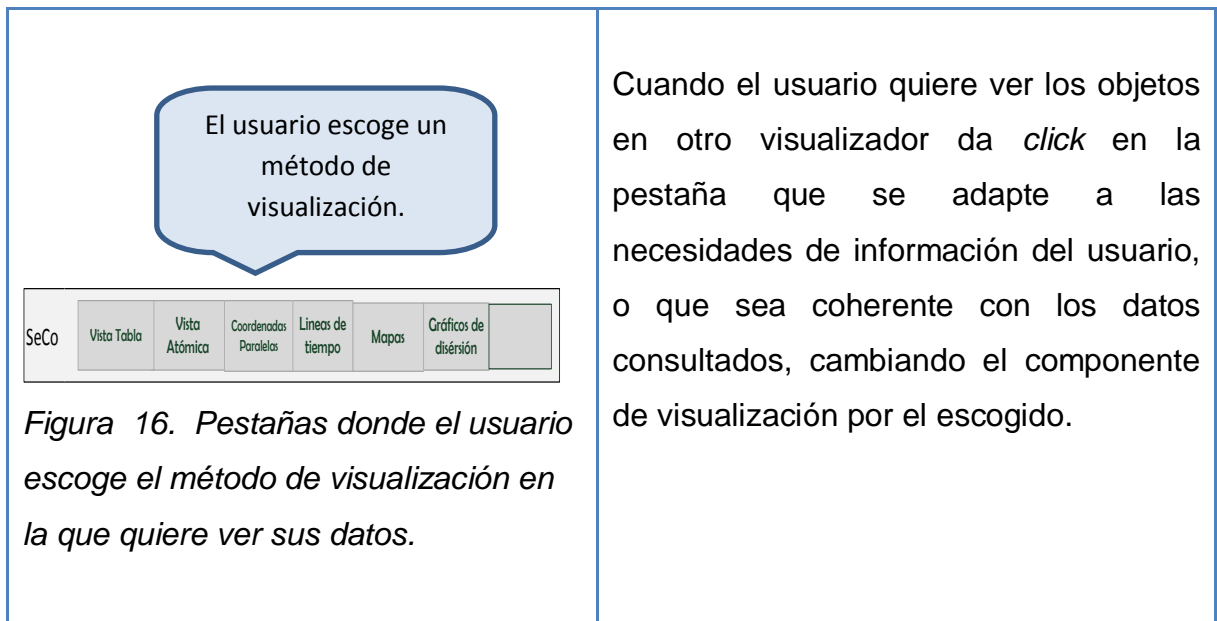


**Figura 14.** Al dar click en uno de los días se despliega una ventana con las combinaciones en las horas del día.

El usuario tiene la opción de escoger otras combinaciones dando *click* en cada una de las combinaciones de la barra de combinaciones y posteriormente ser mostrada en el área de visualización del componente.



**Figura 15.** Combinaciones para que el usuario escoja.



### 4.3.3. Mockups

En la fabricación o diseño, un *mockup* es un modelo a escala o tamaño completo de un diseño o dispositivo, es utilizado para la enseñanza, demostración y evaluación de diseño, un mockup es también llamado prototipo si proporciona al menos una parte de la funcionalidad de un sistema y a su vez permite la inspección del diseño. Los mockups se utilizan prácticamente en el diseño de un nuevo producto. Los mockups son una forma de diseñar interfaces de usuario en papel o en imágenes computarizadas con la ayuda de herramientas software.

En el caso de este proyecto se realizaron varios Mockups, tratando de buscar el diseño adecuado para este tipo de visualizador, algunos se desecharon por representar complejidad para el usuario que los ve o confusión en la información. Se describen algunos Mockups que se analizaron y estudiaron en el transcurso de la elaboración de este trabajo:

- Mockup en lápiz

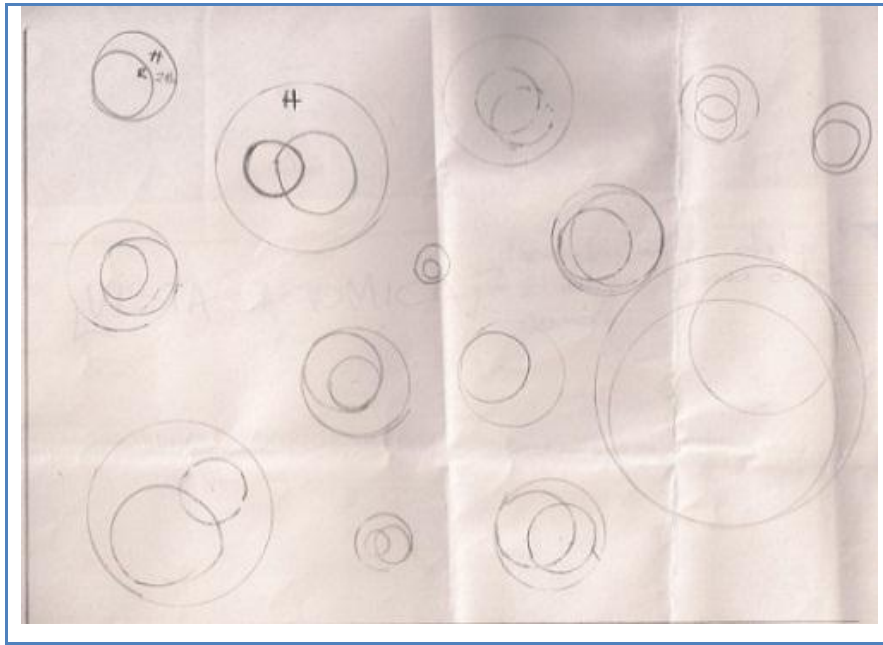


Figura 17. Imagen inicial hecha a lápiz y en papel, de un visualizador de esferas por horas.

Con esta técnica, se realizaron los primeros diseños, se inició con un pliego de papel *bond* y en él se plasmaban las ideas ocasionales.

#### 4.3.3.1 Mockup basado en diagramas de Venn

Tabla. 5. Descripción diseño de componente en diagramas de venn

**Descripción diseño del componente basado en diagramas de Venn**

Figura 18. Vista panorámica del segundo diseño de componente de visualización.

Este visualizador fue diseñado basándose en los diagramas de *Venn*, particularmente en los diagramas de *Euler*. ¿Qué problema quiere resolver este visualizador? Se quiere mostrar la intersección de las horas de un grupo de objetos (Hoteles, Restaurantes y Museos), así el usuario podrá ver de manera gráfica en que momento difieren entre sí y en qué momento están disponibles los tres al tiempo, cuál es la combinación que más le gusta, o cuál le sirve en el momento de hacer una elección, de acuerdo a los criterios de búsqueda propuestos por el usuario.

El visualizador presenta una pantalla con conjuntos de esferas que intersectan entre sí, con tres colores diferentes, que indican los tres objetos, la parte más oscura quiere decir que son las horas en que los tres objetos se encuentran disponibles al público, y el número en el centro del diagrama indica la cantidad de combinaciones de estos objetos con las mismas horas de intersección.

Como lo muestra el ejemplo del mockup, no tiene en cuenta direcciones específicamente, en este caso, son las mejores combinaciones de hoteles, restaurantes y museos de una ciudad determinada, por tanto la mejor combinación será mostrada por la esfera más grande, esto quiere decir que puede aparecer en cualquier parte de la pantalla sin un orden aparente. En el caso que se le coloque un criterio como, cerca de, o un lugar determinado, las esferas se mostrarán de izquierda a derecha, de la más grande a la más pequeña. El usuario tendrá una imagen con múltiples esferas y una columna al lado izquierdo de la pantalla que indica el nombre del criterio seleccionado (hoteles, restaurantes y museos) junto al resto de especificaciones tales como la ciudad y las horas del día, en la misma pantalla al pasar el cursor, el usuario observará que en cada combinación de esferas hay un número en el centro y además en cada una de las esferas sobresale el horario del hotel, museo y restaurante, indicando en el centro el

horario en que los tres se intersectan, como realizando un tipo de rollover, pero si el usuario quiere información más detallada solo dará un click y desplegará otra pantalla donde aparecerá una tabla con las distintas combinaciones que se encuentran en esas esferas y así podrá ser impresa o guardada.

Proceso de búsqueda:

1. El usuario quiere saber qué hoteles, museos y restaurantes pueden estar disponibles a ciertas horas del día en una ciudad determinada sin importar su ubicación, y para ello ingresa como parámetros de búsqueda la ciudad y las horas posibles de interés, y da aceptar en la información que ya tiene.

2. Inmediatamente el sistema despliega una ventana donde se encontrará una visualización de esferas unidas entre sí, que no indica la cantidad de combinaciones que hay entre hoteles, museos y restaurante en la ciudad que el usuario indicó, con un colorido llamativo y con diferente tamaño, y en desorden.

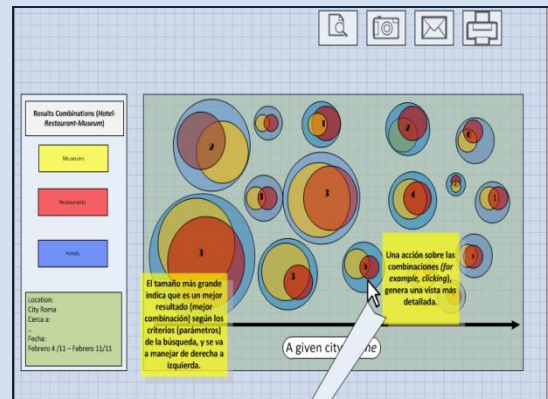


Figura 19. Imagen que muestra las combinaciones

3. El usuario notará que en el centro de cada imagen aparece un número, lo cual indica la cantidad de combinaciones con el mismo horario que existen.

4. El usuario pasa el cursor por una de

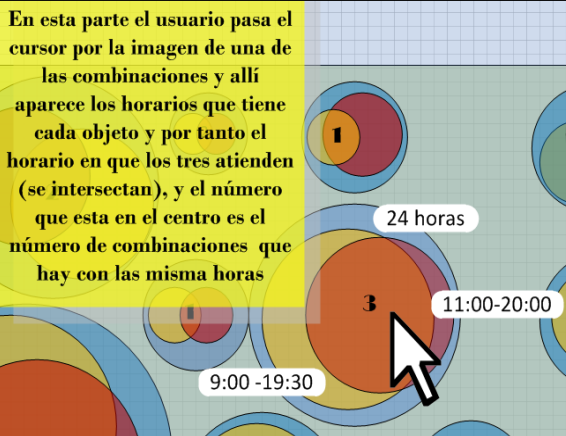


Figura 20. Vista detallada de la acción sobre cada conjunto de combinaciones.

las combinaciones y allí se realiza un *rollover* en cada una de las esferas indicando el horario en que atiende cada uno por separado y en la mitad el horario en que están disponibles los tres y por supuesto, el número de combinaciones.

5. Aquí el usuario puede observar cual es el horario que más se ajusta a su búsqueda, y puede dar click en la que más le llame la atención para ampliar la información, y entonces se despliega otra pantalla con la información de las combinaciones que nos indicaba la esfera anterior, esta información se muestra en una tabla representada con los mismos colores que identificaban a cada una de las esferas.

Results Combinations (Hotel-Restaurant-Museum)			
Com b	Hotels	Restaurants	Museums
1	<p>Hotel name:</p> <p>Address:</p> <p>Phone:</p> <p>City:</p> <p>Country:</p>	<p>Restaurant name:</p> <p>Address:</p> <p>Phone:</p> <p>City:</p> <p>Country:</p>	<p>Museum name:</p> <p>Address:</p> <p>Phone:</p> <p>City:</p> <p>Country:</p>
2	Hotels	Restaurants	Museums
3	Hotels	Restaurants	Museums
4	Hotels	Restaurants	Museums
5	Hotels	Restaurants	Museums

Figura 21. Vista de tabla después de realizarle click en una de las combinaciones.

6. Teniendo esta información el usuario, puede:

Guardar en su computador la imagen como una foto, ya sea de las esferas, o



ScreenShot

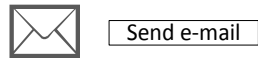


Figura 22. Imágenes de las herramientas que ofrece el componente de visualización.

de la tabla de combinaciones.

Dar *click* en el icono de impresiones, y si la computadora está conectada con una impresora se dan la facilidad de imprimir tanto la imagen de las esferas como las de la tabla.

Si no cuenta con una impresora o una computadora personal y quiere tener la información disponible, y no tener que buscar de nuevo, puede enviarla a una dirección de correo.

7. En el caso de que el usuario quiere buscar cerca a: la visualización aparecerá de manera ordenada de izquierda a derecha, el de mayor tamaño es la combinación que indica cercanía, los de menor tamaño indica lejanía, además contará con los eventos anteriores.

#### 4.3.3.2. Descripción del mockup inicial del visualizador *Timeline*

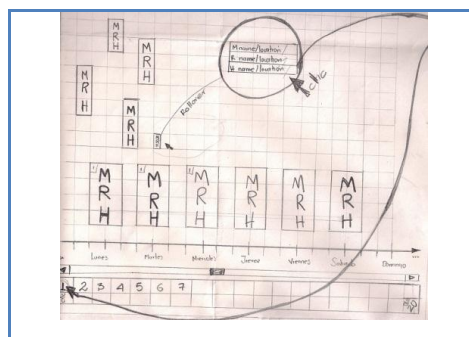


Figura 23. Primer diseño en papel y lápiz del componente de visualización en barras.

Tabla. 6. Descripción de la primera versión del componente

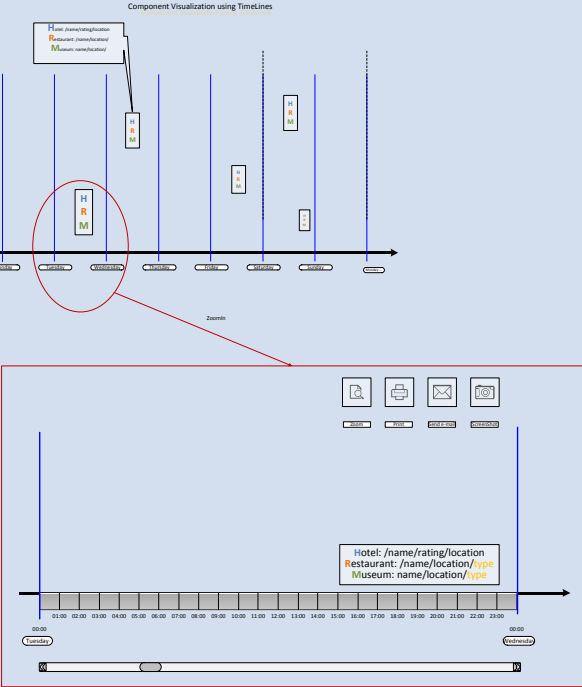
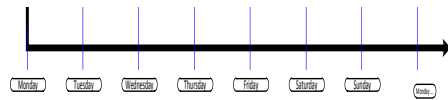
Descripción primera versión del diseño del componente	
	<p>Un usuario quiere saber qué restaurantes, museos y hoteles están disponibles a ciertas horas del día y que se encuentren cerca de un lugar específico dentro de una ciudad en particular.</p> <p>Ejemplo de una consulta: Encontrar hoteles, restaurantes y museos en la ciudad de Roma, ordenados de acuerdo a la menor distancia promedio con respecto a un lugar específico, mostrando los horarios de atención al público y precio entre-semana.</p> <p>El usuario realiza la consulta cuyos resultados de búsqueda serán combinaciones, en este caso, de tres objetos (hotels, restaurants and museums) y cuyos criterios de búsqueda serán: ciudad, localización o ubicación, fecha.</p> <p>El usuario ingresa estos datos y da <i>click</i> en aceptar e inmediatamente se despliega una ventana donde se tiene, al lado izquierdo, los</p>

Figura 24. Visualizador de barras mockup realizado en Visio.

datos que se utilizaron como criterios de búsqueda.



En esta imagen, se observa que el intervalo es una semana.

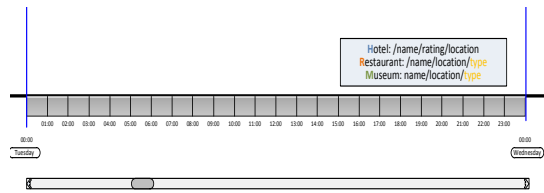


Figura 25. Escala de tiempo por horas. Vista zoom de un intervalo de un día.

Al lado derecho, en el área principal de la pantalla, aparece una barra horizontal que indica la línea de tiempo, a su vez con la representación de la escala de tiempo (*days, hours*), indicando el intervalo determinado en la búsqueda.

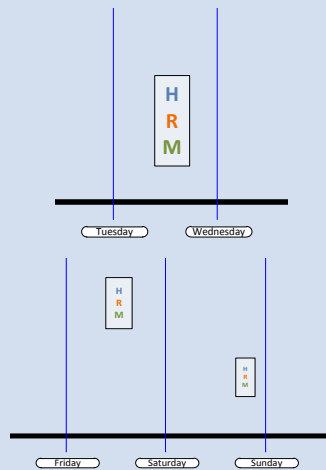
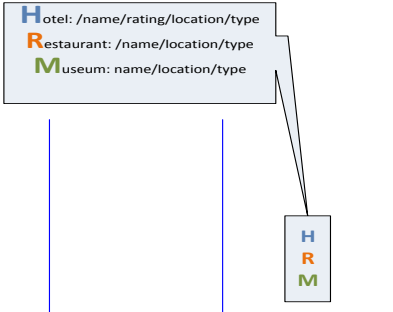



Figura 26. Las combinaciones en los días.

Sobre la línea de tiempo, aparecen representadas en rectángulos, las combinaciones ( $H-R-M$ ), destacando que los rectángulos de mayor tamaño son los mejores resultados de acuerdo al *rank* de la consulta.

 <p><i>Figura 27. Rollover con la información de la combinación.</i></p>	<p>Cuando el usuario hace <i>clicking</i> o <i>rollover</i> sobre cualquier combinación, se genera una vista con mayor detalle sobre los objetos y sus atributos.</p>
<p><i>En la parte superior de la ventana, aparecerían los siguientes iconos:</i></p>  <p><i>Figura 28. Herramientas del visualizador.</i></p>	<p><i>Zoom:</i> Al dar <i>click</i> en una parte de la pantalla (intervalo de tiempo), se muestra una escala de tiempo más detallada (e.g. <i>hours scale</i>).</p> <p><i>Print:</i> Muestra el cuadro de control para imprimir la sección actual.</p> <p><i>Send E-mail:</i> Envía la información actual por correo electrónico.</p> <p><i>ScreenShot:</i> Genera una captura de pantalla en formato de imagen.</p>

#### 4.3.3.3 Mockup última versión del componente de visualización

El prototipo del último componente de se realizó en Microsoft Visio, una herramienta fácil de usar que permite el manejo de colores y diagramas.

Tabla. 7. Descripción componente final

### Descripción última versión de diseño del componente

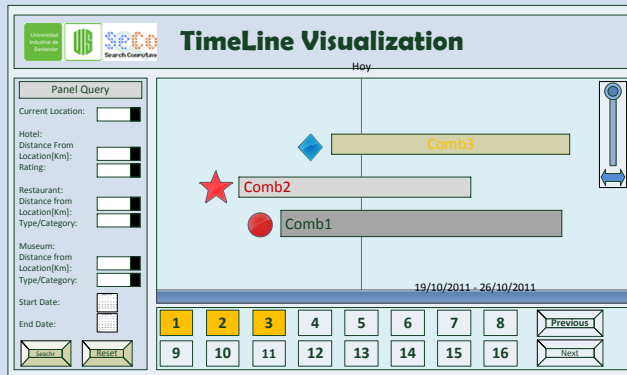


Figura 29. Componente de visualización Time Line

Componente de visualización con un área de visualización amplia en donde se puede observar las diferentes combinaciones, con iconos o imágenes que identifiquen el evento de visualización, a la izquierda se encuentra el panel de consulta con la información de la consulta inicial del usuario, en la parte de abajo las diferentes combinaciones existentes entre los objetos.

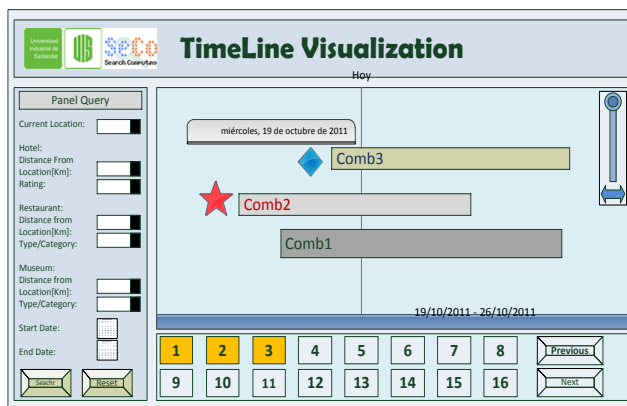


Figura 30. Rollover con la fecha del evento de la combinación.

Cuando se pase el mouse sobre un evento o combinación se despliega la fecha en la que sucede (inicia) el evento. Se tiene una línea en la mitad de todo el visualizador que sirve de referencia de la fecha actual.

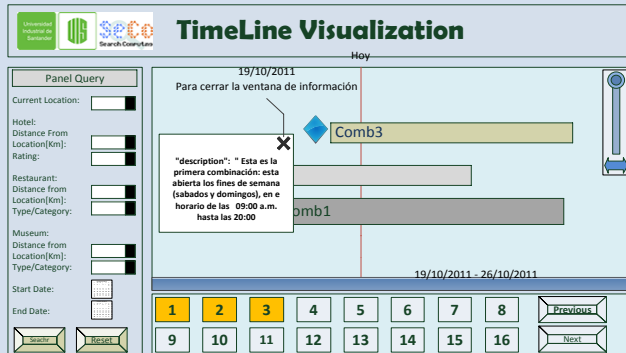


Figura 31. Se despliega una ventana con la información de la combinación.

Cuando se da clic en un evento se despliega una ventana con la descripción detallada de la combinación (hotels, restaurants and museums) y los horarios de atención de los tres objetos.

Esta ventana se pueda acomodar en cualquier parte del área de visualización por poseer el efecto *drag & drop*.



Figura 32. Herramienta para dar zoom y pasar fechas

Esta herramienta sirve para dar *Zoom* en el visualizador a los eventos que queremos ver detalladamente por horas. Las flechas dan al visualizador la habilidad de avanzar o retroceder en la línea de tiempo.

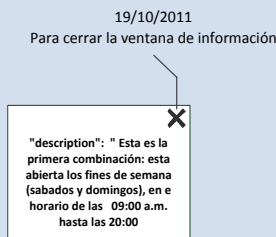


Figura 33. Ventana con la información de la combinación, se cierra en la X.

Cuando se da clic en algún evento se despliega esta pantalla con la información relevante de la combinación (Hotels, restaurants and museums), con sus horas de atención, se cierra en la parte superior dando clic.

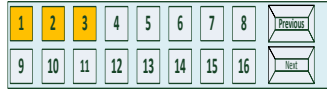


Figura 34. Herramienta para escoger otras combinaciones.

Esta herramienta le da al usuario la posibilidad de escoger entre las combinaciones posibles a visualizar en el componente, con solo dar clic se resalta con otro color y se dibuja en el área de visualización.

#### 4.4. SIMULACIÓN CON DATOS

*Objects: Hotels, Restaurants and Museums.*

Para el ejemplo tomamos 3 Hoteles, 3 Restaurantes y 3 Museos. Especificamos un conjunto de datos (atributos) de la siguiente manera: Cada objeto tiene la distancia en metros con respecto a un punto específico (para el ejercicio de prueba tomamos como punto de referencia el *Castillo Sant'Angelo*). Cada combinación tiene la suma total de distancias y su promedio correspondiente. En la tabla de datos (puede verse en [34]) se agrega un *ranking* de menor a mayor distancia promedio. Aparece también los días de disponibilidad de la combinación y las horas.

El análisis hecho en cada uno de los componentes diseñados, fue enviado a:

Marco Brambilla (mbrambil@elet.polimi.it), Piero Fraternali (piero.fraternali@polimi.it), Alessandro Bozzon (bozzon@elet.polimi.it), Maristella Matera (matera@elet.polimi.it), investigadores del proyecto SeCo:

**Marco Brambilla dice:**

The timeline is fine. The second example instead (*with venn diagrams*) is very complex. I think nobody would understand it. Furthermore, I think it has an intrinsic problem: you cannot show separate *venn* diagrams in the same XY plot with no overlaps like this, because it means there are cluster of independent objects, while I don't think it's the case in the example. To do:

1. See if you find a much simpler use case.
2. Position everything in the same venn diagram.

If impossible, I would drop this option.

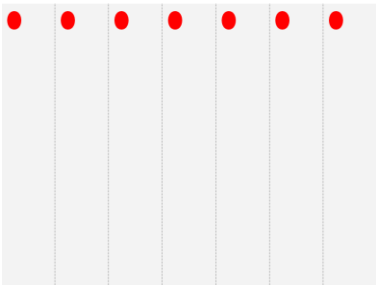
**Piero Fraternali dice:**

As a general remark, try to be simple...

I think it is a good start. Did you also start practising with the technologies (e.g., some JavaScript libraries)?

## 4.5. PRÁCTICA CON LIBRERÍAS JAVASCRIPT Y CANVAS

Tabla. 8. Descripción de las prácticas realizadas con el componente

 <p><i>Figura 35. Primera interacción con</i></p>	<p>1. Descripción Mi primer <i>canvas</i>: En esta parte teniendo en cuenta el diseño aceptado por los profesores de Italia, se iniciaron pruebas para insertar dibujos en cada <i>canvas</i>, que a su vez estaban dentro de los <i>div</i>'s, para ello se utilizó la librería <i>Jcanvas</i> un <i>plugins</i> que facilita</p>
--	--

<p><i>Canvas, días de la semana con imágenes que representaran las posibles combinaciones.</i></p>	<p>la interacción con el <i>canvas</i>.</p>
<div data-bbox="376 726 784 1005" data-label="Image"> </div> <p><i>Figura 36. Insertando imágenes en el Canvas, describe como aparecen las imágenes prediseñadas en el Canvas.</i></p>	<p>2. Descripción Imágenes en el <i>canvas</i>:  Teniendo en cuenta el diseño, se iniciaron pruebas para insertar imágenes prediseñadas en cada <i>canvas</i>, que a su vez estaban dentro de los <i>div's</i>, para ello se utilizó la librería <i>Jcanvas</i> un <i>plugins</i> que facilita la interacción con el <i>canvas</i>, en esta parte se puede observar la cantidad de posibles combinaciones que se pueden visualizar en el componente, de la importancia de los colores y el tamaño de acuerdo a cada combinación, además de tener la franja que identifica cada día con su respectivas combinaciones.</p>
<div data-bbox="380 1388 781 1705" data-label="Image"> </div> <p><i>Figura 37. Distribución del espacio del componente de visualización.</i></p>	<p>3. Descripción distribución de espacios del componente: teniendo en cuenta la información que se va a llevar en el componente, se distribuye el espacio del componente, como lo muestra la siguiente figura.</p>

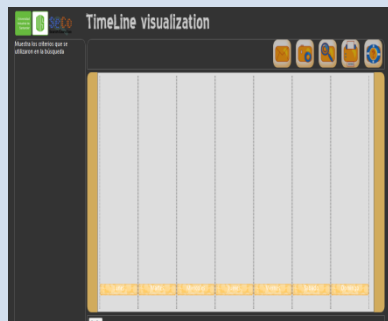


Figura 38. Utilizando JQueryUI

4. Descripción para el manejo de colores: Se distribuyó como iban a quedar los espacios en el visualizador además de querer jugar con colores, para ello de su uso librerías de JavaScript, en este caso se usó JQueryUI, que proporciono ayuda en el manejo de colores y en el diseño.

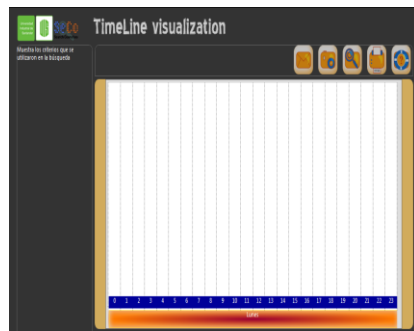


Figura 39. El componente de visualización en horas de cierto día.

5. Descripción por días: Al dar *click* en el día aparece otra ventana con la información de la combinación, pero con las horas en la que las tres combinaciones atienden.

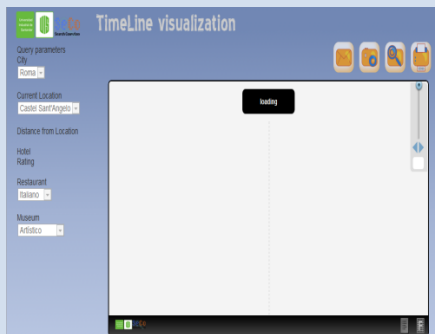


Figura 40. Diseño final del

En el estudio de *Jquery* y de las diferentes librerías de *JavaScript* y atendiendo sugerencias de los investigadores de *SeCo*, se encontró una librería que facilita el mostrar los días y las horas de manera más agradable así que se tomó esta librería



## Capítulo 5. Implementación

En este capítulo especificamos detalles del desarrollo y las herramientas utilizadas. También presentamos algunos pantallazos con los resultados de funcionamiento.

Como ya ha sido mencionado, este es un componente software para la visualización de resultados de consultas multidominio basado en líneas de tiempo.

### 5.1. FUNCIONAMIENTO DEL COMPONENTE

El componente de visualización pertenece a la parte *front-end* de la arquitectura de SeCo. El usuario empieza a construir la consulta a través de interfaces de búsqueda y mediante la aplicación de la herramienta flexible de búsqueda de información denominada *Liquid Query* puede refinar la consulta. Los datos que resultan de la consulta, son visualizados de acuerdo a un método de visualización que sea coherente con la naturaleza de los mismos. El componente desarrollado en este proyecto tiene un enfoque en los datos orientados en el tiempo, por lo tanto es adecuada la visualización en una línea de tiempo.

Los resultados de la consulta suceden a través de una petición al servidor. Los resultados son devueltos hacia la aplicación del lado del cliente en formato *JSON* (*JavaScript Object Notation*). Este es el formato adecuado para que los datos sean procesados y mostrados en el componente de visualización.

Como se detalló en el capítulo de diseño, se creó una tabla que contiene las combinaciones de los objetos junto con sus atributos. Después se adaptó al formato *JSON*, el cual es el indicado para recibir los resultados de la consulta y ser utilizados por las librerías del componente de visualización.

## 5.2. FRAMEWORK Y LIBRERIAS UTILIZADAS

El componente de visualización corresponde con una interfaz gráfica que debe desarrollarse con base en la creación de aplicaciones enriquecidas para internet (*RIA*), por lo tanto debe ser construido y desarrollado con tecnologías orientadas hacia la web.

El lenguaje utilizado para manejo de objetos mediante *scripts*, como ya se dijo en el capítulo 2, es *javascript* y para su fácil manejo se aprendió y se utilizó la librería o framework *jQuery*, la cual simplifica la programación en *javascript*.

También se utilizaron las siguientes librerías complementarias cuyo desarrollo es basado también en *jQuery*.

### ***jQueryUI***

Es una librería de código abierto de componentes e interacciones de interfaz gráfica. Proporciona abstracciones de bajo nivel de interacción y animación, efectos avanzados y *widgets* con fuentes y colores personalizables.

### ***TimeGlider*** [35]

Es una librería de código abierto construida en javascript especialmente para la representación de datos en líneas de tiempo.

Con esta librería se puede obtener características como: zoom, línea de tiempo *arrastrable* y continua, y detalles de los eventos.

Se puede hacer *zoom* en el área de la línea de tiempo arrastrando el *slider* vertical: subir para aumentar (*zooming in*, se muestra menos tiempo, escala

mayor); bajar para reducir (*zooming out*, muestra más tiempo, escala más pequeña). La rueda del *mouse* también funciona como control de *zoom*.

En cualquier parte en blanco del escenario o en la franja azul donde se indica el tiempo se puede arrastrar a la derecha y a la izquierda para observar la continuidad “en línea” del tiempo. Al hacer *click* en un evento, se abre una pequeña ventana modal que puede contener elementos como la descripción, imágenes y enlaces.

### ***BubbleUp* [46]**

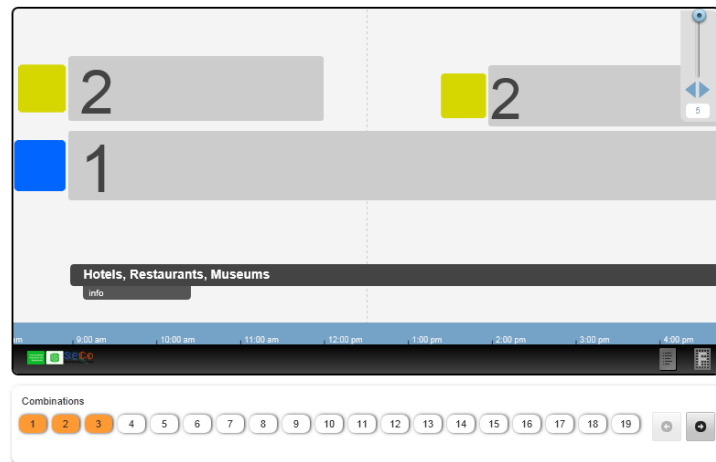
Librería que permite hacer listas de imágenes con un efecto suave cuando se pasa el mouse sobre alguna de éstas; cuando sucede el evento *mouseover* la imagen se agranda y cuando el evento deja de existir, la imagen vuelve a su tamaño original.

## **5.3. DISEÑO Y FUNCIONALIDAD**

En la parte izquierda del componente de visualización se muestran los criterios que se utilizaron en la búsqueda multidominio. Se denomina búsqueda multidominio porque asocia múltiples objetos a partir de uno o varios criterios de tal forma que se convierte en una búsqueda a la que se agrega semántica y por tanto, su complejidad aumenta comparada con una búsqueda común de palabras en un buscador de propósito general.

En este componente se debe visualizar los resultados de una consulta que asocia tres objetos (hoteles, restaurantes y museos) relacionados mediante el criterio de distancia con respecto a un lugar específico, el cual a su vez determina el ordenamiento (*rank*) de los resultados.

Como el esquema de visualización está enfocado en datos orientados en el tiempo, con representación de líneas de tiempo, los atributos importantes en la visualización son precisamente datos relacionados con el tiempo: el momento en que los objetos (hoteles, restaurantes y museos) están funcionando o están abiertos al público. En las siguientes figuras se observan combinaciones enfocadas al nivel de *zoom* de horas; en la Figura 42, dos combinaciones, de las cuales el evento 2 está dividido el día en dos rangos de horas; en la Figura 43, se observan tres eventos diferentes. En la parte inferior, se aprecian botones que representan cada una de las combinaciones



*Figura 42. Combinaciones en la línea de tiempo con escala por horas.*

Conceptualmente aparece la noción de *evento* como definición de lo que se puede representar en el tiempo. Se debe tener en cuenta la apreciación de que en el tiempo ocurren sucesos o eventos. Así pues, una *combinación* (o un resultado) se considera un evento por el sentido de lo que representa.

Cada barra horizontal representa una combinación (evento) que identifica en el tiempo, la disponibilidad o funcionamiento de cada hotel, restaurante y museo que hace parte de un resultado, es decir, cada evento corresponde a una combinación de dichos objetos.

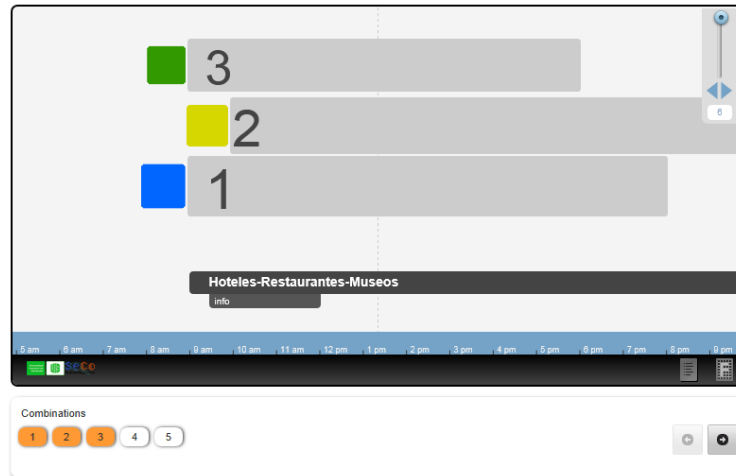


Figura 43. Combinaciones (eventos) en la línea de tiempo.

## Capítulo 6. Evaluación

## 6.1. EVALUACIÓN HEURÍSTICA

En este trabajo se utiliza el método de inspección, pero en especial la técnica de evaluación heurística, es una variante de la inspección de usabilidad donde se juzga si cada elemento de la interfaz de usuario sigue los principios de usabilidad establecidos. La Evaluación Heurística puede ser utilizada en, prácticamente, cualquier momento del ciclo de desarrollo, aunque probablemente se adapta mejor en etapas tempranas, cuando no hay material lo suficientemente firme para efectuar un test. No requiere, pues, gran despliegue de medios y puede ser llevado a cabo por personal no especializado e incluso por usuarios tradicionales. Ya son un clásico en la materia las heurísticas de Jakob *Nielsen*.

¿Cómo llevar a cabo una evaluación heurística?

La evaluación heurística es un método de ingeniería de la usabilidad para encontrar los problemas de usabilidad en un diseño de interfaz de usuario de modo que pueda ser atendida como parte de un proceso iterativo. La evaluación heurística consiste en tener un pequeño número de evaluadores que examinan la interfaz y juzgan su conformidad con los principios de usabilidad reconocidas (la "heurística").

La evaluación heurística es una evaluación difícil, para ser realizada por un solo usuario; porque una persona no es capaz de encontrar todos los problemas de usabilidad en una interfaz. La experiencia de la evaluación heurística en muchos proyectos ha demostrado que diferentes personas encuentran diferentes problemas de usabilidad, por lo tanto, es posible mejorar la eficacia del método mediante la participación de múltiples evaluadores.

La evaluación heurística se lleva a cabo haciendo que cada evaluador inspeccione la interfaz solo o individualmente. Este procedimiento es importante para asegurar evaluaciones independientes e imparciales de cada evaluador. Los resultados de

la evaluación se pueden presentar como informes escritos de cada evaluador o un informe verbal de cada evaluador a un observador, pero el informe escrito tiene ventaja porque presenta un registro formal de la evaluación realizada.

Cuando los evaluadores tienen problemas para utilizar la interfaz, se les puede dar pistas sobre cómo proceder a fin de no perder el tiempo valioso de evaluación que luchan con la mecánica de la interfaz. Es importante señalar, sin embargo, que los evaluadores no deben recibir ayuda hasta que están claramente en problemas y han comentado el problema de usabilidad que se trate.

Por lo general, una sesión de evaluación heurística de un individuo evaluador dura una o dos horas. Durante la sesión de evaluación, el evaluador va a través de la interfaz varias veces inspeccionando cada elemento de diálogo comparando con la lista de principios de usabilidad reconocidos (la heurística).

### **Diez heurísticas de usabilidad para nuestro componente de visualización [37]:**

#### **1. El estado del sistema ha de estar siempre visible.**

El sistema siempre debe mantener informado al usuario sobre lo que está pasando, a través de información adecuadas en un tiempo razonable.

El componente de visualización time line esta visible, siempre y cuando el usuario lo llame desde SeCo y lo tenga en uso.

#### **2. Coincidencia del sistema y el mundo real.**

La información que se muestra en el componente de visualizador hace referencia a las diferentes combinaciones, por tanto, se muestra la información con frase o palabras con son familiares en el diario vivir del usuario.

#### **3. Libertad y control por parte del usuario.**

El usuario puede escoger la combinación que quiere ver o ser mostrada por el componente de visualización, además de jugar con el componente a través de los días u horas, con el *drag&drop*, aumentando o disminuyendo los eventos según quiera verlos.

#### 4. **Consistencia y estándares.**

El componente cumple con el ideal de un usuario satisfecho para ello ha escogido iconos e imágenes con los que el usuario podrá identificarse, además del buen uso de colores y fuentes.

#### 5. **Prevención de errores.**

El componente está diseñado solo para mostrar los que el usuario ha refinado en SeCo, por tanto, evita que haya errores en el uso del componente de visualización.

#### 6. **Reconocimiento en lugar de recordar.**

El usuario no recordara, si no que reconocerá los iconos de relevancia y eventos de combinación por sus colores.

#### 7. **Flexibilidad y eficiencia.**

A menudo se puede acelerar la interacción para un usuario experto de tal manera que lo que se realice pueda ser útil tanto al usuario experto como inexperto. Permitir a los usuarios adaptar acciones frecuentes.

El componente es flexible ya que en él se pueden observar varias combinaciones y jugar de manera que al usuario le parezca agradable y eficiente al mismo tiempo. Además de contar con herramientas que a un usuario experto le serán fáciles de reconocer y de utilizar.

#### 8. **Diseño estético y minimalista.**

Se consideró que el componente no debe estar recargado de información y de rótulos, por ello se hizo que fuera un diseño ergonómico y fácil de usar, con colores suaves y fuertes que resalten la importancia de las partes del componente. El cuadro de dialogo que sale al realizar clic muestra la información relevante de cada combinación de manera clara y legible.

9. ***Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de los errores.***

Esta parte en el componente no se implementó, ya que se hizo con la idea de evitar errores por parte de usuario.

10. ***Ayuda y documentación.***

A pesar de que es mejor si el sistema puede ser usado sin documentación, puede ser necesario para proporcionar ayuda y documentación. Toda esta información debe ser fácil de búsqueda, centrado en la tarea del usuario, los pasos concretos lista para ser llevado a cabo, y no ser demasiado grande.

El componente de visualización brinda soporte al usuario con *tips* para que sepa cómo utilizarlo, y en que consiste la información que se le brinda.

*Atendiendo las recomendaciones y tips de la evaluación heurística se diseñó la siguiente prueba que se aplicó a cinco usuarios que tienen alguna experiencia con entornos web.*

## 6.2. RESULTADO DE LA EVALUACIÓN

### **DESCRIPCION DEL TRABAJO:**

Se realizaron 3 evaluaciones heurísticas a tres expertos y tres usuarios comunes, en la evaluación se evaluaron aspectos generales, identidad e información, lenguaje y redacción, rotulados, estructura y navegación, Lay-out del componente, búsqueda del componente, elementos multimedia, ayuda, accesibilidad, control y realimentación, para ello se utilizaron dos computadoras con acceso a internet banda ancha de la Universidad Industrial de Santander, se realiza la evaluación a cada experto por aparte y en recinto cerrado, permitiendo a cada uno tener accesibilidad total del componente.

Al evaluador se le entrega dos hojas con una serie de preguntas para evaluar cada ítem mencionado, cada ítem se debe responder con un sí o un no.

### PRINCIPALES HALLAZGOS:

#### **Aspectos Generales:**

Todos los evaluadores concluyeron que conocían el objetivo del componente, y que el contenido se adaptaba al objetivo del componente, sintieron que la estructura estaba orientada al usuario y todos coincidieron en este aspecto, el manejo de los colores y la coherencia del componente fue aceptada y aprobada con un sí rotundo en todas las casillas.

#### **Identidad e Información:**

El logotipo y el eslogan fueron aprobados con un sí, por parte de todos los evaluadores.

#### **Lenguaje y redacción:**

En este caso hubo tres de los usuarios que no estaban familiarizados con el idioma en que se planteó el componente y por lo tanto su respuesta fue un no en la casillas de respuesta, y sin embargo el componente les pareció amigable, familiar y cercano.

**Rotulado:**

El título de la página les pareció correcto a todos los evaluadores, y los rótulos usados fueron adecuados porque en todas las casillas respondieron sí.

**Estructura y navegación:**

Hubo confusión en cuanto si a algún elemento se le podía hacer click, porque esta parte la mayoría de los evaluadores respondieron que no, reconocieron que no había enlaces huérfanos, ni sobrecarga de estos, que la estructura y navegación del componente es adecuada.

**Lay-Out del componente:**

Se aprovecha de las áreas de mayor tamaño para mostrar información relevante, evitando la sobrecarga de información, con una interfaz limpia y sin ruido, existen áreas o zonas blancas para que el usuario pueda descansar la vista usando correctamente el espacio de visualización del componente, controlando el tamaño del componente.

**Búsqueda:**

Se resalta que el componente de visualización *TimeLine* es reconocible como tal, es fácil de encontrar, mostrando los resultados de la búsqueda de forma comprensible para el usuario.

**Elementos multimedia:**

No se estuvo de acuerdo con los iconos que referenciaban las combinaciones, pero que el uso de iconos es adecuado para diferenciarlas.

**Ayuda:**

La sección de ayuda es necesaria, el componente la proporciona para dar información de algunas características que el usuario puede aprovechar, está ubicada en un área legible cumpliendo con estándares.

**Accesibilidad:**

El tamaño de la fuente y demás elementos fueron usados correctamente, contrastando con el color de fondo. El componente funciono adecuadamente con los navegadores en los que los evaluadores cargaron el componente.

**Control y realimentación:**

El usuario tiene control sobre la interfaz, teniendo libertad para actuar con un adecuado tiempo de respuesta.

**CONCLUSIONES DE LA EVALUACIÓN:**

En la evaluación no se tuvo en cuenta el manejo de errores, fue una observación que realizaron los evaluadores, además teniendo en cuenta sugerencias verbales sobre el manejo de los colores se han hecho cambios pertinentes para suavizarlos, de forma que sean más agradable a la vista; la ayuda que se le ofrece al usuario debe ser más clara y amplia, con una interfaz más agradable.

Refiere un evaluador experto que se debe dar prioridad al manejo de las horas, y por supuesto mejorar la apariencia de los iconos para que sean más llamativos y resalten cada combinación.

## 7. CONCLUSIONES

El Desarrollo de Software Basado en Componentes es una metodología que no se puede tomar más como una metodología de desarrollo nueva, por el contrario se trata de una idea que está evolucionando desde sus inicios para tomar más fuerza en el mundo de la ingeniería de software, existen ya grandes ideas, conceptos y un mundo de analistas, diseñadores que desean hacerla realidad y lo están haciendo.

El trabajo de investigación nos llevó al conocimiento de nuevas tecnologías y estudios que se están realizando en diferentes partes del mundo, mostrándonos a su vez que el ser humano está interesado en buscar nuevas alternativas para manejar la cantidad de datos existentes en el mundo, por ello, este proyecto nos llamó atención porque con él se aprendieron nuevos conceptos.

Se tuvo la oportunidad de interactuar con personas expertas en diferentes materias (investigadores de SeCo), quienes con su conocimiento evocaron a que la tecnología entre más sencilla se muestre más puede enriquecer sus servicios.

La utilización de *paper prototype* como técnica central en el desarrollo centrado en el usuario para generar ideas de diseño, permite el desarrollo de prototipos de bajo costo y rápida realimentación de los usuarios.

El estudio de la visualización de la información está relacionado con Human-Computer Interaction dado que en el problema está directamente implicado el usuario pues influyen factores como la cognición y la percepción.

Se ha desarrollado de un *widget front-end* en la arquitectura de SeCo ha aportado un componente de visualización para la representación de datos orientados o conectados en el tiempo.

En todos los pasos y resultados obtenidos y refinados durante la fase de diseño se tuvo en cuenta que la representación gráfica es el medio para comprender, ampliar conocimientos y tomar decisiones. Con base en esto, tratamos de crear una implementación interactiva donde se alcance el verdadero significado de un *widget*: presentación de datos + control.

## BIBLIOGRAFÍA

- [1] <http://www.search-computing.it/>
- [2] A. Bozzon, D. Braga, M. Brambilla, S. Ceri, F. Corcoglioniti, P. Fraternali, S. Vadacca, Search Computing: Multi-domain Search on Ranked Data, Politecnico di Milano, Milan, Italy.
- [3] Alessandro Bozzon, Marco Brambilla, Stefano Ceri, Piero Fraternali, *Liquid Query: Multi-Domain Exploratory Search on the Web*, WWW2010 - 19th International World Wide Web Conference - Raleigh, North Carolina, Monday, 26 April 2010.
- [4] A metasearch engine is a search tool, Sandy Berger's Great Age Guide to the Internet By Sandy Berger. Que Publishing, 2005, ISBN 0789734427
- [5] D. Norman, User Centered System Design: New Perspectives on Human-Computer Interaction (1986) (editor en colaboración con Stephen Draper).
- [6] [http://en.wikipedia.org/wiki/Metasearch\\_engine](http://en.wikipedia.org/wiki/Metasearch_engine)
- [7] Stefano Ceri, Marco Brambilla. Search Computing Challenges and Directions, Chapter 1: Search Computing, Springer Verlag in the LNCS State-of-art series, Vol. 5950, March 2010.
- [9] A. Bozzon, M. Brambilla, T. Catarci, S. Ceri, P. Fraternali, M. Matera, Visualization of Multi-Domain Ranked Data. Politecnico di Milano, Milano. Università "La Sapienza", Roma.
- [10] Simona Bonardi, Data Visualization Techniques for Multi-Domain Result Sets, Facoltà di Ingegneria dell'Informazione, Politecnico di Milano, 2009 – 2010.
- [11] Stuart K. Card, Jock D. Mackinlay, Ben Shneiderman, Readings in Information Visualization, Using Vision to Think, Morgan Kaufmann, 1999.
- [12] Yusef Hassan & Francisco J. Martín Fernández & Ghzala Iazza. Diseño Web Centrado en el Usuario: Usabilidad y Arquitectura de la Información [en línea]. "Hipertext.net", núm. 2, 2004. <http://www.hipertext.net>
- [13] <http://www.flickr.com/photos/unicauca/sets/72157610351013952/>

- [14] <http://www.w3c.org>
- [15] <http://www.ajaxya.com.ar/temarios/descripcion.php?cod=8&punto=1>
- [16] [http://eprints.rclis.org/bitstream/10760/8705/1/texto\\_099b.pdf](http://eprints.rclis.org/bitstream/10760/8705/1/texto_099b.pdf)
- [17] Juan C. Dürsteler, Revista InfoVis, [en línea] <http://www.infovis.net/>, Julio de 2000. Mensaje 180: Visualizando el Tiempo, 2006.
- [18] <http://www.emn.fr/z-info/spiraclock/index.html>
- [19] <http://www.aec.at/annualreport/>
- [20] <http://www.infovis.net/printMag.php?num=71&lang=1>
- [21] A. Bozzon, M. Brambilla, S. Ceri, P. Fraternali, I. Manolescu, Search Computing Challenges and Directions, Chapter 13, Liquid Queries and Liquid Results, Search Computing, Springer Verlag in the LNCS, March 2010.
- [22] Nota: *Usuario experto* en algunos contextos de SeCo, puede ser sinónimo de desarrollador.
- [23] [http://es.wikipedia.org/wiki/Rich\\_Internet\\_Applications](http://es.wikipedia.org/wiki/Rich_Internet_Applications)
- [24] Librería jQuery, <http://www.jquery.com>
- [25] <http://es.wikipedia.org/wiki/JQuery>
- [26] ACM SIGCHI Curricula for Human-Computer Interaction. Definition of HCI. Hewett, Baecker, Card, Carey, Gasen, Mantei, Perlman, Strong, Verplank, [http://old.sigchi.org/cdg/cdg2.html#2\\_1](http://old.sigchi.org/cdg/cdg2.html#2_1)
- [27] [http://internetng.dit.upm.es/wp-content/uploads/File/3\\_Entorno%20colab\\_centros\\_secund.pdf](http://internetng.dit.upm.es/wp-content/uploads/File/3_Entorno%20colab_centros_secund.pdf)
- [28] Google Chrome Experiments, <http://www.chromeexperiments.com/>
- [29] MORVILLE, P; ROSENFELD, L.; Information Architecture for the WWW (2002). 2a. ed. Cambridge (Massachusetts). O'Reilly. Citado en: <http://www.uoc.edu/web/esp/art/uoc/cobarsi0103/cobarsi0103.html>
- [30] <http://www.hipertext.net/web/pag206.htm>
- [31] Marti Hearst, Search User Interfaces, Cambridge University Press, 2009, [searchuserinterfaces.com](http://searchuserinterfaces.com).
- [32] "First Principles of Interaction Design", de Bruce Tognazzini, <http://www.galinus.com/es/articulos/principios-diseno-de-interaccion.html>.

- [33] <http://www.4shared.com/video/gK6QLCK-/CVTimeLine-Storyborad-SeCoUIS.html?>
- [34] Tabla con datos para realizar simulación del mockup  
[http://www.4shared.com/file/laGMu-5y/Datos\\_prueba\\_Mockup\\_CVTL.html?](http://www.4shared.com/file/laGMu-5y/Datos_prueba_Mockup_CVTL.html?)
- [35] Página oficial de desarrollo de TimeGlider <http://timeglider.com/jquery/>
- [36] Pagina de la W3C para HTML5 <http://www.w3.org/TR/html5/>
- [37] Por Jakob Nielsen, 1994.  
[http://www.useit.com/papers/heuristic/heuristic\\_list.html](http://www.useit.com/papers/heuristic/heuristic_list.html)
- [38] Snyder Carolyne, Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces. Morgan Kaufmann, 2003. [www.paperprototyping.com](http://www.paperprototyping.com)
- [39] I. Sommerville. Software engineering. Addison-Wesley Pub Co, 6ta edición, Agosto 2000.
- [40] J. Sametinger. Software engineering with reusable components. Springer Verlag, Agosto 1997.
- [41] J. Sodhi, y P. Sodhi. Software reuse: Domain analysis and design process. McGraw-Hill. 1999
- [42] Lidia Fuentes, José M. Troya, Antonio Vallecillo “Desarrollo de Software Basado en Componentes”, p 3-4. 2003.
- [43] Philippe Krutchen. “Rational Software”
- [44] Gartner Group.
- [45] Clemens Szyperski, “Component Software Component Software Beyond Object – Oriented Programming”. P 20 -22. 1998
- [46] Documentación librería BubbleUp, <http://aext.net/2010/04/bubbleup-jquery-plugin/>.