CAMPUS
DE EXCELENCIA
INTERNACIONAL

Graduado en Ingeniería Informática

Universidad Politécnica de Madrid

Escuela Técnica Superior de Ingenieros Informáticos

TRABAJO FIN DE GRADO

Editor web de conexiones para la creación visual de aplicaciones composicionales mediante mashup

Autor: Jaime Pajuelo Chávez

Director: Francisco Javier Soriano Camino

MADRID, ENERO DE 2015

ÍNDICE DEL DOCUMENTO

TABLA DE ILUSTRACIONES

RESUMEN

A lo largo de este documento se describe el trabajo llevado a cabo para el logro de todos objetivos de este Trabajo Fin de Grado, el cual tiene como objetivo principal la mejora de la herramienta de edición de las conexiones internas de un mashup proporcionada actualmente por la plataforma web WireCloud.

WireCloud es una plataforma web centrada en la construcción visual de mashups de aplicaciones a partir de la interconexión de pequeñas aplicaciones web denominadas widgets. Los principales inconvenientes presentes en el actual editor de conexiones que incluye esta plataforma afectan principalmente a sus usuarios con poca experiencia en diseño web. Estos usuarios tienen dificultades a la hora de interpretar el esquema de conexiones de un mashup ajeno y también, de crear el esquema de conexiones de un mashup propio.

La mejora realizada supone un cambio en la metáfora utilizada para la creación de las conexiones, que ahora se organiza en torno a unidades conceptuales denominadas *comportamientos* que representan subconjuntos cohesionados de conexiones con significado (representando por si mismos comportamientos relevantes del mashup). Con este cambio se logra solventar los inconvenientes que presenta el actual sistema, principalmente la necesidad de crear (y visualizar) simultáneamente todas las conexiones requeridas por el mashup, lo cual supone que:

a) es difícil identificar con qué propósito se ha creado cada conexión y qué relación guardan unas conexiones con otras.
b) existe un riesgo de olvidar alguna conexión, fundamentalmente por ser difícil la interpretación del propósito de cada conexión y por la imposibilidad de identificar y nombrar conjuntos de conexiones que tienen un propósito determinado.

Antes de implementar el código fuente se realizó un estudio pormenorizado de las tecnologías que actualmente utiliza WireCloud, además de un estudio en profundidad de la situación del anterior editor y de las nuevas características y ventajas buscadas en este nuevo editor. Con este último propósito se definieron varios casos de estudio que ayudaron a concretar qué se ha de entender por un *comportamiento* en el diseño de las conexiones de un mashup y también, ayudaron a definir la mejor organización visual del nuevo editor en torno al concepto de *comportamiento*. El resto del trabajo consistió en la implementación del nuevo editor y en la elaboración de toda la documentación relacionada: principalmente el manual de uso del nuevo editor que estará disponible como parte de la documentación online de WireCloud.

This document describes the work carried out for the achievement of all targets of this Final Project, which has as its main objective the improvement of the edition tool of the mashup's internal connections that is currently provided by WireCloud.

WireCloud is a web platform focused on building visual of web mashups from the interconnection of web applications called web widgets. The main drawbacks present in the current web editor of connections, includes on this platform, mainly affect to users with little experience in web design. These users have difficulties for interpreting the wiring diagram of any web mashup and also, creating the wiring diagram of an own web mashup.

The improvement made a change in the metaphor used for creating connections, now managing a conceptual units called behaviors that represent subsets of meaningful connections. This change overcomes the drawbacks of the current system, mainly the need to create and display simultaneously all connections required by the web mashup, which means that:

a) Difficultly identify what purpose is created each connection and how they relate to each other connections.
b) There is the risk of forgetting some connection, mainly for being difficult to interpret the purpose of each connection and the inability to identify and name sets of connections that have a specific purpose.

Before deploying the source code, the study of the technologies currently used WireCloud was performed, plus the study of the situation of the previous editor and new features and advantages searched for this new editor was performed too. For the latter purpose was defined several case studies that helped to specify what understood by behavior in the design of the connections of a web mashup and also that helped to define the best visual organization of the new behavior-oriented wiring editor.

The rest of the work involved in implementing the new wiring web editor and in the preparation of all documentation related: mainly user manual for using the new wiring web editor that will be available as part of the WireCloud online documentation.

CAPÍTULO 1

INTRODUCCIÓN Y OBJETIVOS

En Internet, el concepto de una aplicación web 2.0 [1] fue impulsado en el año 2004 permitiendo a un usuario final convertirse en un contribuidor más; en otras palabras, pasar de ser un simple observador de contenidos web a ser un creador capaz de participar e interactuar dentro de una comunidad virtual.

Dentro de este ámbito, el auge de las aplicaciones web de tipo mashup ha revolucionado la forma de diseñar una aplicación web. Un mashup [2] [3] brinda a un usuario final la oportunidad de combinar el contenido de más de una fuente, con la finalidad de crear un nuevo contenido completo e incluso, en algunos casos, innovador de una manera fácil y relativamente rápida.

Con el paso del tiempo han surgido plataformas centradas en el diseño de este tipo de aplicación web, entre las cuales se encuentra WireCloud [4]. Esta plataforma destaca por múltiples funcionalidades novedosas, sobre todo por dirigir un paradigma de desarrollo que en cada nueva publicación pretende acercarse más a usuarios finales. Actualmente, WireCloud incluye un editor web de conexiones que permite elaborar un mashup a través de la interconexión de diferentes aplicaciones web.

Gracias a la disponibilidad del código fuente de la plataforma WireCloud, liberado bajo la Licencia Publica General de Affero (GNU AGPLv3, del inglés GNU Affero General Public v3) y alojado en un repositorio remoto público, se permite el estudio y la modificación de los módulos relativos al editor de conexiones mencionado anteriormente, Dicho esto, el estudio de estos módulos se ha tomado como punto de partida y la modificación de estos hará posible el logro de los objetivos principales de este Trabajo Fin de Grado.

1.1 Motivaciones del trabajo

A día de hoy, pocas plataformas de diseño de mashups contienen la funcionalidad necesaria para que las aplicaciones web composicionales de un mashup interactúen entre sí; es decir, compartan cualquier tipo de información o recurso. Esta funcionalidad abre sin embargo un abanico de posibilidades en el diseño de este tipo de aplicación web, consiguiendo así que el límite se encuentre en las ideas innovadoras del diseñador.

La plataforma WireCloud proporciona dicha funcionalidad mediante un paradigma de composición de datos guiado por eventos. Por medio del cual, su estable editor web de conexiones permite a cualquier usuario visualizar todas las aplicaciones web que componen cierto mashup. Y además, ofrece la oportunidad de conectarlos entre sí de una manera sencilla e intuitiva, a través de los diferentes puntos de entrada y de salida que estos incluyen.

En la versión de desarrollo más reciente de esta plataforma, se puede observar un inconveniente, que concierne a usuarios con poca experiencia en diseño web, relativo a la dificultad para interpretar el esquema de conexiones de un mashup ajeno o a la dificultad para organizar las conexiones entre un gran número de aplicaciones web composicionales presentes en un mashup propio. A consecuencia de la observación citada, se pretende llevar a cabo la mejora del editor web de conexiones para así solventar estas dos situaciones, de modo que WireCloud incluya un paradigma de desarrollo más orientado a usuarios finales sin conocimientos de programación ni experiencia en diseño web.

1.2 Objetivos del trabajo

Este trabajo tiene como objetivo principal el desarrollo de un nuevo editor web de conexiones para la plataforma WireCloud, el cual sustituya a la versión actual, respetando los puntos fuertes como es su trabajado motor de conectividad, e incluya un diseño del esquema de conexiones orientado a comportamientos. Haciendo posible que cualquier usuario pueda visualizar más cómodamente y con más información un complejo esquema de conexiones e incluso, pueda organizar de forma más intuitiva un gran número de aplicaciones web composicionales de un mashup propio.

Un comportamiento, en el contexto de diseño web de un mashup, se refiere al grupo de aplicaciones web composicionales de un mashup, incluyendo ciertas conexiones existentes entre ellas, que persiguen una finalidad concreta que puede describirse mediante una frase (representando un comportamiento específico del mashup). En numerosas ocasiones, el comportamiento global de un mashup puede distribuirse en más de un comportamiento parcial.

Finalmente, el nuevo editor web tendrá un seguimiento continuo dentro de la plataforma WireCloud, de una forma no intrusiva, a través de pruebas de unidad y de integración; y tendrá en cuenta también, la posibilidad de instalación en otras plataformas de diseño de mashups.

1.3 Organización del resto del documento

El resto del documento se organiza del siguiente modo:

- En el capítulo 2 se mencionan las tecnologías de diseño y programación web que actualmente WireCloud hace uso además de un mejor análisis de esta plataforma web, explicando a parte las diferentes aplicaciones web que permite instalar y la plataforma de alojamiento donde se encuentra con acceso público la versión más reciente de WireCloud.
- Como consiguiente, el capítulo 3 detalla las fases de estudio y de implementación que se llevaron a cabo para lograr el desarrollo del nuevo editor web. También se incluye, la demostración del diseño de un mashup dada a partir de los comportamientos parciales identificados.
- Finalmente, el capítulo 4 describe los resultados del trabajo realizado y las principales conclusiones técnicas y personales extraídas de su desarrollo. Concluyendo, se ofrece algunas líneas futuras de desarrollo para su continuación.

CAPÍTULO 2

ESTADO DEL ARTE

Como punto de partida para la evolución de este trabajo, se ha realizado en primer lugar un estudio exhaustivo de las tecnologías de diseño y programación web que actualmente la plataforma WireCloud hace uso. Estas tecnologías son descritas a lo largo de este capítulo además de los diferentes tipos de aplicaciones web, que surgieron a partir del nacimiento del concepto Web 2.0, que la plataforma citada permite.

También se incluye una descripción más detallada de la plataforma en la se llevará a cabo la implementación del nuevo editor web de conexiones orientado a comportamientos, citando los puntos fuertes y la trabajaba herramienta de edición por los cuales fue elegida. En última instancia, y no menos importante, se menciona la plataforma de alojamiento de repositorios remotos donde se encuentra, hoy en día, alojada públicamente WireCloud. El buen uso de las funcionalidades que ofrece esta plataforma de alojamiento permitirá controlar la evolución del código fuente del nuevo editor web.

2.1 Tecnologías de Diseño y Programación Web

En primer lugar, se estudian las tecnologías de diseño y programación web que utiliza la plataforma WireCloud para su continua evolución. A día de hoy, estas tecnologías web son muy conocidas mundialmente y además están soportadas por cualquier navegador web actual, por esta razón se decidió continuar con la misma línea de desarrollo para lograr la correcta realización del código fuente del nuevo editor web.

2.1.1 Python

Figura 1: logo de Python

Python es un lenguaje de programación interpretado cuyo paradigma de desarrollo persigue una filosofía basada en la mejora de una sintaxis que favorezca un código fácilmente legible. Además de ser interpretado, lo que entre una multitud de posibilidades permite ejecutar un código fuente sin necesidad de compilarlo previamente, es un lenguaje de programación multiparadigma debido a que ofrece la posibilidad de

diferentes estilos de programación. Estos estilos de programación pueden ser: orientado a objetos, imperativo y, en menor medida, funcional.

Guido van Rossum, creador y principal autor de este lenguaje, publicó la primera versión estable, la 0.9.0, en 1991. A partir de esta fecha, Python fue haciéndose y ganándose un hueco entre los lenguajes más utilizados hoy en día. Actualmente, la Python Software Foundation (PSF) [6] administra todo el código, documentación y especificaciones añadidas desde la fecha del lanzamiento de la versión alfa de Python 2.1.

Esta organización sin ánimo de lucro fue fundada en el año 2001, posee una licencia de código abierto compatible con la Licencia Pública General de GNU [7] a partir de la versión de Python 2.1.1. Por último, en estos últimos años este lenguaje de programación se ha hecho muy popular gracias a varias razones:

- En primer lugar, incorpora una gran cantidad de bibliotecas que permiten realizar multitud de tareas, en su mayoría habituales, sin la necesidad de programarlas desde cero.
- En segundo lugar, es de destacar la sintaxis inmensamente visual basada en una notación indentada, que obliga a adaptarse a unas mismas notaciones con el objetivo de que todos los programas tengan un aspecto muy similar.
- Para acabar, la compatibilidad que ofrece con un gran número de plataformas.

Respecto al uso de este lenguaje en la plataforma WireCloud, se debe mencionar que la rama principal de todas las funcionalidades que esta ofrece desde el lado del servidor. Python, acompañado de la tecnología de diseño web que a continuación se menciona, hacen posible el despliegue del servicio web además de gestionar las diferentes pruebas que evalúan el correcto funcionamiento de esta.

2.1.2 Django

Figura 2: logo de Django

Django [8] es un entorno de desarrollo web, escrito en el lenguaje de programación Python, que fomenta el desarrollo rápido y el diseño limpio y pragmático; lo cual indica que permite construir aplicaciones o servicios web de una manera más rápida y utilizando un menos número de líneas de código. Actualmente, este entorno continúa mejorándose

casi a diario; y esta liberado bajo la licencia BSD [9], lo que significa que es gratuito y de código abierto.

Entre sus principales características, Django se adapta a la filosofía Modelo-Vista-Controlador (MVC) [10] que permite separar el proyecto en las diferentes partes que componen una aplicación web; y se adhiere al principio conocido como "Una vez y sólo una" (DRY, del inglés *Don't Repeat Yourself*) que promueve la reducción de la duplicación de cualquier pieza de información.

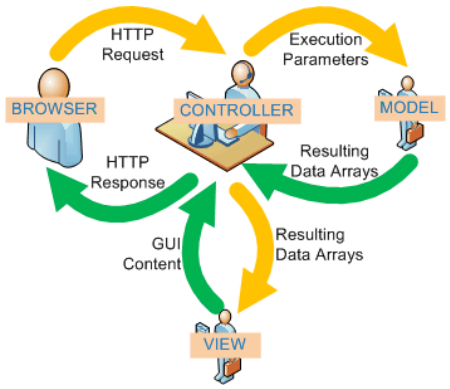

Figura 3: arquitectura Modelo-Vista-Controlador

2.1.3 HyperText Markup Language (HTML)

Figura 4: logo de HTML5

HTML [11] es un lenguaje de marcado utilizado para la elaboración de páginas web. Al ser también un estándar, este lenguaje tolera que una página web pueda ser interpretada por cualquier navegador web.

Este lenguaje se escribe en forma de etiquetas, rodeadas por corchetes angulares (<, >), que consisten en breves instrucciones de comienzo y final, mediante las cuales se determina la forma en la que debe aparecer, en el navegador, el texto y demás elementos.

La primera versión de HTML fue publicada en 1991 por Tim Berners-Lee, creador del World Wide Web [12] y director del World Wide Web Consortium (W3C) [13]. En esta publicación se describen 22 elementos que mostraban un diseño inicial y relativamente simple de HTML. A lo largo de la historia de Internet, han existido distintas versiones en

las cuales se han ido incorporando elementos nuevos o suprimiendo elementos que habían quedado obsoletos, con el fin de hacerlo más eficiente.

Para crear o editar un documento HTML puede utilizarse cualquier editor de texto plano. Existen otros editores con características WYSIWYG (del inglés *What You See Is What You Get*) que permiten ver el resultado de lo que se está editando en tiempo real. Algunos ejemplos de estos últimos son KompoZer, Microsoft FrontPage o Adobe Dreamweaver. Asimismo, existen otros editores conocidos como WYSIWYM (del inglés *What You See Is What You Mean*), que dan más importancia al contenido y al significado que a la apariencia visual. Entre los objetivos que tienen estos editores encontramos la separación que realizan entre el contenido y la presentación.

HTML5 [14] es la quinta revisión importante de este lenguaje. Esta versión establece una serie de nuevos elementos y atributos que reflejan el uso típico de los sitios web modernos. Alguno de los elementos de HTML 4.01 ha quedado obsoleto, incluyendo aquellos elementos puramente de presentación. También hay un renovado énfasis en la importancia del manejo del DOM [15] para el comportamiento de la Web. 2.0.

El W3C ha puesto su sello de recomendación a esta última versión, indicando con ello que nos encontramos ante un estándar ya finalizado. Esta denominación pone punto final a casi ocho años de trabajo continuado para sustituir al obsoleto HTML 4.01 que debutó como lenguaje básico de Internet en 1999.

2.1.4 JavaScript

Figura 5: logo JavaScript

JavaScript es un lenguaje de programación interpretado, creado por Netscape [16], que ofrece un paradigma de programación orientado a objetos basado en prototipos. A pesar de su nombre, JavaScript no guarda relación alguna con el lenguaje de programación Java.

En un principio, este lenguaje fue nombrado como LiveScript. Posteriormente, la unión entre Netscape y Sun Microsystems [17] (el creador del lenguaje de programación Java) supuso la modificación de su nombre por el que conocemos en la actualidad, este es JavaScript.

Hay que resaltar dos tipos de JavaScript: uno centrado en el lado del servidor y otro, por el contrario, en el lado del cliente. En este último, éste es muy popular respecto a otros debido a que cualquier navegador web soporta su uso sin necesidad de procesos intermedios. Hoy por hoy, JavaScript se utiliza principalmente para la creación de aplicaciones web dinámicas.

Para interactuar con una página web, JavaScript contiene el Modelo de Objetos del Documento (DOM, del inglés "Document Object Model") que permite ver el mismo documento HTML de otra manera, describiendo el contenido del documento como un conjunto de objetos accesibles y manipulables dinámicamente; y ofreciendo la capacidad de manipular los eventos que se producen tanto en el navegador web como en las acciones del usuario.

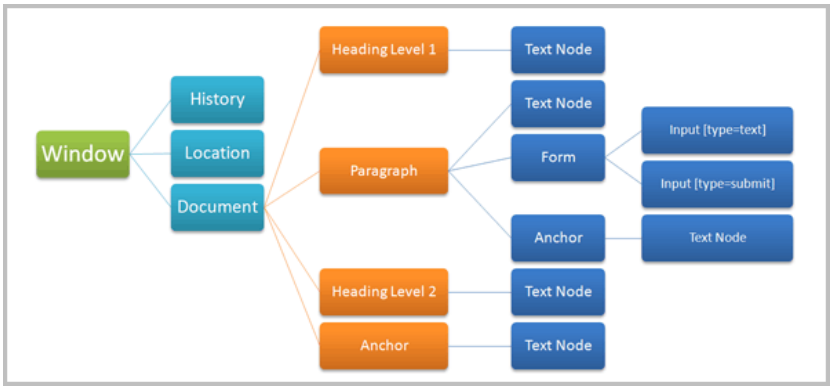

Figura 6: esquema simplificado de la jerarquía DOM

2.1.5 JavaScript Object Notation (JSON)

Figura 7: logo de JSON

JSON [18] es un formato ligero para el intercambio de datos. Básicamente, este estándar, es un texto plano basado en pares atributo-valor. Entre los tipos de valores que se puede encontrar en JSON podemos ver los siguientes: enteros, cadenas de caracteres, booleanos, listas y objetos.

Este formato nació como una disyuntiva a XML [19], pero su fácil uso en JavaScript ha generado un gran número de seguidores a esta alternativa. Una de las ventajas del uso de JSON, es que al ser un formato independiente a cualquier lenguaje de programación

admite que pueda ser usado para el intercambio de información entre distintas tecnologías.

2.1.6 Cascading Style Sheets (CSS)

Figura 8: logo de CSS3

CSS [20] es un lenguaje utilizado en la presentación de un documento estructurado, escrito en HTML o XML. Este lenguaje permite, a los programadores web, controlar el estilo y el aspecto de un documento facilitando el diseño y mantenimiento de múltiples páginas web.

La sintaxis de este lenguaje está basada en reglas; es decir, declaraciones sobre el estilo de uno o más elementos. Las hojas de estilo están compuestas por una o varias de esas reglas aplicadas a un documento. Estas reglas pueden dividirse en dos partes: un selector y una declaración. A su vez, esta última está compuesta por una propiedad junto con el valor que se le asigne. Por otro lado, el selector funciona como enlace entre el documento y el estilo, especificando los elementos que se van a ver afectados por esa declaración. La declaración es la parte de la regla que establece cuál será el efecto.

Tras la aparición de este lenguaje, surgió la necesidad de estandarizar su uso para todos los navegadores web. El organismo encargado de la estandarización fue W3C, la cual definió la primera versión en 1996. Posteriormente, se han desarrollado diferentes versiones hasta alcanzar CSS3 [21], la versión más actual.

A diferencia de CSS2, que fue una gran especificación que definía varias funcionalidades, su actual versión, mencionada anteriormente, está dividida en varios módulos. A día de hoy, en definición esta versión nos ofrece una gran variedad de opciones muy significativas para las necesidades del diseño web actual. Estas van desde opciones de sombreado y redondeado, hasta funciones avanzadas de movimiento y transformación, CSS3 es el estándar que dominará la web por los siguientes años.

2.1.7 Syntactically Awesome Style Sheets (SASS)

Figura 9: logo de SASS

Sass [22] es un metalenguaje de la parte superior de CSS que se utiliza para describir el estilo de un documento de manera limpia y estructural. Este proporciona una sintaxis más simple y elegante para CSS e implementa varias características útiles para la creación de hojas de estilo manejables.

Se pueden distinguir dos sintaxis: la primera, conocida como Sassy CSS (SCSS), es un superconjunto de las sintaxis en CSS3; y la segunda, conocida como una sintaxis de sangría, está diseñada para gente que prefiere similitud con CSS.

La implementación oficial de Sass es de código abierto y se encuentra escrita en Ruby [23]; sin embargo existen otras implementaciones, entre las que se incluye una en Python.

2.2 Aplicaciones Web

A lo largo de este punto se hará mención de las aplicaciones web que han sido estudiadas para un mejor entendimiento de los elementos con los que trabaja la plataforma en la cual se realiza este Trabajo Fin de Grado.

2.2.1 Widget

Un widget [24] es una pequeña aplicación web que puede ser instalado y ejecutado fácilmente por un usuario final, definiendo funcionalidades limitadas o nuevos contenidos. El origen de la palabra widget proviene de la combinación de las palabras window y gadget. Su objetivo principal es enriquecer los contenidos y las funcionalidades actuales del servicio web donde es aplicado, sin la necesidad de programar o crear nuevos contenidos.

Internamente, este tipo de aplicación puede interactuar con otros servicios distribuidos en Internet con el objetivo de ofrecer contenidos, como texto, imagen, audio o video, de casi cualquier temática y funcionalidad. La facilidad de diseñar estos contenidos se debe a que pueden desarrollarse en pocas líneas de código y utilizando lenguajes de programación tan conocidos como JavaScript o Adobe Flash [25].

En el contexto de la programación de aplicaciones visuales, los widgets adquieren otro significado más amplio como el de componente visual que forma parte de una Interfaz Gráfica de Usuario (GUI, del inglés “Graphical User Interface”) [26]. Otros términos utilizados para describir un widget son portlet, gadget, module, snippet y flake.

Figura 10: ejemplos de aplicaciones web widget

2.2.2 Mashup

Un mashup también es una aplicación web. En este caso, utiliza el contenido de más de una aplicación o recurso web para crear, de manera relativamente fácil, un nuevo contenido completo visualizado en una única interfaz gráfica. Por esta razón este tipo de aplicación es uno de los pilares de la Web 2.0, una web social y colaborativa donde los usuarios finales también pueden participar e interactuar entre sí.

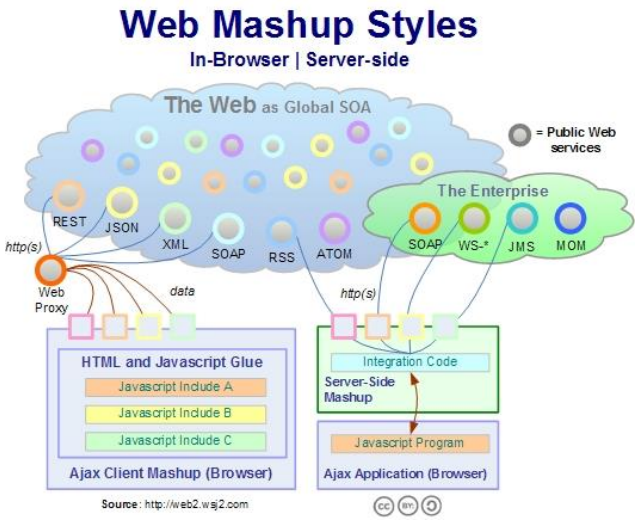

Figura 11: arquitectura cliente-servidor de un mashup

La potencia de un mashup radica en la facilidad de utilizar los recursos de las fuentes originales gracias a un API [27], sencillo y bien documentado, o una fuente RSS [28]; y en la imaginación de los usuarios llevada hasta límites inesperados. Entre estas fuentes se encuentran las APIs de Amazon, eBay, Flickr, Google, Microsoft, Yahoo o YouTube.

Según el objetivo que presenten pueden diferenciarse los siguientes tipos:

- Para consumidores, enfocados al usuario final.
- Para negocios, enfocados al cliente.
- Para datos, enfocados a la combinación de datos de dos o más fuentes.

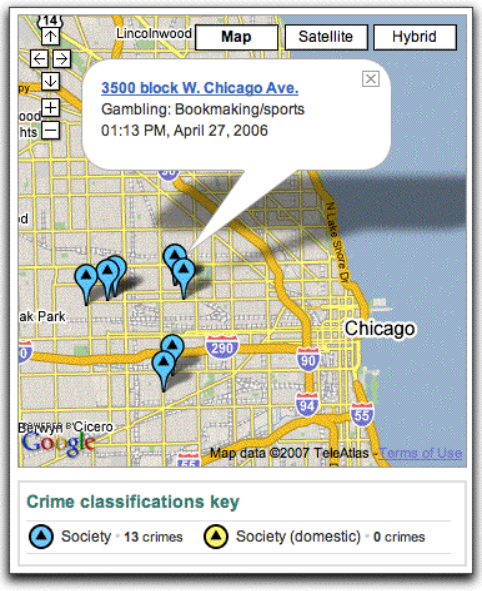

Figura 12: ejemplo de aplicación web mashup

2.3 Plataformas de Desarrollo de Mashups

En lo que concierne a las plataformas de desarrollo de mashups a continuación se dará una descripción detallada de la plataforma WireCloud, en la que desarrollará el nuevo editor web de conexiones.

2.3.1 WireCloud

Figura 13: logo de WireCloud

WireCloud es una plataforma web, soportada en Django, que ofrece a cualquier usuario, sin importar mucho el nivel técnico en diseño web, la posibilidad de construir un mashup web de manera rápida e intuitiva. Esta plataforma fue desarrollada en la Escuela Técnica Superior de Ingenieros Informáticos de la Universidad Politécnica de Madrid, más

concretamente por los miembros del laboratorio de "Computer Networks & Web Technologies" (CoNWeT) [29].

Un punto interesante respecto a esta plataforma web, es que forma parte de un proyecto europeo conocido como FI-WARE [30]. En este proyecto europeo, juega un papel importante entre los conocidos "Generic Enablers" donde hace referencia al mashup de aplicaciones web. Como consecuencia de esto, WireCloud implementa todas las APIs definidas en este Generic Enabler y cuya documentación se encuentra en la página oficial de FI-WARE.

Los servicios más destacados que ofrece esta plataforma, permiten al usuario lo siguiente:

- Subir un componente web compatible, como un widget, operador o mashup, para utilizarlo posteriormente en la creación de un mashup o compartirlo con el resto de usuarios.
- Añadir un componente web subido a un espacio de trabajo donde se diseña un nuevo mashup. En el caso de un mashup subido, crea un espacio de trabajo a partir de este.
- Descargar un componente web desde una tienda conectada, en la cual se comparten diferentes tipos de componentes web por parte de todos los usuarios.
- Modificar la conectividad entre los diferentes componentes web que compone un entorno de trabajo.

El editor del mapa de conexiones que ofrece WireCloud es la base fundamental de esta plataforma, el cual se diferencia con otros muchos editores. Gracias a su complejo motor de conexiones, un usuario puede editar o crear conexiones entre componentes web de un mashup de una manera fácil y rápida.

2.4 Plataformas de Alojamiento

Para concluir, en el subapartado que sigue, se dará una explicación de las ventajas que ofrece el uso de la plataforma GitHub [31] para llevar a cabo un proyecto como es el de WireCloud. Es por estas ventajas, aun no descritas, que se hará uso de esta plataforma para controlar la evolución del código fuente del proyecto.

2.4.1 GitHub

Figura 14: logo de GitHub

GitHub es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git [32]. El código se almacena de forma pública, aunque también se ofrece la opción de privatizar creando una cuenta de pago.

Esta plataforma permite alojar repositorios de código y brindar herramientas muy útiles para el trabajo en equipo, dentro de un proyecto. Además de eso, un usuario puede contribuir a mejorar el software de los demás. Para poder alcanzar esta meta, GitHub provee de funcionalidades para hacer un fork y solicitar pulls.

Realizar un fork es simplemente clonar un repositorio ajeno (genera una copia en tu cuenta), para eliminar algún bug o modificar cosas de él. Una vez realizadas tus modificaciones puedes enviar un pull al dueño del proyecto, el cual podrá analizar los cambios que has realizado fácilmente, y si considera interesante tu contribución, adjuntarlo con el repositorio original.

El sistema de control de versiones que utiliza, como se ha mencionado previamente, es Git. Git es un software de control de versiones, diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente. Se basa en un sistema distribuido de control de código fuente (SCM, del inglés “Source Code Management”), una herramienta que resuelve una serie de problemas que puedan surgir cuando se trabaja con código fuente.

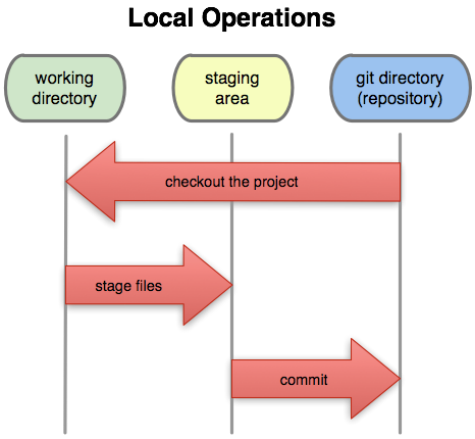

Figura 15: funcionamiento básico de un repositorio local

Este sistema aporta auditoria del código (conocimiento de quién ha modificado qué y cuándo), control sobre cómo ha cambiado el proyecto con el paso del tiempo, vuelta a versiones anteriores de una forma rápida, capacidad de trabajo en equipo, control y seguridad en las actualizaciones realizadas en las estructuras internas del código, y mezcla y creación de ramas extremadamente eficientes.

CAPÍTULO 3

DESARROLLO

A lo largo de este capítulo, se describe el desarrollo y la implementación del nuevo editor web de conexiones orientado a comportamientos para la plataforma WireCloud. Como se ha mencionado con anterioridad, esta plataforma de diseño de aplicaciones web de tipo mashup incluye un editor web de conexiones cuyo paradigma de desarrollo necesita ser actualizado con el propósito de dirigirse cada vez más a usuarios finales. Y el alcance de este propósito se logrará resolviendo los objetivos principales de este trabajo.

Tras llevar a cabo un análisis profundo de la situación actual del editor web de conexiones que facilita esta plataforma, y detallar las nuevas características que el nuevo diseño debe tener en cuenta, se dispuso la implementación de un nuevo sistema prácticamente desde cero. Además, este nuevo sistema debe cumplir también con las necesidades actuales, satisfechas por el actual editor web de conexiones, de la plataforma WireCloud; e incluso, tener en mente la posibilidad de instalación en otras plataformas similares. A continuación, se citan los sistemas que se han mantenido inmunes a posibles modificaciones debido a su trabajado código:

- Sistema de arrastrado: permite modificar la posición actual de cualquier aplicación web dentro de los límites establecidos por el sistema.
- Motor de conectividad: permite crear una conexión entre diferentes aplicaciones web, indicando el punto de entrada de una y el punto de salida de otra.
- Sistema de recomendaciones: permite conocer las posibles conexiones entre diferentes aplicaciones web a partir de las descripciones dadas en los puntos de entrada y de salida de estas.

Relativo a la versión de desarrollo más reciente de esta plataforma, el nuevo editor web de conexiones soluciona los inconvenientes respectivos a la visualización del esquema de conexiones inentendible de un mashup ajeno o a la organización de un gran número de aplicaciones web composicionales de un mashup propio. Haciendo posible que cualquier usuario pueda realizar lo anterior de una manera rápida e intuitiva, y conseguir de este modo, que WireCloud incluya un paradigma de desarrollo más orientado a usuarios finales.

Profundizando en el funcionamiento del nuevo sistema, este debe permitir a cualquier usuario visualizar todas las aplicaciones web que componen cierto mashup; además de, ofrecer la oportunidad de conectarlas entre sí de una manera sencilla e intuitiva, a través de los diferentes puntos de entrada y de salida que estas incluyen. Todo esto, por medio de un sistema orientado a comportamientos.

Un comportamiento, en el contexto de diseño web de un mashup, se refiere al grupo de aplicaciones web composicionales de un mashup, incluyendo ciertas conexiones existentes entre ellas, que persiguen una finalidad común. Incluyendo la posibilidad de que el comportamiento global de un mashup pueda distribuirse en uno o más comportamientos parciales, los cuales puedan visualizarse independientemente y puedan compartir cualquier aplicación web o conexión que participen en otros comportamientos.

Después de esta aclaración, un editor web de conexiones orientado a comportamientos permite gestionar los diferentes comportamientos parciales que un usuario considere dentro de cierto mashup, y visualizarlos de forma independiente o respecto al comportamiento global. Así como, la fácil ubicación de todas las aplicaciones web, disponibles en la cuenta del usuario, con el objetivo de utilizarlos posteriormente en el esquema de conexiones.

Por último, se mantiene el estilo y la logística de la plataforma WireCloud utilizando JavaScript, en la implementación del código fuente; Sass, en el diseño web; y Python tanto para las pruebas de unidad y de integración, como para el código de migración de datos a este nuevo sistema.

3.1 Fase de Análisis

En este apartado se describe la fase de análisis, comenzando con una explicación detallada del estado actual de la plataforma WireCloud y una identificación de los actores principales del sistema y los posibles casos de uso. Para concluir, se mencionan los requisitos específicos del nuevo sistema que satisfacen los objetivos principales de este trabajo.

3.1.1 Estado Actual de la Plataforma WireCloud

WireCloud es una plataforma web centrada en el diseño de aplicaciones web de tipo mashup. Está desarrollada en la Escuela Técnica Superior de Ingenieros Informáticos de la UPM, específicamente en el laboratorio CoNWeT. Entre las herramientas que ofrece para la edición de un mashup, esta plataforma proporciona un editor web de conexiones

que permite interconectar diferentes aplicaciones web de tipo widget u operador a través de los puntos de entrada y de salida que estas disponen.

También ofrece la posibilidad de conectarse a una tienda web con la finalidad de proporcionar a los usuarios una forma de publicar cualquier entorno de trabajo como un mashup y de instalar cualquier aplicación web (widget, operador o mashup) aportada por algún diseñador registrado. En otras palabras, los usuarios pueden compartir sus aplicaciones web a través de este sistema de publicaciones.

Figura 16: la tienda web de WireCloud en FI-WARE

La pantalla principal de esta plataforma permite a los usuarios gestionar sus propios entornos de trabajo, los cuales pueden dividirse en pestañas. Cada entorno de trabajo contiene los widgets, que el usuario ha añadido de una forma fácil e intuitiva, visualizados por su interfaz gráfica. Estos widgets se pueden redimensionar, modificar su posición en el entorno de trabajo, configurar según las preferencias que el diseñador ha proporcionado y finalmente interactuar con la interfaz gráfica que dispone. También, el usuario puede distribuir los widgets de un entorno de trabajo en las diferentes pestañas de las que disponga.

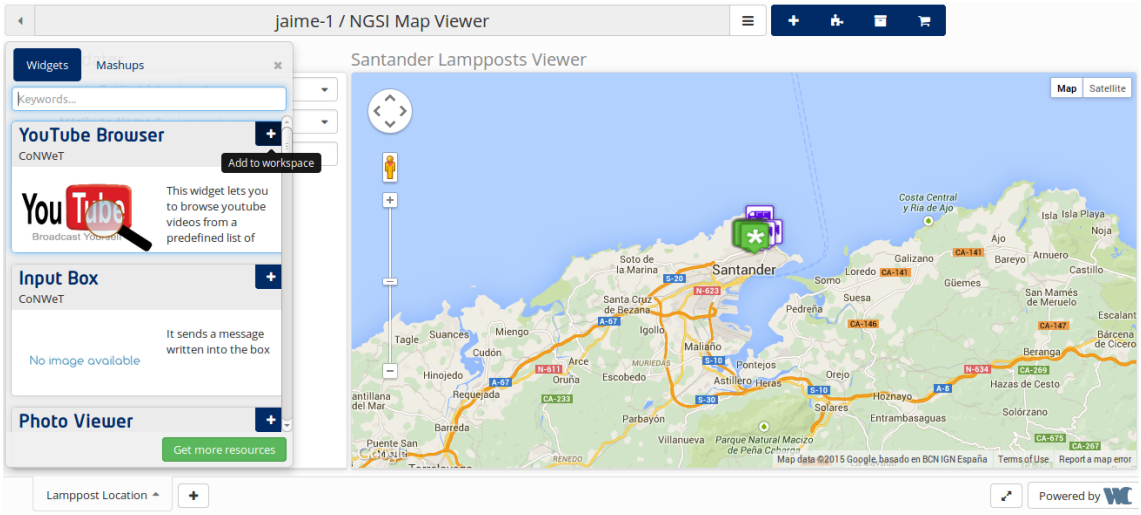

Figura 17: el menú deslizante de aplicaciones en un entorno de desarrollo

Los widgets dentro de WireCloud se pueden ver como pequeñas aplicaciones web que disponen de una interfaz gráfica, con la cual un usuario pueda interactuar, y de un conjunto de puntos de entrada y de salida que permiten recibir y enviar información. La funcionalidad de un widget está definida por el diseñador. Y la utilidad principal de estas aplicaciones web en esta plataforma, es la de conectarse con otras aplicaciones web de tipo widget u operador de modo que tengan un papel importante dentro de un mashup.

Volviendo al editor web de conexiones del que dispone ahora WireCloud, se debe destacar la importancia que los operadores están teniendo a la hora de conectar aplicaciones web que intercambian información casi compatible. En esta plataforma, los operadores pueden verse como pequeñas aplicaciones web que no disponen de una interfaz gráfica pero sí disponen de un conjunto de puntos de entrada y de salida. Desde la parte visual de un mashup, estos operadores no serían visibles al carecer de interfaz gráfica. Aun así, la utilidad principal de estos reside en el esquema de conexiones de un mashup, donde puede proporcionar datos de servicios externos o modificar la estructura o los datos que pasan entre dos aplicaciones web.

Como ya se había adelantado en apartados anteriores, el editor web de conexiones es una herramienta fundamental a la hora de diseñar un mashup dentro de la plataforma WireCloud. Pero este editor web necesita ser actualizado para que el objetivo principal de la plataforma, de orientarse cada vez más a usuarios finales, sea posible. En esta actualización se desea que esta herramienta web esté orientada a comportamientos de tal modo que se solventen los inconvenientes mencionados anteriormente. A continuación,

se detalla un análisis profundo de los puntos fuertes y débiles que se observan en el actual editor web.

El editor web de conexiones que integra actualmente WireCloud permite al usuario lo siguiente:

1. Disponer de las aplicaciones web accesibles en el entorno de trabajo actual para utilizarlas en el área de edición del esquema de conexiones. Estas se encuentran listadas en el lateral izquierdo de la pantalla y pueden ser arrastradas con el cursor a la posición que el usuario desee dentro de los límites del área mencionada antes.
2. Crear, modificar y borrar conexiones entre las diferentes instancias disponibles en el área de edición del esquema de conexiones de una manera rápida e intuitiva.
3. Reconocer fácilmente el contenido de la instancia de un widget u operador; es decir, el título y los puntos de entrada y de salida. Además de, diferenciar la instancia de un widget de la de un operador.
4. Recomendar posibles conexiones a partir de las descripciones dadas por los desarrolladores en los diferentes puntos de entrada y de salida de los que disponen cada instancia de aplicaciones web utilizadas en el área de edición del esquema de conexiones.
5. Minimizar un operador y reordenar los puntos de entrada y de salida de una instancia cualquiera con el objetivo de visualizar de mejor manera el esquema de conexiones.

En cuanto a los puntos débiles, este editor web de conexiones tiene inconvenientes que solamente afectan a usuarios con poca experiencia en diseño de mashups. Estos inconvenientes pueden reflejarse en dos situaciones: a la hora de conocer el esquema de conexiones complejo por el número de componentes de un mashup y a la hora de construir desde cero un mashup propio que poco a poco empieza a contener un gran número de aplicaciones web y conexiones.

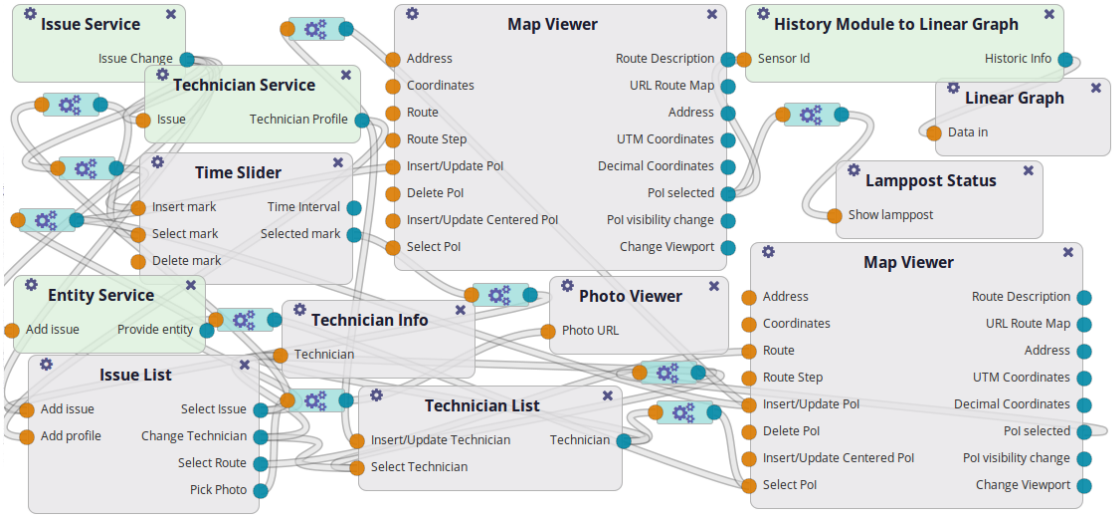

Figura 18: esquema de conexiones de un mashup complejo

En la ilustración de arriba pueden contemplarse las dos situaciones que este trabajo intenta solventar. Desde el punto de vista de un usuario ajeno, con un nivel técnico medio-bajo en diseño, puede ver este esquema de conexiones como inentendible. Y desde el punto de vista del propietario, la continuación o edición de este esquema de conexiones puede ser de difícil ejecución.

Después de explicar el estado actual del editor web de conexiones, se enumeran los puntos a mejorar respecto a este estado:

1. La interfaz general debe modificarse para gestionar una mejor barra de herramientas, de esta manera permitir añadir nuevas funcionalidades en este trabajo e incluso más adelante.
2. La nueva barra de herramientas debe ofrecer la opción de listar tanto los widgets y operadores accesibles en ese momento, como los comportamientos parciales que se vayan identificando.
3. La nueva barra de herramientas debe ofrecer, de una manera fácil y rápida, cambiar de perspectiva; es decir, de la vista global del esquema de conexiones frente a un comportamiento parcial a la vista independiente de un comportamiento parcial.
4. La nueva barra de herramientas debe ocupar el menor espacio posible con el fin de aprovechar el espacio de la pantalla en el área de edición del esquema de conexiones.

5. El área de edición del esquema de conexiones debe mejorarse para contrastar los componentes de un comportamiento parcial y de ese modo, tener visibles el resto de componentes del mashup como si estuvieran en segundo plano.

3.1.2 Identificación de los Actores Principales

En esta sección, se identifican los actores principales del nuevo sistema. Como el objetivo principal del paradigma de desarrollo, que incluye la plataforma WireCloud, que se desea mejorar para dirigirse más usuarios finales, estos son los únicos actores principales que se pueden encontrar. Entonces, se incluyen tanto usuarios con un nivel técnico avanzado en diseño web, como usuarios con poca experiencia en ello.

Teniendo identificado al actor principal, este tiene como objetivo principal la construcción de un mashup a partir de la interconexión de diferentes aplicaciones web y la descripción de cada comportamiento parcial identificado en el esquema de conexiones final, con el propósito de que otros usuarios o el mismo puedan comprender el comportamiento global de dicho mashup de manera rápida y simple, sin importar la complejidad del esquema de conexiones o el número elevado de aplicaciones web usadas.

Otros objetivos y no menos importantes que los actores principales pueden perseguir, se enumeran a continuación:

1. Disponer, en todo momento, de las aplicaciones web disponibles en el entorno de trabajo actual para su futuro uso en el esquema de conexiones dado.
2. Distribuir, de manera fácil e intuitiva, las diferentes aplicaciones web que componen o compondrán el mashup en cuestión, además de las conexiones existentes, en los comportamientos parciales que el usuario ha identificado.
3. Visualizar el esquema de conexiones de cada comportamiento parcial creado de forma independiente o respecto al comportamiento global del mashup.
4. Modificar el esquema de conexiones de un comportamiento creado visualizando las aplicaciones web y conexiones existentes en otros comportamientos parciales.
5. Conectar diferentes aplicaciones web del esquema de conexiones, visualizando en todo momento las recomendaciones que el sistema ofrece como ayuda a partir de las descripciones dadas en los puntos de entrada o de salida de cada aplicación web.

3.1.3 Identificación de los Casos de Uso

En esta sección, se presentan los diferentes casos de uso identificados en este nuevo sistema. Esta identificación se ha realizado a partir del estudio del actual editor web de conexiones que proporciona WireCloud y el análisis de los requisitos iniciales especificados en este trabajo. Los siguientes casos de uso son el resultado del estudio y el análisis mencionado antes, los cuales fomentan una idea más concreta de los puntos a implementar e indican como el sistema debería interactuar con los actores principales.

3.1.3.1 Visualizar el Esquema de Conexiones

Precondiciones: el usuario debe estar registrado e identificado en la plataforma.

Objetivo: visualizar el esquema de conexiones del entorno de trabajo o mashup actual.

Procedimiento: el usuario selecciona de la barra de herramientas del entorno de trabajo actual, la opción de visualizar el esquema de conexiones. Esta acción muestra una nueva barra de herramientas y el esquema de conexiones del mashup en cuestión.

Postcondiciones: el sistema crea y activa un comportamiento parcial en el caso de que el entorno de trabajo o mashup actual no contenga ninguno.

3.1.3.2 Añadir un Widget desde la Barra de Herramientas

Precondiciones: el usuario debe estar registrado e identificado en la plataforma; además de, tener al menos un widget disponible en el entorno de trabajo actual y tener activa la vista del esquema de conexiones del comportamiento global del mashup.

Objetivo: utilizar un widget en el esquema de conexiones del comportamiento activo.

Procedimiento: el usuario despliega una lista de aplicaciones web de tipo widget disponibles en el entorno de trabajo actual, selecciona una de estas que se encuentre habilitada y la arrastra hasta la posición que desee, dentro del esquema de conexiones del comportamiento global del mashup.

Postcondiciones: el sistema solo permite una instancia por cada widget, lo cual indica que el widget seleccionado quedará inhabilitado.

3.1.3.3 Añadir un Operador desde la Barra de Herramientas

Precondiciones: el usuario debe estar registrado e identificado en la plataforma; además, tener al menos una aplicación web de tipo operador instalada en la cuenta actual y tener activa la vista del esquema de conexiones del comportamiento global del mashup.

Objetivo: utilizar un operador en el esquema de conexiones del comportamiento activo.

Procedimiento: el usuario despliega una lista de aplicaciones web de tipo operador instaladas en la cuenta actual, selecciona una de estas que se encuentre habilitada y la arrastras hasta la posición que desee, dentro del esquema de conexiones del comportamiento global del mashup.

Postcondiciones: el sistema permite crear más de una instancia por cada operador, lo cual indica que el operador seleccionado seguirá habilitado.

3.1.3.4 Identificar de los puntos de entrada o de salida

Precondiciones: el usuario debe estar registrado e identificado en la plataforma; además de, tener al menos una instancia de una aplicación web en el esquema de conexiones del comportamiento global del mashup.

Objetivo: distinguir los puntos de entrada y de salida de cualquier instancia de aplicación web en el esquema de conexiones del comportamiento global del mashup.

Procedimiento: cada instancia de aplicación web distribuye los puntos de entrada al lateral izquierdo y los puntos de salida al lateral derecho.

3.1.3.5 Distinguir una instancia

Precondiciones: el usuario debe estar registrado e identificado en la plataforma; además de, tener al menos una instancia de un widget y otra de operador en el esquema de conexiones del comportamiento global del mashup.

Objetivo: distinguir las instancias de las aplicaciones web de tipo widget y de tipo operador en el esquema de conexiones del comportamiento global del mashup.

Procedimiento: cualquier instancia de una aplicación web de tipo widget tiene un color diferente al otro tipo de aplicación web. En el caso de que estuvieran visibles en el fondo, dicho color simplemente será más transparente.

3.1.3.6 Crear una Conexión

Precondiciones: el usuario debe estar registrado e identificado en la plataforma; además de, tener al menos dos instancias de aplicaciones web, que incluyan puntos de entrada y de salida, disponibles en el esquema de conexiones del comportamiento global del entorno de trabajo actual o mashup y tener activa la vista del esquema de conexiones del comportamiento global del mashup.

Objetivo: crear una conexión entre el punto de salida y el punto de entrada de dos instancias diferentes de aplicaciones web en el comportamiento activo.

Procedimiento: el usuario selecciona el punto de salida de la instancia de una aplicación web y desplaza el cursor hasta el punto de entrada de otra instancia, o viceversas, creando de este modo una conexión en el comportamiento activo.

Postcondiciones: el sistema añade una o ambas instancias de aplicaciones web en el caso de que estuvieran visibles en el fondo. Por defecto, si la nueva conexión ya existía en el comportamiento activo, esta será eliminada ya que no se permite duplicidad en las conexiones.

3.1.3.7 Resaltar una Instancia

Precondiciones: el usuario debe estar registrado e identificado en la plataforma; además de, tener al menos una instancia de una aplicación web disponible en el esquema de conexiones del comportamiento global del entorno de trabajo actual.

Objetivo: destacar visualmente una instancia de una aplicación web respecto al resto de instancias del esquema de conexiones del comportamiento global del entorno de trabajo actual.

Procedimiento: el usuario selecciona una instancia de una aplicación web del esquema de conexiones.

Postcondiciones: el sistema resalta la instancia seleccionada, incluido las conexiones salientes y entrantes, con un color y sombreado diferente al resto. En caso que dicha instancia se encuentre en el fondo del esquema de conexiones, este resaltado será un poco transparente.

3.1.3.8 Resaltar una Conexión

Precondiciones: el usuario debe estar registrado e identificado en la plataforma; además de, tener al menos una conexión disponible en el esquema de conexiones del comportamiento global del entorno de trabajo actual.

Objetivo: destacar visualmente una conexión respecto al resto de conexiones del esquema de conexiones del comportamiento global del mashup.

Procedimiento: el usuario selecciona una conexión del esquema de conexiones del comportamiento global del mashup.

Postcondiciones: en el caso de que esta conexión no pertenezca al comportamiento activo, el resaltado será un poco transparente.

3.1.3.9 Modificar la Posición Actual de una Instancia

Precondiciones: el usuario debe estar registrado e identificado en la plataforma; además de, tener al menos una aplicación web disponible en el entorno de trabajo actual y tener al menos una instancia disponible en el esquema de conexiones del comportamiento global del mashup.

Objetivo: modificar la posición actual de una instancia de una aplicación web disponible en el esquema de conexiones del comportamiento global del mashup actual.

Procedimiento: el usuario selecciona y arrastra una instancia de una aplicación web a otra posición dentro del área del esquema de conexiones del comportamiento global del mashup.

3.1.3.10 Modificar la Posición Actual de más de una Instancia

Precondiciones: el usuario debe estar registrado e identificado en la plataforma; además de, tener más de una aplicación web disponible en el entorno de trabajo actual y tener más de una instancia en el esquema de conexiones del comportamiento global del mashup.

Objetivo: modificar la posición actual de más de una instancia en el esquema de conexiones del comportamiento global del mashup.

Procedimiento: el usuario selecciona y arrastra más de una instancia del esquema de conexiones del comportamiento global del mashup a otra posición dentro de dicha área.

3.1.3.11 Modificar el Orden de los Puntos de Entrada y de Salida

Precondiciones: el usuario debe estar registrado e identificado en la plataforma; además de, tener al menos una instancia en el esquema de conexiones del comportamiento activo.

Objetivo: modificar el orden de los puntos de entrada o de salida de una instancia de una aplicación web perteneciente al comportamiento activo.

Procedimiento: el usuario selecciona la opción de configuración de una instancia en la parte superior izquierda, esta acción despliega una lista de opciones, entre las cuales está la de reordenar los puntos de entrada o de salida de dicha instancia. Ahora estará disponible la modificación de la lista de puntos, lo cual el usuario puede seleccionar cualquier punto y desplazarlo a la posición que el desee.

3.1.3.12 Modificar la Curvatura de una Conexión

Precondiciones: el usuario debe estar registrado e identificado en la plataforma; además de, tener al menos una conexión disponible en el esquema de conexiones del comportamiento activo.

Objetivo: modificar la curvatura de una conexión del esquema de conexiones del comportamiento activo.

Procedimiento: el usuario selecciona una conexión perteneciente al comportamiento activo, tras esta acción, aparecen las opciones para modificar la curvatura tanto en el lado del punto de entrada como en el de salida, en forma de tiradores. Estos tiradores permiten modificar dicha curvatura como el usuario desee.

3.1.3.13 Modificar el Punto de Entrada o de Salida de una Conexión

Precondiciones: el usuario debe estar registrado e identificado en la plataforma; además de, tener al menos una conexión disponible en el esquema de conexiones del comportamiento activo.

Objetivo: modificar el punto de entrada o de salida de una conexión perteneciente al comportamiento activo.

Procedimiento: el usuario selecciona una conexión del esquema de conexiones perteneciente al comportamiento activo, habilitando la posibilidad de mover el punto de entrada o de salida a otro punto disponible.

3.1.3.14 Crear un Comportamiento

Precondiciones: el usuario debe estar registrado e identificado en la plataforma.

Objetivo: crear un comportamiento en el esquema de conexiones del comportamiento global del mashup.

Procedimiento: en la barra de herramientas, el usuario puede crear un nuevo comportamiento a partir de un formulario a rellenar con los campos de título y descripción. Estos campos son opcionales. En el formulario se le ofrece la posibilidad de crear un nuevo comportamiento, además de activarlo en la misma acción. Por defecto, este comportamiento se crea vacío.

3.1.3.15 Añadir un Widget desde el Esquema de Conexiones

Precondiciones: el usuario debe estar registrado e identificado en la plataforma; además de, tener al menos un widget disponible en el entorno de trabajo actual y tener activa la vista del esquema de conexiones del comportamiento global del mashup.

Objetivo: utilizar una instancia de un widget ya existente en el comportamiento global del mashup en el esquema de conexiones del comportamiento activo.

Procedimiento: el usuario selecciona la instancia visible en el fondo del esquema de conexiones, y selecciona la opción de añadir al comportamiento activo.

3.1.3.16 Añadir un Operador desde el Esquema de Conexiones

Precondiciones: el usuario debe estar registrado e identificado en la plataforma; es decir, tener al menos una instancia de una aplicación web de tipo operador disponible en el esquema de conexiones y tener activa la vista del esquema de conexiones del comportamiento global del entorno de trabajo actual.

Objetivo: utilizar una instancia de un operador ya existente en el comportamiento global del mashup en el esquema de conexiones del comportamiento activo.

Procedimiento: el usuario selecciona una instancia de una aplicación web de tipo operador visible en el fondo del esquema de conexiones, y selecciona la opción de añadir al comportamiento activo disponible en el parte superior derecha.

3.1.3.17 Añadir una Conexión desde el Esquema de Conexiones

Precondiciones: el usuario debe estar registrado e identificado en la plataforma; además de, tener al menos una conexión disponible en el esquema de conexiones y tener activa la vista del esquema de conexiones del comportamiento global del entorno de trabajo actual.

Objetivo: utilizar una conexión existente en el comportamiento global del mashup en el esquema de conexiones del comportamiento activo.

Procedimiento: el usuario selecciona una conexión visible en el fondo del esquema de conexiones, mostrando de este modo la opción de añadirla en el comportamiento activo.

Postcondiciones: el sistema la añade al comportamiento activo si uno o ambos extremos de esta conexión también estaban visibles en el fondo.

3.1.3.18 Modificar la Información Básica de un Comportamiento

Precondiciones: el usuario debe estar registrado e identificado en la plataforma.

Objetivo: modificar la información básica (título y descripción) de un comportamiento creado.

Procedimiento: desde la barra de herramientas, el usuario puede listar los comportamientos parciales existentes en el entorno de trabajo actual. Entonces, el usuario selecciona la opción de preferencias en la parte derecha superior de un comportamiento, abriendo de este modo un formulario con los datos actualmente guardados de dicho comportamiento parcial. El usuario modifica la información básica disponible y selecciona la opción de guardar los cambios producidos.

3.1.3.19 Borrar una Conexión

Precondiciones: el usuario debe estar registrado e identificado en la plataforma; además de, tener al menos una conexión disponible en el esquema de conexiones del comportamiento activo.

Objetivo: borrar una conexión existente en el esquema de conexiones del comportamiento activo.

Procedimiento: el usuario selecciona una conexión existente en el comportamiento activo, mostrando la opción de borrado. Entonces, el usuario selecciona la opción de borrado y borra dicha conexiones.

Postcondiciones: si dicha conexión pertenece a otro comportamiento parcial, se fijara en el fondo del esquema de conexiones; sino, se borrará también del comportamiento global del mashup.

3.1.3.20 Borrar la instancia de un Widget

Precondiciones: el usuario debe estar registrado e identificado en la plataforma; además de, tener al menos una instancia de un widget disponible en el esquema de conexiones del comportamiento activo.

Objetivo: borrar la instancia de un widget existente en el esquema de conexiones del comportamiento activo.

Procedimiento: el usuario selecciona la opción de borrado mostrada en la parte superior derecha del widget perteneciente al comportamiento activo, borrando además las conexiones entrantes y salientes.

Postcondiciones: si dicha instancia pertenece a otro comportamiento parcial, se fijara en el fondo del esquema de conexiones; sino, se borrará también del comportamiento global del mashup y se habilitará en el menú de herramientas.

3.1.3.21 Borrar la instancia de un Operador

Precondiciones: el usuario debe estar registrado e identificado en la plataforma; además de, tener al menos una instancia de un operador disponible en el esquema de conexiones del comportamiento activo.

Objetivo: borrar la instancia de un operador existente en el esquema de conexiones del comportamiento activo.

Procedimiento: el usuario selecciona la opción de borrado mostrada en la parte superior derecha del operador perteneciente al comportamiento activo, borrando también las conexiones entrantes y salientes.

Postcondiciones: si dicha instancia pertenece a otro comportamiento parcial, se fijará en el fondo del esquema de conexiones; sino, se borrará también del comportamiento global del mashup.

3.1.3.22 Activar la Vista Global

Precondiciones: el usuario debe estar registrado e identificado en la plataforma.

Objetivo: conocer el esquema de conexiones del comportamiento global del mashup.

Procedimiento: en la sección de herramientas, el usuario puede activar la vista del esquema de conexiones del comportamiento global. Esta acción, mostrará todas las instancias de aplicaciones web y las conexiones del esquema de conexiones del comportamiento global del mashup, resaltando las instancias y las conexiones pertenecientes al comportamiento activo y fijando al fondo el resto de componentes.

Postcondiciones: el sistema oculta todas las opciones accesibles en los componentes fijados al fondo.

3.1.3.23 Desactivar la Vista Global

Precondiciones: el usuario debe estar registrado e identificado en la plataforma.

Objetivo: conocer el esquema de conexiones del comportamiento activo.

Descripción: en la sección de herramientas, el usuario puede desactivar la vista del esquema de conexiones del comportamiento global del mashup. Esta acción, muestra una vista independiente del comportamiento activo; es decir, se muestran los componentes pertenecientes a este.

Postcondiciones: cualquier modificación realizada en esta vista, no afectará al resto de comportamientos parciales ni mucho menos al comportamiento global.

3.1.3.24 Borrar un Comportamiento

Precondiciones: el usuario debe estar registrado e identificado en la plataforma.

Objetivo: borrar un comportamiento parcial disponible en el esquema de conexiones del entorno de trabajo actual.

Procedimiento: en el formulario de edición de un comportamiento, el usuario puede observar la opción de borrado de este. Esta opción solo se muestra si no es el último comportamiento parcial de dicho esquema de conexiones. Al seleccionar esta opción, el comportamiento parcial específico será borrado del mashup.

Postcondiciones: el sistema borrará todos los componentes de dicho comportamiento.

3.1.3.25 Activar un Comportamiento

Precondiciones: el usuario debe estar registrado e identificado en la plataforma.

Objetivo: mostrar el esquema de conexiones de un comportamiento parcial.

Procedimiento: el usuario selecciona el comportamiento que desea visualizar desde el panel de comportamientos parciales. Esta acción resalta los componentes del esquema de conexiones del comportamiento seleccionado por encima del resto de componentes del comportamiento global del mashup.

3.1.4 Requisitos Específicos

En esta sección, se presentan los requisitos específicos que serán satisfechos por el nuevo editor web de conexiones. Esta lista es el resultado de un estudio profundo de los posibles casos de uso. Para la definición de los requisitos se ha seguido la siguiente nomenclatura <Tipo de requisito>.<Identificador numérico>:

- Primer campo: define el tipo de requisito lo cual permite diferenciar entre requisitos funcionales, no funcionales o de diseño.
- Segundo campo: define el identificador número de cada requisito y sigue una notación por niveles.

3.1.4.1 Requisitos Generales del Sistema

Los requisitos mencionados a continuación, tienen relación a la presentación general de la herramienta de edición una vez el usuario seleccione abrir este desde el entorno de trabajo actual.

DIS.1.1: la interfaz del sistema presentará una barra de herramientas para gestionar los comportamientos parciales, las aplicaciones web y activar la vista independiente del comportamiento activo. Esta barra se ubicará en el lateral izquierdo, facilitando así su ubicación, y se mostrará siempre en todo momento.

FUN.1.1: el sistema permitirá cambiar en cualquier momento de la vista del esquema de conexiones del comportamiento global a la vista independiente del esquema de conexiones del comportamiento activo.

FUN.1.2: si un widget es borrado del entorno de trabajo actual, el sistema borrará la instancia de este de los comportamientos y del comportamiento global, incluido las conexiones salientes y entrantes.

FUN.1.3: si un operador es desinstalado de la cuenta del usuario, el sistema no lo borrará del mapa de conexiones, sino se mostrará de otra manera para notificar al usuario que ha sido borrado y pueda mover las conexiones entrantes y salientes.

3.1.4.2 Requisitos del Panel de Comportamientos Parciales

Los requisitos que se mencionan a continuación, prestan su atención al panel deslizante hacia la derecha, accesible desde la barra de herramientas, dedicada únicamente a los comportamientos parciales identificados por el usuario.

DIS.2.1: la interfaz del sistema presentará un panel relativo a los comportamientos parciales. Este panel se ubicará en el lateral izquierdo, facilitando así su ubicación, y se ocultará siempre que se realice cualquier operación fuera de este.

DIS.2.2: en el panel de comportamientos parciales, cada comportamiento será diferenciado por un título y una descripción; y a su vez, por el número de componentes (widgets, operadores y conexiones) del esquema de conexiones.

DIS.2.3: en el panel de comportamientos parciales, el comportamiento activo se resaltará más que el resto de comportamientos creados.

FUN.2.1: el sistema permitirá visualizar una lista con los comportamientos parciales pertenecientes al mashup. Si el mashup no tiene ningún comportamiento parcial creado, el sistema creará uno por defecto.

FUN.2.2: el sistema permitirá crear un comportamiento parcial a través de un formulario con los siguientes campos: título y descripción. Estos campos son opcionales. Además, el sistema ofrecerá la posibilidad de activarlo en la misma operación.

FUN.2.3: el sistema permitirá activar cualquier comportamiento parcial. Si el mashup no tiene ningún comportamiento activo, el sistema activará el primero de la lista por defecto.

FUN.2.4: el sistema no permitirá activar más de un comportamiento parcial al mismo tiempo, ni tampoco desactivarlos todos. Por defecto, siempre se tendrá un comportamiento activo.

FUN.2.5: el sistema permitirá modificar la información de un comportamiento parcial a través del mismo formulario de creación, cuyos campos estarán rellenados con los datos actualmente guardados. Esta operación podrá realizarse en cualquier momento.

FUN.2.6: el sistema permitirá borrar un comportamiento parcial. Esta operación eliminará el comportamiento del panel dedicado a ellos, además de, los componentes pertenecientes a él del esquema de conexiones del comportamiento global del mashup. Si dicho comportamiento estaba activado, el sistema activará el primero de la nueva lista por defecto. Esta operación no se mostrará accesible en el caso de que sea el último comportamiento.

FUN.2.7: el sistema permitirá visualizar previamente el esquema de conexiones de un comportamiento parcial con la finalidad de encontrar de forma más rápida un componente. Esta operación se llevará a cabo, exclusivamente en la vista del esquema de conexiones del comportamiento global, cuando el cursor esté por encima de un comportamiento parcial. Dicha operación no implicará la activación de este.

3.1.4.3 Requisitos del Panel de Aplicaciones Web

Los requisitos mencionados a continuación, están dedicados al panel deslizante hacia la derecha, accesible desde la barra de herramientas, que se centra únicamente a las aplicaciones web accesibles en ese momento desde el entorno de trabajo actual.

DIS.3.1: la interfaz del sistema presentará un panel relativo a las aplicaciones web. Esta sección se ubicará en el lateral izquierdo, facilitando así su ubicación, y se ocultará siempre que se realice cualquier operación fuera de este.

FUN.3.1: el sistema permitirá visualizar una lista con los widgets utilizados en el entorno de trabajo actual.

FUN.3.2: el sistema permitirá crear una única instancia por cada widget. El widget instanciado aparecerá como inhabilitado en la lista.

FUN.3.3: el sistema permitirá visualizar una lista con los operadores instalados en la cuenta del usuario.

FUN.3.4: el sistema permitirá crear varias instancias por cada operador.

FUN.3.5: el sistema permitirá crear una nueva instancia de cualquier aplicación web habilitada en el panel relativo a estos, arrastrándolo a la sección del esquema de conexiones del mashup. Esta operación solo se permitirá en la vista del esquema de conexiones del comportamiento global.

3.1.4.4 Requisitos Generales del Esquema de Conexiones

Respecto a los requisitos siguientes, estos se llevan a cabo sin tener en cuenta si la vista global esta activada o no.

DIS.4.1: la interfaz del sistema presentará una sección relativa al esquema de conexiones del mashup. Esta sección abarcará prácticamente todo el espacio posible.

DIS.4.2: el sistema mostrará en todo momento las operaciones posibles con respecto a los componentes pertenecientes al comportamiento activo.

DIS.4.3: el sistema permitirá distinguir visualmente entre las aplicaciones web instanciadas de tipo widget y de tipo operador.

FUN.4.1: el sistema permitirá actualizar la posición de cualquiera aplicación web instanciada, dentro del área de esta sección.

FUN.4.2: el sistema permitirá minimizar o recuperar el tamaño de las aplicaciones web instanciadas de tipo operador.

FUN.4.3: el sistema permitirá actualizar la posición de un conjunto de aplicaciones web instanciadas previamente seleccionadas; es decir, una selección múltiple.

FUN.4.4: el sistema permitirá ofrecer recomendaciones sobre las posibles conexiones entre las aplicaciones web instanciadas. Estas recomendaciones estarán basadas en palabras claves que cada entrada y salida tendrán en su descripción.

3.1.4.5 Requisitos de la Vista Global

Respecto a estos requisitos, solo se llevan a cabo cuando el usuario tiene activado la vista global del esquema de conexiones frente a un comportamiento parcial indicado por el usuario.

DIS.5.1: el sistema mostrará la posibilidad de añadir una instancia visible en el fondo cuando el cursor este sobre ella.

DIS.5.2: el sistema mostrará la posibilidad de añadir una instancia visible en el fondo cuando esta esté seleccionada.

FUN.5.1: el sistema permitirá distinguir las aplicaciones web instanciadas y las conexiones pertenecientes al comportamiento activo, de una forma más contrastada, del resto de componentes del comportamiento global del mashup.

FUN.5.2: el sistema ocultará casi siempre las operaciones posibles en los componentes visibles en el fondo con la finalidad de no entorpecer la visualización del comportamiento activo.

FUN.5.3: el sistema permitirá crear una instancia de la aplicación web seleccionada en el panel de aplicaciones web. Esta instancia formará parte del comportamiento activo.

FUN.5.4: el sistema permitirá reorganizar los puntos de entrada y de salida de una instancia perteneciente al comportamiento activo.

FUN.5.5: el sistema permitirá añadir una instancia, visible en el fondo, al comportamiento activo.

FUN.5.6: el sistema permitirá borrar la instancia de una aplicación web perteneciente al comportamiento activo. Esta operación borrará también las conexiones salientes y entrantes pertenecientes a dicho comportamiento. Si dicha instancia también pertenece a otro comportamiento, se fijará en el fondo del esquema de conexiones; sino, será borrado también del comportamiento global del mashup.

FUN.5.7: el sistema permitirá borrar en cascada la instancia de una aplicación web perteneciente al comportamiento activo, si este pertenece al menos a otro comportamiento. Dicha instancia será borrada de todos los comportamientos parciales y del comportamiento global del mashup, incluido las conexiones salientes y entrantes.

FUN.5.8: el sistema permitirá crear una nueva conexión entre dos instancias diferentes pertenecientes al comportamiento activo, seleccionando el punto de entrada y de salida de cada uno. Dicha conexión formará parte del comportamiento activo.

FUN.5.9: el sistema permitirá crear una nueva conexión entre la instancia de una aplicación web perteneciente al comportamiento activo y otra instancia diferente visible en el fondo, seleccionando el punto de entrada y de salida de cada una. Dicha conexión formará parte del comportamiento activo, incluido el extremo visible en el fondo.

FUN.5.10: el sistema permitirá crear una nueva conexión entre dos instancias visibles en el fondo, seleccionando el punto de entrada y de salida de cada una. Dicha conexión formará parte del comportamiento activo, incluido los extremos.

FUN.5.11: el sistema no permitirá crear una conexión ya existente en el comportamiento activo; es decir, duplicar una conexión existente. Por defecto, se borrará la conexión duplicada.

FUN.5.12: el sistema permitirá crear una conexión ya existente en el caso de que dicha conexión este visible en el fondo. Dicha conexión existente formará parte del comportamiento activo y la conexión creada será borrada.

FUN.5.13: el sistema permitirá añadir cualquier conexión visible en el fondo. Dicha conexión formará parte del comportamiento activo, incluido si alguno de los extremos también esta visible en el fondo. Esta operación estará oculta en un principio para no sobrecargar de operaciones posibles el mapa de conexiones.

FUN.5.14: el sistema mostrará la posibilidad de añadir una conexión, visible en el fondo, cuando el cursor este sobre dicha conexión.

FUN.5.15: el sistema mostrará la posibilidad de añadir una conexión, visible en el fondo, cuando dicha conexión esté seleccionada.

FUN.5.16: el sistema mostrará la posibilidad de añadir una conexión, visible en el fondo, cuando uno de los extremos, de dicha conexión, esté seleccionado.

FUN.5.17: el sistema permitirá modificar la curvatura de una conexión perteneciente al comportamiento activo. Esta modificación también afectará al resto de comportamientos.

FUN.5.18: el sistema permitirá borrar una conexión perteneciente al comportamiento activo. Si esta conexión pertenece también a otro comportamiento, esta estará visible en el fondo. En cualquier otro caso, dicha conexión será borrada también del comportamiento global del mashup.

3.1.4.6 Requisitos de la Vista Independiente

Finalmente, estos últimos requisitos solo tienen cabida cuando se visualiza el comportamiento parcial indicado por el usuario de forma independiente; es decir, que la vista global del esquema de conexiones este desactivado.

FUN.6.1: el sistema mostrará únicamente los componentes pertenecientes al comportamiento activo.

FUN.6.2: el sistema no permitirá borrar componentes del comportamiento activo.

FUN.6.3: el sistema no permitirá crear nuevos componentes en el comportamiento activo.

3.2 Fase de Implementación

Una vez finalizada la fase de análisis del nuevo sistema, se realizó un primer diseño de alto nivel que ante todo plasmase las nuevas características y permitiese observar los casos de uso realizamos anteriormente. Tras toda observación dada, se dispuso la implementación del código fuente del lado cliente de este editor web de conexiones. Las tecnologías elegidas para este desarrollo fueron JavaScript y Sass, debido a que son las más actuales hoy en día y que son las que utiliza la plataforma WireCloud.

Después de varias semanas, se realizaron una serie de pruebas unitarias y de sistema para comprobar con certeza la correcta funcionalidad del sistema. También se tuvo en cuenta todos los posibles errores que pueden darse dentro y fuera del editor web; por ejemplo, cuando se borra un widget u operador de la cuenta del usuario. Respecto al modo de gestionar e identificar los diferentes elementos creados en el área de edición, se decidió seguir la misma línea ya que otras herramientas de WireCloud hacían uso de algunos de ellos.

Por último, aparte de documentar las nuevas clases introducidas en el código fuente del nuevo sistema, también se tuvo en cuenta como incorporar el nuevo editor web de conexiones orientado a comportamientos a la plataforma WireCloud. Entre las soluciones aportadas, se eligió por realizar una migración de datos ya que así los usuarios ya registrados en esta plataforma pudiesen mantener prácticamente intactos los mashups creados antes de dicha incorporación. Estos mashups solo sufrirían unos pequeños cambios para permitir el uso de comportamientos parciales dentro de un mashup.

3.2.1 Estructura del sistema

En esta sección se detalla la actual estructura del nuevo editor web de conexiones que incorpora la plataforma WireCloud. Respecto al antiguo sistema, algunos módulos fueron modificados casi en su totalidad, otros parcialmente modificados y los últimos se mantuvieron prácticamente intactos. La razón de estos últimos es su trabajado código fuente y que en el nuevo sistema, las funcionalidades proporcionadas por estos se quería que continuaran.

También hay que mencionar, que se introdujeron nuevos módulos los cuales se encargan básicamente de la gestión de los comportamientos parciales que el usuario vaya

identificando. A continuación, se ilustra la distribución de los paquetes del editor web dentro de la plataforma WireCloud.

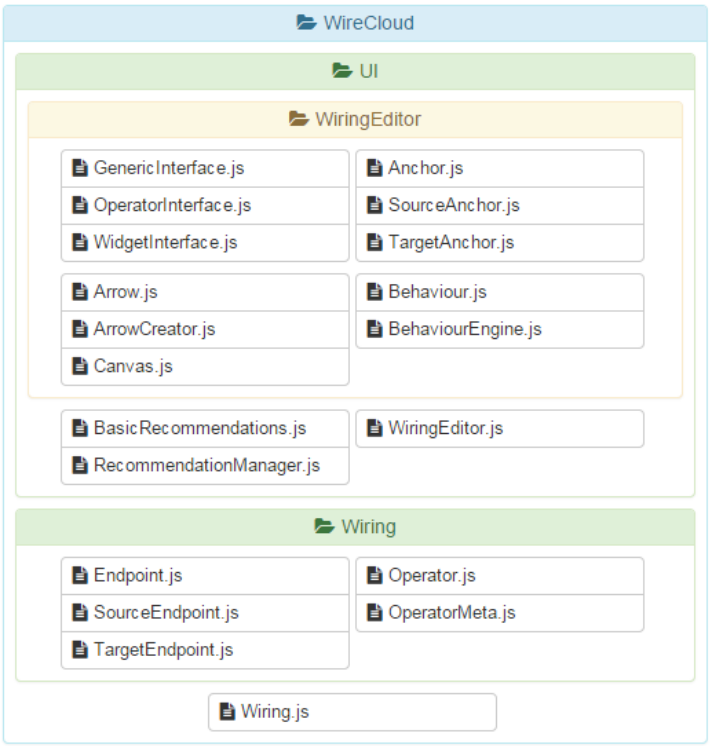

Figura 19: distribución de paquetes y módulos

En la ilustración anterior, se pueden observar todos los paquetes y módulos que intervienen durante el uso del editor web de conexiones. Todo el código fuente del editor web se encuentra ubicado dentro del paquete UI que proporciona WireCloud y los módulos que hacen posible la identificación de las diferentes instancias que puede tener una aplicación web de tipo operador y la gestión de los operadores y las conexiones, que han sido utilizados en el esquema de conexiones de cierto mashup, en el entorno de trabajo actual de cualquier usuario.

Los módulos OperatorMeta y Operator se encargan de definir el comportamiento de cada operador genérico y de identificar las instancias que se han producido de cada operador dado respectivamente. Y el módulo Wiring se encarga de funcionar como motor de conexiones de un entorno de trabajo dado.

Por último, se muestra una ilustración que resalta con azul oscuro los módulos modificados y los nuevos módulos de este editor web de conexiones orientado a comportamientos. El resto como se ha dicho con anterioridad, se han mantenido prácticamente intactos en su totalidad.

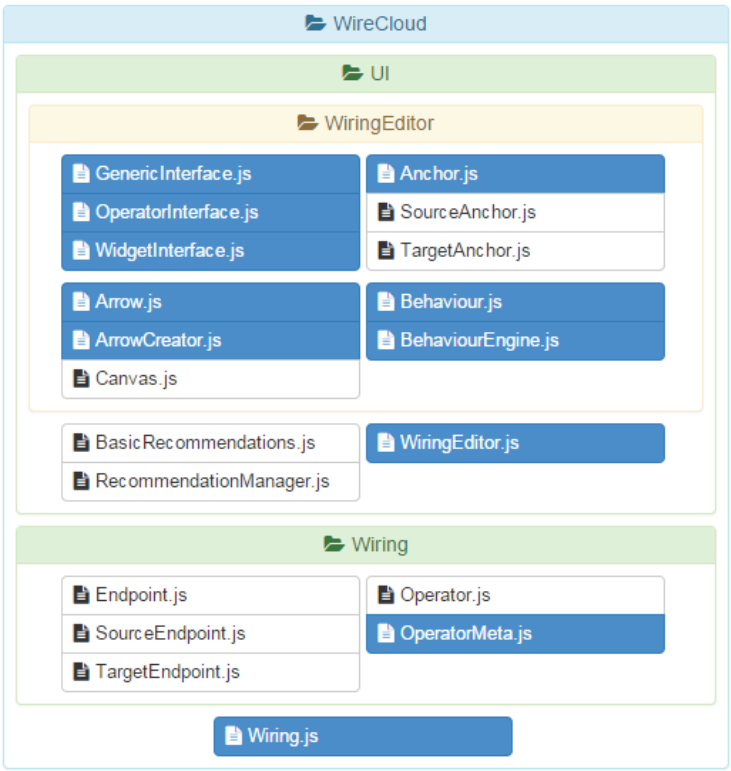

Figura 20: identificación de los módulos modificados y nuevos

3.2.2 Documentación del sistema

En esta sección se describe el funcionamiento general de las clases que este nuevo editor web incluye. La documentación de las clases se realizó una vez acabado la ejecución de las pruebas realizadas para evaluar el correcto funcionamiento del sistema. En la ilustración siguiente se puede visualizar un diagrama general de las clases que participan en la edición visual del esquema de conexiones de un mashup. Hay que mencionar, que este diagrama de clases ha sido reducido quitando las funciones y los atributos menos importantes de diferentes clases para una mejor resolución de la imagen. Además se han resaltado más las flechas blancas de *herede de* y las flechas negras de *instanciado por* para una mejor comprensión.

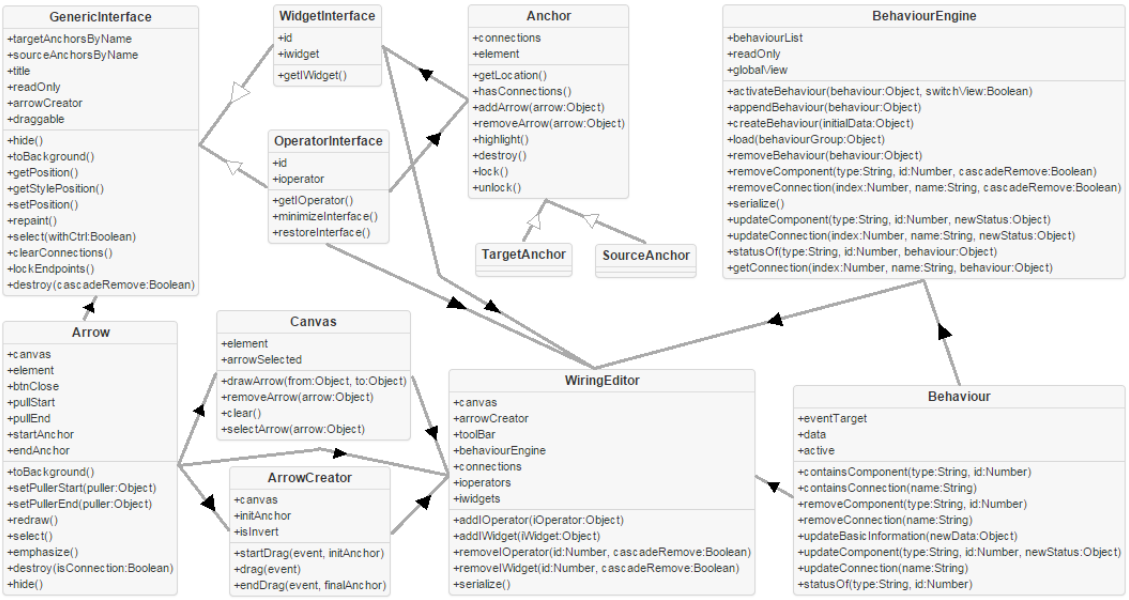

Figura 21: diagrama general de clases

Las siguientes subsecciones están dedicadas a la explicación en profundidad de las clases más importantes que este nuevo editor web de conexiones contiene. Además, solo se detallaran las funciones y las propiedades más destacables debido a que la mayoría de los miembros de una clase fueron implementados para servir como soporte a otras clases o a la misma clase.

3.2.2.1 Módulo WiringEditor

La clase WiringEditor como el propio nombre dice es la clase principal del editor web de conexiones. Esta clase se encarga básicamente de lo siguiente:

- Cargar la visualización entera del sistema; es decir, la barra de herramientas y el área de edición.
- Gestionar de los eventos producidos en la barra de herramientas y en el área de edición.
- Almacenar las aplicaciones web disponibles, las aplicaciones web instanciadas y las conexiones existentes.
- Interactuar con el motor de comportamientos que más adelante se describe.
- Serializar los datos del estado actual del esquema de conexiones que después serán enviados al servidor web.

Entre todos los métodos públicos que ofrece esta clase, se pueden encontrar *serialize*. Este método se encarga de serializar los datos necesarios para almacenar y volver a

restaurar posteriormente el estado actual del esquema de conexiones de un mashup. Estos datos son enviados al servidor que corre en ese momento. Respecto a los datos serializados, en la organización interna de estos datos pueden distinguirse dos partes muy claramente: la parte de visualización y la parte de negociación. La razón de esta separación es debida a un deseo de independencia por parte de ambas, ya que si en caso de fallo en una de ellas puede reutilizarse la otra para prácticamente recuperarse en su totalidad. Además de que la parte de negociación es la que realmente necesita el motor de conexiones que trabaja por debajo de un entorno de trabajo. A continuación, se describe detalladamente estas dos partes:

- Parte de negociación: la estructura de estos datos permite almacenar los operadores utilizados y las conexiones creadas en la última modificación realizada en el área de edición.
    - Cada conexión contiene las aplicaciones web (los números de identificación y tipos de aplicación web) y los puntos de entrada y salida (los nombres proporcionados) que participan en ella. Es la información mininamente necesaria para que el motor de conexiones pueda interpretarla.
    - Cada operador contiene el identificador del operador instanciado y el identificador de la instancia dentro del área de edición. Esta información es necesaria para reconocer que instancia de que operador está siendo utilizada en una conexión.

  Estos datos, incluida la información relativa a los widgets utilizados en el entorno de trabajo, son necesarios a la hora de que ejecutar todos los widgets presentes en un entorno de trabajo.
- Parte de visualización: la estructura de estos permite almacenar los comportamientos parciales identificados por el usuario final y el comportamiento global del mashup. Estos datos son importantes de cara al usuario final ya que es la parte imprescindible para la restauración correcta de todas las posiciones y modificaciones de cada uno de los componentes presentes en el esquema de conexiones de un mashup. La razón de separar ambas partes, otorga la posibilidad de que hubiera un error en estos datos restaurar todo lo posible desde los datos presentes en la parte de negociación.
    - Cada comportamiento parcial está ubicado según su fecha de creación y almacena el título y la descripción dada por el usuario, las aplicaciones web utilizadas y también las conexiones existentes. Además de un booleano para saber si estaba activo o no. Toda aplicación web viene dado

por un identificador, el orden de los puntos de entrada y de salida, la posición actual en el área de edición y en el caso de los operadores, si estaba minimizado o no. En relación a la información guardada de cada conexión, es la necesaria para conocer la curvatura que tiene y los puntos extremos de esta. Todo esto permite que cada comportamiento parcial permita la modificación independiente de sus elementos.

- Por último, el comportamiento global recolecta todos los componentes, tanto aplicaciones web y conexiones, existentes en el esquema de conexiones. Y sigue la misma estructura de datos que el resto de comportamientos parciales.

Los métodos de *addIWidget* y *addIOperator* realizan el último paso de agregación de una aplicación web instanciada al comportamiento parcial activo en ese momento. Estos métodos comprueban en primer lugar si la aplicación web instanciada es una nueva creada desde la barra de herramientas o simplemente el usuario ha querido compartir una instancia existente en el fondo de la vista global del esquema de conexiones.

En cambio, los métodos de removeIWidget y removeIOperator realizan exactamente lo opuesto. Estos métodos son los encargados de verificar si la aplicación web instanciada que se quiera borrar del comportamiento activo pertenece a otro comportamiento parcial también. Y de esto, saber si mostrar la ventana de borrado en casada o no.

Cualquier evento producido durante la edición del esquema de conexiones de un mashup, que sea de suma importancia, es gestionado desde esta clase. Entre estos eventos se puede encontrar eventos como conexión creada, conexión borrada, compartir instancia, borrar instancia, abrir panel de comportamientos parciales y otras muchas más.

3.2.2.2 Módulo Behaviour

La clase Behaviour es completamente nueva y se encarga de los datos actuales de un comportamiento visual además de la parte visual de este en el panel dedicado a los comportamientos parciales identificados por el usuario. También es el encargado de actualizar en tiempo real los componentes que han sido borrados o añadidos en la barra de estado que cada comportamiento parcial muestra en la representación ubicada dentro del panel citado anteriormente.

Hay que destacar métodos como *updateComponente* o *updateConnection* que actualizan o crean un componente en el comportamiento parcial que representa. Estos métodos, y el

resto de métodos, son llamados exclusivamente por la clase BehaviourEngine que en la siguiente sección se describe.

También podemos encontrar métodos similares a los anteriores, como *containsComponent*, *containsConnection*, *removeComponent* y *removeConnection* los cuales comprueban si un componente o una conexión existe en ese comportamiento parcial y borran el componente y la conexión pasada por entrada del comportamiento parcial, respectivamente.

Por último, el método de *statusOf* retorna la información de la conexión o componente que se desea para que sea visualizada en la vista independiente. A través de este método el motor de comportamientos puede cargar de forma rápida toda la información necesaria para cargar la vista independiente de un comportamiento parcial específico.

3.2.2.3 Módulo BehaviourEngine

Como el propio nombre indica, la clase BehaviourEngine es la encargada de gestionar todos los comportamientos parciales identificados por el usuario y la actualización en tiempo real de todos ellos. La instancia de esta clase trabaja junto con el editor web de conexiones y es la que permite que esta se oriente a comportamientos. Cualquier operación que se realice en el editor de conexiones, esta se presenta como intermediario. Prácticamente, las funciones que esta proporciona están presentes en todos los eventos importantes que pueden producirse durante la edición del esquema de conexiones de un mashup.

El cambio de la vista global a la independiente o viceversa, lo gestiona está a través de sus propios eventos. Esta clase a través del método *activateBehaviour* permite cambiar de manera fluida la visualización del esquema de conexiones, ocultando o mostrando los componentes específicos además de cargar los datos correctos en ese momento.

Incluye métodos como *createBehaviour* y *removeBehaviour* para crear y borrar un comportamiento parcial respectivamente. El primer método permite crear un comportamiento parcial a partir de los datos del formulario de creación, los cuales pueden estar vacíos. Y el otro método, es el que permite borrar completamente un comportamiento parcial es decir, todos los componentes pertenecientes a este en el área de edición.

A través del método *load* puede restaurar los comportamientos parciales y el comportamiento global guardados en el servidor web. También comprueba que los datos

estén correctos y que solamente un comportamiento parcial este activo además de encargarse de que si el usuario no tiene ningún comportamiento parcial, crearle uno por defecto.

El método de *serialize* es el encargado de serializar la parte de visual del esquema de conexiones de un mashup; es decir, todos los comportamientos parciales y el comportamiento global, y enviárselos al editor de conexiones. El cual los adjuntará con la parte de negociación y de esta manera enviárselos finalmente al servidor web.

Respecto al resto de métodos públicos que ofrece esta clase, están relacionados con la actualización del esquema de conexiones del comportamiento activo. Métodos como *updateComponent*, *updateConnection*, *removeComponent* y *removeConnection* permiten al editor de conexiones gestionar correctamente las operaciones realizadas por el usuario sobre los componentes y las conexiones en relación al comportamiento parcial activo en ese instante.

Finalmente, el método *statusOf* es de suma importancia a la hora de cargar la información correcta sobre una aplicación web instanciada o una conexión. En otras palabras, cuando se visualiza previamente un comportamiento parcial, se activa un nuevo comportamiento parcial y se cambia de visualización el editor de conexiones consulta al motor de comportamientos sobre los datos correctos de cada componente presente en el esquema de conexiones a través del método citado.

3.2.2.4 Módulo GenericInterface

Esta clase tiene un único cometido, funcionar como si fuera una clase abstracta de la cual heredaran los operadores y widgets instanciados en el editor web. Esta clase contiene todos los miembros (atributos y funciones) que deberían tener en común OperatorInterface y WidgetInterface. Además, estas clases también son utilizadas en la representación gráfica de una aplicación web en el panel dedicado a ellas, accesible desde la barra de herramientas.

El sistema de arrastrado mencionado al principio de este capítulo, es usado por las instancias de esta clase. Permitiendo de este modo, la posibilidad de arrastrar la representación gráfica de una aplicación web, que se desee utilizar, al área de edición de una manera intuitiva y fluida. Estas interfaces producidas, están compuestas por el título de dicha aplicación web, el cual funciona de identificador, en la cabecera de la misma; además de los puntos de entrada, ubicados en el lateral izquierdo, y los puntos de salida, ubicados en el lateral derecho. Cada punto de entrada o de salida puede visualizarse como

la composición del nombre identificador y el ancla que permite crear o enlazar conexiones.

Dentro del área de edición, el elemento HTML que contiene esta clase hace uso del sistema de arrastrado también lo cual permite que el usuario pueda movilizarlo a otras posiciones permitidas. Respectivo al sistema de arrastrado, este sistema es muy versátil ya que se utiliza en diferentes herramientas de WireCloud, como por ejemplo en los widgets visualizados en el entorno de trabajo. Su funcionamiento se base en tres fases: comienzo del arrastrado, arrastrando un elemento y finalmente, finalización del arrastrado.

Por último, el método *makeSlotsDraggable* permite al usuario modificar el orden de los puntos de entrada o de salida de la interfaz asignada. Esta funcionalidad también hace uso del sistema de arrastrado mencionado anteriormente, aunque en este caso los movimientos de un punto final desplazándose están más limitados. De tal modo, que el usuario tenga la sensación de subir o bajar la posición de un punto de entrada o de salida seleccionado. Luego los cambios producidos son notificados al motor de comportamientos a través del editor de conexiones.

3.2.2.5 Módulo Arrow

La última clase a detallar es la encargada de representar visualmente una conexión existente entre los puntos de entrada y de salida de dos aplicaciones web. La complejidad de algunos de los métodos de esta clase, es debido al uso de la curva de Bezier. Dentro del editor de conexiones, esta clase es conocida como el motor de conexiones del área de edición la cual en un principio se mencionó que iba a permanecer prácticamente intacta. Aun así se realizó alguna modificación superficial que afecta sobre todo a la ejecución de los eventos.

La edición de la curvatura de una conexión se mantiene inmune a cambios, ya que su trabajado código se quería mantener en esta nueva versión. Pero la posibilidad de editar una conexión solo debería estar permitida cuando esta pertenece al comportamiento parcial activo en ese momento, y por esta razón el evento que permite editar una conexión también está controlado por el motor de comportamientos.

Con la alteración del código fuente relacionado con los eventos de edición y selección, ahora se puede visualizar conexiones en el fondo, resaltar una conexión sin necesidad de abrir la edición de esta y sobre todo abrir las puertas a una mejor gestión de la superposición de unas conexiones con otras.

Para acabar, métodos como *emphasize* o *select* permite al usuario resaltar una conexión, sin importar si está en el fondo cuando la vista global esta activada, sobre el resto de las demás.

3.2.3 Evaluación del sistema

Para evaluar el correcto funcionamiento del nuevo sistema, se han realizado diferentes tipos de pruebas. Entre las cuales se pueden encontrar pruebas unitarias, de integración y finalmente de sistema. Sin importar el tipo de prueba, cada una de las realizadas tienen como objetivos en común la finalización esperada de cada una de las funcionalidades permitidas en el editor web de conexiones, la adaptación al servidor web que utiliza la plataforma WireCloud y el correcto comportamiento con el resto de aplicaciones proporcionadas en esta.

A través de las pruebas de sistema, se ha podido controlar el correcto funcionamiento de cada una de las acciones dadas durante la edición del esquema de conexiones de un mashup. Este tipo de pruebas se realizan con un sistema de pruebas visuales llamado Selenium. Este sistema permite a un desarrollador, desde el lado servidor, evaluar el comportamiento del lado cliente de un servicio web utilizando cualquier navegador web actual.

Por último, las pruebas unitarias y de integración bastaron para conocer que la incorporación de este nuevo editor web de conexiones fue la deseada. Además de probar que interactúa correctamente con las otras aplicaciones presentes en WireCloud.

3.2.4 Persistencia de los datos

Con respecto a la creación, modificación o eliminación de conexiones durante la visualización del esquema de conexiones, el entorno de trabajo actual tiene en cuenta los cambios respectivos a dichas acciones solamente cuando el usuario retorna a la pantalla principal. Esto es debido a que en ese momento el sistema serializa los datos relacionados con el estado actual de los esquemas de conexiones de los comportamientos parciales y el comportamiento global en formato JSON, para luego ser enviados al servidor.

Todas las instancias de aplicaciones web de tipo widget utilizadas en el esquema de conexiones se identifican según el número asignado en el entorno de trabajo actual por parte de la plataforma WireCloud. De este modo, si el extremo de una conexión es la instancia de un widget, este se identificada con el mismo número. Esto también afecta a

cada comportamiento parcial, e incluso al comportamiento global; es decir, si la instancia de un widget pertenece a algún comportamiento, se identifica de igual manera.

En relación con las instancias de aplicaciones web de tipo operador utilizadas también en el esquema de conexiones, estas son almacenadas igual que en el antiguo sistema. Se almacenan teniendo en cuenta dos identificadores; uno para diferenciarlo del resto de operadores instalados en la cuenta del usuario, y otro para diferenciarlo en el esquema de conexiones. Este último identificador, es similar al de los widgets. La información relativa a estas instancias, una vez almacenadas en el servidor, se pierde cuando el usuario vuelve a la pantalla principal debido a que es prescindible.

Para acabar, la información relativa a los comportamientos parciales también se pierde una vez el usuario vuelva a la pantalla principal. En caso de que el usuario al entrar a la pantalla de edición del mapa de conexiones del entorno de trabajo no tenga un comportamiento parcial creado, el sistema crea uno por defecto.

3.2.5 Publicación de un mashup

Entre el abanico de posibilidades que ofrece la plataforma WireCloud, se puede encontrar la posibilidad de publicar tu propio entorno de trabajo como si fuera un mashup. Permitiendo de este modo, que cualquier usuario registrado y con acceso a la misma tienda web pueda descargarse el nuevo mashup en cuestión. Lo cual, incluir la información respectiva a los comportamientos parciales identificados por el diseñador dentro de los datos de esta publicación, es necesaria ya que es uno de los objetivos principales de este trabajo.

Aun así, WireCloud incluía los datos relacionados con el estado visual del esquema de conexiones en los datos de la publicación a partir de la integración del antiguo sistema. Esto facilita, la modificación de la estructura de los datos relacionados con la parte visual del esquema de conexiones para incluir los comportamientos parciales y soportar un funcionamiento orientado a comportamientos. Con esto, se consigue alcanzar el objetivo principal que concierne al uso y comprensión de cualquier mashup, sin importar su complejidad, por parte de un usuario ajeno. Todo esto es posible gracias a la colaboración por parte del diseñador a la hora de identificar y detallar los diferentes comportamientos parciales que cierto mashup pueda tener.

3.2.6 Gestión de Errores

Los posibles errores a tener en cuenta están relacionados con cargar por primera vez un nuevo mashup o de borrar aplicaciones web como son los widgets u operadores. Tras esta mención, si nos centramos en este tipo de situaciones pueden darse los siguientes casos:

- Cuando un mashup no tiene información respectiva a la parte visual del esquema de conexiones; es decir, los posibles comportamientos parciales y el comportamiento global del mashup. Por defecto, el sistema crea un comportamiento parcial y un comportamiento global, ambos vacíos.
- Cuando un mashup tiene algún error de estructura o incoherencia de datos relativo a las conexiones existentes en los comportamientos parciales y el comportamiento global. Por defecto, el sistema mostrará las instancias de aplicaciones web utilizadas en el esquema de conexiones, pero vacío de conexiones.
- Cuando un mashup tiene algún error en los datos visuales de cualquier instancia. Por defecto, el sistema fijara a todas las instancias con errores en una posición inicial para que luego el usuario las recoloque en las posiciones que desee.
- Cuando una aplicación web de tipo widget es borrada del entorno de trabajo. Por defecto, también será borrado del esquema de conexiones en el caso de que esté siendo utilizado en algún comportamiento parcial incluyendo las conexiones entrantes y salientes de este.
- Cuando una aplicación web de tipo operador es borrada de la cuenta del usuario. Por defecto, se mantendrán en el esquema de conexiones en el caso de que estuviera siendo utilizada en algún comportamiento parcial. Esto permite que el usuario pueda decidir si volver a instalar dicha aplicación web o mover las conexiones entrantes y salientes.

Para proporcionar estabilidad a la plataforma WireCloud, el nuevo sistema debe tener en cuenta todos los posibles errores a la hora de cargar un mashup ajeno o simplemente cuando el propio usuario interactúe con herramientas de la plataforma que puedan afectar al sistema. De modo que el usuario pueda abrir el esquema de conexiones de cualquier mashup sin fallo alguno por parte del sistema. A su vez, se intenta mostrar soluciones en caso de que haya ocurrido algún inconveniente.

3.2.7 Migración de Datos

Además de satisfacer con los requisitos específicos mencionados con anterioridad, se tuvo en cuenta la adaptación del nuevo sistema en una plataforma WireCloud. Debido a que

ya existen usuarios que hacen uso de una versión que incluye el antiguo editor web de conexiones, se optó por llevar a cabo una migración de datos que afecta a los datos almacenados en el estado actual del esquema de conexiones de cualquier entorno de trabajo.

Entonces, el estado del esquema de conexiones de cada entorno de trabajo existente en una plataforma WireCloud será modificado para permitir el funcionamiento orientado a comportamientos. Esta modificación afectará sobre todo a la parte visual de los componentes de un mashup, la cual incluirá estos componentes tanto en el comportamiento global como en un comportamiento parcial que el sistema creará por defecto.

3.3 Diseño de un mashup a partir de comportamientos

Seguidamente se presenta a modo de conclusión la nueva forma en que se diseñan las conexiones de un mashup concreto a partir de los comportamientos parciales identificados en el mismo. En la primera ilustración se muestra el entorno de trabajo que se tiene para la puesta en escena de este nuevo editor web. En el cual se pueden contemplar cuatro widgets, uno para listar técnicos, uno para ver el perfil del técnico seleccionado, uno para realizar una videollamada al técnico seleccionado y finalmente otro para mostrar la localización del técnico en cuestión.

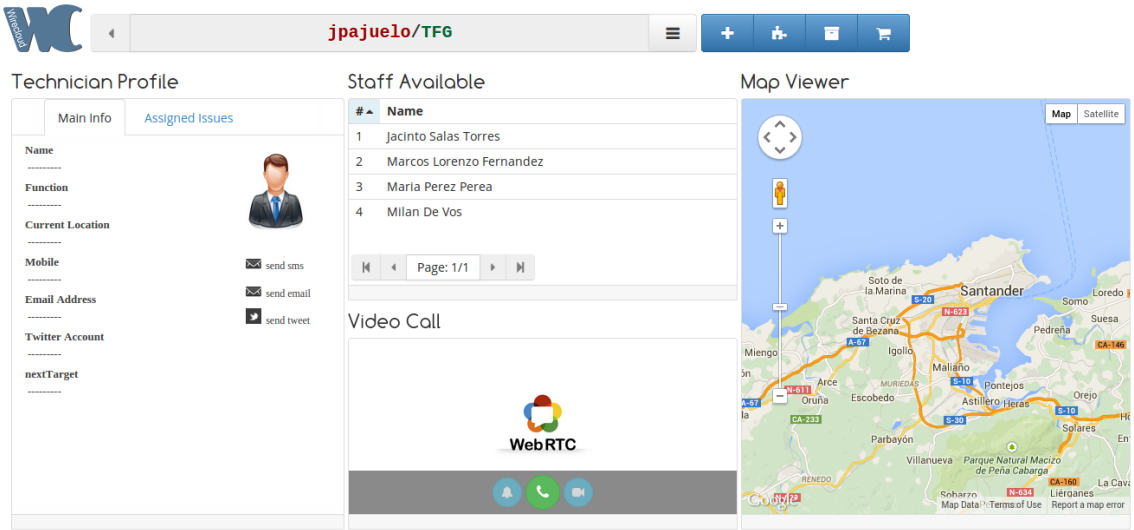

Figura 22: visualización del entorno de trabajo

En un primer lugar se puede identificar un comportamiento parcial dado en el mashup en cuestión, el cual trata de visualizar siempre la localización del técnico que se desea conocer. En la

ilustración de abajo podemos ver los componentes que pertenecerían a este comportamiento parcial descrito.

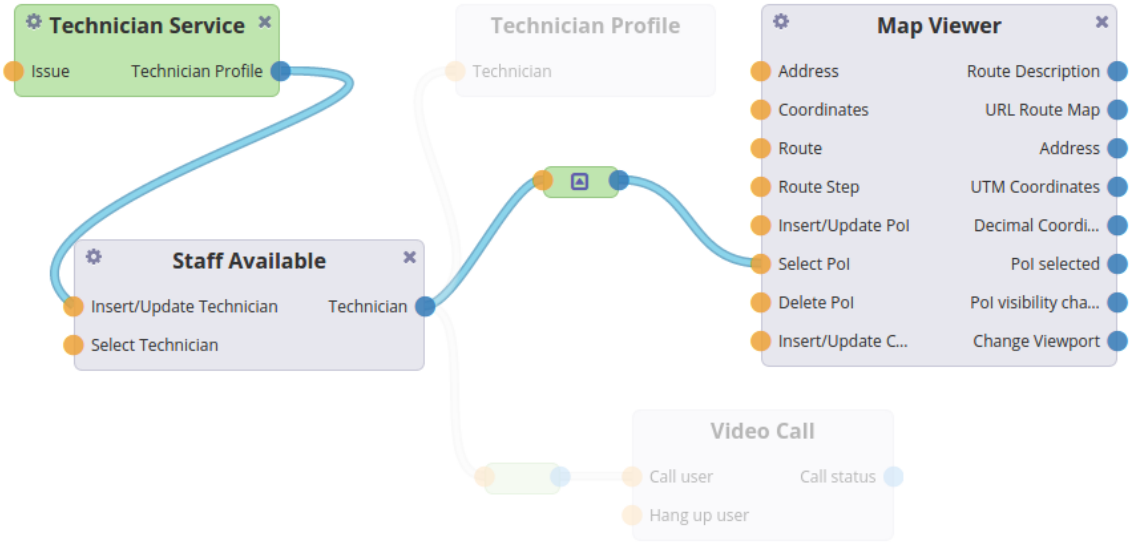

Figura 23: primer comportamiento parcial identificado

En la barra de herramientas se podría ver el título y la descripción que representa a este comportamiento parcial en la que también aparece el número de cada tipo de componente presente en este.

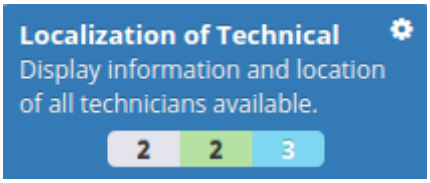

Figura 24: representación del primer comportamiento parcial

Finalmente, se puede identificar el otro comportamiento parcial que puede darse en el mashup. Este se encargaría de mostrar los datos de perfil de un técnico y en el caso de querer solicitarlo, realizar una videollamada para contactar con él. A continuación, se muestra el esquema de conexiones relativo a este comportamiento parcial y la representación de este en la barra de herramientas.

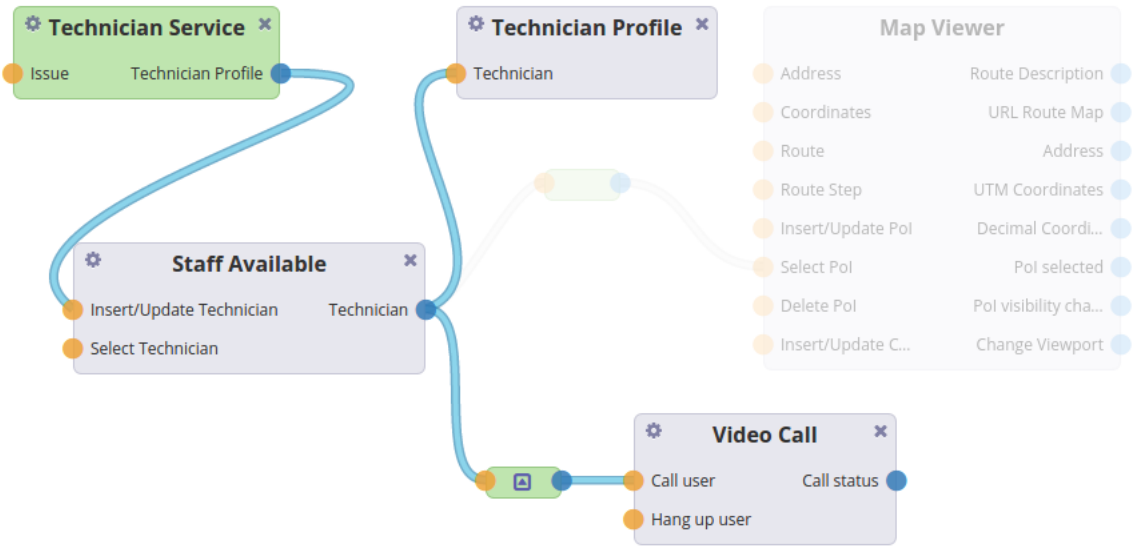

Figura 25: segundo comportamiento parcial identificado

Como la propia descripción indica la cohesión de todos los componentes de este comportamiento parcial permite realizar una videollamada a un técnico disponible previamente indicado por el usuario a través de la GUI que proporciona cada widget.

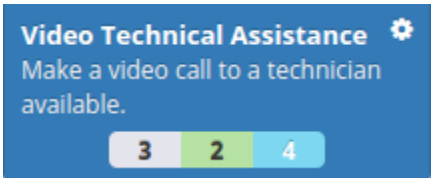

Figura 26: representación del segundo comportamiento parcial

CAPÍTULO 4

RESULTADOS Y CONCLUSIONES

En este Trabajo Fin de Grado, se ha diseñado y construido un nuevo editor web de conexiones orientado a comportamientos, que mantiene la metáfora visual basada en flujo de datos y eventos que proporcionaba el anterior diseño y la mejora para permitir un diseño gradual y por partes del esquema de conexiones.

La evolución de este nuevo sistema partió de un análisis exhaustivo del editor web de conexiones que incluye actualmente la plataforma WireCloud y de todas las posibilidades que el nuevo editor web de conexiones debe ofrecer, a la vista de los principales problemas descritos por sus usuarios.

Posteriormente, durante la fase de implementación se han realizado pruebas de sistema que evalúan el correcto funcionamiento en los distintos casos de uso considerados.

A continuación, se recogen las principales conclusiones técnicas del trabajo, consecuencia de las decisiones técnicas tomadas en algunas de las reuniones realizadas para el logro de los objetivos principales, además de las conclusiones personales que la realización de este trabajo me ha aportado. Finalmente se perfilan algunas líneas futuras que podrían seguir los desarrolladores de la plataforma WireCloud en relación con el nuevo editor contribuido desde este trabajo.

4.1 Conclusiones Técnicas

A lo largo del estudio previo a la realización del nuevo editor web de conexiones orientado a comportamientos, se ha tenido en cuenta el uso de las tecnologías actuales en el ámbito de las aplicaciones web. Como la plataforma WireCloud está en continuo desarrollo, y además, tiene presente la evolución y el uso de nuevas tecnologías, se decidió utilizar las mismas tecnologías de las que se hacía uso. Entre las cuales, Sass ha permitido que la modificación del diseño del nuevo sistema implementado sea fácil e intuitivo para otros desarrolladores en el caso que quieran personalizar su propio editor web de conexiones a su manera.

Durante la evolución de este trabajo, se han tomado varias decisiones técnicas que han fijado el rumbo hacia el logro de los objetivos principales. Tales como el uso de tecnologías web compatibles con los navegadores web de hoy en día o el diseño de la

nueva barra de herramientas, para así otorgar más espacio de la pantalla al esquema de conexiones. Además de la posibilidad de añadir nuevas funcionalidades en un futuro. De igual forma, la utilización de los eventos que puede gestionar JavaScript para facilitar al usuario un mejor entendimiento de la herramienta; por ejemplo, ocultando opciones para no sobrecargar la pantalla de acciones.

En definitiva, el nuevo concepto de comportamientos en un mashup ha demostrado una mejora respecto al anterior. Sobre todo a la hora de visualizar un mashup con un gran número de componentes. La nueva metáfora visual adoptada permitirá a usuarios con poca experiencia sentirse capaces de diseñar un mashup diferente y a la par que complejo.

Como pudimos ver en el capítulo anterior, la antigua versión del editor web de conexiones podía dar lugar a la visualización del esquema de conexiones de un mashup muy complejo, haciéndose costoso tanto su entendimiento como su edición. Y después de llevar a cabo este trabajo, podemos apreciar en las ilustraciones siguientes la resolución de dichos inconvenientes gracias a la adopción de la nueva metáfora visual citada anteriormente. En ellas pueden apreciarse cada uno de los comportamientos que pueden darse en ese mashup; y comprobar la mejora visual que se ha adquirido tras la adopción de tan mencionada metáfora.

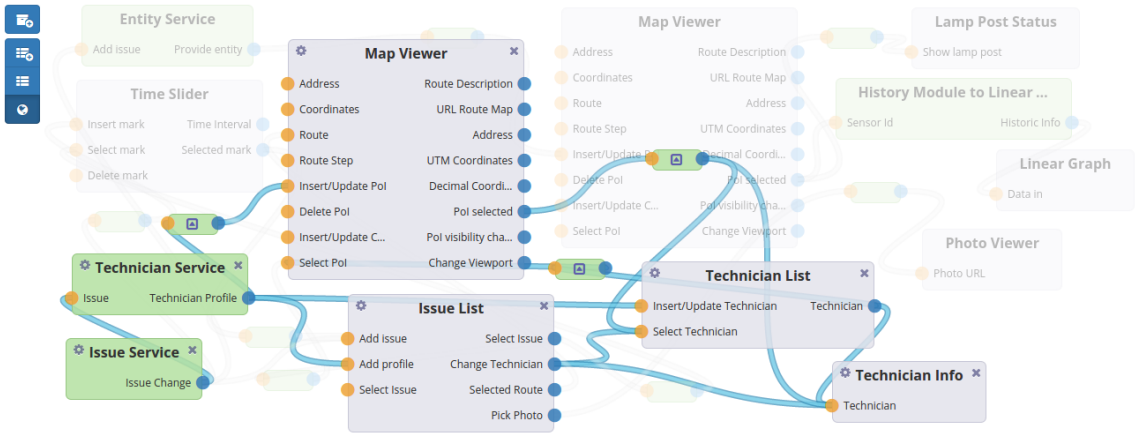

Figura 27: vista global del primer caso de uso

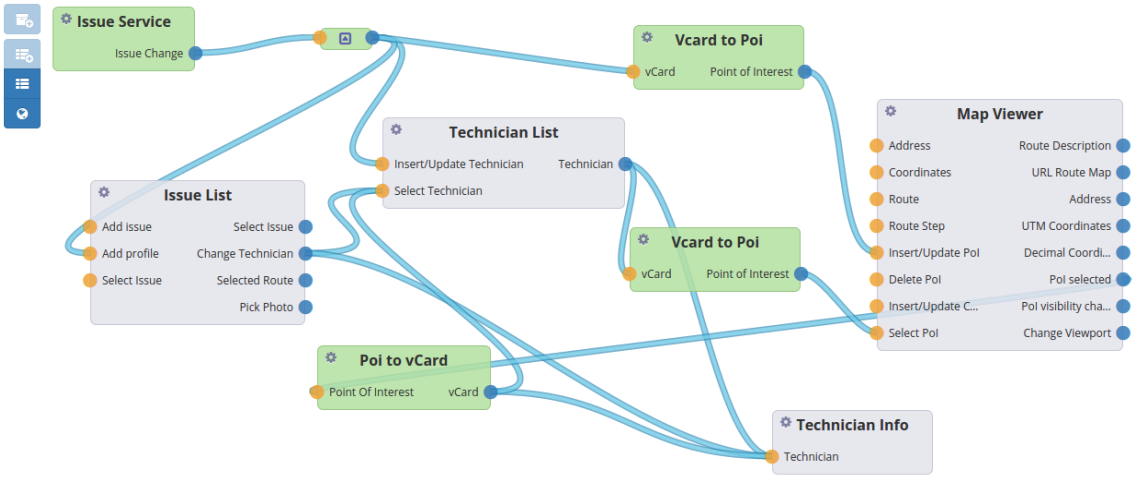

Figura 28: vista independiente de un comportamiento parcial

4.2 Aportaciones Personales

En primer lugar, me gustaría mencionar la buena experiencia que me ha supuesto trabajar en el laboratorio CoNWeT del grupo de investigación CETTICO, ubicado en la Escuela Técnica Superior de Ingenieros Informáticos de la UPM. A lo largo de toda mi estancia realizando este trabajo, he adquirido conocimientos útiles en relación con el manejo de servicios web, ya sea en el lado servidor o en el lado cliente. Enfatizando sobre todo en este último, es decir el lado cliente, con el que he aprendido a utilizar mejor tecnologías actuales como son JavaScript o Sass, las cuales están muy demandadas en profesiones referidas a este entorno. Aun así, todos estos nuevos conocimientos me servirán de igual manera en campos de investigación, o ámbitos empresariales, todos ellos relacionados con las aplicaciones web.

Respecto a mis aptitudes, he comprendido mejor el trabajo en grupo, permitiéndome alcanzar en conjunto metas a corto plazo y con fecha limite; o simplemente, cumplir con los objetivos propuestos. Además de tener en cuenta todas las decisiones tomadas en una reunión y luego llegar a una conclusión en común para solventar ciertas dudas o trazar nuevas líneas. Con todo esto, he conseguido organizarme mejor a la hora de llevar a cabo un trabajo; por ejemplo, que requisitos funcionales tengo que satisfacer primero o que pruebas de unidad o de sistema tengo que implementar para saber con certeza que el comportamiento de una funcionalidad es correcta.

En cuanto a las tecnologías nuevas que he conocido durante este periodo, he de referirme a Selenium y Canvas. La primera me ha aportado una forma más de realizar código de evaluación haciendo uso de los navegadores actuales y de esta manera comprobar desde

otro punto de vista el funcionamiento correcto de una nueva aplicación. Por otra parte, el manejo de figuras 2D a través de Canvas en el lado cliente de un servicio web me ha sido de gran utilidad para conocer más a fondo las posibilidades que ofrecen actualmente las aplicaciones web con el uso de HTML5.

En último lugar, quiero destacar el buen ambiente que se tiene en el laboratorio, ya que he recibido la ayuda necesaria en los momentos de duda o incertidumbre ante alguna línea a seguir o estancamiento por falta de alguna documentación en el código fuente. Inclusive en las reuniones realizadas durante este periodo, donde se ha asentado mejor la idea de los comportamientos dentro de un mashup, se han tenido en cuenta mis ideas o soluciones para así conseguir una línea principal que poder seguir. Todo ello, hizo que la evolución y el logro de los objetivos de este Trabajo Fin de Grado tuviera a penas inconvenientes.

4.3 Líneas Futuras

Para concluir, en este apartado, se mencionan las líneas futuras que este sistema, dentro de la plataforma WireCloud, que se llevarían a cabo. Una primera línea sería la mejora de este nuevo concepto, el de los comportamientos de un mashup, hasta el punto de ofrecer una mejor barra de herramientas para que cualquier usuario interesado en la creación de mashups, sin importar el nivel técnico, sea capaz de lograr un mashup innovador y muy documentado internamente, de manera inteligible para él mismo y para otros usuarios.

Siguiendo esta mejora, otra línea tendría que ver con los diseñadores a la hora de identificar bien los comportamientos parciales que contiene un mashup. Esta buena identificación, implica a los diseñadores a realizar una tarea más, pero esta facilita la comprensión posterior de otros usuarios que puedan estar interesados en usar su creación. De este modo, la realización de esta tarea mejoraría la incorporación de nuevos usuarios a la plataforma WireCloud, lo cual es un punto importante a tener en cuenta.

Con respecto a los comportamientos parciales, otra línea a tener en cuenta sería la posibilidad de descargar un comportamiento parcial como un nuevo mashup independiente. En otras palabras, crear un nuevo mashup descargable a partir de los componentes que integran un comportamiento parcial. Esta funcionalidad, permitiría a los usuarios descargarse el comportamiento parcial de un mashup debido a que únicamente es la parte que le interesa, sin tener la necesidad de descargar el mashup completo.

Ahora la vista independiente de un comportamiento parcial solo permite modificaciones visuales, lo cual nos lleva a tener en cuenta otra línea consistente en la mejora de dicha

visualización, permitiendo nuevas funcionalidades como crear, borrar o modificar conexiones o instancias de aplicaciones web. Incluso la posibilidad de realizar una captura de imagen de la visualización de un comportamiento parcial para el uso de la descripción de un mashup o en una presentación.

Finalmente, el nuevo diseño de la barra de herramientas permitirá la agregación de nuevas funcionalidades como deshacer una acción o, disminuir y aumentar el tamaño de los elementos de una vista, de una manera fácil para los desarrolladores. Con el objetivo de que con el paso del tiempo esta barra empiece a parecerse más a un menú completo de edición de contenidos. E incluso asociar a cada función un atajo con el fin de agilizar la ejecución de una serie de acciones de manera rápida.

ANEXOS

1. MANUAL DE USUARIO

1.1 Introducción

En este documento se expone el modo de funcionamiento, detallado en profundidad, para llevar a cabo un uso correcto del sistema web de edición de conexiones orientado a comportamientos que se encuentra integrado en la plataforma WireCloud. Esta plataforma está centrada en la construcción de mashups y se encuentra supervisada, y en continuo desarrollo, por el proyecto europeo FI-WARE. WireCloud proporciona un paradigma de diseño web de mashups dirigido a usuarios finales. Estos, a través de las múltiples herramientas que ofrece, pueden construir un mashup a partir de la interconexión de pequeñas aplicaciones web (hoy día, widgets u operadores).

La herramienta de edición que más destaca en la citada plataforma es este editor web de conexiones. En los siguientes apartados se describen paso a paso todas las funcionalidades que ofrece para que cualquier usuario conozca los pasos suficientes a seguir, con el propósito de que sea capaz de conectar los diferentes puntos de entrada y de salida de que disponen las pequeñas aplicaciones web, además de gestionar los comportamientos parciales que se vayan identificando de manera sencilla.

En primer lugar, debemos conocer más a fondo cuales son las aplicaciones web de las que actualmente WireCloud permite hacer uso en este editor web:

- Widget: es una pequeña aplicación web que trabaja por si sola y puede ser instalado y ejecutado por un usuario final en los entornos de trabajo o mashups de WireCloud. La funcionalidad de esta es diversa, abarcando desde recolectar información de otros servicios web y mostrarlas a través de una interfaz gráfica de usuario hasta servir como una aplicación auxiliar en transacciones de datos.
- Operador: es una pequeña aplicación web similar a un widget web pero con algunas diferencias. Estas diferencias son la ausencia de una interfaz gráfica de usuario y la innecesaria interacción con el usuario final. Normalmente, la funcionalidad principal de este es la transferencia de datos de servicios web externos. En WireCloud este tipo de aplicación web tiene un papel importante en el área de edición del esquema de conexiones de un mashup.

Finalmente, para la explicación detallada de las funcionalidades ofrecidas por esta herramienta de edición web debemos registrarnos e identificarnos en la plataforma WireCloud, además de disponer de un entorno de trabajo con más de un widget en uso y algún que otro operador instalado en la cuenta de usuario actual.

1.2 Interfaz general

En la interfaz general del editor web pueden diferenciarse dos áreas: un área dedica a la barra de herramientas y otra área centrada en la edición visual del esquema de conexiones de un mashup en concreto. Otra área también presente en la interfaz general es la barra de navegación a través de la cual un usuario puede moverse entre las diferentes aplicaciones que actualmente la plataforma ofrece. Como esta última área no es de suma importancia en la explicación del funcionamiento de este editor, en las próximas ilustraciones no se mostrara con la finalidad de aprovechar mejor la ilustración.

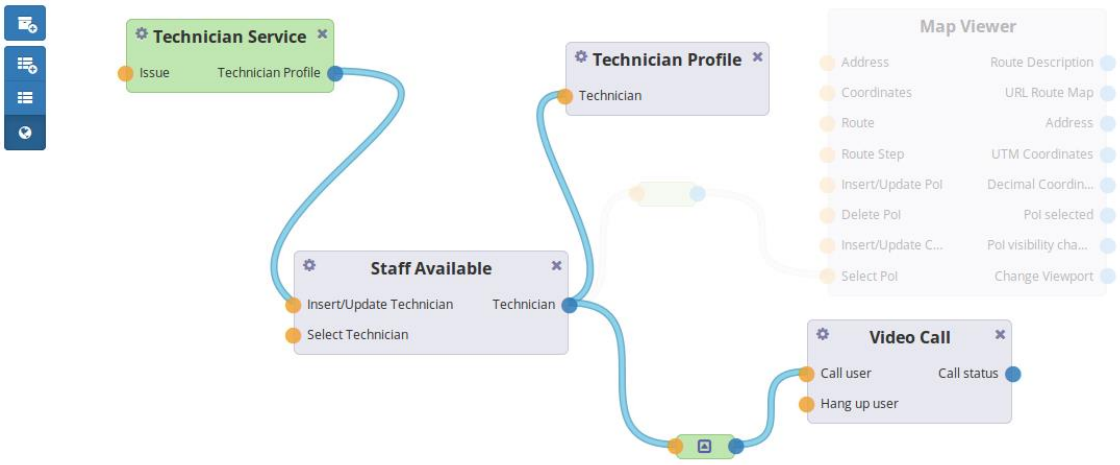

Figura 29: ejemplo de visualización de un mashup por comportamientos

1.2.1 Barra de herramientas

Como puede visualizarse en la ilustración anterior, la barra de herramientas se ubica en el lateral izquierdo de la pantalla y se muestra en todo momento. En esta versión, desde la barra de herramientas un usuario puede visualizar, en forma de lista desordenada, todos los widgets utilizados en el entorno de trabajo actual y todos los operadores instalados en la cuenta de usuario actual. Esto es posible a partir del primer botón en vertical, que despliega un panel deslizante hacia la derecha con las aplicaciones web mencionadas distribuidas en dos pestañas: uno para los widgets y otro para los operadores.

Figura 30: panel de aplicaciones web

Además, un usuario también puede gestionar los diferentes comportamientos parciales identificados durante la edición de un mashup; es decir, puede crear, modificar y borrar cualquier comportamiento parcial según algunas restricciones que más adelante se mencionan. El segundo botón en vertical despliegue una nueva ventana en medio de la pantalla incluyendo un formulario a rellenar, el cual tiene los campos de título y descripción, para la creación de un nuevo comportamiento parcial. Estos campos son opcionales y se completaran con unos valores por defecto en el caso que no sean rellenados. La ventana desplegada también muestra la posibilidad de crear y activar el nuevo comportamiento parcial en cuestión en la misma operación.

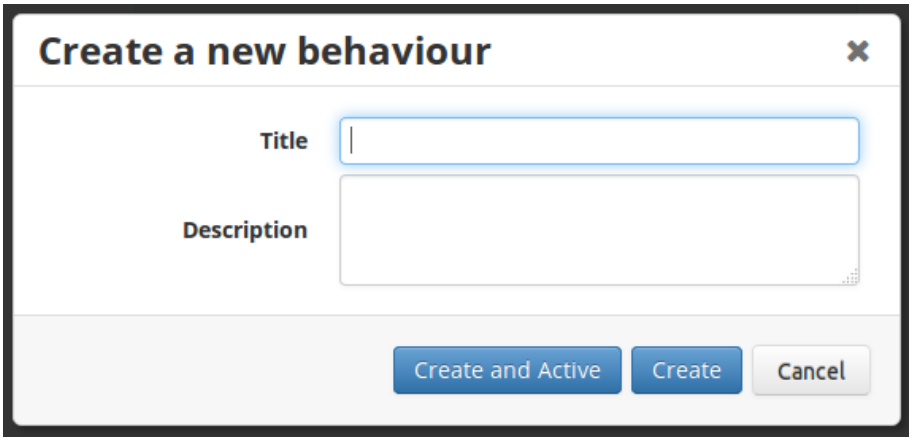

Figura 31: formulario de creación de un comportamiento parcial

Respecto al penúltimo botón en vertical, este despliega un panel deslizante hacia la derecha con los comportamientos parciales identificados por el usuario. Cada comportamiento parcial puede visualizarse como el elemento de una lista en el cual se identifica por un título y una descripción dada por el usuario, además del número de cada tipo de componente utilizado en este. En la parte superior derecha de cada comportamiento parcial también se incluye un botón de preferencias.

Figura 32: panel de comportamientos parciales

El botón de preferencias mencionado antes despliega una ventana en mitad de la pantalla incluyendo un formulario con la información guardada en un comportamiento parcial específico. Esta información, título y descripción, pueden ser modificados por el usuario en todo momento.

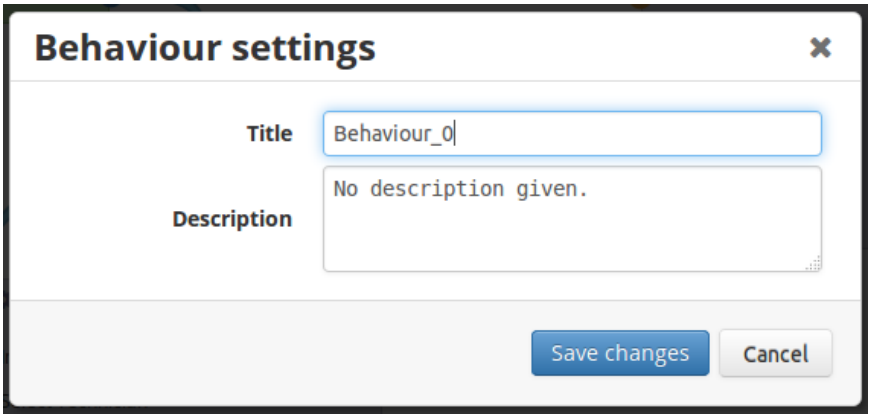

Figura 33: formulario de actualización de un comportamiento parcial

Por último, el botón de debajo de la barra de herramientas permite cambiar la vista actual del esquema de conexiones del mashup. Cuando este botón está activado muestra la vista

global del esquema de conexiones frente al comportamiento parcial activo en ese momento. En caso contrario, se muestra la vista independiente del esquema de conexiones perteneciente al comportamiento parcial activo. Por defecto, este botón está activo cada vez que accedes al sistema.

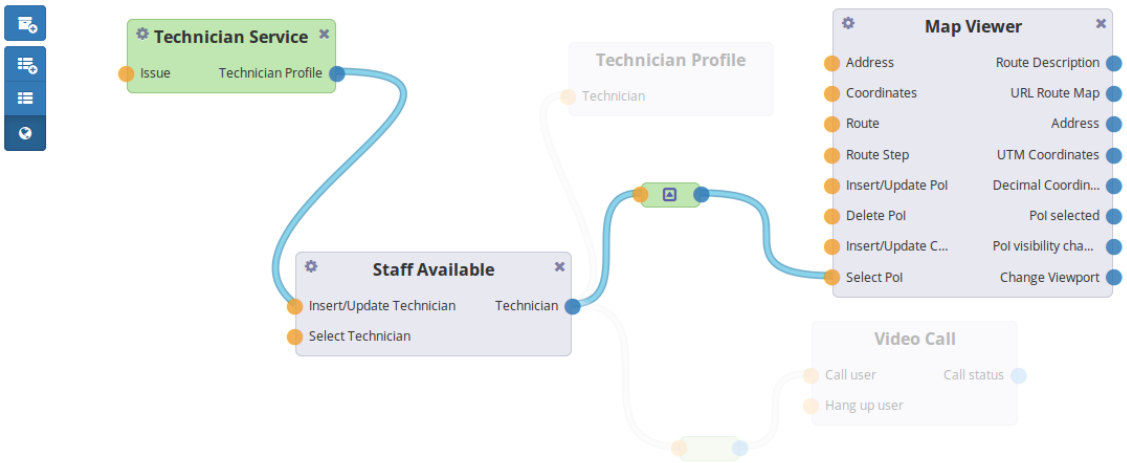

Figura 34: vista global frente al comportamiento parcial activo

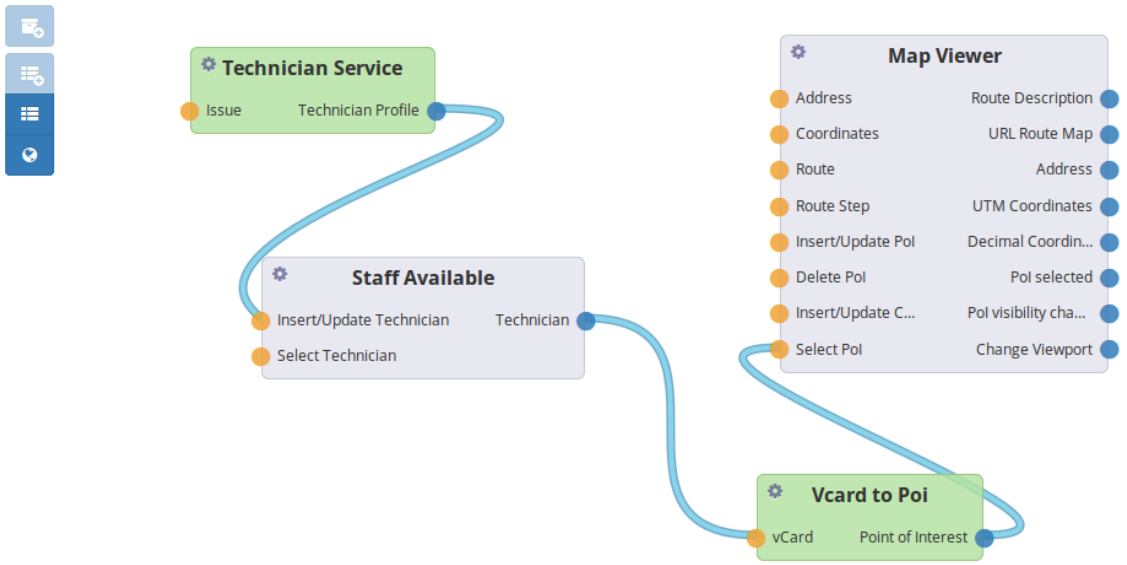

Figura 35: vista independiente del comportamiento activo

1.2.2 Área de visualización del esquema de conexiones

Para finalizar este apartado, se detalla la especificación del área centrada a la edición visual del esquema de conexiones de un mashup. En esta área, un usuario puede visualizar

los componentes interconectados de un comportamiento parcial según las dos vistas explicadas anteriormente. Aun así, el objetivo principal de esta área es permitir al usuario utilizar aplicaciones web, conectarlas y distribuirlas en los diferentes comportamientos parciales que este identifique.

1.3 Especificación de las aplicaciones web

Como se ha mencionado en un principio, las aplicaciones web accesibles desde el editor pueden visualizarse a través de un panel destinado a clasificar en dos listas, los widgets y los operadores disponibles. Cada aplicación web se identifica por el título dado por el desarrollador y pueda arrastrarse de fácil manera al área de edición, en el cual cambiará su aspecto visual. Este nuevo aspecto visual a partir de ahora lo llamaremos instancia de una aplicación web.

1.3.1 Instancia de una aplicación web

La instancia de una aplicación web es la representación visual de las preferencias y los diferentes puntos de entrada y de salida que esta dispone. En todas las instancias creadas pueden distinguirse dos zonas: la parte superior está dedicada al menú desplegable de opciones, al botón de borrado y al título asignado; y la parte central distribuye los puntos de entrada en el lateral izquierdo y los puntos de salida en el lateral derecho. Cada punto de entrada o de salida está identificado por un título e internamente por una descripción que el sistema de recomendaciones hace uso; y contiene en un extremo una especie de ancla con el cual un usuario puede crear conexiones.

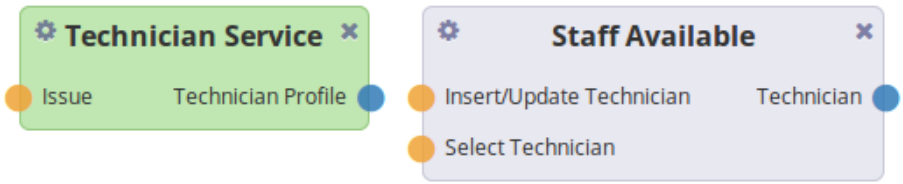

Figura 36: instancias de aplicaciones web

Se puede observar que la instancia de un widget y la de un operador se distinguen fácilmente dentro del área de edición. Además, también difieren en las opciones que el menú de configuración despliega.

1.4 Sistema de recomendaciones

Este editor hace uso de un sistema de recomendaciones basado en el casado de palabras entre las descripciones dadas en cada uno de los puntos de entrada y de salida que

contienen las instancias de aplicaciones web. En otras palabras, este sistema busca entre las descripciones de los puntos de entrada y de salida disponibles que dos palabras casen al menos. Esta búsqueda se realiza siempre en los puntos opuestos; es decir, en el caso que el usuario quiera conectar el punto de entrada de una instancia de aplicación web, este sistema buscará en los puntos de salida pertenecientes al resto de instancias disponibles y los puntos de salida resultantes a esta búsqueda quedaran resaltadas para que el usuario pueda visualizar estos puntos de manera rápida.

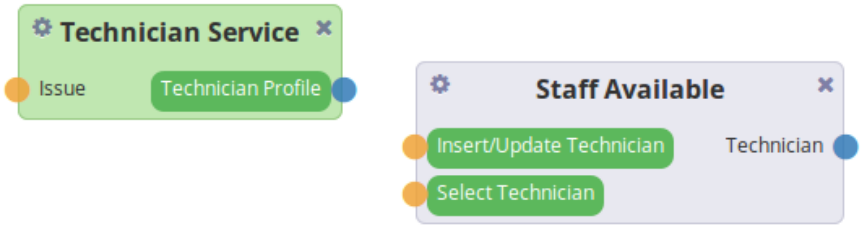

Figura 37: recomendaciones mostradas por el sistema

Por cierto, este sistema no tiene en cuenta si una instancia pertenece o no al comportamiento activo en la vista global del esquema de conexiones frente a este comportamiento. Además, estas recomendaciones resaltadas son obtenidas de las descripciones escritas por los desarrolladores lo cual no indica la conexión entre dos puntos resaltados puedan funcionar correctamente (en estas ocasiones en cuando tienen importancia el uso de operadores).

1.5 Vista global frente a un comportamiento parcial

En este apartado se mencionan las funcionalidades accesibles desde la vista global del esquema de conexiones frente al comportamiento parcial activo. Para tener visible esta vista debe estar activo el último botón en vertical de la barra de herramientas. Estas funcionalidades combinan una serie de acciones en las que entran en juego tanto la barra de herramientas como el área de edición del esquema de conexiones.

1.5.1 Crear la instancia de una aplicación web

1. Abrimos el panel dedicado a las aplicaciones web pulsando el primer botón en vertical de la barra de herramientas ubicada en el lateral izquierdo de la pantalla.
2. Seleccionamos el tipo de aplicación web, widget u operador, para mostrar las aplicaciones web accesibles desde el entorno de trabajo actual en este instante.

3. Buscamos la aplicación web, identificada por el título asignado, que deseamos utilizar en el área de edición. Esta aplicación web debe estar habilitada para continuar con los siguientes pasos.

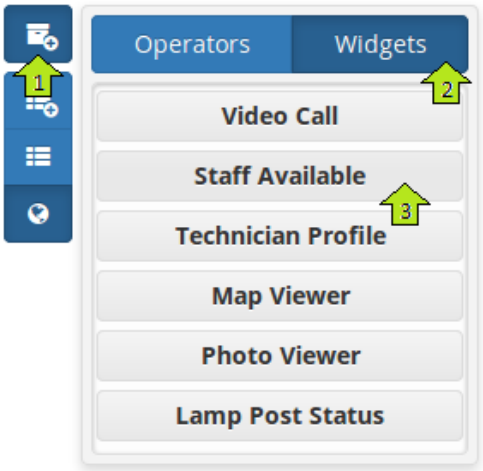

Figura 38: selección de una aplicación web

4. Seleccionamos la aplicación web buscada con el botón izquierdo del ratón y sin soltar dicha tecla, deslizamos el elemento hasta la posición, dentro del área de edición, que se desee. La posición indicada puede ser modificada en cualquier momento una vez dentro del área de edición.

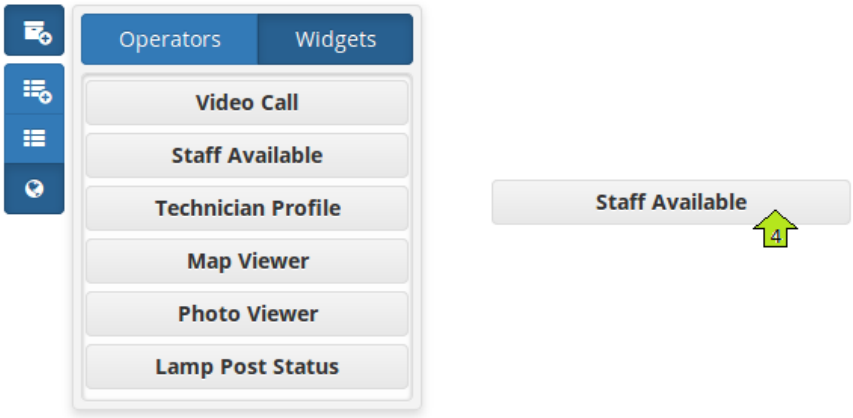

Figura 39: paso intermedio en la creación de una instancia

5. Una vez dentro del área de edición, soltamos el botón izquierdo del ratón creando de esta manera, una nueva instancia de la aplicación web buscada. Esta nueva

instancia estará disponible en el área de edición y se visualizará como en la ilustración de abajo.

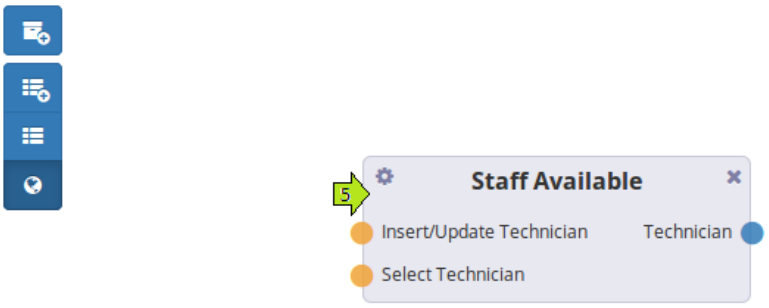

Figura 40: resultado de la creación de una instancia

Cualquier nueva instancia de una aplicación web creada desde la barra de herramientas formará parte del comportamiento parcial que esté activado en ese momento. Por defecto, los widgets solo pueden tener una instancia presente en el área de edición. Como consecuencia de esto, el widget instanciado estará inhabilitado en el panel dedicado a las aplicaciones web. Sin embargo, los operadores pueden tener más de una instancia disponible en el área de edición.

1.5.2 Modificar la posición actual de una aplicación web instanciada

1. Seleccionamos con el botón izquierdo del ratón la aplicación web instanciada, presente en el área de edición, que deseamos movilizar a una nueva posición.

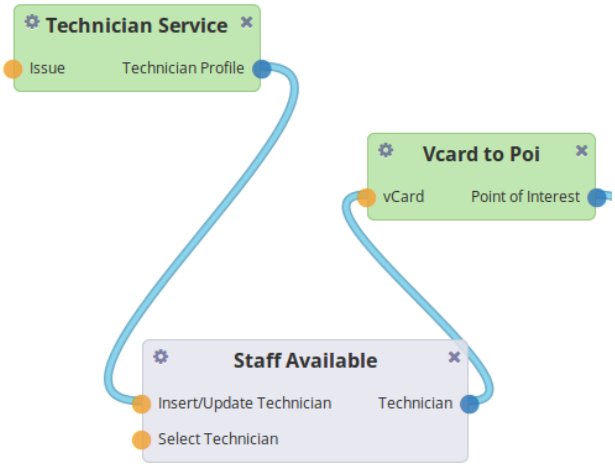

Figura 41: selección de una instancia a mover

2. Una vez seleccionada, sin soltar dicho botón, movemos la aplicación web instanciada a la nueva posición dentro de los límites del área de edición.

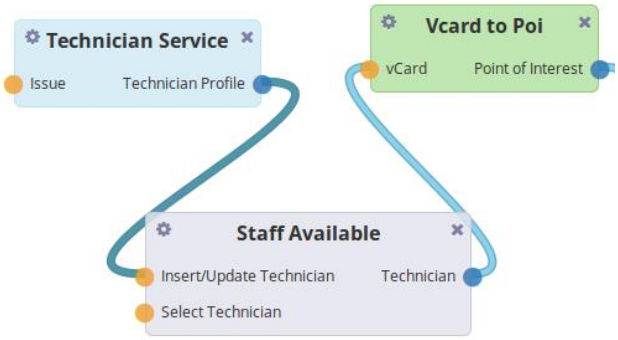

Figura 42: paso intermedio en la movilización de una instancia

3. Finalmente, soltamos el botón izquierdo del ratón para actualizar la nueva posición de la aplicación web instanciada.

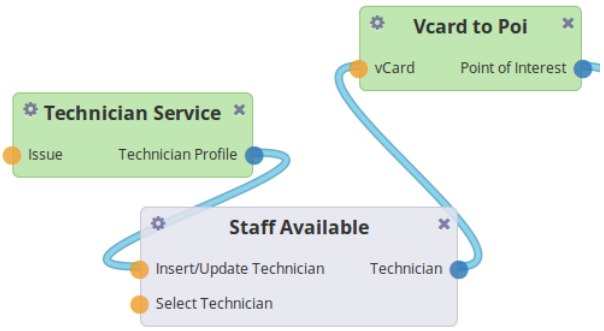

Figura 43: nueva posición de una instancia

Esta operación puede realizarse tanto en widgets como en operadores. Como se puede observar en las ilustraciones anteriores, las conexiones entrantes y salientes de la instancia de una aplicación web también son movilizadas.

También se permite la selección múltiple en este editor. Para ello debemos incluir una acción previa al paso 2:

- Si se desea realizar una selección múltiple, la selección de las aplicaciones web instanciadas que deseamos movilizar a una nueva posición debe realizarse por medio de una combinación de teclas (CTRL + botón izquierdo del ratón) a la hora de seleccionar una.

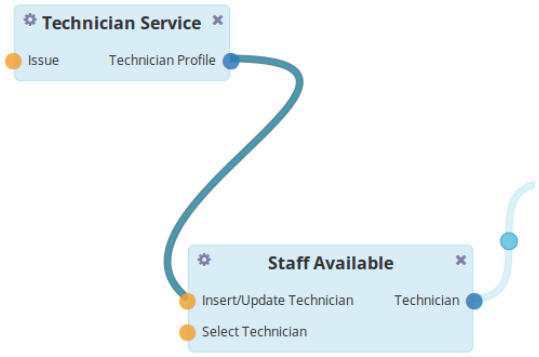

Figura 44: selección simultanea de dos instancias

1.5.3 Modificar el orden de los puntos de una aplicación web instanciada

1. Seleccionamos el botón de configuración, situado en la esquina superior izquierda, de la aplicación web instanciada que deseamos modificar el orden de los puntos de entrada o de salida que esta dispone.

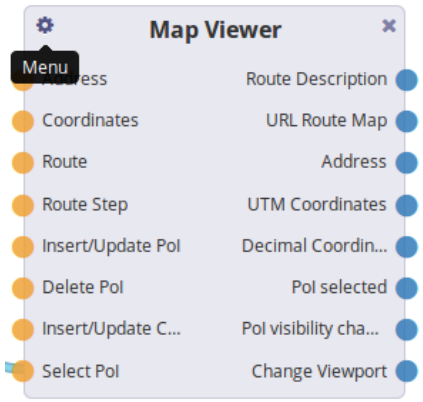

Figura 45: ubicación del botón de configuración

2. En el menú desplegable, seleccionamos la opción de *reordenar los puntos finales.* Esta opción permitirá modificar la posición de cualquier punto de entrada o de salida visible, siempre que dicha modificación no sea fuera de la columna del tipo de punto al que pertenece.

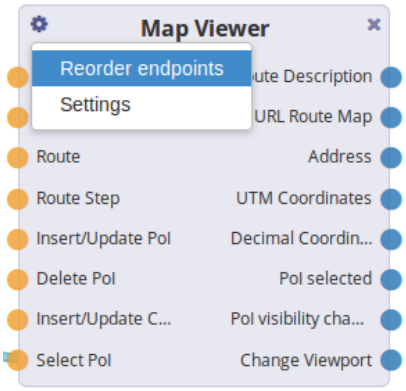

Figura 46: opción de reordenar

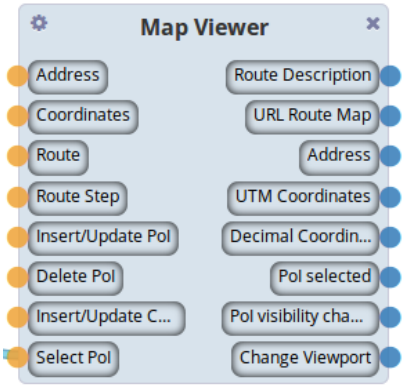

Figura 47: visualización para reordenamiento de entradas o salidas

3. Seleccionamos con botón izquierdo del ratón el punto de entrada o de salida que deseamos modificar de posición.

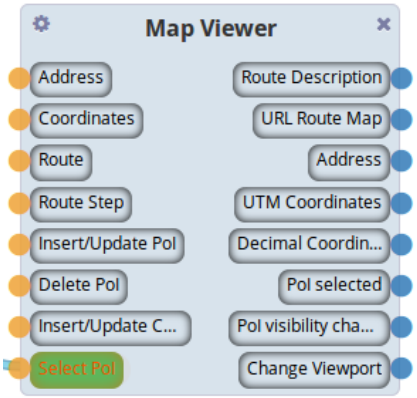

Figura 48: selección de un punto de entrada

4. Ahora, sin soltar este botón, desplazamos el punto de entrada o de salida seleccionado a la nueva posición dentro de la columna del mismo tipo de punto.

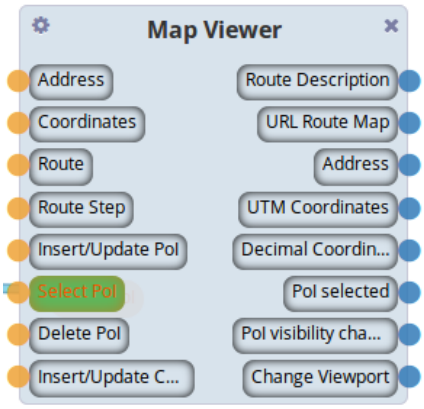

Figura 49: nueva posición del punto de entrada seleccionado

5. Soltamos el botón izquierdo del ratón para indicar el nuevo orden de los puntos de entrada o de salida de la aplicación web instanciada.

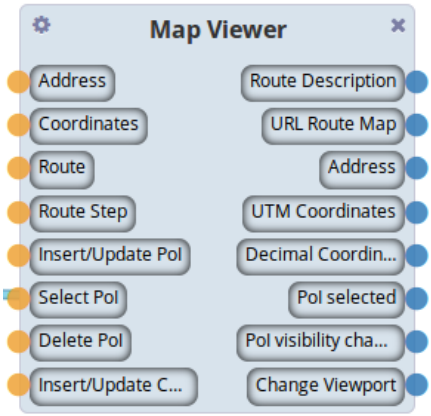

Figura 50: nuevo estado de las listas de entradas y salidas

6. Finalmente, seleccionamos fuera de la aplicación web instanciada para actualizar el nuevo orden de los puntos de entrada o de salida y finalizar la edición de la posición de estos.

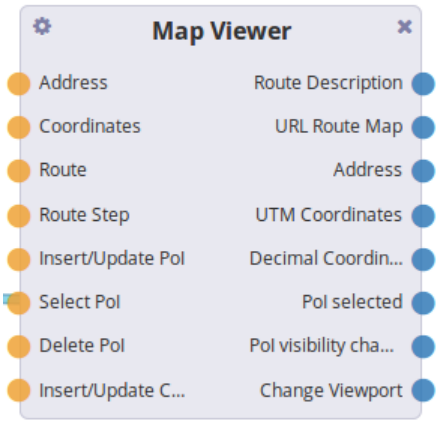

Figura 51: reordenamiento finalizado

Esta operación solo puede realizarse en las aplicaciones web instanciadas que pertenecen al comportamiento parcial activo en ese momento.

1.5.4 Modificar el tamaño de un operador instanciado

1. Seleccionamos el botón de configuración, situado en la esquina superior izquierda, del operador instanciado que deseamos minimizar.

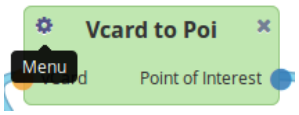

Figura 52: localización del botón de configuración

2. Finalmente, desde el menú despegable seleccionamos la opción de *minimizar* para visualizar una nueva forma más reducida en todos los aspectos.

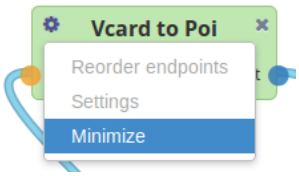

Figura 53: selección de la opción de minimizar

En el caso de que se desee restaurar el tamaño de un operador instanciado que se encuentra actualmente minimizado, debemos seleccionar la opción de *restaurar tamaño* situada en el centro de la figura y representada por un icono intuitivo.

Figura 54: visualización de una instancia minimizada

1.5.5 Modificar la información básica de un comportamiento parcial

1. Seleccionamos el penúltimo botón en vertical de la barra de herramientas para deslizar hacia la derecha el panel dedicado a los comportamientos parciales identificados por el usuario en el mashup en cuestión.
2. Dentro de este panel, podemos encontrar una lista ordenada por fecha de creación de los diferentes comportamientos parciales. En dicha lista, seleccionamos el botón de preferencias, situado en la esquina superior derecha, del comportamiento parcial que deseamos modificar su información básica.

Figura 55: selección del botón de preferencias

3. Ahora podemos visualizar una nueva ventana, centrada en mitad de la pantalla, que proporciona un formulario con la información actualmente guardada del comportamiento parcial seleccionado.

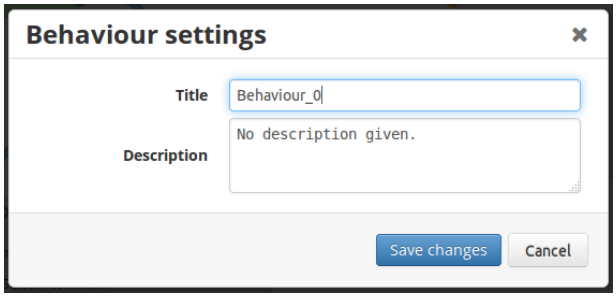

Figura 56: formulario de actualización del comportamiento parcial

4. Modificamos la información de los campos específicos de este formulario.

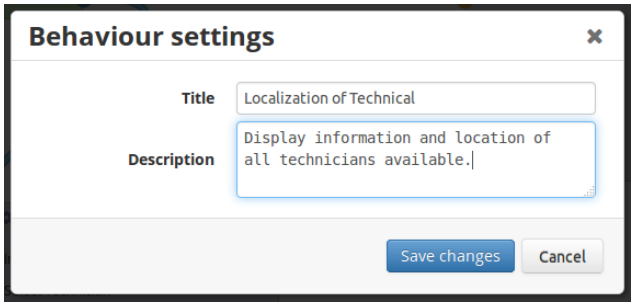

Figura 57: introducción de nuevos datos

5. Finalmente, seleccionamos el botón de *guardar cambios*, situado en la parte inferior de la ventana, para actualizar la información guardada en el comportamiento parcial seleccionado.

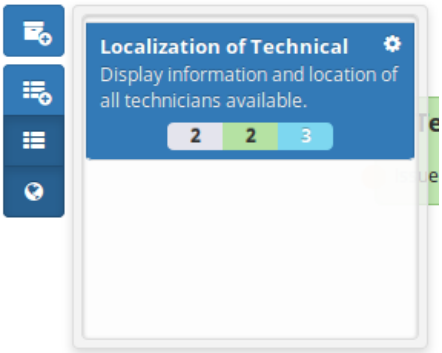

Figura 58: actulización completada de la información básica

1.5.6 Crear un comportamiento parcial

1. Seleccionamos el segundo botón en vertical de la barra de herramientas para visualizar una nueva ventana centrada en mitad de la pantalla el cual proporciona un formulario para crear un nuevo comportamiento parcial.

Figura 59: fomulario de creación de un comportamiento parcial

2. Rellenamos los campos del formulario dado. En el caso de que el usuario todavía no sepa el objetivo de este nuevo comportamiento parcial, el sistema permite dejar estos campos vacíos.

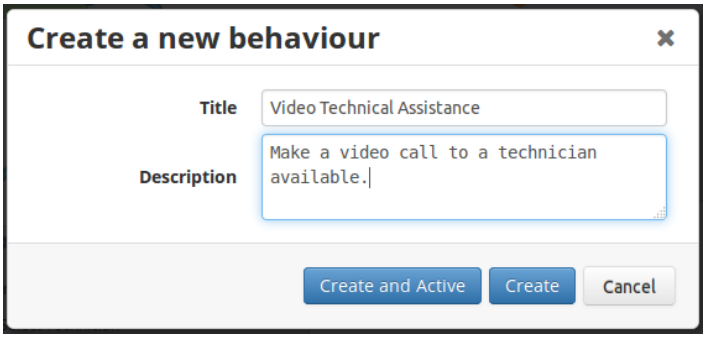

Figura 60: introducción de datos en el formulario

3. Finalmente, seleccionamos la opción de *crear* ubicada en la parte inferior de esta ventana para crear un nuevo comportamiento parcial. Si deseamos utilizar el nuevo comportamiento parcial en el mismo instante que se crea, debemos seleccionar la opción de *crear y activar* ubicada también en la parte inferior de la ventana.

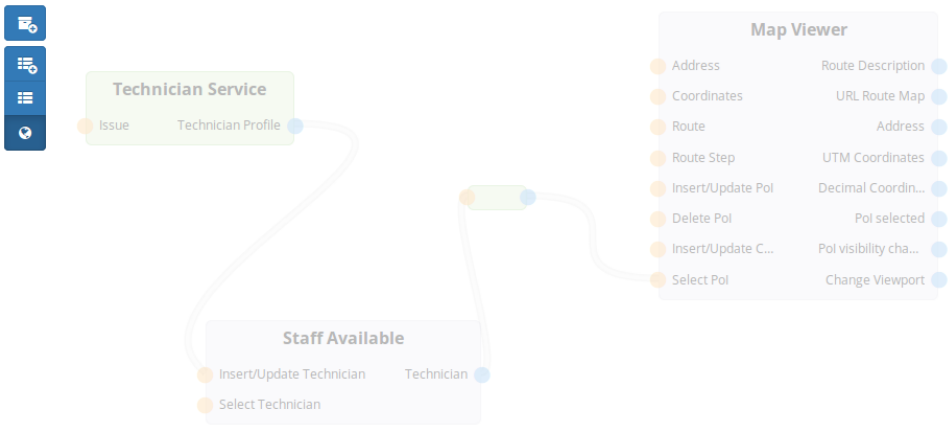

Figura 61: visualización del nuevo comportamiento parcial

1.5.7 Activar un comportamiento parcial

1. Seleccionamos el penúltimo botón en vertical de la barra de herramientas para deslizar hacia la derecha el panel dedicado a los comportamientos parciales identificados por el usuario.

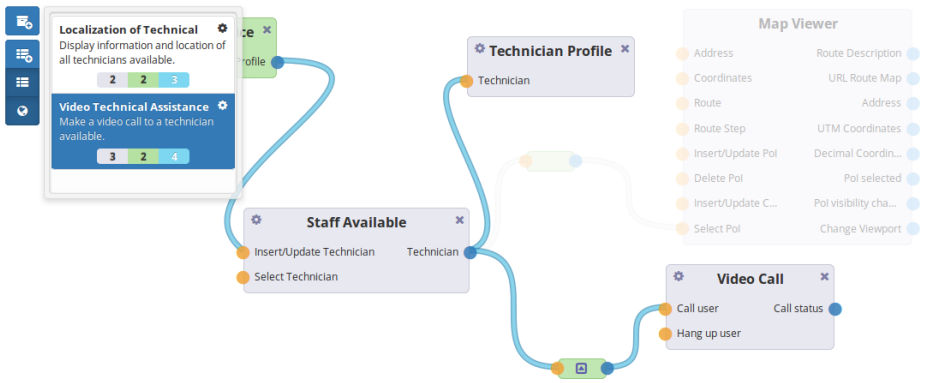

Figura 62: búsqueda del comportamiento parcial a activar

2. En este panel deslizante, podemos observar los diferentes comportamientos parciales en una lista ordenada por fecha de creación. Finalmente, seleccionamos el comportamiento parcial que deseamos activar.

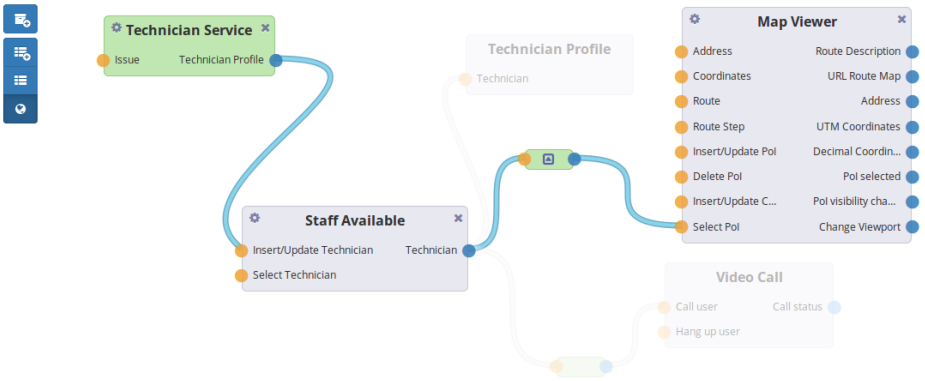

Figura 63: nuevo comportamiento parcial activo

1.5.8 Compartir la instancia de una aplicación web

1. Seleccionamos la aplicación web instanciada visible en el fondo que deseamos compartir al comportamiento parcial activo para así, mostrar la opción de *añadir*. Esta opción esta oculta para no entorpecer la visualización del comportamiento parcial actualmente activo. También, podemos deslizar el cursor por encima de la instancia a compartir para mostrar esta opción.

Figura 64: visualización de una instancia en el fondo

2. Finalmente, seleccionamos dicha opción para compartir la instancia deseada también en el comportamiento parcial activo. De tal manera, que ahora esta instancia mostrará las opciones permitidas desde un comportamiento parcial activo.

Figura 65: localización del botón de añadir

Figura 66: nueva intancia en el comportamiento activo

1.5.9 Crear una conexión

1. Seleccionamos el punto de entrada de la aplicación web instanciada, con el botón izquierdo del ratón, que deseamos que sea el punto izquierdo extremo de la nueva conexión. Si esta aplicación web instanciada esta visible en el fondo, como consecuencia de la creación de la nueva conexión, esta será compartida también con el comportamiento parcial activo en ese momento.

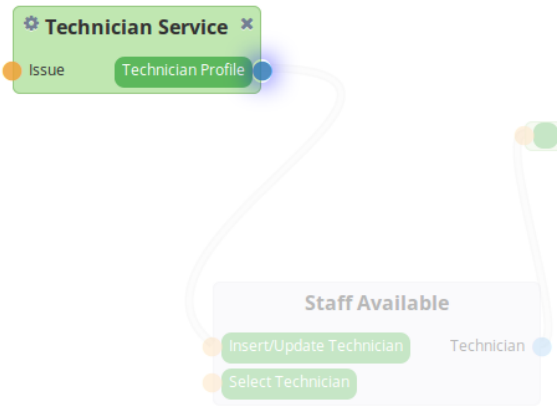

Figura 67: selección del punto extremo izquierdo de una conexión

2. Una vez seleccionado el punto de entrada, sin soltar dicho botón, deslizamos el cursor para crear la forma visual de una futura conexión.

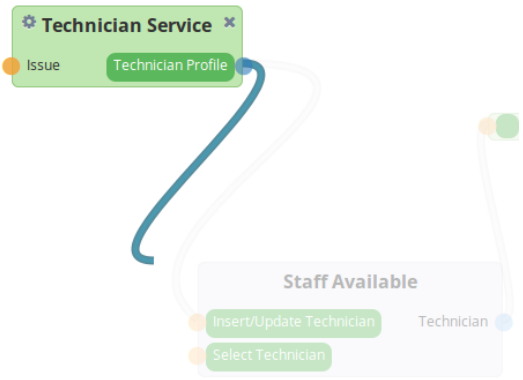

Figura 68: paso intermedio en la creación de una conexión

3. Finalmente, soltamos el botón izquierdo del ratón sobre el punto de salida de la otra aplicación web instanciada que deseamos que sea el punto extremo derecho de la nueva conexión. Si esta aplicación web instanciada esta visible en el fondo, formara también parte del comportamiento parcial activo. De este modo, creamos una nueva conexión entre dos aplicaciones web.

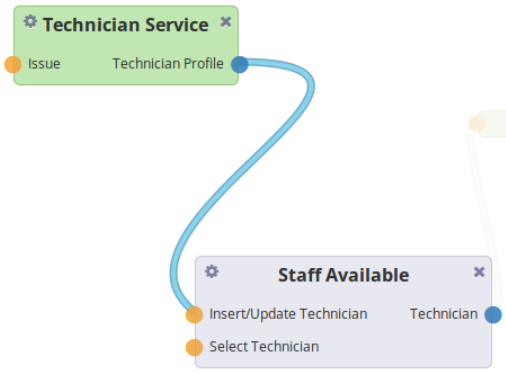

Figura 69: conexión creada

Por defecto, el sistema no permite la duplicidad de conexiones existentes en el mismo comportamiento parcial.

1.5.10 Compartir una conexión

1. Seleccionamos la conexión visible en el fondo que deseamos compartir al comportamiento parcial activo para mostrar la opción de *añadir* ubicada en la mitad de la conexión. Esta opción tiene forma de círculo azul. También, podemos visualizar esta opción si deslizamos el cursor sobre dicha conexión.

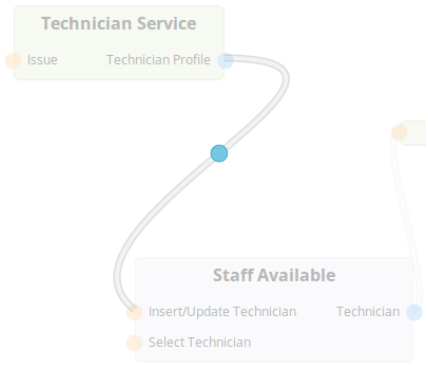

Figura 70: visualización de una conexión en el fondo

2. Finalmente, seleccionamos la opción de añadir para que la conexión visible en el fondo forme parte del comportamiento parcial activo en ese momento. Por defecto, si alguno o ambos extremos de la conexión también estaba visible en el fondo, formaran parte del comportamiento parcial activo.

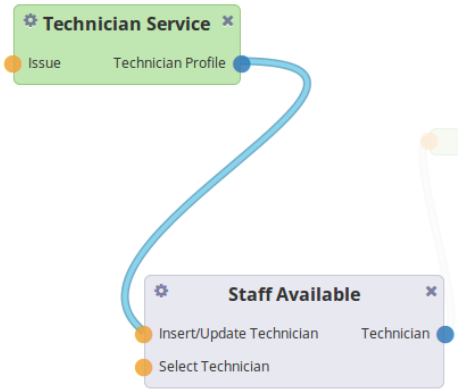

Figura 71: nueva conexión en el comportamiento activo

El sistema no permite compartir una conexión entre más de un comportamiento parcial sin que ambos extremos de esta pertenezcan al nuevo comportamiento parcial. Por esta razón, cuando uno o ambos extremos de una nueva conexión añadida desde el fondo también están visibles en el fondo, son añadidos al comportamiento parcial activo en ese instante en la misma operación.

1.5.11 Modificar la curvatura de una conexión

1. Seleccionamos la conexión, perteneciente al comportamiento parcial activo, que deseamos modificar su curvatura para visualizar la opción de *borrar* y los

tiradores que modifican la curvatura de esta. Estos tiradores se encuentran ubicados en los extremos de la conexión.

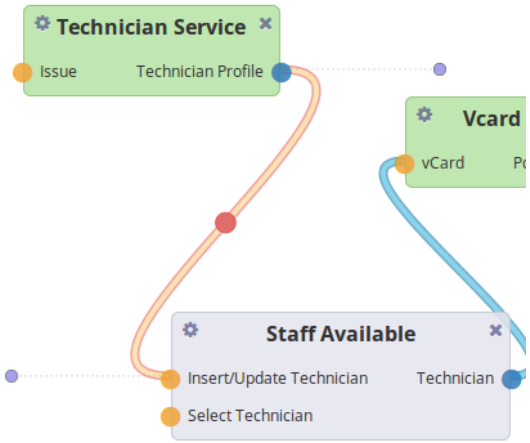

Figura 72: estado de edición de una conexión

2. Deslizamos en la dirección que deseamos de uno o ambos tiradores hasta conseguir la curvatura buscada.

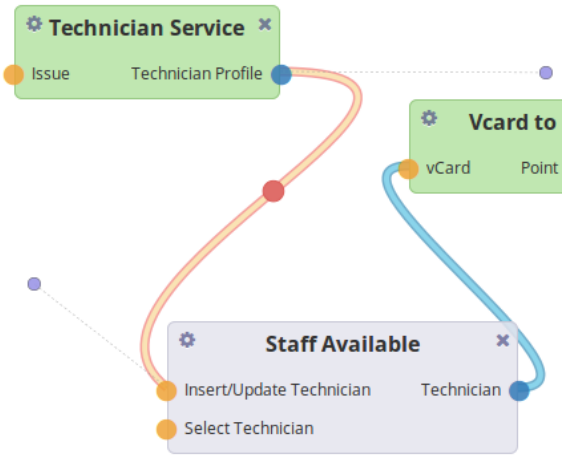

Figura 73: modificación de la curvatura de una conexión

3. Finalmente, seleccionamos fuera de esta conexión para salvaguardar los cambios realizados en esta.

1.5.12 Modificar el punto extremo de una conexión

1. Seleccionamos la conexión, perteneciente al comportamiento parcial activo, que deseamos modificar alguno de los puntos extremos.

2. Ahora, seleccionamos con el botón izquierdo del ratón uno de los puntos extremos de la conexión seleccionada.
3. A continuación, deslizamos el cursor sin soltar dicho botón pulsado hacia otro punto del mismo tipo. Este nuevo punto de entrada o de salida puede pertenecer o no al comportamiento parcial activo en ese instante.
4. Finalmente, soltamos el botón izquierdo del ratón sobre el nuevo punto extremo de la conexión seleccionada. De este modo, la conexión anterior se borrará y se creara una nueva con el nuevo punto de entrada o de salida señalado.

1.5.13 Borrar la instancia de una aplicación web

1. Seleccionamos el botón de borrar ubicado en la esquina superior derecha de la instancia de una aplicación web que deseamos borrar. Este operación solo aparece en las aplicaciones web instanciadas que pertenecen al comportamiento parcial activo en ese momento.

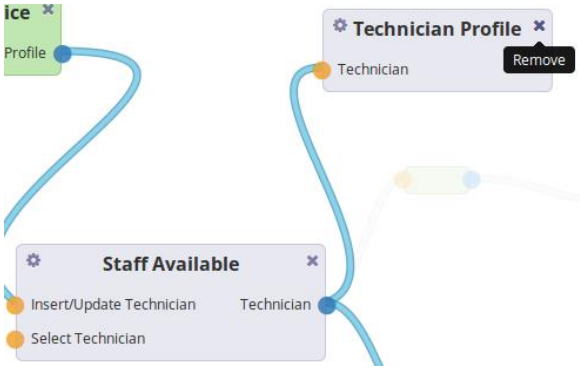

Figura 74: localización del botón de borrar

2. Después del paso anterior, la aplicación web instanciada será borrada del comportamiento parcial activo incluido las conexiones entrantes y salientes de esta. En el caso de que dicha instancia pertenezca también a otros comportamientos parciales, se visualizará una ventana que permite el borrado en cascada.

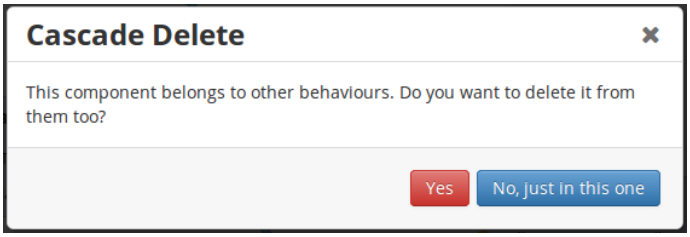

Figura 75: posibilidad de borrado en cascada

3. Finalmente, si seleccionamos la opción de borrar en cascada la aplicación web instanciada, esta será borrada de todos los comportamientos parciales incluyendo todas las conexiones entrantes y salientes de esta también.

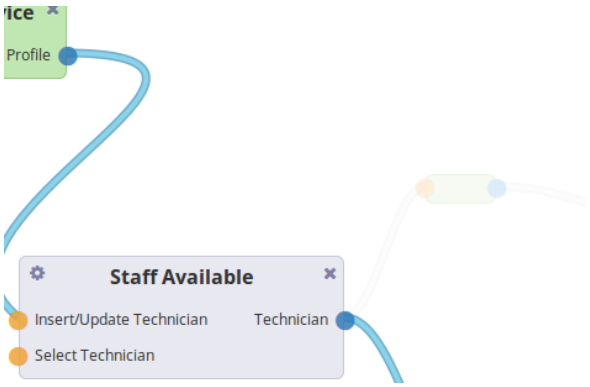

Figura 76: borrado completo de una instancia

En el caso de que la aplicación web instanciada era de tipo widget, y tras la operación de borrado esta ya no está disponible en ninguna de los comportamientos parciales, podemos observar en el panel centrado a las aplicaciones web que esta se encuentra habilitada.

1.5.14 Borrar una conexión

1. Seleccionamos la conexión, que deseamos borrar, para visualizar la opción de borrar la cual se encuentra ubicada en el centro de la conexión como un círculo rojo. Esta conexión debe pertenecer al comportamiento parcial activo.

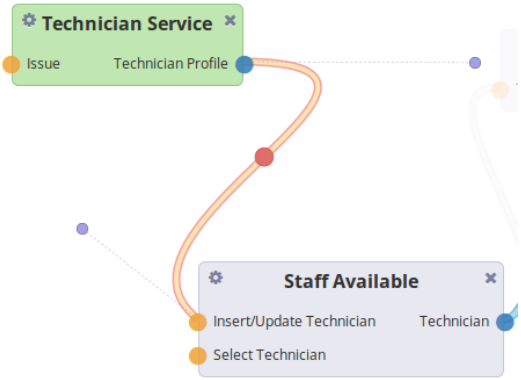

Figura 77: estado de edición de una conexion

2. Finalmente, seleccionamos dicha opción la cual borra la conexión únicamente del comportamiento parcial activo en ese momento. En el caso de que esta también pertenecía a otro comportamiento parcial, esta estará visible en el fondo.

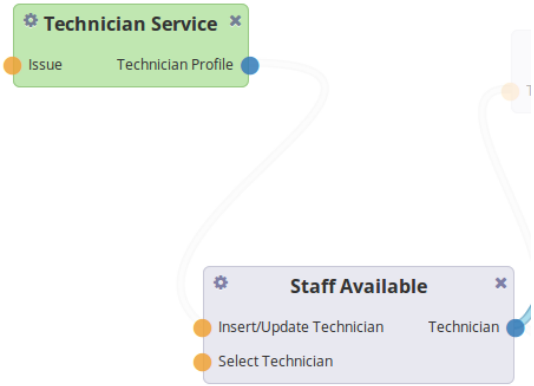

Figura 78: conexión borrada y visible en el fondo

1.5.15 Borrar un comportamiento parcial

1. En primer lugar, seleccionamos el penúltimo botón en vertical de la barra de herramientas. Con esto, visualizaremos el panel dedicado a los comportamientos parciales identificados por el usuario.

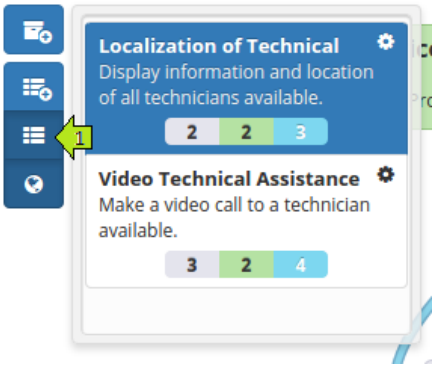

Figura 79: despliegue del panel de comportamientos parciales

2. Seleccionamos el botón de preferencias ubicado en la esquina superior derecha de la representación del comportamiento parcial que deseamos borrar. Este botón despliega una ventana, ubicada en el centro de la pantalla, que incluye un formulario con la información guardada.

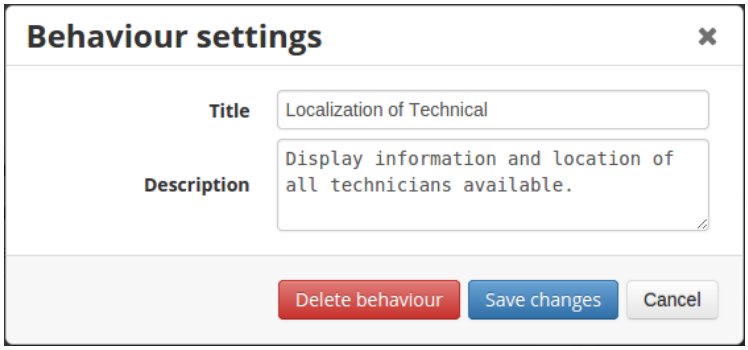

Figura 80: visualización del botón de borrar un comportamiento parcial

3. En la parte inferior de esta nueva ventana, podemos visualizar la opción de *borrar el comportamiento parcial* seleccionado. Seleccionamos dicha opción para borrar el comportamiento parcial que deseamos. Tras esta operación, todos los componentes (aplicaciones web instanciadas y conexiones) pertenecientes a dicho comportamiento parcial también serán borrados.

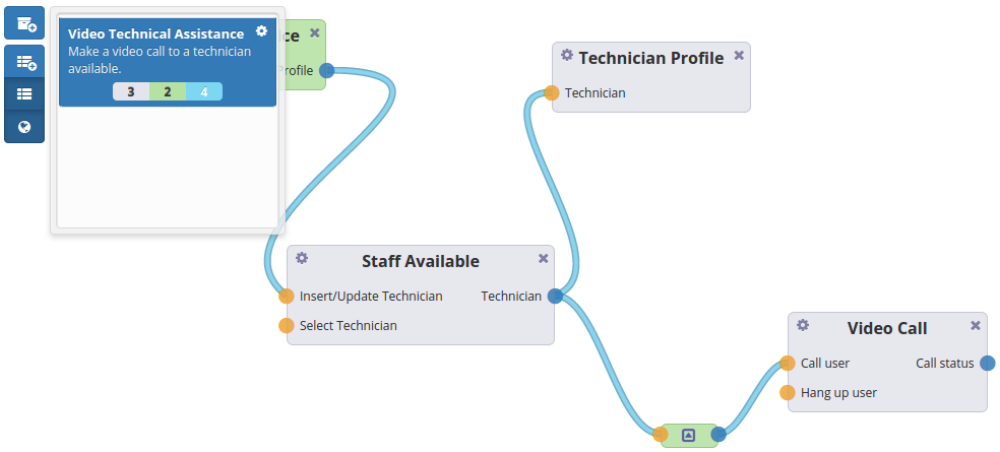

Figura 81: vista posterior al borrado del comportamiento parcial

Por defecto, el sistema no muestra la opción de borrar un comportamiento parcial específico cuando este es el último de la lista. También se debe mencionar, que el comportamiento parcial que se desea borrar estaba actualmente activo, el sistema activará el primer comportamiento parcial de la nueva lista de estos.

1.5.16 Visualizar previamente un comportamiento parcial

1. Primeramente, seleccionamos el penúltimo botón en vertical de la barra de herramientas el cual despliega un panel deslizante hacia la derecha dedico a los comportamientos parciales identificados por el usuario.

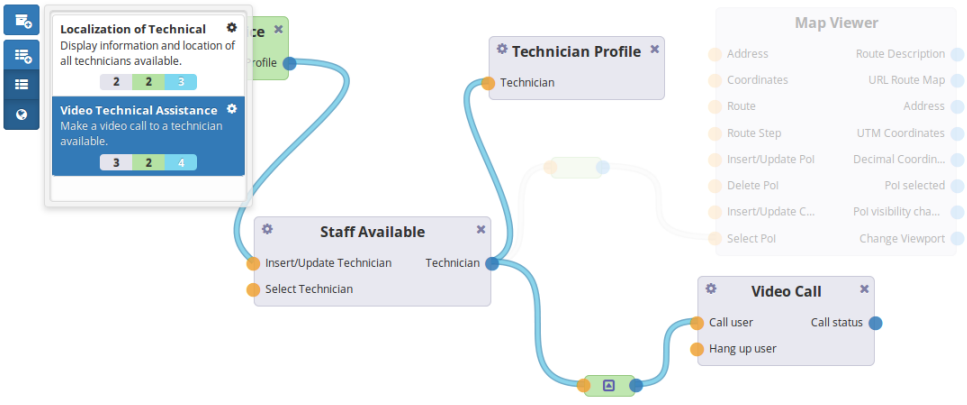

Figura 82: apertura del panel de comportamientos parciales

2. Finalmente, deslizamos el cursor sobre el comportamiento parcial, no activo en ese momento, que deseamos visualizar sin la necesidad de activarlo. Tras esta operación, podremos observar que los componentes pertenecientes a dicho

comportamiento parcial pasan a primer plano y el resto se visualizan en el fondo por un corto periodo.

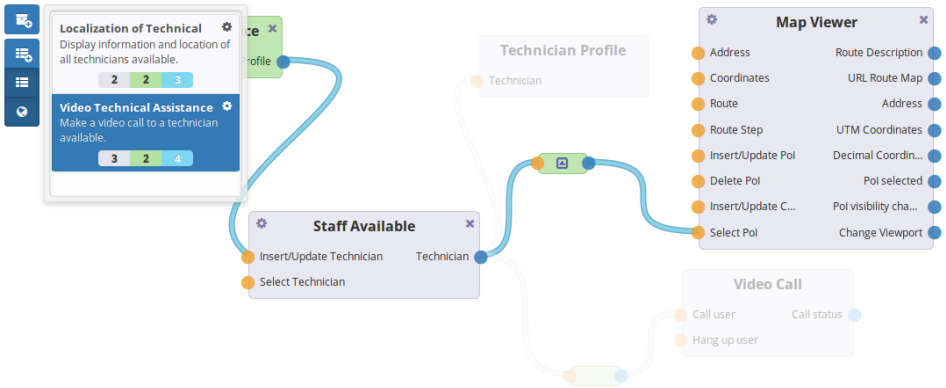

Figura 83: visualización previa de un comportamiento parcial

1.6 Vista independiente de un comportamiento parcial

En esta versión de este editor web, la lista de funcionalidades accesibles desde la vista independiente del esquema de conexiones de un comportamiento parcial es corta y limitada. Esta razón es debida a que de momento solo se utiliza para una mejor visualización o compresión de un comportamiento parcial dado. Para activar esta vista, el último botón en vertical de la barra de herramientas debe estar desactivado. A continuación, se mencionan las funcionalidades que están disponibles en esta vista:

- Modificación de la posición actual de una o más aplicaciones web instanciadas.
- Modificación del tamaño de cualquier operador instanciado.
- Modificación del orden de los puntos de entrada y de salida de cualquier aplicación web instanciada.
- Modificación de la curvatura de cualquier conexión existente.
- Modificación de la información básica de un comportamiento parcial
- Activación de un comportamiento parcial

Las modificaciones que provocan estas funcionalidades solo afectan al comportamiento parcial activo en ese instante; es decir, que al resto de comportamientos parciales identificados y al propio comportamiento global del mashup no se ven alterados.

Por último, si es la primera vez que un comportamiento parcial se visualiza desde esta vista, las últimas modificaciones realizadas en el comportamiento global del mashup serán las que se cargaran.

Figura 84: visualización de un comportamiento parcial en la vista global

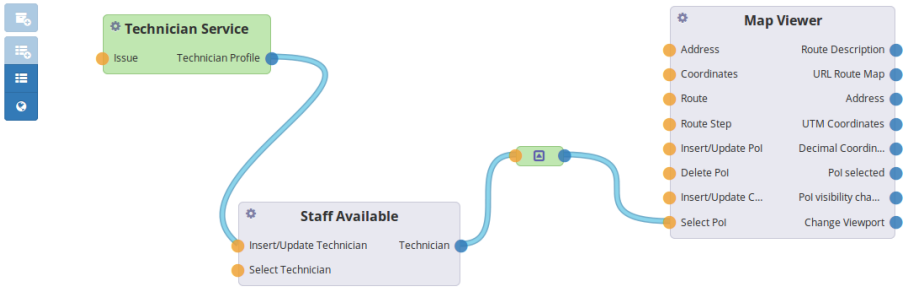

Figura 85: primera visualización del comportamiento parcial en la vista independiente

2. CÓDIGO FUENTE

2.1 Plataforma GitHub

Al tratarse de un desarrollo de tipo OpenSource el código fuente de todo el trabajo se encuentra disponible en la plataforma de alojamiento conocida como GitHub. El repositorio remoto en el cual se encuentra ubicado este trabajo, dentro esta plataforma, tiene un perfil de acceso público, por lo que a través del enlace al proyecto WireCloud del que forma parte cualquiera puede hacer uso de él (atendiendo a las cláusulas de la Licencia Pública General de Affero mencionada anteriormente en el Capítulo 1).

BIBLIOGRAFÍA

[1] Concepto del movimiento web 2.0 [Online]. Disponible en este enlace.

[2] Documentación de un mashup web [Online]. Disponible en este enlace.

[3] Explicación de un mashup web con ejemplos [Online]. Disponible en este enlace.

[4] Página oficial de la plataforma WireCloud [Online]. Disponible en este enlace.

[5] Página oficial del lenguaje Python [Online]. Disponible en este enlace.

[6] Objetivos de la Python Software Foundation [Online]. Disponible en este enlace.

[7] Documentación de la GNU GPL [Online]. Disponible en este enlace.

[8] Página oficial del entorno web Django [Online]. Disponible en este enlace.

[9] Documentación de la licencia BSD 3-Clause [Online]. Disponible en este enlace.

[10] Concepto del diseño Model-View-Controller [Online]. Disponible en este enlace.

[11] Tutorial de uso del lenguaje HTML [Online]. Disponible en este enlace.

[12] Definición del concepto WWW [Online]. Disponible en este enlace.

[13] Página oficial de la comunidad W3C [Online]. Disponible en este enlace.

[14] Documentación del HTML5 [Online]. Disponible en este enlace.

[15] Tutorial de uso del DOM [Online]. Disponible en este enlace.

[16] Historia del lenguaje JavaScript [Online]. Disponible en este enlace.

[17] Historia de la empresa Sun Microsystems [Online]. Disponible en este enlace.

[18] Página oficial del formato JSON [Online]. Disponible en este enlace.

[19] Introducción al formato XML [Online]. Disponible en este enlace.

[20] Guía de uso del lenguaje CSS [Online]. Disponible en este enlace.

[21] Nuevos módulos en CSS3 [Online]. Disponible en este enlace.

[22] Página oficial del lenguaje Sass [Online]. Disponible en este enlace.

[23] Página oficial de lenguaje Ruby [Online]. Disponible en este enlace.

[24] Historia del widget web [Online]. Disponible en este enlace.

[25] Definición de Adobe Flash [Online]. Disponible en este enlace.

[26] Definición de la GUI [Online]. Disponible en este enlace.

[27] Definición de una API [Online]. Disponible en este enlace.

[28] Explicación de RSS [Online]. Disponible en este enlace.

[29] Página oficial del laboratorio CoNWeT [Online]. Disponible en este enlace.

[30] Página oficial de proyecto europeo FI-WARE [Online]. Disponible en este enlace.

[31] Página oficial de la plataforma GitHub [Online]. Disponible en este enlace.

[32] Página oficial del lenguaje Git [Online]. Disponible en este enlace.

Este documento esta firmado por

	**Firmante**	CN=tfgm.fi.upm.es, OU=CCFI, O=Facultad de Informatica - UPM, C=ES
	Fecha/Hora	Wed Jan 07 23:54:39 CET 2015
	Emisor del Certificado	EMAILADDRESS=camanager@fi.upm.es, CN=CA Facultad de Informatica, O=Facultad de Informatica - UPM, C=ES
	Numero de Serie	630
	Metodo	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)