Guía presentación

1. Introducción

El presente trabajo se enmarca concretamente dentro del proyecto LUCA. El objetivo general de dicho proyecto es proporcionar una interfaz de acceso uniforme a un conjunto de fuentes de datos heterogéneas.

1. Motivación

En los últimos años, el volumen de datos recogidos y manipulados por las empresas ha aumentado de forma vertiginosa. Estos datos se han ido almacenando en diferentes tipos de fuentes conforme las empresas crecían y sus sistemas evolucionaban y se fusionaban. Como resultado de este proceso no es extraño actualmente encontrar empresas que tengan sus datos almacenados en sistemas tan dispares como bases de datos relacionales, hojas XML o repositorios FTP.

Como consecuencia de esta nueva situación, cuando un usuario quiere obtener una información concreta cuyos datos residen en varios de estos sistemas, ´este necesita acceder a cada uno de estos sistemas, extraer de cada sistema la información que precisa, y finalmente filtrarla y unificarla para finalmente obtener los datos requeridos.

Por ejemplo, una cadena de venta de electrodomésticos podría tener sistemas informáticos diferentes para el departamento de atención al cliente, para el departamento técnico de postventa y para el departamento de compras y adquisiciones. Por tanto, para conocer con precisión el estado actual de una reparación, podríamos necesitar:

1. Acceder al sistema de atención al cliente para obtener el identificador de la incidencia y en qué fase de su gestión se encuentra.

2. Una vez corroborado que la incidencia esta actualmente siendo atendida, recuperaríamos del sistema de gestión de reparaciones el estado detallado de la reparación. Como resultado de esta operación, supongamos que averiguamos que la reparación está a la espera de recibir una pieza que se ha de sustituir.

3. Finalmente, para poder hacer una estimación de cuando podría estar lista la reparación, accederíamos al sistema de compra y adquisiciones para averiguar cuando está prevista la entrega de la pieza solicitada.

Como hemos comentado anteriormente, a cada uno de estos sistemas podría accederse de manera diferente. Por ejemplo, el primero podría consultarse utilizando un servicio web. La información del segundo podría recuperarse accediendo directamente a una base de datos relacional, mientras que la información del tercero se obtendría analizando órdenes de compra en formato puf almacenadas en un repositorio de ficheros compartido.

Por tanto, el usuario, para poder realizar este proceso, necesita conocer las particularidades de cada sistema y de su forma de acceso. Para aliviar esta situación, dentro de la empresa CIC, se está desarrollando una aplicación denominada LUCA, a la cual contribuye este Trabajo Fin de Grado. Para facilitar este proceso de recuperación de información, LUCA proporciona un lenguaje común para todas las fuentes de datos a unificar, permitiendo al usuario abstraerse de los detalles de cada fuente.

1. Proceso
   1. Objetivo

El objetivo general de este Trabajo Fin de Grado es integrar en LUCA el concepto de proceso. Para ello, hay que dar soporte a dos cuestiones diferentes: (1) la ejecución de los procesos; y (2) la especificación de procesos. Por tanto, el objetivo general de este trabajo se descompone en estos dos subjetivos principales.

El primer objetivo implica poder tratar procesos en LUCA de la misma forma que se trata las consultas. Es decir, los procesos deberán aparecer como en las consultas bajo una pestaña de gestión y otra de ejecución. Obviamente, la complejidad de ejecutar un proceso es mayor que la de ejecutar una consulta, ya que necesitamos ejecutar varias consultas, resultados intermedios y utilizar estos resultados como entradas para otras consultas.

El segundo objetivo, que es el que implica una mayor complejidad, consiste en facilitar la especificación de procesos en LUCA. Para que un proceso pueda ser ejecutado, primero debe ser especificado, indicando qué consultas lo componen y cómo se relacionan. De acuerdo con los deseos expresados por los responsables del proyecto LUCA y la empresa CIC, dicho mecanismo de especificación debía ser gráfico, permitiendo así componer consultas de manera visual mediante la interconexión de las salidas de unas con las entradas de otras.

* 1. Antecedentes
     1. GoJS

Go.JS es una biblioteca de JavaScript para implementar editores gráficos dentro de interfaces web. GoJS facilita la implementación de tareas tales como definición de símbolos gráficos, gestión de paletas de símbolos, arrastrar y soltar (drag and drop), copiar y pegar, edición de etiquetas de texto asociadas a símbolos gráficos, menús contextuales, función de deshacer o la gestión de eventos de ratón, entre muchos otros elementos.

* + 1. Vaadin

Vaadin es un framework para el desarrollo de aplicaciones web avanzadas, también conocidas como Rich-Internet Applications (RIA). El objetivo del paradigma RIA es desarrollar aplicaciones web con interfaces avanzadas que les haga asemejarse a las aplicaciones de escritorio. La principal ventaja que aporta Vaadin es que permite escribir aplicaciones en código Java, como si fuesen de escritorio, y luego este código es transformado para que funcione en entornos web.

Una de las características diferenciadores de Vaadin es que, al contrario de las librerías de JavaScript tradicionales, Vaadin también contempla la parte del servidor, por lo se generan tanto las llamadas al servidor desde la interfaz gráfica (front-end) como la recepción y tratamiento de esas llamadas en la parte del servidor (back-end).

Para abstraer al usuario de elementos relacionados con HTML o Javascript, Vaadin utiliza los llamados componentes. Un componente representa un elemento gráfico o widget. Para el desarrollo de los componentes, Vaadin proporciona una serie de clases reutilizables que contienen la infraestructura necesaria para facilitar su traducción a código HTML y Javascript.

* 1. Flujo de desarrollo
     1. Editor Grafico

El editor gráfico utiliza GoJS para crear toda la interfaz visual. Este framework utiliza templates para definir el aspecto estético de los diferentes componentes.

* + 1. Integración
       1. Vaadin
       2. LUCA
    2. Implementación
       1. Modelo
       2. Repositorios
       3. Servicios
       4. Presenters

1. Sumario