Guía presentación

Diapositiva 3: Objetivos del proyecto

Proyecto realizado en CIC Consulting Informatico, se integra con el Producto LUCA y realizado desde Agosto de 2017 hasta Enero de 2018. El objetivo del proyecto como se expone en la diapositiva es integrar en el producto LUCA el concepto de *proceso*.

Para entender el concepto de proceso primero es necesario explicar que es LUCA y cuál es su función. 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.

LUCA proporciona un lenguaje común para todas las fuentes de datos a unificar, permitiendo al usuario abstraerse de los detalles de cada fuente.

Con proceso se entiende como la anidación o concatenación de las diferentes consultas, de forma que las salidas de una consulta forman las entradas de otra.

Diapositiva 4: Antecedentes: LUCA

Los objetivos principales del producto LUCA son: centralizar el acceso a los recursos de las diferentes fuentes de datos de una empresa, y automatizar dicho acceso a datos. De esta forma un usuario será capaz de consultar datos de forma sencilla y rápida.

En LUCA se distinguen dos roles o tipos de usuario. Por una parte, el usuario gestor es el encargado de construir las consultas, este rol requiere de un cierto nivel de conocimiento sobre cómo crear y gestionar consultas. Por otro lado, existe el rol de ejecutor de consultas, este rol no precisa de un conocimiento avanzado, se limita a ejecutar las consultas ya construidas para obtener la información deseada, es el rol más destacado ya que es el que más será utilizado por el cliente.

Video de consulta:

En el siguiente video podemos ver como un usuario con rol ejecutor, ejecuta una consulta. El usuario tras seleccionar de la lista de consultas, cual desea ejecutar, se limita a introducir los valores de entrada necesarios para ejecutar dicha consulta, en otras ocasiones, la consulta no precisará de valores de entrada. Por último, tras ejecutar la consulta se visualizan los datos.

Diapositiva 6: Motivación

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 cuándo 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.

LUCA, la voluntad del proceso es unificar todo este flujo de consultas en una consulta grande llamada proceso que se ejecute de forma única, y conseguir así toda la información sin el tedioso trabajo de estar apuntando en notas o similares los resultados de unas consultas para poder utilizarlas en otras.

Diapositiva 7: Objetivos del proyecto

El objetivo principal, como hemos dicho, es facilitar este flujo de consultas, para ello el proyecto constará con un editor gráfico donde el usuario será capaz, utilizando consultas ya creadas, conectarlas entre si mediante sus entradas y salidas, creando así una cadena de consultas para posteriormente, poder ejecutarlas y obtener así la información final.

No obstante, el usuario podrá visualizar el resultado de cada consulta realizada desde el mismo editor gráfico. Por último, el usuario será capaz de exportar los resultados finales en formato Excel.

Diapositiva 8: Requisitos del proyecto

Entrando más en la planificación propia del proyecto. La etapa de requisitos del proyecto ya estaba realizada una vez se me propuso realizar este trabajo. Entre los cuales se incluyen la nidación gráfica de consultas, la propia ejecución de los procesos además de mostrar los resultados de los pasos intermedios.

Diapositiva 9: Arquitectura

Tal y como se comentó con los requisitos, la arquitectura ya estaba definida en un inicio. Pues LUCA se compone de un diseño basado en capas aplicando en patrón Modelo-Vista-Controlador utilizando además Vaadin, este framework permite, entre otras funcionalidades, realizar interfaces gráficas en lenguaje Java abstrayéndose del lenguaje Javascript o HTML necesario para ser visualizado en Navegadores.

Para poder entender mejor este concepto, pondré un ejemplo:

En esta diapositiva podemos ver como se comporta o de que se compone la arquitectura de LUCA. Si un usuario quiere realizar la operación de listado de todas las consultas, este, desde la interfaz grafica o la UI, realizaría una interacción por la que la UI manda una señal o evento al presenter, el presenter es el encargado de la lógica de negocio y se comunica con la vista y con el modelo, por lo tanto, vemos aquí reflejado el patrón MVP, donde la vista está separada del modelo mediante un presentador. Ante la acción previa, el presenter realizara la petición de información, en este caso la lista de consultas, al controlador, el controlador se encarga de asegurarse de que quien realiza la petición posee los privilegios o permisos necesarios y después llama la capa de servicio encargada de comunicarse con el repositorio (utilizando JPA) para que este le retorne la información deseada, y comenzar la vuelta de datos al presenter, donde éste se encargada de cargar los datos en la vista y actualizarla para que el usuario sea capaz de ver los datos.

Para el proyecto podemos separar dos comportamientos. Por un lado tenemos un proyecto que realiza las funciones de componente de Vaadin, éste integra la libreriá GoJs que se explicara posteriormente junto con una lógica de negocio implementada en java para crear una herramienta de visualización de datos genérica que pueda ser utilizada independientemente del proyecto. Por otro lado esta el incremento en LUCA, donde se desea aplicar la funcionalidad de proceso utilizando este componente creado, el cual se comporta bajo la arquitectura definida anteriormente.

Diapositiva 10: Metodología

El proyecto se realizó de forma iterativa siguiendo algunos principios de las metodologías ágiles, como es la existencia de un product backlog, en un inicio se asignaron un conjunto de tareas o ítems que habría que ir completando, como es la definición del editor, la integración con vaadin, o la implementación de capas de servicio y repositorio. Entre otro conjunto de principios estaba el llevar un seguimiento, en este caso semanal, donde yo iba informando de mis avances al product owner, en mi caso fue mi gerente, de forma que este este informado de los avances y de los problemas que me iban surgiendo durante el transcurso del proyecto.

Diapositiva 11: Editor Gráfico

La primera etapa de desarrollo consistió en un aprendizaje y una primera toma de contacto con la librería GoJS, la cual se encargaría de montar toda la estructura gráfica. Esta librería es muy potente y permite tener una lógica compleja en su interior.

En esta diapositiva podemos observar cual fue el diseño obtenido tras finalizar el proyecto. Podemos observar que la caja contenedora es el propio proceso y tiene en su interior subcajas o subprocesos interconectados entre sí.

Diapositiva 12: Conector Javascript

Cuando la librería GoJS desea comunicarse con Vaadin es necesario utilizar un fichero intermedio llamado conector que haga de enlace entre Java y Javascript, este se encarga de informar a Java de los eventos ocurridos en la librería, además de mandar a la librería modificar su estado, por ejemplo, de añadir un nuevo nodo al gráfico.

En ocasiones fue necesario realizar modificaciones del comportamiento por defecto de la herramienta. Por ejemplo, en GoJS, cuando se unen dos puertos, internamente, se crea un nuevo elemento que simula dicho enlace, para mi caso, mi esquema de interacción comienza por cuando el usuario realiza una interacción con la vista, entonces yo lo que trato son eventos, por lo tanto, no necesitaba que se crease dicho enlace, sino que se mandase un evento de enlazado, para posteriormente ser tratado en el presenter y este decidir crearlo o no, debido a que un enlace puede no ser valido si las variables que se conectan son de diferente tipo.

Diapositiva 13: Integración Vaadin

Vaadin utiliza componentes para mostrar información en la vista, por ejemplo, un componente muy utilizado puede ser un simple botón, además de los ya existentes y los que se pueden construir en Vaadin, se pueden utilizar librerías javascript para dar mas potencia grafica o para realizar gráficos que no es capaz Vaadin. Para eso, se cuenta con los componentes abstractos, los cuales se componen de una lógica de negocio y de un estado y se comunican con las librerías mediante conectores, como vimos anteriormente.

El componente se sustenta bajo el modelo siguiente. En este modelo podemos identificar dos elementos principales, las cajas y los enlaces, dentro de las cajas existirán dos tipos, las de procesos y las de subprocesos, un proceso albergara un numero indefinido de subprocesos. Estas clases deberán de ser implementadas por cualquier proyecto que quiera utilizar este componente para dar toda la lógica funcional al mismo.

Con esta integración se cierra el componente genérico, un ejemple de interacción seria crear un nodo. Cuando el presenter se comunica con el componente, le pasa los parámetros necesarios para su creación y este se encarga de cambiar el estado del diagrama y de mandar al conector que realice los cambios en el mismo.

Diapositiva 15: Integración LUCA

El incremento de LUCA consta de la creación de los diferentes presenters y de las capas: controlador, servicio y repositorio. Por lo tanto, el trabajo se centró en dichas áreas, con una mayor complejidad en los presenters que son los que albergan toda la lógica de la aplicación.

Abstrayéndonos del apartado técnico, se crearon todos los menus y contextos necesarios para poder listar los diferentes procesos en estado de gestión y de ejecución en vistas separadas, para poder después, crear o modificar procesos en una vista y en otra ejecutarlos.

Diapositiva 17: Negocio y persistencia