Universidad Zaragoza
1542

Trabajo Fin de Grado

Desarrollo de dos sistemas de visualización de información en una línea de fabricación

Development of two visualization systems in a manufacturing line

Autor/es

Pablo Esteban Alpuente

Director/es

Carlos Catalán Cantero

Jesús Gallardo Casero

Escuela Universitaria Politécnica de Teruel

2018

Resumen ejecutivo

RONAL Ibérica S.A.U. es una empresa dedicada a la fabricación de llantas que tiene la necesidad de introducir nuevos sistemas de visualización de información en dos puestos de trabajo del departamento de pintura en la planta de Teruel.

De forma más concreta, se quiere mostrar información en el puesto en el cual se cubren los agujeros de fijación de las llantas, con bolas o tapones, dependiendo del modelo, para evitar las infiltraciones de pintura. También se quiere mostrar información en los puestos de la zona de trasvase, donde se comprueban los fallos que puede haber en las llantas que llegan por la cadena, y en caso de que existan, corregirlos si es posible, o retirarlas al instante.

El proyecto a realizar ha consistido en desarrollar dos nuevos sistemas haciendo uso de la información ya existente sobre los modelos de llantas que se fabrican, almacenada en una base de datos que se ejecuta en un servidor.

Previamente a la implementación de estos sistemas, se han analizado los requisitos que hemos tenido que cumplir, así como las diversas tecnologías con las que se podrían abordar estos requisitos.

Tras el análisis inicial, se ha realizado el diseño de ambos sistemas, donde se han creado todos los diagramas necesarios con los que representar la arquitectura de hardware, módulos y datos de ambos sistemas en diagramas de clases, secuencia y despliegue.

Una vez realizado el diseño, se han implementado ambos sistemas de visualización, que utilizan los sistemas de información de producción actuales de la planta, con el propósito de mostrar la información necesaria para la correcta realización de las tareas en los puestos.

En el proceso de desarrollo del proyecto, se han buscado soluciones y alternativas a distintos problemas que han surgido durante la evolución del mismo; como no poder modificar la base de datos que utilizarán nuestros sistemas, ni los sistemas que ya están en funcionamiento en la instalación, versiones de los sistemas software que se utilizan, o la falta de información que hemos tenido que completar.

Finalizado el desarrollo y las pruebas de ambos sistemas, se implantarán en distintos equipos en los puestos de trabajo. Más concretamente, un equipo se colocará en el puesto de cubrimientos y dos en la zona del trasvase. Los sistemas se ejecutarán en equipos de 64 bits, cuyo sistema operativo será Linux y la versión de Java 8 o posterior.

Abstract

RONAL Ibérica S.A.U. is a company dedicated to the manufacture of tires that has the need to introduce new information display systems in two jobs in the paint department at the Teruel plant.

More specifically, we want to show information in the post in which the fixing holes of the rims are covered, with balls or plugs, depending on the model, to avoid paint infiltrations. It also wants to show information in the posts of the transfer area, where they check the failures that may be in the tires that arrive through the chain, and if exist, correct them if possible, or remove instantly.

The project to be carried out has consisted of developing two new systems making use of the existing information abour the tire models that are manufactured, stored in a database that runs on a server.

Previusly to the implementation of these systems, we have analyzed the requirements that we have had to comply with, as well as the various technologies with which these requirements could be resolved.

After the initial analysis, the design of both systems has been accomplished, where all the necessary diagrams have been created with which to represent the hardware architecture, modules and data of both systems in class diagrams, sequence and deployment.

Once the design has been carried out, both visualization systems have been implemented, which use the current production information systems of the plant, with the purpose of showing the necessary information for the correct performance of the tasks in the posts.

In the process of developing the project, solutions and alternatives have been searched to different problems that have arisen during the evolution of the same; such as not being able to modify the database that our systems will use or the systems that are already operating in the installation, versions of the software systems that are used, or the lack of information that we have had to complete.

Once the development and tests of both systems have been completed, they will be implemented in different computers at the work stations. More specifically, one computer will be placed in the cover station and the other two in the transfer area. The systems will run on 64-bit computers, whose operating system will be Linux and the Java version 8 or later.

Contenido

1. Introducción

En la empresa RONAL Ibérica surge un proyecto, que consiste en incorporar sistemas de información nuevos en dos puestos de trabajo en el departamento de pintura de la planta de Teruel.

Dentro de la empresa RONAL Ibérica existen distintas zonas bien diferenciadas en las que se realizan distintas fases del proceso de la llanta, las cuáles son:

1. **Fundición:** zona inicial de la cadena donde se generan las llantas, a partir de mezclas de aluminio y otros componentes en las coquillas.
2. **Mecanizado:** zona donde se realizan los distintos agujeros necesarios de las llantas, además de pulirlas y eliminar defectos que se puedan encontrar.
3. **Pintura:** zona donde se realiza un pretratamiento de las llantas, para posteriormente llegar a las distintas zonas de pintado y revisado.
4. **Final de cadena:** última zona de la cadena, en la que las llantas serán validadas o rechazadas según el estado, y se prepararán para su envío, o su fundición por mal estado.

Adicionalmente, en la planta no solo existe la infraestructura mecánica de la cadena, sino también distintos sistemas y aplicaciones software en los que cualquier usuario puede hacer uso de ellos y visualizar información de las llantas y del proceso de la cadena en general.

Los datos que se utilizan en el proceso de fabricación de las llantas se encuentran almacenados en distintas bases de datos. Toda esta información se utiliza, tanto para que la cadena pueda realizar el proceso de forma automática, obteniendo la información necesaria de las distintas bases de datos, como para que los trabajadores en ciertos puestos de trabajo puedan consultar información que resulte relevante para la tarea que se esté llevando a cabo.

Uno de los sistemas con que se interactúa de forma diaria es el software EISENMANN, en el que se guarda información de toda la cadena en la zona de pintura, y con el que se podrán comparar datos de los nuevos sistemas. Toda la información que se puede encontrar y utilizar se divide en los siguientes apartados:

1. **Valores análogos:** visualizar valores de temperatura y conductividad en distintos puntos de la cadena, como hornos de secado o zonas de evaporación. Se pueden introducir de forma manual los rangos de temperatura deseados.
2. **Asignación de la tarea:** visualizar las distintas zonas dentro del departamento de pintura, diferenciadas por los robots que transportan las llantas de una zona a otra. En este apartado, se puede ver las llantas que se están transportando en cada zona de la cadena y en qué lugar se encuentran.
3. **Banco de datos de ruedas:** visualizar y actualizar, si se necesita, los datos almacenados de cada modelo de llanta que están almacenados en la base de datos. En este apartado se pueden observar todos los modelos que se fabrican de forma actual, es decir, unos 1000 modelos distintos, cada uno con su información.
4. **Evaluación de la información:** visualizar las distintas incidencias que puedan suceder en la cadena, pudiendo observar a partir de las señales, en qué punto de la cadena está el problema.

Toda la información de la empresa está interconectada, lo que permite, tanto para el funcionamiento de la cadena, como para visualizar información en un momento concreto, esté actualizada en todo momento y se pueda acceder a cualquier dato en tiempo real.

Dentro de la empresa, los objetivos a grandes rasgos que se han pretendido cumplir con la instalación de los nuevos sistemas son los siguientes:

a) Facilitar a los operarios sus labores de trabajo, dándoles información más completa, concreta, y enfocada a las funciones que deben realizar.
b) Agilizar los procesos de trabajo de la cadena.
c) Obtener una mayor tasa de calidad en su proceso de trabajo y en el producto final.
d) Implementar los sistemas de forma que sean sencillos de mantener, de modificar, y de extender a otras zonas de la planta en el futuro.

1.1. Puestos de trabajo

El desarrollo de los dos nuevos sistemas se realizará teniendo en cuenta el lugar de implantación, por lo que será necesario conocer cuál es el proceso de cada zona, las tareas que se ejecutan y cuál es la información relevante a visualizar en cada uno de los puestos.

Para obtener un mejor conocimiento del proceso, se estudiará la documentación obtenida por parte de la empresa, donde se puede observar el proceso completo de las dos zonas en las que se implantarán los sistemas, y en qué puntos trabajarán los operarios con las llantas.

1.1.1. Puesto de cubrimientos

El primer diagrama de la cadena que se muestra en la figura 1, representa el flujo de la zona en la que las llantas llegan desde la zona del pretratamiento, y posteriormente, los operarios de este puesto de trabajo se encargan de incorporan los cubrimientos en las llantas antes de su entrada en las cabinas de incorporación de polvo.

Las llantas se incorporan a esta zona de la cadena mediante la ayuda del robot que está indicado como “Robot entrega 1” en el diagrama, que transporta las llantas en grupos de tres llantas desde la zona anterior. Estas llantas llegan un minuto después al puesto donde el “Operario” incorpora en la llanta el cubrimiento adecuado en los agujeros de cubrición evitando así la introducción de polvo en estos agujeros, como se puede ver en la imagen 1. Tras pasar por las cabinas de polvo, estas llantas se transportarán a la siguiente zona de la cadena.

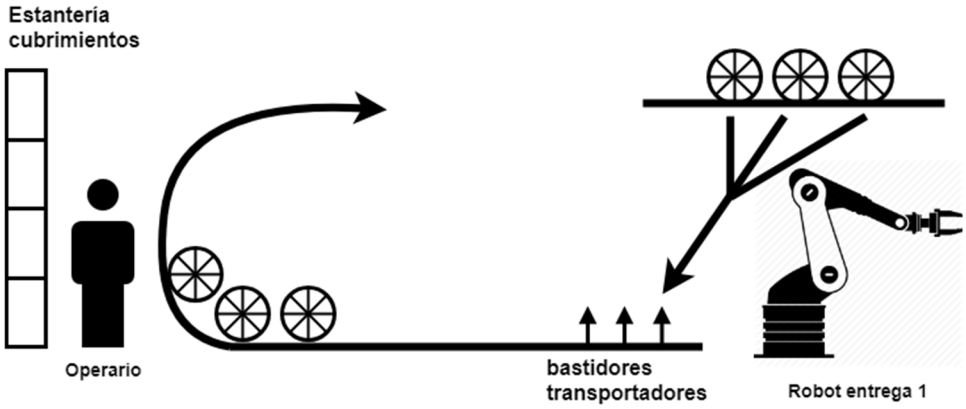


Figura 1. Proceso de la zona de cubrimientos

Cada modelo de llanta debe llevar un modelo de bola o de tapón concreto dependiendo del cliente, o por las especificaciones de la llanta. La información de qué tipo de bola o tapón debe ponerse en cada llanta está almacenada en formato papel en el puesto de trabajo.

Existen varios modelos de tapones distintos, que se pueden modificar en el futuro, pero no es lo esperado, y es esencial tener un conocimiento exacto del cubrimiento que debe llevar incorporado cada modelo.

Si se utiliza un tapón o bola incorrecto, la llanta quedará inutilizada, ya que, al pasar por la cabina de aplicación del polvo, los agujeros quedarán cubiertos por el mismo polvo, caso que ya hemos explicado y que se debe evitar. Actualmente, existe información acerca de los cubrimientos que debe llevar cada modelo localizada en paneles de información.

Imagen 1. Operario de la zona de cubrimientos (Cortesía de RONAL Ibérica)

En el momento en que un operario que no tiene experiencia previa debe introducir un tapón o bola en un modelo desconocido para él. El resultado más común se resume en la colocación de un cubrimiento incorrecto en un gran número de llantas del mismo modelo, que terminarán como llantas “NIO”, es decir, para eliminar.

1.1.2. Puesto de revisión en el trasvase

El segundo diagrama que se muestra en la figura 2, representa el flujo de la zona de trasvase. En esta zona, se encuentran trabajando cuatro operarios, dos al comienzo del trasvase, y dos en el final. Estos operarios se encargan de revisar cada una de las ruedas que llegan, realizando tareas comunes en cada una de las llantas, y tareas específicas según el modelo de la llanta.

Las tareas comunes consisten en revisar en cada una de las zonas destacadas de la llanta en base al puesto del operario, y reparar defectos leves que puedan existir en las llantas haciendo uso de las mesas de reparación. Además, en caso de detectar un defecto grave que no pueda ser corregido en las mesas, el operario deberá retirar la llanta para su procesado posterior en otra zona de la planta.

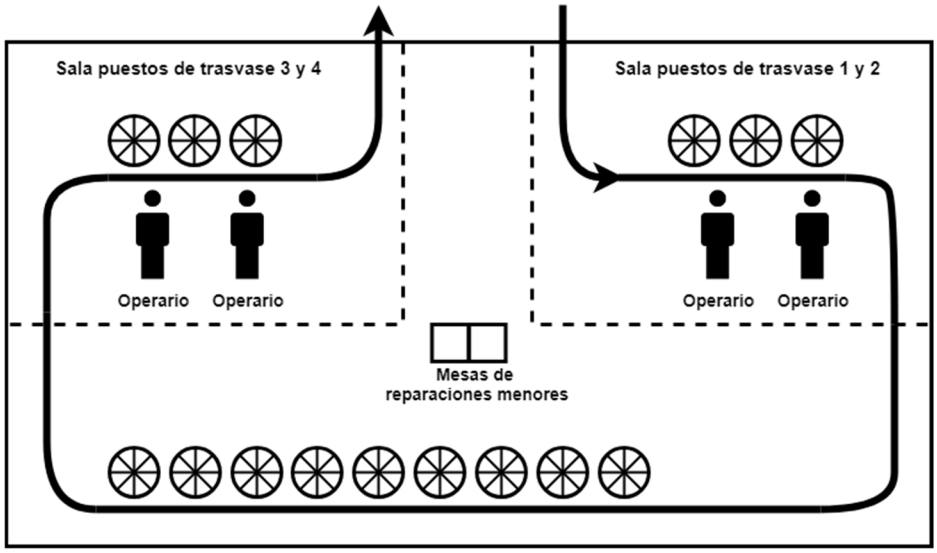


Figura 2. Gráfico de funcionamiento de la zona del trasvase

Cada modelo puede ser más o menos problemático, ya sea por geometría de la llanta o por la pintura que se incorpora. Debido a estos factores, existen llantas que apenas requieren de manipulación por parte de los operarios, y en ciertos modelos se deben revisar diversos problemas que se materializan de forma constante.

La información de la que deben hacer uso los trabajadores en estos puestos consiste en:

a) Acciones a deben realizar continuamente
b) Zonas de las llantas que deben comprobar en cada punto
c) Lijas que se deben utilizar en cada modelo de llanta

Esta información actualmente se encuentra en distintos paneles informativos como se ve en la imagen 2.

Imagen 2. Operario de la zona de trasvase (Cortesía de RONAL Ibérica)

Los problemas en este punto se pueden dar en el momento en que un operario no sabe con certeza qué problemas son los que se suelen dar con más frecuencia en cada

modelo. Existen ruedas más problemáticas que necesitan de una mayor atención, ya sea por problemas complicados de detectar a simple vista o por el tipo de pintura que se incorpora después. Al hacer uso de una lija que no es la adecuada para un modelo puede repercutir en que llantas en mal estado avancen por la cadena y se detecten más tarde, lo que influye primero, en una tasa de calidad menor, y segundo, en una detección tardía de los defectos.

2. Análisis

El punto de partida del proyecto consiste en saber qué funcionalidades se necesitan cubrir con los sistemas, así como conocer qué información se va a utilizar, cómo se va a mostrar, en que formato se almacenará toda la información, etc.

A continuación, se detallan los requisitos que se han obtenido durante la especificación del proyecto, tanto funcionales, como no funcionales, así como las restricciones que se han podido detectar. Estos requisitos de muestran en el diagrama 1.

2.1. Requisitos funcionales

RF1 - Los operarios del puesto de cubrimientos podrán visualizar la información necesaria de la llanta que están procesando.

RF2 - Los operarios del puesto de cubrimientos podrán visualizar las referencias de los modelos que está cargando el robot en esa parte de la cadena.

RF3 - Los operarios del puesto de cubrimientos podrán ver una imagen del tapón o la bola que deben colocar.

RF4 - Los operarios de los puestos del trasvase podrán visualizar la información necesaria de la llanta que están procesando.

RF5 - Los operarios de los puestos del trasvase podrán visualizar las referencias de los modelos que están llegando en esa zona de la cadena.

RF6 - Los operarios de los puestos del trasvase podrán ver los defectos más importantes que deben revisar de cada llanta.

RF7 - Los operarios de los puestos del trasvase podrán ver las funciones generales que se deben realizar en estos puestos, así como información adicional, como la calidad de la cadena del día anterior.

RF8 - Los administradores podrán modificar los datos almacenados en la base de datos, que se mostrarán en los dos sistemas.

RF9 - Los administradores podrán crear, actualizar y eliminar las imágenes que hacen referencia a los cubrimientos en la zona de tapones.

RF10 - Los administradores podrán crear, modificar y eliminar los ficheros PDF que se mostrarán en el sistema de la zona del trasvase.

RF11 - Los administradores podrán actualizar los ficheros utilizados para conectarse a los distintos servidores, que contienen tanto los ficheros como las bases de datos.

2.2. Requisitos no funcionales

RNF1 - La información de la llanta que podrán ver los operarios del puesto de cubrimientos consiste en el número del proyecto, la referencia del modelo, la cabina en la que se usará el polvo, y el tipo de cubrimiento que deben colocar.

RNF2 - La información de la llanta que podrán ver los operarios de los puestos del trasvase consiste en el número del proyecto y la pintura que se le aplicará a la llanta posteriormente.

RNF3 - La información de ambos sistemas se actualizará cada 6 segundos, en caso de que sea necesario.

RNF4 - Los datos de las llantas se extraerán de una base de datos SQL.

RNF5 - Se mostrará en la pantalla un valor “0” en el momento en que los pivotes de la cadena no transporten ninguna llanta.

RNF6 - Las imágenes de los cubrimientos se guardarán en el servidor y serán accesibles por los administradores

RNF7 - El cubrimiento en la base de datos deberá tener el mismo nombre con el que se guarda la imagen correspondiente en el servidor.

RNF8 - En el caso de no existir una imagen correspondiente a un cubrimiento, se mostrará una imagen indicando la falta del dato.

RNF9 - En el caso de que la llanta siguiente sea la misma, no se consultará la información en el servidor ni en la base de datos.

RNF10 - Los ficheros que se mostrarán en el sistema de la zona de trasvase deberán guardarse en formato PDF

RNF11 - Los ficheros PDF deberán guardarse en el servidor y serán accesibles por administradores

RNF12 - En caso de no existir un fichero PDF correspondiente a un modelo de llanta o el pivote no transporte ninguna llanta, se mostrará un PDF que contenga información general y avisos diarios.

RNF13 - En el caso de que la llanta siguiente sea la misma, se cargará la siguiente página del pdf del modelo, en caso de que el pdf tenga más de una página.

RNF14 - Se utilizará la versión 8 de Java en los equipos en los que se ejecuten los sistemas.

RNF15 - Se visualizará en el sistema de cubrimientos la información de la llanta que transporta el bastidor “20”

RNF16 - Se visualizará en el sistema de defectos del trasvase la información de las llantas que transportan los bastidores “813” en los dos primeros puestos, y el “904”, los dos últimos puestos.

2.3. Restricciones de sistema

RT1 - Los ficheros PDF se guardarán utilizando como nombre los cuatro números centrales de los proyectos de las llantas, por ejemplo, “4000.pdf”.

RT2 - Los ficheros PDF se guardarán dentro de carpetas nombradas de la misma forma que el PDF, es decir, con el código de la llanta.

RT3 - Las imágenes se guardarán servidor utilizando como nombre de imagen el tipo de cubrimiento y su código, por ejemplo, “TAPON 95” o “BOLA 26”.

2.4. Diccionario

- **Administrador:** Personal encargado del mantenimiento de la información que se mostrará en ambos sistemas, esto engloba la información de los modelos de las llantas, las imágenes de los cubrimientos y los ficheros PDF.
- **Operario del puesto de cubrimientos:** Personal que visualizará la información del sistema diseñado para este puesto. No tienen permiso para modificar la información del sistema.
- **Operario del puesto del trasvase:** Personal que visualizará la información del sistema diseñado para este puesto. No tienen permiso para modificar la información del sistema.

- **Cubrimiento:** Tapón o bola que se incorpora en la llanta para evitar el filtrado de polvo en los agujeros de fijación.
- **Pivote o bastidor:** Soporte que transporta las llantas a lo largo de toda la cadena durante el proceso productivo.

2.5. Casos de uso

Una vez definidos todos y cada uno de los requisitos que deben cumplir los dos sistemas de forma conjunta, se procederá a plasmar de forma gráfica los requisitos a los que tiene acceso cada tipo de usuario en el diagrama 1.

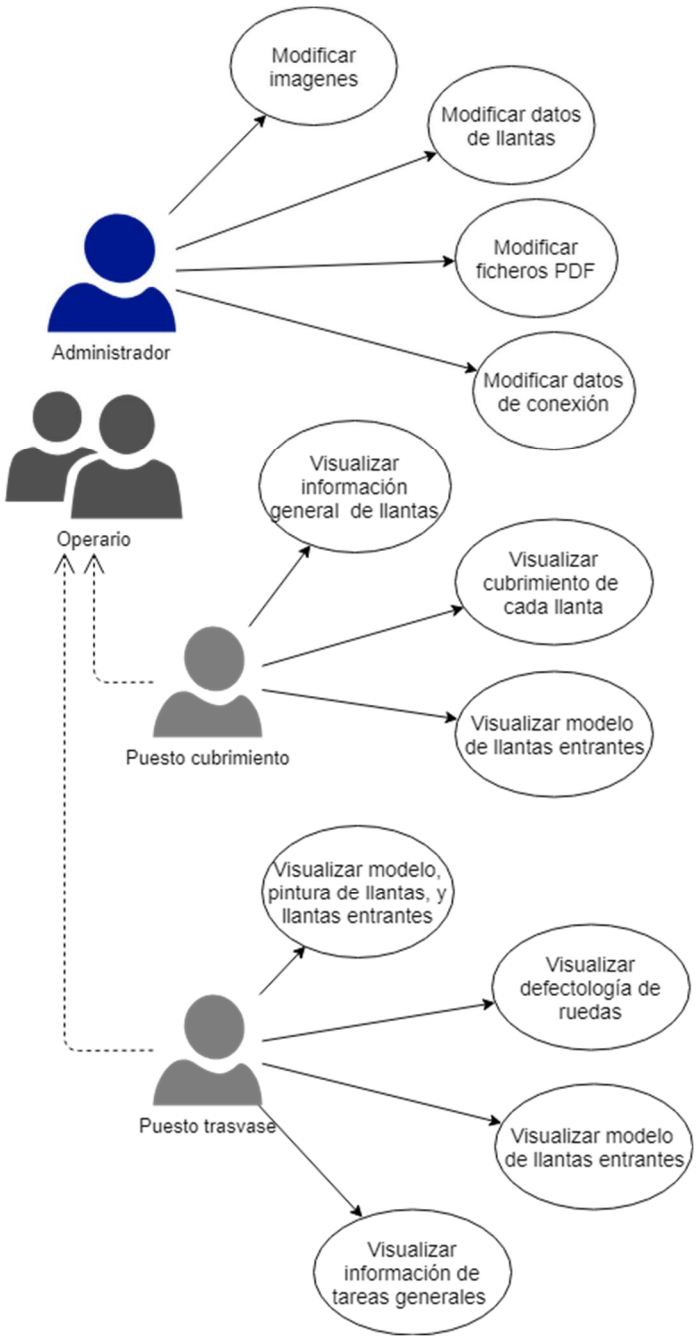


Diagrama 1. Casos de uso

2.6. Tecnologías utilizadas

De forma previa al desarrollo, se realizará un análisis con el fin de tener conocimiento sobre las tecnologías que pueden ser utilizadas en este proyecto.

El análisis se ha realizado sobre distintas combinaciones de tecnologías en las que analizamos tanto la programación *back-end*, como la programación y creación de las interfaces, conocida también como *front-end*, ya que ambas partes son fundamentales, y necesitamos tecnologías que nos faciliten una buena sinergia entre las dos partes.

Por esto, se han analizado tres posibles combinaciones de tecnologías con las que se podrían resolver todos los requisitos, que se explican de forma breve a continuación.

- La primera posibilidad comprende los lenguajes *Tcl* y *Tk* [5], dos tecnologías que van de la mano. Tcl enfocado a la programación *back-end* mediante programación de scripts y prototipos rápidos, cuya principal ventaja reside en poder utilizar métodos escritos en otros lenguajes para cubrir funcionalidades que desde *Tcl* no se pueden realizar. *Tk* nos ofrece la creación de interfaces gracias a su librería de elementos, utilizable desde otros lenguajes como *Ruby* o *Python* utilizando un puente *Tcl*.

- *Python*, junto con *Glade [6]*, son otra alternativa con la que se podría realizar este proyecto, utilizando *Glade* como tecnología para crear las interfaces de los sistemas, y *Python* para proporcionar las funcionalidades solicitadas, analizando librerías que nos permitan cubrir cada uno de los requisitos.

- *Java* y *JavaFX* es la última combinación analizada. *JavaFX* nos proporciona un desarrollo rápido de interfaces, además de sencillo de utilizar junto con *Java*, desde donde se podrían desarrollar todas las funcionalidades descritas.

  Tcl y *Tk* son descartadas debido a que son unas tecnologías ineficientes en lo que respecta a tiempos de ejecución, y estos sistemas deberán de estar en continuo funcionamiento, además de ser tecnologías desconocidas dentro de la empresa. *Python* y *Glade* podría ser una buena combinación, ya que permite implementar las funcionalidades requeridas, pero de cara a los trabajadores de la empresa, no es una tecnología utilizada dentro de este entorno, por lo que se prefiere utilizar una tecnología que resulte conocida en la empresa.

Nos decantamos por el uso de *Java* y *JavaFX* para el desarrollo de ambos sistemas por los siguientes puntos principales:

1) Lenguajes conocidos por los trabajadores de la empresa
2) Librerías adecuadas para cumplir todas las funcionalidades
3) Librerías propias de JavaFX. Scheduled Service [1] y Task [2].
4) Interfaces generadas en lenguaje XML

Java proporciona las herramientas necesarias para obtener la información de las llantas que debemos mostrar en ambos sistemas, haciendo uso de conexiones al servidor y a la base de datos. Un punto importante que ha decantado esta decisión recae sobre el requisito no funcional que nos indica que debemos poder trabajar con ficheros en formato PDF en el sistema del trasvase, el RNF10.

Apache ofrece librerías *Open source* para lenguaje Java, con la que se puede solventar el uso de ficheros en formato pdf en el sistema, y estos se pueden mostrar con facilidad en las interfaces creadas con *JavaFX*. Se detallará más adelante cuáles son estas librerías, y como se han utilizado en la fase de desarrollo.

Por último, *Java* es conocido por los trabajadores de la sección de informática de la empresa, por lo que será sencillo realizar modificaciones, del mismo modo que las interfaces, que serán generadas en lenguaje *XML* haciendo uso de *JavaFX*, y también es conocido por los informáticos.

3. Diseño

En este apartado se va a definir la infraestructura interna y externa de ambos sistemas en los distintos diagramas de diseño, que nos permitirán reflejar cuál será su proceso de funcionamiento, su estructura interna y cómo será la implementación a nivel de hardware.

3.1. Diagramas de clases

El primer diagrama de clases, que se puede ver en el diagrama 2, hace referencia al diseño del sistema de cubrimientos. Como se puede observar, se utilizará un patrón *MVC*, con el que conseguimos separar la lógica de los sistemas, la definición de los datos y la representación de estos en las interfaces, tres módulos que estarán interrelacionados.

Esto supone una facilidad de cara al proceso de desarrollo, y su posterior mantenimiento, así como a la reutilización de código. Haciendo uso de este patrón en nuestros sistemas, la estructura quedaría definida de la siguiente forma:

- **Modelo:** en el modelo se encontrará, tanto la definición de los datos, como la definición de la conexión a la base de datos. Además de los distintos métodos que se usarán para crear y utilizar las conexiones, y los métodos necesarios para obtener la información de las llantas.

- **Vista:** en la vista se pueden observar dos ficheros *fxml*, que implementan las interfaces definidas en lenguaje XML. Los controladores de cada una de las interfaces, que serán los encargados de modificar los campos que se muestran en la vista, se incluyen en el mismo paquete de los *fxml*, ya que, por definición, en *JavaFX* se estructura de esta forma.

- **Controlador:** el controlador se encarga de solicitar datos al modelo para mostrarlo en las vistas. En nuestro caso, el controlador estará formado por distintas clases, en las que se definirán distintos hilos de ejecución para cargar las vistas, solicitar los datos necesarios del modelo, y mostrar esta información en las vistas. Más adelante se detallará como se realiza cada parte del proceso.

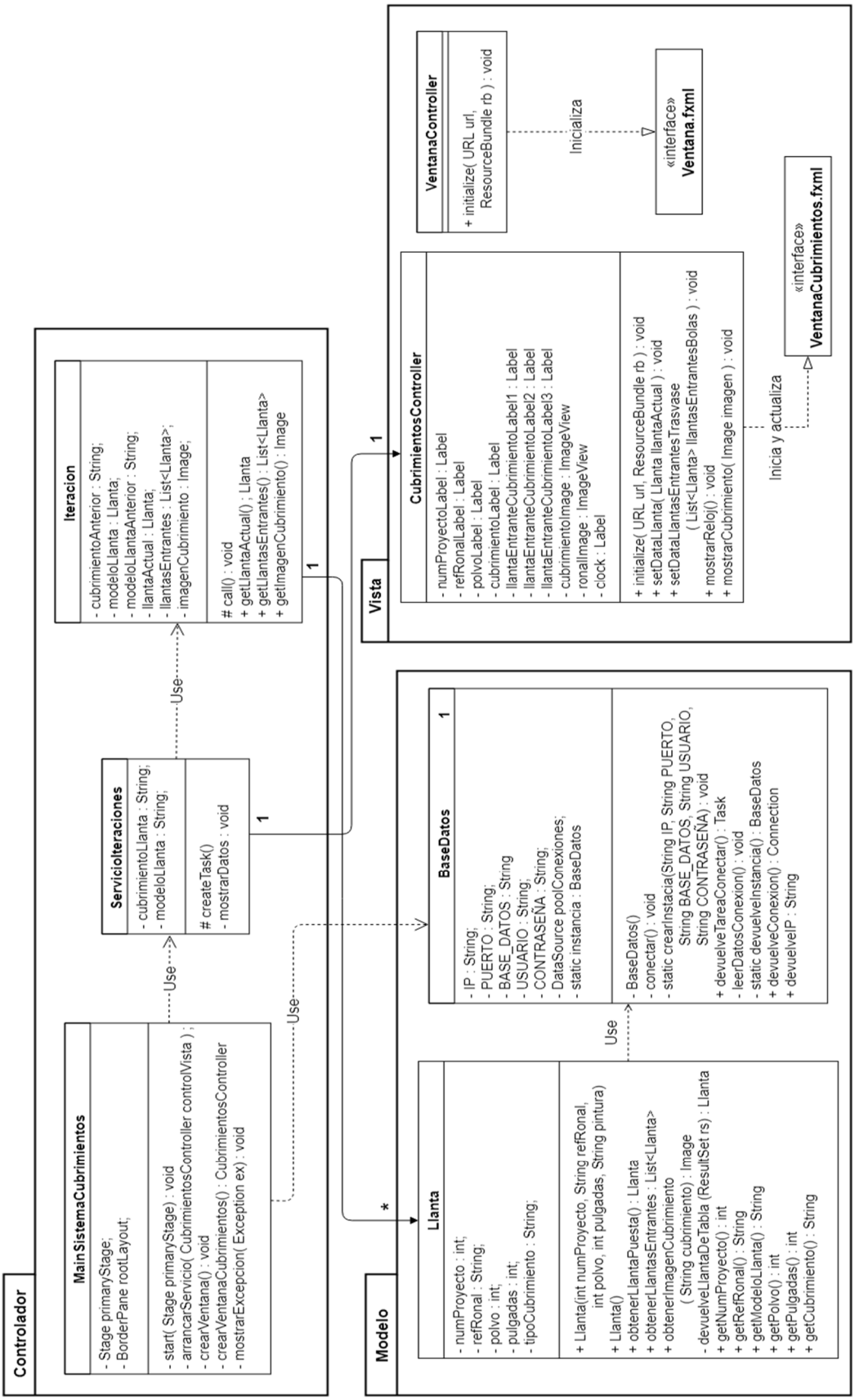


Diagrama 2. Diagrama de clases. Sistema de cubrimientos

El diagrama 3 hace referencia al diseño del sistema que se utilizará en los puestos de la zona de trasvase. La estructura es similar a la del sistema de cubrimientos, modificando los métodos y variables de forma que se puedan utilizar ficheros PDF.

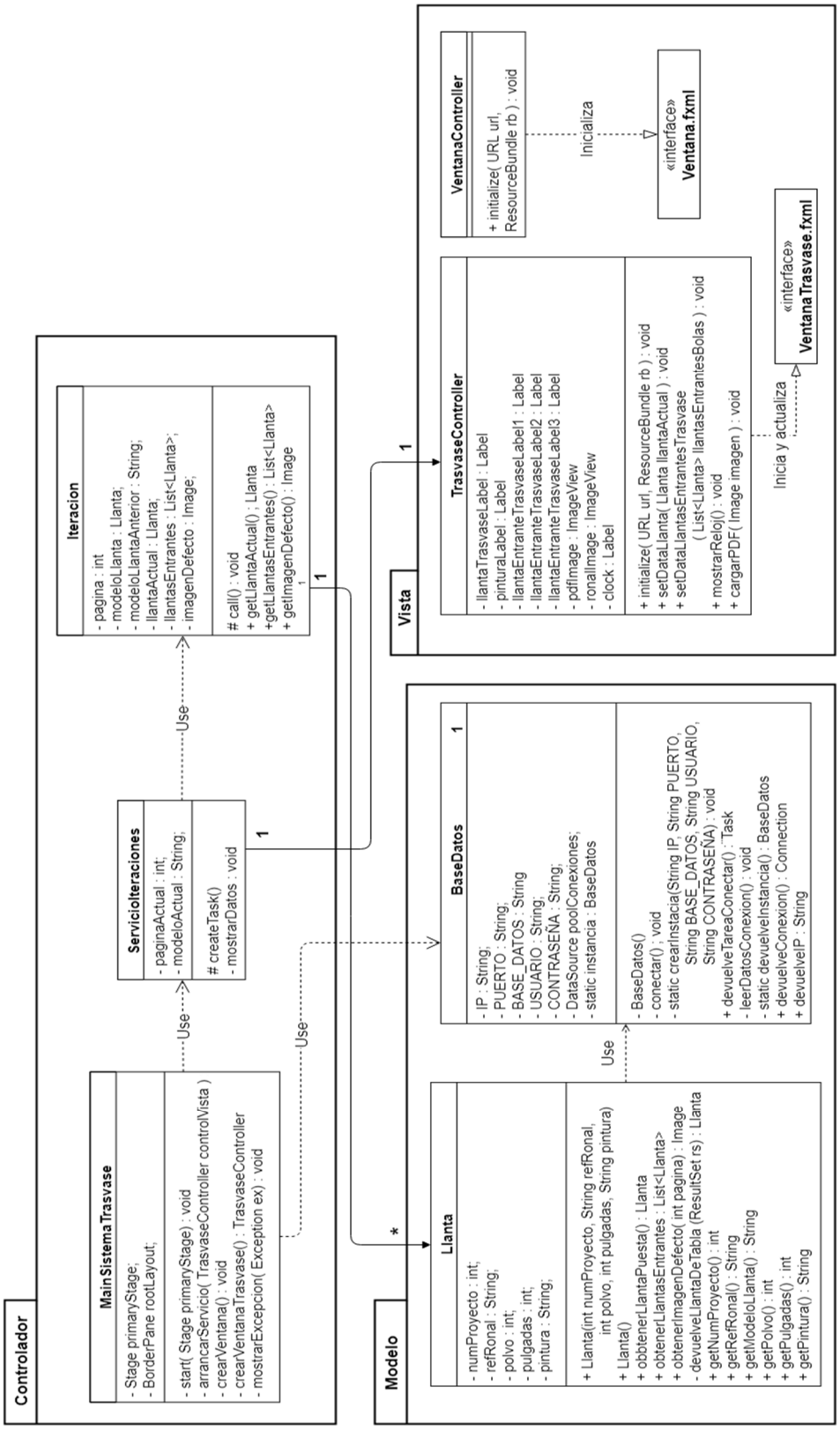


Diagrama 3. Diagrama de clases. Sistema de fallos de trasvase

3.2. Diagrama de despliegue

Se puede observar en el diagrama 4 donde se ejecutará cada uno de los sistemas. Para el puesto de cubrimientos solo se necesitará un equipo, mientras que para la zona de trasvase se necesitarán dos ya que hay dos puestos de trabajo.

Los sistemas se ejecutarán en los equipos, que se conectarán durante la ejecución a un servidor para acceder a la base de datos donde se encuentra toda la información de las llantas. Estos datos se obtendrán haciendo uso de dos tablas de la base de datos, con las que se puede obtener toda la información sobre las llantas que se procesan en la cadena.

Además, estas conexiones hacia el servidor por parte de los sistemas se utilizarán para acceder a las carpetas compartidas, donde se encuentran las imágenes usadas en el sistema de cubrimientos y los ficheros PDF usados en el sistema del trasvase

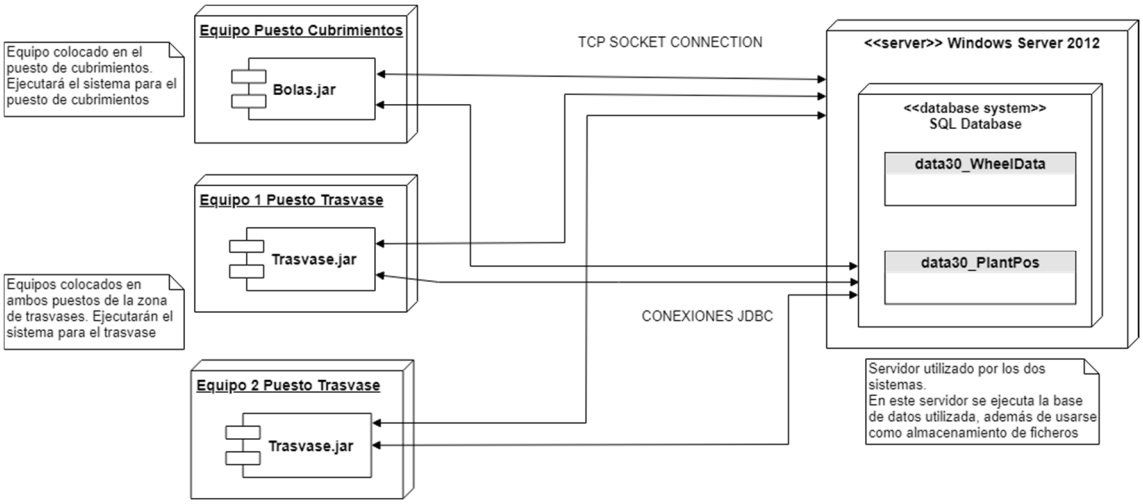


Diagrama 4. Diagrama de despliegue de ambos sistemas

3.3. Diagramas de secuencia

Tanto para el sistema de cubrimientos como para el sistema de la zona de trasvase se van a definir diagramas de secuencia, que representarán para cada uno de los sistemas, la primera ejecución completa de cada uno de los sistemas.

Una vez ejecutada la primera iteración, el proceso de repetirá cada seis segundos en cada uno de los sistemas, desde el punto en que se crean los hilos de ejecución *Task* desde el servicio ScheduledService. Más adelante se detallará el funcionamiento de estos elementos.

El diagrama que representa el sistema de cubrimientos se puede observar en el diagrama 5, y el que representa el sistema de los puestos de trasvase en el diagrama

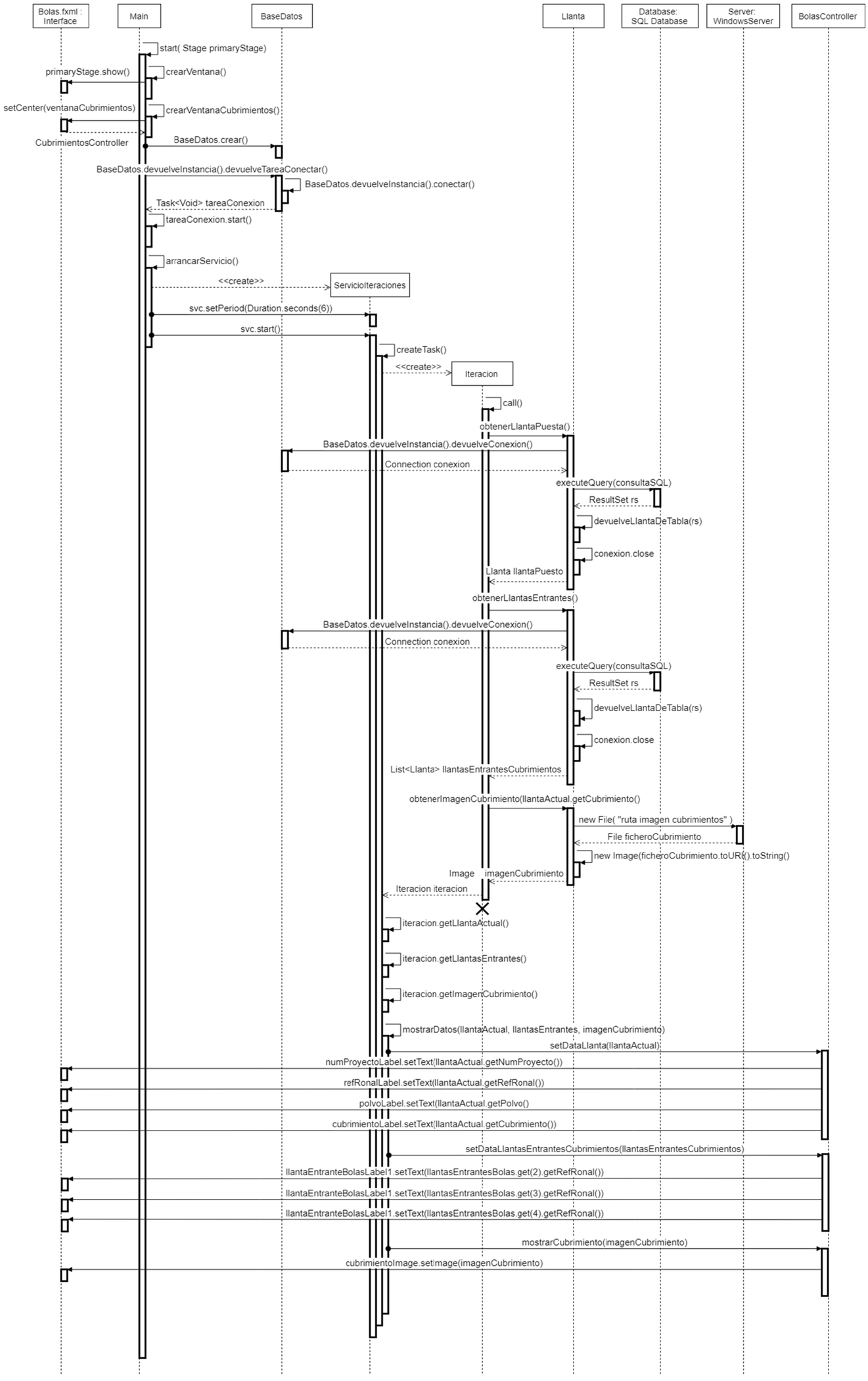


Diagrama 5. Diagrama de secuencia del sistema de cubrimientos

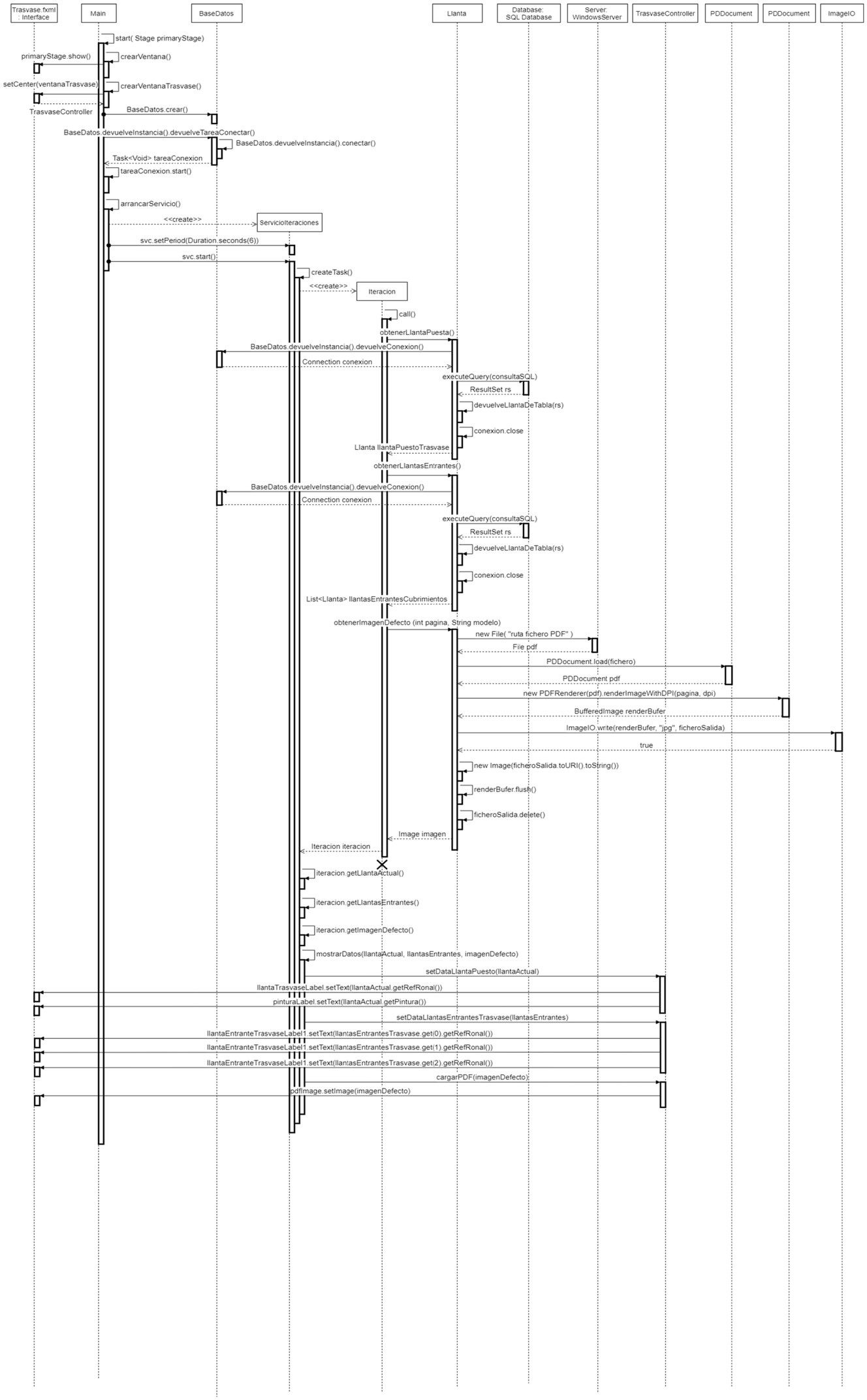


Diagrama 6. Diagrama de secuencia del trasvase

3.4. Prototipos de la interfaz

En este punto, y sabiendo qué información quiere incorporar la empresa en ambos puestos de trabajo de la cadena, se crearán distintos prototipos sencillos con la idea de que los responsables del proyecto los aprueben, o, en caso contrario, se modifiquen y se adecúen a la idea general que existe, acerca de cómo se espera que se muestre la información en los puestos de trabajo.

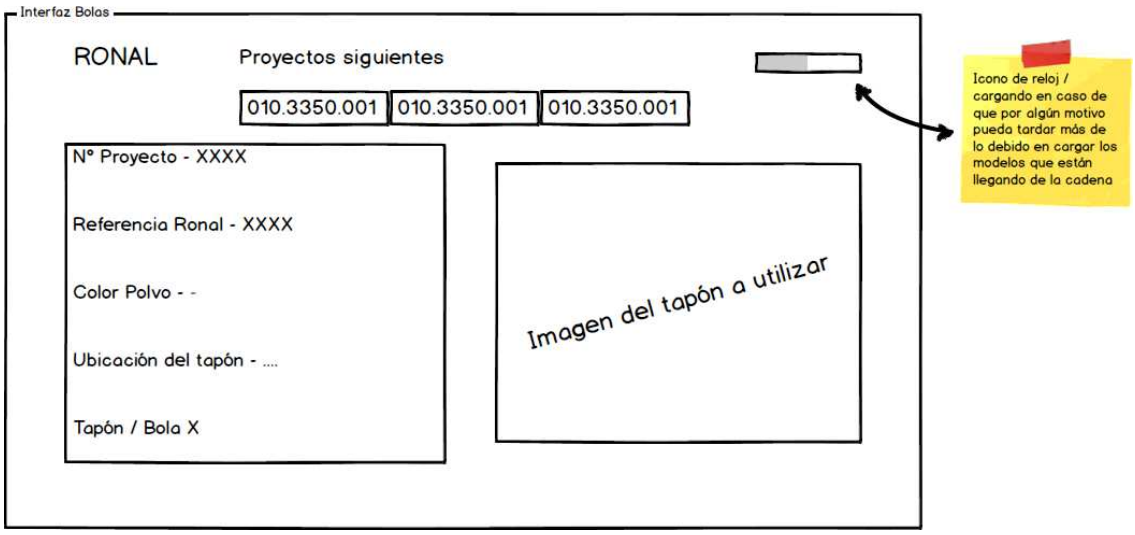
Los prototipos se crearán tanto para el sistema de cubrimientos, como para el sistema de fallos del trasvase, y ambos se pueden observar a continuación.

3.4.1. Prototipo del puesto de cubrimientos

En el prototipo 1 se puede observar un primer acercamiento acerca de cuál podría ser un resultado final de la interfaz, que será visualizada por los trabajadores que estén en el puesto de las bolas y los tapones.

Los operarios de este puesto necesitan información clara y directa, por lo que el prototipo que se creará estará enfocado a una interfaz sencilla, que contenga la información justa y necesaria para que el operario pueda realizar su trabajo de la mejor forma posible.

La información que se deberá mostrar por pantalla se ha detallado previamente en la fase de requisitos, por tanto, se organizará la disposición de los elementos que se deben mostrar en la interfaz de una forma organizada, valorando la importancia que puede tener cada uno de los elementos durante la ejecución del sistema.



Prototipo 1. Prototipo del sistema de cubrimientos

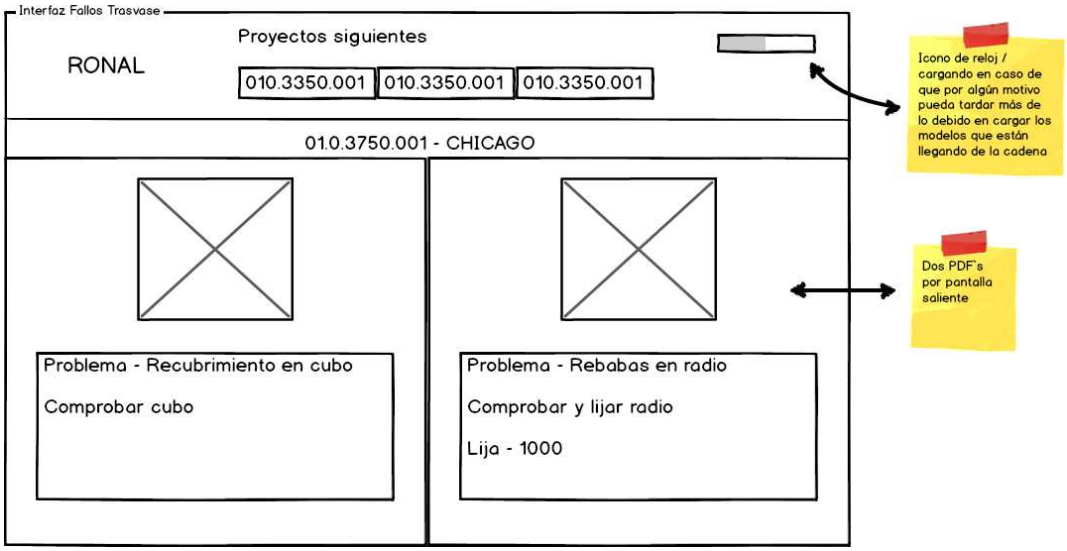
3.4.2. Prototipos de los puestos de fallos en trasvase

Los siguientes prototipos están enfocados a los sistemas que se implantarán en los puestos de trasvase. En estos puestos la información que deben mostrar los sistemas es distinta como se ha podido apreciar en el apartado de requisitos.

De la misma forma, buscamos para estos puestos de trabajo una interfaz sencilla y que muestre una información directa, sin demasiados elementos en la interfaz que puedan

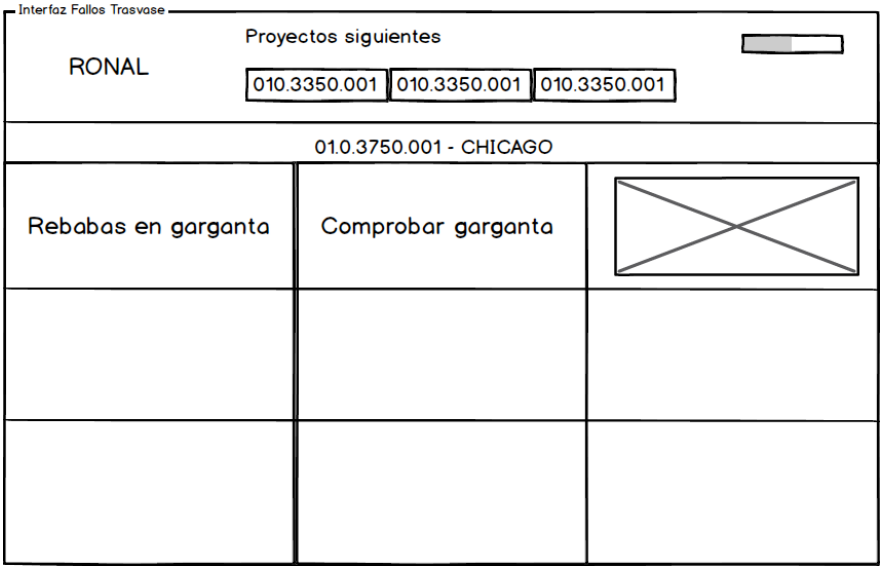
distraer la atención del punto principal, los ficheros que contienen las órdenes de trabajo.

Por esto, en el prototipo 2 se intentará aprovechar la mayor parte de la interfaz para mostrar los ficheros PDF, sin dejar de mostrar información adicional, como, por ejemplo, las referencias de los modelos siguientes que entran por la cadena.



Prototipo 2. Prototipo nº1 del sistema de fallos de trasvase

Se analizaron varios prototipos dentro de la organización de la interfaz para este sistema. Debido a que cada modelo de llanta puede mostrar más o menos información, surge la idea de crear un fichero patrón que se complete para cada modelo de llanta, y mostrar ese fichero según el modelo que se cargue. Como se puede observar en el prototipo 3, el PDF ocuparía la mayor parte de la interfaz, de forma similar al prototipo anterior.



Prototipo 3. Prototipo nº2 del sistema de fallos de trasvase

Aunque cada uno de los prototipos fue aprobado por los encargados del departamento de pintura, las interfaces finales se realizaron en base a los prototipos con pequeñas modificaciones.

4. Desarrollo

En este apartado se detallarán los pasos más relevantes que se han realizado durante el desarrollo de ambos sistemas. Por tanto, será necesario explicar los cambios que se han realizado en los datos de la planta, cómo se obtiene y visualiza la información, de qué forma se han implementado los puntos más importantes del sistema, y cómo se han solucionado los distintos problemas que han surgido durante este proceso.

4.1. Modificaciones en datos de la empresa

La primera tarea que se deberá realizar consiste en analizar los datos que existen dentro de la planta, analizar qué datos se necesitarán mostrar en los sistemas, y completar la información, si es necesario. En este proyecto, se detecta como información relevante, la información de los modelos de las llantas, las imágenes de los cubrimientos y los ficheros PDF.

4.1.1. Información de las llantas

Analizando la información que existe actualmente sobre los modelos de las llantas, se observa que no existe en la base de datos ningún tipo de información que haga referencia al cubrimiento que debe llevar cada modelo, por lo que será necesario actualizar la información de la base de datos.

En la base de datos están almacenados alrededor de mil proyectos de llantas distintos, que son los que actualmente se fabrican en la planta. Estos modelos se van actualizando conforme una llanta ya no se produce, se comienza a fabricar un nuevo modelo, o hay que modificar los datos actuales.

En la información que se almacena de cada modelo comprobamos que hay un campo del que no se hace uso alguno, por lo que, previo permiso de los encargados del departamento de pintura para poder modificarlo, se utilizará ese campo para incluir en cada uno de los modelos el tipo de cubrimiento que se utiliza para cada llanta, obteniendo así una información más completa y actualizada de las llantas.

4.1.2. Imágenes de los cubrimientos

Para tener la información del sistema de cubrimientos completa, será necesario incluir imágenes de cada uno de los modelos de tapones y bolas. Estas imágenes se mostrarán por pantalla completando la información escrita, es decir, si un modelo debe llevar una bola cuyo código es el “26”, aparecerá una imagen de la bola para dar más facilidad a los trabajadores del puesto. Estas imágenes se almacenarán en un servidor al que accederá el sistema para obtener la imagen correspondiente a cada una de las llantas que se transporta por la cadena.

Se incluirán dos imágenes, una con nombre “FALTA IMAGEN.jpg” que será mostrada en el momento en que pase una llanta cuyo campo acerca del cubrimiento que debe llevar esté vacío o no concuerde con ninguna de las imágenes almacenadas en el servidor. La otra imagen que se incluirá tendrá por nombre “0.jpg”, y mostrará un mensaje “Esperando proyecto” cuando los bastidores no transporten ningún modelo. Con estas dos imágenes adicionales se pueden mostrar todas las opciones necesarias en el sistema de cubrimientos.

4.1.3. Ficheros PDF de fallos del trasvase

Por último, la información de los fallos más comunes que se materializan en cada uno de los modelos de las llantas de la fábrica se almacena día a día en formato Excel de Microsoft Office, en los que se adjuntan enlaces a carpetas del equipo donde incluyen imágenes de los fallos que se han materializado.

Los requisitos del proyecto determinan que el sistema de la zona del trasvase deberá mostrar los defectos de las llantas haciendo uso de ficheros en formato PDF, por lo que será necesario generar esta documentación en los modelos que presenten defectos relevantes.

Esta documentación se almacenará en el mismo servidor donde estarán las imágenes de los cubrimientos, donde cada llanta tendrá su carpeta identificada con el número del modelo, y dentro se guardará el fichero PDF correspondiente al modelo, identificado según se ha establecido en los requisitos.

En caso de no existir un PDF correspondiente al modelo que está pasando por la cadena, o el bastidor no contiene ninguna llanta, se mostrará un pdf que contiene información general diaria, almacenado también en el servidor con código "0".

4.2. Librerías analizadas y utilizadas para uso de ficheros PDF

Para poder cumplir todos los requisitos definidos, se deberá encontrar una herramienta o conjunto de herramientas que nos permitan poder manipular los ficheros PDF, por lo que se analizan distintas opciones que nos puedas resultar útiles teniendo en cuenta las tecnologías que se han decidido utilizar.

- *JPedalFX* es la primera opción analizada para solventar este problema [10]. *JPedalFX* es una librería enfocada a lenguaje Java en la que están implementados distintos métodos para poder manipular ficheros PDF. Queda descartada cuando se descubre que esta librería fue *Open source* durante un periodo de tiempo, pero, a día de hoy esta librería es comercial, por lo que no nos sirve.

- *iText* es la siguiente librería analizada [11]. Fue descartada rápidamente por dos motivos muy simples, primero, no integraba las funcionalidades que necesitábamos para nuestro sistema, y segundo, no es una librería gratuita, dos motivos de peso para ser descartada de forma automática.

- *jPDFViewer* es la siguiente posibilidad analizada [9], debido a que buscando información en distintos foros aparece como alternativa al uso de *iText*. *jPDFViewer* es un *bean* para Java, desarrollado por la empresa "*Qoppa Software",* que nos permite realizar las funcionalidades que necesitamos. Esta sería una buena opción, si no fuera por el hecho de que esta librería también es comercial, por lo que debe quedar descartada.

- Analizando otras librerías, surge la idea de cambiar el lenguaje de programación actual, es decir, *Java*, por *Python*. Esto se debe a que aparece una librería llamada *Poppler* [7], una librería desarrollada en lenguaje *C* y basada en *Xpdf* que nos ofrece las funcionalidades que necesitamos. Combinando *Poppler* con *PyGTK*, el intérprete entre *Poppler* y *Python*, se podrían solventar los requisitos solicitados, por lo que

queda como alternativa en caso de que no se encuentre una librería factible para *Java*.

- Se analizan las funciones disponibles en una librería propia de *JavaFX* llamada *HostServices* [12]. Esta librería permite abrir documentos PDF a partir de una URI, y gracias al método *showDocument()* se puede mostrar el contenido del fichero seleccionado en un buscador web o en un componente *Tab*. Se modifica la interfaz, convirtiendo la sección en la que se quiere mostrar los ficheros en una sección de tipo *WebView*, y se realizan pruebas con esta librería. Los resultados se resumen en que nos permite obtener los ficheros PDF, pero no conseguimos mostrar el contenido. Se intenta mostrar el contenido con distintos componentes de *JavaFX*, como *ImageView*, *WebView*, *Tab*, sin obtener resultado con ninguno de ellos.

- Por último se encuentra *PDFBox* [8], una librería *Open source* desarrollada por *Apache* que permite manipular ficheros PDF de cualquier manera, ya que no solo permite obtener un fichero de este tipo y mostrarlo, gracias a esta librería se puede extraer el texto de un pdf, crear PDF, mostrar los ficheros utilizando la API estándar de Java, guardar los PDF como imágenes, etc.

Se comprobará que, utilizando junto a PDFBox otras dos librerías, en este caso *FontBox* y *common-login*, que completarán, por un lado, las funcionalidades de formatos de fuentes de distintos tipos de archivos, y por otro, poder generar distintas salidas de ficheros log para componentes específicos dentro de los sistemas, estas librerías nos permiten mostrar los ficheros PDF en nuestro sistema resolviendo así los requisitos enfocados al uso de ficheros PDF.

4.3. Código de las aplicaciones

El código realizado durante el desarrollo de ambos sistemas no se podrá adjuntar en el documento debido a políticas de privacidad y propiedad intelectual de la empresa.

Las partes más relevantes de los sistemas serán detalladas en los distintos apartados de este documento.

4.4. Conexión a la base de datos

Los sistemas deberán estar en ejecución durante periodos largos de tiempo, y es esencial implementar las conexiones de forma eficiente, por ello, en nuestros sistemas se implementará un patrón *Singleton*, con el que aseguramos que solo existirá una única instancia de conexión abierta en los sistemas, y la instancia de este patrón contendrá un *pool* de conexiones.

Haciendo uso de algunas librerías de *Apache* [3], en este caso, *commons-collecions, commons-pool* y *commons-dbcp* implementaremos un pool de conexiones en cada uno de los sistemas gracias a la clase *DataSource* incluida en las librerías.

Lo que se consigue utilizando un *DataSource* es un *pool* de conexiones del que los sistemas pueden hacer uso de la siguiente forma. Al iniciar el sistema, se creará el *pool* como se ve en la imagen 3 en la siguiente página, y se generarán distintas conexiones marcadas como libres, esperando a ser utilizadas.

```
BasicDataSource ds = new BasicDataSource();

ds.setDriverClassName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
ds.setUrl("jdbc:sqlserver://" + IP + ":" + PUERTO + ";DatabaseName=" +
BASE_DATOS);
ds.setUsername(USUARIO);
ds.setPassword(CONSTRASENA);
this.origenDatos = ds;
```

Imagen 3. Creación del pool de conexiones

Cuando los sistemas necesiten realizar una petición a la base de datos, solicitarán al *pool* una de las conexiones libres y quedará marcada como que está siendo utilizada, realizará con ella la consulta, y una vez finalizada, se cerrará la conexión. La conexión se obtendrá como se puede ver en la imagen 4.

```
public Connection devuelveConexion() throws SQLException {
        return origenDatos.getConnection();
}
```

Imagen 4. Obtener conexión del pool

Al cerrar la conexión esta no se destruye, sino que volverá al *pool* y se volverá a marcar con el estado inicial, es decir, libre, por lo que las conexiones ni se crean ni se destruyen, sino que se reutilizan.

4.5. Obtención y visualización de la información

El punto clave de los sistemas que se van a desarrollar se basa en la información que se muestra en las interfaces, cuál es la información relevante, de qué forma la obtenemos y como la mostramos.

En fases anteriores se han definido cuáles son los datos más relevantes que se deben mostrar en ambos sistemas, y dónde están localizados. En el caso de los datos acerca de las llantas, se realizará una conexión a la base de datos alojada en el servidor y se extraerán los datos necesarios, que se encuentran en varias tablas de la base de datos. Las consultas utilizadas para extraer la información en ambos sistemas se pueden observar a continuación en las imágenes 5 y 6.

Con la consulta de la imagen 5 se obtendrá la información que hace referencia a la llanta que se encuentra en el bastidor número 19. La posición de este bastidor es la que el operario del puesto de cubrimientos se encarga de trabajar, por lo que extraeremos la información de la llanta que transporta ese bastidor para poder mostrarla en el sistema.

```
String consultaSQL = "SELECT TABLA_LLANTA.NUMPROYECTO, "
    + "TABLA_LLANTA.REFRONAL, TABLA_LLANTA.POLVO,"
    + "TABLA_LLANTA.PULGADAS, TABLA_LLANTA.CUBRIMIENTO "
    + "FROM TABLA_BASTIDOR LEFT OUTER JOIN TABLA_LLANTA ON "
    + "TABLA_BASTIDOR.NUMPROYECTO = TABLA_LLANTA.NUMPROYECTO "
    + "WHERE (TABLA_BASTIDOR.NUMPROYECTO = 10020019) ";
```

Imagen 5. Consulta SQL. Datos de llanta actual

Con la consulta de la imagen 6 obtendremos las referencias de los modelos que están siendo introducidos en los primeros bastidores de la zona de cubrimientos. Esta información se mostrará al operario del puesto de cubrimientos, advirtiendo de un cambio de modelo con antelación.

```
   String consultaSQL = "SELECT TOP 5 NUMPROYECTO, REFRONAL "
       + "FROM TABLA_BASTIDOR LEFT OUTER JOIN TABLA_LLANTA ON "
       + "TABLA_BASTIDOR.NUMPROYECTO = TANLA_LLANTA.NUMPROYECTO "
       + "WHERE (LEFT(TABLA_BASTIDOR.NUMPROYECTO, 4) = 1002) "
       + "ORDER BY TABLA_BASTIDOR_NUMPROYECTO";
```

Imagen 6. Consulta SQL. Datos de llantas entrantes

La información de las llantas que se mostrará en la zona del trasvase se extraerá de la misma forma que en la consulta de la imagen 5, modificando el campo *PVDefinitionID*, que hace referencia a la zona de pintura con los cuatro primeros dígitos, y a la posición del bastidor con los cuatro dígitos siguientes.

En cambio, las referencias de los modelos de llantas entrantes en la zona del trasvase se extraerán de forma distinta, como se explicará en la sección *Problemas encontrados* [*4.6*].

Las imágenes de los cubrimientos se obtendrán de forma sencilla. Las imágenes de los cubrimientos se encontrarán en una carpeta compartida en el mismo servidor en el que se ejecuta la base de datos, por lo que, utilizando la IP de la conexión de la base de datos, el sistema podrá acceder a esta carpeta. Conociendo la URL de la carpeta compartida y el tipo de cubrimiento que lleva la llanta, que se obtendrá de la consulta SQL anterior, se podrán extraer las imágenes necesarias. Un ejemplo de URL de acceso a las imágenes se puede observar en la imagen 7.

```
File ficheroImagen = new
File("\\\\"+BaseDatos.devuelveInstancia().devuelveIP()
                +"\\PIB_Bolas\\imgs\\"+tipoCubrimiento+".jpg");
```

Imagen 7. Obtención de imágenes sobre cubrimientos. Ruta de acceso

Una vez se haya accedido a la imagen del cubrimiento, se almacenará en un objeto tipo *File*, objeto que seguidamente será convertido en un objeto *Image* como se puede ver en la imagen 8, que es el apropiado para poder mostrar la imagen del cubrimiento por la interfaz.

```
        if( ficheroImagen.exists() ){
            imagen = new Image(ficheroImagen.toURI().toString());
        }
```

Imagen 8. Conversión de fichero a formato imagen

Por último, la información relevante a la que debemos acceder son los ficheros PDF. El acceso a estos ficheros se realizará de la misma forma utilizada para acceder a las imágenes de los cubrimientos, pero se tendrá que aplicar un proceso posterior para poder mostrar estos ficheros.

Por tanto, el sistema accederá al servidor donde tenemos alojados todos los ficheros PDF correspondientes a los modelos de llantas que resultan más problemáticos. Estos ficheros se encontrarán en una carpeta compartida a la que los sistemas tendrán acceso a través de la IP y el modelo de la llanta. Un ejemplo de ruta se puede observaren la imagen 9.

```
File fichero = new File("\\\\"
+BaseDatos.devuelveInstancia().devuelveIP() +"\\PIB_Trasvase\\"
+ modelo + "\\" + modelo+".pdf");
```

Imagen 9. Obtención de ficheros ".pdf". Ruta de acceso

De esta conexión se extraerá del servidor el fichero y se almacenará en un objeto de tipo *File*, y es en este punto donde se hará uso de las librerías específicas que nos permitirán utilizar y mostrar los ficheros PDF.

Se cargará el fichero extraído del servidor en un objeto de tipo *PDDocument*, y estando en este objeto, se realizará lo que se llama "renderizar" un fichero, que consiste en digitalizar un fichero del tipo que sea y convertirlo en una imagen.

El resultado de renderizar el fichero PDF se almacenará en un objeto tipo *File* haciendo uso de la clase *ImageIO*, preparada para escribir el fichero renderizado en un fichero con el formato que queramos, por lo que su contenido ahora será compatible para poder transformarlo en un objeto de tipo *Image* al igual que transformamos las imágenes de los cubrimientos para poder mostrarlas en la interface.

El funcionamiento en código de este proceso se puede ver en la imagen 10.

```
PDDocument pdf =
PDDocument.load(fichero,MemoryUsageSetting.setupTempFileOnly());

if (pagina == maximoPaginas) {
   pagina = 0;
}
else if( pagina > maximoPaginas) {
   pagina = pagina % maximoPaginas;
}
/*
  El método "renderImageWithDPI" renderiza la página del pdf
   que le indiquemos y la almacena en un búfer
*/
final BufferedImage renderBufer = new PDFRenderer(pdf).
        renderImageWithDPI(pagina, 80, ImageType.RGB);

pdf.close();

final File ficheroSalida = new File("salidaPDF.jpg");
ImageIO.write(renderBufer, "jpg", ficheroSalida);

final Image imagen = new Image(ficheroSalida.toURI().toString());

renderBufer.flush();
ficheroSalida.delete();
return imagen;
```

Imagen 10. Uso de librerías en ficheros ".pdf"

4.6. Concurrencia con JavaFX

Haciendo uso de dos librerías propias de *JavaFX*, en este caso *Task* y *ScheduledService*, ambos sistemas se ejecutarán de forma concurrente. La concurrencia en ambos sistemas se produce debido a que la ejecución de estos sistemas se realiza en varios hilos de ejecución.

Al comienzo de los sistemas, se ejecutará el método *start()*, que arrancará el hilo de ejecución de *JavaFX*, en el que se crearán las conexiones a las bases de datos, el *ScheduledService,* y las distintas ventanas de la interfaz. Este proceso se puede ver en la siguiente página, en la imagen 11.

```
public void start(Stage primaryStage) throws URISyntaxException,
SQLException, IOException {
this.primaryStage = primaryStage;
crearVentana();
TrasvaseController controlVista = crearVentanaTrasvase();
BaseDatos.crear("", "", "", "", "");

Task<Void> tareaConexion =
BaseDatos.devuelveInstancia().devuelveTareaConectar();

        tareaConexion.setOnFailed(evento -> {
            evento.getSource().getException().printStackTrace();
        });
        tareaConexion.setOnSucceeded(evento -> {
            arrancarServicio( controlVista );
        });
        new Thread(tareaConexion).start();
}
```

Imagen 11. Arrancar hilo de JavaFX

En el momento en que se crea el *ScheduledService*, lo que se está creando es un hilo de ejecución que cada seis segundos creará un *Task*, es decir, otro hilo de ejecución. Estos *Task* que se crean cada seis segundos son los encargados de acceder al modelo y obtener la información necesaria.

Una vez finalizado el método *call()* del hilo *Task*, éste se destruirá automáticamente , y en caso de que todo haya ido correctamente, se almacenará la información obtenida en el hilo de ejecución del servicio, es decir, del *ScheduledService*, cosa que podemos saber gracias al método *onSucceded()* de la clase *Task*. Este proceso se puede ver en la imagen 12.

```
tareaIteracion.setOnSucceeded((evento) -> {
            Llanta llantaActual = ((Iteracion)
evento.getSource().getValue()).getLlantaActual();
            List<Llanta> llantasEntrantes = ((Iteracion)
evento.getSource().getValue()).getLlantasEntrantes();
            Image imagenDefecto = ((Iteracion)
evento.getSource().getValue()).getImagenDefecto();

            if(this.modeloLlanta.equals
                      (llantaActual.getModeloLlanta())) {
                paginaActual++;
            }
            this.modeloLlanta = llantaActual.getModeloLlanta();

            mostrarDatos(llantaActual, llantasEntrantes,
                                       imagenDefecto);
});
```

Imagen 12. Iteración con resultado correcto

Para terminar, desde el hilo del servicio, se mandará al hilo de ejecución de *JavaFX* que muestre por pantalla la información que se ha obtenido, esto se realizará ejecutando los métodos del controlador de la vista dentro del método *Platform.runLater()*. La forma de utilizar este método se puede ver en la siguiente página, en la imagen 13.

```
private void mostrarDatos(Llanta llanta, List<Llanta> llantas, Image
imagenDefecto) {
      Platform.runLater(() -> {

          this.controlVista.setDataLlantaPrincipio(llanta);
          this.controlVista.setDataLlantasEntrantesTrasvase(llantas);
          this.controlVista.cargarPDF(imagenDefecto);
          }
      );
```

Imagen 13. Uso de Platform.runLater

El sistema se ejecutará en diversos hilos de ejecución. Por una parte, se creará el hilo de ejecución de *JavaFX* al ejecutar el método *start(),* que se encargará de mantener las vistas y la conexión operativas, y de crear el *ScheduledService.*

Al ejecutar el método *start()* del ScheduledService se creará un hilo de ejecución que se encargará de crear *Task* cada seis segundos, recoger la información de estos *Task* cuando acaben su proceso, y mandar al hilo de *JavaFX* la información para mostrala en pantalla.

Por último, los *Task* creados por el *ScheduledService* ejecutan su método *call()*, en el que accederán al modelo para obtener los objetos que se mostrarán posteriormente. Al finalizar este método, se cerrarán automáticamente.

4.7. Problemas encontrados

Durante todo el proceso se han encontrado diversos problemas a los que se ha tenido que encontrar una solución acorde tanto al correcto desarrollo de los sistemas, como al futuro mantenimiento, respetando las infraestructuras y los sistemas que funcionan actualmente en la empresa.

1. El primer problema aparece en el momento en que no podemos modificar ni el esquema, ni el tipo de datos que se almacenan en esta base de datos. Se hace uso de un campo inutilizado dentro de la base de datos, de forma que no se modifica la infraestructura de la base de datos y completamos la información necesaria para mostrar en los sistemas.

2. El periodo de tiempo que se ha tenido que invertir en el análisis y búsqueda de distintas librerías que nos permitieran poder trabajar en el sistema del trasvase con ficheros PDF. Esto ha supuesto un problema mayor del que nos esperábamos, por la escasez de librerías de código abierto, y por el tiempo invertido en su aprendizaje, lo que ha supuesto un retraso en la finalización del desarrollo del proyecto.

3. Los operarios de los puestos de la zona de trasvase se encargan de las llantas que trasportan los bastidores del final de esta zona, más concretamente, los bastidores “814”, “815”, “904” y “905”. Esto nos lleva a las llantas que se muestran en la parte superior de la interfaz en estos sistemas, donde se indica cuáles van a ser las siguientes llantas que entrarán en esta zona.

   En el sistema de cubrimientos las llantas mostradas en esos campos son las primeras de la zona, por lo que con una consulta haciendo uso de la función “TOP”, obtenemos los datos que necesitamos del principio de la base de datos, pero en este

caso, como las llantas que se deben mostrar no están ni al comienzo ni al final, debemos encontrar la forma de mostrarlas.

Las llantas se encuentran en las posiciones “793”, “794” y “795”, por lo que la primera idea es hacer uso de la función *ROW_NUMBER* con la que se pueden obtener conjuntos de filas de una base de datos indicando el comienzo y fin del conjunto, pero, como la versión de base de datos que se ejecuta es anterior a *SQL Server 2008*, no se puede hacer uso de esta función y se necesita encontrar una alternativa.

La solución adoptada ha supuesto obtener las tres filas de datos a raíz del número de bastidor, haciendo uso de operadores lógicos “OR” dentro de la cláusula *WHERE* de la consulta realizada. De esta forma y como se puede ver en la imagen 14, se puede obtener el conjunto de filas que se necesitan, acoplándonos a la versión de la base de datos que se utiliza.

```
"WHERE (REFRONAL = 10030793) OR (REFRONAL = 10030794) OR "
+ "(REFRONAL = 10030795) ORDER BY REFRONAL";
```

Imagen 14. Alternativa para ROW_NUMBER

4. El tiempo que se ha tardado en definir un formato con el que los encargados de la información de las llantas puedan introducir los cubrimientos en cada modelo, ha sido mayor de lo esperado. Ha sido un problema, ya que cada encargado utilizaba un formato distinto, lo que provocaría que los sistemas no mostrarían la información correcta a raíz de una mala información.

5. Durante la fase de pruebas, nos encontramos con que ambos sistemas consumen gran cantidad de memoria con el paso del tiempo. Los sistemas se ejecutan de forma correcta durante dos horas y media aproximadamente, y comienzan a observarse problemas en la interfaz de ambos sistemas. Se analizará y rediseñará el sistema para solventar el problema.

5. Pruebas

- **Fecha 13/03/2018 – información de las llantas**
  Se realizan pruebas haciendo uso de la base de datos SQL de la que se extraerá la información para ambos sistemas. De estas pruebas se obtienen las consultas necesarias para obtener la información que se mostrará en cada uno de los sistemas.

- **Fecha 03/04/2018 - datos en el sistema de cubrimientos**
  Haciendo uso del sistema EISENNMAN se comprueba si los datos que se muestran en el sistema de cubrimientos son los correctos teniendo en cuenta el puesto de bastidor y las llantas que se transportan. Se modifican los datos de las llantas durante la ejecución del sistema para comprobar si cambian durante la ejecución. Se valida la prueba al ver que se muestran los datos correctos.

- **Fecha 04/05/2018 - prototipo sistema de cubrimientos**
  Se realizan pruebas de visualización en una pantalla de 50'' ejecutando el sistema de cubrimientos para comprobar que todos los elementos de la interfaz tienen el tamaño adecuado. Tras la visualización, se observa que se deben ajustar el tamaño de los textos y el tamaño de la imagen de cubrimiento.

- **Fecha 08/05/2018 – datos en el sistema de defectos**
  Se comprueba en el sistema de los puestos del trasvase que se muestra la información correcta de las llantas, de la misma forma que el sistema de cubrimientos, haciendo uso del sistema EISENNMAN.
  Además, en este sistema se comprueba que los ficheros PDF que se muestran son los correspondientes al modelo de llanta que se está transportando, así como que en el caso de que el bastidor esté vacío o no exista documentación PDF sobre algún modelo de llanta se muestre el PDF de información general.
  Los resultados sobre la información en este sistema son satisfactorios, viendo que la información es correcta y se muestran todas las páginas de los PDF de los modelos.

- **Fecha 09/05/2018 - prototipo sistema de defectos**
  Se realizan pruebas de visualización en una pantalla de 50'' ejecutando el sistema de defectos del trasvase para comprobar que todos los elementos de la interfaz tienen el tamaño adecuado. Tras la visualización, se observa que se deben crear los PDF con el mayor tamaño posible

- **Fecha 29/05/2018 – pruebas de ejecución**
  Se realizan pruebas de ejecución de ambos sistemas haciendo uso del software *jconsole*. Durante las pruebas de ambos sistemas, nos encontramos con que los sistemas dejan de funcionar correctamente y se debe estudiar una solución, ya que, ejecutando ambos sistemas con 4GB de memoria dedicados para la máquina virtual de Java, los sistemas fallan transcurridas dos horas de ejecución.

- **Fecha 04/06/2018 – pruebas de ejecución bajo nuevo diseño**
  Se realiza una reestructuración del código de ambos sistemas y se corrigen diversos problemas que provocaban la caída de los sistemas. Una vez solucionados estos

problemas, se vuelven a ejecutar los sistemas y se analizan con *jconsole*, y esta vez se puede observar que los sistemas no consumen más de 150MB, como se puede observar en la imagen 15, en la que aparece la monitorización del sistema del trasvase durante casi seis horas sin detenerse.

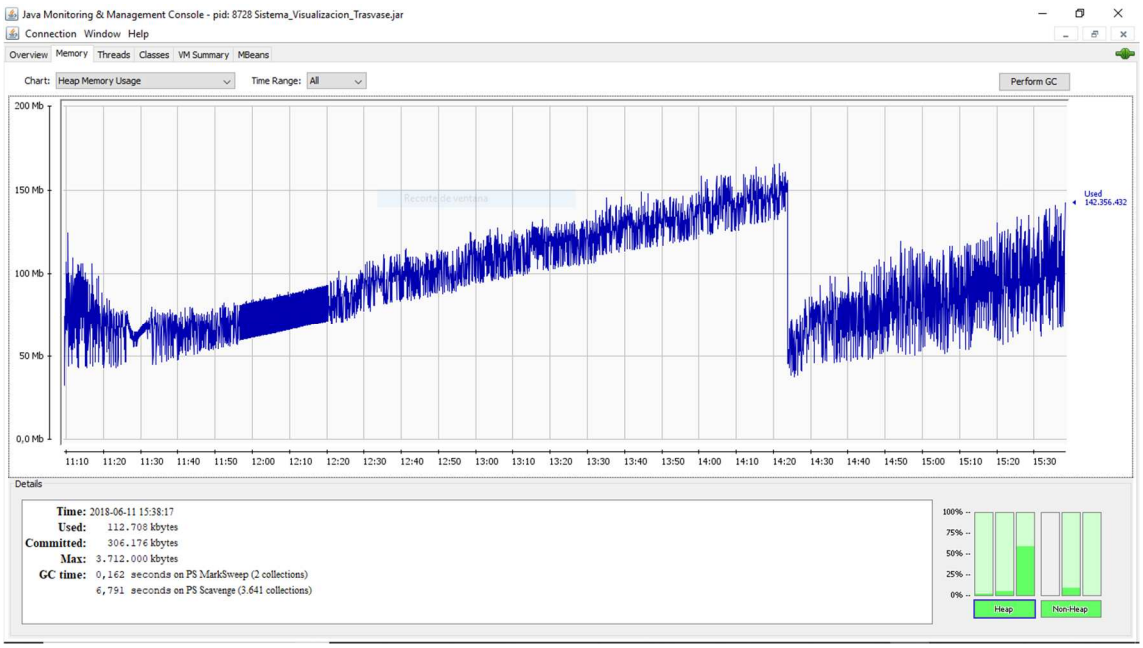


Imagen 15. Monitorización del sistema del trasvase

6. Mantenimiento de los sistemas

Dentro del desarrollo, una parte fundamental es identificar de qué forma y que procedimientos se deben seguir para que los usuarios que vayan a interactuar con el sistema conozcan cómo modificar los datos que se deben mostrar, por lo que se describe de qué manera se debe mantener la información que se utiliza en ambos sistemas.

6.1. Información de las llantas

A través del sistema EISENMANN, los encargados de mantener la información actualizada, pueden modificar cualquier campo incluido el tipo de cubrimiento recientemente añadido, y cualquier campo que se modifique en ese sistema se actualizará en la base de datos al momento, por lo que nuestros sistemas también mostrarán información actualizada.

6.2. Imágenes de los cubrimientos

Respecto a las imágenes de los cubrimientos, en caso de querer modificarlas porque se ha introducido un nuevo cubrimiento en la fábrica, o se ha eliminado algún cubrimiento actual, solo se tendrán que actualizar las imágenes actuales que están en la carpeta compartida del servidor, respetando el formato de nombre que se ha especificado y la ruta de la carpeta.

6.3. Ficheros PDF de defectos

De la misma forma ocurre con los ficheros PDF, la información que se quiera mostrar se preparará utilizando la herramienta PowerPoint de Microsoft Office o cualquier software que permita crear PDF, y una vez preparada, se almacenará en el servidor desde el que se toma la información. Del mismo modo que las imágenes, se deberá respetar la forma de nombrar estos ficheros, que en este caso supone nombrar el fichero con los cuatro números centrales del modelo de la llanta.

6.4. Fichero de conexión a la base de datos

Otros datos que tienen la posibilidad de modificar son los datos de conexión a la base de datos. Los sistemas obtienen los datos de conexión necesarios (IP, puerto, nombre de la base de datos, usuario y contraseña) desde ficheros en texto plano. En caso de que la información se migre a otro servidor, simplemente se debería modificar el contenido del fichero y los sistemas seguirán funcionando de forma correcta. Este fichero solo se debe modificar en caso de migración de información de la base de datos.

6.5. Ficheros de las interfaces

Como último punto, más enfocado a usuarios con conocimientos más concretos de informática, las interfaces de ambos sistemas se han realizado generando ficheros XML, buscando que, en caso de que se desee modificar las interfaces de los sistemas, se pueda realizar de forma fácil y sin necesidad de conocer tecnologías concretas.

Los ficheros escritos en lenguaje XML se pueden modificar haciendo uso de un editor de texto común, del software *JavaFX Scene Builder,* un software intuitivo y sencillo de utilizar con el que agilizar las modificaciones, o cualquier software que esté pensado para realizar este tipo de documentos.

7. Implantación

Una vez desarrollados ambos sistemas, el último paso será colocar ambos sistemas en los puestos de trabajo que se han detallado previamente.

Se prepararán tres equipos de tipo *thin client* en los que se ejecutarán los sistemas, uno para el sistema de cubrimientos, y los dos restantes para los puestos del trasvase. Estos equipos se configurarán de forma que al iniciarlos se ejecute directamente los sistemas desarrollados, ya que se utilizarán únicamente para ejecutar estos sistemas.

En estos equipos se instalará como sistema operativo una versión libre Linux, con lo que se evitarán gastos de licencias, y una versión de Java 8 o superior, ya que es necesario para poder ejecutar todos los componentes de los sistemas.

Cada uno de los equipos se colocará en la parte trasera de los monitores, de esta forma los equipos solo necesitarán una toma de red y alimentación, y la salida de imagen de los equipos estará conectada al puerto del monitor.

Los monitores se colocarán en los puestos de trabajo a una distancia adecuada, de forma que las pantallas estén centradas a los operarios que estén trabajando en los puestos. En caso de que fuera necesario, para colocar los monitores se utilizarán soportes de anclaje.

8. Conclusiones

Al tratarse de un entorno real el proyecto se debe diseñar pensando en los requisitos del entorno, es decir, que los sistemas se ejecuten sin que tengan errores, que no existan problemas de datos y que den a los operarios la información necesaria para realizar sus funciones de la mejor forma posible.

Respecto a la información que se utiliza en el sistema del trasvase se ha podido comprobar la poca variedad de librerías que existen, al menos para Java, para poder gestionar y manipular ficheros PDF de tipo *Open source.* Es posible que en otras tecnologías existan más posibilidades, aunque en nuestro caso se han podido desarrollar los requisitos enfocados hacia este tipo de ficheros gracias a las librerías *Open source* de Apache.

Respecto a la información general de las llantas, que está almacenada en la base de datos, nos hemos encontrado con una versión de base de datos que nos ha obligado a encontrar formas alternativas de obtener cierta información, como, por ejemplo, conjuntos de filas para el sistema del trasvase.

Al ser una versión de hace años, no soporta el uso de ciertos métodos, como *ROW_NUMBER*, por lo que se ha buscado una alternativa con la que poder obtener la misma información que se obtendría haciendo uso de esa función, en este caso, hacer uso de operadores lógicos *OR* en la consulta SQL nos permite seleccionar las distintas filas de las que queremos la información y obtener el mismo resultado que al hacer uso de la función *ROW_NUMBER*.

Las conexiones a una base de datos pueden resultar un problema muy grave en el rendimiento si no se tratan de la forma adecuada, por eso, en nuestros sistemas las conexiones se han implementado haciendo uso del patrón *Singleton*, con el que nos aseguraremos de que en ambos sistemas solo existirá una conexión activa, y no se puedan realizar más instancias.

Además, para hacer más efectivo el uso de la base de datos y de las conexiones en nuestros sistemas, se ha implementado un *pool* de conexiones haciendo uso de distintas librerías de

Apache. Esto nos permite tener un conjunto de conexiones esperando a ser utilizadas, que los sistemas solicitarán cuando las necesiten, se usarán, y al cerrar la conexión, esta quedará en su estado inicial, es decir, esperando a que sea solicitada para realizar nuevas consultas.

Por último, una buena práctica al realizar sistemas en los que las tecnologías utilizadas son Java y JavaFX, consiste en hacer uso del hilo de ejecución de JavaFX únicamente para los mecanismos relacionados con la interfaz y operaciones sencillas, es decir, cargar las distintas interfaces o actualizar los datos que se encuentran en las mismas. Por ello, ejecutaremos las funciones encargadas de modificar interfaz dentro del método *Platform.runLater()*.

8.1. Mejoras de futuro

Los sistemas se han desarrollado en base a los requisitos solicitados por el cliente, en este caso la empresa RONAL Ibérica, lo cual no quiere decir que estos sistemas se han desarrollado de forma que no se pueda extender más sus funcionalidades.

En este apartado se incluyen posibles mejoras con las que los sistemas serían más eficaces y mostrarían más información, por lo que serían más completos de cara a la empresa, ya no al operario concreto de los puestos de trabajo.

- De cara al sistema de cubrimientos, al igual que se muestra información de las llantas y los cubrimientos que deben llevar, se podría mostrar información a nivel general, como normativas o políticas propias de la empresa. Esto se podría realizar haciendo uso de la tecnología que ha sido utilizada en el sistema de los puestos de trasvase, es decir, almacenando la nueva información en formato PDF, y con las librerías utilizadas ya anteriormente, renderizar los ficheros y mostrar estas políticas cada cierto tiempo. De esta forma los operarios estarían más concienciados, y tendrían información sobre qué acciones realizar en caso de ciertas situaciones.

- Otra mejora que se considera importante, se enfoca en compenetrar los sistemas con los que la instalación de la cadena funciona con los desarrollados en este proyecto. La forma en que se podría hacer uso de la información consistiría en lo siguiente. En el momento en que la cadena sufriera una parada, ya sea por una avería o por cualquier otro motivo, reconocer de alguna forma que se ha realizado esa parada, y en ese momento, nuestros sistemas deberían detener la ejecución. Con esto se conseguiría reducir el uso del servidor y de la base de datos, siendo más eficiente el uso de estos sistemas y más completos, ya que se estaría aprovechando información de sistemas actuales de gran importancia para el proceso.

- Una mejora menor dentro del sistema de cubrimientos sería completar la información, incluyendo además del cubrimiento que ha de llevar el modelo, una imagen de la llanta que se está procesando, de forma que estaríamos mostrando información más completa. Con esto conseguiríamos que los operarios inexpertos realizarían sus funciones de forma más sencilla.

8.2. Agradecimientos

Al haberse realizado en un entorno real, he podido vivir una experiencia de trabajo real, donde lo relevante no recae únicamente en las ideas que nosotros mismos tengamos acerca de cómo realizar un proyecto, que tecnologías usar, o cómo nos imaginamos el resultado final de un sistema.

En entornos reales de trabajo, es muy importante en primer lugar conocer qué sistemas están funcionando, bajo qué infraestructuras, qué tipo de bases de datos se están utilizando, y, sobre todo, si se puede realizar alguna modificación en cualquiera de estos puntos, debido a que, por norma general, hay que integrarse de la mejor forma posible al funcionamiento actual del entorno.

En segundo lugar, este proyecto ha resultado muy satisfactorio, primero, por el aprendizaje obtenido de realizar el periodo de análisis de tecnologías desconocidas previamente, y segundo, el conocimiento sobre librerías de código abierto que nos pueden ser útiles en futuros proyectos, pero, sobre todo, por el progreso continuo del proyecto, que ha podido ser posible gracias a la colaboración de todo el equipo de RONAL Ibérica.

Destacaría como punto más importante dentro de esta experiencia la influencia del factor humano. Durante todo el transcurso del proyecto, desde que se comenzó con las conexiones a la base de datos, las modificaciones en los formatos de la información, hasta el desarrollo completo del proyecto, toda la ayuda, conocimientos de la planta, opiniones personales y *feedback* obtenidos de todo el personal que se ha visto involucrado en el proyecto han sido de vital importancia en este proyecto.

Por todo esto, aparte de haber realizado ambos sistemas con los requisitos que se solicitaban y haber obtenido nuevos conocimientos, el punto más relevante que se puede destacar dentro de la experiencia del proyecto realizado es la valoración y el peso que puede llegar a tener el factor humano en su conjunto y colaboración dentro de un proyecto de este campo.

Por ello quisiera mostrar mis agradecimientos personales al personal de RONAL que ha estado involucrado, en especial a Andrés Górriz, Jesús Bella y Carlos Gómez; tutores académicos del proyecto Carlos Catalán y Jesús Gallardo; y, por último, de forma más personal, a Samuel Salvatella y Sergio López, estudiantes de la EUPT, por el apoyo y la ayuda durante la realización del proyecto.

9. Referencias

[1] Class Scheduled Service. API de JavaFX. Disponible en: https://docs.oracle.com/javase/8/javafx/api/javafx/concurrent/ScheduledService.html

[2] Class Task. API de JavaFX. Disponible en: https://docs.oracle.com/javase/8/javafx/interoperability-tutorial/concurrency.htm

[3] Pool de conexiones usando DataSource. Disponible en: http://chuwiki.chuidiang.org/index.php?title=Pool_de_conexiones

[4] JavaFX con Scene Builder. Disponible en: http://code.makery.ch/library/javafx-8-tutorial/

[5] Tcl/Tk. Disponible en: https://www.tcl.tk/

[6] IDE Glade. Disponible en: https://glade.gnome.org/. https://python-gtk-3-tutorial.readthedocs.io/en/latest/builder.html

[7] Poppler. Disponible en : https://stackoverflow.com/questions/18381713/how-to-install-poppler-on-windows. https://github.com/wbsoft/python-poppler-qt5. https://people.freedesktop.org/~aacid/docs/qt5/.

[8] Apache PdfBox. Disponible en: https://pdfbox.apache.org/

[9] jPDFViewer. Qoppa Software. Disponible en: https://www.qoppa.com/pdfviewer/

[10] jPedal. IDR Solutions. Disponible en: https://www.idrsolutions.com/jpedal/

[11] iText. Disponible en: https://itextpdf.com/

[12] Uso de HostServices para mostrar PDF en JavaFX. Disponible en: https://stackoverflow.com/questions/36960844/how-to-open-a-pdf-file-javafx