Descripción TFG:

El Proyecto viene a suplir una necesidad de las empresas de ámbito industrial de hoy en día. Evaluar el coste energético por proceso, con el fin de pasar de un gasto estructural a uno en concreto, aportando un gran valor de ámbito estratégico.

En concreto este proyecto se enmarca en una empresa de elaboración de piensos. El cliente debe poder conocer el coste energético por tonelada, esto les permitirá comparar, sumando el coste energético al coste de las materias primas, el coste de fabricación con el rendimiento en las reses. Cuándo se habla de materias primas, nos referimos a unidades mínimas de información para garantizar la trazabilidad.

El proyecto tiene dos grandes ramas:

1. Elaboración de un cliente con parte gráfica que pueda procesar los datos de los que ya dispone la empresa, para poder obtener un consumo energético por lote, entendiendo como lote: unidad de un producto asociada a un proceso, garantizando la trazabilidad de un proceso.
2. Elaboración de una herramienta de generación de informes, que les facilitará a la empresa, de una forma más intuitiva, la información ya mencionada sobre el consumo energético.

Todo el proyecto va orientado a la empresa Coabre, con origen en Silla (Valencia). Se trata de una empresa a la que el ssitema de automatización ha sido realizado por la empresa donde realizo mi FORTE. Por lo que es muy conveniente que sea esta empresa la que realice la nueva funcionalidad.

Topología de la empresa.

La empresa se inspira en un modelo cliente - servidor.

* Dentro de el servidor se encuentran dos subsistemas de gran calado e la empresa SUITER y REPORTER.
* De la parte del cliente tenemos browser, cliente de fábrica y HMI (Human Machine interface). Dentro de HMI está:
  + Piquera
  + Truck
  + Premezcla
  + Dos. Manual
  + Trasvases

La empresa dispone de un servicio de automatización de procesos llamado SUITER, este sistema. Este sistema se encarga de automatizar el proceso industrial, como podría ser gestión de molinos, gestión de la entrada de materias en los silos. Todo esto requiere como es de prever una gestión en tiempo real.

Queda destacar que se tratan de bases de datos MySQL.

Al suiter le da soporte una base de datos nombrada con el nombre de la fábrica(Coabre). Esta base de datos contiene registros dónde SUITER accede de forma concurrente.

El REPORTER es el sistema encargado de presentar una página web, desde dónde poder gestionar el proceso productivo, este sistema dispone de dos bases de datos, Coabre reporter, que da soporte al servicio web, además de una base de datos replicada de la de Coabre a la que accede para realizar consultas pero esta vez no de tiempo real, ya que una página web no es un sistema de riesgo, además de escribir los datos de entrada cómo puede ser la entrada de materias primas, las mezclas... El motivo de una base replicada viene por el segundo objetivo del REPORTER que es obtener informes. Estos informes suelen tener una grán carga de datos, por lo que ralentizará al SUITER en el que una desviación en segundos puede suponer grandes desviaciones en el proceso. imaginémonos que queremos vaciar un silo si tardamos un segundo más en cerrar la compuerta la fórmula variaría en kilos, algo que no es admisible. Al final tendremos tres bases de datos: coabre\_server como slave, coabre como master y reporter\_coabre para la gestíon de la pagina web.

Mi proyecto afectarí en muy poca medida al SUITER, ya que dispone de todos los datos que se van a necesitar aunque si hay que dominar la comunicación con el REPORTER. Se trataría de añadir una nueva funcionalidad al REPORTER.

Al ser el Reporter el que administra la página web también entra en la parte de Browser, el Browser se trata de servidor web Apache Tomcat. En esta parte la comunicación se realiza mediante peticiones ajax.

En el Browser corre en un tomcat como un contenedor de servlets. Toda la interfaz de interaccion con el cliente esta realizada con java script, jsp, html, jtls y css para darle estilo. El servidor estaría realizado básicamente en java.

los estados por los que pasa una consulta del cliente son los siguientes. El cliente tiene conocimiento de todos los servlets almadenados por el contenedor tomcat que se tratan de las distintas opciones que puede realizar. A partir de ahi se manda una petición request http que resuelve el web.xml de tomcat devolviendo la ruta del servlet solicitado. Pero antes de devolver la ruta y proceder a resolver la petición tiene que pasar un filtro basado en dos partes: inFIlter y genericFilter. El inFilter recibe el ido de ususario y comprueba que tiene acceso a esa opción de forma que filtra los permisos. El genericFilter establece los parámetros de sesión. Una vez pasa el filtro entramos a resolver la petición del cliente.

El servidor trabaja basándose en el modelo en tres capas de gestión de base de datos: persistencia, negocio y presentación. Presentación que corresponde a la interacción directa con el clientes son los servlets, tras ellos está el lib que corresponde a la capa de lógica de negocio y el dao (minúscula) y Dao (mayúscula) que se encargan de la interaccion de la base de datos, de forma más directa el Dao que genera las consultas SQL.

Siguiendo el diagrama de estados en el que nos habíamos quedado, al acceder al servlet se genera un lib que contiene todas las funciones que puede realizar el cliente. Estas funciones son realizadas por el dao que pide al Dao que realice la conexión con la base de datos. El servlet se encarga de añadir este lib a unas variables de contexto a las que accede el cliente. A parte existen peticiones ajax que genera el cliente.

El lib es el caota los datos de el formulario, el servlets carga los privilegios y las opciones, y el dao es el que contine la lógica de negocio. Jquery

Descripción consumo energético.

Como inicio convendría saber a grandes rasgos en que entorno nos estamos moviendo. Este proyecto se enmarca en el ámbito de la fabricación de piensos de una empresa. Se trata de un entorno industrial donde necesitamos asegurar una trazabilidad de todos los procesos que se realizan. Como primera aproximación, el proceso de fabricación de piensos sería, la entrada de materias primas y la salida de piensos o materias. Para poder hacer un seguimiento de el proceso productivo, lo dividimos en lotes. Los lotes nos permiten seguir la trazabilidad de las materias primas.

Pongamos un ejemplo: A la empresa le entra 1000Kg de cebada, entonces esa cebada la anotaremos como LE1 (Lote de entrada 1) este lote nos permite hacer el seguimiento desde que sale de el camión hasta que se almacena para posteriormente ser usado. Parte de este lote podría ser usado para mezclarlo con otros lotes y formar un pienso, esto se llama dosificación. Una vez el lote sale de su almacenamiento pasaría a ser LF1 (Lote de fabricación 1) que puede contener otros lotes de entrada. El proceso seguiría de la misma forma con todos los procesos de los que disponga la fábrica.

A partir de aquí, conociendo lo que es un lote, pasamos a describir el objetivo de el proyecto. El objetivo es obtener el consumo energético por cada lote. Esto se dividirá en dos partes: Elaboración de un cliente que procese los datos de los que ya se dispone para presentárselos a el cliente en una página web y el desarrollo de una herramienta de generación de informes.

Elaboración del cliente:

Tendrá dos tareas principales, mostrar el consumo energético por lote donde se puede ver un cronograma de los motores que intervienen en un lote y comparar los resultados obtenidos con el consumo real de un contador fiscal con el fin de ajustar el error cometido obteniendo consumo auxiliar y consumo ponderado.

Al poder acceder a la base de datos de el servicio de automatización de la empresa, podremos obtener información relevante a los lotes en tiempo real. También dispone de el consumo energético de cada motor que intervenga en el proceso productivo.

Para realizar esta tarea necesitamos el soporte de una base de datos, que en nuestro caso es MySql, crearemos tres nuevos registros:

* reg\_logs
* reg\_consumo\_cuartohorario
* reg\_orden

Haremos uso de las tablas:

* reg\_contaodres
* str\_elementos
* tr\_lotes
* tr\_rel\_lotes

Desarrollo y funcionamiento de las tablas

El cliente al realizar la conexión con el servidor obtiene los objetos nodo que intervienen en la conexión, estos objetos contienen los datos de los motores y de los lotes. La comunicación se realiza por eventos.

En logs se registrará los cambios de estado de los motores asociados a un lote, junto con una marca de tiempo para registrar en qué momento se produjo el cambio de estado. Los motores pueden estar en tres estados: arranque, paro y alarma. A efectos de el cálculo de el consumo energético, una parada y una alarma significan los mismo: el motor deja de funcionar por lo que dejaría de consumir. Pero sería interesante mostrar un cronograma de los logs incluyendo los momentos en los que se han producido alarmas. El cliente captura eventos generados por el servidor SUITER, cada vez que se genera un cambio de estado en los motores, se encolarán usando el modelo de concurrencia productor – consumidor. El consumidor se trata de un proceso en nuestro programa cliente. El factor de degradación es un valor que inicialmente se pondrá a 1, pero que no va a ser usado en este proyecto, tan solo con vistas a futuro. La potencia de degradación y la potencia en vacío será utilizada para el cálculo de el consumo energético. Inicio es la fecha en la que se instaló el motor.

El modelo entidad relación sería:

* reg\_logs(codigo, id\_elemento, id\_lote, arrancado, alarma, potencia\_nominal, potencia\_vacio, factor\_degrdacion, incio, ts)
* reg\_logs.arrancado. reg\_logs.alarma reg\_logs.ts NOT NULL
* reg\_logs.arrancado {0,1}
* reg\_logs.alarma {0,1}

reg\_consumo\_cuartohorario

Dicha tabla es necesaria para poder obtener el consumo de cada lote cada 15min, esta decisión se debe a que los contadores registran la energía real consumida cada cuarto horario.

Esta tabla nos permitirá comparar el consumo obtenido con el consumo de los contadores, con el fin de ajustar el coste obtenido ponderando de todos los lotes en un cuarto de hora.

Antes de realizar la comparación con el contador tendremos que rellenar la tabla con los kw de carga y los kw en vacío, de esto se encargará un proceso del Cliente. Pero surge un problema con el consumo en vacío producido ya que las empresas suelen dejar en cedidos los motores antes que pararlos y arrancarlos debido al gran consumo de él arranque, aunque siempre hay un tiempo límite en el que el motor puede estar en vacío. ¿cómo detectamos que está en vacío?, pues cuando id\_lote esté a null y arrancado en 1 (true)

¿A quién debemos imputarle el consumo en vacío a el lote que sale o a el que entra? Pongamos un ejemplo donde se pueda ver más claramente:

Imaginemos nos que el molino está moliendo un LF1(Lote de fabricación) pero sale de el molino, el motor del molino sigue arrancado y hasta dentro de 10 min no llega el otro lote LF2. En estos casos hemos tomado la decisión de imputar el consumo en esos 10min a el lote entrante LF2 ya que si no estuviera en cola el motor acabaría parando. También se podría producir las sucesión de estados: estado1 🡪 (id\_lote= null, arrancado=1) , estado2 🡪 (id\_lote= null, arrancado=0), esto indica que el lote ha salido, no va a entrar ningún otro lote y el motor supera su tiempo de arrancado en vacío; ese tiempo se le imputaría a el lote saliente LF1.

Además tiene una última columna que dispone de los kw reales, que contiene el consumo real obtenido gracias a la comparación con los contadores, de lo que hablaré más adelante.

Modelo entidad relación de reg\_consumo\_cuartohorario

reg\_consumo\_cuartohorario(codigo, id\_lote, kw\_carga, kw\_vacio, kw\_reales, ts)

reg\_consumo\_cuartohorario.id\_lote NOT NULL

reg\_contadores

En reg\_contadores obtenemos el consumo por cada contador en una fecha, cada cuarto de hora, pero todos los contadores no nos interesan, para discriminar entre contadores necesitamos la tabla str\_elemento que contiene el tipo de contador.

De la tabla reg\_contadores necesitaremos reg\_contadores.id\_contador, reg\_contadores.fecha, reg\_contadores.eae (consumo) y de str\_elemento será necesario str\_elemento.codigo y str\_elemento.nombre.

Modelo:

reg\_contadores(id\_contador, fecha, eae)

str\_elemento(codigo, nombre)

reg\_contadores.id\_contador 🡪 str\_elemento

Una vez obtenido el consumo por cada lote en un mismo cuarto horario y el consumo real en un cuarto horario, podemos ponderar. Sumando el consumo de cada lote tenemos el consumo obtenido en un cuarto de hora x que lo comparamos con el consumo real de los contadores que intervienen, obtendremos una desviación que la tendremos que dividir entre los lotes que intervienen en ese cuarto horario ponderando. Todo este cálculo lo realizará un solo proceso.

El resultado de esta operación irá a reg\_consumo\_cuartohorario.kw\_reales

Finalmente queremos obtener el consumo total de cada lote junto con el heredado de otro lote, este consumo heredado se debe a las dosificaciones. Pongamos un ejemplo:

Supongamos que tenemos un lote de maíz LE1 que dosifica 5 de sus 10 kg a un LF1 entonces el lote LF1 tendrá el consumo en procesar esos 5Kg más el consumo heredado de LE1 que es el consumo de la entrada de ese maíz dividido entre dos.

Por lo tanto, del modelo entidad relación de reg\_orden es:

reg\_orden(codigo, id\_lote, kw\_proceso, kw\_heredado,ts)

reg\_oreden.id\_lote, reg\_oreden.ts NOT NULL

reg\_oreden.id\_lote UNIQUE

Pero surge una posible excepción, el lote que ha dosificado puede cambiar, lo que hace que el consumo heredado cambie, además ese consumo puede cambiar en los lotes que dosificaron a el lote que ha dosificado, es decir, hacia arriba en la gerarquía. También cambia en los hijos de este Lote.

Generalmente se genera una jerarquía en modo de árbol, dónde la raíz empieza la dosificación. Entonces los pasos a seguir serían los siguientes, cada cierto tiempo se revisa el ts de el lote que ha dosificado, si cambia habrá que cambiar el el consumo por kilo de todos los nodos del árbol que le dosificaron hacia arriba hasta llegar a la raíz, en caso de que el que cambia es la raíz no habrá que recorrer el árbol hacia arriba. Además, hay que cambiar el consumo hacia abajo hasta llegar a las hojas, en el caso de que sea una hoja, solo se cambia ella y finalizaría.

Una vez tengamos todos estos datos, lo más interesante sería ver los lotes más próximos a la venta ya que contienen el consumo final y no van a tener que heredar.

Proceso consumidor

Proceso de división de eventos en cuartos horarios

Proceso de cálculo de kw reales

Proceso de gestión de reg\_orden.