



UNIVERSIDAD DE LA RIOJA

TRABAJO FIN DE GRADO

Título
Diseño y reimplementación de partes de producción
Autor/es
Mireia Otín Ros
Director/es
Juan Félix San Juan Díaz
Facultad
Facultad de Ciencias, Estudios Agroalimentarios e Informática
Titulación
Grado en Ingeniería Informática
Departamento
Curso Académico
2014-2015



Diseño y reimplementación de partes de producción, trabajo fin de grado de Mireia Otín Ros, dirigido por Juan Félix San Juan Díaz (publicado por la Universidad de La Rioja), se difunde bajo una Licencia Creative Commons Reconocimiento-NoComercial-SinObraDerivada 3.0 Unported. Permisos que vayan más allá de lo cubierto por esta licencia pueden solicitarse a los titulares del copyright.

© El autor
© Universidad de La Rioja, Servicio de Publicaciones, 2015
publicaciones.unirioja.es
E-mail: publicaciones@unirioja.es



UNIVERSIDAD DE LA RIOJA

Facultad de Ciencias, Estudios Agroalimentarios e Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

Diseño y Reimplementación de Partes de Producción

Alumno:

Mireia Otín Ros

Tutores:

Juan Félix San Juan Díaz

Logroño, julio de 2015

RESUMEN

El objetivo de este Trabajo de Fin de Grado (TFG) es realizar una implementación y diseño nuevos de la aplicación Partes de Producción. Esta herramienta se utiliza en la fábrica para la gestión y control de la producción de cigarrillos e incidencias que puedan ocurrir durante los distintos turnos. También relaciona dichas mediciones con los correspondientes operarios que trabajan en cada máquina a lo largo del turno. Toda esta información que se recoge a través de este programa proporciona una serie de estadísticas que ayudan a reestructurar labores y horarios con el fin de optimizar el rendimiento de las máquinas. Resultan esenciales para conseguir alcanzar los objetivos de producción. Parte de estos datos son incluidos en informes que se elaboran para informar a Imperial Tobacco acerca de la productividad de la fábrica.

Se pretende realizar un estudio de la aplicación existente para detectar posibles mejoras en cuanto a diseño de interfaz de usuario y estructura de clases. La implementación de la herramienta se realizará en la plataforma .NET ya que es la escogida por Imperial para el diseño de otro tipo de aplicaciones.

ABSTRACT

After one year of intership in Imperial Tobacco Company Altadis doing practices through extracurricular program and University curricular subjects, IS team Company represants and Computer Engineer profesor Juan Félix San Juan agreed in chosing Parts Production as a right theme for developing this Final Degree Project.

The Project aims to propose a new design and implementation of production application called Parts Production. This tool is used in the factory for the management and control of the production of cigarettes and incidents that may occur during the different shifts. Also it relates the measurements with corresponding operators working on each machine along the turn.

All this information is collected through this program in order to provide a number of statistics that help to restructure schedules optimizing the machines performance. So the treatment of this data is essential for achieving production company goals. The program generate reports that contribute to inform about the Imperial Tobacco factory productivity.

Tabla de contenido

RESUMEN	3
ABSTRACT	3
1. INTRODUCCIÓN	6
1.1. Antecedentes	6
1.2. Contexto	6
1.3. Motivación del Cambio	7
1.4. Tecnologías.....	7
1.5. Planificación	7
1.5.1. Alcance.....	7
1.5.2. Estructura de Descomposición del Trabajo (EDT).....	8
1.5.4. Descripción de las tareas del EDT	9
1.5.5. Distribución temporal	11
1.3.6. Plan de comunicación	13
1.3.7. Plan de calidad	13
CAPÍTULO 2. ANÁLISIS	14
2.1. Requisitos y roles de la aplicación.....	14
CAPÍTULO 3. ESTRUCTURA DE LA BASE DE DATOS	16
3.1. Base de datos Tet	16
3.1.1. Descripción de las tablas más importantes	16
3.2. Estudio de la interfaz gráfica de usuario y navegabilidad.....	20
3.2.1. Ventana principal.....	20
3.2.2. Ventana Ver Partes	25
3.2.3. Estudio y conclusiones de navegabilidad.....	26
CAPÍTULO 4. IMPLEMENTACIÓN	27
4.1. Estructura general de la aplicación	27
4.1.2. Clases Básicas.....	27
4.1.2. Capa de persistencia	27
4.1.3. Capa de lógica de negocio	28
4.1.4. Capa de presentación	31
4.2.1 Vista de la Ventana Principal.....	31
4.2.2 Autocompletado y ventanas de introducción de datos.....	32
4.2.3 Proceso de guardado de datos.....	35
4.2.3 Gráfico de Tiempo de Operación Efectivo	35
4.2.4 Vista general de la pantalla Ver Partes	36

4.2.5 Drag & Drop	37
PRUEBAS.....	40
CONCLUSIÓN.....	41
Tabla comparativa de horas previstas y horas reales.....	42
BIBLIOGRAFÍA	45
ANEXO	46
Actas de reuniones.....	46
Librería TetServer	50
Breve Guía de Usuario de Partes de Producción	56

1. INTRODUCCIÓN

La memoria de este trabajo pretende recoger los aspectos más importantes del mismo bajo el título de Diseño y Reimplementación de Partes de Producción. Este proyecto se ha desarrollado para el departamento de Informática de la empresa tabacalera Altadis.

1.1. Antecedentes

Para entender el funcionamiento de las tecnologías y programas que se comunican con las máquinas productoras de cigarrillos hace falta explicar y contextualizar el entorno de trabajo.

En la planta de secundario de la fábrica se distribuyen las diferentes líneas de producción. En estas líneas se proporciona el tabaco ya tratado para su liado, empaquetado, encajonado y desplazamiento hasta el almacén. Hay algunos puntos o cabinas situadas a lo largo de cada línea, en los que nos podemos encontrar un equipo y una pantalla auxiliar denominada Dlog. Las conexiones que se establecen en estos equipos con las máquinas permite recoger información y datos sobre la producción: velocidades, cantidades de producto y desperdicio, averías o paradas que se han producido, etc... Una forma de dejar constancia del estado de la producción durante cada turno es realizando un parte, que el operario en máquina se encarga de gestionar y dar de alta. La aplicación utilizada para este cometido es Partes de Producción. El Dlog antes mencionado es una pantalla que informa sobre los cajones producidos y muestra el avance respecto a la planificación de la labor (tipo de tabaco) que se está produciendo.

La necesidad de desarrollar una aplicación como Partes de Producción en la fábrica surgió con el objetivo de informatizar parte de los datos que se recogían de forma manual en las máquinas de producción. Con ello se pretendía obtener datos más fiables y un mayor control sobre los problemas que se producían durante la producción. De este modo podrían realizarse informes más útiles para la planificación y toma de decisiones. Dichos documentos pueden consultarse actualmente tanto a través de la aplicación como por la intranet de la empresa. Esta aplicación se utiliza principalmente para la producción de cigarrillos, aunque también hay opciones para que se puedan dar de alta partes en la sección de filtros y plantas de preparación de rama (primario).

1.2. Contexto

La fábrica de tabaco Altadis se mueve según las directrices que marca Imperial Tobacco. Desde hace algún tiempo es evidente la apuesta por el software de Microsoft y en concreto la plataforma .NET. Hace apenas unos meses que se ha llevado a cabo en la fábrica la instalación de una nueva interfaz de usuario para la visualización del contador de los cajones de los Dlogs. Esta actualización ha sido implementada en .NET y supone una mejora visual y funcional. Permite una mayor flexibilidad a la hora de obtener la planificación de las labores que se producirán en cada máquina. Además el funcionamiento mediante pantalla táctil es mucho más intuitivo y permite obtener con un solo vistazo el estado del proceso de producción, que es supervisado en cada turno por el correspondiente operario.

Con el desarrollo de la aplicación de Partes de Producción en la misma plataforma, se pretende llevar a cabo una serie de mejoras y conseguir adaptarse al diseño utilizado en esta aplicación complementaria que se acaba de describir.

1.3. Motivación del Cambio

A parte de la tendencia internacional antes ya comentada, comparando las tecnologías disponibles con las posibilidades que ofrecía la actual, se llegó a la conclusión de que había llegado el momento de migrar la aplicación a otra plataforma y lenguaje que ofrecieran mejores utilidades con las que poder implementar una mejor versión del programa.

La relación con productos de Microsoft estaba más que consolidada en la compañía, y reforzada internacionalmente. De modo que parecía natural la utilización de Visual Studio 2013 Ultimate como plataforma en la que se desarrollaría la aplicación de escritorio.

Una vez desarrollada la aplicación y después de un periodo de prueba en el que se confirmaría la robustez de la aplicación, no se descartaba el hecho de que en un momento dado pudiese exportarse parte de la funcionalidad a otras fábricas de la familia Imperial Tobacco.

1.4. Tecnologías

En este caso la reimplementación de Partes de Producción se realizará como una aplicación de escritorio. Deberán poder tener acceso a la misma tanto los operarios que trabajen en la línea de producción como el Departamento de Mejora Operacional y Planificación encargados de supervisar y ofrecer mejoras de rendimiento y productividad.

El lenguaje de programación elegido para el desarrollo de la nueva implementación del programa ha sido C# y el sistema de gestión de bases de datos utilizado es Oracle 9i.

1.5. Planificación

En esta sección se recoge una descripción del alcance del producto, la metodología de desarrollo que se ha elegido, la descomposición jerárquica de tareas y la planificación temporal de las mismas.

1.5.1. Alcance

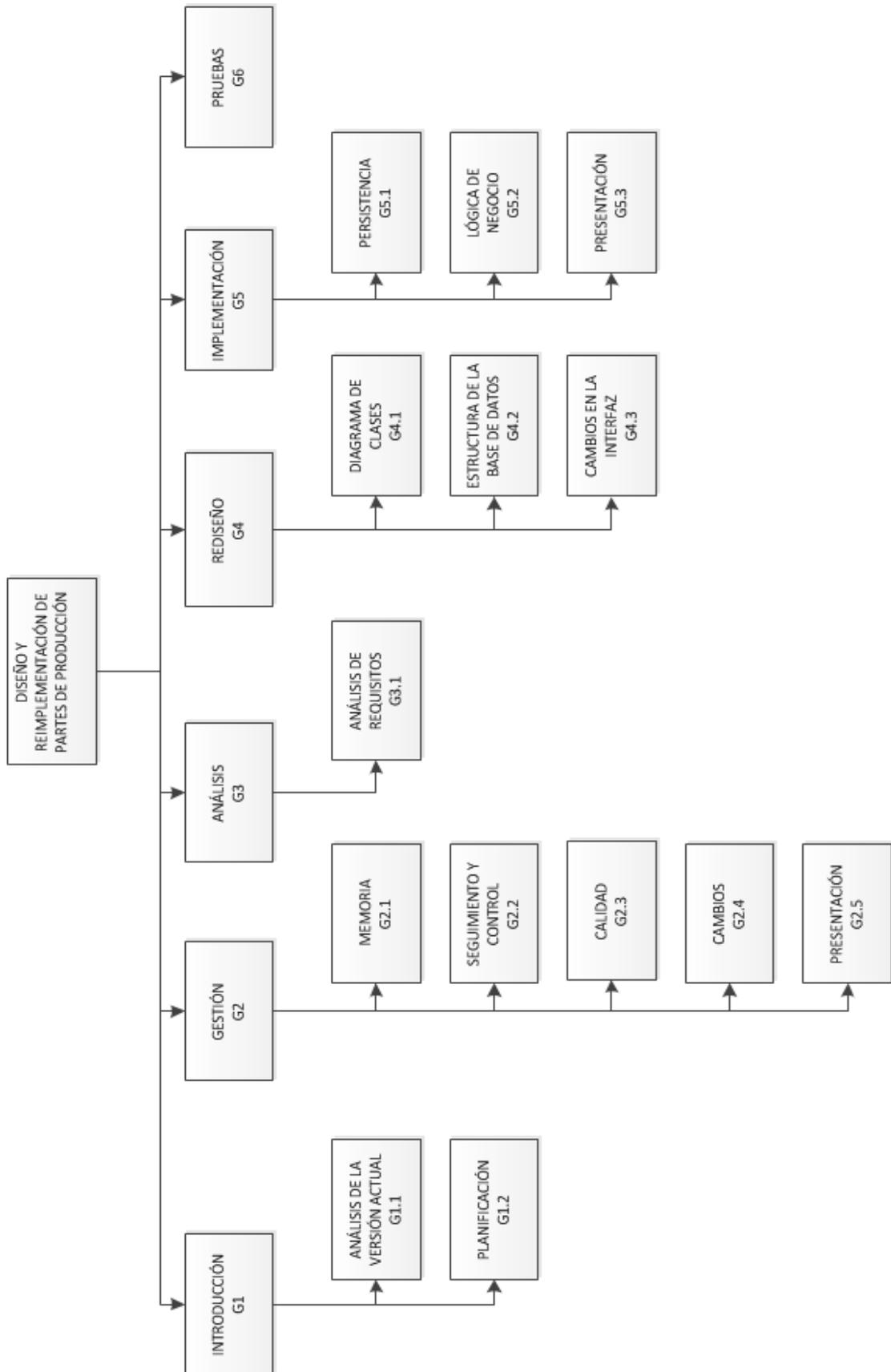
La primera parte del presente Trabajo de Fin de Grado se dedicará al análisis de la aplicación (la estructura de clases o procedimientos, interacción con la base de datos, diseño de la interfaz de usuario, procedimiento de acceso, estudio de la utilización del mismo por parte del usuario). Una vez extraídos los requisitos podremos determinar las carencias y cuál ha sido la parte más costosa económicamente. Se realizará un diseño que incluya dichos cambios y finalmente se implementarán en la plataforma Visual Studio.

La metodología de desarrollo elegida ha sido el desarrollo en cascada. Cada una de las fases del ciclo de vida del proyecto se llevará a cabo una única vez y en un orden lineal: análisis, diseño, implementación y pruebas.

Durante el ciclo de vida del proyecto se llevarán a cabo las siguientes tareas:

- Reuniones con el responsable de IS.
- Reuniones con el director de proyecto.
- Redacción de una memoria de aproximadamente 50 páginas, en la que se detallarán las tareas realizadas durante la duración del trabajo.
- Seguimiento y control del trabajo realizado, en comparativa con el planificado.
- Realización y preparación de una presentación para la defensa del Trabajo Fin de Grado.

1.5.2. Estructura de Descomposición del Trabajo (EDT)



1.5.4. Descripción de las tareas del EDT

Código	Tarea	Descripción	Horas previstas
G1.0	Introducción	Tareas de preparación antes de iniciar el proyecto.	30
G1.1	Análisis de la versión actual	Análisis de la aplicación de Partes de Producción y del entorno en el que se utiliza.	20
G1.2	Planificación	Este apartado engloba la realización y descripción del EDP, las actividades que se van a realizar y como se han programado a lo largo del tiempo (diagrama de Hitos y de Gantt). También todo lo referente a la gestión de calidad, cambios y riesgos.	10
G2.0	Gestión	Abarca las tareas de control y gestión del proyecto, que se llevarán a cabo durante todo el ciclo de vida del mismo.	60
G 2.1	Memoria	La memoria forma parte de los entregables del TFG. De todos ellos es el más esencial puesto que recoge todo el trabajo desarrollado.	40
G 2.2	Seguimiento y control	Tareas destinadas a controlar el desarrollo y evolución del proyecto, así como a detectar posibles desviaciones y proceder a su corrección. Incluye las reuniones con el director y el responsable del departamento de IS, ambos codirectores del proyecto.	10
G 2.3	Calidad	Para evaluar la calidad del proyecto se tendrá en cuenta la lista de requisitos que la nueva implementación del programa tiene que incluir.	6
G 2.4	Cambios	Los cambios que se produzcan durante el desarrollo del proyecto se considerarán en función del estado del mismo.	4
G 2.5	Presentación	Se realizará la defensa del producto final como cierre del proyecto desarrollado.	8 (no incluidas en el recuento)
G 3.0	Ingeniería Inversa	Se estudiarán las tecnologías empleadas para la gestión y funcionamiento de la anterior versión del programa para decidir que estructura se seguirá en la nueva implementación.	25

Diseño y Reimplementación de Partes de Producción

G 3.1	Análisis de requisitos	Descripción de las funcionalidades que cubre la actual aplicación y que influyen en la calidad de la misma.	25
G 4.0	Rediseño	Especificación de los cambios estructurales y visuales que se pretenden implementar en la nueva versión.	30
G 4.1	Diagrama de clases	Realización de un diagrama que muestra la estructura del sistema especificando sus clases.	5
Código	Tarea	Descripción	Horas previstas
G 5.2	Estructura de la Base de Datos	Presentación y explicación de las bases de datos que intervienen en la aplicación y las tablas más significativas que se emplean.	20
G 5.3	Cambios de la interfaz	En esta tarea se pretende proponer una serie de cambios a realizar en la interfaz de la aplicación después de detectar ciertas debilidades en la navegabilidad y presentación.	5
G 6.0	Implementación	Desarrollo de las tres capas: presentación, lógica de negocio y persistencia.	160
G 6.1	Persistencia	Desarrollo de la capa encargada de trabajar con las diferentes bases de datos de la aplicación.	70
G 6.2	Lógica de negocio	Desarrollo de la capa que se comunicará con la de presentación y la de persistencia.	10
G 6.3	Presentación	Desarrollo de la parte visual de la aplicación.	80
G 7.0	Pruebas	Acciones destinadas a comprobar el correcto funcionamiento de la aplicación para detectar y corregir posibles fallos.	10
TOTAL HORAS TRABAJO FIN DE GRADO			315

1.5.5. Distribución temporal

La duración del proyecto se establece en 315 horas de trabajo, que se distribuirán en 24 semanas entre el 2 de febrero y el 29 de julio de 2015. La dedicación semanal al proyecto será de 20 horas semanales, distribuidas entre 3 y 5 horas diarias de lunes a viernes, dependiendo de los asuntos ajenos al proyecto, como las prácticas u otras asignaturas que se deban tratar en la semana.

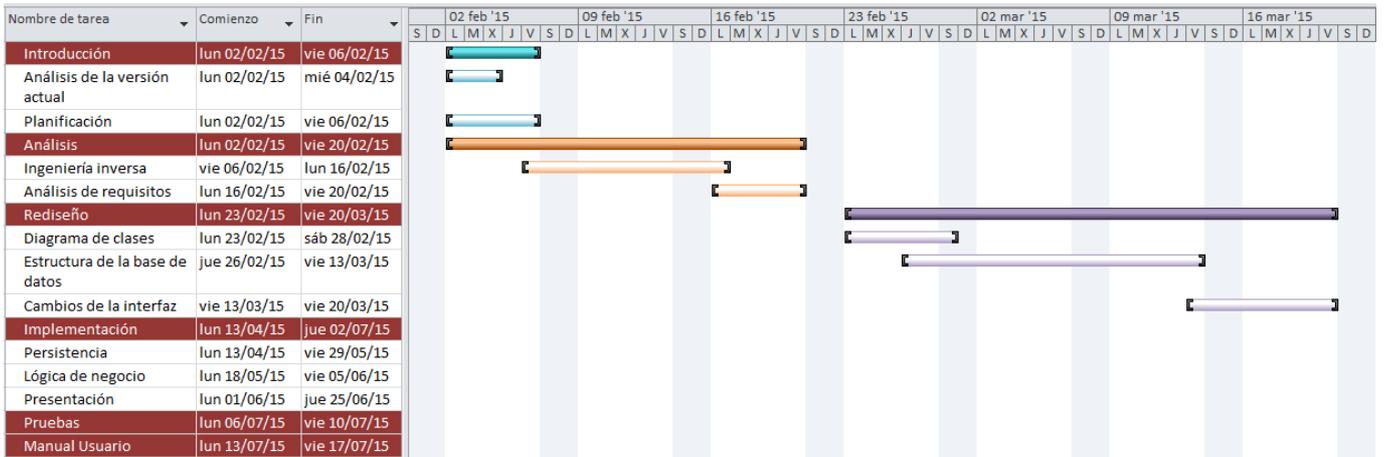
Las reuniones con el director de proyecto se establecen aproximadamente cada dos semanas. En estas reuniones se analizarán el estado y la evolución del TFG. Las reuniones con el responsable de IS se realizarán según las fases establecidas en la planificación del proyecto.

La fecha de fin de implementación de la aplicación queda establecida para el 2 de julio, dejando para la semana siguiente las pruebas y la puesta a punto de la memoria. Por lo tanto el trabajo quedará completamente terminado para el día 24 de julio, quedando pendientes la preparación de la defensa y la presentación final.

A continuación se muestra un Diagrama de Hitos con los hitos más importantes del proyecto:

	1	2	4	5	8	12	21	22	24	30
Hitos	2-5 feb	9-13 feb	23-27 feb	2-6 mar	16-20 mar	13-18 abr	2 jul	6-10 jul	27-29 jul	7-11 sep
Comienzo del TFG	•									
Entrega de la introducción, planificación y análisis	•									
1º reunión con el responsable de IS		•								
Comienzo del diseño			•							
2º reunión con el responsable de IS			•							
Entrega del diseño					•					
	S									
	1	2	4		8	12	21	22	24	30
3ª reunión con responsable de IS				•						
Comienzo de la implementación						•				
Entrega de la implementación							•			
4º reunión con el responsable de IS								•		
Comienzo de las pruebas								•		
Depósito del TFG									•	
Defensa										•

A continuación se muestra en Diagrama de Gantt con las tareas del proyecto:



1.3.6. Plan de comunicación

Durante el desarrollo del proyecto, se establecerá una comunicación con el director de proyecto, así como con el responsable del departamento de informática de Altadis. Dicha comunicación se realizará a través de los siguientes canales:

- Correo electrónico.
- Reuniones presenciales. En el caso del director del proyecto tendrán lugar en su despacho del Edificio Vives de la Universidad de La Rioja. Las fechas en las que se desarrollarán estas reuniones se acordarán previamente entre el director y el alumno. Por otro lado con el responsable de IS de Altadis dichas reuniones se realizarán en el mismo departamento o en otro tipo de salas de reuniones, convocándolas con anterioridad mediante Outlook.

1.3.7. Plan de calidad

Conforme se vayan dando por terminadas las tareas programadas se irá informando y mostrando los avances tanto al director del proyecto como al responsable del departamento de IS. Se les entregará la documentación oportuna con el fin de corroborar si el trabajo se está llevando a cabo conforme a lo establecido y, aclarar y realizar las correcciones oportunas en caso contrario.

El fin del trabajo consiste en establecer una serie de cambios para mejorar la experiencia del usuario con la aplicación, para ello se realizarán test de evaluación heurística a los diferentes tipos de usuarios.

CAPÍTULO 2. ANÁLISIS

Una vez establecido el contexto se recoge el análisis y documentación de los requisitos de la aplicación actual de Partes de Producción. Estos requisitos son esenciales puesto que marcarán el camino a seguir en la implementación en la nueva plataforma, así como los puntos a tener en cuenta a la hora de incluir mejoras.

2.1. Requisitos y roles de la aplicación

REQUISITOS FUNCIONALES

Usuarios de la aplicación:

- **Operarios:** son los trabajadores de planta encargados de las máquinas de los distintos grupos o líneas de producción.
- **Gestores:** son aquellos encargados de la supervisión del proceso de producción (jefes de línea).
- **Administradores:** personal del departamento de informática que se encarga de la gestión de control y mantenimiento de la aplicación.

Hay tres roles de usuario definidos:

- ❖ **Operario:** se le permite crear y eliminar partes siempre que no se hayan validado. En el caso de que ya se hayan validado sólo podrán visualizarlos.
- ❖ **Gestor:** se le permite introducir la producción notificada y validar/invalidar partes.
- ❖ **Administrador:** tiene control total sobre la aplicación.

Requisitos del sistema

La aplicación está destinada a facilitar la gestión de partes, de modo que debe ofrecer funcionalidad para:

- De cada parte se recogerá la siguiente información:
 - Los responsables operarios de la máquina o grupo sobre el que se realiza el parte.
 - La producción y el desperdicio en miles de cigarrillos.
 - Las incidencias que se han producido a lo largo del turno (indicando el código).
 - Gráfica en la que se muestra la proporción de paradas respecto al tiempo de producción a lo largo del turno (Tiempo de operación efectivo). Además en la misma pestaña se incluyen los porcentajes de desperdicio y parada.
 - Información sobre la cantidad de precintos utilizada (si es consumo se mostrará en rojo mientras que las entradas y la existencia inicial se muestra de color verde).
 - Un apartado para realizar observaciones.
 - Sección en la que se acredita el buen desempeño de las tareas realizadas (se especifican con un código abreviado, se las denomina checks)
- **Crear parte de producción.** Se guardará en la base de datos la información introducida en las diferentes secciones que conforman el parte. Es posible crear varios partes para

una misma fecha, turno y máquina si se trata de labores distintas (excepto que sea un producto terminado).

- **Modificación de parte de producción.** Se podrán introducir los cambios una vez se haya cargado la información del parte que queremos modificar. Después deberemos validar los cambios.
- **Eliminar parte de producción.** Tras cargar los datos y pulsar la opción que permite borrar los partes, saldrá un mensaje de confirmación. Después de confirmar la eliminación se borrarán los correspondientes registros de la base de datos.
- **Cancelar parte de producción.** Esta opción permite desechar los datos que hemos estado introduciendo hasta el momento en el parte.
- **Validación de parte de producción.** El parte debe crearse y cargarse previamente para poder validarlo.
- **Refrescar fichajes.** Opción que permita volver a cargar los fichajes de personal.
- **Mantenimiento (Opciones: Consignas, Labores, Maquinas, Formatos, Paradas, Permisos, Variables, Cabeceras parte).** Visualización en tablas de información sobre abreviaturas y características ampliadas de labores y máquinas. Sirve de consulta en caso de que se necesite algún tipo de aclaración a la hora de realizar el parte.
- **Visualizar partes de producción.** Hay diferentes vías para realizar esta acción. Una de ellas consiste en acceder a una ventana en la que se muestran los partes realizados por grupo, turno y fecha (Control Partes Realizados). Por defecto se mostrarán los del día actual. La otra opción de búsqueda es a través de la interfaz principal. Primero se eligen las opciones de grupo, turno y máquina, una vez especificado el filtro pulsando el botón ver partes se cargarán los datos en las diferentes pestañas de la ventana principal. También se debería poder consultar el parte en la intranet aunque esta funcionalidad no está disponible actualmente debido al desuso.
- **Acceso a ayuda en línea.** Enlace a una página de la intranet que contenga una guía o documentación acerca del funcionamiento del programa. Dicha página no se encuentra disponible en la actualidad.
- **Justificación de incidencia.** Por defecto se registran las paradas de las máquinas y se cargan de forma automática. Pero el operario debe encargarse de justificarlas. En caso de que todavía no se hayan justificado el sistema mostrará una alarma parpadeante.

CAPÍTULO 3. ESTRUCTURA DE LA BASE DE DATOS

Al tratarse de una reimplementación, primero habría que entender la estructura de las diferentes bases de datos que intervienen en la captura de datos de la aplicación.

En este caso se trata de tres bases de datos distintas. Sin embargo como se ha indicado antes la parte de la aplicación referente al personal no se va a abordar finalmente, lo que reduce la intervención a sólo dos bases de datos. Ambas gestionadas en Oracle 9i.

Podría parecer raro que se extraigan datos de dos fuentes diferentes, pero esto se debe a la distribución actual de los servidores. En el servidor eslmsr036 se recogen datos de **MEC**, todo lo que tiene que ver con la planificación de la producción y especificación de labores y muestras. Mientras que en el servidor eslmsr035 se guardan datos de **TET** respecto a producciones y partes de producción. Ambas bases de datos, son bases de datos no relacionales.

3.1. Base de datos Tet

3.1.1. Descripción de las tablas más importantes

PRODREP es la tabla principal de la que extrae los datos la aplicación de partes. Se especifica el número de parte, la fecha en la que ha sido creado, el turno, el grupo de producción, el proceso (puede tener los siguientes valores: cigarrillos, filtros, P.P.R) y un campo de comentarios en el que se suelen incluir los problemas que ha surgido durante la producción del grupo durante el turno.

PRODREP

NUM_REP (int)	DAT_REP (date)	SHIFT (string)	LINE (string)	PROC (string)	COMM (string)	DAT_VAL (date)	USER_VAL (string)
------------------	-------------------	-------------------	------------------	------------------	------------------	-------------------	----------------------

PRODUCTION en esta tabla encontramos los datos de producción (miles de cigarrillos por labor) a nivel de máquina y grupo (liadoras y encajonadoras) además de las fechas de inicio, fin y el tiempo de duración. También se incluye el desperdicio, pero estos campos de la tabla se han dejado de utilizar porque se han incluido en otra aparte (WASTE).

En la tabla del **WASTE** deberían registrarse los desperdicios generados durante la producción por turno y grupo. Además se añade la información sobre la hebra de la labor sobre la que se ha producido el desperdicio. Se guardan desperdicios parciales, de cubeta, cajón y contenedor a partir de los que después se calcula el total en miles de cigarrillos.

PRODUCTION

NUM_REP (int)	MACHINE (string)	DAT_INI (date)	DAT_FIN (date)	TIME (int)	BRAND (string)	PRODUCTION (int)	PROD_NOT (int)	LINE (int)
------------------	---------------------	-------------------	-------------------	---------------	-------------------	---------------------	-------------------	------------

WASTE

STRAIN (string)	BRAND (string)	DATE_REP (date)	SHIFT (string)	SHIFT0 (string)	WASTE_CONT (int)	WASTE_CUB (int)	WASTE_CAJ (int)	WASTE_TOTAL (int)
--------------------	-------------------	--------------------	-------------------	--------------------	---------------------	--------------------	--------------------	----------------------

DOCCUR en esta tabla se encuentran reflejadas las paradas que tienen lugar durante un turno en un determinado grupo. Se indica su duración y el código que pretende contextualizar la causa de la parada. También se guarda el orden de parada, si se han producido múltiples.

DOCCUR

NUM_REP (int)	MACHINE (string)	IDOCCUR (int)	IDSOURCE (string)	IDREASON (string)	STOP (string)	DAT_INI (date)	DAT_FIN (date)	TIME (int)	LINE (int)
------------------	---------------------	------------------	----------------------	----------------------	------------------	-------------------	-------------------	---------------	---------------

Si la parada se encuentra “sin justificar” los campos IDOCCUR, IDSOURCE, IDREASON y STOP se muestran con unos valores predeterminados (extracto de la base de datos):

IDOCCUR	IDSOURCE	IDREASON	STOP
999	*	*	*Justificar >>> Se Parada Manual

Árbol de Paradas

Hay una serie de tablas relacionadas con la justificación de las paradas, en la que se guardan los códigos y descripciones más habituales que pretenden aclarar el por qué se ha originado la parada. A estos códigos que se utilizan para definir las posibles causas de las paradas se le denomina “árbol de paradas”, este árbol consta de diferentes niveles:

IDOCCUR -> IDSOURCE -> IDREASON

(Ejemplo: 2-->SC02-->Falta de Consigna)

Tabla **OCCURRENCE**:

OCCURRENCE	
IDOCCUR (int)	DESOCCUR (string)

Raíz. En esta tabla se guardan los tipos de parada. Los posibles valores son:

- 0- Sin Reparación
- 1- Con Reparación
- 2- Sin Carga

Las paradas de tipo 2 (Sin Carga) son las que no hay que tener en cuenta de cara al cálculo de **OEE Overall Equipment Effectiveness (Tiempo de Operación Efectivo)**

Primer nivel de clasificación de paradas.

Todas las clasificaciones pertenecen a un tipo de parada (campo **IDOCCUR** que enlaza con tabla **OCCURRENCE**)

Tabla **OCCSOURCE**:

OCCSOURCE		
IDSOURCE (string)	DESSOURCE (string)	IDOCCUR (int)

Segundo nivel de clasificación de paradas.

Tabla **OCCREASON**:

OCCREASON

IDREASON (string)	DESREASON (string)
--------------------------	---------------------------

Tablas que sirven de nexos

Las tablas **OCCSOURCEMA** y **OCCSOURCERE** son para relacionar tablas entre sí.

Tabla **OCCSOURCEMA**:

No todas las máquinas tienen todos los conceptos del primer nivel de clasificación. En esta tabla se asocia el **IDSOURCE** con la máquina. Si se especifica en **MACHINE** un * es que es válido para todas las máquinas.

OCCSOURCEMA

IDSOURCE (string)	MACHINE (string)
--------------------------	-------------------------

Tabla **OCCSOURCERE**:

Esta tabla relaciona el primer nivel de clasificación (**IDSOURCE**) y el segundo (**IDREASON**).

OCCSOURCERE

IDSOURCE (string)	IDREASON (string)
--------------------------	--------------------------

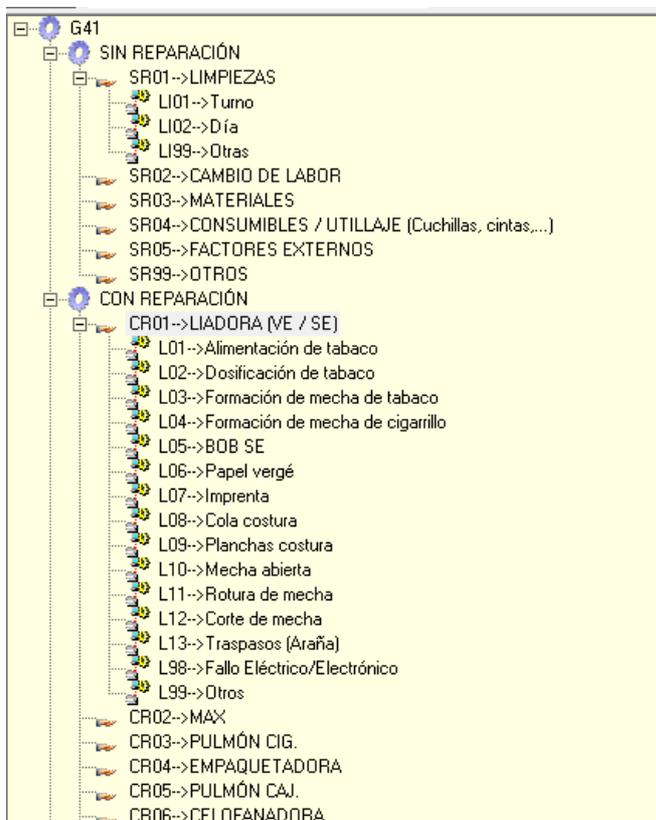


Figura 1 Ejemplo de disposición del árbol de paradas en Partes de Producción.

MOVPREC en esta tabla se almacena información sobre los movimientos de precintos durante la producción de labores (encajonadoras). Al igual que ocurre con las paradas se guarda el número de movimiento (line).

MOVPREC

NUM_REP (int)	MACHINE (string)	BRAND (string)	NUMPREC (int)	COD_MOV (string)	COMM (string)	LINE (int)
----------------------	------------------	----------------	---------------	------------------	---------------	------------

En **CHECKLIST** se almacenan una serie de comprobaciones que se realizan en las liadoras y encajonadoras durante la producción. En la tabla **CODDIS** se encuentra la explicación de las siglas del código de los checks que se realizan. Cuando el estado de un check es incorrecto, es obligatorio especificar en un comentario cuáles han sido las razones por las que la comprobación no es válida.

CHECKLIST

NUM_REP (int)	CODE (string)	STATUS (string)	COMM (string)
----------------------	---------------	-----------------	---------------

CODDIS

COD_FAM (string)	NUM_COD (string)	DES (string)
-------------------------	------------------	--------------

Selección de máquina asociada al grupo

Dentro de los grupos de producción puede haber máquinas distintas aunque realicen la misma función. Es por esto que dichas máquinas se denominan “activas” y se debe guardar su relación con el grupo. Las características que las diferencian unas de otras pueden ser decisivas a la hora de asociar las causas cuando se produce una parada.

Hay una tabla en la que se especifica esta relación:

MACHINEPP

LOMACHINE (string)	IDMACHINE (int)	SPEED (int)	POSITION (int)	ACTIVE (bool)	REFER (bool)
---------------------------	-----------------	-------------	----------------	---------------	--------------

En la tabla además de la relación con el grupo se guardan otras características de las máquinas como la velocidad o capacidad de hacer cigarrillos.

Tabla para gestión de usuarios

El acceso de los usuarios a la aplicación se gestiona a través de una tabla llamada **USERSROLES**. De este modo pueden restringirse zonas dentro del programa según el rol o perfil.

USERSROLES

ROLE (String)	USERCODE (String)	CREATION (Date)	USERNAME (String)
----------------------	-------------------	-----------------	-------------------

Tablas que se utilizan de MEC

Los turnos se establecen durante todo el año, contemplando las rotaciones y los correspondientes horarios. Dicha información se planifica y se almacena en una BD aparte en la tabla **SHIFTPLAN**.

Se almacena el día y hora en que comienza y acaba el turno además del tipo de turno (A,B,C) y su orden según el horario que se le ha asignado. Esta tabla es necesaria para establecer el turno actual en la aplicación.

SHIFTPLAN

DSTART (date)	DEND (date)	SHIFTNUMBER (int)	SHIFTID (string)
----------------------	-------------	-------------------	------------------

3.2. Estudio de la interfaz gráfica de usuario y navegabilidad

3.2.1. Ventana principal

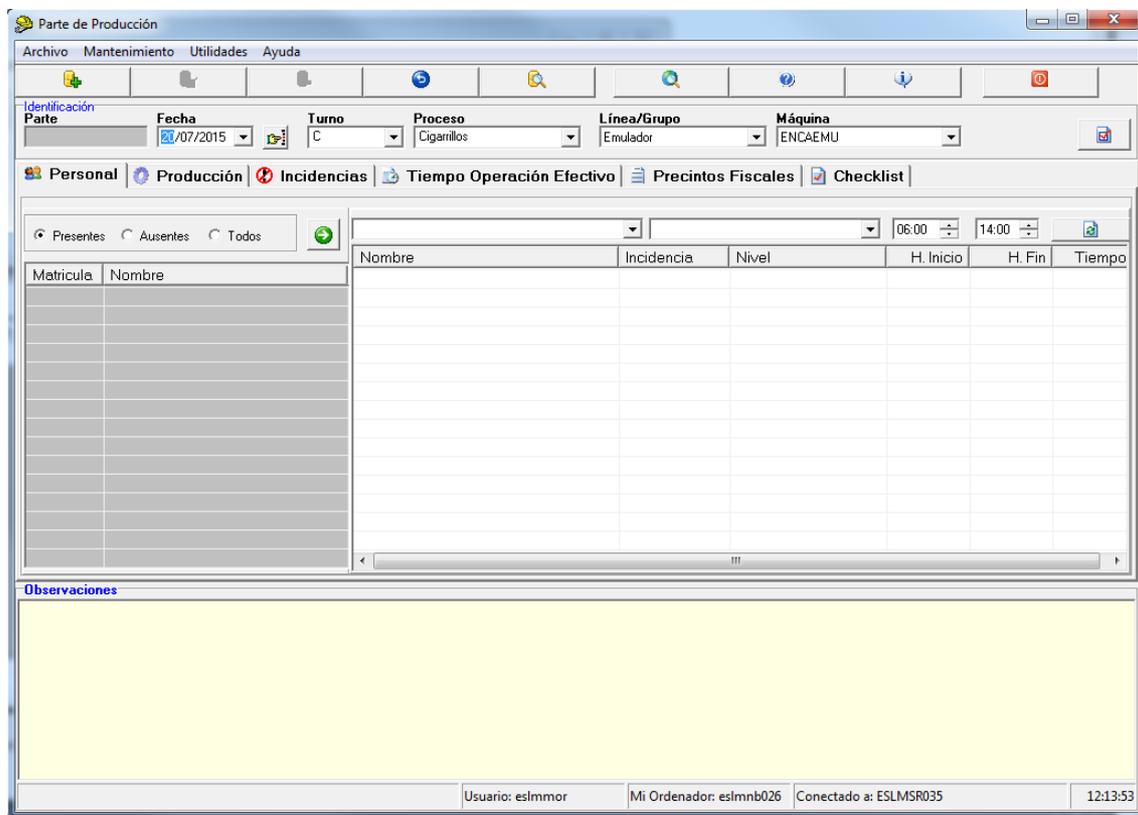


Figura 2 Vista de la ventana principal de la actual aplicación de Partes de Producción.

El diseño de la ventana principal viene condicionado por las distintas secciones esenciales que estructuran un parte de producción. Estas secciones se reparten en diferentes pestañas (Personal, Producción, Incidencias, Tiempo Operación Efectivo, Precintos Fiscales y Checklist).

En la barra de estado aparece el nombre del usuario de acceso a la aplicación y el terminal desde donde se conecta.

En la ventana principal se encuentran:

- Representación de las funcionalidades básicas (Crear Parte, Modificar Parte, Eliminar Parte, Cancelar cambios, Buscar Parte, Consultar e Imprimir Partes, Ayuda, Acerca de, Terminar) mediante botones justo debajo de la barra principal de tareas.
- Filtro de búsqueda por Fecha, Turno, Proceso, Línea o Grupo y Máquina. En la sección del filtro se incluye el botón para la validación del parte.

Sección Producción

La introducción de datos se agiliza mediante una ventana en la que nos permite introducir todos los valores referentes a la producción

Figura 3 Ventana para la introducción de producciones.

Identificación									
Parte	Fecha	Turno	Proceso	Línea/Grupo	Máquina				
	20/07/2015	C	Cigarrillos	Emulador	ENCAEMU				
Personal Producción Incidencias Tiempo Operación Efectivo Precintos Fiscales Checklist									
Seleccione		06:00 - 14:00							
Máquina	H.Inicio	H.Fin	Tiempo	Labor	Prod. Li...	Desp. Li...	Prod. E...	Desp. E...	P. Notific.
<input type="checkbox"/> Emulador	06:00	14:00	08:00	→	14,157,474	45,454	554,545	2,365	

Figura 4 Estado de la interfaz después de introducir los datos.

Sección Incidencias

En este apartado se debe especificar las paradas, la causa que se elige del “árbol” ya descrito anteriormente y la duración de la misma.

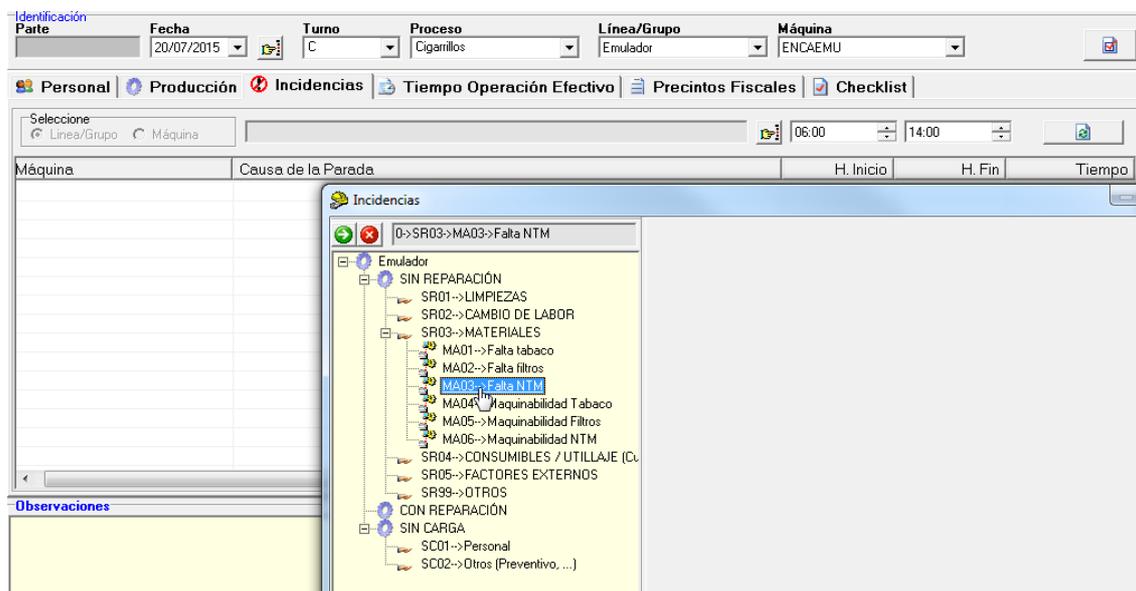


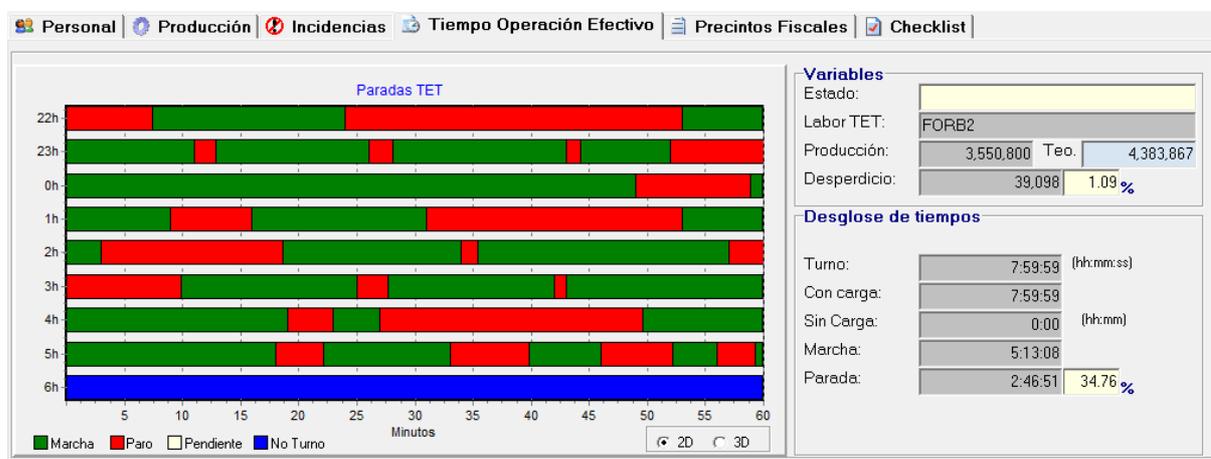
Figura 5 Ventana principal y elección de causa de paradas en la ventana Incidencias.

Conforme va pasando el tiempo del turno, en el parte se irán cargando las paradas que se produzcan en el grupo y máquina seleccionada. Aunque esto aparezca de forma automática, el usuario también tiene la posibilidad de introducir una parada no justificada. Esta situación no debería producirse si se crea el parte cuando finaliza el turno correspondiente.

Máquina	Causa de la Parada	H. Inicio	H. Fin	Tiempo
<input type="checkbox"/> G55	*Justificar >>> Max Parada Manual	06:13	12:57	6:44
<input type="checkbox"/> G55	*Justificar >>> Cargador Bobinas Se	13:06	13:44	0:38

Figura 6 Paradas sin justificar en la sección de Incidencias de la ventana principal de Partes.

Sección Tiempo de Operación Efectivo



Esta parte resulta la más visual de la aplicación puesto que se muestra un gráfico que abarca todas las horas que constituyen el turno y las paradas que se producen a lo largo del mismo.

Además se facilitan valores de algunas variables, que posteriormente se utilizan en cálculo de índices de producción.

Figura 7 Gráfico de Paradas de la ventana principal de Partes de Producción.

Sección Precintos Fiscales

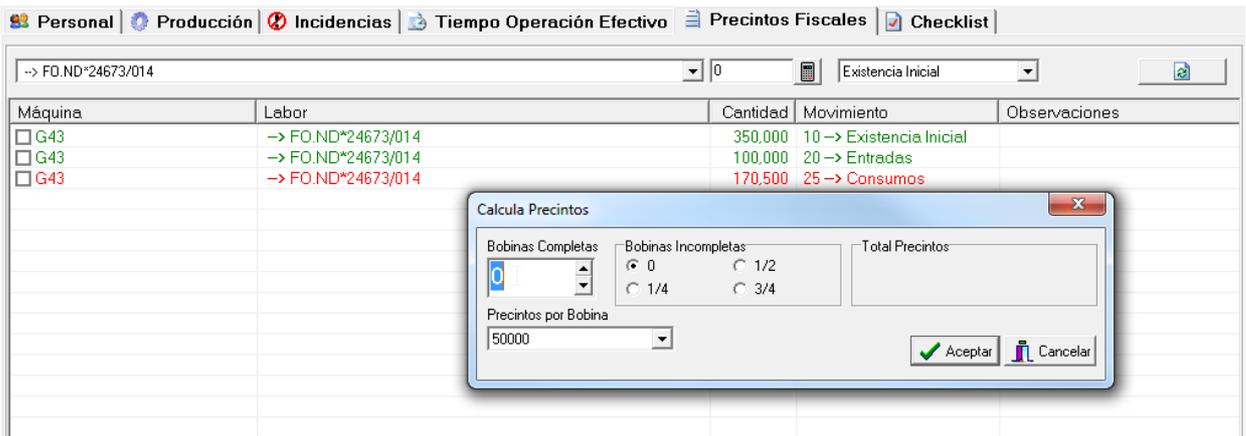


Figura 8 Ventana para la introducción de la cantidad de precintos a partir de bobinas.

En esta sección debe quedar presente los movimientos de precintos que tienen lugar cuando se está produciendo una labor. Se puede calcular e introducir el número de precintos con facilidad desde una ventana auxiliar.

Sección Checklist

Para cada parte se guardan una serie de checks con un estado de cumplimiento. En caso de que no se cumpla alguno de los establecidos es obligatorio indicar la causa en el apartado de Observaciones.

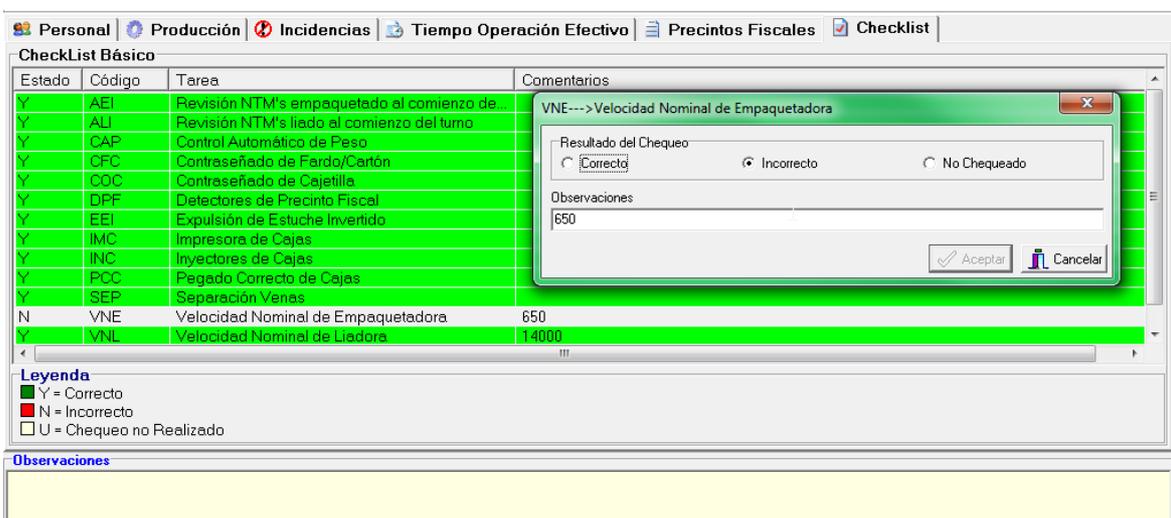


Figura 9 Ventana para el cambio de estado de un check y vista general de la pestaña Checklist.

Sección para Usuarios

Dentro del programa de partes podemos encontrar una opción en el menú que sirve para introducir los usuarios que tienen permisos para utilizar el programa. De este modo se controla el acceso de forma restrictiva. Sólo tienen acceso a esta sección aquellos que tienen un rol definido como Administrador en la base de datos.

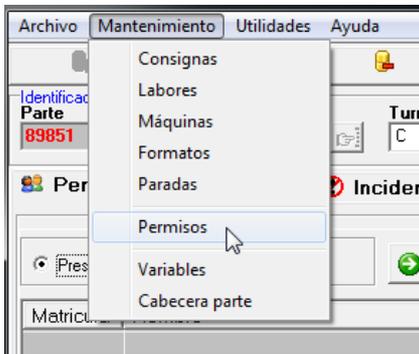


Figura 10 Vista detalle del Menú Mantenimiento.

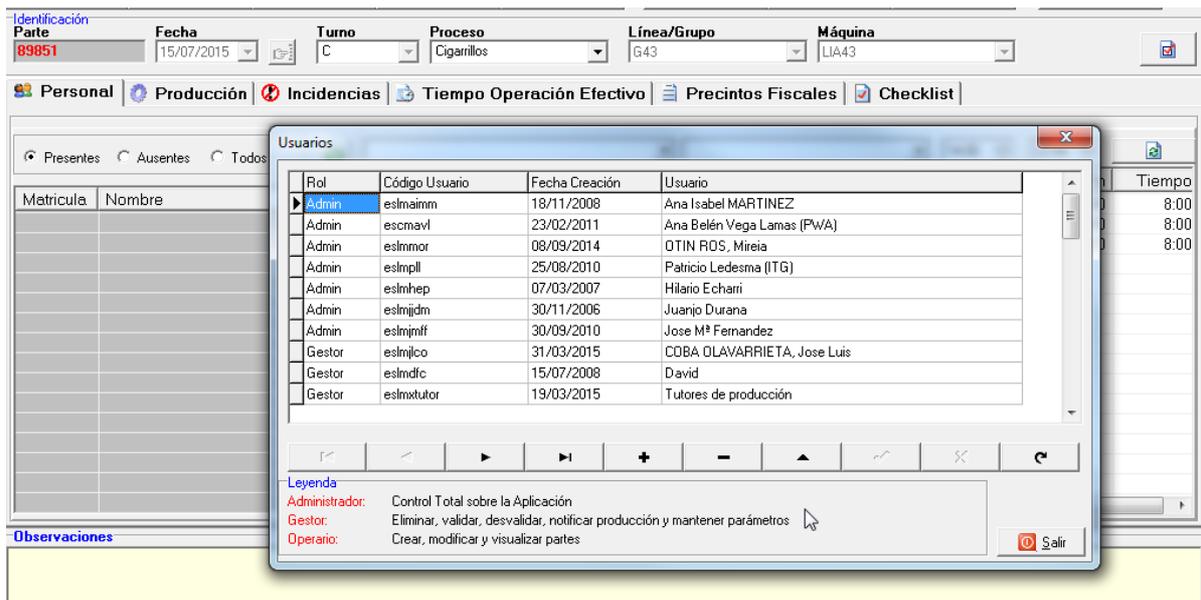


Figura 11 Vista de la Ventana de Permisos donde el administrador gestiona los usuarios.

3.2.2. Ventana Ver Partes

Aquí pueden verse todos los partes que se han guardado en la base de datos sobre una fecha determinada. Aparecen todas las máquinas, aunque algunas no tengan partes disponibles para consultar. La vista del parte es parecida a la que puede verse en la ventana principal. Esta ventana es sólo de consulta, no está destinada para realizar ni guardar modificaciones.

Fecha: 01/07/2015

Personal

Nombre	Incidencia	Nivel	H. Inicio	H. Fin	Tiempo
00019789 GARCIA DOMINGUEZ, PE...			14:00	22:00	8:00
00019802 MAYORA GARCIA DE GAL...			14:00	22:00	8:00
00019886 CARRO FRIAS, ALBERTO			14:00	22:00	8:00

Producción

Máquina	H.Inicio	H.Fin	Tiempo	Labor	Prod. Lia...	Desp. Li...	Prod. E...	Desp. E...	P. Notific...
G30	14:00	22:00	8:00	-> 51141/005D	3,100,000	15,500	155,000	420	0.0000

Incidencias

Máquina	Causa de la Parada	H. Inicio	H. Fin	Tiempo
G30	0->SR01->LI01->Turno	14:00	14:30	0:30
G30	1->CR02->M05->Alimentación de papel hidrofugado	16:45	17:00	0:15
G30	0->SR04->CU01->Cuchillas, cintas, bobinas...	21:15	21:25	0:10

Precintos

Máquina	Labor	Cantidad	Movimiento	Observaciones
G30	-> 51141/005D	345,000	10 -> Existencia In...	

Observaciones

Figura 12 Vista general de la ventana Ver Partes.

- G30
 - 89659->A
 - 89667->B
 - 89647->C
 - G31
 - G33
 - G35
 - G41
 - 89654->A
 - 89665->B
 - 89651->C
 - G43
 - 89658->A
 - 89661->B
 - 89652->C

Figura 13 Vista detalle del despliegue de partes de la ventana Ver Partes.

3.2.3. Estudio y conclusiones de navegabilidad

Con el fin de reafirmar algunas de las debilidades que ya se habían detectado en la versión anterior de Partes de Producción se procedió a realizar un test heurístico a dos tipos de usuarios con respecto a la siguiente serie de parámetros.

Generales:

- Utilidad de todos los botones disponibles en la interfaz de la ventana principal del programa.

Específicos de los usuarios que crean partes en la zona de producción:

- Facilidad de tiempo de introducción de información en el parte.
- Problemas al dar de alta el parte.

Específicos de los usuarios que crean partes en la zona de producción:

- Transición entre la ventana principal y la de Ver Partes.
- Datos de importancia que se consultan en la ventana Ver Partes.

Uno de los perfiles de usuarios es aquel que se suele encargar de dar de alta los partes durante el turno.

El otro perfil que se tuvo en cuenta para la prueba era la de aquellos usuarios que posteriormente realizaban un estudio y extraían comportamientos y actuaciones a partir del uso de la aplicación de partes.

Conclusiones importantes que se obtuvieron del estudio:

- Los usuarios tienen que cerrar la ventana de Ver partes y volver a la principal si necesitan hacer modificaciones del mismo.
- Los usuarios tienen que introducir todos los datos, el programa no ayuda de ninguna manera.
- Si el usuario quiere comparar las producciones de un grupo tiene que ir abriendo los diferentes partes e irse fijando en los datos o apuntarlos en algún sitio.

Se han introducido una serie de cambios y mejoras que se explican posteriormente, sabiendo de antemano que iban a redundar en beneficios ya que así lo mostraban los resultados obtenidos en este test heurístico.

CAPÍTULO 4. IMPLEMENTACIÓN

Este capítulo recoge los aspectos más importantes de la implementación de la aplicación.

4.1. Estructura general de la aplicación

Para la implementación de la nueva versión de Partes se ha decidido seguir una estructura de tres capas: Presentación, Lógica y Persistencia.

4.1.2. Clases Básicas

Tras realizar un análisis de la estructura de las tablas de bases de datos se ha establecido una serie de Clases Básicas esenciales para el tratamiento de datos.

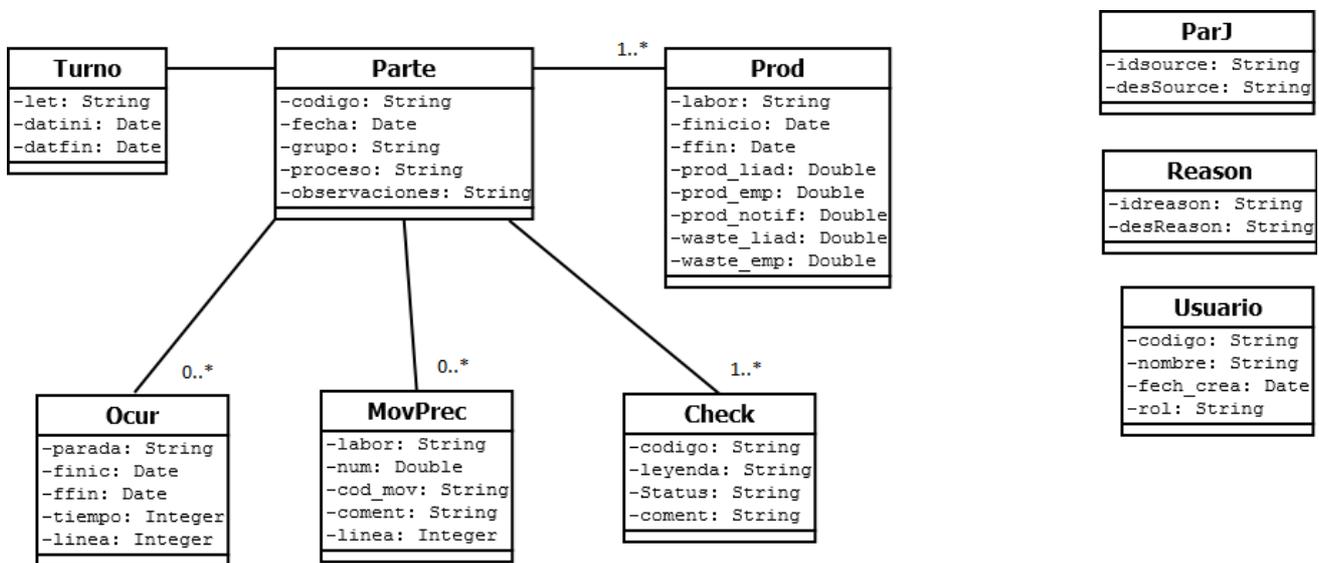


Figura 14 Diagrama UML de las clases básicas

No se han incluido en el diagrama las propiedades mediante las que se tiene acceso de lectura y modificación de los atributos.

4.1.2. Capa de persistencia

En la capa de persistencia se han programado los métodos correspondientes para la obtención y modificación de datos.

Se ha utilizado el espacio de nombres **System.Data.OracleClient** y las siguientes cadenas para establecer conexión con las bases de datos:

Conexión con **Tet**:

```
private static String conexion = "Server=(DESCRIPTION =(ADDRESS = (PROTOCOL = TCP)(HOST = ESLMSR035)(PORT = 1521))(CONNECT_DATA = (SID = XE)))";
```

Conexión con **MEC**:

```
private static String conexMec = "Server=(DESCRIPTION =(ADDRESS = (PROTOCOL = TCP)(HOST = ESLMSR036)(PORT = 1521))(CONNECT_DATA = (SID = XE)))";
```

Además se han establecido transacciones para la creación y actualización de tablas:

```
OracleTransaction trans;
using (trans = conn.BeginTransaction(IsolationLevel.Serializable))

catch (OracleException ex)
{
    com.Transaction.Rollback();
    throw;
}
finally
{
    conn.Close();
    conn.Dispose();
}
```

De este modo cualquier error que pudiera ocurrir en las consultas de actualización de información nos asegura la consistencia de los datos con los que se está trabajando. No se hace commit hasta que todas las operaciones o actualizaciones se han realizado con éxito.

4.1.3. Capa de lógica de negocio

En este caso la capa de Lógica de negocio establece un mero canal entre la persistencia y la capa de presentación. Los métodos realizan llamadas a la capa de persistencia con los parámetros adecuados. En esta capa se han incluido además los métodos que se conectan con la librería **TetServer** para extraer información sobre paradas, y datos de producción.

Al tratarse de una aplicación para la extracción, actualización y visualización de datos no se ha incluido en esta capa otro tipo de funcionalidad que no sea la de traspaso de información.

Es necesario que la ventana Ver Partes tenga acceso a la lógica de negocio ya que debe mostrar los partes disponibles en tiempo real. Muchos de los usuarios una vez creado el parte, acceden a esta ventana para certificarse únicamente de que el parte ha sido creado. También podrían utilizar la funcionalidad de buscar parte de la ventana principal, pero los test han puesto de manifiesto que la mayoría opta por la Ventana de Ver Partes porque resulta más rápido.

Cómo Ver Partes está pensado sólo para la visualización de los partes, no crea ningún tipo de inconsistencia el hecho de pasar un objeto de la lógica de negocio en el constructor del formulario.

Librería TetServer

TetServer.dll proporciona una serie de métodos que se conectan con el servicio encargado de recoger los datos de las diferentes máquinas, de modo que podemos extraer tiempos de paradas que se están produciendo a tiempo real. También nos permite acceder a los históricos con datos que pertenecen a otros turnos y grupos.

Para poder hacer uso de la dll se debe registrar ejecutando el comando regsvr32.exe de la consola de comandos de Windows.

Los métodos utilizados establecen una conexión **TetConnection** y a través de una serie de parámetros como máquina, turno y variable se incluyen en una query. Los datos obtenidos se recogen en un array de cadenas de texto que contiene la información que hemos requerido especificando la variable adecuada. Las variables son diferentes y pueden hacer referencia al contador de cajones, a las paradas u otro tipo de parámetros.

Este tipo de información nos interesa en el caso de que estemos creando un parte en un grupo y máquina, ya que se podrá visualizar en tiempo real las paradas que hasta el momento han tenido lugar.

El acceso a históricos es necesario en el caso de que deseemos cargar algún parte de días anteriores, o de otros turnos. La variable utilizada es la misma, y el proceso es similar sólo que se debe especificar una fecha.

Todos estos datos obtenidos sobre las paradas han servido como información que después se ha tratado para presentar un gráfico de paradas desglosado en las horas que conforman un turno. Se ampliará la información de carácter general sobre esta librería en el anexo de la memoria.

Estableciendo Conexión

```
TetConnection tc = new TetConnection();
tc.ConnectionString = "ESLMSR035";
..
```

Definición de Query y Variables

```
TetQuery tq = new TetQuery();
tq.GroupParam = "";
tq.MachineParam = maquina;
tq.Attr2Param = turno;
tq.RangeParam = "";
tq.TReport = "SHFT";
tq.AddLVar("TEMP008");
tc.ExecuteTetQuery(tq);
String texto = tq.GetVariable("TEMP008");
..
```

Almacenamiento de la información

```
String texto = tq.GetVariable("TEMP008");
TetArray ar = new TetArray();
ar.Text = texto;
```

Método completo

A través de este método se obtienen las paradas en tiempo real.

```

public List<Ocur> paradas_tiempoReal(String maquina, String turno)
{
    List<String> maqs = getMaquinas();
    List<Ocur> lisoc = new List<Ocur>();
    List<String> lisin = new List<String>();
    if (turno != null && maqs.Contains(maquina))
    {
        TetConnection tc = new TetConnection();
        tc.ConnectString = "ESLMSR035";
        TetQuery tq = new TetQuery();
        tq.GroupParam = "";
        tq.MachineParam = maquina;
        tq.Attr2Param = turno;
        tq.RangeParam = "";
        tq.TReport = "SHFT";
        tq.AddLVar("TEMP008");
        tc.ExecuteTetQuery(tq);
        String texto = tq.GetVariable("TEMP008");
        TetArray ar = new TetArray();
        ar.Text = texto;
        int i;

        for (i = 0; i < ar.RowCount; i++)
        {
            int k;
            String incidencia = "";
            for (k = 0; k < ar.ColCount; k++)
            {
                incidencia = incidencia + ar.GetString(i, k) + "&";
            }
            lisin.Add(incidencia);
        }

        lisoc = new List<Ocur>();
        foreach (String s in lisin)
        {
            String[] partes = s.Split('&');

            String fechin = partes[0];
            String[] horain = fechin.Split(':');
            String year = horain[0].Substring(0, 4);
            String moi = horain[0].Substring(4, 2);
            String numd = horain[0].Substring(6, 2);

            String hor = horain[1].Substring(0, 2);
            String min = horain[1].Substring(2, 2);
            String seg = horain[1].Substring(4, 2);

            DateTime dti = new DateTime(Convert.ToInt32(year),
                Convert.ToInt32(moi), Convert.ToInt32(numd), Convert.ToInt32(hor),
                Convert.ToInt32(min), Convert.ToInt32(seg));

            String fechfin = partes[1];
            String[] horafin = fechfin.Split(':');
            hor = horafin[1].Substring(0, 2);
            min = horafin[1].Substring(2, 2);
            seg = horafin[1].Substring(4, 2);
        }
    }
}

```

```
DateTime dtf = new DateTime(Convert.ToInt32(year),  
Convert.ToInt32(moi), Convert.ToInt32(numd), Convert.ToInt32(hor),  
Convert.ToInt32(min), Convert.ToInt32(seg));  
  
String dur = partes[2];  
String par = partes[3];  
  
Ocur c = new Ocur(dti, dtf, Convert.ToInt32(dur), par);  
lisoc.Add(c);  
  
}  
}  
  
return (lisoc);  
}
```

4.1.4. Capa de presentación

Esta sección recoge una descripción de los mecanismos que se han seguido para implementar las funcionalidades, que tras el test heurístico, se han contemplado como las adecuadas para ofrecer al usuario una mejor interacción con el programa. Se ha pretendido no cambiar mucho la estructura general con respecto a la anterior interfaz, para que el usuario no se hallara demasiado desubicado, ya acostumbrado al funcionamiento y apariencia de la otra.

4.2.1 Vista de la Ventana Principal

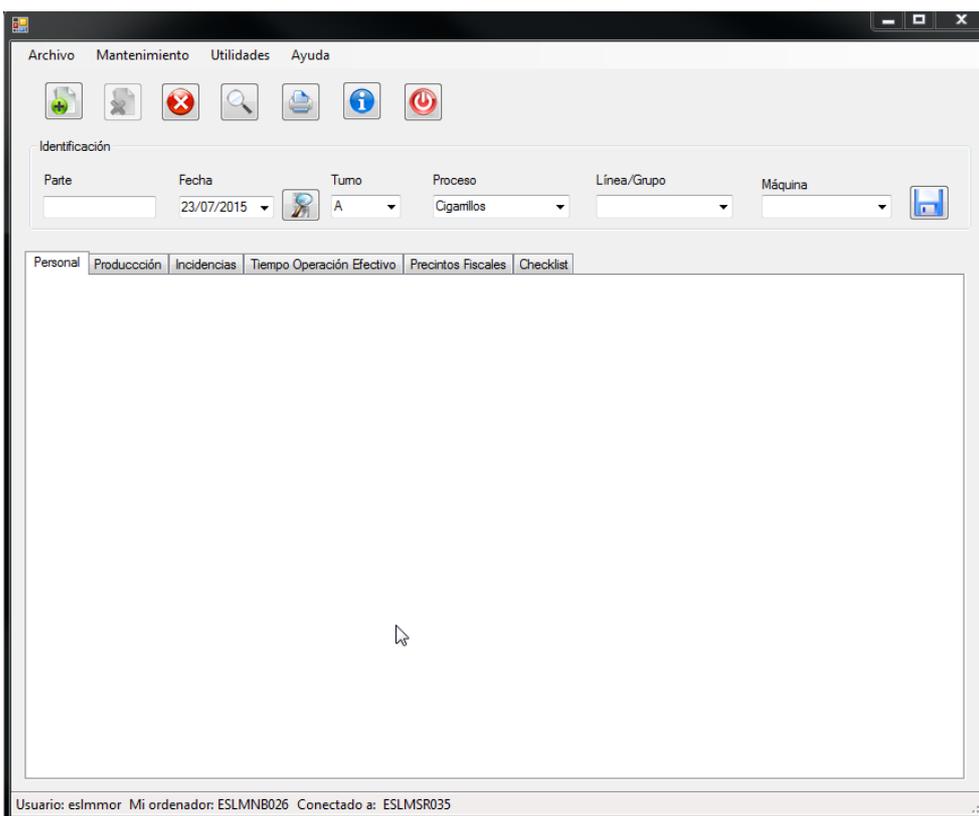


Figura 14 Ventana principal de la nueva interfaz de Partes de Producción

4.2.2 Autocompletado y ventanas de introducción de datos

Utilizando el evento *EditingControlShowing* se ha implementado un autocompletado de las distintas partes que componen el parte para facilitar la introducción de la información.

Producción

En los campos máquina y labor se despliegan los grupos de producción y las labores registradas en la base de datos. El usuario sólo tiene que poner los primeros caracteres para poder elegir la opción que desea sin tener que acabar de escribirlos.

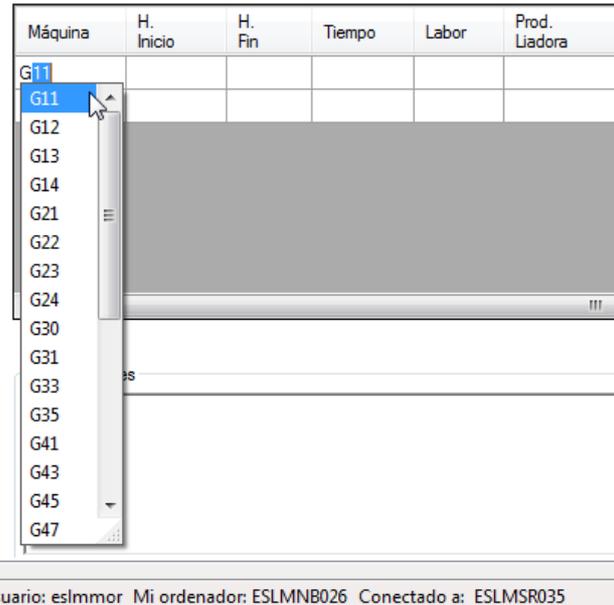


Figura 15 Detalle de la opción de autocompletado de los grupos de máquinas.

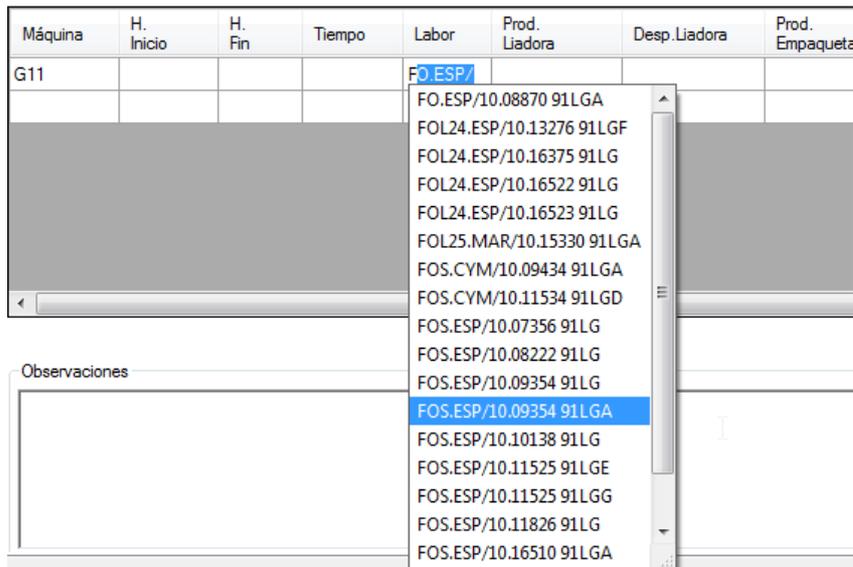


Figura 16 Detalle de la opción de autocompletado de las labores de producción.

Precintos Fiscales

Los campos Máquina y Labor funcionan de forma análoga a lo descrito anteriormente en producción.

Al hacer clic en la celda Cantidad aparece una ventana que facilita el cálculo de cantidad de precintos con respecto al número de bobinas utilizado o a utilizar.

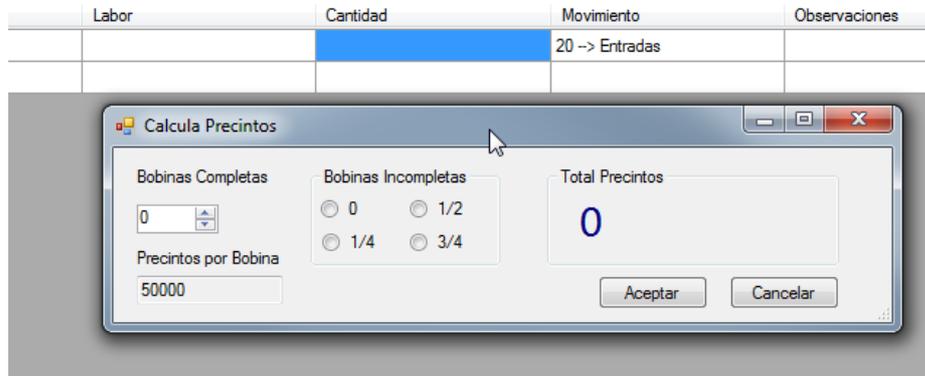


Figura 17 Detalle de la ventana de introducción del número de precintos.

En el campo Movimiento se muestran como opciones los posibles valores que puede tomar. Además después de seleccionar una de las opciones se autocompleta con toda la información necesaria que se debe incluir en la base de datos.

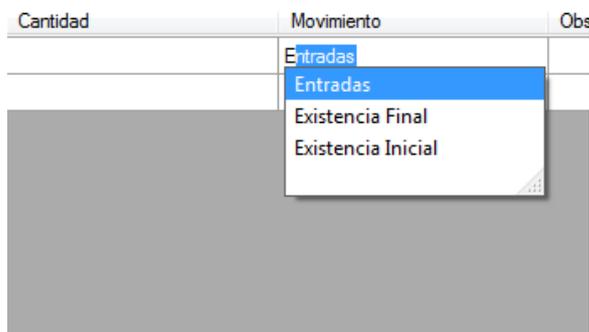


Figura 18 Detalle de la opción de autocompletado de los movimientos de precintos.

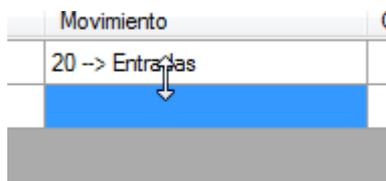


Figura 19 Detalle del autocompletado del código de movimiento del precinto.

Justificación de Paradas

Al cargar un parte las paradas sin justificar se muestran en rojo. Si se hace clic en la causa de la parada se abre una ventana que nos permite seleccionar la causa a partir de un “árbol” que se carga particularmente en función de cada máquina. Esto se debe a que hay distintos tipos de liadoras y empaquetadoras. Algunas por ejemplo, poseen pulmón de cigarrillos.

Al pulsar directamente en la causa se carga la traza de códigos que se utiliza para identificar apropiadamente el error.

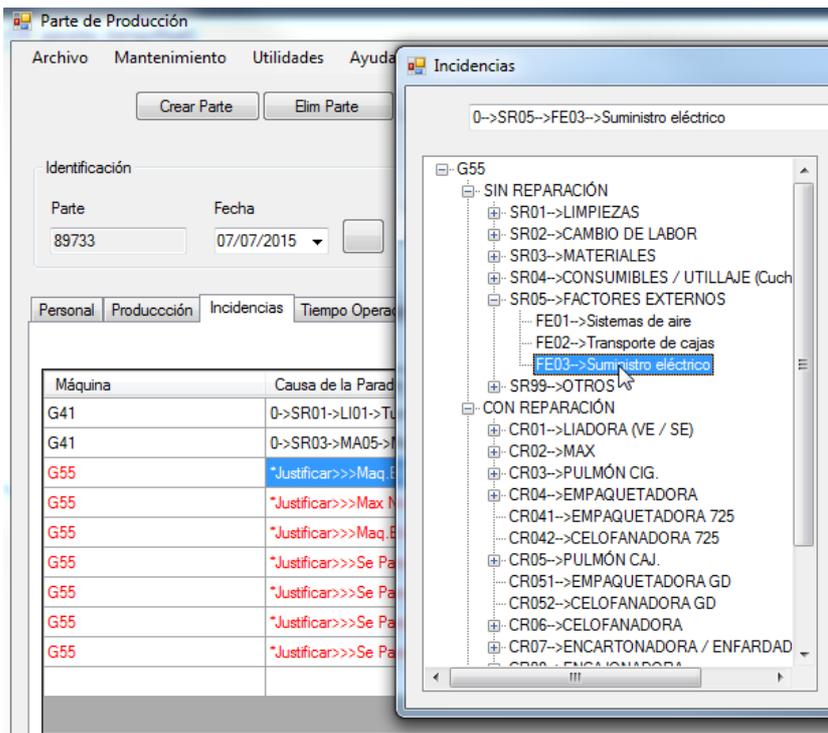


Figura 20 Vista de la sección de incidencias y la ventana de la justificación de paradas.

Máquina	Causa de la Parada	H.Inicio	H.Fin
G55	0->SR99->OT01->Otros	14:00	15:20
G55	1->CR05->PU01->Pulmón caj...	15:38	16:30
G55	0->SR02->CL01->Cambio de l...	16:55	17:05
G55	1->CR051->EMPAQUETAD...	17:08	20:49
G55	*Justificar >>> Se Parada Man...	20:54	21:45

Figura 21 Detalle de la causa de la parada una vez seleccionada de la ventana de justificación.

4.2.3 Proceso de guardado de datos

Se han utilizado diferentes eventos para detectar cambios en el estado de los datos de forma que una vez el usuario decida salirse o cambiar de ventana se le advierta de la posibilidad de guardar los cambios realizados. Además se ha habilitado un botón para ello.

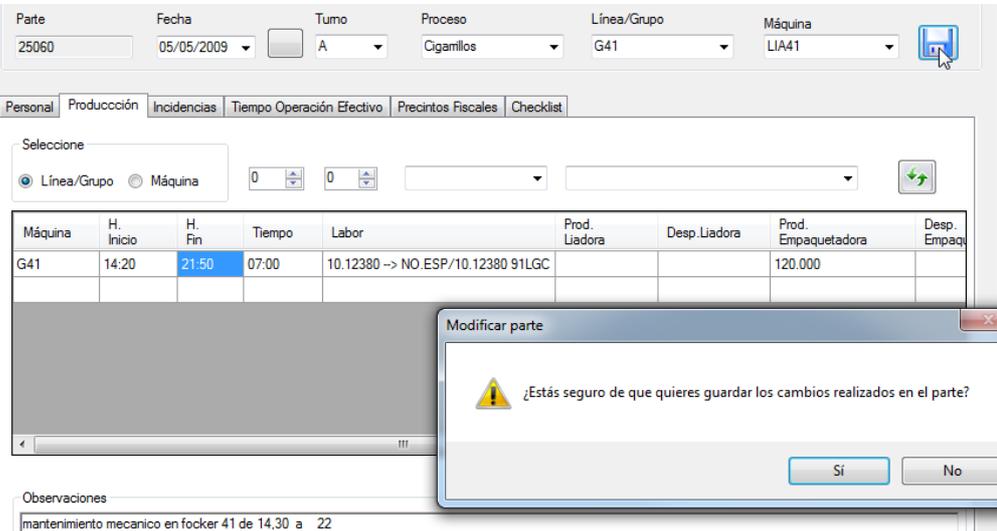


Figura 22 Vista del aviso para guardar las modificaciones realizadas en el parte.

4.2.3 Gráfico de Tiempo de Operación Efectivo

Para la elaboración del gráfico, se han extraído los datos de la librería **TetServer** de la que ya hemos hablado anteriormente. Para construir el gráfico se han utilizado elementos de la clase **System.Windows.Forms.DataVisualization.Charting** que proporciona Visual Studio. Cómo se pretendía que el gráfico se asemejara lo máximo posible al que se utilizaba anteriormente en la versión desarrollada en Delphi se eligió presentarlo en barras horizontales, un tipo de gráfico **Range bar**. Se han representado las áreas de color rojo como el tiempo de parada mientras que las parcelas verdes indican que el funcionamiento de la máquina ha sido continuo.

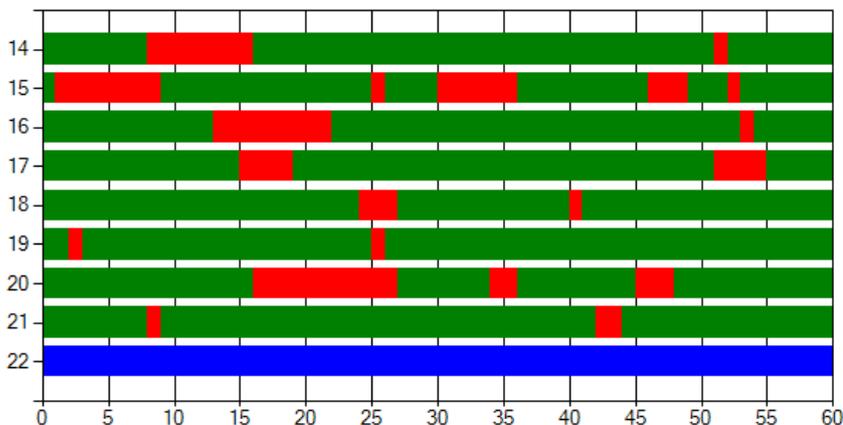


Figura 23 Visualización del gráfico actual.

4.2.4 Vista general de la pantalla Ver Partes

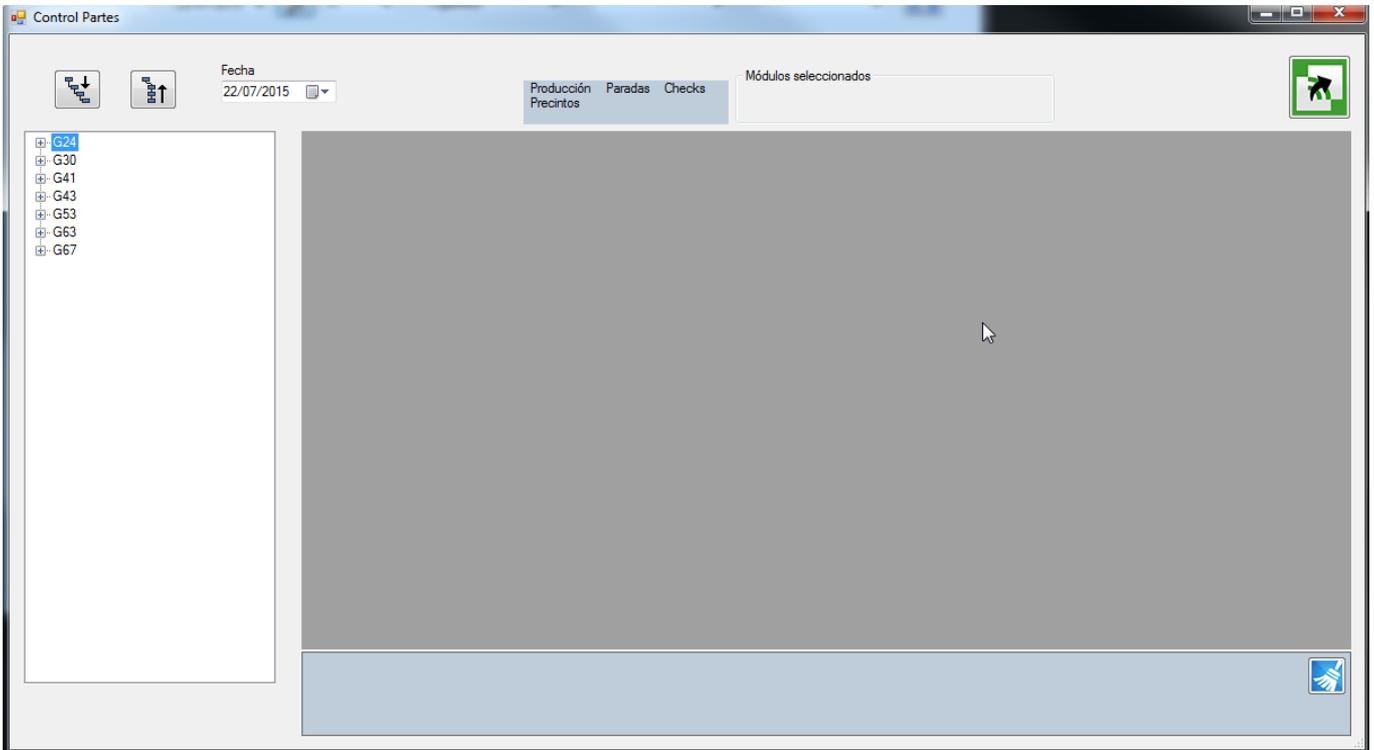
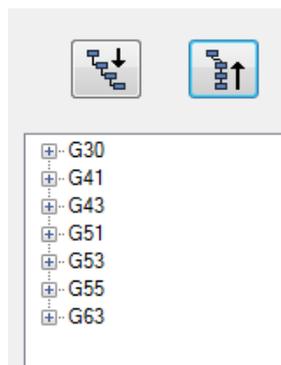
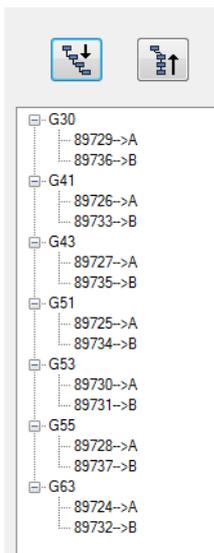


Figura 24 Vista de la nueva interfaz de la ventana Ver Partes.

Como se puede ver en la figura las partes esenciales de la ventana se han mantenido en el nuevo diseño. En una de ellas se muestran todos los partes por turno y grupo, se ha decidido utilizar un **Treeview** con la opción de expandir y contraer nodos en su totalidad.



Figuras 25 y 26 Detalle del expandido de los nodos de los grupos de partes.

Se han eliminado aquellos grupos de los que no se han guardado partes en la base de datos, el hecho de incluirlos vacíos no aporta ningún tipo de información relevante.

4.2.5 Drag & Drop

Después de utilizar esta funcionalidad en una de las asignaturas del grado, se pensó en aprovechar estos conocimientos adquiridos para mejorar la interfaz y la interacción de la ventana Ver partes. Para que dicha interacción fuera posible se han diseñado una serie de controles de usuario que representan resúmenes de las partes esenciales de un parte.

Con ello se pretendía que el usuario pudiera visualizar en un mismo contenedor las partes que él considere necesarias. Con ello se facilita la comparación entre distintos turnos, en mucho menos tiempo permite al usuario hacerse una idea general o visión global.

Una vez seleccionados los módulos que se quieren visualizar, en el contenedor se arrastra uno de los elementos mostrándose en el contenedor principal.

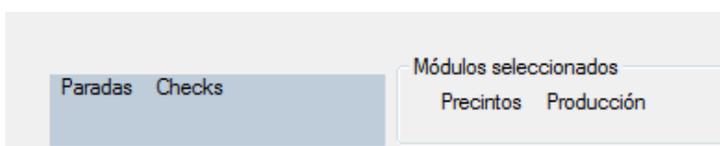


Figura 27 Contenedor de selección de las secciones del parte a visualizar.

Control de usuario de producción

Para realizar un balance general del turno con respecto a la labor que se estaba produciendo, se ha elegido mostrar la producción y desperdicio de la liadora y encajonadora del grupo.



Figura 28 Componente resumida de producción en tiempo de diseño.

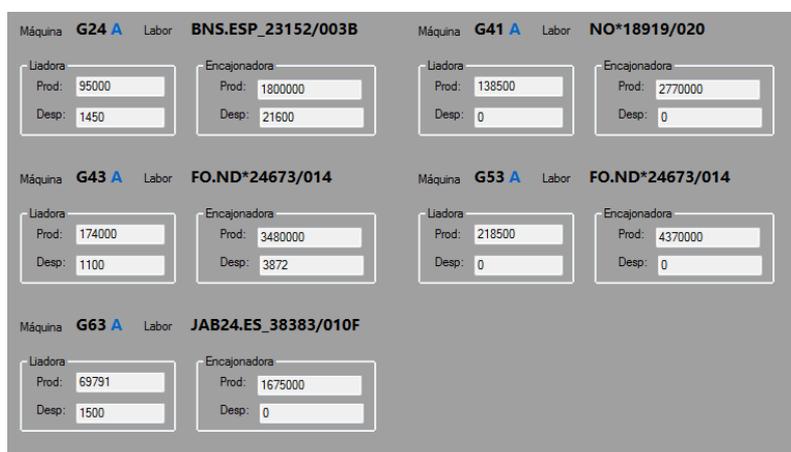


Figura 29 Módulo de producción en tiempo de ejecución.

De este modo se pueden realizar balances por líneas en el mismo turno como se puede ver en la imagen anterior.

Control de usuario de precintos

El control muestra los movimientos de precintos que se han producido.

Máquina	G41 B	Máquina	G43 B
167000	10 -> Existencia Inicial	400000	20 -> Entradas
300000	20 -> Entradas	175000	10 -> Existencia Inicial
		230000	25 -> Consumos

Figura 30 Vista del módulo de precintos en tiempo de ejecución.

Control de usuario de paradas

Para visualizar las paradas, el usuario especificó que debía tratarse de un gráfico aunque no era necesario que fuera tan específico ni elaborado como el que podía consultarse en la ventana principal. De modo que se ha utilizado de nuevo la librería *System.Windows.Forms.DataVisualization.Charting* para crear un gráfico de tipo *Pie*, indicando mediante porcentajes el tiempo de parada respecto del total del turno.

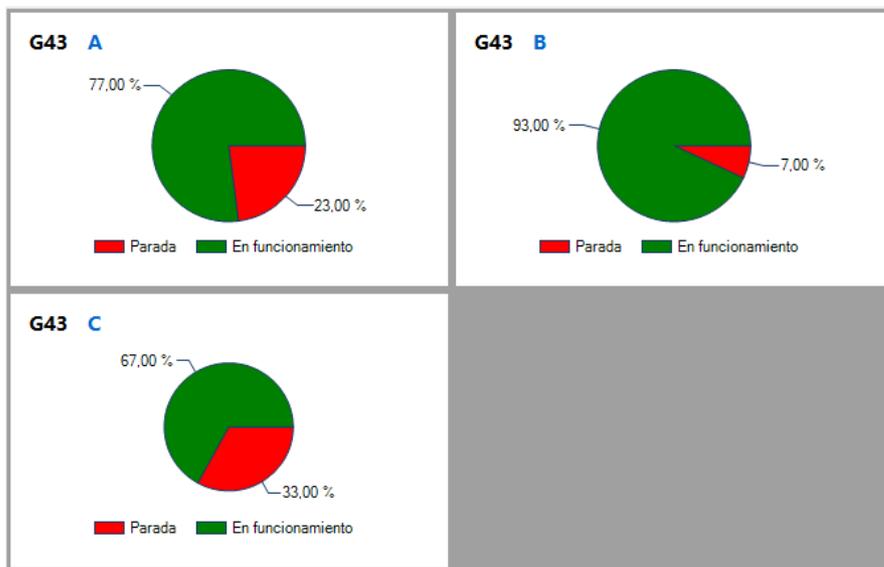


Figura 31 Vista del módulo de paradas en ejecución

Control de usuario de checks

En este control se especifican cuántos de los checks son correctos, incorrectos o no se han realizado.

Section	Checks Correctos	Checks Incorrectos	Checks No Realizados	Total
G43 A	11	2	0	13
G43 B	11	1	1	13
G43 C	12	1	0	13

Figura 32 Vista del módulo de checks en tiempo de ejecución.

Anteriormente al seleccionar uno de los partes se cargaba en el contenedor el parte entero, con lo que el usuario debía buscar qué apartado era el que le interesaba visualizar. En este aspecto se considera una mejora importante la implementación de esta parte puesto que permite focalizar la información en menor tiempo.

PRUEBAS

Las pruebas unitarias se han realizado utilizando el **Unit Test Generator** de Visual Studio. Se ha creado una clase, donde se han definido los correspondientes métodos test para probar los métodos de las capas de Persistencia y Lógica.

Ejemplos de Tests de la capa de Persistencia:

```
[TestMethod]
0 | 0 referencias
public void TestMethod18()
{
    PartesPersistencia pp = new PartesPersistencia();
    this.p = pp.getParte("72616");
    Assert.IsTrue(pp.eliminarParte(p.Codigo));
}
```

```
[TestMethod]
0 | 0 referencias
public void TestMethod1()
{
    PartesPersistencia pp = new PartesPersistencia();
    DateTime dt = new DateTime(2015,7,15);
    Parte p=pp.buscarParte(dt, "A","Cigarrillos","G43","LIA43" );
    Assert.IsNotNull(p);
}
```

Ejemplos de Tests de la capa de Lógica:

```
[TestMethod]
0 | 0 referencias
public void TestMethod19()
{
    PartesLogica pl = new PartesLogica();
    List<Ocur> lis = pl.paradas_tiempoReal("43", "B");
    Assert.IsNotNull(lis);
}
```

```
[TestMethod]
0 | 0 referencias
public void TestMethod20()
{
    PartesLogica pl = new PartesLogica();
    DateTime dt = new DateTime(2015, 7, 15);
    List<Ocur> lis = pl.getParadasHistorico("43", "B",dt);
    Assert.IsNotNull(lis);
}
```

Resumen de las pruebas ejecutadas

Jerarquía	No cubiertos (bloqu...	No cubiertos (% de bloques)	Cubiertos (bloques)	Cubiertos (% bloques)
eslmmor_ESLMNB0...	1042	49,78 %	1051	50,22 %
▸ ppftests.dll	0	0,00 %	90	100,00 %
▸ clases_basicas.dll	155	60,08 %	103	39,92 %
▸ ppf_log.dll	278	83,73 %	54	16,27 %
▸ ppf_pers.dll	609	43,10 %	804	56,90 %

CONCLUSIÓN

Este Trabajo Fin de Grado ha sido desarrollado como propuesta del departamento de Informática de Altadis, como ya se ha indicado anteriormente.

Ha sido una continuación natural tras haber estado realizando prácticas en dicha empresa durante prácticamente un año. La gestión y la coordinación del mismo dentro de la empresa como con el director del proyecto ha sido una forma de tomar experiencia sobre cómo adaptarse a los tiempos y los imprevistos respecto a la planificación que se había realizado inicialmente.

El hecho de haber escogido realizar la reimplementación de Partes de Producción me ha permitido sumergirme en la dinámica de trabajo de otros departamentos esenciales de la fábrica, como el de Producción y Mejora Operacional. Realmente la dificultad del proyecto residía en entender el funcionamiento ya no tanto de las máquinas, sino la infraestructura de comunicaciones y cómo se realizaba la extracción de datos de las mismas. Por ese lado he tenido el apoyo del responsable del departamento de Informática, Patricio Ledesma, que en todo momento ha intentado resolver mis dudas y explicarme los procesos esenciales para que el funcionamiento de la aplicación fuera consistente.

Este constituye el primer proyecto de grandes dimensiones, un contacto con la realidad que me ha permitido desarrollar aptitudes transversales necesarias para mi futura trayectoria profesional. Además he tenido la oportunidad de conocer otra plataforma y lenguaje de programación (Delphi) en los que se había basado el desarrollo de la aplicación de Partes de Producción que se está utilizando actualmente. No se ha podido realizar la reimplementación al completo puesto que durante la realización del trabajo se estaba poniendo en marcha la implantación de un nuevo mecanismo de control de presencia, de modo que la parte de la aplicación dedicada a la visualización del personal presente en fábrica no se ha podido incluir.

De todos modos la aplicación resultante ha logrado cumplir con los objetivos iniciales que se habían fijado, cubriendo las expectativas del representante del departamento de IS de la fábrica.

Tabla comparativa de horas previstas y horas reales

Código	Tarea	Descripción	Horas previstas	Horas reales
G1.0	Introducción	Tareas de preparación antes de iniciar el proyecto.	30	30
G1.1	Análisis de la versión actual	Análisis de la aplicación de Partes de Producción y del entorno en el que se utiliza.	20	20
G1.2	Planificación	Este apartado engloba la realización y descripción del EDP, las actividades que se van a realizar y como se han programado a lo largo del tiempo (diagrama de Hitos y de Gantt). También todo lo referente a la gestión de calidad, cambios y riesgos.	10	10
G2.0	Gestión	Abarca las tareas de control y gestión del proyecto, que se llevarán a cabo durante todo el ciclo de vida del mismo.	60	60
G 2.1	Memoria	La memoria forma parte de los entregables del TFG. De todos ellos es el más esencial puesto que recoge todo el trabajo desarrollado.	40	40
G 2.2	Seguimiento y control	Tareas destinadas a controlar el desarrollo y evolución del proyecto, así como a detectar posibles desviaciones y proceder a su corrección. Incluye las reuniones con el director y el responsable del departamento de IS, ambos codirectores del proyecto.	10	10
G 2.3	Calidad	Para evaluar la calidad del proyecto se tendrá en cuenta la lista de requisitos que la nueva implementación del programa tiene que incluir.	6	6
G 2.4	Cambios	Los cambios que se produzcan durante el desarrollo del proyecto se considerarán en función del estado del mismo.	4	4

G 2.5	Presentación	Se realizará la defensa del producto final como cierre del proyecto desarrollado.	8	8
G 3.0	Ingeniería Inversa	Se estudiarán las tecnologías empleadas para la gestión y funcionamiento de la anterior versión del programa para decidir que estructura se seguirá en la nueva implementación.	25	25
G 3.1	Análisis de requisitos	Descripción de las funcionalidades que cubre la actual aplicación y que influyen en la calidad de la misma.	25	25
G 4.0	Rediseño	Especificación de los cambios estructurales y visuales que se pretenden implementar en la nueva versión.	30	35
G 4.1	Diagrama de clases	Realización de un diagrama que muestra la estructura del sistema especificando sus clases.	5	5
Código	Tarea	Descripción	Horas previstas	
G 5.2	Estructura de la Base de Datos	Presentación y explicación de las bases de datos que intervienen en la aplicación y las tablas más significativas que se emplean.	20	20
G 5.3	Cambios de la interfaz	En esta tarea se pretende proponer una serie de cambios a realizar en la interfaz de la aplicación después de detectar ciertas debilidades en la navegabilidad y presentación.	5	10 (realización de una reunión más no prevista)
G 6.0	Implementación	Desarrollo de las tres capas: presentación, lógica de negocio y persistencia.	160	165
G 6.1	Persistencia	Desarrollo de la capa encargada de trabajar con las diferentes bases de datos de la aplicación.	70	70
G 6.2	Lógica de negocio	Desarrollo de la capa que se comunicará con la de presentación y la de persistencia.	10	10

Diseño y Reimplementación de Partes de Producción

G 6.3	Presentación	Desarrollo de la parte visual de la aplicación.	80	85 (costo del gráfico de Tiempo de Operación efectivo mayor de lo previsto)
G 7.0	Pruebas	Acciones destinadas a comprobar el correcto funcionamiento de la aplicación para detectar y corregir posibles fallos.	10	10
TOTAL HORAS TRABAJO FIN DE GRADO			315	325

BIBLIOGRAFÍA

Catálogo de Referencia y API de Microsoft

<https://msdn.microsoft.com/es-es/library/ms123401.aspx>

Msdn Foros

<https://social.msdn.microsoft.com/Forums/es-ES/home?forum=vcses>

Visual Studio Toolbox

<https://channel9.msdn.com/Shows/Visual-Studio-Toolbox>

Dot Net Perls

<http://www.dotnetperls.com/>

Unit Test Generator

<https://visualstudiogallery.msdn.microsoft.com/45208924-e7b0-45df-8cff-165b505a38d7>

ANEXO

Actas de reuniones

ACTA REUNIÓN I TFG: DISEÑO Y REIMPLEMENTACIÓN DE PARTES DE PRODUCCIÓN		
10-02-2015	11:00 -11:45	Departamento de IS Zona de
Reunión convocada por	Mireia Otín Ros	
Tipo de	Primera reunión con el departamento de IS	
Asistentes	Patricio Ledesma Lardiés – responsable del departamento de IS Jose María Fernandez Fuentes – Técnico IS Mireia Otín Ros – Alumno del TFG	
Orden del día		
<ol style="list-style-type: none"> 1. Presentación general del Proyecto 2. Discusión sobre los requisitos de la nueva implementación 		
Asuntos tratados y		
Presentación general del Proyecto		
Se explica las líneas generales del Proyecto, sus partes y las fases y su planificación en el tiempo. Los asistentes están de acuerdo en seguir el camino expuesto.		
Discusión sobre los requisitos de la nueva implementación		
<p>Se expone por parte del alumno las primeras conclusiones sobre el funcionamiento de la actual aplicación de Partes de Producción. A este respecto Patricio Ledesma está de acuerdo en la mayoría de los puntos y realiza aclaraciones sobre la extracción de la información de personal de fábrica que se puede consultar actualmente en Partes de Producción. Procede a explicar que durante el tiempo de desarrollo del Trabajo Fin de Grado comenzará un Proyecto gestionado en conjunto con el departamento de RRHH que tiene como objetivo la implantación de un nuevo Sistema de Control de Presencia. Por lo tanto se acuerda que esta parte por el momento no se abordará en la nueva implementación.</p> <p>Se expone brevemente la idea y la estructura en capas de cómo debe ser la nueva implementación y la metodología (iterativa-incremental) que se usará durante el desarrollo de la misma.</p> <p>Se acuerda que para la próxima reunión se procederá a aprobar el esquema de clases UML definido y las propuestas de cambio en el diseño de la capa de presentación de la aplicación.</p>		

ACTA REUNIÓN I I		
TFG: DISEÑO Y REIMPLEMENTACIÓN DE PARTES DE PRODUCCIÓN		
25-02-2015	11:00 -12:00	Departamento de IS Zona de
Reunión convocada por	Mireia Otín Ros	
Tipo de reunión	Segunda reunión con el departamento de IS	
Asistentes	Patricio Ledesma Lardiés – responsable del departamento de IS Jose María Fernandez Fuentes – Técnico IS Mireia Otín Ros – Alumno del TFG	
Orden del día		
<ol style="list-style-type: none"> 1. Presentación del esquema UML de clases básicas. 2. Discusión sobre la extracción de algunos datos de producción 3. Presentación de la estructura de métodos a implementar en la persistencia. 4. Discusión sobre los cambios que se realizarán en la interfaz. 		
Asuntos tratados y		
Presentación del esquema UML de clases básicas		
El alumno explica la estructura del mismo y el por qué de la elección de los atributos		

Discusión sobre la extracción de algunos datos de producción

Se expone por parte del alumno la incertidumbre acerca de cómo se extrae la información relacionada con las paradas de las máquinas durante la producción. Patricio Ledesma a esta cuestión responde explicando que hay una librería llamada TetServer que se puede utilizar para facilitar ese tipo de información. Se acuerda que él se encargará de proporcionar la librería junto con la información para su manejo y las variables que intervienen para la correcta extracción de la información.

Presentación de la estructura de métodos a implementar en la persistencia

El alumno procede a presentar las cabeceras de la capa de persistencia. Patricio Ledesma no tiene ninguna objeción al respecto.

Discusión sobre los cambios que se realizarán en la interfaz

En primera instancia se aborda el diseño de la ventana principal. Se establecen en consenso las siguientes mejoras:

- Se quiere mantener la estructura general de la aplicación que ahora se utiliza.
- Reducción de botones que habitualmente ya no se utilizan
- Opción de autocompletado para que la introducción de la información sea más rápida.
- Respecto a la función de búsqueda, debe de permitirse realizar la siguiente búsqueda sin la obligación de cancelar la visualización del parte que se esté mostrando.
- Se quiere mantener el mismo tipo de gráfico en la sección de Tiempo de Operación Efectivo.

A continuación se abordan aspectos a tener en cuenta en el diseño de la ventana Ver Partes:

- Debe seguir siendo una ventana de sólo visualización.
- Se incluirá un botón que permita cargar el parte que se está consultando a la ventana principal.
- Se quitarán del árbol de partes aquellos grupos de los que no exista parte en la base de datos.

Respecto a la forma de presentar el parte que se va a visualizar, el alumno propone la realización de módulos para una presentación más gráfica de las secciones importantes de un parte de producción. A este respecto Patricio Ledesma está de acuerdo. José María Fernández propone que la definición de dichos módulos debería ser consensuada con los técnicos del departamento de Mejora Operacional cuyo objetivo es implantar mejoras tras recabar y realizar cálculos con los datos de producción. Se acuerda que el alumno propondrá una reunión con Patricio Ledesma y los implicados para resolver esta cuestión.

ACTA REUNIÓN I I I TFG: DISEÑO Y REIMPLEMENTACIÓN DE PARTES DE PRODUCCIÓN		
06-03-2015	12:00 -13:00	Sala de Reuniones Control III
Reunión convocada por	Mireia Otín Ros	
Tipo de reunión	Reunión con el departamento Mejora Operacional	
Asistentes	Patricio Ledesma Lardiés – responsable del departamento de IS Juan José Durana Murillo – responsable del departamento de Mejora Operacional Eduardo Sáenz Fernández – técnico de Mejora Operacional Jesús Angel Palomar Ortega – técnico de Mejora Operacional asociado a Producción Mireia Otín Ros – Alumno del TFG	
Orden del día		
<ol style="list-style-type: none"> 1. Presentación del contexto del Proyecto 2. Presentación de las modificaciones de la interfaz a realizar 3. Presentación gráfica de la idea de los módulos o controles para la visualización del parte en la ventana Ver Partes 4. Discusión sobre los módulos presentados 		
Asuntos tratados y acuerdos		
<p>Presentación del contexto del Proyecto</p> <p>Se explica las líneas generales del Proyecto y la fase actual del mismo para poner en antecedentes al responsable y técnicos del departamento de Mejora Operacional.</p>		
<p>Presentación de las modificaciones de la interfaz a realizar</p> <p>Se expone por parte del alumno un resumen sobre los cambios que se pretenden realizar a nivel de interfaz en la nueva implementación de la aplicación.</p>		
<p>Presentación gráfica de la idea de los módulos o controles para la visualización del parte en la ventana Ver Parte</p> <p>El alumno procede a presentar un prototipo de los módulos que representarán las distintas secciones esenciales de un parte.</p> <p>Además insta a que se propongan modificaciones al respecto. Aclaraciones consensuadas sobre algunos aspectos:</p> <ul style="list-style-type: none"> - En cada módulo debe especificarse el grupo y el turno - Sería interesante el hecho de poder comparar módulos de distintas partes - El módulo que represente las paradas debe ser un gráfico de sectores, en el que se vea el tiempo de parada mediante porcentajes. - De la parte de precintos se debe incidir en el código de movimientos de precintos conservando los colores presentes en la visualización del parte en la ventana principal. - La producción notificada no es de utilidad en el modulo de producción. <p>Se acuerda que una vez realizada la implementación, los presentes técnicos del departamento de Mejora Operacional estarán presentes en la reunion de presentación de la funcionalidad de la aplicación.</p>		

ACTA REUNIÓN IV		
TFG: DISEÑO Y REIMPLEMENTACIÓN DE PARTES DE PRODUCCIÓN		
06-07-2015	9:00 -10:00	Sala de reuniones 314
Reunión convocada por	Mireia Otín Ros	
Tipo de	Primera reunion con el departamento de IS	
Asistentes	Patricio Ledesma Lardiés – responsable del departamento de IS Jose María Fernandez Fuentes – Técnico IS Juan José Durana Murillo – responsable del departamento de Mejora Operacional Eduardo Sáenz Fernández – técnico de Mejora Operacional Jesús Angel Palomar Ortega – técnico de Mejora Operacional asociado a Producción Mireia Otín Ros – Alumno del TFG	
Orden del día		
<ol style="list-style-type: none"> 1. Presentación de la funcionalidad de la aplicación 2. Discusión sobre la necesidad de corregir algún comportamiento de la aplicación 		
Asuntos tratados y		
Presentación de la funcionalidad de la aplicación		
<p>El alumno procede a explicar el funcionamiento de la nueva implementación mediante la ejecución en vivo de la aplicación.</p>		
Discusión sobre la necesidad de corregir algún comportamiento de la aplicación		
<p>Sobre el funcionamiento de la Ventana Principal:</p> <ul style="list-style-type: none"> - Cada vez que se realice alguna modificación y se cancele el parte actual o se quiera realizar una nueva búsqueda, deberá aparecer una ventana que avise sobre guardar las modificaciones en la base de datos. - Cuando se produzca un error de base de datos aparecerá una ventana de error avisando que el usuario debe contactar con el administrador. <p>Los técnicos de Mejora Operacional no han realizado ninguna observación relacionada con la ventana Ver Partes. Están de acuerdo con el comportamiento de esta parte de la aplicación.</p>		

Librería TetServer

Librería de componentes ActiveX "TetServer"

Componentes:

- **TetConnection:** Componente para acceder a la base de datos de Tet y al TetRobot
- **TetConnectionMgr:** Gestiona una batería de componentes TetConnection para que pueda haber clientes concurrentes.
- **TetQuery:** Componente en el que se define una consulta a Tet y contiene los resultados.
- **TetArray:** Componente para manejar arrays de dos dimensiones de texto o números y convertirlo a/de texto.

Instalación:

Copiar el archivo "TetServer.dll" a un directorio del ordenador. Ejecutar regsvr32.exe pasándole como parámetro el nombre del archivo copiado.

Desinstalación:

Ejecutar regsvr32.exe con la opción /u pasándole como parámetro el nombre del archivo TetServer.dll. Después se puede borrar el archivo (si está en uso no se puede borrar).

TetConnection

Este componente permite conectarse a la base de datos de Tet y al TetRobot y efectuar diversas consultas.

Propiedades

ConnectionString: string. Cadena de conexión a la base de datos. Puede ser un nombre TNS o una cadena completa.

Métodos

ExecuteSQL (sql,params:string): string. Ejecuta una sentencia SQL (o bloque PL/SQL) contenida en el primer parámetro de la función. Esta sentencia puede contener parámetros de sustitución de tipo cadena, a los que se da valor en el segundo parámetro, que es una lista de parejas nombre=valor separadas por avances de línea. Si la sentencia es de tipo select, la función devuelve un array rectangular de texto (tabuladores-linefeeds) con las filas devueltas por la consulta; si no, hace un commit y no devuelve nada.

ExecuteTetQuery (q:ITetQuery). Ejecuta una consulta especificada en el objeto de tipo TetQuery, y deja el resultado de la consulta en el mismo objeto.

TetConnectionMgr

Este componente contiene una lista de objetos de tipo TetConnection, de forma que las consultas de tipo TetQuery se ejecutan en el primer TetConnection disponible, permitiendo de esta manera que varios usuarios ejecuten consultas simultáneamente. También lleva una estadística del número de consultas, duración, etc.

Propiedades

ConnectionString: string. Cadena de conexión a la base de datos. Puede ser un nombre TNS o una cadena completa.

Métodos

AddConnection (bHist,bRT:Boolean). Añade un nuevo objeto TetConnection a su lista interna, de forma que esté disponible para los usuarios que lo soliciten. Según los parámetros bHist y bRT, se especifica si este nuevo TetConnection se usará para consultas de históricos, de tiempo real o ambos.

ExecuteTetQuery (q:ITetQuery). Ejecuta una consulta especificada en el objeto de tipo TetQuery, y deja el resultado de la consulta en el mismo objeto.

ResetStat (). Pone a 0 los contadores de estadística.

GetMgrStat ():string. Devuelve una cadena con 8 valores de estadística separados por tabulador. Los valores son:

1. Tiempo en segundos desde que se creó el componente.
2. Tiempo en segundos desde que se pusieron a 0 los contadores de estadística.
3. Número de consultas de histórico ejecutadas.
4. Número de consultas en tiempo real ejecutadas.
5. Tiempo en segundos empleado en consultas históricas.
6. Tiempo en segundos empleado en consultas en tiempo real.
7. Tiempo en segundos de espera media en comenzar una consulta histórica (debido a consultas simultáneas).
8. Tiempo en segundos de espera media en comenzar una consulta en tiempo real (debido a consultas simultáneas).

GetConnStat ():string. Devuelve una cadena con una línea de 6 estadísticas separadas por tabulador por cada TetConnection presente. Los valores de cada una de las líneas son:

1. Vale 1 si este TetConnection se usa para históricos. 0 si no.
2. Vale 1 si este TetConnection se usa para tiempo real. 0 si no.
3. Número de consultas de histórico ejecutadas.
4. Número de consultas en tiempo real ejecutadas.
5. Tiempo en segundos empleado en consultas históricas.
6. Tiempo en segundos empleado en consultas en tiempo real.

TetQuery

Este componente contiene una consulta para ser ejecutada por un TetConnection y el resultado de la misma. También proporciona unas funciones útiles para el formateo de valores numéricos o de tiempo.

La aplicación fija los parámetros de la consulta en unas propiedades del objeto. A continuación llama al método ExecuteTetQuery de un TetConnection o TetConnectionMgr. El resultado de la consulta lo guarda internamente en una lista de registros devueltos y un puntero que

inicialmente apunta al primer registro. La aplicación puede mover el puntero con diferentes métodos y acceder a los campos del registro seleccionado mediante propiedades del TetQuery.

Propiedades para parametrizar la consulta

GroupParam: string. Grupo de máquinas sobre las que se realiza la consulta. El uso de esta propiedad es incompatible con la MachineParam. Véase la propiedad UseStoredGroups para conocer su función exacta.

MachineParam: string. Máquina sobre la que se realiza la consulta. El uso de esta propiedad es incompatible con la GroupParam.

RangeParam: string. Indica sobre qué informes se realiza la consulta. Puede contener los siguientes valores:

- "" (Cadena vacía). Indica que es sobre los informes en tiempo real.
- "f" (efe). Indica que es sobre todos los informes históricos.
- "fyyyyymmddnnn". Indica que es sobre los informes cuya fecha efectiva esté dentro de los nnn primeros días desde la fecha yyyy/mm/dd.
- "l" (ele). Indica que es sobre el último informe histórico. No se puede usar con grupos de máquinas.
- 'pyyyymmddhhmmss'. Indica que es sobre el último informe histórico cuya fecha de comienzo es menor que la indicada. No se puede usar con grupos de máquinas.
- 'nyyyymmddhhmmss'. Indica que es sobre el primer informe histórico cuya fecha de comienzo sea mayor que la indicada. No se puede usar con grupos de máquinas.
- 'xyyyymmddhhmmss'. Indica que es sobre el informe histórico cuya fecha de comienzo es exactamente la indicada. No se puede usar con grupos de máquinas.

Attr1Param: string. Si esta propiedad tiene un valor distinto a la cadena vacía, indica que sólo se quieren obtener los informes históricos cuyo campo Attr1 tenga exactamente ese valor. Se ignora en informes en tiempo real.

Attr2Param: string. Si esta propiedad tiene un valor distinto a la cadena vacía, indica que sólo se quieren obtener los informes históricos cuyo campo Attr2 tenga exactamente ese valor. Se ignora en informes en tiempo real.

ProductParam: string. Si esta propiedad tiene un valor distinto a la cadena vacía, indica que sólo se quieren obtener los informes históricos cuyo campo Product tenga exactamente ese valor. Se ignora en informes en tiempo real.

TReport: string. Indica qué tipo de informe se desea obtener. No puede dejarse vacía.

OrderFields: string. Indica en qué orden se quieren obtener los informes. Consiste en una letra por cada campo de ordenación. Por ejemplo, si vale 'FA' indica que se quieren ordenar primero por fecha efectiva y luego por el atributo 1. Además de para ordenar, esta propiedad se usa en los métodos para mover el puntero de registro seleccionado. Las letras que especifican cada uno de los posibles campos de ordenación son:

UseStoredGroups: Boolean. Sólo se usa con los informes históricos por grupo de máquinas. Indica si la propiedad GroupParam se refiere a los grupos a los que pertenecía una máquina cuando se grabó el informe (True) o si se refiere a los grupos a los que pertenece una máquina actualmente (False).

Métodos para parametrizar la consulta

AddNVar (nom: string). Añade a la lista de variables de la consulta la variable numérica con nombre 'nom'.

AddCVar (nom: string). Añade a la lista de variables de la consulta la variable de texto con nombre 'nom'.

AddBVar (nom: string). Añade a la lista de variables de la consulta la variable booleana con nombre 'nom'.

AddLVar (nom: string). Añade a la lista de variables de la consulta la variable de texto largo con nombre 'nom'.

ClearVars (). Vacía la lista de variables de la consulta.

Métodos para analizar o mover el puntero de registro devuelto

Clear (). Elimina la lista de registros resultado de la consulta, como si esta no se hubiera efectuado. Se llama internamente cada vez que se modifica una propiedad de parametrización de la consulta.

Count (n: integer):integer. Cuenta, a partir del registro actual e incluyéndolo, cuántos registros hay cuyos n primeros campos de ordenación sean iguales. Si la lista no contuviera ningún registro devolvería 0. Si n=0, cuenta cuántos registros hay hasta el último de la lista.

Next (n: integer):Boolean. Si los n primeros campos de ordenación del registro actual son iguales a los del siguiente en la lista, avanza el puntero en una posición y devuelve True. Si no son iguales o no hay más registros, devuelve false. Si n=0, avanza y devuelve True si el actual no es el último. Si la lista estuviera vacía, devuelve False.

Rewind(n:integer). Retrocede el puntero hasta el primer registro cuyos n primeros campos de ordenación sean iguales a los del actual. Si n=0, se posiciona en el primer registro de la lista.

Propiedades globales para el resultado de la consulta (sólo lectura)

LastException: string. Si al ejecutar la consulta el TetConnection la interrumpiera por alguna excepción, esta propiedad contiene un texto descriptivo de la excepción; si no, es una cadena vacía.

FirstPointer: string. Si la consulta es sobre históricos de una máquina, aquí devuelve una cadena con la fecha y hora de comienzo del informe más antiguo obtenido en la consulta, en formato yyyyymmddhhmmss.

LastPointer: string. Si la consulta es sobre históricos de una máquina, aquí devuelve una cadena con la fecha y hora de comienzo del informe más moderno obtenido en la consulta, en formato `yyyymmddhhmmss`.

Propiedades para el registro actual (sólo lectura)

Machine: string. Nombre de la máquina.

MachineDesc: string. Descripción de la máquina.

StartDate: string. Fecha y hora de comienzo del informe en formato `yyyymmdd:hhmmss`.

EndDate: string. Fecha y hora de final del informe en formato `yyyymmdd:hhmmss`.

EffectiveDate: string. Fecha efectiva del informe en formato `yyyymmdd:000000`.

Attr1: string. Atributo 1.

Attr2: string. Atributo 2.

Product: string. Nombre del producto.

ProductDesc: string. Descripción del producto.

CurPointer: string. Para los informes históricos, contiene la fecha de comienzo del informe en formato `yyyymmddhhmmss`.

GetVariable(nom:string):variant. Devuelve el valor de la variable 'nom'. Si la variable no tuviera valor devuelve una cadena vacía.

GetVariableDef(nom:string; def:variant):variant. Devuelve el valor de la variable 'nom'. Si la variable no tuviera valor devuelve lo especificado en 'def'.

Métodos para formateo de texto

StrToHtml(s:string):string. Devuelve la cadena pasada como parámetro sustituyendo cada carácter por su equivalente en HTML.

TetDateToStr(d,f:string):string. Siendo 'd' una fecha-hora en formato `yyyymmdd:hhmmss`, la devuelve formateada según se especifique en el parámetro f, que es una cadena que puede contener una serie de los siguientes identificadores, separados por cualquier otro texto.

SecToHMS(s:integer):string. Convierte un número de segundos a una cadena con el formato `h:mm:ss`, donde h puede tener más de 1 dígito si se requiere.

SecToHM(s:integer):string. Convierte un número de segundos a una cadena con el formato `h:mm`, donde h puede tener más de 1 dígito si se requiere.

SecToMS(s:integer):string. Convierte un número de segundos a una cadena con el formato `m:ss`, donde m puede tener más de 1 dígito si se requiere.

TetDateDiff(s1,s2:string):integer. Devuelve la diferencia en segundos de dos fechas (s2-s1) con el formato yyyyymmdd:hmmss.

IntToDotStr(n:number):string. Devuelve el valor pasado como una cadena de texto con el valor de ese entero separando los miles, millones, etc con un punto.

FormatFloat(n:number; ndec:integer):string. Devuelve el valor pasado como una cadena de texto con tantos decimales como indique 'ndec'. Si se indica un número de decimales negativo, se utiliza su valor absoluto, pero se le quitan todos los ceros finales de la parte decimal y hasta el separador de decimales si éste fuera el último carácter.

TetArray

Este componente contiene una matriz de dos dimensiones de cadenas de texto, a cuyas celdas se puede acceder directamente. Las dos dimensiones de la matriz son indefinidas, por lo que puede contener todas las celdas que se quiera. Las propiedades RowCount y ColCount se refieren a las de una matriz que pudiera contener todas las celdas utilizadas.

Propiedades

RowCount: integer (read-only). Número de la última fila utilizada más 1.

ColCount: integer (read-only). Número de la última columna utilizada más 1.

Text: string. En lectura, devuelve una cadena de texto con RowCount líneas separadas por avance de línea y cada línea con ColCount valores de celda separados por tabulador. En escritura, carga las celdas incluidas en la cadena con el formato anterior, y modifica RowCount y ColCount para adecuarse al nuevo tamaño.

Métodos

Clear(). Borra las todas las celdas de la matriz. Las propiedades RowCount y ColCount pasan a valer 0.

GetString(row,col:integer):string. Devuelve el contenido de la celda indicada, una cadena vacía si no existiera.

GetStringDef(row,col:integer; def:string):string. Devuelve el contenido de la celda indicada. 'def' si no existiera.

GetNumber(row,col:integer):number. Devuelve el contenido de la celda indicada convertido a número, 0 si no existiera o no contuviera un formato numérico válido.

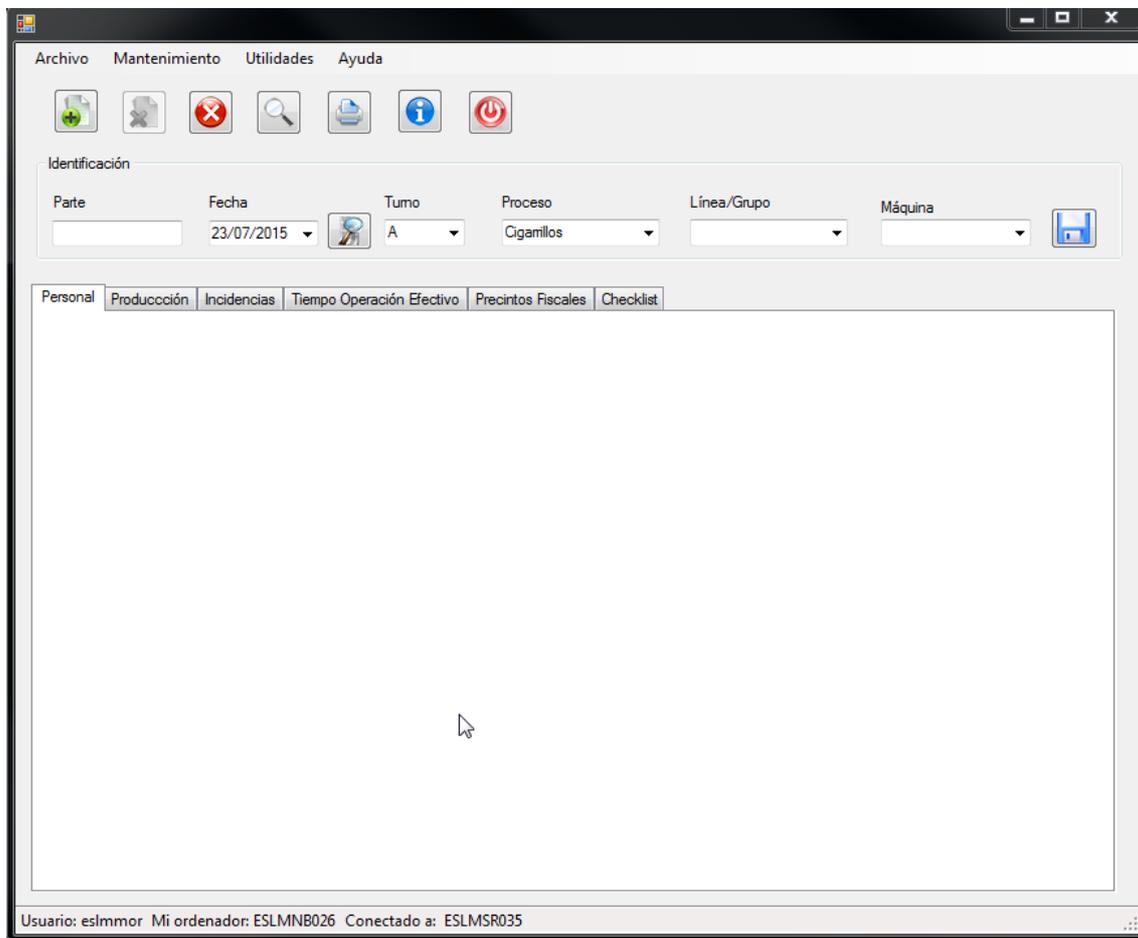
GetNumberDef(row,col:integer; def:number):number. Devuelve el contenido de la celda indicada convertido a número, 'def' si no existiera o no contuviera un formato numérico válido.

SetString(row,col:integer; s:string). Coloca 's' en la celda indicada, aumentando RowCount o ColCount si fuera necesario.

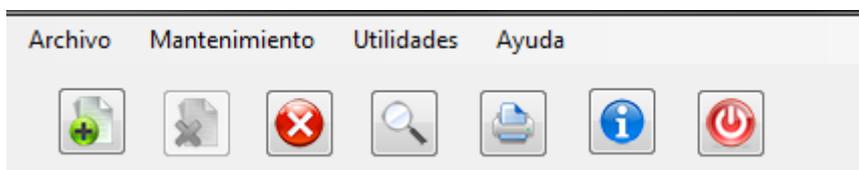
SetNumber(row,col:integer; n:number). Coloca 'n' en la celda indicada, aumentando RowCount o ColCount si fuera necesario.

Breve Guía de Usuario de Partes de Producción

Una vez ejecutado el programa aparecerá la ventana principal:



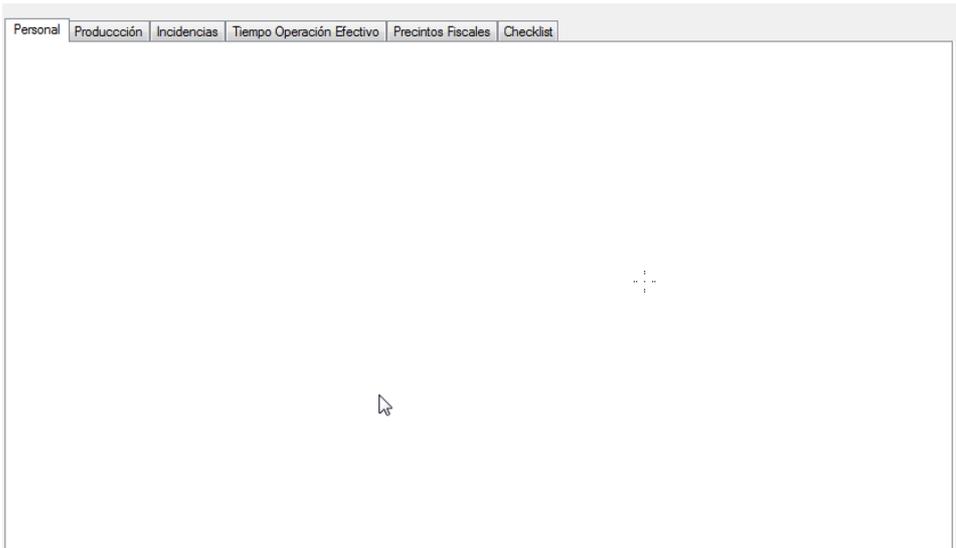
Como podemos observar en esta ventana nos aparecen seleccionados la fecha y el turno actual. Podemos distinguir varias zonas en la interfaz:



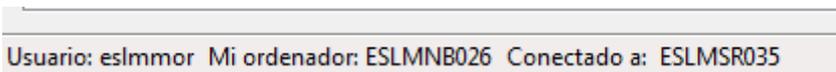
Menú principal y botones para las funcionalidades esenciales de un parte.



Zona de identificación para el cargado y modificación de partes.

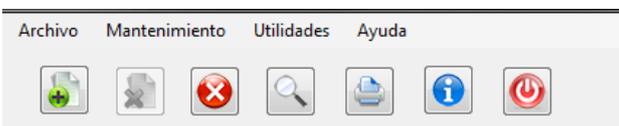


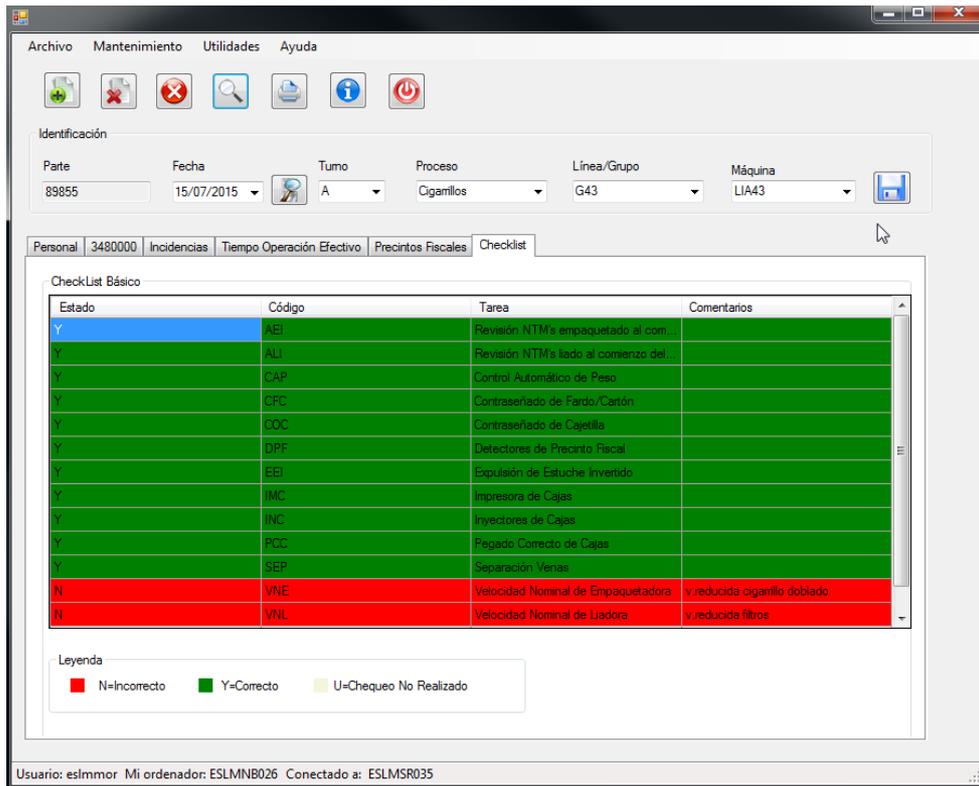
Zona de cargado de datos para la visualización de las distintas secciones de un parte.



Barra de estado indicando el usuario, ordenador y servidor al que el programa se encuentra conectado.

Inicialmente el botón de la funcionalidad de Cancelar parte se encuentra deshabilitado hasta que se carga un parte:

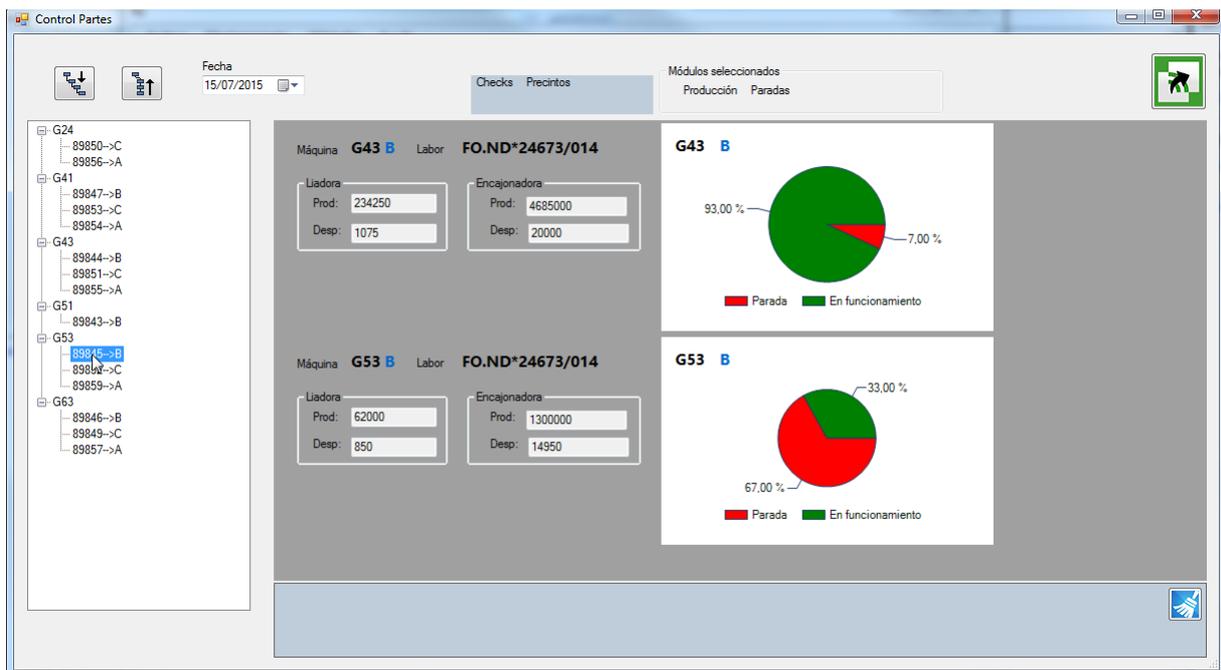




Todos los botones de la interfaz muestran el texto correspondiente a su funcionalidad:



Ventana Ver Partes



Diseño y Reimplementación de Partes de Producción

En la vista general de Ver Partes podemos observar una zona donde se encuentran todos los partes disponibles según la fecha escogida.



Con el siguiente botón se eliminan los elementos que se han arrastrado al contenedor.

Una vez seleccionado un parte del árbol podemos cargarlo en la ventana principal.

Control Partes

Fecha: 15/07/2015

Módulos seleccionados: Producción, Paradas

Máquina **G43 B** Labor **FO.ND*24673/014**

Liadora: Prod: 234250, Desp: 1075; Encajonadora: Prod: 4685000, Desp: 20000

Máquina **G53 B** Labor **FO.ND*24673/014**

Liadora: Prod: 62000, Desp: 850; Encajonadora: Prod: 1300000, Desp: 14950

93.00 % En funcionamiento, 7.00 % Parada (G43 B)

33.00 % En funcionamiento, 67.00 % Parada (G53 B)

Archivo Mantenimiento Utilidades Ayuda

Identificación

Parte: 89845, Fecha: 15/07/2015, Tumo: A, Proceso: Cigarrillos, Línea/Grupo: G53, Máquina: LIA53

Personal Producción Incidencias Tiempo Operación Efectivo Precintos Fiscales Checklist

Máquina	H. Inicio	H. Fin	Tiempo	Labor	Prod. Liadora	Desp. Liadora	Prod. Empaquetadora	Desp. Empaquetadora
G53	06:00	14:00	08:00	-> FO.ND*24673/014	1.300.000	14.950	62.000	850

Observaciones

Usuario: eslmmor Mi ordenador: ESLMNB026 Conectado a: ESLMSR035