

# **DisperSATE**

**Sistema de ayuda al trabajo en equipo de Dispersiones**

**Juan Manuel González Naranjo**

UOC - ETIS

**Director: David Gañán Jiménez**

Junio 2006

## AGRADECIMIENTOS

A mi mujer, María José, por su paciencia y comprensión, pero también por llevar a cabo mi parte de las tareas del hogar. A mis pequeñines, Alba y Mario, por todo el tiempo que no les he dedicado, en una época de su vida dónde todo el tiempo de dedicación a ellos es poco (espero recuperarlo con creces). Y, por supuesto, a mis suegros, ya que sin su presencia y ayuda con los niños todo el trabajo no hubiera sido posible.

Sería injusto no mencionar a mi jefe de turno, el Sr. Mañé, ya que su permisividad durante algunas horas de trabajo ayudó a que completara alguna entrega de documentos.

Por último, hay que reconocer el ánimo que proporciona una crítica constructiva a tiempo, así como una buena calificación del trabajo entregado. Gracias, David.

## RESUMEN DEL PROYECTO

**DisperSATE** está basado en una aplicación que podría ser el núcleo de un sistema de información de ayuda al trabajo en equipo en la planta de *dispersiones poliméricas* de BASF en Tarragona, que es una *planta química de tipo discontinuo*.

Utilizando la *tecnología .NET*, tal como corresponde al área del proyecto, se ha creado desde cero dicha aplicación, cuya puesta en producción permitirá racionalizar y automatizar los procesos de comunicación entre los trabajadores de la planta, respetando al máximo los sistemas de trabajo y atribuciones actualmente establecidos para cada persona y grupo, pero eliminando ambigüedades en la transmisión de información.

Se trata de una *aplicación web*. En el *servidor web*, situado en algún equipo de la intranet de la planta, se implementa la lógica y la gestión de la base de datos. Los usuarios acceden a la aplicación mediante un *cliente web* (navegador), siendo la interfaz de usuario una serie de páginas web, con *contenido estático y/o dinámico*, generadas por el servidor en tiempo de ejecución.

El objetivo principal de este proyecto era la realización de la aplicación, pero hay que destacar todo el proceso de aprendizaje realizado sobre la tecnología .NET y los elementos de desarrollo utilizados, fundamentalmente las *páginas ASP.NET*, el *lenguaje de programación C#*, el entorno de desarrollo de *Visual Studio.NET* y el SGBD *SQL Server*.

## ÍNDICE DE CONTENIDO

<i>Agradecimientos</i> _____	2
<i>Resumen del proyecto</i> _____	3
<i>Índice de contenido</i> _____	4
<i>Índice de figuras</i> _____	5
<b>1.- Introducción</b> _____	<b>6</b>
1.1.- Descripción y funcionamiento general de DisperSATE _____	6
1.2.- Justificación y aportación del TFC _____	7
1.3.- Objetivos del TFC _____	8
1.4.- Enfoque y método utilizado _____	9
1.5.- Planificación del TFC _____	10
1.6.- Productos obtenidos _____	12
1.7.- Descripción de los capítulos siguientes _____	12
<b>2.- Análisis de requisitos</b> _____	<b>13</b>
2.1.- Composición de la aplicación _____	13
2.2.- Requisitos funcionales _____	14
<b>3.- Arquitectura de la aplicación</b> _____	<b>15</b>
3.1.- Introducción _____	15
3.2.- MVC: Modelo – Vista – Controlador _____	15
3.3.- Adaptación de la arquitectura MVC en .NET _____	16
<b>4.- Diseño de clases</b> _____	<b>17</b>
4.1.- Clases del Modelo _____	17
4.1.1.- Diagrama estático de clases _____	17
4.1.2.- Descripción de las clases _____	18
4.2.- Clases del Controlador _____	20
4.3.- Clases de la Vista _____	21
4.3.1.- Arquitectura de la interfaz (Vista) _____	21
4.3.2.- Relación de páginas _____	24
<b>5.- Diseño de la base de datos</b> _____	<b>26</b>
5.1.- Diagrama ER _____	26
5.2.- Diagrama físico _____	27

<b>6.- Descripción de DisperSATE</b>	<b>28</b>
6.1.- Módulo de identificación y acceso	28
6.2.- Módulo de administración	30
6.3.- Módulo de producción	33
6.3.1.- Entorno del jefe de turno (JT)	33
6.3.2.- Entorno del ayudante del jefe de turno (AJT)	35
6.3.3.- Entorno de panelista (PA)	36
6.3.4.- Entorno de preparador de cargas (PR)	37
6.3.5.- Entorno de operario (OP)	37
<b>Conclusiones</b>	<b>38</b>
<b>Líneas futuras de trabajo</b>	<b>38</b>
<b>Bibliografía</b>	<b>39</b>

## ÍNDICE DE FIGURAS

Figura 1.- Etapas del TFC	10
Figura 2.- Diagrama de Gantt	11
Figura 3.- Modelo-Vista-Controlador	15
Figura 4.- Diagrama estático de clases del Modelo	17
Figura 5.- Estructura general de la interfaz de usuario	21
Figura 6.- Detalle del momento del turno en la interfaz de usuario	22
Figura 7.- Relación de páginas de DisperSATE	24
Figura 8.- Flujo de páginas de la interfaz	25
Figura 9.- Diagrama de Entidad-Relación	26
Figura 10.- Diagrama físico de la base de datos	27
Figura 11.- Portada de DisperSATE	28
Figura 12.- Control de acceso	28
Figura 13.- Relevé de un puesto	29
Figura 14.- Cambio de contraseña personal	29
Figura 15.- Índice del CM	30
Figura 16.- Ejemplo de creación de una nueva carga por el CM	31
Figura 17.- Ejemplo de modificación de una carga por el CM	32
Figura 18.- Índice del JT	33
Figura 19.- Página de supervisión de personal y puestos del JT	34
Figura 20.- Índice del AJT	35
Figura 21.- Índice de PA	36
Figura 22.- Índice de reactor	36
Figura 23.- Índice de PR	37
Figura 24.- Índice de OP	37

## **1.- INTRODUCCIÓN**

### **1.1.- Descripción y funcionamiento general de DisperSATE**

**DisperSATE** es una aplicación web que funciona como un sistema de información de ayuda al trabajo en equipo en una *planta química de tipo discontinuo*. El personal accede a la aplicación desde terminales distribuidos por toda la planta, o mediante terminales portátiles tipo PDA, que disponen de un cliente web, e interactúa con ella durante su jornada de trabajo.

Una planta química de tipo discontinuo es aquella cuyos procesos tienen un ciclo que dura varias horas, tras las cuales las instalaciones de cada proceso se detienen y se acondicionan, limpiándolas y reparándolas si es preciso, para después iniciar un nuevo ciclo de proceso. En contraposición, una planta de tipo continuo se caracteriza por unos procesos de ciclo mucho más largos, del orden de semanas, meses, o incluso más.

#### **Control de acceso**

El puesto del CM es único y tiene privilegios de administrador. El CM accede a la aplicación seleccionando “CM” en la lista de puestos e introduciendo su contraseña. Para el resto de puestos, existe un titular del puesto, que adquiere tal condición en el momento del relevo del puesto. Tras dicho relevo, los accesos a la aplicación se realizan, como en el caso del CM, seleccionando el puesto en la lista de puestos e introduciendo la contraseña del titular.

#### **Administración**

El CM proporciona las cargas de producto que se deben ir fabricando, que aparecerán en el apartado de planificación del JT, para que éste pueda seleccionarlas e iniciar su fabricación. Proporciona también reglas adicionales (sobre la limpieza de recipientes, sobre el trasiego de productos, etc.) que son interpretadas por el JT y por el AJT a la hora de planificar las tareas. No son órdenes estrictas, sino pautas que deberán plasmarse posteriormente en la planificación. También puede modificar el estatus del personal.

#### **Producción**

La actividad de los puestos de producción, en cada turno, se inicia con el relevo del turno actual y se concluye con el relevo del turno siguiente. Cada puesto realizará las tareas que le son propias en el desarrollo de la producción, poniendo en común los datos sobre las mismas.

## 1.2.- Contexto, justificación y aportación del TFC

### Los procesos de la planta de dispersiones

La planta de dispersiones de BASF, que es de *tipo discontinuo*, fabrica fundamentalmente *dispersiones poliméricas*, aunque también *soluciones poliméricas*.

Las dispersiones son *polímeros* formando una suspensión en agua, lo que les da un aspecto blanquecino y opaco, y una textura viscosa; cada producto presenta una proporción específica de agua y polímero. Estos productos son muy utilizados: bases para pinturas, pegamentos, acondicionamiento de papel, acondicionamiento de tejidos, etc., etc.

Las soluciones poliméricas están formadas por polímeros disueltos, y su disolvente puede ser agua o bien otros, por ejemplo etanol, dependiendo del producto. Su aspecto es transparente y su textura es también viscosa. Así mismo, sus aplicaciones son también muy amplias: cosmética, detergentes, etc.

Tanto en un caso como en otro, el proceso de fabricación genera los polímeros a partir de monómeros como estireno, butadieno, acrilonitrilo, acrilamida, etc., mediante una adición proporcional en un reactor de polimerización, que produce una reacción fuertemente exotérmica. Dicha reacción condiciona en gran medida el tiempo de ciclo, ya que para controlarla la adición proporcional debe tener una velocidad determinada, no más, porque podría excederse la capacidad de refrigeración del reactor.

Pero el proceso de producción se compone de mucho más que la reacción de polimerización. Parte de la adición proporcional se realiza mediante líneas directas de adición de monómeros, pero otra parte se realiza desde un depósito llamado “reactor de preparación”, en el que se debe preparar previamente una mezcla de diferentes ingredientes, como agua, emulsionantes y otros, muchos de ellos extraídos de bidones y contenedores. Por otro lado, tras la polimerización el producto debe acondicionarse para cumplir con las especificaciones (ph, monómeros residuales, viscosidad, concentración, etc.). Una vez acondicionado, debe enviarse al parque de almacenamiento mediante líneas de trasiego.

Además de la producción, se realiza el mantenimiento y limpieza de las instalaciones, tareas de carácter mecánico que son llevadas a cabo por el mismo personal de producción. Estas tareas incluyen el desmontaje y posterior montaje de los recipientes, así como el uso de instalaciones de limpieza con agua a alta presión (300 kg/cm<sup>2</sup>).

### El personal de la planta de dispersiones

El personal de producción se distribuye en una serie de puestos. En primer lugar, el puesto de administración de producción, ocupado por el *contramaestre* (CM), con *horario normal*. Existe un sistema de producción de *turno rotativo*, con 5 turnos, que se alternan en horarios de mañana, tarde y noche. En cada turno hay 14 personas: el *jefe de turno* (JT); el *ayudante del jefe de turno* (AJT); 7 trabajadores con categoría de *panelista*, que ocupan de forma rotativa, cuatro puestos de panel en la *sala de control* (PA1, PA2, PA3, PA4), y tres de *preparación de cargas* (PR1, PR2, PR3); y por último 5 *operarios* (OP1, ..., OP5).

## Justificación

Dado que el autor de este TFC es un miembro del equipo de producción, con 9 años de experiencia, 4 en un puesto de operario y 5 en el de panelista, podría dominar adecuadamente el punto de vista del cliente en la realización del proyecto. Por otra parte, los conocimientos adquiridos durante los estudios de ETIS, pero sobre todo el entusiasmo y la motivación que supone el realizar un proyecto aplicado al trabajo diario que ha proporcionado al autor el medio de vida hasta el momento, auguran un resultado satisfactorio.

## Aportación

En cuanto a métodos de trabajo y organización interna de la planta de Dispersiones, se puede observar una gran falta de implantación de *TTIC* que optimicen el trabajo diario. De hecho una gran parte de las tareas habituales carecen de la más mínima infraestructura informática, a nivel de organización, coordinación y gestión del equipo humano.

La aplicación **DisperSATE** facilitará el trabajo de cada individuo y favorecerá la coordinación y comunicación a distintos niveles, respetando al máximo los sistemas de trabajo y atribuciones actualmente establecidos para cada persona y grupo.

La información contenida en el sistema se actualizará constantemente, recuperándola y utilizándola durante cada jornada de trabajo, pudiendo acceder a ella de forma segura y deslocalizada, a través de diferentes terminales distribuidos por toda la planta gracias a una intranet ya existente, o incluso mediante terminales móviles tipo PDA, con la especificación EX para evitar riesgo de explosión. De esta manera se eliminarán ambigüedades en la transmisión de información, que suelen ser fuente de errores, tiempos muertos y retrasos en la producción,

## 1.3.- Objetivos del TFC

### Objetivo principal

Crear desde cero la aplicación **DisperSATE**. La intención final es ponerla en producción para aumentar la productividad actual mejorando la comunicación, aunque esto es ya un objetivo personal, al margen del alcance de este TFC.

Se puede considerar como parte del objetivo principal la iniciación y profundización en el uso de la tecnología .NET, dado el papel que juega dicha tecnología en el panorama actual de tendencias de desarrollo de software.

### Objetivos personales

- Implantación de la aplicación, aunque este punto no se considerará fundamental y quedará condicionado por el tiempo disponible tras la consecución del objetivo principal.



- Actualmente existen estudios sobre los tiempos de cada tarea del proceso de producción que podrán compararse con los tiempos obtenidos tras la implantación de la aplicación y comprobar si se ha obtenido una mejora real.
- Creación de componentes reutilizables en futuras aplicaciones, maximizando la fiabilidad.
- Conclusión de los estudios de ingeniería técnica en informática de sistemas.
- Toma de contacto con la práctica profesional real en relación a la implantación de sistemas informáticos y desarrollo de software nuevo.

#### **1.4.- Enfoque y método utilizado**

Teniendo en cuenta que ésta es una aplicación destinada a facilitar el trabajo diario de un grupo de personas habituadas a su manera de hacer las cosas, reticentes a los cambios, y teniendo en cuenta además que la posible implantación de la misma tiene que suponer un proceso fácil y rápido, se ha previsto un producto final sencillo de aprender, en un principio, y de utilizar continuamente durante la actividad habitual.

La estética es muy simple, casi “espartana”, para no distraer la actividad del trabajador mientras se utiliza. Se contempla la posibilidad de usar, además de terminales fijos en puntos estratégicos, terminales portátiles tipo PDA, que algunos trabajadores necesitarían.

La aplicación se ha dividido, para distinguir los tipos de funcionalidades, en tres subsistemas:

- **Subsistema de administración.** Es el encargado de gestionar la entrada de nuevas cargas, reglas de trabajo y estatus del personal, incluida la asignación de contraseña. Solo podrá acceder a él el CM.
- **Subsistema de producción.** En él se realiza la asistencia al trabajo en equipo, guardándose en la base de datos todo tipo de acciones y datos generados durante el proceso de producción por los diferentes actores. Podrán acceder a él todos los usuarios, previa identificación.
- **Subsistema de identificación y acceso.** Se encarga de permitir o denegar el acceso a un puesto mediante la contraseña personal del titular del puesto. Los usuarios, una vez acceden al puesto, pueden cambiar la contraseña personal.

La metodología que se ha utilizado para construir la aplicación es la que comúnmente se utiliza en la actualidad para el desarrollo de software nuevo, adaptada a los requisitos en la realización del TFC:

En primer lugar se diseña un plan de trabajo razonable, que permita llevar a cabo los objetivos previstos; a continuación se realiza el análisis de requisitos, el diseño, orientado a objetos, y el primer prototipo, consistente en una serie de páginas web con el aspecto provisional de la aplicación, que permita describir su funcionamiento y ayudar a realizar cambios adicionales, tanto de aspecto como de funcionalidades. Finalmente se realiza la implementación, las pruebas y corrección de errores, y la generación de la documentación.

## 1.5.- Planificación del TFC

ETAPAS DEL TFC		
Etapa	Documentos	Actividades
<u>Planificación del TFC.</u>  <i>DESDE: 28/2/2006</i> <i>HASTA: 13/3/2006</i>	Plan de trabajo del TFC.  <b>jgonzaleznar_platreball</b>  <b>(PAC1)</b>	Recopilación y lectura de documentación sobre .NET.  Instalación del entorno de desarrollo .NET.  Selección de un proyecto.  Planificación temporal (etapas y diagrama de Gantt).
<u>Análisis.</u>  <i>DESDE: 14/3/2006</i> <i>HASTA: 29/3/2006</i>	Documento de análisis.  <b>jgonzaleznar_analisi</b>  Prototipo.  <b>jgonzaleznar_prototipo_v1_0</b>	Definir detalladamente las necesidades de información que deberá resolver el software al margen de la tecnología que se utilizará para el desarrollo del proyecto.  Diseño de un prototipo (interfaz de usuario).
<u>Diseño.</u>  <i>DESDE: 30/3/2006</i> <i>HASTA: 18/4/2006</i>	Documento de diseño.  <b>jgonzaleznar_pac2</b>  <b>(PAC2)</b>  El prototipo de la fase anterior fue aceptado por el consultor como prototipo definitivo.	Arquitectura general del sistema.  Estructuras de datos.  Interfaces de usuario.
<u>Implementación.</u>  <i>DESDE: 19/4/2006</i> <i>HASTA: 18/5/2006</i>	Código de los programas. Carpeta del servidor web con todos los ficheros de los programas.  <b>DisperSATE</b>  Guión para generar la BD.  <b>guionSQLPrueba</b>  Documentación de los programas y manual de usuario.  <b>jgonzaleznar_pac3</b>  <b>(PAC3)</b>	Implementación del diseño con la tecnología .NET.  Prueba del software siguiendo la especificación de la prueba de la fase de diseño.  Localizar y corregir los errores que se detecten.
<u>Presentación.</u>  <i>DESDE: 18/5/2006</i> <i>HASTA: 19/6/2006</i>	Memoria del TFC.  <b>jgonzaleznar_memoria</b>  Presentación virtual.  <b>jgonzaleznar_presentacio</b>	Redacción de la memoria.  Diseño de la presentación.

Figura 1 – Etapas del TFC

## DIAGRAMA DE GANTT

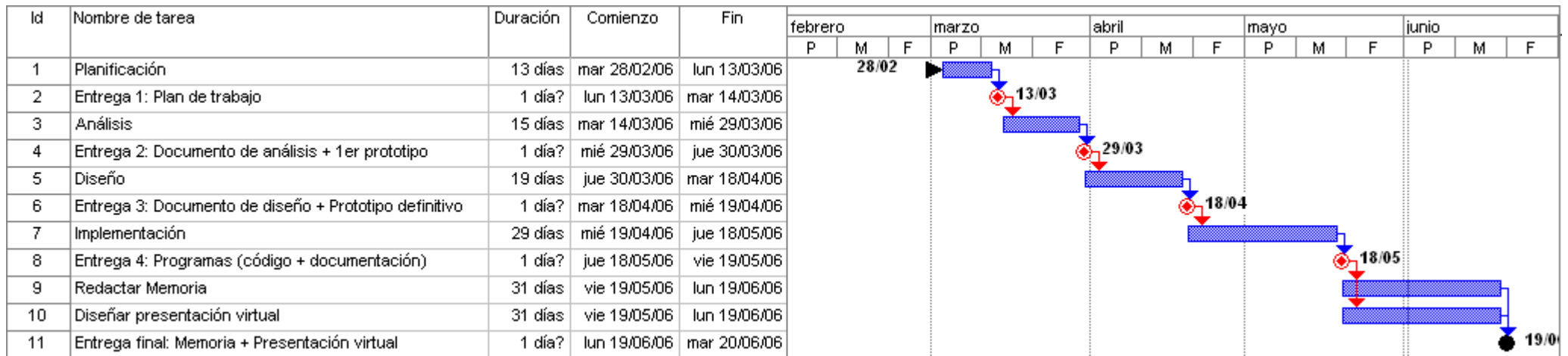


Figura 2 – Diagrama de Gannt

## **1.6.- Productos obtenidos**

Aunque ya se habían incluido en el apartado anterior, a modo de información complementaria, en este apartado se van a enumerar los productos generados durante la realización de este TFC para las diferentes entregas, en su orden cronológico:

- Documento del plan de trabajo.
- Documento de análisis de requisitos.
- Prototipo de la aplicación, formado por páginas html creadas mediante Microsoft FrontPage y un editor de texto.
- Documento de diseño.
- La aplicación DisperSATE, implementada mediante Microsoft Visual Studio .NET.
- Manual de instalación de la aplicación y manual de usuario, en el documento de implementación.
- Memoria del TFC.
- Presentación virtual realizada con Microsoft Power Point.

## **1.7.- Descripción de los capítulos siguientes**

Los capítulos que vienen a continuación son un resumen de las etapas del TFC. No se describe detalladamente cada etapa para no aumentar de forma exagerada la extensión de la memoria. Los detalles se pueden encontrar en los documentos relacionados en el apartado anterior.

El segundo capítulo se refiere a la etapa de análisis, centrándose en los requisitos, y sin tener en cuenta los casos de usos por razón de extensión.

El tercer capítulo explica el modelo arquitectónico de la aplicación y la manera en que se ha aplicado en el caso de **DisperSATE**.

El cuarto capítulo trata sobre la base de datos de la aplicación: cómo se ha diseñado y la manera de utilizarse.

En el capítulo quinto se explica el diseño de clases, desde el diseño inicial, hasta la implementación final, tanto de las clases de la interfaz y los controladores, como las clases del modelo, con el acceso a la base de datos.

El capítulo sexto describe DisperSATE, sus principales funcionalidades y elementos.

Para terminar, se incluyen las conclusiones y un apartado sobre líneas futuras de trabajo.

## 2.- ANÁLISIS DE REQUISITOS

El análisis de requisitos es el punto de partida en la realización práctica de un proyecto, y posiblemente una de sus fases más importantes. En el análisis tiene que establecerse todo lo que se quiere que haga la aplicación. Para ello se deberían realizar una serie de reuniones con el cliente para dejar definidos todos los aspectos posibles. En el caso de **DisperSATE**, el autor jugará también el papel de cliente, puesto que forma parte de la plantilla de producción de la planta, con 9 años de experiencia, 4 en un puesto de operario y 5 en el de panelista, y conoce las funciones del contraamaestre, del jefe de turno y del ayudante, pudiendo dominar así adecuadamente su punto de vista.

### 2.1.- Composición de la aplicación

La aplicación se ha dividido en tres subsistemas:

- **Subsistema de administración.** Es el encargado de gestionar la entrada de nuevas cargas, reglas de trabajo y estatus del personal, incluida la asignación de contraseña. Solo podrá acceder a él el CM.
- **Subsistema de producción.** En él se realiza la asistencia al trabajo en equipo, guardándose en la base de datos todo tipo de acciones y datos generados durante el proceso de producción por los diferentes actores. Podrán acceder a él todos los actores, previa identificación.
- **Subsistema de identificación y acceso.** Se encarga de la identificación de cada usuario mediante la contraseña de su puesto en su turno y una contraseña personal. No será posible el cambio de contraseña de puesto por parte de los usuarios, solo el CM está autorizado para ello, en el subsistema de administración. Los usuarios si podrán cambiar las contraseñas personales.

En el futuro, se podrían incluir otros subsistemas, para mejorar la aplicación (no se han tenido en cuenta para este TFC porque el tiempo disponible no hubiera permitido concluirlos):

- **Subsistema de evaluación del rendimiento.** Se encargaría de recuperar la información sobre la producción realizada por cada turno y elaborar perfiles de rendimiento. Su aplicación inmediata sería el establecimiento de un sistema de incentivos adicionales basado en objetivos de producción.
- **Subsistema de importación/exportación de datos.** Se encargaría de enlazar la base de datos con el sistema SAP/R3 de la fábrica, con el objeto de disponer de las existencias de materias primas en el almacén y de productos terminados en el parque de tanques.

## **2.2.- Requisitos funcionales**

Según las necesidades del cliente, se han identificado los siguientes requisitos:

- La aplicación debe ser fácil de instalar y de acceder, aprovechando para ello la infraestructura disponible en la planta: equipos de sobremesa conectados en una red de área local, con sistema operativo Windows NT y XP. Se tratará de una aplicación web, con un servidor web situado en uno de los equipos, realizándose el acceso mediante los navegadores Explorer de los mismos.
- Dada la distribución física del espacio de trabajo, con un edificio principal con 6 pisos de altura, dónde se ubican los reactores y la sala de control, y un parque de almacenamiento y un almacén de materias primas a una cierta distancia del edificio principal, se impone la necesidad de habilitar el acceso mediante dispositivos móviles. Por ello se debe diseñar una interfaz sencilla e intuitiva, fácil de visualizar en el equipo, en cualquier situación ambiental y de actividad (por ejemplo, cuando se intenta montar una línea de trasiego, o cuando se va a buscar materias primas al almacén).
- En la introducción de datos se minimizará la necesidad de escribir texto, utilizándose, siempre que sea posible, elementos de selección de valores, como por ejemplo listas desplegables. La finalidad será reducir al máximo la posibilidad de cometer errores en las entradas y agilizar la operación, dado que en algunos momentos la actividad es frenética y no es posible detenerse demasiado para cumplimentar formularios.
- El acceso a la aplicación requiere la selección del puesto del usuario y la introducción de su contraseña personal. La contraseña inicial de cada usuario la establecerá el convalidador al dar de alta al trabajador; el convalidador podrá cambiar dicha contraseña si es necesario. El usuario, una vez haya accedido, dispondrá de la posibilidad de cambiar también su contraseña.
- La interfaz de la aplicación, independientemente del perfil de usuario, deberá ofrecer información contextual del momento de trabajo: la hora actual, el turno que está trabajando y el turno que debe entrar a continuación, y si es el momento del relevo, para lo que podríamos considerar la última media hora de la jornada (ver el apartado 1.1. de la memoria para obtener información de los horarios de trabajo).
- El convalidador podrá acceder a la aplicación en cualquier momento, dado que será el administrador único y no está sometido al sistema de turnos. Los demás usuarios podrán acceder a la aplicación después de realizar el relevo de su puesto. Ya no podrán acceder a él cuando hayan sido relevados. Solo podrá relevar un puesto un integrante del turno que debe entrar a continuación del existente.
- Cada usuario dispondrá de las funcionalidades propias de su puesto de trabajo. El convalidador, al ser el administrador y conocer las contraseñas, podrá acceder a cualquier puesto de trabajo en cualquier momento.

Del guión textual de los actores se extrae la información de los casos de uso, que no se van a incluir en esta memoria por cuestión de espacio. Se puede obtener más información en el documento de análisis de requisitos (*jgonzaleznar\_analisi*).

## 3.- ARQUITECTURA DE LA APLICACIÓN

### 3.1.- Introducción

Hoy por hoy la mejor forma de diseñar aplicaciones web, según las opiniones que se pueden leer en diferentes fuentes, esta basada en el patrón de diseño arquitectónico MODELO-VISTA-CONTROLADOR (MVC).

Si tuviéramos todo el código de la aplicación en una sola página ASP.NET, sería difícil el mantenimiento y actualización de la aplicación. Además deberíamos introducir código redundante en muchas partes de la misma.

Poniendo el código en diferentes módulos independientes eliminaríamos la redundancia, pero añadiríamos más complejidad, lo que incrementaría el problema del mantenimiento, actualización y mejora.

El patrón de diseño MVC consigue eliminar redundancia en el código, aporta flexibilidad y no penaliza excesivamente la complejidad.

### 3.2.- MVC: Modelo – Vista – Controlador

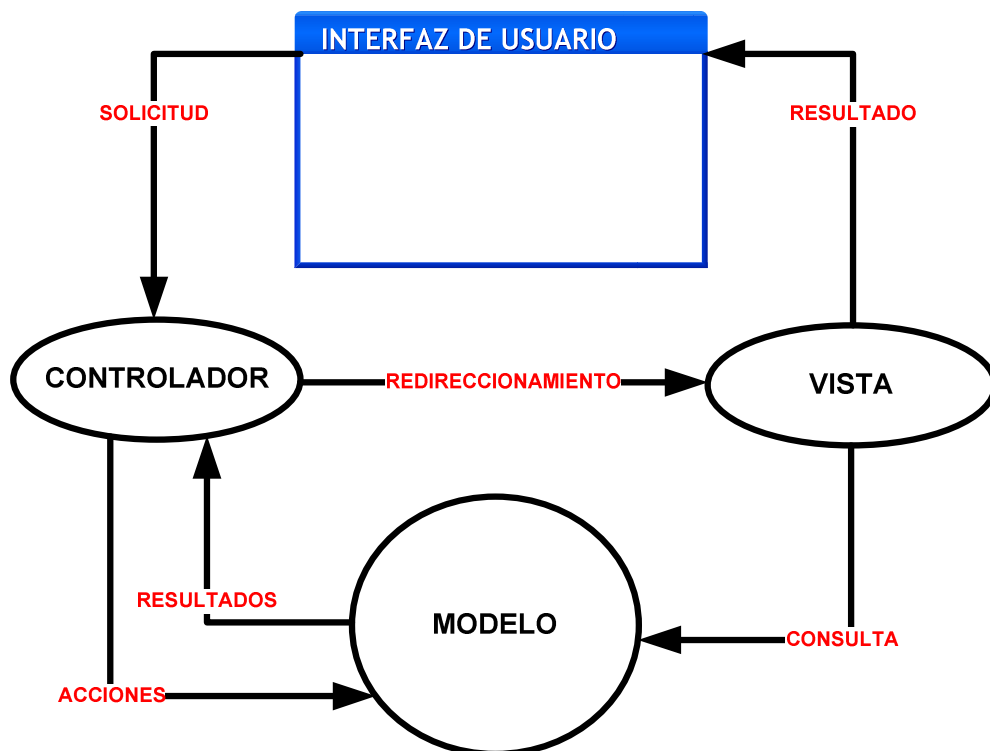


Figura 3 – Modelo-Vista-Controlador

En el patrón de diseño MVC se distinguen tres componentes:

**Modelo**.- Contiene la lógica y los datos del dominio de la aplicación. Define el modelo de clases del problema sin tener en cuenta la presentación. Incluye el código de las partes comunes y los cálculos que se tienen que hacer más de una vez. Es decir, implementa el comportamiento común de las diferentes partes de la aplicación.

**Vista**.- Consulta el estado del modelo y lo muestra al usuario. Se definirá una vista para cada forma diferente de presentar los datos al usuario: una tabla, un gráfico, etc. Una vista no contiene el código que permite acceder a los datos, sino que éste se encuentra en el modelo.

**Controlador**.- Los controladores se encargan de interpretar las acciones del usuario, ya sea para obtener vistas o bien para solicitar acciones sobre el modelo.

Las vistas y los controladores dependen del modelo, pero el modelo no depende ni de las vistas ni de los controladores. Esta es la manera de independizar la presentación y el modelo, de forma que el modelo se puede diseñar y probar totalmente al margen de la presentación, y ésta, a su vez, puede consistir en una serie de vistas simultáneas y diferentes.

Por otro lado, si la aplicación se tuviera que realizar entre diferentes personas, cada una podría realizar su tarea de forma autónoma. Por ejemplo, unas podrían dedicarse al diseño de las vistas y otras a la implementación del modelo, de forma totalmente independiente.

### **3.3.- Adaptación de la arquitectura MVC en .NET**

**Arquitectura MVC pasiva**.- Es el tipo de arquitectura utilizada en aplicaciones web. Se caracteriza porque las vistas solo se actualizan cuando se realiza alguna acción a través del controlador, independientemente de los cambios que pueda sufrir el modelo. Los controladores solo tendrán el código necesario para que los usuarios puedan utilizar la vista.

**Las vistas**.- Las vistas estarán formadas por páginas ASP.NET.

**Los controladores**.- Habrá uno para cada página que se presente al usuario. Éste se encargará de recibir la solicitud de página del cliente, de realizar las operaciones sobre el modelo y de determinar cual es la siguiente página que se debe presentar al usuario. En ASP.NET los controladores son clases que gestionan los eventos generados por la página del usuario. Solo hay que determinar que acción utilizar frente a un evento concreto. La redundancia de código en los controladores se puede solucionar mediante la herencia. Se define una clase con las funciones comunes, y otras más que heredan lo común e implementan las peculiaridades de cada controlador.

**Las clases del modelo**.- En un principio, las clases del modelo iban a ser de dos tipos: las clases de entidad y las de acceso a la base de datos, una por cada clase de entidad. De esta forma se pretendía distribuir mejor el trabajo, especializando las clases que se dedicaban a modificar la persistencia. Más tarde, la falta de tiempo aconsejó prescindir de esta distinción, de forma que hay un solo tipo de clases en el modelo, las de entidad, que al mismo tiempo implementan la lógica de acceso a la base de datos.



## 4.- DISEÑO DE CLASES

### 4.1.- Clases del Modelo

#### 4.1.1.- Diagrama estático de clases

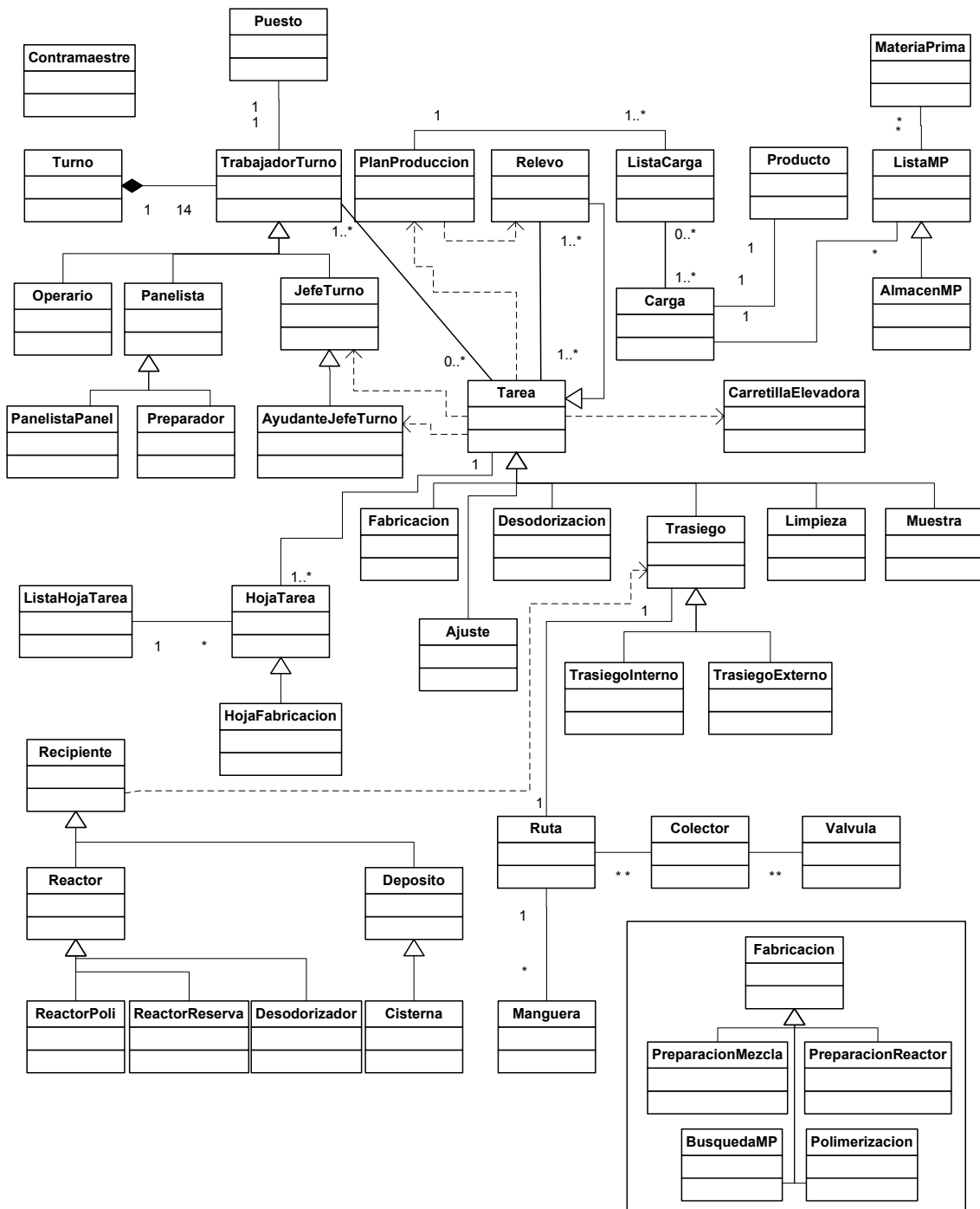


Figura 4 – Diagrama estático de clases del Modelo

#### **4.1.2.- Descripción de las clases**

En un principio, las clases del modelo iban a ser de dos tipos: las clases de entidad y las de acceso a la base de datos, una por cada clase de entidad. De esta forma se pretendía distribuir mejor el trabajo, especializando las clases que se dedicaban a modificar la persistencia. Más tarde, la falta de tiempo aconsejó prescindir de esta distinción, de forma que hay un solo tipo de clases en el modelo, las de entidad, que al mismo tiempo implementan la lógica de acceso a la base de datos. En este apartado voy a describir brevemente cada una de las clases anteriores para establecer su significado y utilidad.

Voy a utilizar una lista ordenada alfabéticamente por el nombre de las clases. En caso necesario se puede localizar con facilidad una clase en el diagrama para ayudar a comprender su papel en el conjunto. Todas las clases tendrán sus atributos, uno o más constructores y los métodos necesarios, pero no se van a describir aquí por cuestión de espacio.

**Clase Ajuste.-** Hereda de la clase **Tarea**. Describe la tarea de ajuste de una carga recién fabricada, en la que se deben añadir los aditivos finales para que los parámetros de análisis del producto queden dentro de especificaciones.

**Clase BusquedaMP.-** Hereda de la clase **Fabricacion**. Describe una subtarea de fabricación consistente en ir a buscar las materias primas necesarias al almacén.

**Clase Carga.-** Describe una carga de fabricación de producto.

**Clase Cisterna.-** Hereda de la clase **Deposito**, que hereda a su vez de **Recipiente**. Describe una cisterna de almacenamiento de producto acabado.

**Clase Colector.-** Describe un colector de líneas de trasiego.

**Clase Deposito.-** Hereda de la clase **Recipiente**. Describe un recipiente que carece de agitación.

**Clase Desodorizador.-** Hereda de la clase **Reactor**. Describe un reactor destinado a desodorización física (proceso consistente en la inyección de vapor en el producto acabado para arrastrar moléculas residuales de monómero de hidrocarburos aromáticos, como estireno, butadieno, ACN, etc., que causan un olor característico, no deseable por parámetros de calidad).

**Clase Desodorizacion.-** Hereda de la clase **Tarea**. Describe la tarea de desodorización física.

**Clase Fabricacion.-** Hereda de la clase **Tarea**. Describe la tarea de fabricación de una carga nueva de producto.

**Clase Limpieza.-** Hereda de la clase **Tarea**. Describe una tarea de limpieza de un recipiente.

**Clase Manguera.-** Describe una manguera de trasiego de producto acabado. Se trata de una manguera de material plástico con un blindaje especial para resistir presiones de hasta 15 bar. Sufren un cierto desgaste con el uso prolongado, perdiendo así parte de su resistencia y poniendo en peligro la integridad física de los trabajadores, por eso la descripción incluirá información actualizada sobre su estado.

**Clase MateriaPrima.-** Describe una materia prima de las utilizadas en el proceso de producción.

**Clase ListaMP.-** Contiene una lista de objetos de clase MateriaPrima, asociando a cada uno de ellos un valor, que es la cantidad de materia prima.

**Clase Muestra.-** Hereda de la clase **Tarea**. Define una tarea de obtención de muestra.

**Clase Polimerizacion.-** Hereda de la clase **Fabricacion**. Describe una subtarea de fabricación que consiste en controlar la reacción de polimerización de una carga en un reactor.

**Clase PreparacionMezcla.-** Hereda de la clase **Fabricacion**. Describe una subtarea de fabricación que consiste en preparar una mezcla de monómeros, emulsionantes y disolvente que más tarde se debe adicionar durante la polimerización.

**Clase PreparacionReactor.-** Hereda de la clase **Fabricacion**. Describe una subtarea de fabricación que consiste en acondicionar y comprobar la estanqueidad del reactor antes de comenzar la polimerización.

**Clase Producto.-** Describe un producto del catálogo de productos que ofrece la marca.

**Clase Puesto.-** Describe un puesto de trabajo dentro de la planta de producción.

**Clase Recipiente.-** Describe un recipiente genérico de la planta de producción.

**Clase Reactor.-** Hereda de **Recipiente**. Describe un recipiente con agitación y camisa, con circuito de agua y vapor, para calentar y enfriar el contenido.

**Clase ReactorPoli.-** Hereda de **Reactor**. Describe un reactor de polimerización, que es aquel en el que se realiza la reacción de polimerización.

**Clase ReactorPreparacion.-** Hereda de **Reactor**. Describe un reactor de preparación de la mezcla de monómeros.

**Clase ReactorReserva.-** Hereda de **Reactor**. Describe un reactor de acondicionamiento de producto acabado.

**Clase Relevo.-** Hereda de **Tarea**. Describe una tarea de relevo, en la que hay que preparar el informe para el trabajador entrante en el mismo puesto y realizar dicho relevo.

**Clase Tarea.-** Describe una tarea genérica.

**Clase TrabajadorTurno.-** Describe un trabajador a turnos.

**Clase Trasiego.-** Hereda de **Tarea**. Describe una tarea de Trasiego.

**Clase Turno.-** Describe un turno de producción.

**Clase Valvula.-** Describe una válvula de las líneas de trasiego de producto acabado.

## 4.2.- Clases del Controlador

Habr  un controlador para cada p gina ASP.NET, que recoger  las peticiones y pasar  los eventos al modelo, para que  ste interact e con la base de datos. Tras obtener los resultados, el controlador llamar  a la siguiente p gina ASP.NET y le pasar  los datos correspondientes.

La raz n de utilizar un controlador para cada p gina ASP.NET, y no uno general, es que voy a trabajar con Microsoft Visual Studio .NET, que permite, al mismo tiempo que se crea una p gina ASPX, crear una p gina controladora “code-behind” asociada, con extensi n .VB, si se utiliza Visual Basic .NET, o .CS, si se utiliza C#. De esta forma se simplifica bastante el trabajo.

Todas estas clases tienen unos m todos comunes:

### **private void Page\_Load(object sender, EventArgs e)**

Es el m todo que carga la p gina cuando se abre. Es aqu  d nde tenemos que especificar, por ejemplo, que se llene una tabla con datos, o que inicialice listas de elementos.

### **override protected void OnInit(EventArgs e)**

Este m todo tambi n se llama al cargar la p gina y su papel es llamar a InitializeComponent y al m todo OnInit de la clase padre.

### **private void InitializeComponent()**

Es un m todo necesario para tener un formulario web. Llama al m todo Page\_Load.

Las clases controladoras heredan de la clase padre *System.Web.UI.Page*. En el apartado de dise o de la vista se ha incluido un diagrama d nde aparecen todas las p ginas que formar an la vista, obtenidas en el an lisis, y una tabla d nde se asocia cada una de estas p ginas con la p gina .ASPX real que la contiene, cuyo nombre ser  tambi n el nombre de la clase del controlador, y cuyo c digo se implementar  en la p gina code-behind asociada.

## 4.3.- Clases de la Vista

### 4.3.1.- Arquitectura de la interfaz (Vista)

En el siguiente diagrama se indican los diferentes apartados de la interfaz general de usuario:

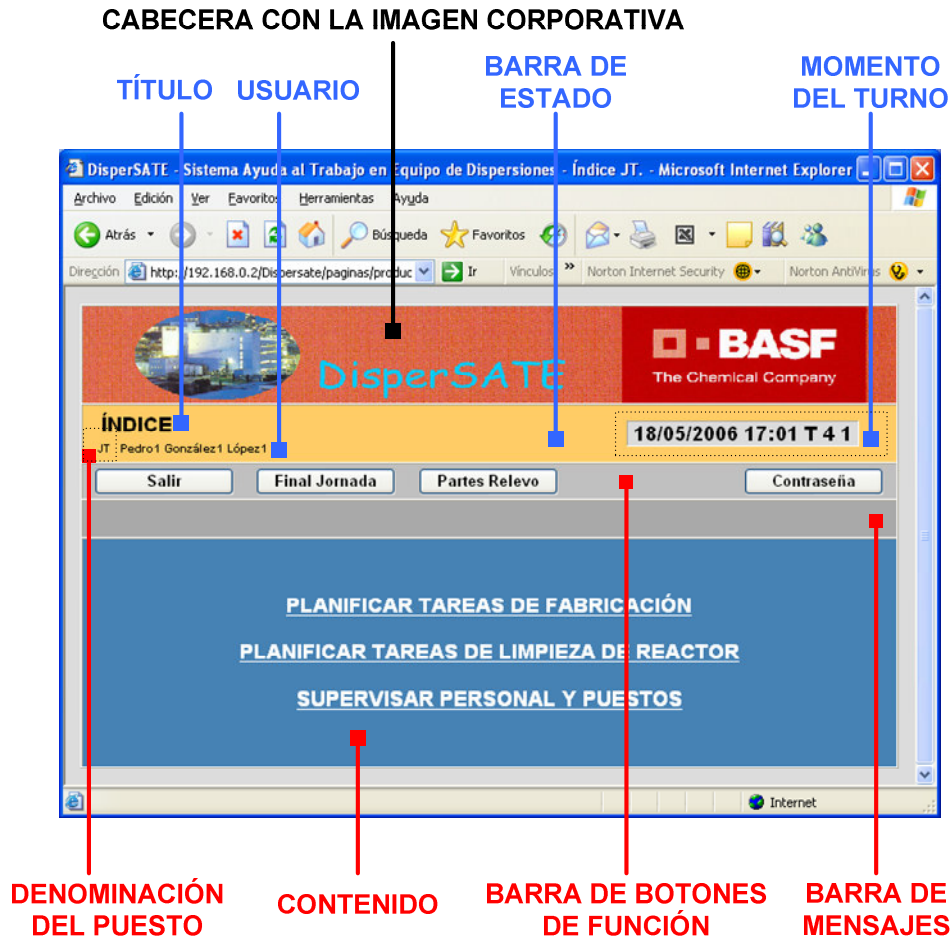


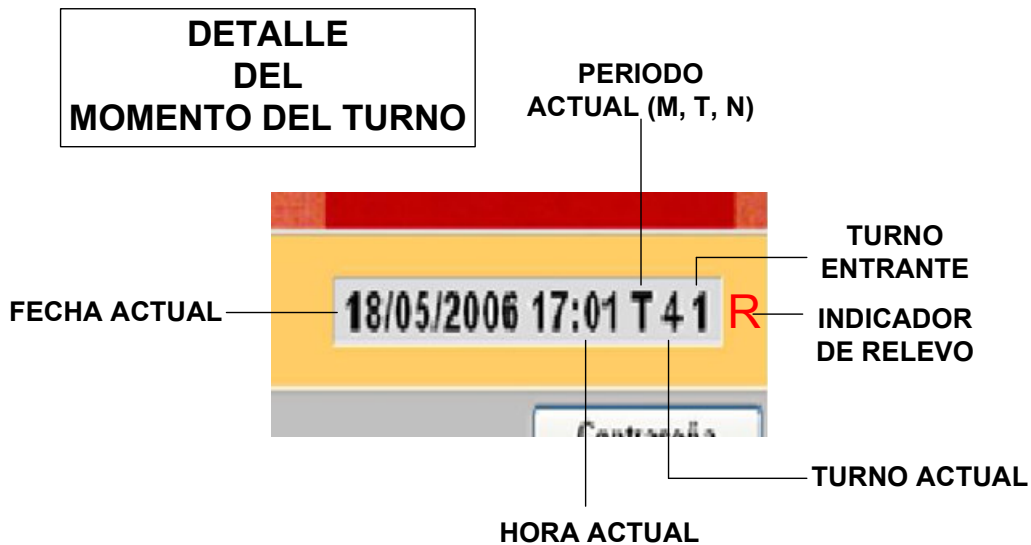
Figura 5 – Estructura general de la interfaz de usuario

**Cabecera corporativa.** - Elemento decorativo que identifica la aplicación con la empresa.

**Barra de estado.** - Información que indica el contexto de usuario. Sus elementos son:

- **Título:** título de la página en curso.
- **Denominación del puesto:** puesto al que corresponden las funcionalidades mientras no se pulse el botón “Salir” o el “Final jornada”.
- **Nombre del usuario:** último trabajador que accedió al puesto. En el caso del contraamaestre el usuario es siempre el mismo.

- **Momento del turno:** a continuación se muestra un esquema sobre este elemento.



**Figura 6 – Detalle del momento del turno en la interfaz de usuario**

La información que muestra este elemento se actualiza cada vez que se carga o se refresca la página. Dado que las funcionalidades incluidas en la aplicación suponen una navegación constante entre páginas, la frecuencia de actualización es suficiente para orientarse, con una simple mirada, sobre el momento de trabajo. La “R” es un indicador que aparece durante la última media hora de trabajo del turno actual e indica que ha comenzado el periodo de relevo.

**Barra de mensajes.**- Es un espacio destinado a ofrecer textos informativos al usuario (errores, por ejemplo) y a pedir confirmación de algunas operaciones, presentando en este último caso algunos botones sobre esta barra.

**Contenido.**- Espacio destinado a mostrar el grueso de información generada por la funcionalidad propia de cada página. Si la cantidad de información necesita más espacio para ser presentada, la ventana se amplía hacia abajo, pudiéndose utilizar la barra de desplazamiento vertical del navegador para visualizarla.

**Implementación.**- La aplicación utilizará páginas ASP.NET. Hay varias formas de conseguir el aspecto anterior con estas páginas:

- Generar las páginas tal como se muestran, duplicando todo el código común en cada página.
- Utilizar frames, lo que no es aconsejable porque puede haber problemas con los navegadores.
- En ASP.NET se puede crear una relación de herencia entre páginas y utilizar un método genérico definido en la clase base, que se implementará de forma diferente en cada

subclase. En primer lugar se crea la página ASP.NET que servirá de base: *BasePage.cs* (hereda de *System.Web.UI.Page*). En esta clase se incluirá un método virtual y realizaremos todos los métodos comunes para las diferentes clases. Se crea un fichero con la parte común del aspecto visual de las páginas: *BasePage.inc*. Finalmente se crean las diferentes páginas de la aplicación añadiendo a los ficheros .aspx la directiva #include en el lugar donde queremos que aparezca la cabecera:

```
<!--#include virtual="BasePage.inc"--!>
```

- Utilizar la directiva #include para incluir dentro de una página ASP.NET otras páginas ya creadas antes en el lugar donde se desea.

El diseño actual de la interfaz de usuario comenzó con la primera opción. La distribución de los elementos se consigue mediante una plantilla con la cabecera (una simple imagen de archivo estática) y diferentes tablas, que se copia en cada página para iniciar el diseño.

Visual Studio .NET permite diseñar páginas en dos modos diferentes:

- en el modo html tradicional, en el que los elementos se sitúan mediante etiquetas y marcado estándar;
- en modo rejilla, en el que se puede situar con el ratón cada elemento en la pantalla en la posición (x, y) deseada, de forma visual.

Se ha optado por el modo tradicional, ya que en algunos momentos el trabajo se ha desarrollado con un editor de textos fuera de Visual Studio, por necesidades de desplazamiento.

Las etiquetas, botones y demás elementos visibles comunes están incluidos en la plantilla. Solo hay que asignarles contenido y funcionalidades a sus eventos en la página code-behind de C#, que implementa la clase del controlador, y serán propios de cada página.

Las funcionalidades comunes de una cierta dificultad se implementan como métodos estáticos de clase, en la clase *UtilidadesDispersate*, utilizándolos en cada página mediante una llamada. Así se elimina el código redundante, aunque no existen los beneficios de la herencia desde una página base.

Esta forma de trabajar proviene de un primer momento en el que se desconocía .NET, C# y Visual Studio .NET, pero era necesario comenzar a diseñar las páginas. Más tarde, con los conocimientos ya adquiridos, se trazó el objetivo de transformar el producto para adaptarlo a la herencia. Y así se hizo, pero sin concluir el proceso, de forma que en este momento las páginas, con una estructura común bien definida, están preparadas para transformarlas y relacionarlas mediante herencia. Pero esto será objeto de un trabajo posterior, una vez entregado este proyecto.

### 4.3.2.- Relación de páginas

<b>RELACIÓN DE PÁGINAS DE DisperSATE</b>		
<b>MODULO</b>	<b>TÍTULO DE LA PÁGINA</b>	<b>FICHERO .ASPX</b>
<b>Conexión y Acceso</b>	<b>Planta de Dispersiones Sistema de Ayuda al Trabajo en Equipo</b>	<b>portada.aspx</b>
	Créditos	acercadeDispersate.aspx
	<b>Control de acceso</b>	<b>control_acceso.aspx</b>
	Firmar relevo	firmar_relevo.aspx
	Cambiar contraseña	cambiar_contra.aspx
<b>Administración</b>	<b>Índice CM</b>	<b>indice_CM.aspx</b>
	Gestión producción	gestion_produccion.aspx
	Nueva carga	nueva_carga.aspx
	Modificar o borrar carga n.º	modificar_borrar_carga.aspx
	Gestión de normas	gestion_normas.aspx
	Nueva norma	nueva_norma.aspx
	Modificar o borrar norma	modificar_borrar_norma.aspx
	Gestión de personal	gestion_personal.aspx
	Alta trabajador	alta_trabajador.aspx
Modificación o baja del trabajador	modificacion_baja_trabajador.aspx	
<b>Producción</b>	<b>Índice JT</b>	<b>indice_JT.aspx</b>
	Planificar tareas de fabricación	fabricacion1.aspx
	Añadir tarea de fabricación	nueva_tarea_de_fabricacion1.aspx
	Configurar tarea de fabricación	nueva_tarea_de_fabricacion2.aspx
	Modificar o borrar tarea de fabricación	modificar_borrar_fabricacion.aspx
	Planificar tareas de limpieza de reactor	limpiezareactores.aspx
	Añadir nueva tarea de limpieza de reactor	nueva_limpieza.aspx
	Modificar o borrar tarea de limpieza	modificar_borrar_limpieza.aspx
	Supervisar personal y puestos	supervisar_personal_puestos.aspx
	<b>Índice AJT</b>	<b>indice_AJT.aspx</b>
	Planificar trasiegos	trasiegos.aspx
	Nuevo trasiego	nuevo_trasiego.aspx
	Modificar o borrar tarea de trasiego	modificar_borrar_trasiego.aspx
	Planificar desodorización	desodorizacion.aspx
	Nueva desodorización	nueva_desodorizacion.aspx
	Modificar o borrar desodorización	modificar_borrar_desodorizacion.aspx
	Planificar muestras	muestra.aspx
	Nueva muestra	nueva_muestra.aspx
	Modificar o borrar muestra	modificar_borrar_muestra.aspx
	<b>Índice PA[1,2,3,4]</b>	<b>indice_PA.aspx</b>
	Índice R-[307,520,620,419,720,130,820,920]	indice_reactor.aspx
	Tarea (num. tarea)	tarea_PA.aspx
	<b>Índice PR</b>	<b>indice_PR.aspx</b>
	No se han implementado las funcionalidades para los preparadores.	
	<b>Índice PA[1,2,3,4,5]</b>	<b>indice_OP.aspx</b>
	Tarea (num. tarea)	tarea_OP.aspx

**Figura 7 – Relación de páginas de DisperSATE**



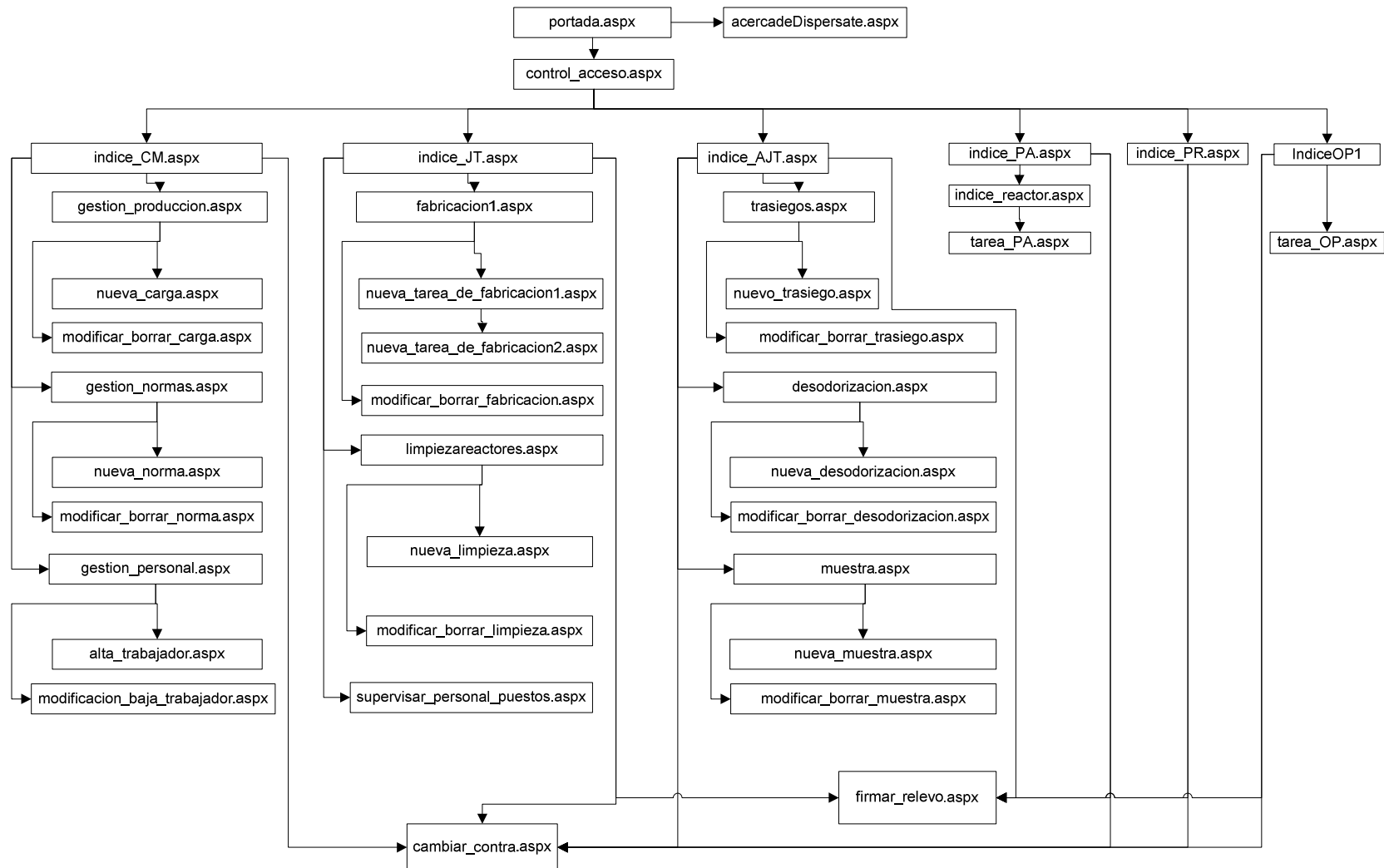


Figura 8 – Flujo de páginas de la interfaz

## 5.- DISEÑO DE LA BASE DE DATOS

La base de datos contendrá la persistencia del modelo. Por tanto su diseño se realizará a partir de la observación de las clases del modelo, lo que permite deducir las entidades existentes y sus relaciones.

### 5.1.- Diagrama ER

Existen varias formas de representar este tipo de diagramas. En este caso he utilizado la simbología vista en los materiales didácticos de la asignatura *Bases de datos I*, ya que resulta bastante clara.

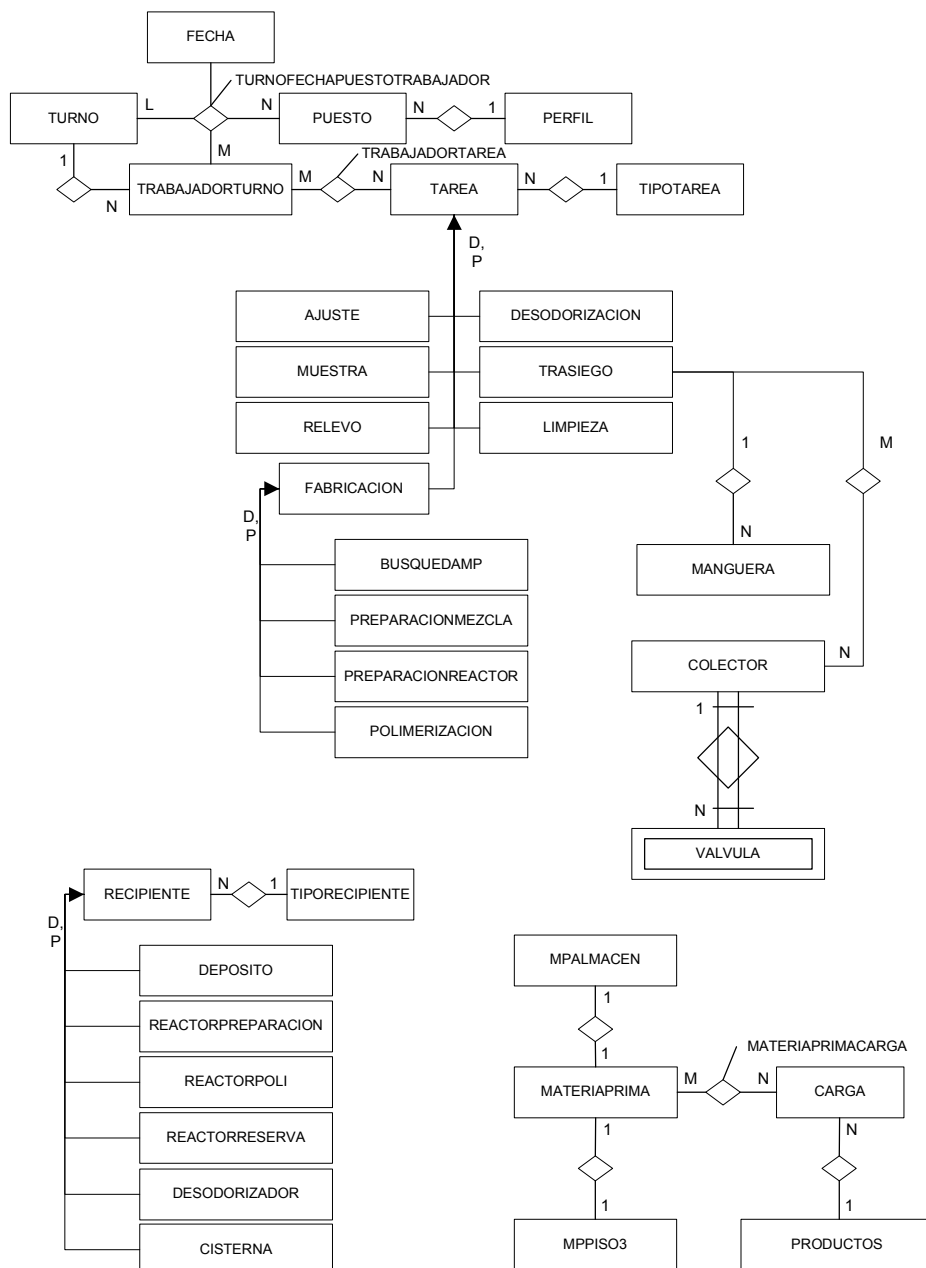


Figura 9 – Diagrama de Entidad-Relación

## 5.2.- Diagrama físico

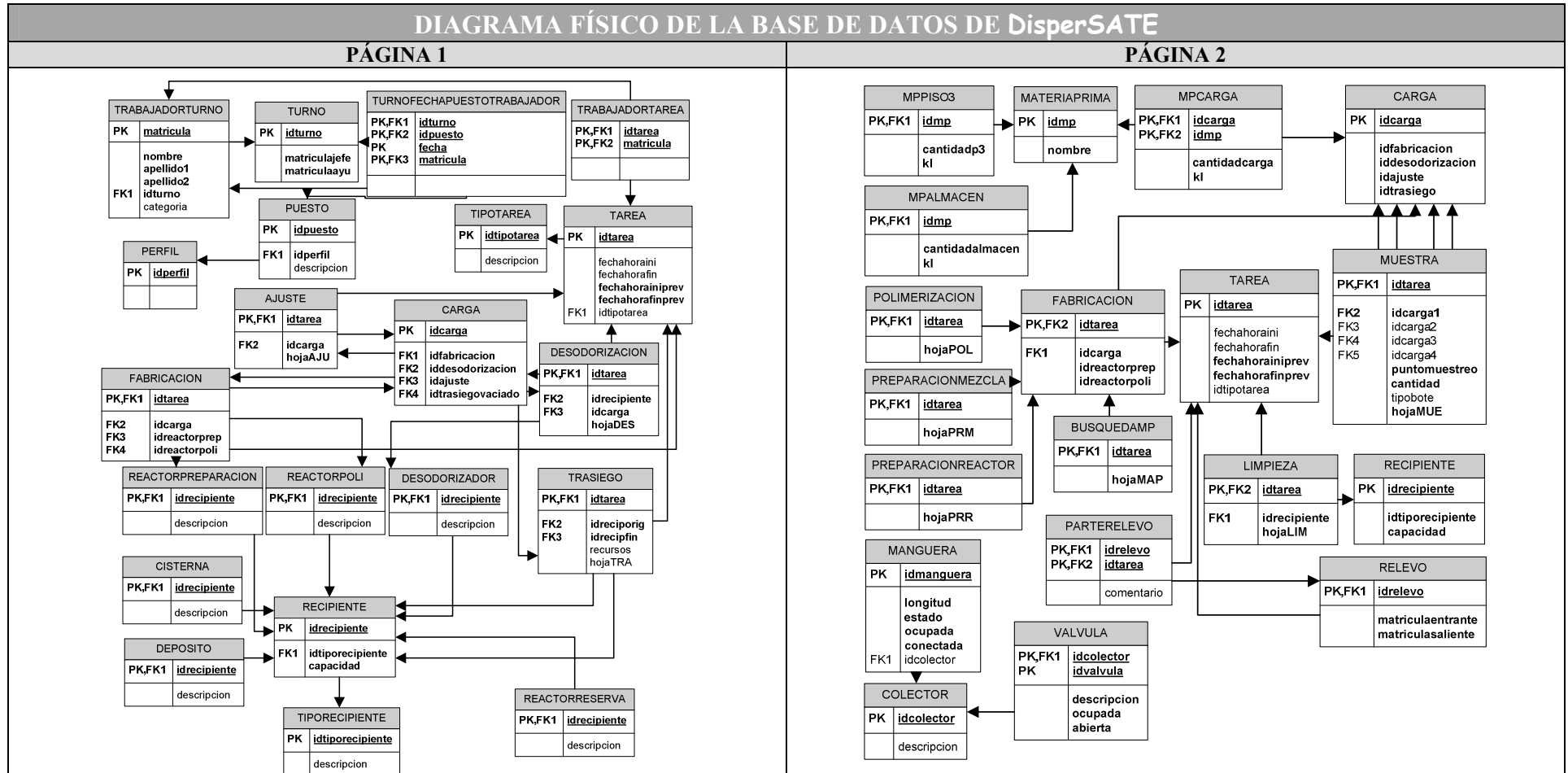


Figura 10 – Diagrama físico de la base de datos

## 6.- DESCRIPCIÓN DE DisperSATE

A continuación se va comentar con más detalle el funcionamiento de DisperSATE, a través de algunas páginas de su interfaz gráfica. En primer lugar se accede a la portada mediante un enlace directo situado, por ejemplo, en el escritorio del Terminal.



Figura 11 – Portada de DisperSATE

### 6.1.- Módulo de identificación y acceso

Desde el enlace **ENTRAR** de la portada se entra en la página de control de acceso. Aquí se debe seleccionar un puesto de la lista y a continuación escribir la contraseña de su titular.



Figura 12 – Control de acceso

En la figura se observa que se ha utilizado el puesto **CM**, cuyo usuario es el administrador y no está sujeto al sistema de relevos. Para cualquier otro usuario, se debe tener en cuenta que sólo puede acceder a un puesto su titular, es decir, aquel usuario que haya relevado por última vez el puesto. La página de relevo se implementa en el módulo de identificación, aunque se accede a ella desde el índice de cada puesto, menos del **CM**, con el botón **FINAL JORNADA**.



Figura 13 – Relevo de un puesto

El puesto de CM es único, y su horario de trabajo es normal con jornada partida, de lunes a viernes. En cambio, para entender el papel del resto de los actores, hay que tener en cuenta que para éstos existe un sistema rotativo de 5 turnos. Cada 8 horas entra en juego un turno diferente, con lo que es necesario un nuevo acceso al sistema para cada puesto al menos cada 8 horas, con la correspondiente identificación.

Cada usuario, tras acceder a su puesto, puede cambiar su contraseña personal si lo desea. Al igual que en el caso anterior, la página de cambio de contraseña pertenece al módulo de control de acceso, aunque se accede a ella desde el índice de cada puesto, incluido el del **CM**, con el botón **CONTRASEÑA**.



Figura 14 – Cambio de contraseña personal

## 6.2.- Módulo de administración

A este módulo pertenecen las funciones del puesto del CM. En el índice del CM se encuentran los enlaces a las páginas dónde se realizan dichas funciones.



**Figura 15 – Índice del CM**

El CM Proporciona las cargas de producto que se deben ir fabricando, en su orden correspondiente, con el que deberán aparecer en el apartado de planificación del JT (jefe de turno), para que éste pueda seleccionarlas e incluirlas en su plan de trabajo diario.

Proporciona también reglas adicionales (sobre la limpieza de recipientes, sobre el trasiego de productos, etc.) que son interpretadas por el JT y por el AJT (ayudante del jefe de turno) a la hora de planificar las tareas. No son órdenes estrictas, sino pautas que deberán plasmarse posteriormente en la planificación.

Modifica el estatus del personal, pudiendo cambiar a una persona de turno, o establecer ciertas restricciones u órdenes extraordinarias.

El funcionamiento es similar para los tres apartados disponibles: al entrar se presenta una página con el listado de elementos dados de alta en el momento actual.

Se puede crear un elemento nuevo utilizando el botón disponible a tal efecto, que da acceso a la página de creación de dicho elemento. Tras aceptar los datos, se vuelve a la página anterior y se puede ver el nuevo elemento en el listado.

Se puede modificar o borrar un elemento existente en la lista pulsando el enlace existente en su identificador. Tras realizar la acción, se puede ver que el elemento se ha modificado o se ha eliminado de la lista.

Este esquema de funcionamiento, útil y práctico, se repite en otros apartados de la aplicación. Vamos a ilustrar su uso con **GESTIÓN DE PRODUCCIÓN**:

## Ejemplo de creación de una nueva carga por el CM

Al pulsar el enlace **GESTIÓN DE PRODUCCIÓN** vamos a la siguiente página:



Añadimos una nueva carga:



Al guardar, volvemos a la página anterior, dónde aparece la nueva carga:



El ejemplo de funcionamiento es válido para los demás apartados del CM.

Figura 16 - Ejemplo de creación de una nueva carga por el CM

## Ejemplo de modificación de una carga por el CM

Al pulsar el enlace **GESTIÓN DE PRODUCCIÓN** vamos a la siguiente página:



Pulsamos el enlace de la primera línea:



Al guardar, volvemos a la página anterior, dónde aparece la carga modificada:



El ejemplo de funcionamiento es válido para los demás apartados del CM.

Figura 17 - Ejemplo de modificación de una carga por el CM



## 6.3.- Módulo de producción

### 6.3.1.- Entorno del jefe de turno (JT)



Figura 18 – Índice del JT

El JT decide qué cargas del plan de producción va a comenzar a fabricar durante el periodo de su turno y qué reactores se van a limpiar. Además debe tener la información sobre la situación del personal de su turno y el estado de relevo de puestos.

En “**PLANIFICAR TAREAS DE FABRICACIÓN**” se accede a una página donde aparece la lista de cargas que ya están en proceso de producción, tanto si han sido añadidas por el actual JT o si ya estaban fabricándose y aún no han terminado de hacerlo.

En el número de carga de cada una existe un enlace con el que se accede a una página de modificación y borrado (solo es posible borrar si aún no se ha iniciado la subtarea asociada de preparación de mezcla de monómeros). Borrar consiste en devolver la carga a la lista del plan de producción, para comenzar a fabricarla en otro momento.

Mediante el botón “**Nueva carga**” se accede a la lista de cargas del plan y se seleccionan y configuran las cargas necesarias, que irán desapareciendo de la lista del plan de producción. Al volver a la página de planificación del JT, aparecerán todas las cargas que se hayan ido configurando.

Al configurar una carga para fabricarla también se podrá determinar si después de vaciar el reactor se desea limpiarlo. De esta manera se prevé con antelación la limpieza y se la asocia a una carga de producto. Pero también se pueden planificar las limpiezas de forma independiente mediante “**PLANIFICAR TAREAS DE LIMPIEZA DE REACTOR**”.

La dinámica de este apartado es similar a otros ya explicados: aparece una página con la lista de limpiezas ya programadas; en cada fila existe un enlace sobre el identificador (número de limpieza) que permite modificar o borrar la limpieza. Cuando se elimina la limpieza deja de

aparecer en la lista. También se encontrará el botón “**Nueva limpieza**”, que permitirá configurar una nueva tarea de limpieza, que a su vez aparecerá en la lista.

Por último, el apartado “**SUPERVISAR PERSONAL Y PUESTOS**”, que necesita una descripción más profunda:



Figura 19 – Página de supervisión de personal y puestos del JT

En la lista de personal asignado al turno tenemos los números de matrícula remarcados en rojo: esto significa que en estos momentos los usuarios referidos no han realizado el relevo. Efectivamente: en la barra de estado del turno aparece el 5 como turno actual, pero actualmente los puestos están ocupados por el personal del turno 1, que es con el que se crea la base de datos. En su uso real la finalidad será informar, con un simple vistazo, de la ausencia de un trabajador, debiendo el JT tomar las medidas adecuadas.

Por otro lado, tenemos la lista de personal titular actual de los puestos. En este caso los puestos están remarcados en rojo porque supuestamente el puesto debería haberse relevado, ya que el turno actual no es el 1, sino otro, y no ha sido así. La casuística es:

- si estamos en el periodo normal del turno y el puesto está ocupado por personal del turno que corresponde, indicado en la barra de estado, la etiqueta del puesto no estará remarcada;
- si estamos en el periodo normal del turno y el puesto está ocupado ya por un integrante del turno siguiente, indicado en la barra de estado, significará que el usuario ha realizado el relevo antes de tiempo (ha venido temprano) y se iluminará la etiqueta en verde claro;
- si estamos dentro del periodo de relevo (la última media hora), indicado por la R en la barra de estado, o dentro del turno siguiente, u otro, y el puesto no se ha relevado, la etiqueta del puesto se ilumina en rojo, tal como ocurre en la pantalla de muestra anterior.

Se trata de un doble control de presencia, por personal previsto y por puestos. Teniendo en cuenta que las distancias en la planta no permiten comprobar físicamente la presencia de una persona con facilidad, que el puesto no tiene una sede fija y que la megafonía no siempre es efectiva ni se permite una confirmación verbal rápida, esta funcionalidad tiene una gran utilidad.

### 6.3.2.- Entorno del ayudante del jefe de turno (AJT)



Figura 20 – Índice del AJT

El AJT se encargará de planificar los trasiegos de producto acabado, de planificar los procesos de desodorización y de planificar las tareas de muestreo.

La operativa es similar a la descrita para otros apartados: en las páginas de planificación se encontrará una lista con las tareas programadas; existirá un enlace sobre el identificador de la tarea para poder acceder a la página de modificación y eliminación; también existirá un botón para crear y configurar una nueva tarea, que aparecerá en la lista correspondiente.

**6.3.3.- Entorno de panelista (PA)**

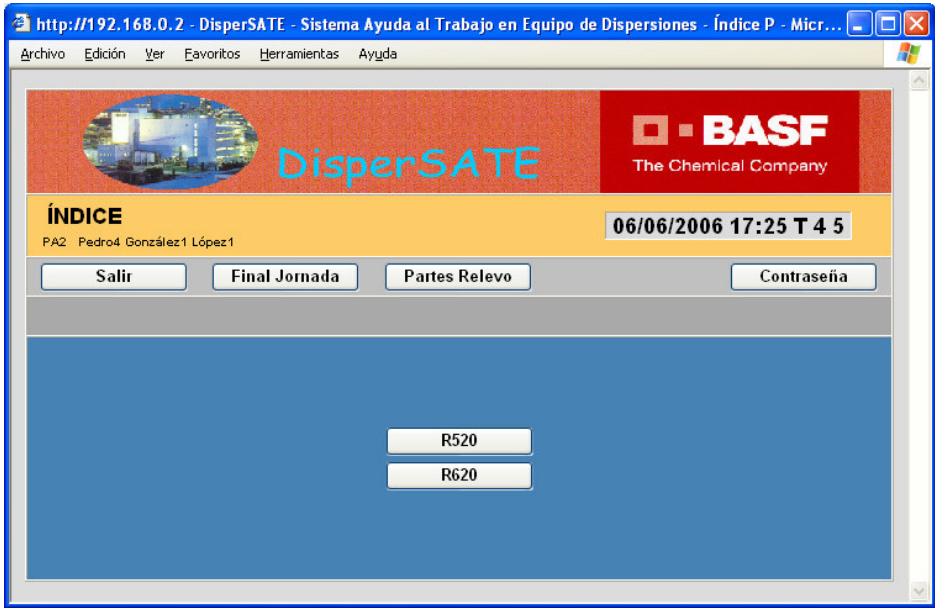


Figura 21 – Índice de PA

Para acceder a las tareas de uno de los reactores hay que pulsar el botón correspondiente. La página del reactor tiene el aspecto siguiente:

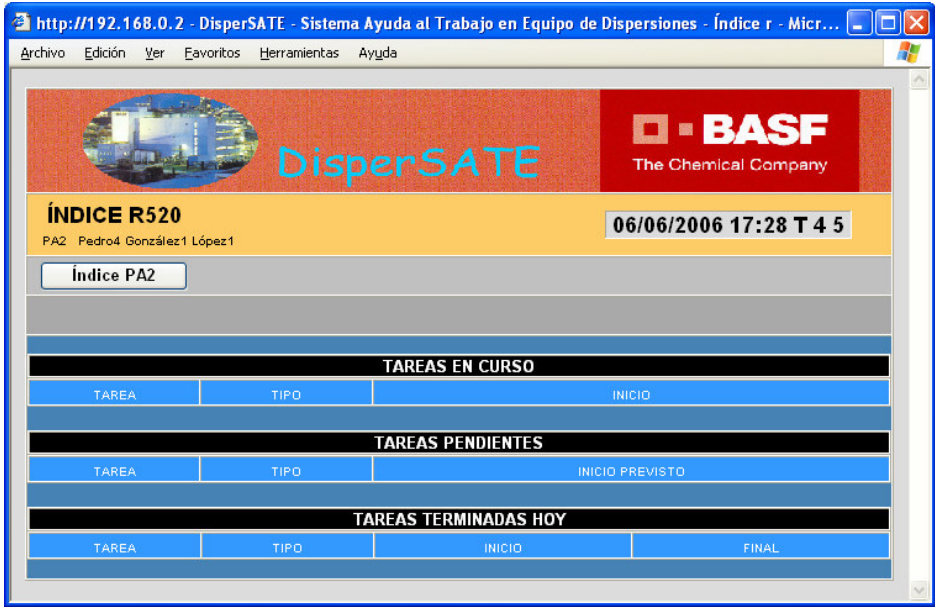


Figura 22 – Índice de reactor

Aparecen listadas las tareas asociadas a cada reactor, que son las subtareas que componen la fabricación de las cargas, o tareas de limpieza. Los identificadores de las tareas tendrán un enlace: una tarea en curso podrá terminarse y una tarea pendiente podrá iniciarse, cambiando de lista.

### 6.3.4.- Entorno de preparador de cargas (PR)

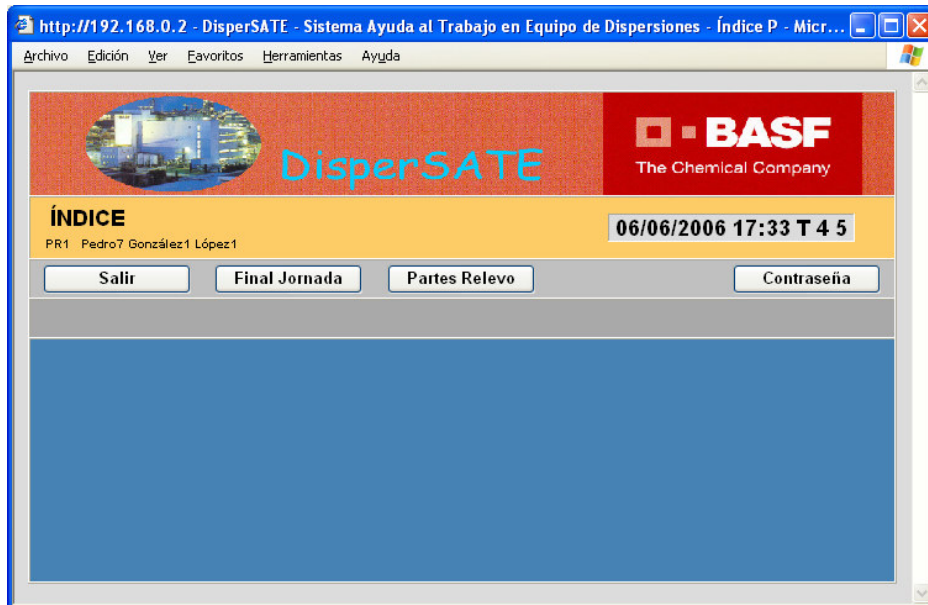


Figura 23 – Índice de PR

Actualmente los PR no tienen implementada ninguna funcionalidad específica debido a la falta de tiempo. Queda pendiente este apartado. Los botones de la barra funcionan igual que para el resto de puestos.

### 6.3.5.- Entorno de operario (OP)

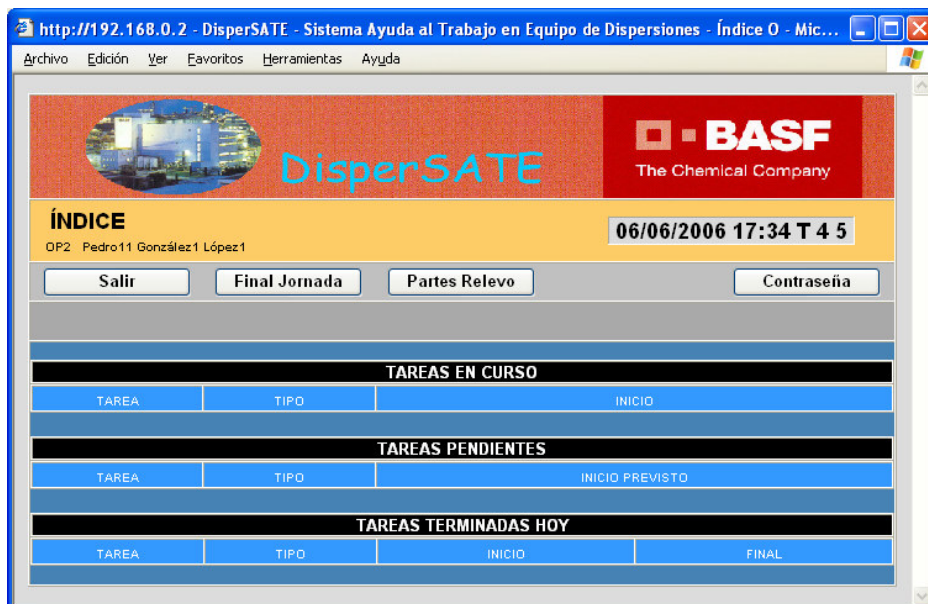


Figura 24 – Índice de OP

Aparecen listadas las tareas asociadas a cada OP. Los identificadores de las tareas tendrán un enlace: una tarea en curso podrá terminarse y una tarea pendiente podrá iniciarse, cambiando de lista.

## CONCLUSIONES

Hay que reconocer que la cantidad de funcionalidades previstas en un principio era excesivamente ambiciosa, dado el tiempo disponible y la falta de experiencia inicial en la tecnología .NET.

No obstante, desde mi punto de vista, las funcionalidades desarrolladas son suficientes para considerar que el producto final es digno y puede servir de base para desarrollar una aplicación más completa, en el ámbito de la empresa que ha servido de marco para el proyecto: la planta de dispersiones de BASF en Tarragona.

La aplicación está diseñada de forma correcta: separa con claridad la lógica de negocio de la interfaz de usuario, y agrupa de forma razonable los métodos que acceden a la base de datos. Sin embargo, podría mejorarse la conexión entre la lógica y la persistencia, añadiendo una serie de clases dedicadas exclusivamente a funcionar como puente entre ambas.

En lo que se refiere a la interfaz, no se ha alcanzado un nivel suficiente de uso de la herencia como camino para la reutilización, pero se han sentado las bases para introducirla, identificando y aislando los elementos comunes entre las páginas.

He avanzado mucho en lo que concierne al uso de la tecnología .NET y de sus elementos relacionados (C#, Visual Studio.NET, ASP.NET, ADO.NET, etc.), más teniendo en cuenta que partía de 0, pero me queda la sensación de haberme quedado corto en la profundización de algunos aspectos, quizás por la excesiva extensión de la aplicación, que obligaba a desarrollar mucho código en poco tiempo.

Finalmente tengo que añadir que considero bien aprovechado el tiempo empleado, desde el punto de vista de organización y dinámica de trabajo diario, aunque la dedicación me ha hecho rozar el límite de las necesidades humanas y familiares. Considero este TFC como la culminación de un ciclo de mi formación y espero que marque un punto de inflexión en mi vida.

## LINEAS FUTURAS DE TRABAJO

De lo anteriormente expuesto, se pueden deducir de forma inmediata varias acciones:

- **Mejorar** el acceso a la base de datos, añadiendo clases especializadas, introducir la herencia entre las páginas de la interfaz y profundizar en el uso de ASP.NET y de C#.
- **Implementar las funcionalidades previstas originalmente** que han sido descartadas para ajustar el desarrollo al tiempo disponible.
- **Subsistema de evaluación del rendimiento.** Se encargaría de recuperar la información sobre la producción realizada por cada turno y elaborar perfiles de rendimiento.
- **Subsistema de importación/exportación de datos.** Se encargaría de enlazar la base de datos con el sistema SAP/R3 de la fábrica, con el objeto de disponer de las existencias de materias primas en el almacén y de productos terminados en el parque de tanques.

## **BIBLIOGRAFÍA**

### **Libros**

[Frohock et al.] Frohock, M.; Reding, J.; Whalen, E.; Adrien, S.; (2001) “Guía completa de Microsoft SQL Server 2000”.- Editorial Mc. Graw Hill.

[González] González, O.; (2002) “ASP .NET”.- Editorial Anaya Multimedia.

[Charte] Charte, F.; (2002) “ASP .NET”.- Editorial Anaya Multimedia.

[Charte et al.] Charte, F.; Serrano, J.; (2002) “Programación con Visual Studio .NET”.- Editorial Anaya Multimedia.

[Ferguson et al.] Ferguson, J.; Patterson, B.; Beres, J.; (2003) “La Biblia de C#”.- Editorial Anaya Multimedia.

[Arora et al.] Arora, G.; Aiaswamy, B.; Pandey, N.; (2002) “Proyectos profesionales C#”.- Editorial Anaya Multimedia.

[Sceppa] Sceppa, D.; (2000) “Programación avanzada con ADO”.- Editorial Mc. Graw Hill.

### **Referencias a Internet**

**Microsoft .net**

<http://www.microsoft.com/latam/net/default.asp>

**Microsoft.com “MSDN Home”**

<http://msdn.microsoft.com/netframework/>

**Windows.com “MSDN Solution Architecture Center”**

<http://msdn.microsoft.com/architecture/>

**Visual Studio**

<http://www.microsoft.com/spanish/msdn/latam/vstudio/default.aspx>

**El rincón del programador – “.Net Framework”**

<http://www.elrincondelprogramador.com/default.asp?pag=secciones/net/net.asp>

**developer.com “Enterprise Web Services and Model-View-Controller”**  
<http://www.developer.com/design/article.php/3490056>

**Sanghavi, k. “Hello World: An Anatomy of an ASP .NET Project”**  
<http://www.asp101.com/articles/kaushal/aspnethelloworld/default.asp>

**Peterson, J. “Writing your first ASP.NET page”**  
<http://www.asp101.com/lessons/firstpage.asp>

**Microsoft Corp “GotDotNet: The Microsoft .Net Framework Community ”**  
<http://www.gotdotnet.com/>

## **Revistas**

**SÓLO PROGRAMADORES.- Coleccionable ASP . Net**  
(A partir del n. ° 116)

**SÓLO PROGRAMADORES.- Coleccionable Visual C# . Net Standard**  
(A partir del n. ° 104)