

# Trabajo Final de Carrera.

Inspecciones de Calidad basado en páginas web con  
tecnología J2EE.

Universitat Oberta de Catalunya.  
E.T.I.S.  
Curso 2004 - 2005

Alumno: Santi Valverde Montseny.  
[svalverde@uoc.edu](mailto:svalverde@uoc.edu)

Dirigido: Antoni Oller Arcas.

## Resumen.

Este trabajo pretende utilizar la arquitectura J2EE para desarrollar un sistema completo y real. Para ello ha sido necesario aplicar los conocimientos sobre Ingeniería de Software para conseguir elaborar el producto final.

Ello nos ha llevado a todo un proceso de elaboración en diversas fases en la cuales aplicando diversas técnicas, se han obtenido la recogida y documentación de requisitos, el análisis, diseño e implementación del sistema.

A lo largo de esta memoria se podrá observar la secuencia de pasos necesarios en la obtención de un producto software y como a partir de una fase obtenemos la información necesaria para construir la siguiente, mejorada y ampliada.

Mediante la utilización de software libre como MySQL, Tomcat, Java,... se ha conseguido toda la operativa necesaria para construir la aplicación, además tiene la posibilidad de integrarse sobre diferentes plataformas.

Por último apuntar que el producto final obtenido tiene muchas vías de ampliación futuras, lo que permitirá aglutinar diferentes sistemas autónomos actuales en uno solo, independiente de las plataformas.

---

## Índice de contenidos.

<b>Resumen.</b>	2
<b>Índice de contenidos.</b>	3
<b>Índice de Figuras.</b>	4
<b>Introducción.</b>	5
Contexto.	5
Objetivos del trabajo.	5
<b>Recogida y documentación de requisitos.</b>	6
Requisitos funcionales.	6
Requisitos de la interficie de usuario.	7
Especificaciones de usabilidad.	7
<b>Análisis.</b>	8
Casos de uso.	8
Especificación de las clases de análisis.	13
Análisis de la Interficie de Usuario.	19
<b>Diseño.</b>	21
Patrones de diseño.	21
Diseño de la implementación de los casos de uso.	21
Revisión diagrama estático de diseño.	24
Diseño de la persistencia.	27
Diseño de la interficie gráfica de usuario.	29
<b>Implementación.</b>	32
Descripción del producto obtenido.	32
Software utilizado.	32
<b>Conclusiones y recomendaciones.</b>	33
Nivel de consolidación de los objetivos.	33
Líneas futuras de trabajo.	33
<b>Bibliografía.</b>	34
Libros utilizados.	34
Referencias de Internet.	34
<b>Anexo A.</b>	35
Código SQL para la creación de las tablas.	35

## Indice de Figuras.

<i>Figura 1</i> Diagrama Casos de Uso.	8
<i>Figura 2</i> Diagrama clases asociaciones.	13
<i>Figura 3</i> Diagrama estático análisis, caso de uso 1.	14
<i>Figura 4</i> Diagrama estático análisis, caso de uso 2.	14
<i>Figura 5</i> Diagrama estático análisis, caso de uso 3.	14
<i>Figura 6</i> Diagrama estático análisis, caso de uso 4.	15
<i>Figura 7</i> Diagrama estático análisis, caso de uso 5.	15
<i>Figura 8</i> Diagrama estático análisis, caso de uso 6.	15
<i>Figura 9</i> Diagrama estático análisis, caso de uso 7.	16
<i>Figura 10</i> Diagrama estático análisis, caso de uso 8.	16
<i>Figura 11</i> Diagrama estático análisis, caso de uso 9.	16
<i>Figura 12</i> Diagrama estático análisis, caso de uso 10.	17
<i>Figura 13</i> Diagrama estático análisis, caso de uso 11.	17
<i>Figura 14</i> Diagrama estático análisis, caso de uso 12.	18
<i>Figura 15</i> Diagrama estático análisis, caso de uso 13.	18
<i>Figura 16</i> Diagrama estático análisis, caso de uso 14.	18
<i>Figura 17</i> Menu Principal.	19
<i>Figura 18</i> Menu Identificación.	19
<i>Figura 19</i> Menu Administrador.	19
<i>Figura 20</i> Menu Inspector.	20
<i>Figura 21</i> Menu Cliente.	20
<i>Figura 22</i> Menu Jefe Departamento.	20
<i>Figura 23</i> Model-View-Controller.	21
<i>Figura 24</i> Diagrama estático diseño: servlet.	24
<i>Figura 25</i> Diagrama estático diseño: Modelo.	24
<i>Figura 26</i> Diagrama estático diseño: usuario.	25
<i>Figura 27</i> Diagrama estático diseño: Lcd	25
<i>Figura 28</i> Diagrama estático diseño: Informe.	26
<i>Figura 29</i> Diagrama estático diseño: complementos.	26
<i>Figura 30</i> Diagrama Base datos.	27
<i>Figura 31</i> Interficie gráfica: alta modelo.	29
<i>Figura 32</i> Interficie gráfica: alta categoria.	30
<i>Figura 33</i> Interficie gráfica: lista test.	30
<i>Figura 34</i> Interficie gráfica: Informe diario.	31

## **Introducción.**

### ***Contexto.***

Nuestro cliente es una empresa dedicada a la fabricación de televisores de cristal líquido. Dentro de su proceso productivo hay un departamento encargado de realizar el control de la calidad del producto en fabricación. Dicho control consiste, en verificar una serie de pruebas y tomar ciertas medidas de una muestra de la producción diaria que permitirá la aceptación o rechazo del producto fabricado.

Actualmente los datos son obtenidos por unos operarios, a partir de ahora inspectores, que van rellenando formularios de papel a medida que van realizando las pruebas. Una vez completados estos formularios son introducidos en el sistema informático para su posterior evaluación y análisis.

### ***Objetivos del trabajo.***

El objetivo principal del trabajo es la eliminación de los formularios de papel y la introducción de los datos en el sistema informático a medida que se van generando. Para conseguir estos objetivos se creará una aplicación basada en páginas web que ofrecerá los siguientes servicios:

- Los inspectores podrán introducir el resultado de las pruebas en el momento de generarlas.
- El jefe del departamento podrá consultar y crear informes en tiempo real.
- El administrador del sistema gestionará las autorizaciones de los usuarios y la creación de las pruebas a realizar.
- Los usuarios autorizados podrán consultar los informes de calidad.

## Recogida y documentación de requisitos.

### ***Requisitos funcionales.***

El sistema que vamos a implementar esta compuesto por diversos actores: administrador del sistema, inspector, jefe departamento y cliente final. Las actividades que van a desarrollar cada actor son las siguientes:

Administrador Sistema. Es la persona que pondrá en marcha todo el sistema. Primero realizará el alta de los diversos usuarios y sus perfiles asignados, para ello introducirá los datos personales, asignara un identificador de usuario y password, también definirá el perfil asociado a cada usuario.

Una segunda labor es la creación y mantenimiento de los modelos en fabricación, ello comporta definir los datos característicos de cada modelo así como las pruebas a realizar en cada uno.

Inspector. Es la persona que introducirá el resultado de las pruebas realizadas a cada televisor, para ello es necesario seleccionar un modelo determinado, elegir un grupo de pruebas y una vez obtenido el formulario ir introduciendo el resultado de las pruebas.

En el caso de detectar algún defecto o prueba no superada el inspector decidirá el tipo y clasificación del defecto, así como si lo cree necesario, añadirá los comentarios pertinentes.

Jefe Departamento. Su función principal es la de generar los informes de calidad para su posterior consulta. En una primera fase dichos informes corresponderán a los datos obtenidos diariamente. Estos informes se obtendrán de los datos almacenados en una BBDD y mostrarán la cantidad de televisores inspeccionados, cantidad de pruebas realizadas, pruebas superadas y pruebas no superadas, estos datos se completarán con los comentarios adecuados y se almacenarán para su posterior consulta por parte de los clientes autorizados.

En el futuro, a partir de los datos de la BBDD, se podrán realizar diversos tipos de informes y cálculos estadísticos.

Cliente. Su función principal es la consulta de los informes creados por el jefe del departamento de calidad, para ello es necesario introducir la fecha del informe requerido y si este ha sido creado se le mostrarán los resultados.

Como característica general del sistema remarcar que cualquier tipo de actor necesita un nombre de usuario y password para entrar al sistema y solo accederá a la funcionalidad del perfil que tiene asignado.

### ***Requisitos de la interficie de usuario.***

Podemos realizar una división entre los usuarios que interactuan de manera ocasional y los que necesitan interactuar de manera continuada durante toda la jornada laboral.

Los que actuán de forma ocasional son los clientes, jefe departamento y administrador del sistema, este último tendrá un inicio intensivo para crear usuarios y modelos pero su intensidad descenderá una vez puesto en marcha el sistema.

Todos estos usuarios trabajan sobre PCs de sobremesa, por lo que el diseño de la interficie no necesita grandes restricciones, se busca que las opciones a realizar tengan una clara alusión a la función que se va a ejecutar, de manera que aunque los usuarios no tengan grandes conocimientos de informática su interacción sea cómoda y clara.

Por el contrario tenemos al grupo de inspectores, para los cuales, debido a su trabajo a pie de fábrica y movilidad se ha pensado la utilización de un Pocket PC, concretamente se piensa adquirir el iPAQ H4150 de Hewlett Packard. Dicho dispositivo consta de pantalla táctil TFT de 3,5 pulgadas y puntero por lo que las páginas web de interacción deben de suministrar el máximo de visibilidad y claridad en sus funciones. Por otro lado debido a su utilización diaria los inspectores alcanzarán un grado de utilización óptimo en poco tiempo.

### ***Especificaciones de usabilidad.***

Los clientes deben de interactuar inmediatamente despues de una explicación de la funcionalidad de su perfil.

Los inspectores, al cabo de una semana de utilización deben de poder realizar el mismo número de muestras que rellenando los formularios actuales de papel.

En el caso del admistrador el alta de usuarios debe de realizarse de forma inmediata así como la introducción de nuevos modelos no debe de superar los 15 minutos.

## Análisis.

### Casos de uso.

El diagrama básico de los casos de uso muestra la gestión que realizará cada actor sin entrar en detalle en las funciones de cada uno.

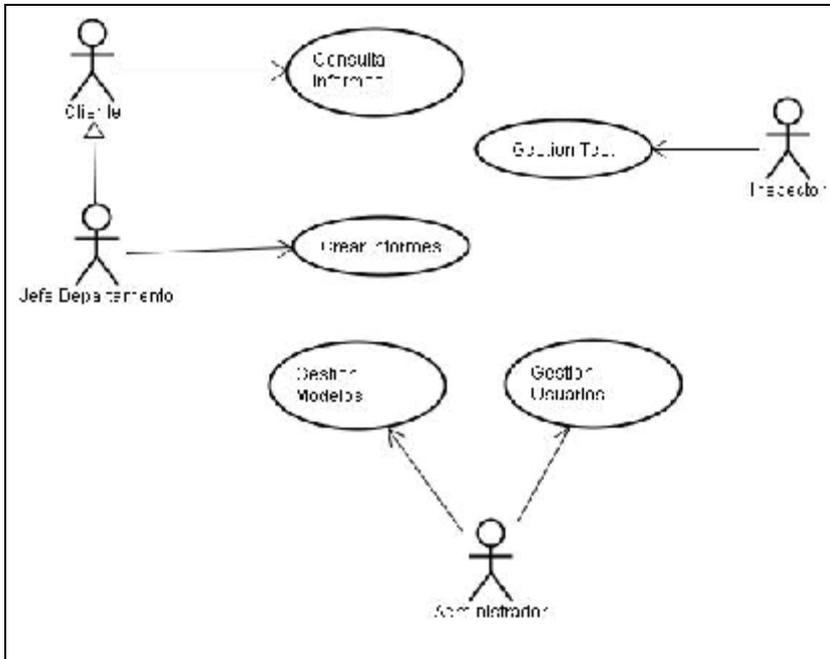


Figura 1 Diagrama Casos de Uso.

Ahora describiremos de manera textual las acciones a realizar por parte de los actores que componen el sistema.

#### Caso de Uso nº1: "Alta Usuario"

- **Resumen Funcionalidad:** Se añade un usuario nuevo a la BBDD.
- **Papel dentro del Trabajo:** Es una de las funciones básicas del administrador, una vez dados de alta los usuarios en la puesta en marcha del sistema, su frecuencia es baja.
- **Actores:** Administrador
- **Casos de Uso Relacionados:** Modificar Usuario.
- **Precondición:** El usuario no existe en la BBDD
- **Postcondición:** El nuevo usuario es incorporado a la BBDD
- El administrador introduce los datos del nuevo usuario, estos datos contienen el nº de empleado, el nick asignado, password, nombre, apellido y perfil de usuario. El perfil de usuario se selecciona de una lista desplegable donde se enumeran los diversos perfiles.
- **Excepciones:** Si el nº de empleado ya existe se lanzara un mensaje de error informado de la situación y se recupera la página principal del administrador.

#### Caso de Uso nº2: " Baja Usuario"

- Resumen Funcionalidad: Se elimina un usuario de la BBDD.
- Papel dentro del Trabajo: Su uso es ocasional.
- Actores: Administrador
- Casos de Uso Relacionados:
- Precondición: El usuario debe existir en la BBDD
- Postcondición: Los datos del usuario son borrados de la BBDD.
- Cuando el administrador selecciona este caso de uso se le abre una página para que pueda introducir el nº de empleado, una vez aceptada la acción, los datos son eliminados de la BBDD.
- Excepciones: Si el nº de empleado no existe se lanzará un mensaje informando de la situación y se volverá a la página principal del administrador.

#### Caso de Uso nº3: " Modificar Usuario"

- Resumen Funcionalidad: Se modifican los datos asociados a un usuario almacenándose los nuevos valores.
- Papel dentro del Trabajo: Su uso es ocasional.
- Actores: Administrador.
- Casos de Uso Relacionados:
- Precondición: El usuario debe existir en la BBDD.
- Postcondición: Los nuevos valores pertenecientes al usuario son almacenados en la BBDD.
- El administrador introduce el nº de empleado, aparece una página con los datos del usuario para su modificación, al validar los nuevos datos se almacenan en la BBDD.
- Excepciones: Si el nº de empleado no existe se recibe un mensaje de error y se vuelve a la página principal del administrador

#### Caso de Uso nº4: " Alta Modelo"

- Resumen Funcionalidad: Crea un modelo nuevo que se almacena en la BBDD
- Papel dentro del Trabajo: Es una de las tareas principales del administrador, su frecuencia es baja unos 4 modelos mensuales.
- Actores: Administrador
- Casos de Uso Relacionados: Baja Modelo, Alta Categoría, Alta Test
- Precondición: El nombre del modelo no existe en la BBDD
- Postcondición: El modelo nuevo se incorpora a la BBDD.
- Para introducir un modelo nuevo es necesario suministrar el nombre del modelo, su descripción y de una lista de check boxes marcar los test de los que va a constar el modelo.
- Señalar que para poder dar de alta un modelo ha sido necesario previamente el alta de categorías y tests.
- Excepciones: En los casos de nombre de modelo no válido o de la existencia del modelo se informará al usuario y se pasará a la página principal del administrador.

#### Caso de Uso nº5: " Baja Modelo"

- Resumen Funcionalidad: Consiste en inhabilitar un modelo para su inspección.
- Papel dentro del Trabajo: Ocasional, un modelo acostumbra a tener una vida de dos o tres años.
- Actores: Administrador
- Casos de Uso Relacionados:
- Precondición: El nombre del modelo existe en la BBDD
- Postcondición: El modelo queda inhabilitado para inspecciones.
- El administrador debe seleccionar el nombre del modelo desde un menu desplegable para inhabilitar, una vez validada la acción el modelo queda inutilizado para las inspecciones.
- Excepciones:

#### Caso de Uso nº6: " Alta Categoría "

- Resumen Funcionalidad: Se incorpora la definición de una nueva categoría a la BBDD
- Papel dentro del Trabajo: En la puesta en marcha del sistema será imprescindible la definición de todas las categorías, posteriormente se crean categorías de forma muy esporádica.
- Actores: Administrador
- Casos de Uso Relacionados: Alta Modelo, Alta Test
- Precondición: La categoría no existe en la BBDD
- Postcondición: La nueva categoría se incorpora a la BBDD
- El administrador debe suministrar el nombre de la categoría y su descripción también cuenta con una tabla donde aparecen las categorías ya definidas actualmente.
- Excepciones: Si la categoría ya existe o el nombre no es válido se informa al usuario de la situación y se vuelve a la página principal del administrador.

#### Caso de Uso nº7: " Alta Test"

- Resumen Funcionalidad: Se incorpora un nuevo test a la BBDD
- Papel dentro del Trabajo: En la puesta en marcha será preciso añadir todos los test que se realizan actualmente, posteriormente se añadirán de forma ocasional.
- Actores: Administrador
- Casos de Uso Relacionados: Alta Categoría
- Precondición: El test no existe en la BBDD y la categoría a la que pertenece ya ha sido dada de alta.
- Postcondición: El nuevo test ha sido incorporado a la BBDD
- El administrador debe suministrar el nombre del test, su descripción y la categoría a la cual pertenece. La categoría se seleccionará desde un menu desplegable donde aparecen todas las categorías definidas.
- Excepciones: Si el nombre del test existe o no es válido se informará al usuario de dicha situación y se volverá a la página principal del administrador.

#### Caso de Uso nº8: " Informe Diario"

- Resumen Funcionalidad: Permite la visualización del informe diario de calidad.
- Papel dentro del Trabajo: Es la función habitual por parte de los clientes
- Actores: Cliente, Jefe Departamento.
- Casos de Uso Relacionados: Informe Detallado
- Precondición: El informe debe haber sido creado con anterioridad.
- Postcondición: Se visualizan los datos en pantalla.
- Los clientes autorizados pueden ver los informes diarios creados por el jefe del departamento de calidad, en ellos se reflejan la marcha de la calidad de producción. El jefe del departamento puede visualizar los informes creados por él. Para ello deben introducir la fecha que quieren visualizar.
- Excepciones: Si el informe no existe se advierte al usuario de esta situación y se vuelve a la página principal del cliente.

#### Caso de Uso nº9: " Informe Detallado"

- Resumen Funcionalidad: Permite la visualización de los defectos encontrados por tipo de test y número de serie.
- Papel dentro del Trabajo: Es una de las funciones habituales por parte de los clientes.
- Actores: cliente, jefe departamento
- Casos de Uso Relacionados: Informe Diario
- Precondición: El informe detallado ha sido creado con anterioridad
- Postcondición: Se visualizan los datos en pantalla
- Los clientes autorizados, introduciendo previamente la fecha que quieren visualizar, pueden ver en pantalla los datos inspeccionados y los resultados obtenidos por televiso.
- Excepciones: Si el informe no existe se advierte al usuario de esta situación y se vuelve a la página principal del cliente.

#### Caso de Uso nº10: " Llenar Test"

- Resumen Funcionalidad: Consiste en rellenar los tests asignados a un modelo
- Papel dentro del Trabajo: Es el trabajo principal de los inspectores, su uso será continuado durante la jornada laboral.
- Actores: Inspector
- Casos de Uso Relacionados: Modificar test, Consulta Test
- Precondición: El modelo ha sido dado de alta en la BBDD
- Postcondición: Se incorporan los datos del televisor bajo inspección, así como los datos obtenidos de las diferentes pruebas pasadas.
- El inspector selecciona desde un menú desplegable un modelo a inspeccionar, introduce el nº de serie del televisor y posteriormente aparece una lista de categorías de entre las cuales debe elegir una y validar. Para cada categoría seleccionada se le desplegará una página donde aparecen los test asociados a cada categoría, donde podrá ir completando la información solicitada.
- Excepciones: Si el nº de serie ya existe, se informa al usuario de dicha situación y se retorna a la página principal del inspector.

#### Caso de Uso nº11: " Modificar Test"

- Resumen Funcionalidad: Permite modificar el resultado de algún test ya introducido o finalizar algún test pendiente.
- Papel dentro del Trabajo: Habitual dentro del trabajo básico del inspector, ya que las pruebas realizadas pueden ser compartidas por varios inspectores.
- Actores: Inspector
- Casos de Uso Relacionados: Llenar Test
- Precondición: El televisor a inspeccionar debe haber sido previamente introducido en la BBDD.
- Postcondición: Los datos asociados a algún test han sido modificados o se ha incorporado algún test nuevo.
- El inspector debe suministrar el modelo y nº de serie, recibirá la lista de categorías de donde podrá seleccionar una para modificar algún test asociado a la categoría.
- Excepciones: Si el nº de serie del modelo introducido no existe se informara al usuario de la situación y se retornará a la página principal del inspector.

#### Caso de Uso nº12: " Consulta Test"

- Resumen Funcionalidad: Informa de los televisores pendientes de finalizar alguna inspección.
- Papel dentro del Trabajo: Ocasional dentro del trabajo de los inspectores.
- Actores: Inspector
- Casos de Uso Relacionados: Llenar Test, Modificar Test
- Precondición:
- Postcondición: Visualiza en pantalla los televisores pendientes de finalizar la inspección.
- El inspector visualiza en la pantalla los números de serie de los televisores con algún test pendiente de realizar.
- Excepciones:

#### Caso de Uso nº13: " Crear Informe"

- Resumen Funcionalidad: Crea el informe de calidad diario a partir de los datos almacenados en la BBDD.
- Papel dentro del Trabajo: Es una de las tareas básicas dentro del trabajo del jefe del departamento
- Actores: Jefe Departamento.
- Casos de Uso Relacionados:
- Precondición: Deben existir datos sobre televisores inspeccionados almacenados en la BBDD
- Postcondición: Se crea un fichero con los datos obtenidos y los comentarios adjuntados por parte del jefe
- Cuando el jefe del departamento selecciona este caso de uso aparecen en pantalla los datos diarios sobre los cuales puede añadir los comentarios oportunos.
- Excepciones:

Caso de Uso nº14: " Crear Informe Detallado"

- Resumen Funcionalidad: Crea un informe detallado a partir de los datos almacenados en la BBDD.
- Papel dentro del Trabajo: Es una de las tareas básicas dentro del trabajo del jefe del departamento
- Actores: Jefe Departamento
- Casos de Uso Relacionados:
- Precondición: Deben existir datos sobre televisores inspeccionados almacenados en la BBDD
- Postcondición: Se crea un fichero con los datos obtenidos y los comentarios adjuntados por parte del jefe
- Cuando el jefe del departamento selecciona este caso de uso aparecen en pantalla los datos diarios sobre los cuales puede añadir los comentarios oportunos.
- Excepciones:

**Especificación de las clases de análisis.**

A partir de los casos de uso definidos anteriormente obtenemos las clases de entidades que van a participar en nuestro sistema así como las relaciones que van a existir entre ellas.

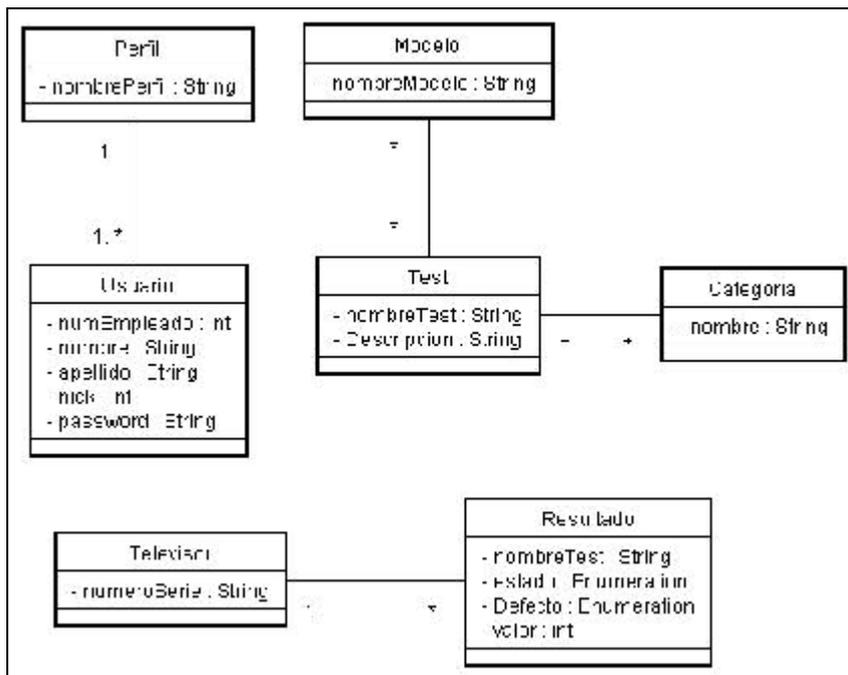


Figura 2 Diagrama clases asociaciones.

Para cada caso de uso crearemos un diagrama de colaboración simplificado en el cual podremos ver las clases de frontera, las clases de control y las operaciones que se aplican.

Caso de Uso nº1: "Alta Usuario"

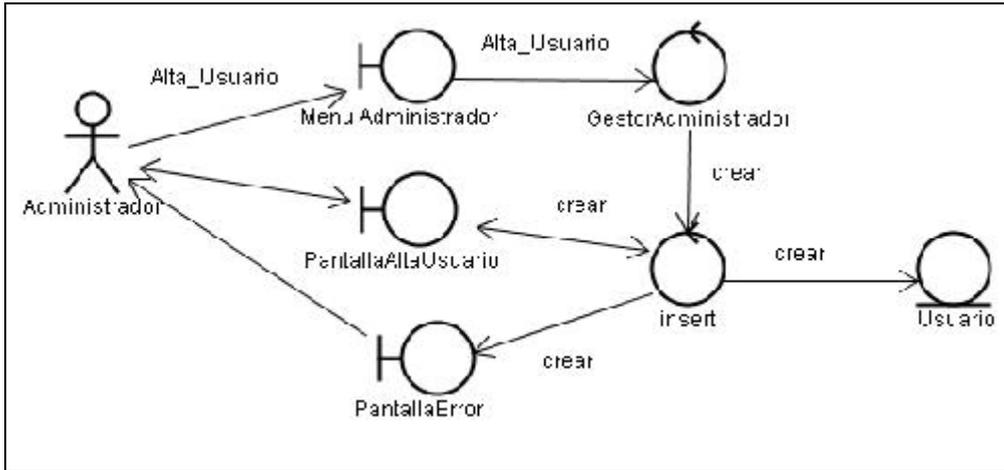


Figura 3 Diagrama estático análisis, caso de uso 1.

Caso de Uso nº2: "Baja Usuario"

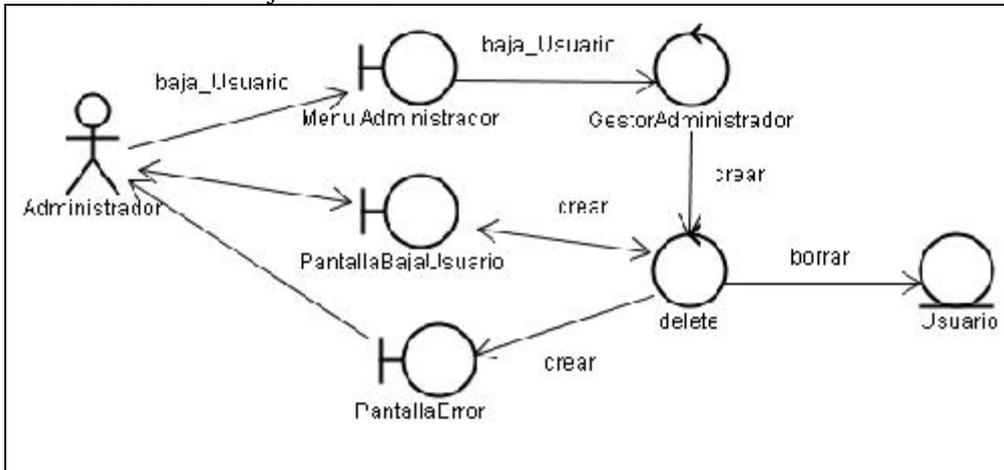


Figura 4 Diagrama estático análisis, caso de uso 2.

Caso de Uso nº3: "Modificar Usuario"

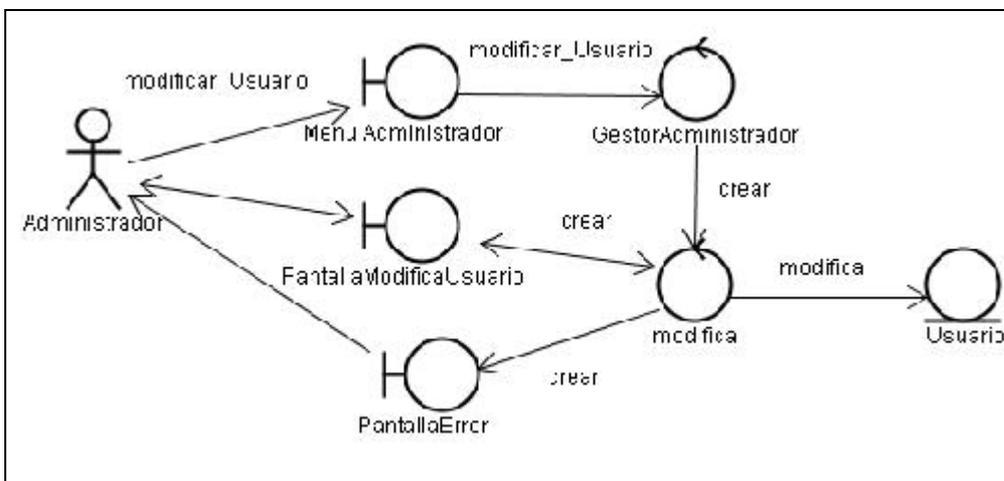


Figura 5 Diagrama estático análisis, caso de uso 3.

Caso de Uso nº4: "Alta Modelo"

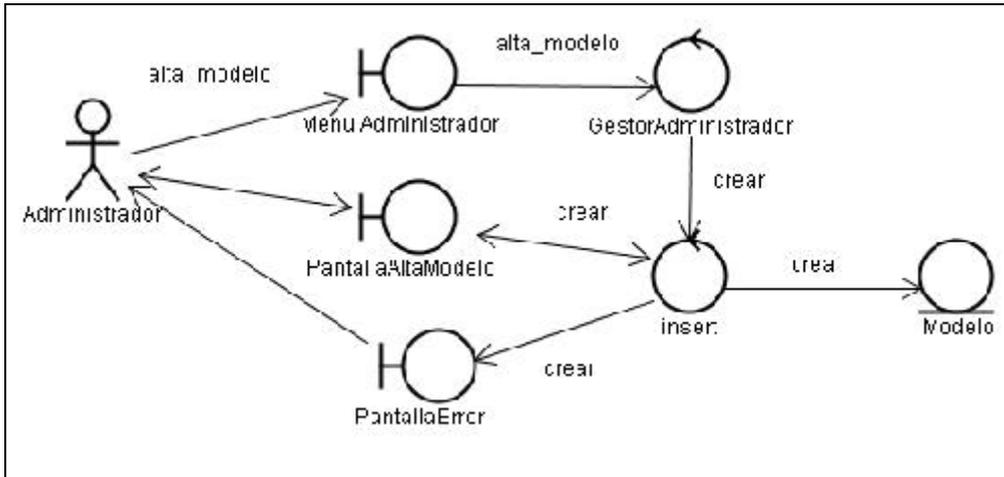


Figura 6 Diagrama estático análisis, caso de uso 4.

Caso de Uso nº5: "Baja Modelo"

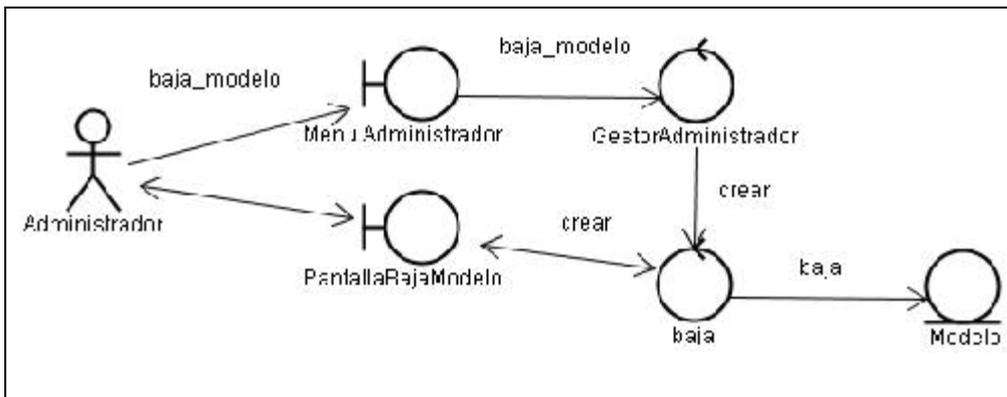


Figura 7 Diagrama estático análisis, caso de uso 5.

Caso de Uso nº6: "Alta Categoría"

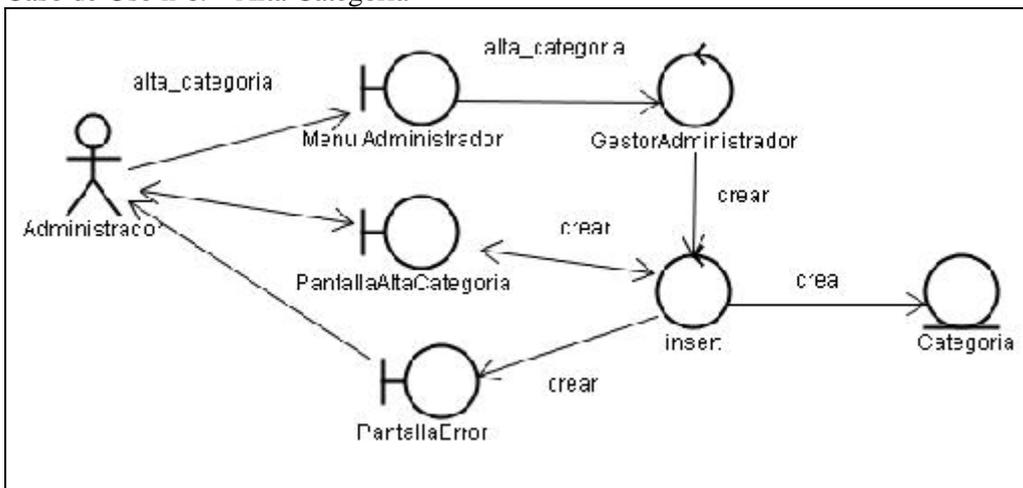


Figura 8 Diagrama estático análisis, caso de uso 6.

Caso de Uso nº7: " Alta Test "

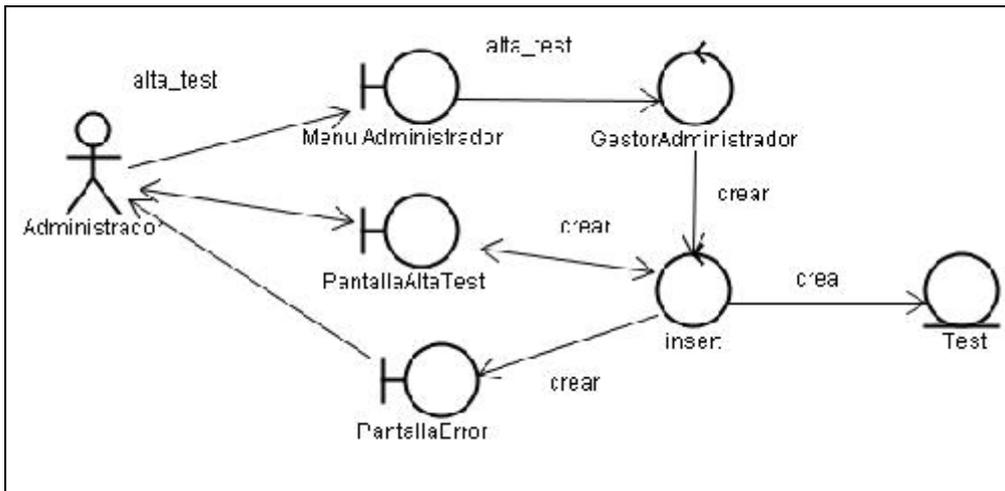


Figura 9 Diagrama estático análisis, caso de uso 7.

Caso de Uso nº8: " Informe Diario "

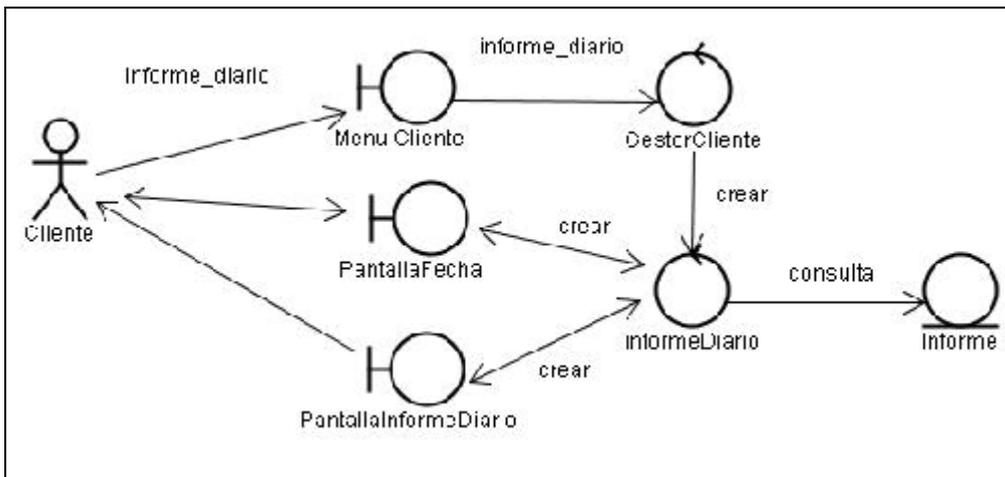


Figura 10 Diagrama estático análisis, caso de uso 8.

Caso de Uso nº9: " Informe Detallado "

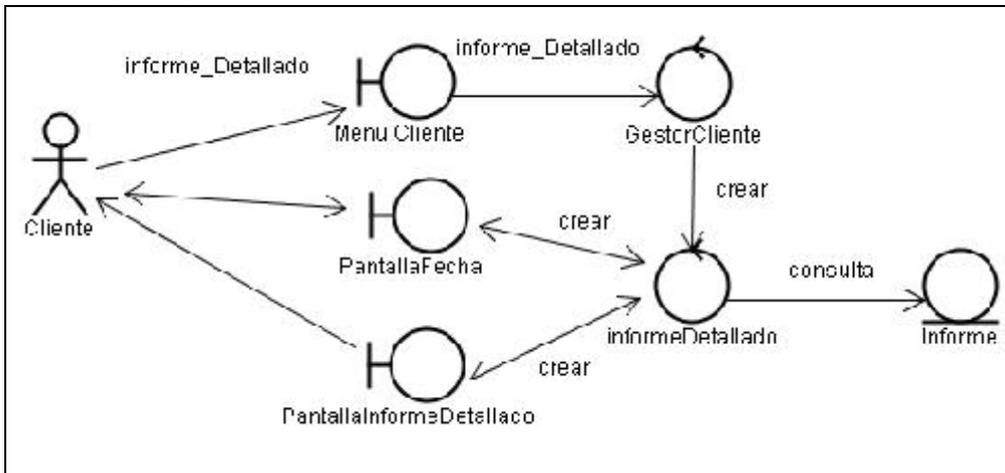


Figura 11 Diagrama estático análisis, caso de uso 9.

Caso de Uso nº10: " Llenar Test "

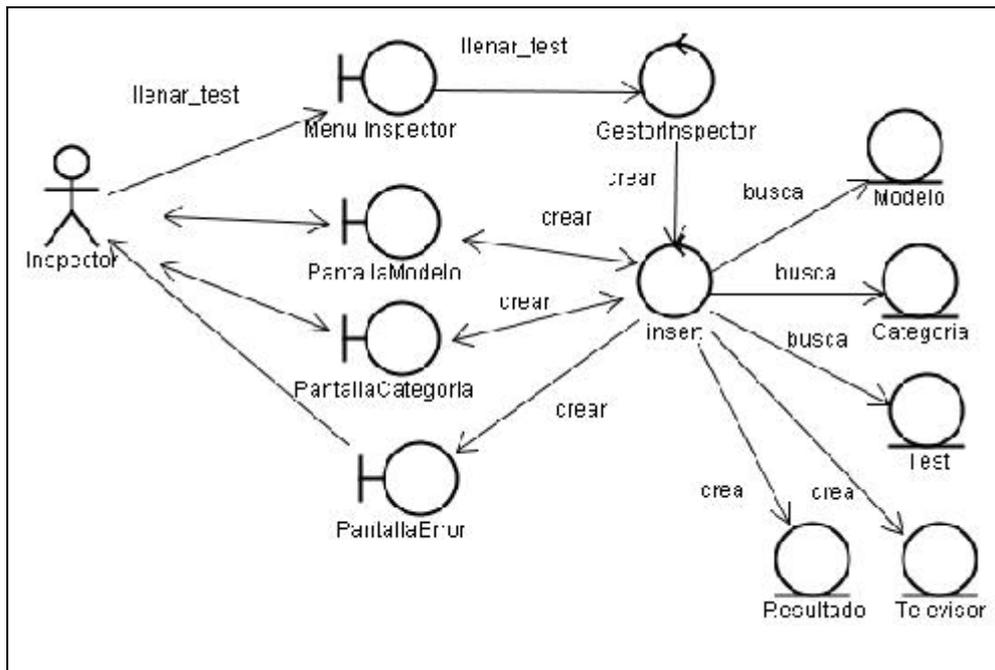


Figura 12 Diagrama estático análisis, caso de uso 10.

Caso de Uso nº11: " Modificar Test "

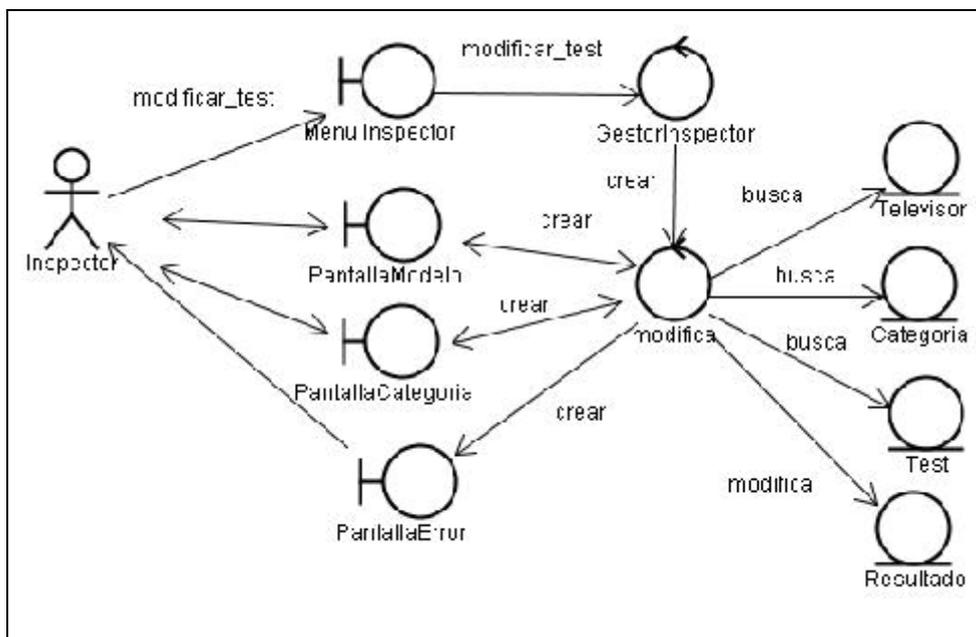


Figura 13 Diagrama estático análisis, caso de uso 11.

Caso de Uso nº12: " Consulta Test "

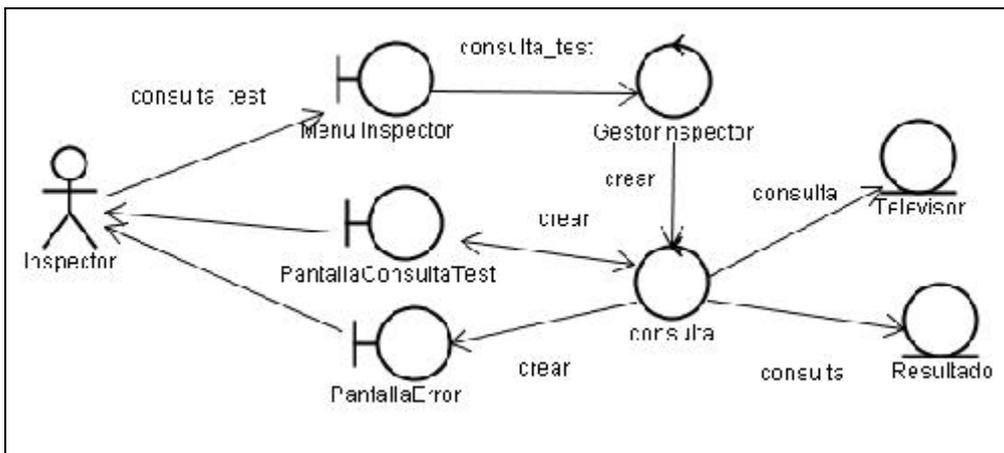


Figura 14 Diagrama estático análisis, caso de uso 12.

Caso de Uso nº13: " Crear Informe Diario "

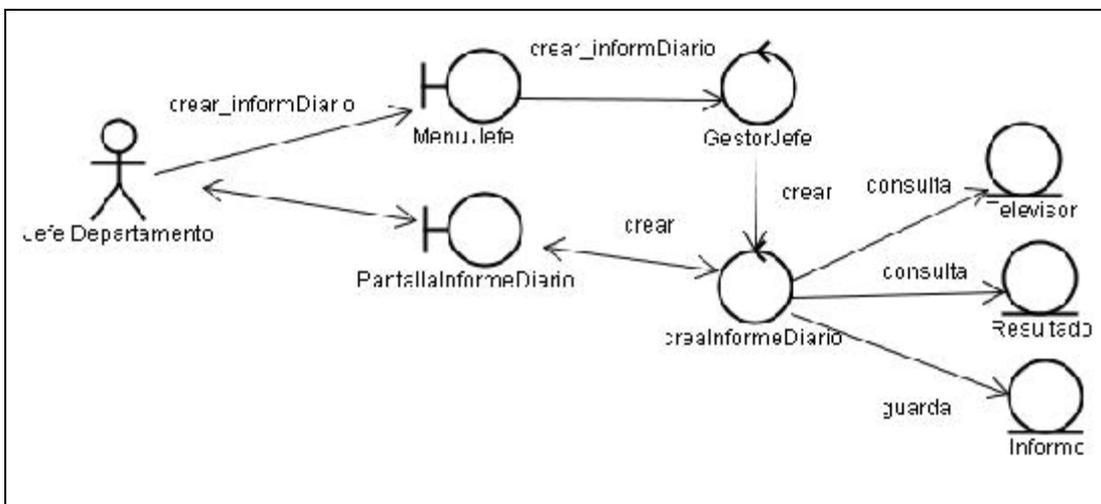


Figura 15 Diagrama estático análisis, caso de uso 13.

Caso de Uso nº14: " Crear Informe Detallado "

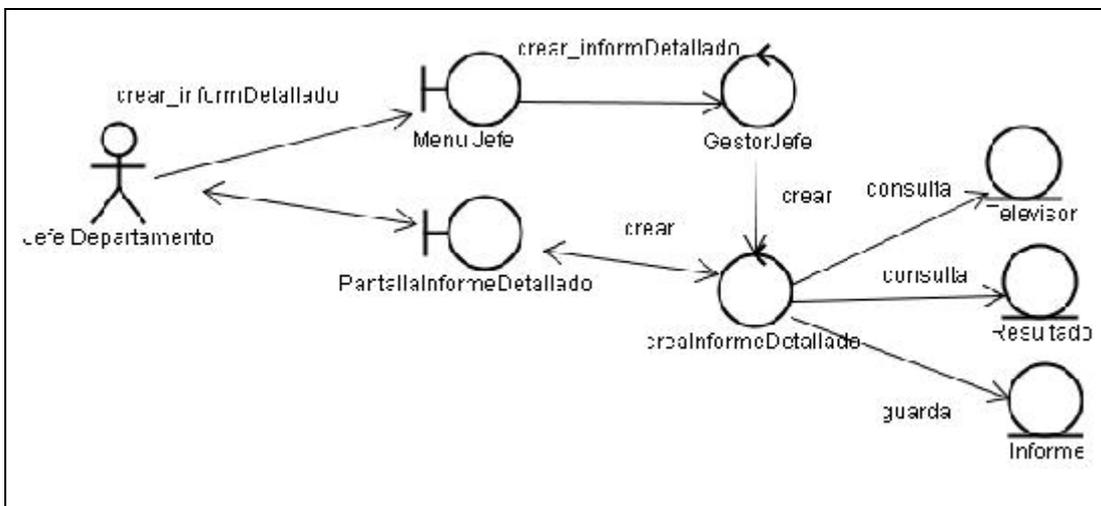


Figura 16 Diagrama estático análisis, caso de uso 14.

## ***Análisis de la Interficie de Usuario.***

El sistema esta compuesto por una página de bienvenida, para posteriormente pasar a validar al usuario que quiere acceder al sistema mediante un nombre de usuario y una contraseña. Esta acción permitirá acceder a un menu principal para cada perfil de usuario.

Página Bienvenida.



**Figura 17** Menu Principal.

Página Identificación.



**Figura 18** Menu Identificación.

Página Administrador.



**Figura 19** Menu Administrador.

Página Inspector.



Figura 20 Menu Inspector.

Página Cliente.



Figura 21 Menu Cliente.

Página Jefe Departamento.



Figura 22 Menu Jefe Departamento.

## Diseño.

### **Patrones de diseño.**

Para la realización de este sistema se ha utilizado el paradigma MVC (Model-View-Controller). De esta manera hemos separado por un lado la presentación mediante la interficie de usuario con sus páginas \*.html y \*.jsp, el control del programa que trata las entradas y salidas de información y por último el modelo que engloba toda la funcionalidad necesaria y el acceso a los datos.

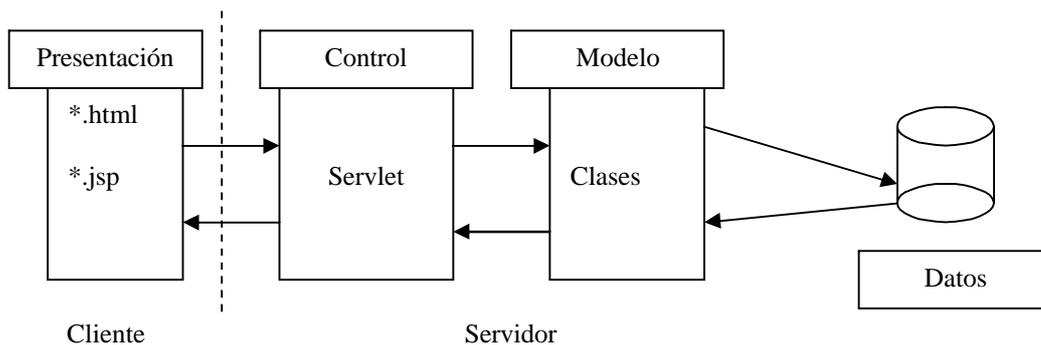


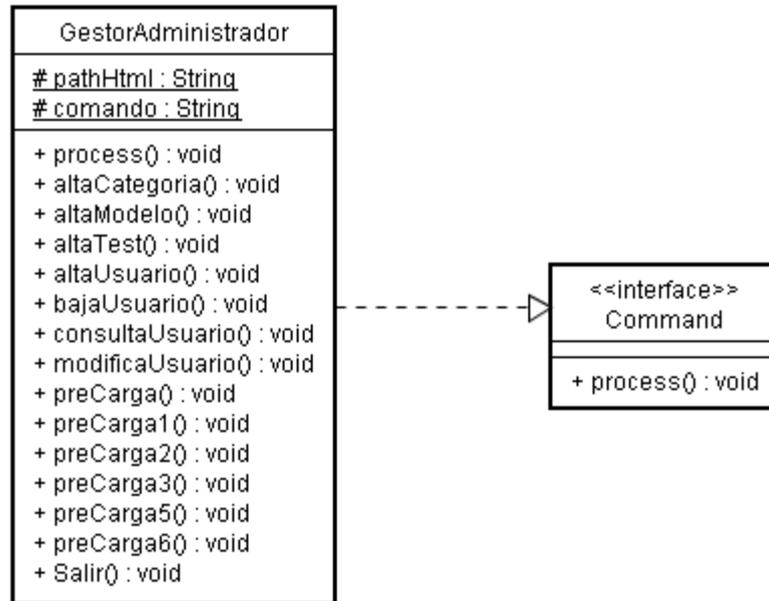
Figura 23 Model-View-Controller.

### **Diseño de la implementación de los casos de uso.**

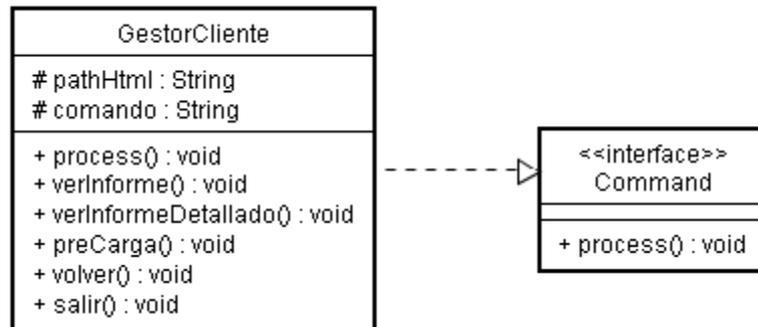
Para implementar los casos de uso, como se ha visto en los diagramas de análisis, se han creado gestores para cada perfil de usuario y de ellos depende toda la funcionalidad del perfil. Destacar que las clases de frontera se han incluido dentro de los propios gestores, aunque perfectamente se podrían crear clases que suministraran las diversas pantallas al usuario.

Añadir que todos los metodos que implementan los gestores tienen dos parámetros en su llamada estos son (`HttpServletRequest req, HttpServletResponse resp`) que son necesarios para la comunicación con el servlet principal y no se han incluido en las clases para facilitar la claridad en los diagramas

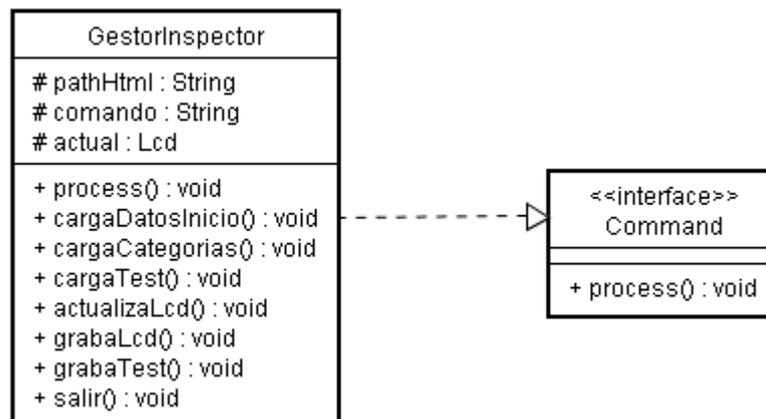
Casos de uso asociados al administrador (casos 1..7).



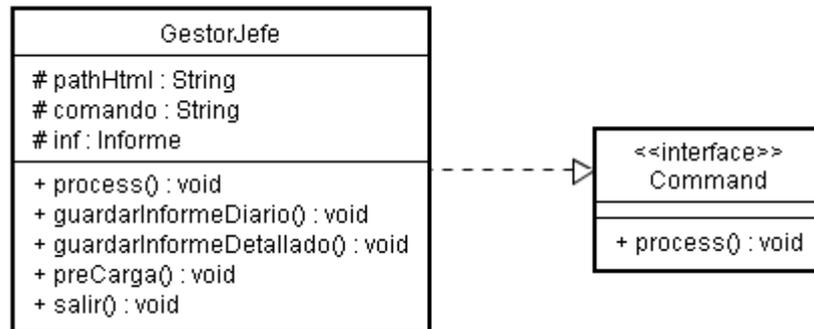
Casos de uso asociados al Cliente ( casos 8,9).



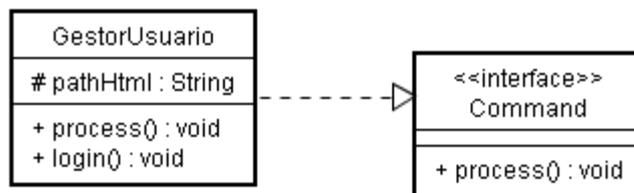
Casos de uso asociados al Inspector (casos 10,11,12)



Casos de uso asociados al jefe (casos 13,14)



También ha sido necesario implementar una clase que no aparecía explícitamente en los casos de uso, esto se debe a que se ha supuesto que los usuarios ya se habían identificado con anterioridad. Este gestor es el **GestorUsuarios** y proporciona la identificación de los diversos usuarios. Su firma es la siguiente:



### Revisión diagrama estático de diseño.

Para poder comunicar los gestores con las peticiones del navegador del cliente se ha definido un servlet principal que mediante una clase Factory recibe las peticiones de los clientes y carga el gestor correspondiente, a su vez canaliza el envío de las respuestas \*.jsp, \*.html a los clientes.

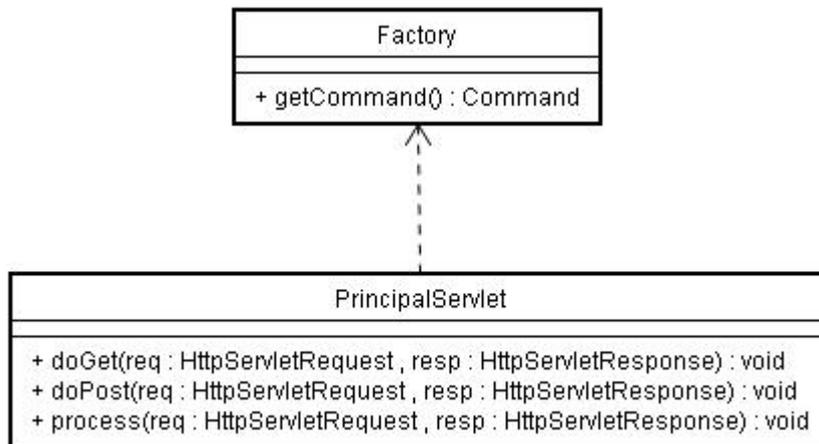


Figura 24 Diagrama estático diseño: servlet.

Las clases Modelo, Categoría y Test han quedado de la siguiente manera:

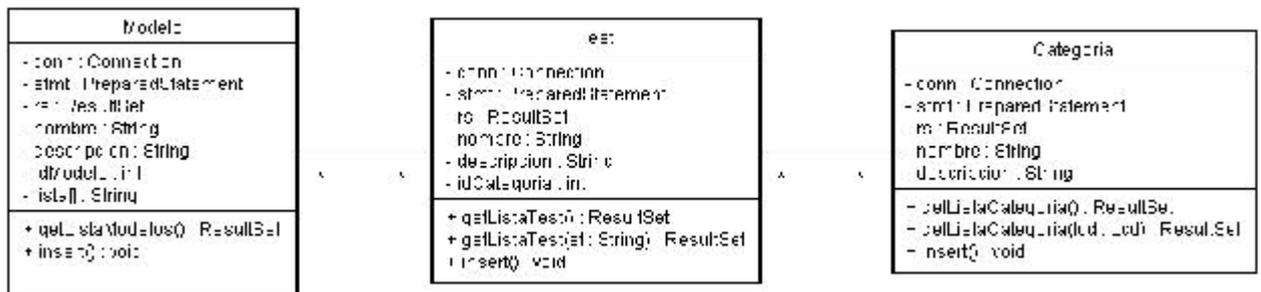
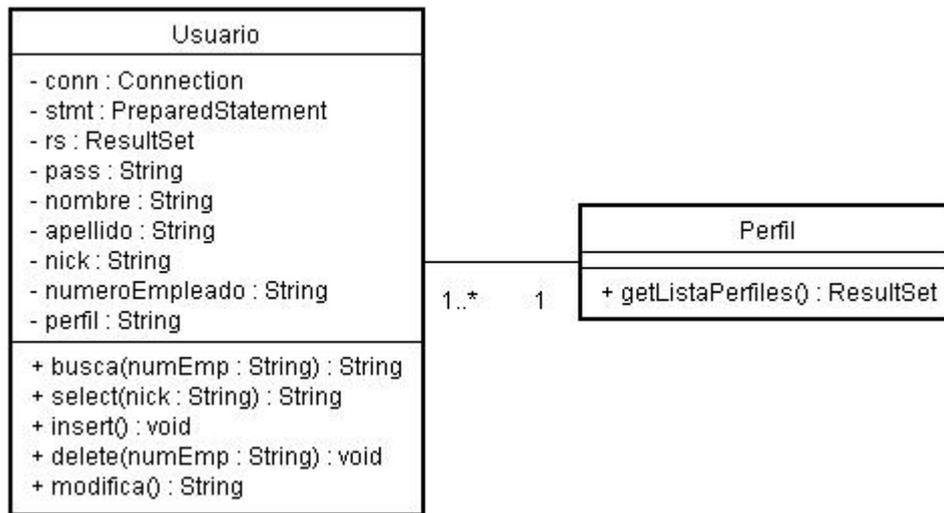


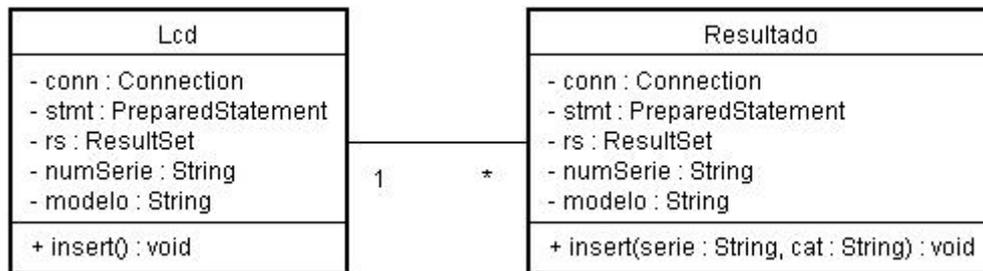
Figura 25 Diagrama estático diseño: Modelo.

Las clases Perfil y Usuario quedan:



**Figura 26 Diagrama estático diseño: usuario.**

La clase Televisor, se ha sustituido por Lcd, y la clase Resultado quedan:



**Figura 27 Diagrama estático diseño: Lcd**

Una clase que habíamos pasado por alto en el diagrama de análisis es la clase Informe que nos proporciona los métodos necesarios para recuperar los datos de los resultados, mostrarlos y almacenarlos.

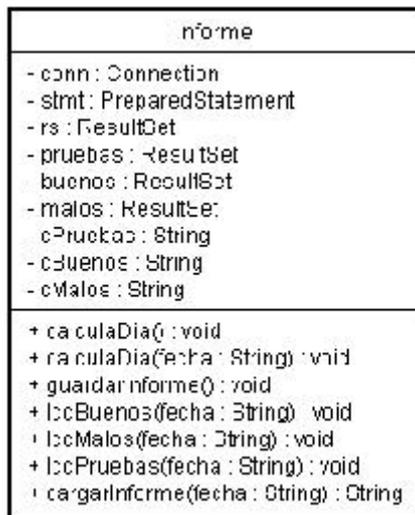


Figura 28 Diagrama estático diseño: Informe.

Por último se han creado dos clases que facilitan algunas operaciones, ellas són: la clase Util que devuelve la fecha y la hora en el formato apropiado para la base de datos y la clase GestorBD que proporciona la conexión y desconexión con la BBDD del sistema

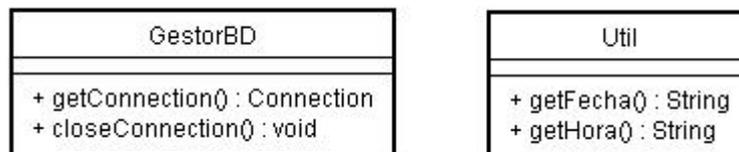


Figura 29 Diagrama estático diseño: complementos.

## Diseño de la persistencia.

Para poder mantener los datos una vez que la aplicación se ha cerrado, se ha optado por combinar una base de datos relacional para la mayoría de clases de nuestro sistema y una estructura de ficheros para las consultas de los informes, de esta manera las consultas complejas sobre la base de datos solamente se realizarán una sola vez al crear los diversos informes.

Para diseñar la base de datos se ha utilizado un entorno de diseño que nos proporciona la posibilidad de diseñar gráficamente las tablas, atributos, claves y relaciones construyendo más tarde la consulta SQL que nos proporcionará la creación de todas las tablas, atributos y relaciones.

Este entorno de diseño se llama Zdesign, y el resultado obtenido es el siguiente:

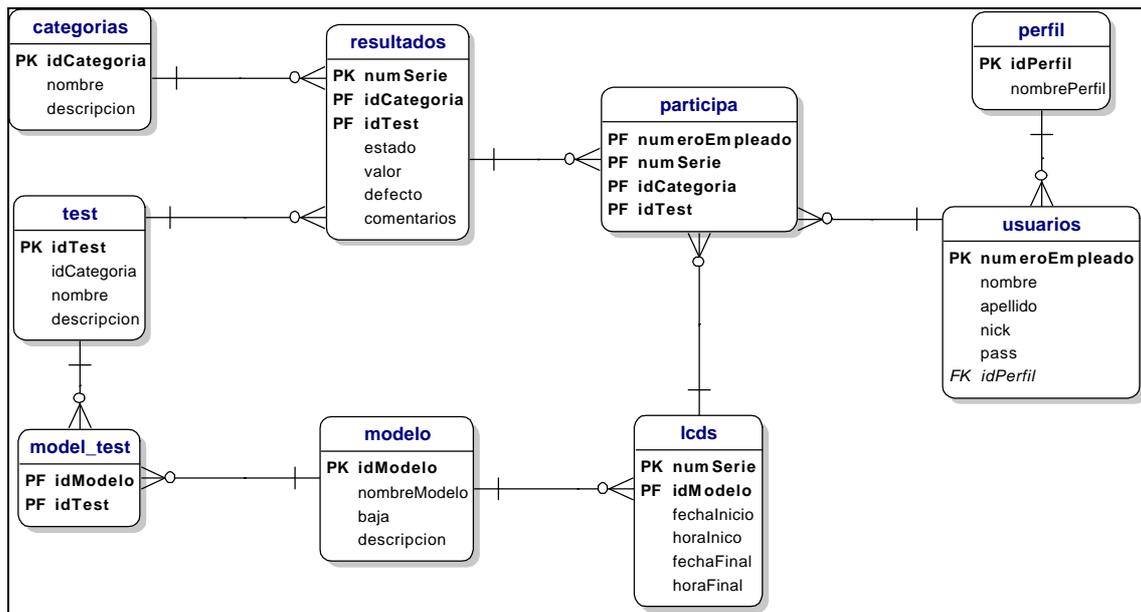


Figura 30 Diagrama Base datos.

Para que nuestro sistema pueda funcionar de forma autónoma una vez instalado es necesario incluir algunos datos sobre las tablas creadas. Ello nos permitirá entrar como administrador y empezar a generar categorías, test, modelos, usuarios,...etc.

```

# Connection: root@localhost:3306
# Host: localhost
# Saved: 2005-01-10 08:50:30
#
#===== #
# Project Filename: C:\temp\trabajo.dez
#
# Project Name: Inspeccion Calidad
#
# Author: Santi Valverde
#
# DBMS: MySQL 3
#
# Copyright:
#
# Generated on: 10/01/2005 8:13:18
#
#===== #
    
```

```
#===== #  
# Inserts  
#  
#===== #
```

```
# Perfiles #  
insert into perfil (nombrePerfil) values ("administrador");  
insert into perfil (nombrePerfil) values ("inspector");  
insert into perfil (nombrePerfil) values ("jefe");  
insert into perfil (nombrePerfil) values ("cliente");
```

```
# usuario inicial #  
insert into usuarios  
(numeroEmpleado,nombre,apellido,nick,pass,idPerfil)  
values  
(0,"inicio","sistema","administrador","1234",1);
```

El contenido del fichero SQL para la creación de las tablas se puede consultar en el [Anexo A..](#)

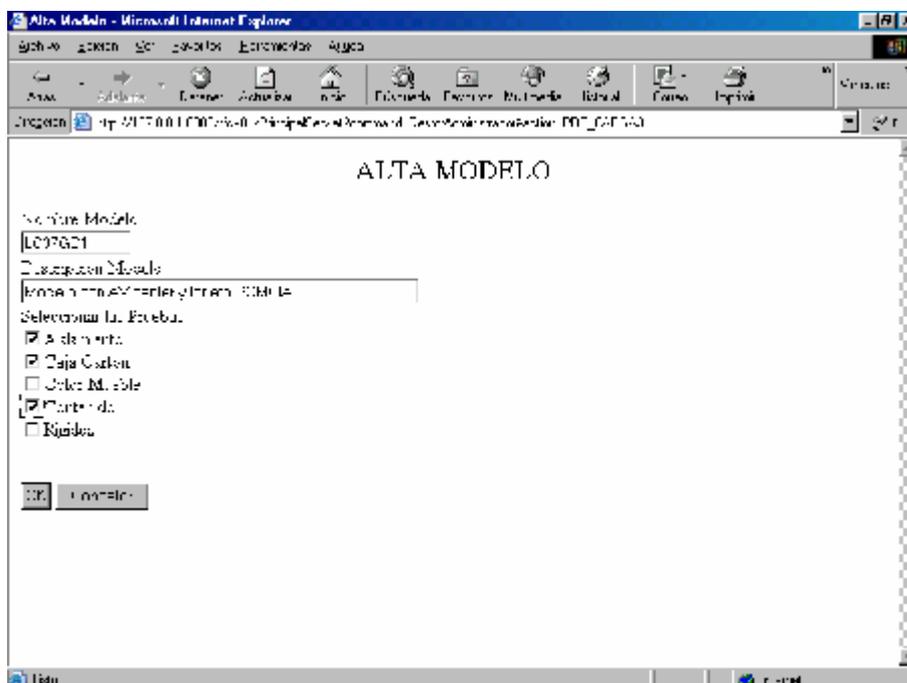
### **Diseño de la interficie gráfica de usuario.**

En líneas generales se ha buscado una interficie de usuario clara con pocos cambios entre los diversos menus, ordenes claras y siempre que ha sido posible menus desplegable para facilitar la elección de las diversas opciones. De esta manera se consigue que todo tipo de usuario, incluso con pocos conocimientos de informática puedan, mediante un periodo muy corto de aprendizaje, sacar el máximo partido al sistema.

Señalar que se ha dejado para una segunda fase el diseño corporativo, no se han incluido logotipos, banners identificativos de departamentos, imágenes para mejorar aún más la comprensión de las diversas acciones.

Algunos ejemplos de la interficie gráfica puede ser los siguientes:

Alta de un modelo.



**Figura 31 Interficie gráfica: alta modelo.**

En este caso se han utilizado check boxes para facilitar la selección de los diversos test que contendrá dicho modelo.

Otro ejemplo puede ser la creación de una categoría nueva.

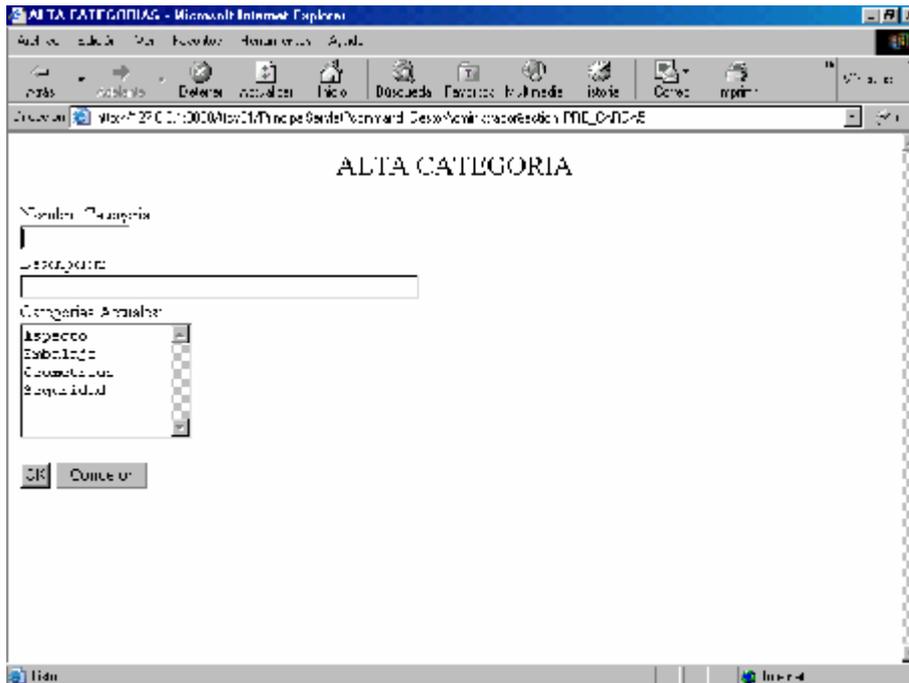


Figura 322 Interficie gráfica: alta categoria.

En este caso se ha utilizado una lista para mostrar las diferentes categorías ya creadas.

Lista de pruebas a realizar.

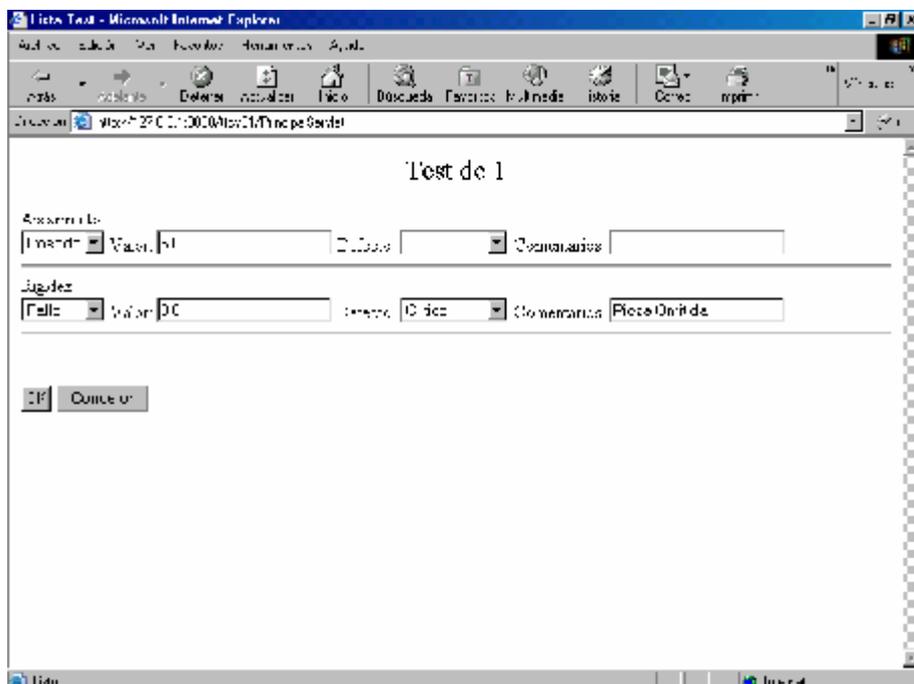


Figura 33 Interficie gráfica: lista test.

Tanto los defectos como el estado de la prueba se seleccionan desde menus desplegables.

Por último un ejemplo de la creación de un informe.

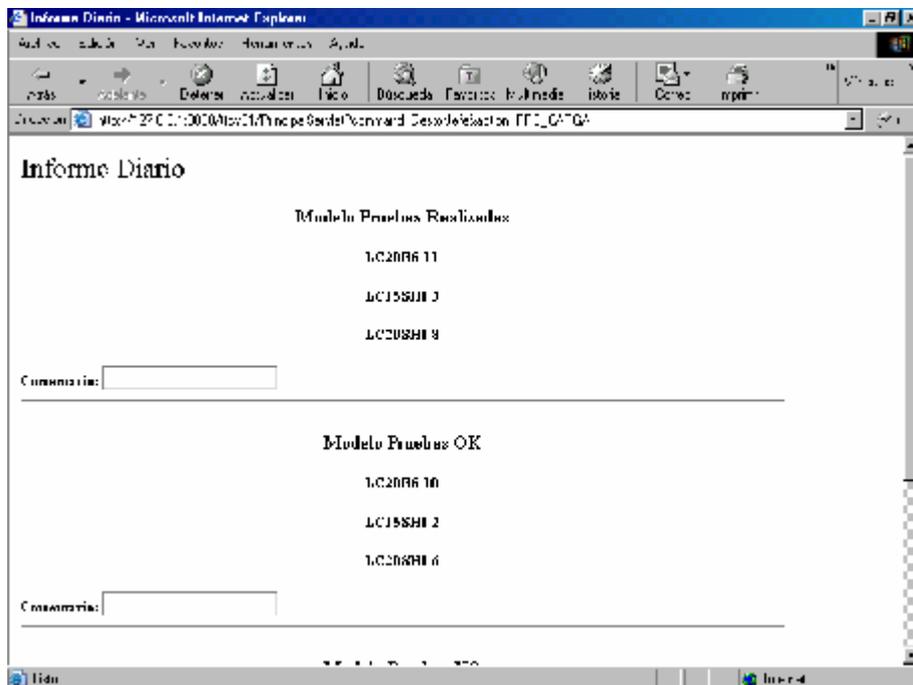


Figura 34 Interficie gráfica: Informe diario.

## **Implementación.**

### ***Descripción del producto obtenido.***

El sistema obtenido lo podemos considerar como la primera fase para establecer un sistema que proporcione por un lado, la posibilidad de automatizar ciertos procesos dentro del area productiva y por otro la posibilidad de no depender de aplicaciones independientes que necesitan un mantenimiento individual de cada sistema.

Rápidamente se podrían integrar a este sistema la gestión del etiquetado del producto y la gestión de las reparaciones.

Así mismo, aprovechando la red inalámbrica ya instalada, se esta contemplando la posibilidad de adquirir Pocket PC para el área de producción con la intención de realizar un seguimiento en tiempo real de las producciones.

### ***Software utilizado.***

Para la puesta en marcha del sistema anteriormente descrito, ha sido necesario la instalación de diversos paquetes que seguidamente se detallan:

- Lenguaje de programación Java J2SDK 1.4.1
- Base de datos relacional MySQL ver 3.23
- Servidor de servlets TOMCAT ver. 4

El sistema ha sido probado bajo Windows 98 y Windows 2000 con idéntico comportamiento.

El resto de recursos software han sido utilizados para la creación de la aplicación y su posterior documentación:

- IDE de programación Eclipse Ver. 1.2
- Diseño BBDD relacional Zdesign.
- Redacción memoria, paquete Microsoft Office
- Creación de diagramas JUDE.

## **Conclusiones y recomendaciones.**

### ***Nivel de consolidación de los objetivos.***

Destacar que al tratarse de un proyecto real, el planteamiento inicial desborda las perspectivas del más optimista. Lo que parecía en un principio una gestión sencilla de páginas web, se convierte poco a poco en un trabajo en el cual debería tomar parte un equipo de varias personas, esto me ha llevado a centrarme en la lógica del sistema más que en poder explorar otros campos como podrían ser el diseño web, la optimización para Pocket PCs o la obtención de datos estadísticos a partir de los datos de inspección.

Esto demuestra la necesidad de estructurar un proyecto en paquetes donde de forma más o menos independiente se puedan ir construyendo los diversos servicios.

A nivel personal y debido a que no me dedico a la programación, me he dado cuenta de las lagunas que se habrían en ciertos campos y que era preciso añadir un esfuerzo complementario para conseguir los objetivos de igual modo agradecer la ayuda suministrada por parte del consultor Toni Oller, que sin su ayuda habría sido imposible arrancar dicho trabajo.

Por último señalar que una vez visto el sistema funcionando, he podido ver la dimensión de las posibilidades de la tecnología J2EE aplicado a la comunicación a través de páginas web y obteniendo un producto no ligado a ninguna plataforma concreta e independiente de los sistemas software individuales.

### ***Líneas futuras de trabajo.***

Como se ha comentado anteriormente se podrán integrar por un lado el sistema de gestión de etiquetado y el control sobre las reparaciones del área de producción.

Centrandonos en el departamento de donde surgió la necesidad del proyecto, las líneas futuras son:

- La incorporación por parte de los inspectores de imágenes tomadas en la detección de los defectos y que puedan ser incorporadas a los informes.
- Un paquete de análisis estadístico que proporcione diversas informaciones complementarias de los actuales informes.

## **Bibliografía.**

### ***Libros utilizados.***

Título: Java 2  
Autor: Agustín Froufe.  
Editorial: RA-MA

Título:MySQL para Windows y Linux  
Autor: César Pérez.  
Editorial: RA-MA

Título: El Lenguaje Unificado de Modelado. Manual de Referencia.  
Autor: James Rumbaugh,Ivar Jacobson,Grady Booch  
Editorial: Addison-Wesley

También se ha utilizado la documentación de la UOC de BDI,BDII,XCII y EPI.

### ***Referencias de Internet.***

IDE de desarrollo Eclipse.  
<http://www.eclipse.org/>

Información sobre Tomcat, descarga de últimas versiones,...etc  
<http://jakarta.apache.org/tomcat/>

Información base de datos relacional y descarga de últimas versiones.  
<http://www.mysql.com/>

Tecnología java, paquetes,...etc Sun Microsystems.  
<http://www.sun.com/>

Código java, ejemplos,...etc.  
<http://www.programacion.com/>

## Anexo A.

### **Código SQL para la creación de las tablas.**

```
##### #
# Project Filename:      C:\temp\trabajo.dez          #
# Project Name:         #                          #
# Author:               #                          #
# DBMS:                 MySQL 4                     #
# Copyright:           #                          #
# Generated on:         10/01/2005 8:13:18          #
##### #

##### #
# Tables                                                       #
##### #

CREATE TABLE categorias (
    idCategoria INTEGER NOT NULL AUTO_INCREMENT,
    nombre VARCHAR(15) NOT NULL,
    descripcion TEXT,
    PRIMARY KEY (idCategoria),
    UNIQUE KEY IDX_categorias1(idCategoria)
);

CREATE TABLE modelo (
    idModelo INTEGER NOT NULL AUTO_INCREMENT,
    nombreModelo VARCHAR(15) NOT NULL,
    baja BOOL,
    descripcion TEXT,
    PRIMARY KEY (idModelo),
    UNIQUE KEY IDX_modelo1(idModelo)
);

CREATE TABLE test (
    idTest INTEGER NOT NULL AUTO_INCREMENT,
    idCategoria INTEGER NOT NULL,
    nombre VARCHAR(15) NOT NULL,
    descripcion TEXT,
    PRIMARY KEY (idTest),
    UNIQUE KEY IDX_test1(idTest)
);

CREATE TABLE model_test (
    idModelo INTEGER NOT NULL,
    idTest INTEGER NOT NULL,
    PRIMARY KEY (idModelo, idTest),
    KEY IDX_model_test1(idTest),
    KEY IDX_model_test2(idModelo)
);

CREATE TABLE lcds (
    numSerie VARCHAR(15) NOT NULL,
    idModelo INTEGER NOT NULL,
    fechaInicio DATE,
    horaInico TIME,
    fechaFinal DATE,
    horaFinal TIME,
    PRIMARY KEY (numSerie, idModelo),
    KEY IDX_lcds1(idModelo),
    UNIQUE KEY IDX_lcds2(numSerie)
);

CREATE TABLE resultados (
    numSerie VARCHAR(15) NOT NULL,
    idCategoria INTEGER NOT NULL,
```

```

        idTest INTEGER NOT NULL,
        estado ENUM('0','1','2') NOT NULL,
        valor VARCHAR(5),
        defecto ENUM('0','1','2','3','4') NOT NULL,
        comentarios TEXT,
        PRIMARY KEY (numSerie, idCategoria, idTest),
        KEY IDX_resultados1(idCategoria),
        KEY IDX_resultados2(idTest),
        UNIQUE KEY IDX_resultados3(numSerie, idCategoria, idTest)
    );

CREATE TABLE perfil (
    idPerfil INTEGER NOT NULL AUTO_INCREMENT,
    nombrePerfil VARCHAR(25) NOT NULL,
    PRIMARY KEY (idPerfil),
    UNIQUE KEY IDX_perfil1(idPerfil)
);

CREATE TABLE usuarios (
    numeroEmpleado INTEGER NOT NULL,
    nombre VARCHAR(40) NOT NULL,
    apellido VARCHAR(40) NOT NULL,
    nick VARCHAR(15) NOT NULL,
    pass VARCHAR(6) NOT NULL,
    idPerfil INTEGER NOT NULL,
    PRIMARY KEY (numeroEmpleado),
    KEY IDX_usuarios1(idPerfil),
    UNIQUE KEY IDX_usuarios2(numeroEmpleado)
);

CREATE TABLE participa (
    numeroEmpleado INTEGER NOT NULL,
    numSerie VARCHAR(15) NOT NULL,
    idCategoria INTEGER NOT NULL,
    idTest INTEGER NOT NULL,
    PRIMARY KEY (numeroEmpleado, numSerie, idCategoria, idTest),
    KEY IDX_participa1(numSerie, idCategoria, idTest),
    KEY IDX_participa2(numSerie),
    KEY IDX_participa3(numeroEmpleado)
);

##### #
# Foreign Keys #
##### #

ALTER TABLE model_test
    ADD FOREIGN KEY (idTest) REFERENCES test (idTest);

ALTER TABLE model_test
    ADD FOREIGN KEY (idModelo) REFERENCES modelo (idModelo);

ALTER TABLE lcds
    ADD FOREIGN KEY (idModelo) REFERENCES modelo (idModelo);

ALTER TABLE resultados
    ADD FOREIGN KEY (idCategoria) REFERENCES categorias (idCategoria);

ALTER TABLE resultados
    ADD FOREIGN KEY (idTest) REFERENCES test (idTest);

ALTER TABLE usuarios
    ADD FOREIGN KEY (idPerfil) REFERENCES perfil (idPerfil);

ALTER TABLE participa
    ADD FOREIGN KEY (numSerie,idCategoria,idTest) REFERENCES resultados
    (numSerie, idCategoria, idTest);

ALTER TABLE participa
    
```

```
ADD FOREIGN KEY (numSerie) REFERENCES lods (numSerie);  
  
ALTER TABLE participa  
ADD FOREIGN KEY (numeroEmpleado) REFERENCES usuarios (numeroEmpleado);
```