

## INDICE

<b>INTRODUCCION .....</b>	<b>1</b>
<b>OBJETIVOS .....</b>	<b>2</b>
<b>ANALISIS DE REQUISITOS.....</b>	<b>3</b>
Estructura de la encuesta .....	3
Perfiles de Usuario.....	3
Estructura de la aplicación.....	4
<b>ANALISIS Y DISEÑO .....</b>	<b>6</b>
Introducción .....	6
Construcciones básicas .....	7
Diseño de la Base de Datos .....	9
Diagramas de Flujo.....	11
Especificaciones de la aplicación .....	17
Interficie gráfica.....	20
<b>IMPLEMENTACIÓN.....</b>	<b>21</b>
Introducción .....	21
<i>Scripts</i> de Servidor.....	22
Objetos integrados de ASP.....	22
Entorno de trabajo .....	23
Diseño de la aplicación .....	27
<b>DOCUMENTACION .....</b>	<b>41</b>
<b>CONCLUSIONES .....</b>	<b>42</b>

## INTRODUCCIÓN

El Grupo Finaf92 consta de una División Servicios formada por varios departamentos: Tecnologías de la Información, Recursos Humanos, Tesorería y Contabilidad, Análisis y Control, E\_Business,... encargados de dar servicio a todas las sociedades del grupo. Uno de los objetivos de esta división es analizar el grado de satisfacción de los empleados del grupo mediante encuestas.

Las encuestas se realizan anualmente de forma anónima y constan de los datos del encuestado (sociedad a la que pertenece, división, cargo y tiempo en la empresa), y una serie de variables de análisis que se evalúan con una puntuación del 1 al 10.

Actualmente, la encuesta se realiza con una hoja de cálculo que se envía a través de correo electrónico a todos los empleados. Una vez cumplimentada, los empleados depositan el resultado impreso en un buzón ubicado en la recepción de su sociedad, para posteriormente ser enviadas a la división servicios. El departamento de organización es el encargado de introducir los resultados manualmente en un programa de análisis estadístico (por ejemplo, MiniTab) y analizar los resultados obtenidos.

Los puntos débiles que se han detectado con el uso de este procedimiento son:

- Falta de Coherencia: Ciertos encuestados contestan que no son usuarios de un departamento y posteriormente realizan la valoración del departamento
- Fiabilidad de los datos: Las valoraciones obtenidas se introducen manualmente en un aplicativo de análisis por lo que se pueden producir errores en la introducción de los datos
- Procedimiento poco ágil: El tiempo que transcurre entre la divulgación de la encuesta y el análisis de los resultados obtenidos es considerable.

## OBJETIVO

El objetivo del trabajo es diseñar, desarrollar e implantar una aplicación que permita paliar las deficiencias del sistema actual.

La aplicación ha de cumplir las siguientes funcionalidades:

- Cada empleado podrá realizar la encuesta desde su puesto de trabajo en un entorno HTML, sin necesidad de imprimir los resultados y depositarlos en un buzón.
- Las encuestas realizadas quedarán guardadas en una base de datos. De esta forma se permite mantener un histórico de los resultados obtenidos cada año.
- Las preguntas y secciones podrán ser modificadas por un responsable previamente autenticado.
- Conocer el % de participación por cada sociedad y departamento. Las encuestas son anónimas, únicamente disponemos de datos orientativos del encuestado.
- Extraer los resultados para poder analizar los datos obtenidos con un programa de análisis estadístico. De esta forma se evita introducir manualmente los resultados de las encuestas.

El resultado de este trabajo será una aplicación en formato HTML que permita recoger las encuestas realizadas por todos los empleados del grupo que tienen acceso a la red de ordenadores de la empresa.

## **ANÁLISIS DE REQUISITOS**

### **1.- Estructura de la encuesta**

Las encuestas se realizan anualmente y constan de:

#### **a).- Datos del encuestado**

El encuestado deberá indicar la sociedad a la que pertenece, el área, la posición y la antigüedad en la empresa

#### **b).- Valoraciones**

Por cada departamento a encuestar existen una serie de preguntas que han de ser valoradas del 1 al 10 según grado de satisfacción. Dentro de cada departamento se valorarán una serie de áreas, por ejemplo, dentro del departamento de Tecnologías de la Información se valorarán el soporte de hardware, soporte de software, soporte de SAP,...etc.

En cada área, el encuestado deberá indicar si es usuario o no. En caso de que no utilice el área concreto quedarán deshabilitadas las preguntas que dependan de este área.

Las preguntas que se realizan en cada área de un departamento pueden coincidir, pero no siempre desde todas las áreas se realizarán las mismas preguntas. Por lo tanto, las preguntas realizadas dependerán del departamento y área a encuestar.

### **2.- Perfiles de Usuario**

En la aplicación se han de tener dos tipos de acceso:

#### **a).- Administrador**

Persona que podrá gestionar las diferentes opciones de la aplicación, por ejemplo, creación de nuevos departamentos, nuevas preguntas, añadir nuevas opciones a los datos del encuestado, ...

Para las tareas administrativas será necesario la autenticación mediante un *password*.

#### **b).- Encuestado**

Todos los empleados que tengan acceso a la red del grupo Finaf92 han de realizar la encuesta incluidas las personas de los departamentos encuestados.

### **3.- Estructura de la aplicación**

La aplicación constará de tres áreas:

#### **3.1.- Mantenimiento de la encuesta**

Todas estas funciones podrán ser realizadas únicamente por el administrador de la aplicación:

- Datos encuestado
  - o Añadir/Modificar/Eliminar Sociedad
  - o Añadir/Modificar/Eliminar Departamento
  - o Añadir/Modificar/Eliminar Posición
  - o Añadir/Modificar/Eliminar Antigüedad
- Datos encuesta
  - o Añadir/Modificar departamentos a encuestar
  - o Añadir/ Modificar áreas
  - o Añadir/ Modificar preguntas
  - o Generar nueva encuesta
  - o Mantenimiento de departamentos, áreas y preguntas de la encuesta
  - o Modificar periodo de aplicación: Las encuestas se realizan anualmente pero no se pueden introducir encuestas durante todo el año ya que para realizar los análisis estadísticos es necesario finalizar un periodo. Al generar una nueva encuesta se define el periodo de validez.

#### **3.2.- Encuesta**

La única opción posible será generar encuesta. La encuesta se podrá cancelar en la mitad del proceso pero no podrá ser anulada una vez realizada.

El encuestado deberá valorar todos los departamentos y áreas de los que es usuario.

#### **3.3.- Estadísticas**

Una vez realizadas las encuestas es necesario analizar los datos obtenidos para poder mostrar los resultados.

Se ha de estudiar la posibilidad de guardar los resultados de las encuestas por año, con el fin de poder realizar análisis estadísticos comparativos en diferentes etapas.

Se ha de disponer de estadísticas básicas dentro de la aplicación, como:

- % Participación por Sociedad: Se indicará el número de encuestas realizadas por cada sociedad.
- Medias de puntuación por departamento. De cada departamento a encuestar se visualizará la valoración obtenida.
- Medias de puntuación por sociedad. De cada sociedad se visualizará las puntuaciones que ha realizado sobre cada departamento a encuestar.

Independientemente de que funciones de las anteriores sean implementadas se podrán extraer los resultados de la base de datos para poder analizar los datos obtenidos con un programa de análisis estadístico. De esta forma se evita introducir manualmente los resultados de las encuestas.

Este área podrá ser utilizado únicamente por el administrador de la aplicación.

# ANÁLISIS Y DISEÑO

## 1.- Introducción

Para la realización de la aplicación se parte de una serie de premisas que son importantes para el análisis y diseño:

- El desarrollo de la aplicación se realizará en un entorno Web.
- La cantidad de accesos al sistemas no serán elevados y únicamente se darán en un periodo establecido.
- El sistema operativo sobre el que se instalará la aplicación es un Windows NT Server 4.0
- Los gestores de base de datos propuestos debido al rendimiento que ha de proporcionar la aplicación son Microsoft SQL Server o Microsoft Access (Jet).

Las tres soluciones posibles para el desarrollo de la aplicación dependiendo del servidor Web utilizado y de la plataforma elegida son los siguientes:

### a).- Solución J2EEE basada en productos libres y Open Source

El servidor Web utilizado sería Apache-Tomcat que sería el soporte para las páginas estáticas y como contenedor de JSPs (Java Server Pages) y Servlets, sobre los que se desarrollaría la aplicación.

Las principales ventajas que aporta son la portabilidad a otras arquitecturas que sigan el estándar J2EEE y la reducción de costes al utilizar productos libres. La principal desventaja es que no está recomendado para entornos de producción fuertes, con gran cantidad de accesos de usuarios, ...etc.

### b).- Solución J2EEE basada en productos comerciales

El servidor Web utilizado sería iPlanet que sería el soporte para las páginas estáticas y como contenedor de JSPs (Java Server Pages) y Servlets, sobre los que se desarrollaría la aplicación.

Las principales ventajas que aporta esta solución son la portabilidad a otras arquitecturas al igual que la solución anterior y la utilización de productos líderes en el sector. La principal desventaja es el coste de los productos.

### c).- Solución basada en arquitectura Microsoft

El servidor Web utilizado sería el IIS (*Internet Information Server*) que sería el soporte para las páginas estáticas HTML y ASPs (*Active Server Pages*), sobre los que se desarrollaría la aplicación.

Las principales ventajas que aporta son la fácil integración con la mayoría de servicios Microsoft y el coste de los productos. La principal desventaja es que la tecnología Microsoft es propietaria y no es portable a otras arquitecturas.

Dada la magnitud de la aplicación a desarrollar y el grado de utilización, es evidente que la segunda opción queda descartada debido al elevado coste. Respecto las otras dos soluciones propuestas la más acorde al problema presentado es la tercera solución debido a la fácil integración con la mayoría de los servicios Microsoft de los que pudiera disponer la empresa, teniendo en cuenta que toda la infraestructura de la empresa utiliza tecnología Microsoft. Además se añade la mayor facilidad de programación en entornos Microsoft, considerando que el desarrollo de la primera solución es en JAVA.

Por estos motivos la solución con la que se desarrollará la aplicación será la basada en arquitectura Microsoft (programación de ASPs).

## **2.- Construcciones básicas**

Antes de todo es necesario definir las entidades con las que se trabajarán, que fueron concretadas en la etapa de análisis de requisitos.

### a).- Datos del encuestado

- **SOCIEDAD:** Indica la sociedad a la que pertenece el encuestado. En caso de que la sociedad disponga de ciertas divisiones se crearán entradas de la forma:

Nombre\_Empresa DIV A  
Nombre\_Empresa DIV B

- **DEPARTAMENTO:** Departamento en el que trabaja el encuestado. Los departamentos serán comunes a todas las sociedades aunque no todas dispongan de todos los departamentos indicados.
- **POSICION:** Cargo que ocupa el encuestado.
- **ANTIGÜEDAD:** Antigüedad en la empresa. Se definirán unos márgenes de antigüedad para ubicar al encuestado.

### b).- Datos a encuestar

- **DPTO. A ENCUESTAR:** Todos aquellos departamentos que pertenecen a la Div. Servicio y que se pretenden valorar.
- **AREAS:** Dentro de cada departamento se valorarán un serie de áreas.
- **PREGUNTAS:** Quedarán determinadas por el departamento y área.

La estructura general del formulario será:



# DEPARTAMENTO A ENCUESTAR 1

## AREA 1

	NU	0	1	2	3	4	5	6	7	8	9	10
Pregunta 1												
Pregunta 2												
.												
.												
Pregunta n												

## AREA 2

	NU	0	1	2	3	4	5	6	7	8	9	10
Pregunta 1												
Pregunta 2												
.												
.												
Pregunta n												

...

## AREA N

	UN	0	1	2	3	4	5	6	7	8	9	10
Pregunta 1												
Pregunta 2												
.												
.												
Pregunta n												

## COMENTARIOS / PUNTOS DE MEJORA

De cada pregunta se valorará el grado de satisfacción. Si no se es usuario del área indicado se marcará la opción *NU* (no usuario). Por defecto, esta opción estará siempre predeterminada y variará en caso de querer valorar esta pregunta.

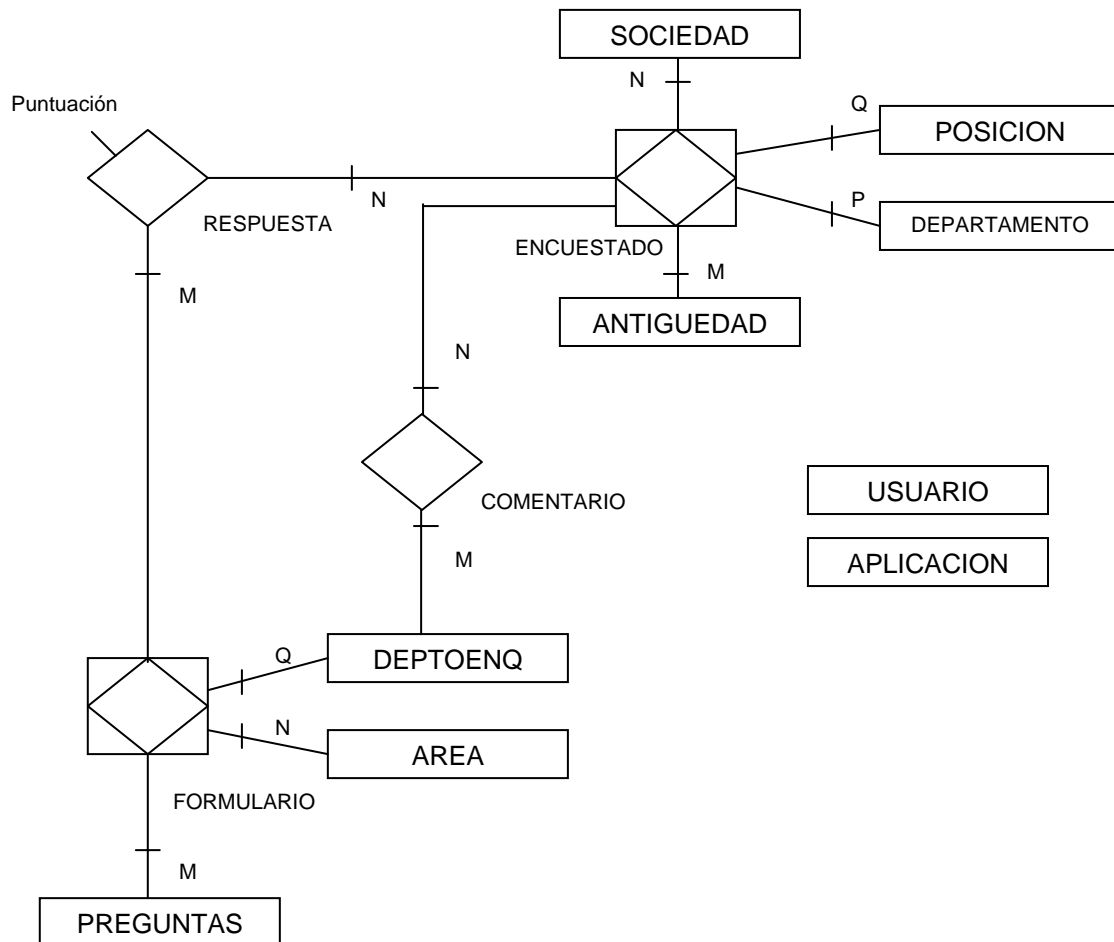
No existirán más niveles definidos en las preguntas, de cada área se realizarán una serie de preguntas que pueden ser valoradas. La implementación de la aplicación no permitirá la creación de subpreguntas.

### c).- Respuestas

Las respuestas quedarán determinadas por los datos del encuestado y los datos a encuestar. Las respuestas almacenadas dependerán del año para poder mantener un histórico.

### 3.- Diseño de la base de datos

#### 3.1.- Diseño Conceptual – Modelo Entidad-Relación (ER)



#### 3.2.- Diseño Lógico

Se transformará el modelo ER en un modelo relacional, de tal forma que tanto las entidades definidas, como la interrelación RESPUESTA (M:N), como la interrelación COMENTARIO (M:N), como las entidades asociativas, se transformarán en relaciones.

SOCIEDAD (id\_sociedad, nom\_sociedad, activado)

DEPARTAMENTO (id\_departamento, nom\_departamento, activado)

POSICION (id\_posicion, nom\_posicion, activado)

ANTIGÜEDAD (id\_antiguedad, nom\_antiguedadd, activado)

ENCUESTADO (id\_encuestado, sociedad, departamento, antigüedad, posicion, valido, fecha\_acceso)

Donde {sociedad} referencia SOCIEDAD

{departamento} referencia DEPARTAMENTO  
{antigüedad} referencia ANTIGÜEDAD  
y {posicion} referencia POSICION

DPTOENQ (id\_dptoeng, nom\_dptoeng)

AREA (id\_area, nom\_area)

PREGUNTA (id\_pregunta, texto)

FORMULARIO (id\_formulario, dptoeng, area, pregunta)

Donde {dpteng} referencia DPTOENQ

{area} referencia AREA

y {pregunta} referencia PREGUNTA

RESPUESTA (id\_encuestado, id\_formulario, puntuacion)

Donde {id\_encuestado} referencia ENCUESTADO

y {id\_formulario} referencia FORMULARIO

USUARIO (user, password)

APLICACIÓN (id\_aplicacion, finicio, ffin)

En la mayoría de las relaciones se ha definido un identificador a pesar de que en muchos casos no era necesario porque la entrada quedaba bien identificada con el resto de atributos, pero de esta forma se consigue trabajar de una forma más clara y cómoda.

Las entidades SOCIEDAD, DEPARTAMENTO, POSICION y ANTIGÜEDAD tienen el atributo *activo* que indica si esa sociedad, departamento, posición o antigüedad está activa para ser mostrada en el formulario. No se pueden eliminar los registros de estas entidades porque es posible que estén asociados a algún formulario de años anteriores.

La entidad ENCUESTADO contiene el atributo *valido* que indica si ese encuestado ha finalizado el proceso de realización de la encuesta correctamente. Al finalizar la encuesta el atributo tomará el valor de *cierto*, en caso de abandonar el procesos o producirse error el valor será *falso*.

Las dos últimas relaciones son de sistema, la relación USUARIO permitirá identificar al administrador, contendrá su *login* y *password*. La relación APLICACIÓN permite marcar la fecha de inicio y de fin de la aplicación, es decir, el periodo en el cual se podrán realizar las encuestas. Dentro de este periodo no se permitirá la modificación del formulario y fuera del periodo no se permitirá la realización de ninguna encuesta.

### 3.3.- Diseño Físico

A la hora de seleccionar el servidor de base de datos para almacenar las estructuras en forma de tablas relacionales, es necesario especificar parámetros como capacidad, nº registros que dispondrá la aplicación, grado de utilización de la aplicación,...etc.

Los dos servidores de base de datos a analizar son SQL Server y Microsoft Jet (de Access). La ventaja principal de SQL Server frente a Jet se basa en que el primer gestor es un ejecutable autónomo mientras que Jet son archivos DLL, es por esto que SQL Server es mucho más conveniente para la publicación Web.

En la medida de lo posible se intentará utilizar el motor Jet de Microsoft por su sencillez en la implantación y por la disposición de licencia. Cabe analizar si cubre nuestras necesidades.

La aplicación se utilizará durante un periodo de aproximadamente dos meses y se almacenarán las respuestas de cada año. Teniendo en cuenta el nº de preguntas y el nº de encuestados, el nº de registros puede ser de aproximadamente 50.000 entradas que únicamente contendrán el identificador del encuestado, el identificador del formulario y la puntuación.

Para evitar el crecimiento anual de respuestas en una misma relación, al comenzar un año se generará otra tabla de respuestas vacía con el identificador del año (por ejemplo, RESPUESTA\_2002).

Las limitaciones que presenta Microsoft Access 97 que más afectan a la implementación de la aplicación son:

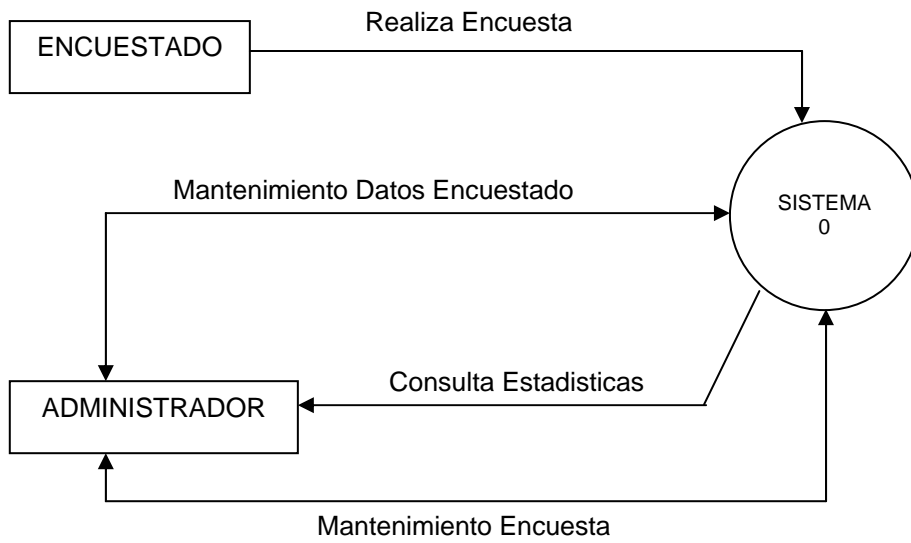
- Tamaño Archivo MDB: 1GB
- Nº objetos en una base de datos: 32.768
- Tamaño en la tabla: 1GB
- Nº Caracteres en un campo memo: 65.535

Con estas limitaciones y teniendo en cuenta la utilización y complejidad de la aplicación, aparentemente no existe ningún obstáculo en la implantación de la aplicación con el gestor de base de datos de Microsoft Access.

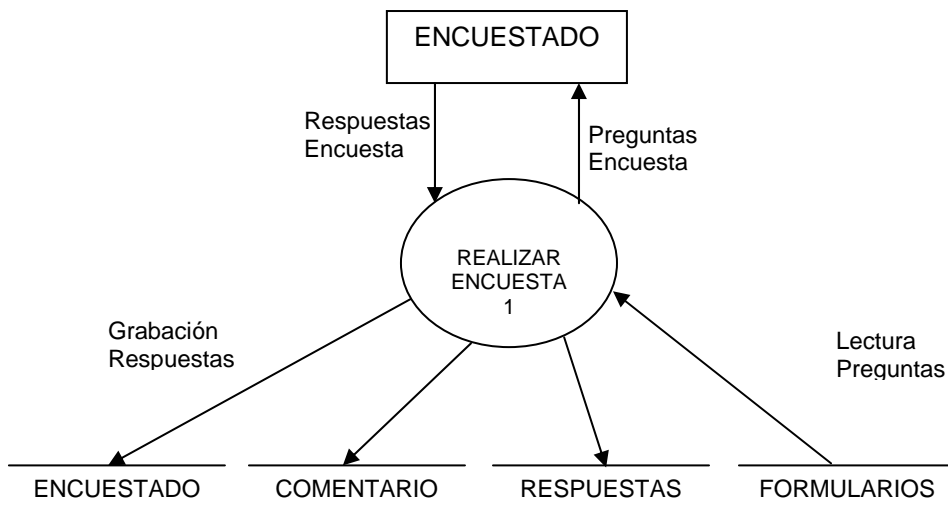
#### **4.- Diagramas De Flujo**

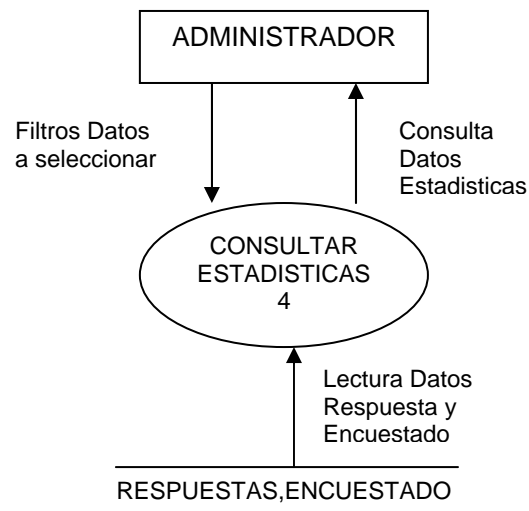
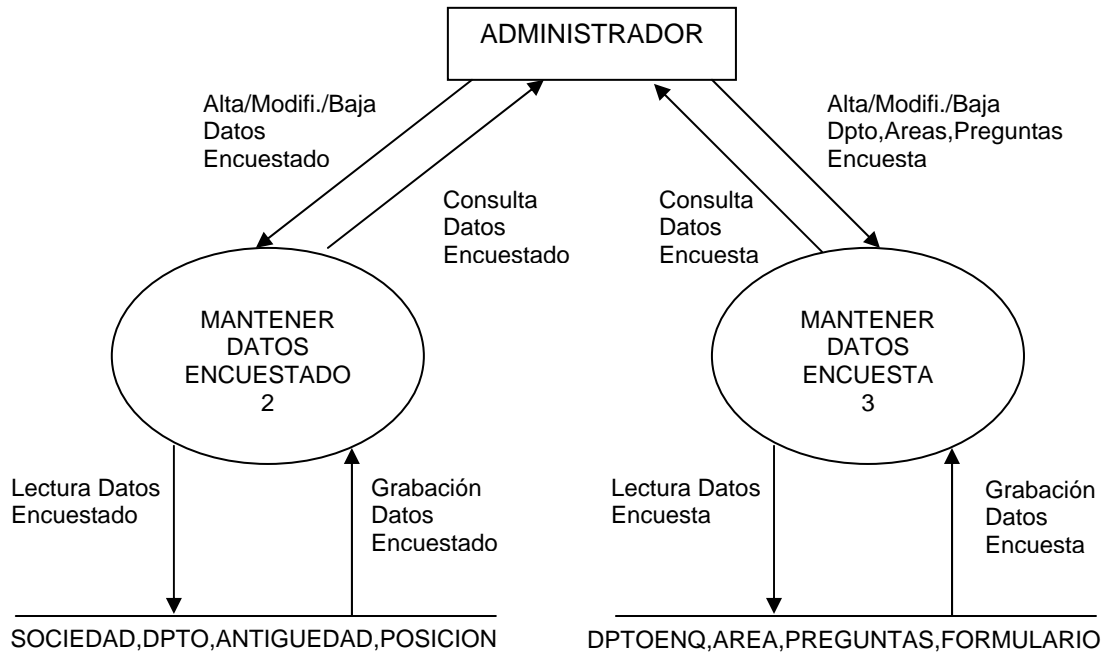
La aplicación constará básicamente de tres áreas: mantenimiento de la encuesta, realización de la encuesta, y extracción de estadísticas.

### 4.1.- Diagrama de Contexto – Nivel 0

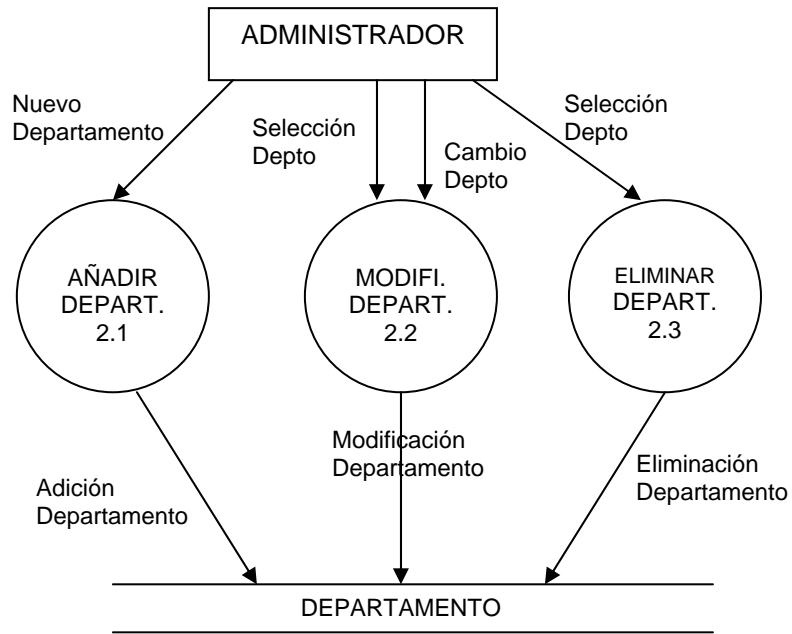


### 4.2.- DFD 0 – Nivel 1





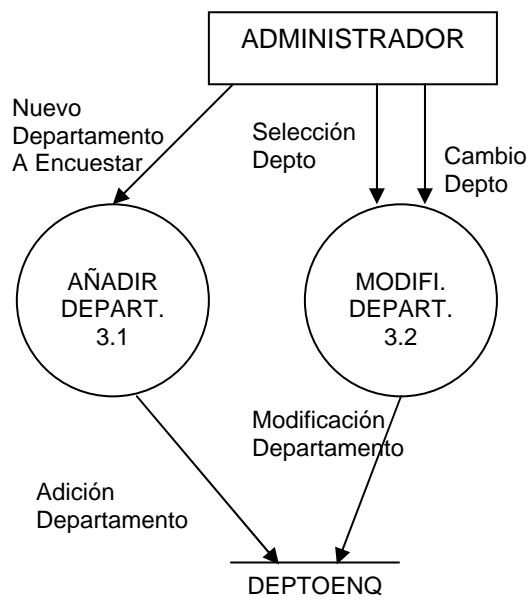
### 4.3.- DFD 2: Proceso Mantener Datos Encuestado – Nivel 2



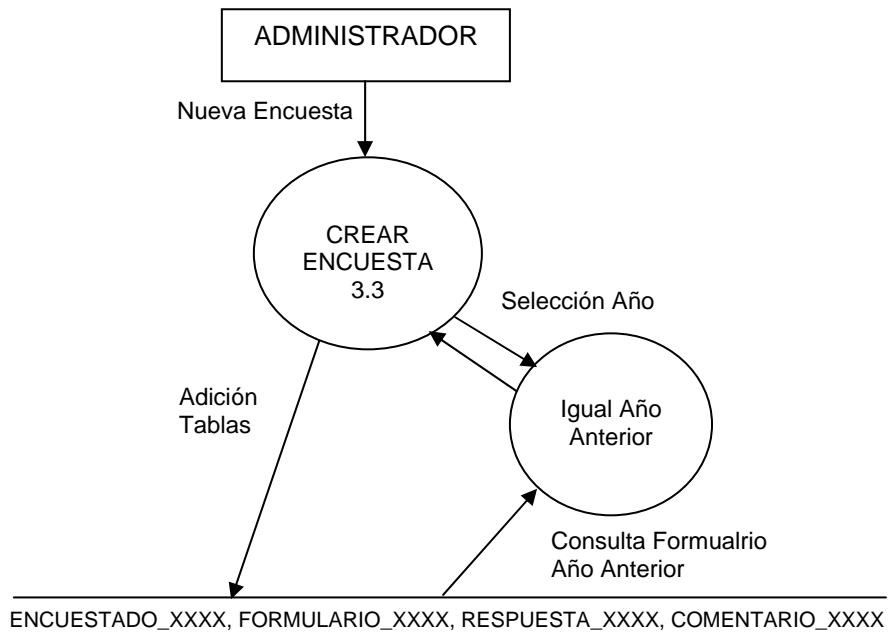
La opción eliminar departamento hace referencia a desactivar este departamento, modificar el atributo *activo* a falso.

Se repite el mismo esquema para las opciones de Añadir/Modificar/Eliminar Sociedad, Antigüedad y Posición.

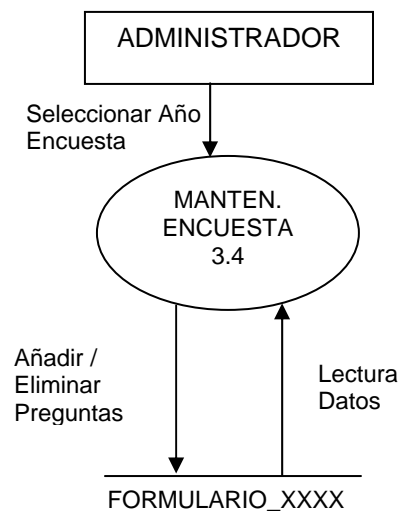
### 4.4.- DFD 3: Proceso Mantener Datos Encuesta – Nivel 2



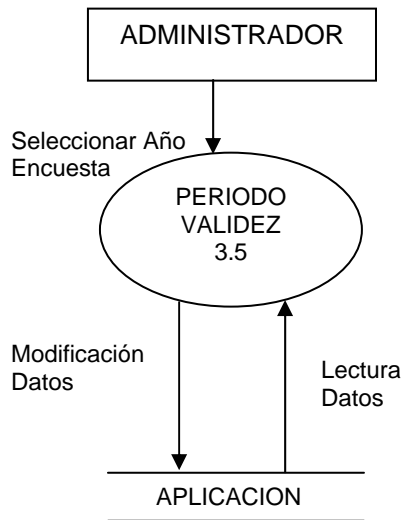
Se repite el mismo esquema para las opciones de Añadir/Modificar Área y Pregunta.



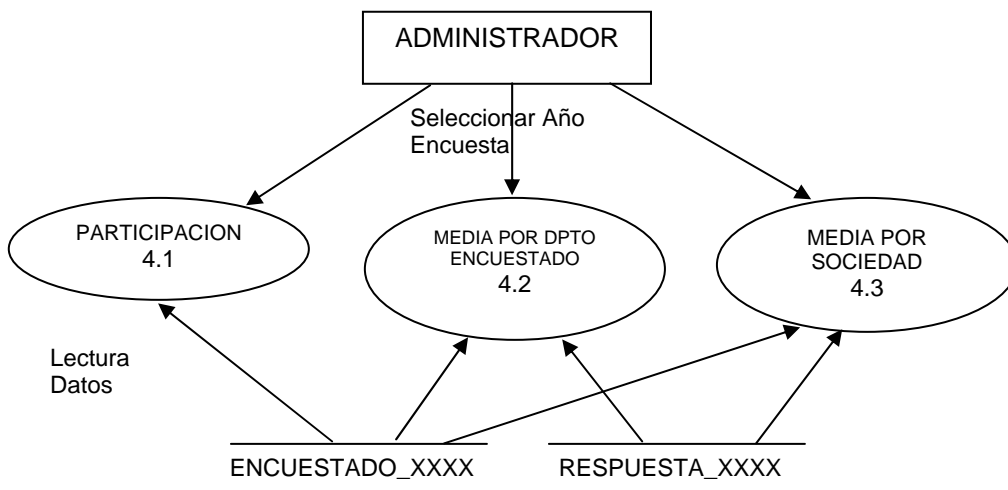
El sufijo \_XXXX en las tablas creadas indica el año al que hace referencia la encuesta. De esta forma se consigue mantener un histórico de las valoraciones obtenidas, de datos sobre los encuestados y del diseño del formulario de años anteriores.







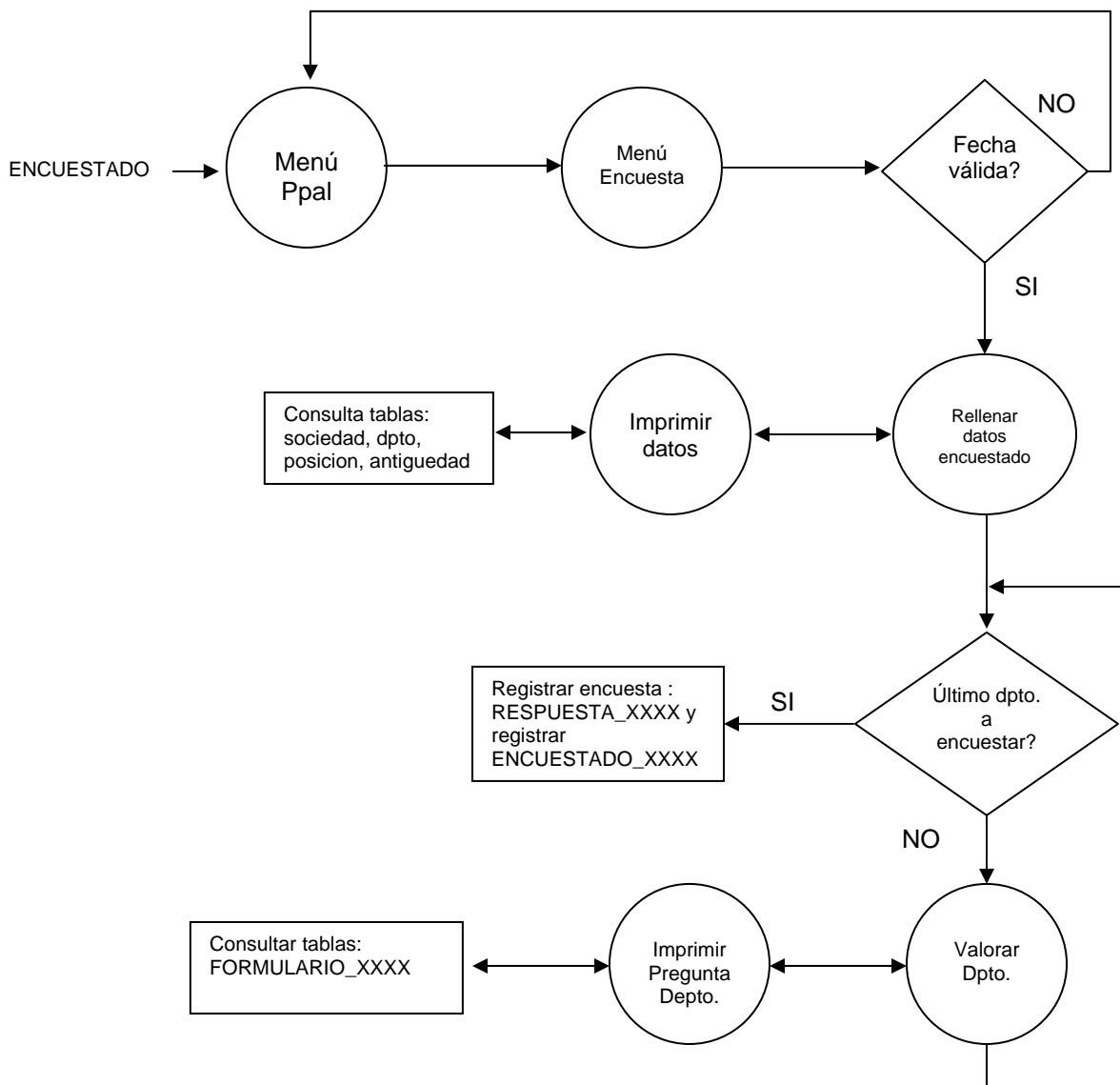
**4.5.- DFD 4: Proceso Consultar Estadísticas – Nivel 2**



## 5.- Especificaciones de la aplicación

### 5.1.- Proceso Realizar Encuesta

Cualquier persona dispondrá de la opción que le permitirá realizar la encuesta. El proceso será guiado: en primer lugar se cumplimentarán sus datos (sociedad, departamento, antigüedad y posición) y a continuación se valorará uno por uno todos los departamentos que le prestan servicios. La encuesta será grabada en la base de datos únicamente cuando el proceso sea finalizado. Si en la mitad del proceso se produce una cancelación la encuesta registrada no será válida.

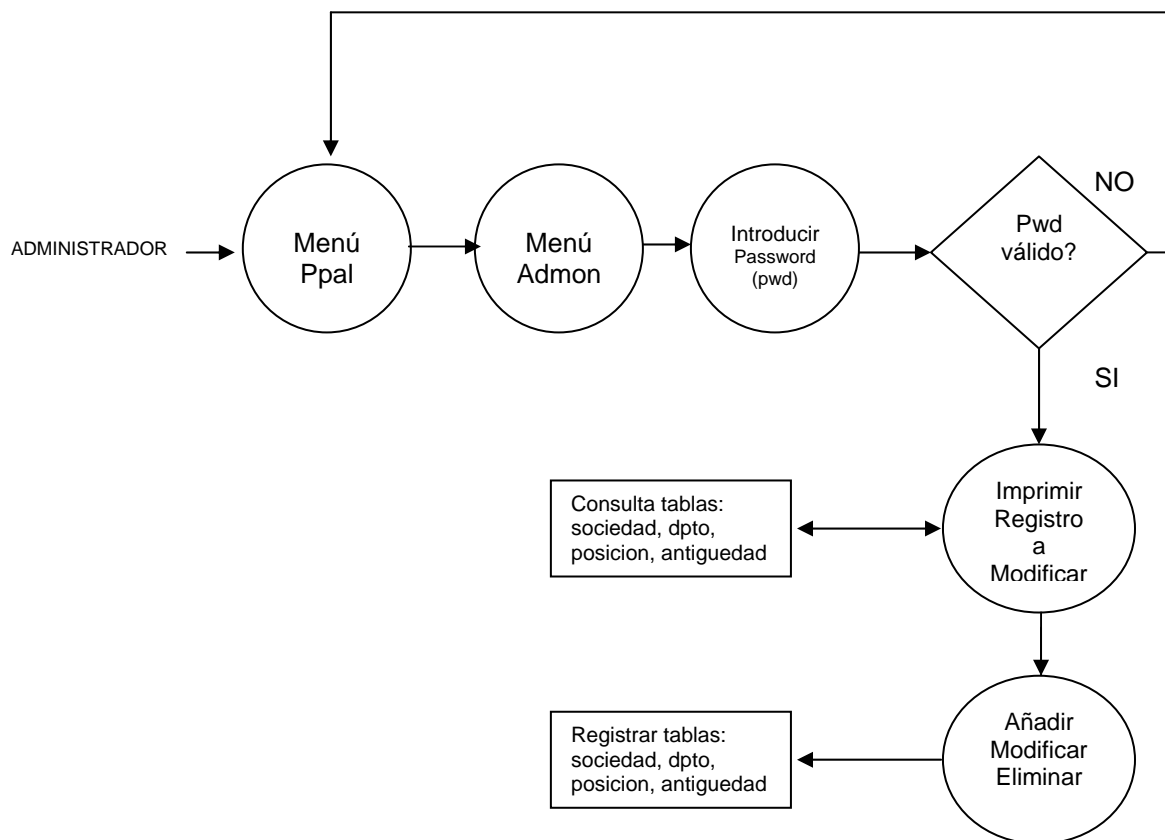


### 5.2.- Proceso Mantenimiento de los Datos del Encuestado

Las cuatro tablas implicadas son independientes entre sí, por lo que no existen relaciones entre ellas. Las opciones que se presentarán serán poder añadir un nuevo registro, modificar uno existente y eliminar registro.

La opción eliminar, no borra el registro de la tabla, únicamente lo desactiva para su no impresión. La eliminación de un registro implicaría la no coherencia en formularios de años anteriores que hiciesen referencia a este registro.

Las operaciones con las tablas SOCIEDAD, DEPARTAMENTO, POSICION y ANTIGÜEDAD seguirán el mismo procedimiento.

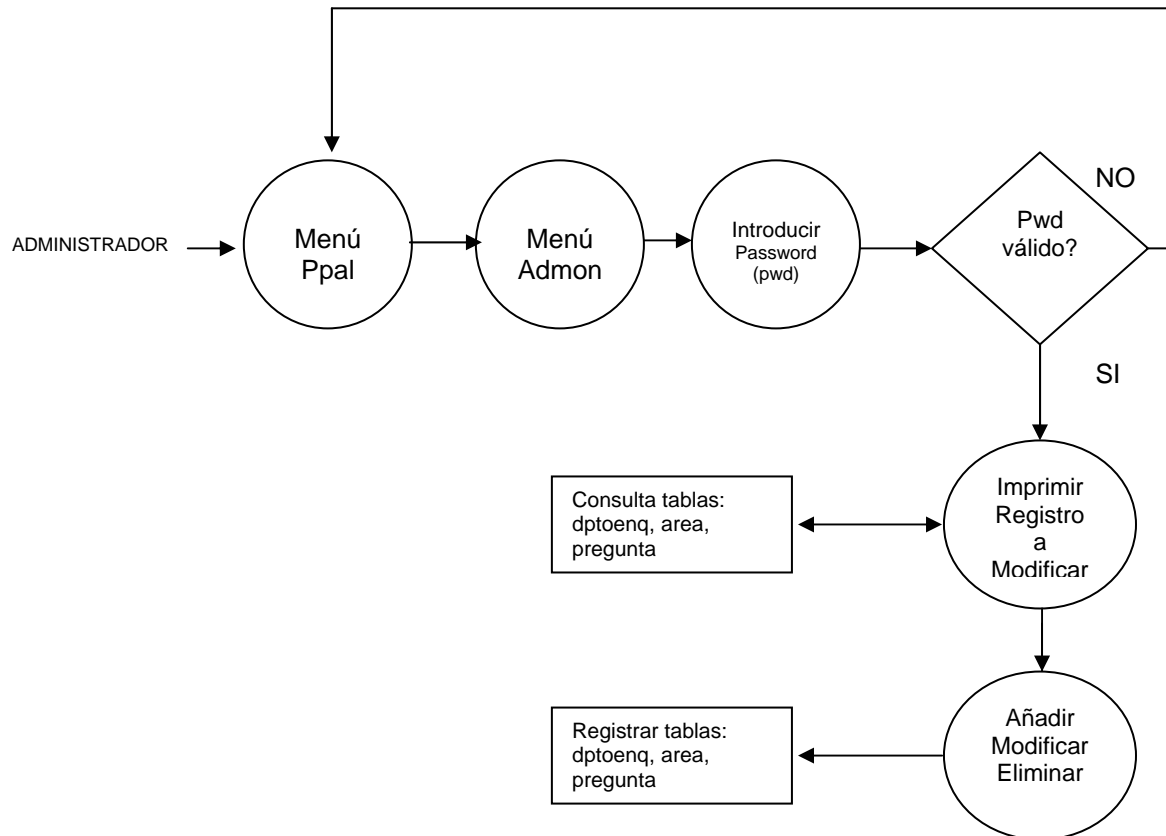


La opción eliminar no borrará el registro de la tabla si no que únicamente quedará desactivado (atributo activado a falso).

### 5.3.- Proceso Mantenimiento de los Datos de la Encuesta

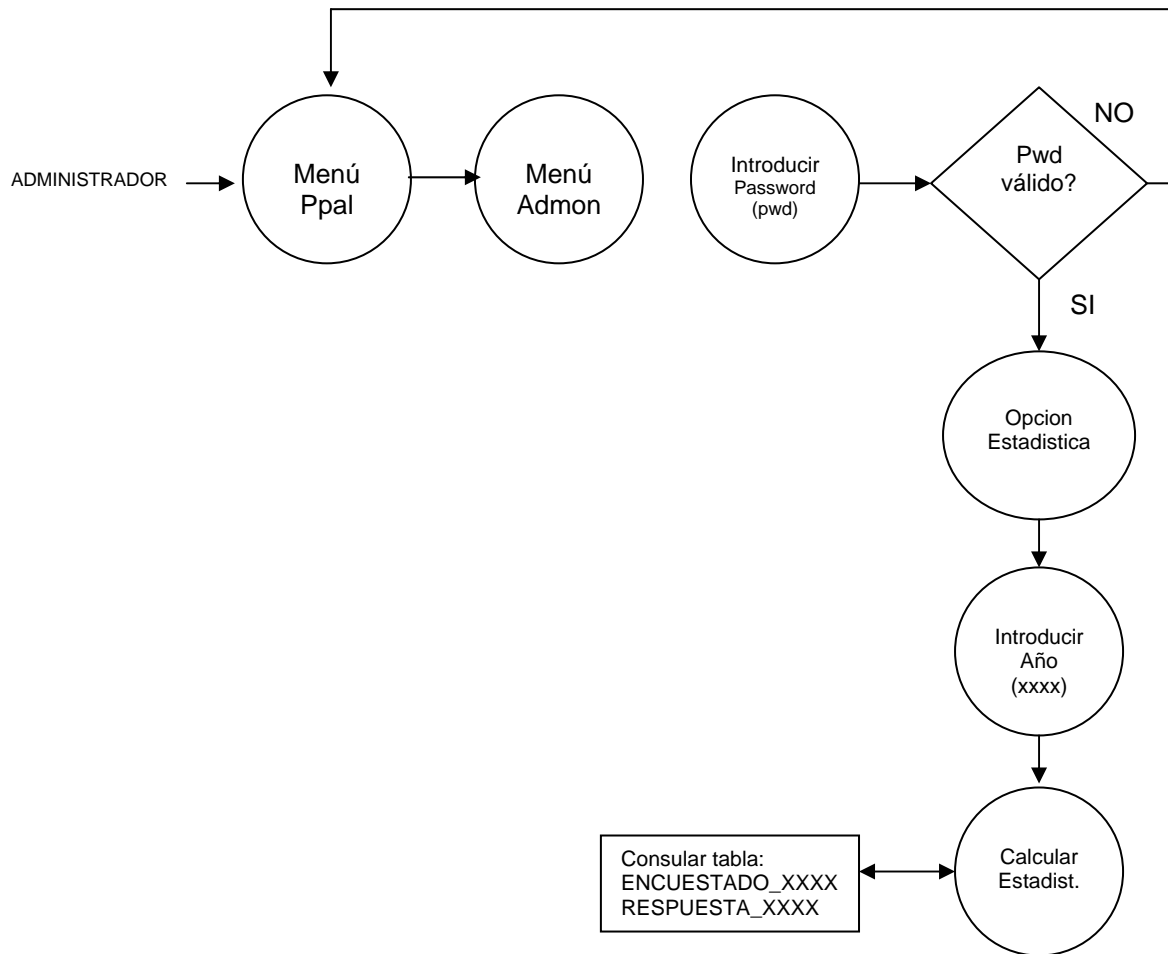
Se permiten las mismas opciones que en el caso anterior. Hay que tener en cuenta que la opción modificar se refiere a modificaciones referentes a mala escritura o falta de información, en ningún caso cambio de sentido de la pregunta porque podría desvirtuar el sentido de encuestas anteriores.

Al igual que en el caso anterior las tablas implicadas son independientes entre sí en el diseño aunque lógicamente sí que tienen relación. La relación entre las tablas DPTOENQ, AREA y PREGUNTA viene reflejada en la tabla FORMULARIO.



#### 5.4.- Proceso Consulta Estadísticas

Los procedimientos referentes a estadísticas se abordarán de nuevo en un futuro para poder evaluar los pros y contras que puede conllevar.



En la tabla encuestado se guardará un registro por cada persona que realiza la encuesta, nos permitirá saber el nº total de encuestas contestadas.

## 6.- Interficie gráfica

La interficie gráfica de usuario se realizará con páginas HTML. En los apartados anteriores se ha tratado el funcionamiento del programa desde un punto de vista interno, cuando se habla de interficie gráfica se trata de aquellos efectos del funcionamiento que son visibles para el usuario.

No existen requisitos en el diseño de la interficie, durante la fase de implementación se irán realizando los cambios oportunos en el diseño.

# IMPLEMENTACIÓN

## 1.- Introducción

La solución con la que se ha desarrollado la aplicación ha sido la basada en arquitectura Microsoft (programación páginas *Active Server* - ASPs) tal y como se determinó en la fase de análisis y diseño.

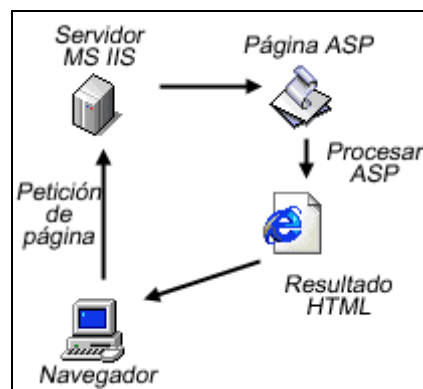
ASP es una tecnología que se basa en los lenguajes de *Script*, que van insertados dentro de páginas HTML, y se utiliza para realizar contenidos dinámicos en Internet. Por lo tanto, dentro de una página HTML, se va a poder acceder de forma on-line a una base de datos, consultar los datos que se necesitan y mostrarlos al usuario como HTML.

ASP no necesita compilación, es un lenguaje que se interpreta cada vez que se realiza una petición al servidor de una de sus páginas. La desventaja que presenta es que al ser un lenguaje interpretado, se gasta más tiempo que si fuera compilado ya que el servidor ha de interpretar la página cada vez que se realiza una petición.

De todas formas, este tiempo se puede considerar insignificante comparado con el tiempo que va a utilizar en otras aplicaciones, como puede ser el acceso a bases de datos.

Las páginas ASP comienzan a ejecutarse cuando un usuario solicita un archivo *.asp* al servidor Web a través del explorador. El servidor web llama a ASP, que lee el archivo solicitado, ejecuta las secuencias de comandos que encuentra y envía los resultados al explorador del cliente.

Puesto que las secuencias de comandos se ejecutan en el servidor, y no en el cliente, es el servidor el que hace todo el trabajo necesario para generar las páginas que se envían al explorador. Las secuencias de comando quedan ocultas a los usuarios, estos sólo reciben el resultado de la ejecución en formato HTML.



**Fig. I** Esquema ejecución página ASP

De esta forma desaparece el problema de si el cliente puede o no ejecutar sentencias de comandos, el servidor Web sólo envía el resultado en código HTML estándar interpretable por cualquier explorador.

## 2.- *Scripts de Servidor*

Tal y como se ha comentado en el apartado anterior, una página ASP mezcla en el mismo archivo secuencias de comandos con código HTML estándar. Las secuencias de comando *asp* se distinguen del resto del texto del archivo mediante delimitadores.

En el caso de HTML, dichos delimitadores son los símbolos `<` y `>` que enmarcan las etiquetas html. ASP utiliza los delimitadores `<%` y `%>` para enmarcar las secuencias de comando.

ASP viene de forma predeterminada con dos motores de secuencia de comandos, *Microsoft Visual Basic Script* (VBScript) y *Micorosot Jscript*. De todas formas, se pueden instalar y utilizar motores de otros lenguajes como Perl.

VBScript únicamente se ejecuta en el navegador de Microsoft. Esto lo hace menos universal que JavaScript, que es capaz de ejecutarse en los dos navegadores, aunque es difícil dar con un desarrollo en JavaScript que funcione correctamente en los dos sistemas.

Para establecer el lenguaje principal de secuencia de comandos en todas las páginas de una aplicación, se ha de establecer la propiedad Lenguaje ASP predeterminado en la configuración del servidor web.

Para establecer el lenguaje principal de secuencia de comandos en una única página, hay que incluir la siguiente línea al inicio del archivo *asp*.

```
<% @LANGUAGE = VBScript %>
<% @LANGUAGE = JScript %>
```

## 3.- *Objetos integrados de ASP*

Los cinco objetos que el servidor ofrece para un mayor control sobre aspectos de la comunicación cliente-servidor y sobre el funcionamiento del sistema son los siguientes:

- *Objeto Application*: Se utiliza para compartir información entre todos los usuarios de una aplicación
- *Objeto Request*: Se utiliza para tener acceso a la información que se pasa en las peticiones http. En esta información se incluyen los parámetros que se pasan desde los formularios HTML mediante el método POST o el método GET, *cookies* y certificados del cliente.

- Objeto Response: Se utiliza para controlar la información que se envía al usuario. Esto incluye el envío de información directamente al explorador, las redirección del explorador a otra dirección URL o el establecimiento de valores de las *cookies*.
- Objeto Server: Proporciona acceso a los métodos y la propiedades del servidor.
- Objeto Session: Permite almacenar la información necesaria para una determinada sesión de usuario. Las variables almacenadas en el objeto Session persisten durante todo el tiempo que el usuario tiene acceso a las páginas de la aplicación.

#### **4.- Entorno de trabajo**

Para el desarrollo de la aplicación y para la ejecución de páginas *Active Server* se han tenido en cuenta los siguientes puntos:

##### **4.1.- Servidor Web**

Para la implantación de un servidor Web que soporte ASP, el software necesario depende del sistema operativo que está instalado en la máquina.

- Windows 2000 → Internet Information Server (IIS) 5.0
- Windows NT → Internet Information Server (IIS) 4.0 o IIS 3.0 + ASP.exe
- Windows 9x → Personal Web Server (PWS) 4.0 o PWS 1.0 + ASP.exe

La aplicación de las encuestas se ha configurado en un servidor Windows NT Server 4.0 y el servidor Web utilizado es el IIS 4.0.

La instalación es relativamente sencilla y se realiza por medio de un asistente de instalación. Para acceder a la aplicación se ha creado un directorio virtual llamado encuestas. Un directorio virtual actúa como un alias de un directorio real, es decir, es un directorio de nuestro disco duro que se asigna a un directorio de Internet.

El acceso a la aplicación se realiza desde el explorador con la siguiente sentencia:

`http://www.finaf92.es/encuestas`

La seguridad de un sitio Web es especialmente importante, se ha de garantizar la correcta utilización de los usuarios remotos. Para poder ejecutar páginas ASP en el servidor es necesario establecer permisos de lectura y ejecución en el directorio donde se encuentre el código de las páginas.

IIS 4.0 utiliza la seguridad de Windows NT por lo que se ha reforzado la seguridad mediante permisos NTFS. Desde IIS se pueden establecer permisos de Lectura y Ejecución y desde NT cualquier permiso implementado en NTFS. En caso de discrepancia se toma la opción más restrictiva.



## 4.2.- Entorno de programación

Los archivos *asp* son archivos de texto normales, no es necesario ningún editor especial para crearlos, puede utilizarse cualquier editor que genere código ASCII.

De todas formas, aunque no es necesario conviene utilizar un editor que sea suficientemente cómodo y productivo.

Para el desarrollo de las páginas de la aplicación se ha utilizado el programa *Visual InterDev 6.0*, un producto creado por *Microsoft* dentro del paquete de *Visual Studio 6.0*.

La razón de haber utilizado este *software*, y no otro de las múltiples opciones que hay, es la facilidad que ofrece en la programación de páginas ASP. Reconoce el código de servidor introducido entre los delimitadores, tanto VBScript como JScript, y proporciona ayuda en tiempo de diseño. Por ejemplo, al escribir objetos ASP que disponen de métodos y propiedades aparece un desplegable con todas las opciones disponibles para ese objeto.

Permite generar proyectos para trabajar de forma más cómoda y organizada y desde la pantalla principal permite visualizar el explorador de proyectos por lo que el acceso al código de nuestras páginas es cómodo y sencillo.

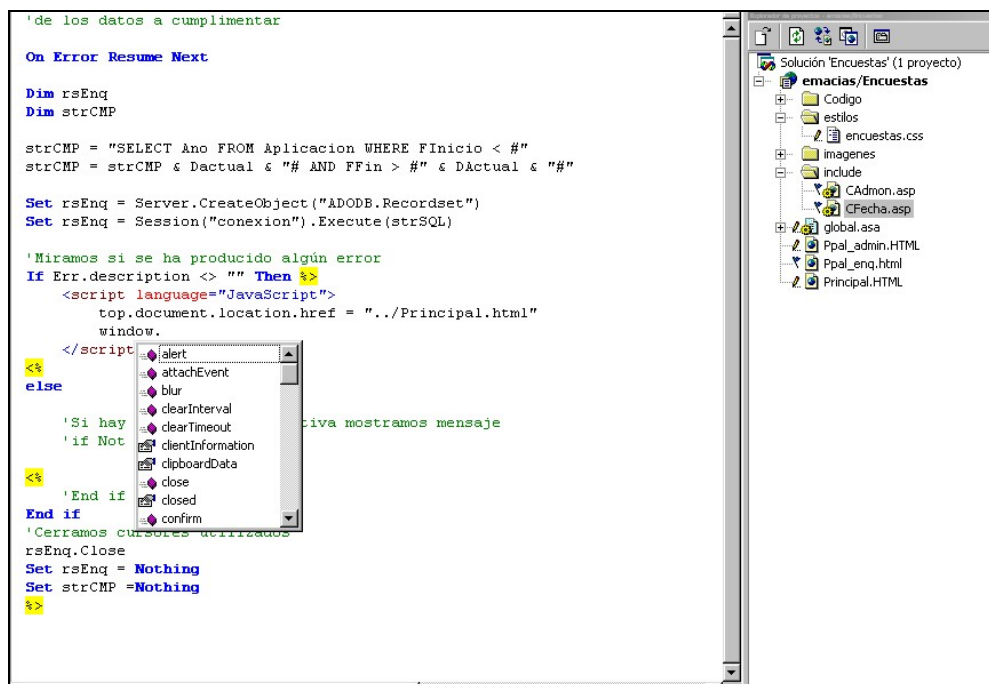


Fig. II Entorno de programación *Microsoft Visual InterDev 6.0*

### 4.3.- Fuentes de datos ODBC

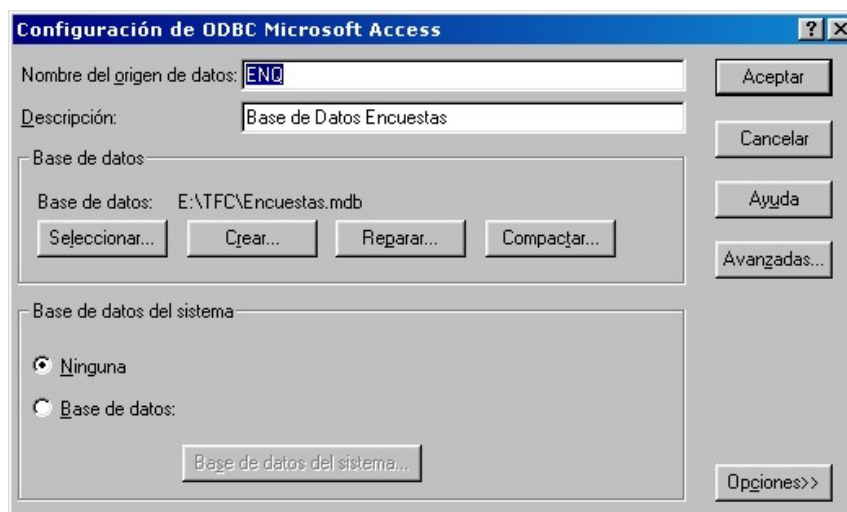
El sistema ODBC (Open DataBase Connectivity) es un sistema estructurado de “filtros” mediante los cuales podemos comunicarnos y extraer datos de cualquier base de datos sin preocuparnos de qué sistema la gestiona.

Lo único que se necesita para comunicarnos con una de estas bases de datos, es tener instalado en el sistema el correspondiente controlador ODBC.

Cuando se dispone de una base de datos y del controlador ODBC correspondiente, lo que en realidad se tiene es un fichero que contiene una serie de datos a los que no se pueden acceder de forma directa, y un filtro que es el que permite acceder a ellos en un lenguaje común para todas las bases de datos. Este lenguaje es el SQL, es el más difundido de consulta a base de datos.

Para crear una conexión ODBC a Access, una vez se tiene el fichero de base de datos guardado en el disco duro, se ha de hacer doble *click* en el icono ODBC que se halla en el panel de control.

Se ha de crear un controlador ODBC de sistema, para que lo pueda utilizar el *Internet Information Server* sin problemas, elegir la opción Microsoft Access Driver y pulsar el botón finalizar.



**Fig. III** Configuración ODBC Microsoft Access

En el cuadro de diálogo que aparece se han de rellenar los siguientes campos:

- Nombre de origen de datos: Nombre con el que se identificará a la base de datos
- Descripción: Descripción de la base de datos (sin utilidad práctica)
- Base de datos: Base de datos que contendrá los datos
- Base de datos del sistema: Si la base de datos tiene implementado un sistema de seguridad se necesitará especificar cuál es su base de datos de

sistema. Si no la tiene, lo más normal, por defecto se marcará la opción Ninguna

Una vez cumplimentados todos los datos se tendrá una conexión ODBC con las base de datos Access, identificada por el nombre que se le dio en el campo Data Source Name.

#### 4.4.- Lenguaje de consulta estructurado (SQL)

El lenguaje de consulta estructurado (SQL) es un lenguaje de base de datos normalizado, utilizado por el motor de base de datos de Microsoft Jet.

El lenguaje SQL está compuesto por comandos, cláusulas, operadores y funciones de agregado. Estos elementos se combinan en las instrucciones para crear, actualizar y manipular las bases de datos.

Existen dos tipos de comando SQL:

- DLL: Permiten crear y definir nuevas bases de datos, campos e índices.

Comando	Descripción
CREATE	Utilizado para crear nuevas tablas, campos e índices
DROP	Empleado para eliminar tablas e índices
ALTER	Utilizado para modificar las tablas agregando campos o cambiando las definición de los campos

**Tabla I Comandos DLL**

- DML: Permiten generar consultas para ordenar, filtrar y extraer datos de la base de datos

Comando	Descripción
SELECT	Utilizado para consultar registros de la base de datos que satisfagan un criterio determinado
INSERT	Utilizado para cargar lotes de datos en la base de datos en una única operación
UPDATE	Utilizado para modificar los valores de los campos y registros especificados
DELETE	Utilizado para eliminar registros de una tabla de una base de datos

**Tabla II Comandos DML**

#### 4.5.- Componentes de acceso a datos

Para poder acceder a la base de datos de la cual se ha creado el correspondiente nombre de origen de datos (DSN) y poder manipular su información, es necesario utilizar una conexión en la página ASP desde la que se pretenda acceder a la base de datos. Esto se consigue mediante el uso de ADO (*ActiveX Data Object*).

ADO utiliza ODBC para el acceso a base de datos, independizando la aplicación de la tecnología de la base de datos. Esto implica que si se cambia la tecnología y se mantiene la misma estructura de datos, las aplicaciones desarrolladas con ADO seguirán funcionando sin cambiar código de las páginas.

ADO está formado por varios objetos organizados de forma jerárquica, cada uno con sus métodos y propiedades específicos.

- Connection: Proporciona una conexión a una base de datos ODBC desde una página ASP. Ninguno de los otros objetos puede existir si este no se ha declarado.
- Error: Colección en la que se almacenarán los posibles errores del objeto.
- Command: Representa un comando SQL que se ejecuta contra la base de datos declarada en el objeto *Connection*. Si el resultado de ese comando es un conjunto de datos, estos se almacenarán en un objeto de tipo *RecordSet*.
- RecordSet: Representa una tabla o el resultado de una consulta ejecutada contra la base de datos. Como en todo modelo relacional, los datos se presentarán en filas y columnas.
- Field: Representa la información relativa a un campo de un *Recordset*.

## **5.- Diseño de la aplicación**

### **5.1.- Introducción**

La aplicación consta básicamente de dos áreas bien diferenciadas, el área de realización de la encuesta y el referente a la gestión y mantenimiento de la misma.

Tal y como se especificó en la fase de análisis de requisitos el administrador es la persona encargada de realizar la gestión y mantenimiento de la encuesta y para ello es necesario la autenticación ante el sistema mediante un *password*.

Desde la pantalla principal de la aplicación se puede acceder a las dos áreas. Por un lado se dispone de una *link* para la realización de la encuesta y por otro lado se dispone de una petición de usuario y contraseña para la realización de la gestión.

Se ha de garantizar que para acceder a cualquier página de administración sea necesario, previamente, la autenticación ante el sistema. Para asegurar este criterio se utilizan variables de sesión.

Existe la posibilidad de que una persona se aprendiese la dirección URL de una página de gestión de la encuesta y accediese a ella directamente sin necesidad de validarse ante el sistema.

Para solucionar este problema se utilizan variables de sesión. Una vez el administrador introduce el nombre de usuario y contraseña, el sistema comprueba si los datos introducidos son correctos. Si la respuesta es afirmativa, se declara una variable de sesión con el usuario introducido.

```
'Guardamos en una variable de sesión el usuario
Session("Usuario") = Request.Form("Login")
```

Al inicio de todas las página ASP, donde únicamente puede acceder el administrador, incluimos el siguiente fichero:

```
<!-- #include file = "../include/CAdmon.asp" -->
```

De esta forma antes de ejecutar el código de la página ASP ejecuta el código del fichero *Cadmon.ASP*. En este fichero se comprueba si existe una variable de sesión llamada Usuario; si existe se continúa ejecutando el código de la página correspondiente y si, por el contrario, no existe se redirige la aplicación a la página de inicio.

Esta medida de seguridad se realiza en el nivel de aplicación pero no en el nivel de red. Los datos introducidos por el usuario, desde el explorador cliente, viajan por la red hasta el servidor sin ningún tipo de protección. Para evitar la interceptación de estos datos se podría estudiar la posibilidad de utilizar una conexión segura, mediante el protocolo S-HTTP (*Secure HTTP*).

La recogida de datos del cliente se realiza por medio de formularios HTML en los que se pone a disposición del usuario una serie de elementos en los que se puede escribir o seleccionar los datos que se le pide.

La sintaxis del método FORM es la siguiente:

```
<FORM>
  NAME = "nombre_Formulario"
  TARGET="nombre_Ventana"
  ACTION="URL_destino"
  METHOD=GET ó POST
  ENCTYPE="Tipo_codificación"
  [onSubmit="Codigo_JavaScript"]
</FORM>
```

- **Action:** Representa la dirección URL a la que llama el formulario al hacer un *submit* (enviar datos).
- **Target:** Nombre de la ventana donde se ejecutará la página a la que se llama al hacer el *submit*.
- **Name:** Nombre del formulario
- **Method:** El método utilizado para la transmisión de los datos introducido puede ser GET O POST.
- **OnSubmit:** Se produce cuando se pulsa el botón se *submit* del formulario.

Después de crear el formulario HTML se tendrá que procesar los datos proporcionados por el usuario, lo que implica enviar la información a un archivo

.asp para que la analice y manipule. El archivo al que se envía la información se indica en la propiedad ACTION.

```
<form method = "POST" action="codigo/Login.asp" name="Login"
onSubmit="javascript: return Validar();" >
  <table align="right" cellpadding="0" cellspacing="0">
    <tr>
      <td align="right" class="textoLogin">LOGIN:&nbsp;&nbsp;&nbsp;</td>
      <td><input type="text" name="login" class="textoLogin"></td>
    </tr>
    <tr>
      <td align="right" class="textoLogin">PASSWORD:&nbsp;&nbsp;&nbsp;</td>
      <td><input type="password" name="pwd" class="textoLogin"></td>
    </tr>
    <tr>
      <td colspan=2><INPUT type="submit" value="Entrar" name=Admon> </td>
    </tr>
  </table>
</form>
```

**Fig. IV** Ejemplo formulario

El objeto *Request* de ASP proporciona dos colecciones que facilitan la tarea de recuperar información de los formularios enviados con las peticiones URL.

La colección *QueryString* recupera los valores del formulario pasados al servidor Web como texto a continuación del signo de interrogación de la dirección URL de la petición. Los valores del formulario se pueden anexar a la dirección URL de la petición mediante el método GET de HTTP o, manualmente, si se agregan los valores del formulario a la dirección URL.

```
Login.asp?login=nomusuario&password=password
Request.QueryString("login")
Request.QueryString("password")
```

**Fig. V** Envío/Recogida Información método GET

La colección *Form* del objeto *Request* de ASP recupera los valores enviados por el método POST. El método POST, que envía los datos de los formularios en el cuerpo de la petición HTTP, puede enviar un número casi ilimitado de caracteres a un servidor.

```
Request.Form("login")
Request.Form("password")
```

**Fig. VI** Recogida Información método POST

Por último, antes de enviar la información al servidor es conveniente realizar las comprobaciones posibles en el cliente para evitar operaciones en el servidor. Las validaciones se pueden realizar con código JavaScript al producirse el evento *onSubmit*, de tal forma que si la validación es correcta se envía la información del formulario y si no es correcta se indica al usuario.

```

<script language="JavaScript">
  //Función para validar que se halla introducido login y password
  function Validar() {
    if ((document.Login.login.value == "") ||
        (document.Login.password.value == "")) {
      alert("Se ha de indicar Usuario y Password");
      return(false);
    }
    return(true);
  }
</script>

```

**Fig. VII** Validación de un formulario

Finalmente, para facilitar el diseño de las páginas se han utilizado Hojas de Estilo. Los estilos utilizados se pueden definir de tres formas diferentes:

- Directamente sobre el tag con el modificador Style

```
<FONT STYLE="font-family=Arial;font-size=11">Texto</FONT>
```

- Definiendo los estilos en la página HTML

```

<HEAD>
<STYLE TYPE="text/css">
  .txt_titulo {font-family:Arial, Verdana;font-size:3px}
  .txt_tachado {text-decoration:line-through}
</STYLE>
</HEAD>

<BODY>
  <FONT CLASS="txt_titulo">Este es el titulo</FONT><BR><BR>
  <FONT CLASS="txt_tachado">Este texto aparecerá tachado</FONT>
</BODY>

```

- Definiendo los estilos en un fichero de estilos

Este es el método que se ha utilizado ya que realizando modificaciones únicamente en este fichero, los cambios son aplicados a todas las páginas de la aplicación.

En la cabecera de las páginas se ha de incluir la siguiente línea:

```

<HEAD>
<LINK rel="stylesheet" type="text/css" href="../estilos/encuestas.css">
</HEAD>

<BODY>
  <FONT CLASS="txt_titulo">Este es el titulo</FONT><BR><BR>
  <FONT CLASS="txt_tachado">Este texto aparecerá tachado</FONT>
</BODY>

```

La hoja de estilos utilizada se ha implementado, al igual que el resto de código, con Microsoft Visual Interdev. Permite crear clases de una forma rápida, cómoda y sencilla.

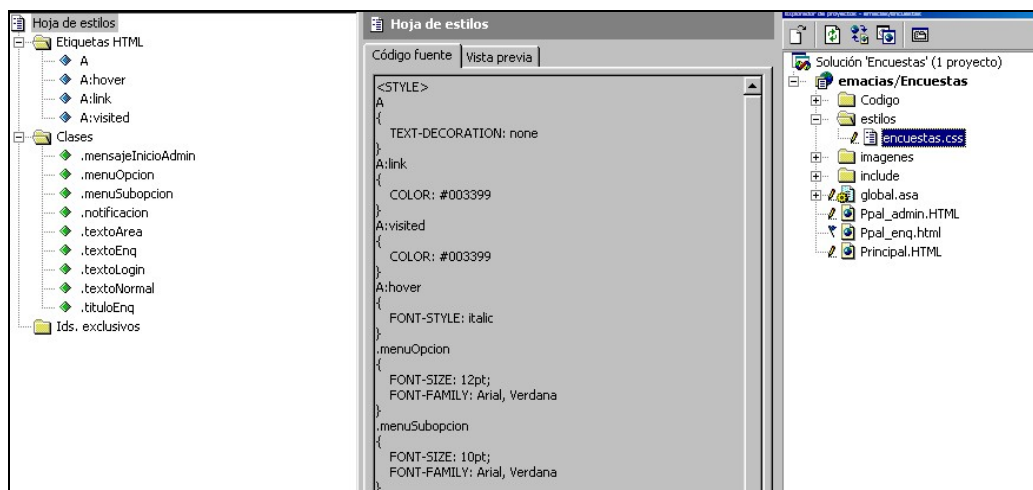


Fig. VIII Hoja de Estilos con Visual Interdev

## 5.2.- Estructura de la aplicación

A continuación se describe la estructura de páginas que contiene la aplicación, la relación existente entre ellas y una breve descripción de qué se ha implementado en cada una.

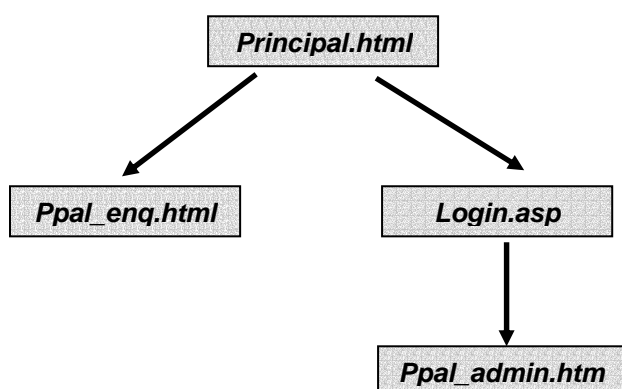


Fig. IX Bloque inicial de la aplicación

### 1. *Principal.html*

Página principal de la aplicación. Permite seleccionar la opción de realizar la encuesta anual o bien, en el caso del administrador, introducir el usuario y el *password* para validarse ante el sistema.

### 2. *Ppal\_enq.html*

Página que muestra la cabecera de la aplicación (logotipo y nombre de las sociedades que forman el grupo) y el inicio de la encuesta con ayuda de un *frame*.



### 3. *Login.asp*

Comprueba si el usuario y password introducidos son correctos.

Realiza una conexión con la base de datos y se guarda el parámetro conexión como una variable de sesión.

```
Set Session("conexion")= Server.CreateObject("ADODB.Connection")
Session("conexion").Open="DSN=ENQ;UID=' ';PWD=' '"
```

Comprueba si los datos introducidos son correctos. Si la respuesta es afirmativa, lanza la página *Ppal\_Admin.html*.

```
<script language="JavaScript">
    top.document.location.href = "../Ppal_Admin.html"
</script>
```

En caso contrario se muestra un mensaje de error, se abandona la sesión (destruye todos los objetos y variables almacenadas en el objeto Session) y vuelve a la página *Principal.html*.

### 4. *Ppal\_admin.html*

Inicia el módulo de administración de la encuesta y permite acceder a todas las funcionalidades de gestión.

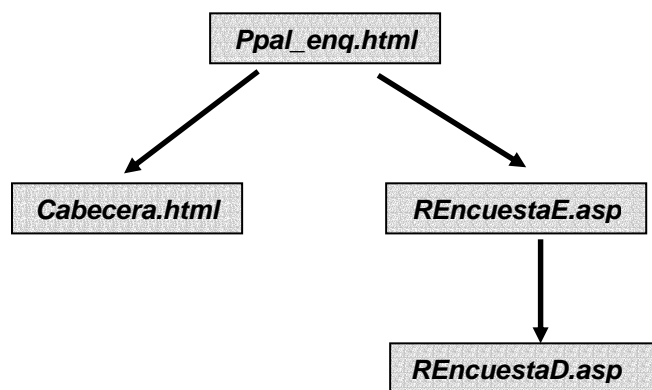


Fig. X Bloque realización encuesta

### 5. *RencuestaE.asp*

Inicio de la realización de la encuesta.

Realiza una conexión con la base de datos y se guarda el parámetro conexión como una variable de sesión. Este parámetro estará disponible mientras dure la realización de la encuesta.

En primer lugar, busca cuál es la encuesta que está activa. Por el diseño de la aplicación, únicamente podrá haber una encuesta activa por

periodo, que será aquella en la que la fecha actual es mayor que su fecha de inicio y menor que la fecha final de la aplicación. Si no existe ninguna encuesta que cumpla estos requisitos se notifica que no hay encuestas activas.

En esta página se cumplimentan los datos referentes al encuestado y la información obtenida es enviada a la página *RencuestaD.asp*.

Para mostrar la opciones posibles se selecciona de la base de datos, las sociedades, departamentos, posiciones y antigüedades que están activas y se muestran por pantalla.

## **6. *RencuestaD.asp***

Valoraciones de todos los departamentos pendientes de encuestar.

Si se accede a la página por primera vez, justo después de cumplimentar los datos del encuestado, el sistema almacena la información recibida y muestra por pantalla la cuestiones referentes al primer departamento a encuestar.

Las valoraciones introducidas serán reenviadas a la misma página que almacenará los datos recibidos y buscará e imprimirá por pantalla las preguntas del siguiente departamento.

Cuando se llega al último departamento se muestra un mensaje indicando que el proceso ha finalizado.

Es posible que la realización de la encuesta se suspenda en mitad del procesos, por ejemplo, el usuario cierre el navegador. Esta encuesta no debería contar ya que el proceso ha sido interrumpido.

Para prevenir el caso anterior, al almacenar los datos referentes al encuestado se actualiza el campo valido a *false*. Si todo el proceso ha sido correcto, el sistema actualiza el valor de este campo a *true*.

Es posible que la tabla que almacena los datos referentes a las respuesta crezca debido a las encuestas que no han finalizado correctamente. Es por esto, que en el módulo de administración se ha habilitado una opción para purgar los registros no válidos.

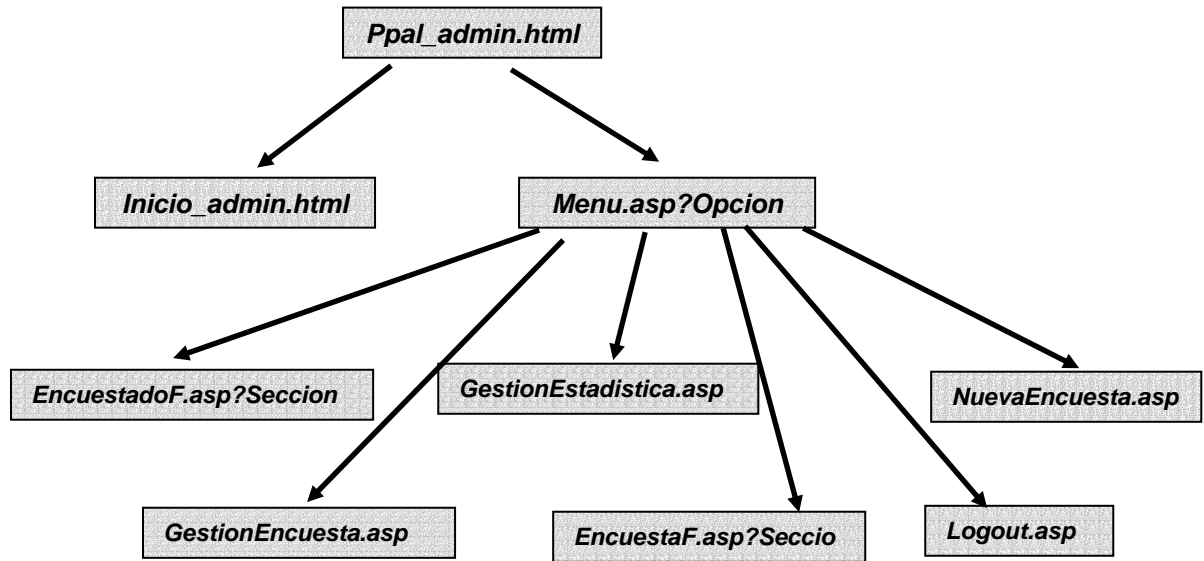


Fig. XI Bloque modulo administración

### 7. Inicio\_admin.html

Muestra un mensaje indicando que se ha iniciado una sesión con el módulo de administración. En el fichero *Login.asp* se crea la variable de sesión que contiene el parámetro de conexión y no se libera hasta que no se finaliza este modulo.

### 8. Menu.asp

Muestra las opciones disponibles del area de gestión. La opción seleccionada se reenvía a la propia página mediante el método GET del formulario.

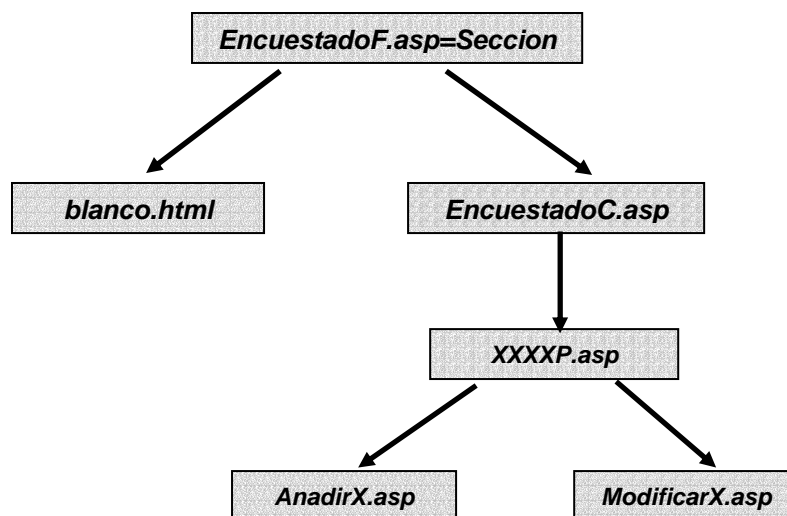


Fig. XII Bloque datos encuestado

### **9. *EncuestadoF.asp***

Frame que muestra la página *EncuestadoC.asp* y *blanco.html*.  
En la variable de sesión Sección guardará la opción pulsada en el menú.

### **10. *Blanco.html***

Página en blanco

### **11. *EncuestadoC.asp***

Dependiendo del valor de la variable de sesión Sección, muestra las sociedades, departamentos, posiciones o antigüedades que están almacenadas en la base de datos.

Muestra en cada lista de color negro las que están activas y de color gris las que están desactivadas. Las opciones que se permiten seleccionar son Añadir, Eliminar y Modificar.

### **12. *XXXXP.asp***

Existe una página para cada tabla: *SociedadP.asp*, *DepartamentoP.asp*, *PosicionP.asp* y *AntigüedadP.asp*.

En esta página se muestran los datos a cumplimentar dependiendo de la opción pulsada. Si se ha seleccionado añadir se muestra un formulario con los datos a cumplimentar y un botón Añadir que envía la información a la página AnadirX (AnadirS, AnadirD, AnadirP, AnadirA dependiendo de la variable sesión).

Si por el contrario se ha seleccionado Modificar, se muestra un formulario con los actuales datos y se ofrece la posibilidad de cambiar. Los resultados son enviados a la página *ModificarX.asp* dependiendo de la variable Sección.

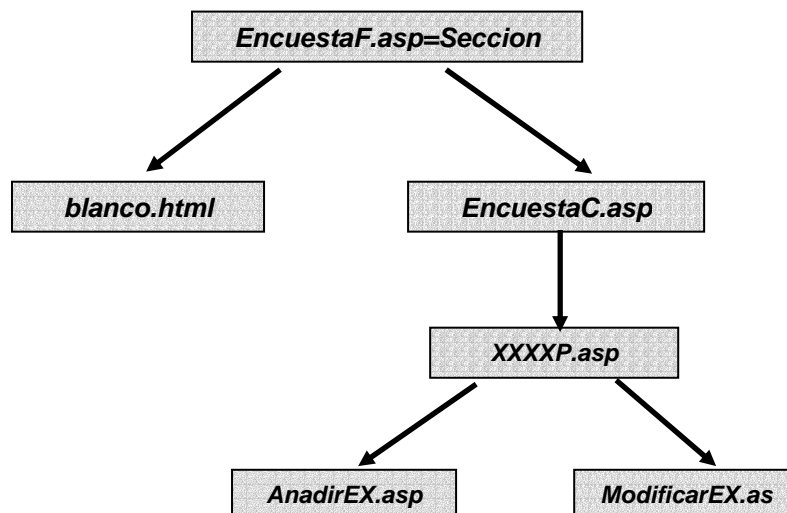
Antes de enviar los datos del formulario siempre se realizarán validaciones básicas como que todo los campos estén cumplimentados, no permitir cadenas vacías,...

### **13. *AnadirX.asp***

Añade un nuevo registro a la base de datos.

### **14. *ModificarX.asp***

Modifica un registro de la base de datos.



**Fig. XIII** Bloque datos encuesta

### 15. EncuestaF.asp

Frame que muestra la página *EncuestaC.asp* y *blanco.html*.

En la variable de sesión SEncuesta guardará la opción pulsada en el menú.

### 16. EncuestaC.asp

Dependiendo del valor de la variable de sesión SEncuesta, muestra los departamentos a encuestar, areas o preguntas que están almacenadas en la base de datos.

Muestra en cada lista de color negro las que están activas y de color gris las que están desactivadas. Las opciones que se permiten seleccionar son Añadir y Modificar.

### 17. XXXXP.asp

Existe una página para cada tabla: *DptoEnqP.asp*, *AreaP.asp* y *PreguntaP.asp*.

En esta página se muestran los datos a cumplimentar dependiendo de la opción pulsada. Si se ha seleccionado añadir se muestra un formulario con los datos a cumplimentar y un botón Añadir que envía la información a la página AnadirX (AnadirED, AnadirEA, AnadirEP dependiendo de la variable sesión).

Si por el contrario se ha seleccionado Modificar, se muestra un formulario con los actuales datos y se ofrece la posibilidad de cambiar. Los resultados son enviados a la página *ModificarEX.asp* dependiendo de la variable SEncuesta.

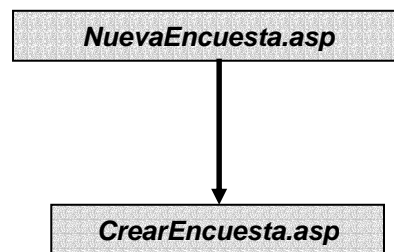
Antes de enviar los datos del formulario siempre se realizarán validaciones básicas como que todo los campos estén cumplimentados, no permitir cadenas vacías,...

### **18. AnadirEX.asp**

Añade un nuevo registro a la base de datos.

### **19. ModificarEX.asp**

Modifica un registro de la base de datos.



**Fig. XIV** Bloque crear encuesta

### **20. NuevaEncuesta.asp**

Permite crear una nueva encuesta. Los datos que se han de seleccionar son el año al que hará referencia la encuesta (clave primaria de la tabla aplicación) y las fechas de inicio y fin de aplicación.

Permite generar la encuesta igual que alguna de las realizadas anteriormente, ya que el caso más típico será mantener la misma estructura de preguntas del año anterior para poder realizar comparativas con las valoraciones obtenidas.

Al enviar la información se comprueba que el año que se indica en las fechas de inicio y fin sea el mismo que el año de la encuesta, que la fecha final sea mayor que la fecha inicial, que el formato de las fechas sea el correcto y que estén rellenos todos los campos posibles.

### **21. CrearEncuesta.asp**

Genera las tablas Formulario\_Año, Encuestado\_Año, Respuesta\_Año y Comentarios\_Año y añade un nuevo registro en la tabla Aplicación.

Si está seleccionada la opción igual que algún año anterior, el contenido de la tabla Formulario de aquel año se inserta en la tabla Formulario\_Año creada.

Para que esta opción funcione correctamente es necesario que las creaciones de todas las tablas se haga de forma conjunta ya que en

caso de error en la creación de alguna de ellas se debería suspender todo el proceso y anular las tablas creadas.

Para mantener esta coherencia se abre una transacción de tal forma que todas las operaciones que se realicen a partir de ese momento no serán efectivas hasta que no se cierre la transacción.

```
Session("conexion").BeginTrans  
Session("conexion").Execute(strSQL)  
Session("conexion").Execute(strSQLF)  
Session("conexion").Execute(strSQLE)  
Session("conexion").Execute(strSQLR)  
Session("conexion").Execute(strSQLC)
```

En caso de producirse algún error se deshacen todos los cambios efectuados en la base de datos desde el inicio de la transacción.

```
Session("conexion").RollBackTrans
```

En caso contrario, se cierra la transacción haciendo efectivos los cambios efectuados dentro de ella.

```
Session("conexion").CommitTrans
```

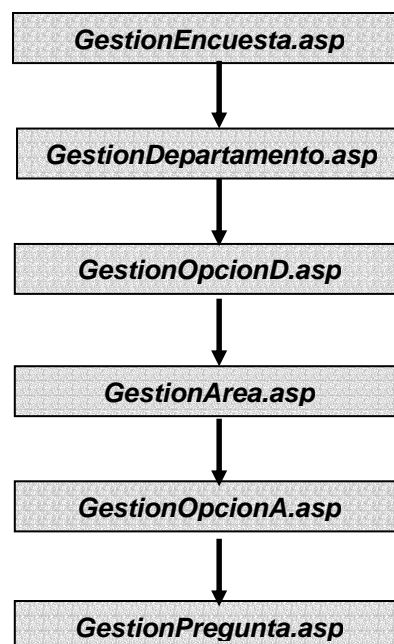


Fig. XV Bloque mantenimiento encuesta

## 22. *GestionEncuesta.asp*

Página que permite seleccionar, indicando el año, de que encuesta se pretende realizar el mantenimiento. De las encuestas disponibles únicamente se podrá realizar mantenimiento de aquellas encuestas que todavía no se hallan cursado (fecha inicio > fecha actual) ya que la

modificación de una encuesta ya valorada podría generar falta de coherencia.

### **23. *GestionDepartamento.asp***

Guarda como variable de sesión el parámetro año recibido, muestra los departamentos disponibles y permite las opciones de Gestión, Eliminar y Vista Previa.

### **24. *GestionOpcionD.asp***

Si la opción eliminada es eliminar, borra de la tabla Formulario\_año todos aquellos registros que tienen asignado el departamento seleccionado.

Si la opción es Gestión ejecutamos la página GestionArea.asp.

### **25. *GestionArea.asp***

Muestra de color negro las areas asociadas al departamento seleccionado y de color gris el resto de areas que pueden ser incluidas.

Las opciones que se permiten son Gestión y Eliminar.

### **26. *GestionOpcionA***

Si la opción pulsada es Gestion, se muestra un lista con todas las preguntas de este departamento y area, y con las posibles preguntas que pueden ser insertadas.

Las opciones que permite son Insertar y Eliminar.

Si por el contrario, la opción seleccionada es eliminar, se borrarán de la tabla Formulario\_año todos aquellos registros formados por el area y departamento seleccionados.

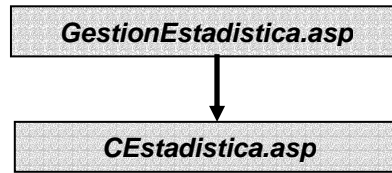
### **27. *GestionPregunta.asp***

Si la opción pulsada es insertar, comprueba que la pregunta que se pretende insertar no este insertada ya que la clave principal de la tabla formulario es un identificador y no los campos departamento, area y pregunta.

Si la pregunta no existía la inserta y si ya existía muestra un mensaje informando al usuario que la pregunta ya está insertada.

Si la opción pulsada es eliminar, se borra el registro con la pregunta seleccionada.





**Fig. XV** Bloque estadísticas

### **28. *GestionEstadistica.asp***

Página que permite seleccionar, indicando el año, de que encuesta se pretende realizar los cálculos estadísticos.

Envía a la página *CEstadistica.asp* información sobre el año seleccionado y sobre la opción pulsada en el menú (Cálculo de participación, media obtenida en los departamentos a encuestar o media de las valoraciones realizadas por las sociedades).

### **29. *CEstadistica.asp***

Dependiendo de la opción pulsada y del año seleccionado calcula la participación o las medias obtenidas.

Los resultados se muestran por pantalla en un taba.

### **30. *Logout.asp***

Pregunta al usuario si realmente quiere abandonar el módulo de administración. Si la respuesta es afirmativa abandona la sesión y redirige la aplicación a la pantalla principal.

# DOCUMENTACIÓN

## **1.- Documentación para el programador**

La documentación técnica de cómo se ha desarrollado la aplicación se corresponde a este propio documento en el que se explican todos los pasos realizados para la implementación de la aplicación.

Con este documento cualquier otra persona sería capaz de entender los criterios seguidos durante el desarrollo de la aplicación y realizar modificaciones en el código de la misma.

## **2.- Documentación para la instalación**

Para utilizar la aplicación implementada basta con tener instalado un servidor Web, disponer de la base de datos de la aplicación y crear una conexión ODBC a esta base de datos con el nombre de origen de datos ENQ.

El código de la aplicación se ha de copiar en algún directorio del servidor Web y en el propio servidor, crear un directorio virtual alias de este directorio físico y con página de inicio el fichero *Principal.html*.

Ejecutando [http://nombre\\_servidor/alias](http://nombre_servidor/alias) en el explorador, obtendríamos la página de inicio de la aplicación.

## **3.- Documentación para el usuario**

En este manual se incluye la información necesaria para el uso de la aplicación, desde el punto de vista del encuestado y desde el punto de vista del administrador.

Debido a la extensión del fichero, se adjunta como *Manual\_Usuario.doc*

## CONCLUSIONES

Los objetivos de este trabajo eran diseñar e implementar una aplicación que permitiese realizar las encuestas anuales que se gestionan desde la División Servicios del Grupo Finaf92 para medir el grado de satisfacción.

Estos objetivos se han cumplido, el producto final cumple con las expectativas iniciales aunque la aplicación permite continuar desarrollando y mejorando nuevas funcionalidades.

El principal aspecto a remarcar de la realización de este proyecto son las ventajas que ofrece respecto al procedimiento que estaba en curso.

- Evita la falta de coherencia en las respuestas: El proceso de realización de la encuesta es guiado, si se produce algún error el proceso se cancela y los datos guardados hasta el momento quedan inutilizables.
- Mejora la fiabilidad de los datos: Los datos son insertados directamente en la base de datos, por lo que no se pueden dar errores de introducción manual de valoraciones.
- Agiliza el procedimiento: Una vez los empleados han realizado los valoraciones no es necesario realizar ningún proceso adicional, ya se puede realizar el análisis estadístico.

Por otra parte, se permiten modificar los márgenes de tiempo para la realización de la encuesta, cambiando la fecha fin de la encuesta en curso

El único punto débil que no es posible solventar con la solución propuesta es la duplicación de respuestas. Una persona podría cumplimentar tantas encuestas como quisiese.

Este problema viene determinado por el hecho de que las encuestas son anónimas. En la realización de encuestas anónimas, el 'fraude' se detecta en el recuento de respuestas. En este caso, como la encuesta no es obligatoria se podría dar el caso de que 10 personas no contestasen y uno contestase 10 veces. Si pretendemos que la encuesta sea anónima no es posible solucionar este punto débil.

Posibles puntos de mejoras para la aplicación serían:

- Cálculo y depuración de nuevas estadísticas
- Aumentar el alcance de la aplicación: Esta aplicación está diseñada para que la encuesta se realice desde cualquier puesto operativo de la red de Finaf92. Cabe comentar que el grupo consta de una red de ventas que no están conectados a la red de Finaf92 pero de los cuales también nos interesa su opinión.

Se podría estudiar una posible solución, como por ejemplo, que envíen las respuestas por correo y se integrasen de forma automática. La solución propuesta de forma temporal es habilitar un ordenador conectado a la red

de la empresa en las convenciones, para poder realizar las valoraciones de la división.

Finalmente comentar que desde la División Servicios del Grupo se pretende implantar una Intranet desde donde se gestionarán diversas aplicaciones. Actualmente está en curso el proyecto de implantación de un portal corporativo y la aplicación de gestión de las encuestas formará parte de las aplicaciones del portal.