

# **Geslab 1.0**

***Memoria del  
desarrollo de la  
aplicación***

*Autor: Álvaro de Luis de Miguel*

# **Memoria del desarrollo de una aplicación realizada para la Universidad de Burgos**

En esta memoria se van a detallar las fases del desarrollo de una aplicación para la gestión del inventario de los laboratorios de la Universidad de Burgos.

## **1. Entrevistas con los futuros usuarios**

Al comienzo de la beca, en Febrero, tuvo lugar una primera entrevista (muy larga) con un usuario en la que se especificaron los requisitos fundamentales del sistema.

Durante las dos semanas siguientes se realizaron entrevistas puntuales con ese mismo usuario para detallar alguno de los requisitos que no habían quedado claros.

Más tarde, en Junio, la coordinadora de Seguridad y Salud convocó una reunión llamando a todas las áreas interesadas y se contó con la asistencia de unos diez representantes. En esta reunión se introdujeron cambios importantes que obligaron a modificar en profundidad el sistema, desde el análisis hasta la implementación, que ya estaba bastante avanzada, en especial la interfaz de usuario.

## **2. Especificación de requisitos del sistema**

Durante la primera entrevista se elaboró una especificación de requisitos del sistema inicial, pero en la siguiente reunión con los usuarios se añadieron varios requisitos, como la gestión de proveedores, y algunos cambios en los campos de las tablas.

A continuación se incluye una especificación de requisitos del sistema. Esta especificación es la que se definió en la segunda entrevista.

### **2.1. Descripción general del sistema**

El sistema deberá ser capaz de gestionar el inventario de todos los laboratorios de química de la Universidad de Burgos, controlando la fecha en que entran los productos y la fecha en que salen, y a partir de estos datos se calculan las existencias. Debe permitir

consultar las existencias de los otros laboratorios, de tal forma que unos laboratorios puedan pedir a otros productos que les faltan, y así aprovechar mejor los recursos de la Universidad.

## **2.2. Descripción detallada de los requisitos del sistema**

1. El sistema deberá tener terminales en cada uno de los laboratorios interesados en gestionar su inventario de productos.
2. Cada área deberá gestionar solamente sus propios productos. En la Universidad hay varios departamentos y varias áreas por cada departamento. Un área puede encontrarse repartida por distintos centros, y lógicamente en un centro hay varias áreas.
3. En cada área y centro hay varias localizaciones, es decir, ubicaciones donde se pueden guardar los productos (tales como armarios, estanterías...).
4. Cada área debe tener la posibilidad de ocultar una localización a las demás áreas, de tal forma que éstas no puedan conocer la existencia de los productos almacenados en dichas localizaciones (véase más adelante).
5. Se almacenarán los productos con su nombre, fórmula y precauciones. Este último campo será de texto libre.
6. Los productos deberán estar organizados en fichas. Cada ficha se corresponde con un producto guardado en una localización (no es lo mismo un bote de NaCl guardado en el armario que un saco de NaCl guardado en el almacén), con una calidad determinada, de una marca determinada y proporcionado por un proveedor determinado. Este último campo se podrá dejar en blanco.
7. Se almacenarán datos de los proveedores, como su nombre, dirección, teléfono, fax, e-mail y una persona de contacto.
8. Se guardarán además los nombres de las marcas y calidades de los productos. Se deberá permitir su gestión por parte de cualquier usuario.
9. También se gestionará la relación entre marcas de productos y proveedores. Un proveedor llevará varias marcas, y una marca puede ser llevada por varios proveedores.
10. En los envoltorios de los productos (frascos...) suele venir un número de lote. Este se guardará también, opcionalmente.
11. En caso de que el producto tenga fecha de caducidad, se deberá permitir almacenar ésta.

12. En un mismo pedido suelen llegar varias unidades de un producto que corresponde a una determinada ficha.
13. Se almacenará la capacidad de cada frasco, bote... Dicha capacidad puede venir dada en gramos(g) o en mililitros(ml).
14. Se podrá especificar si un frasco es un residuo o no, es decir, si el área quiere deshacerse de ese producto o no.
15. Se guardarán las entradas y salidas de los productos, de tal forma que cada ficha tendrá varias entradas y salidas. De cada entrada y salida se guardan los mismos datos: fecha, fecha de caducidad, lote, unidades, capacidad, en gramos o en mililitros, residuo.
16. Basándose en las entradas y salidas de los productos, se calcularán las existencias de los productos del área. En este cálculo no se tendrá en cuenta si un producto es residuo o no, ni la fecha de caducidad. Se calculará el número de unidades de una determinada capacidad (Ejemplo: 10 frascos de 200 g, 25 de 500 g, 40 de 1000 ml)
17. Deberá ser posible consultar las existencias de otras áreas. Esto es útil para aprovechar mejor el material que hay en la Universidad, de tal forma que un laboratorio pueda pedir a algún otro un producto que le falta al primero y le sobra al último. Será posible ocultar productos a las demás áreas (véase requisito 4). Se podrán consultar todas las fichas situadas en localizaciones no ocultas. Se sabrá el área y el centro donde están las existencias, pero no la localización exacta, por seguridad.
18. Se podrán generar informes a partir de las existencias de cada área así como de todas las áreas, pudiendo seleccionar los campos que aparecerán en dichos informes.
19. Los datos de departamentos, áreas y centros sólo podrán ser modificados por un usuario administrador, que tendrá el control total sobre el sistema.
20. Cada área podrá gestionar sus propias localizaciones, pero no las de las demás áreas. Se puede añadir, modificar y borrar una localización.
21. Cada área también podrá gestionar solamente sus fichas, entradas y salidas. Para estos datos sólo se permitirán las opciones de añadir y modificar; no será posible borrar, por seguridad.
22. Los productos, calidades, proveedores y marcas serán comunes para todas las áreas y los podrán gestionar cualquiera de éstas. Se supone que cada usuario sabe lo que hace. Las

- operaciones permitidas para estos datos serán añadir y modificar. Se podrán borrar aquellos que no estén incluidos en ninguna ficha.
23. Cada producto tiene una hoja de seguridad. El programa deberá ser capaz de visualizar la hoja de seguridad de cada producto.
  24. Se elaborará documentación de ayuda al usuario, así como ayuda a los operadores para las tareas de mantenimiento y administración del sistema.

### **3. Análisis**

Una vez realizadas la primera entrevista y la especificación de requisitos, se procedió a la elaboración del análisis, que duró casi todo el primer mes de la beca. Se ha utilizado la metodología estructurada de Yourdon. La realización del análisis no ha sido muy detallada, ya que posteriormente se realizará un análisis y un diseño siguiendo una metodología orientada a objetos, para ser presentado el trabajo como proyecto de fin de carrera.

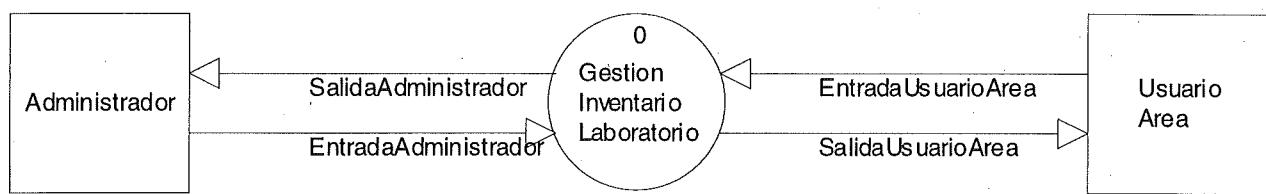
La segunda reunión obligó a cambiar el análisis, lo cual llevó casi otras dos semanas. La especificación de proceso no se actualizó por razones de tiempo.

Para la realización de los DFD se ha utilizado la herramienta EasyCase Professional 4.23.

### 3.1. Diagramas de Flujo de Datos (DFD)

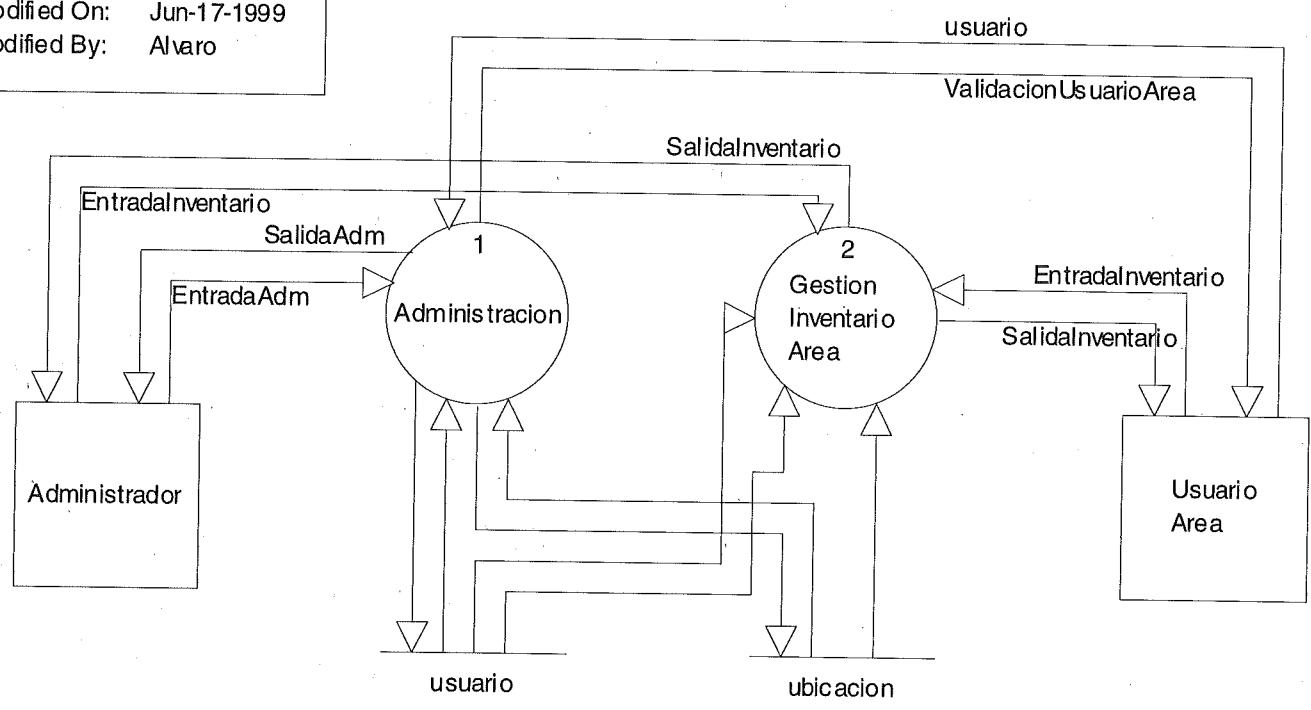
#### 3.1.0. Diagrama de Contexto

Project Name:	GesLab 1.0
Project Path:	c:\bd\dfd\
Chart File:	dfd0.dfd
Chart Name:	Contexto
Created On:	Mar-22-1999
Created By:	Alvaro
Modified On:	Jul-23-1999
Modified By:	Alvaro



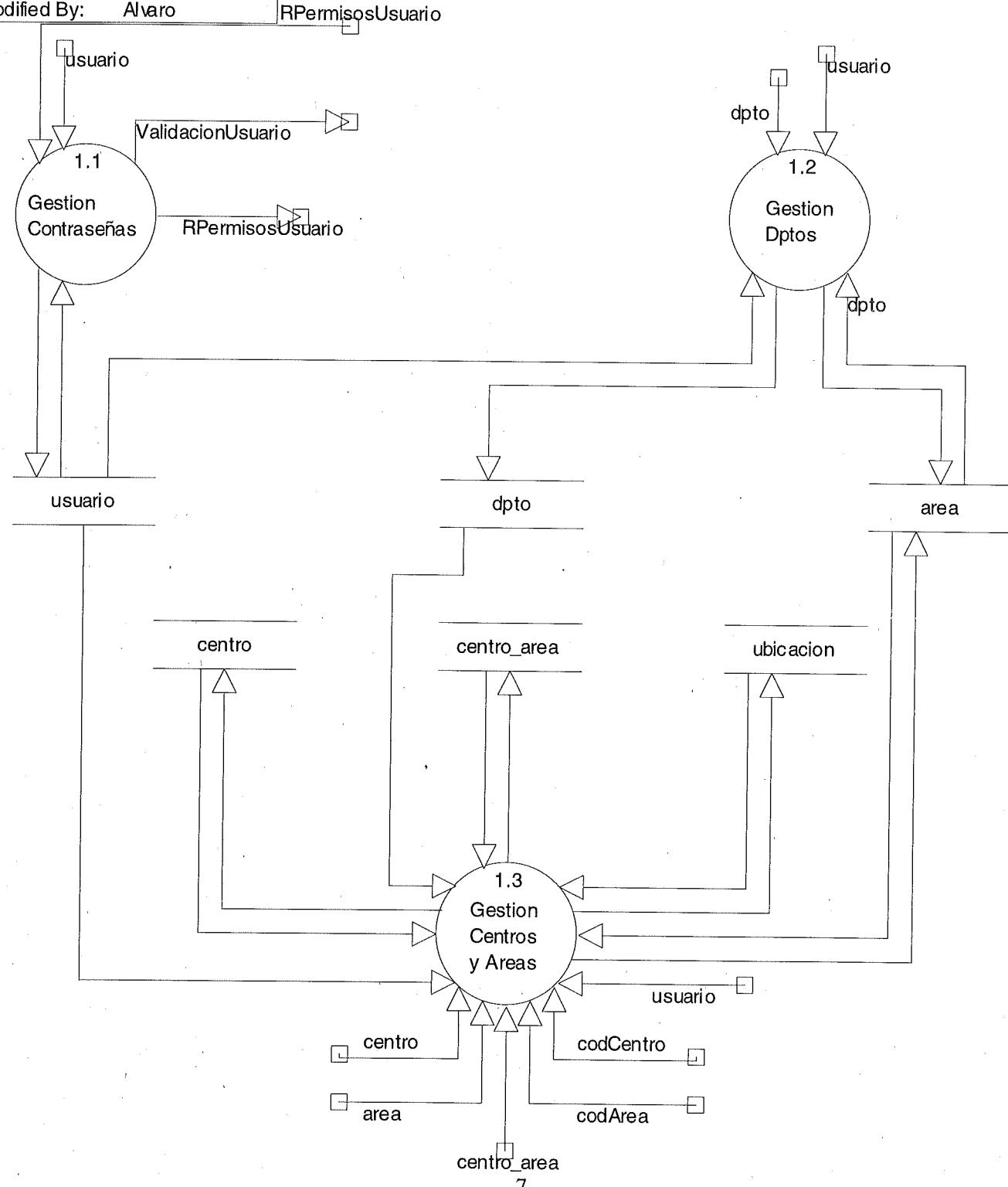
### 3.1.1. Diagrama de Sistema

Project Name: GesLab 1.0  
Project Path: c:\bd\dfd\  
Chart File: sistema.dfd  
Chart Name: Sistema  
Created On: Mar-22-1999  
Created By: Alvaro  
Modified On: Jun-17-1999  
Modified By: Alvaro



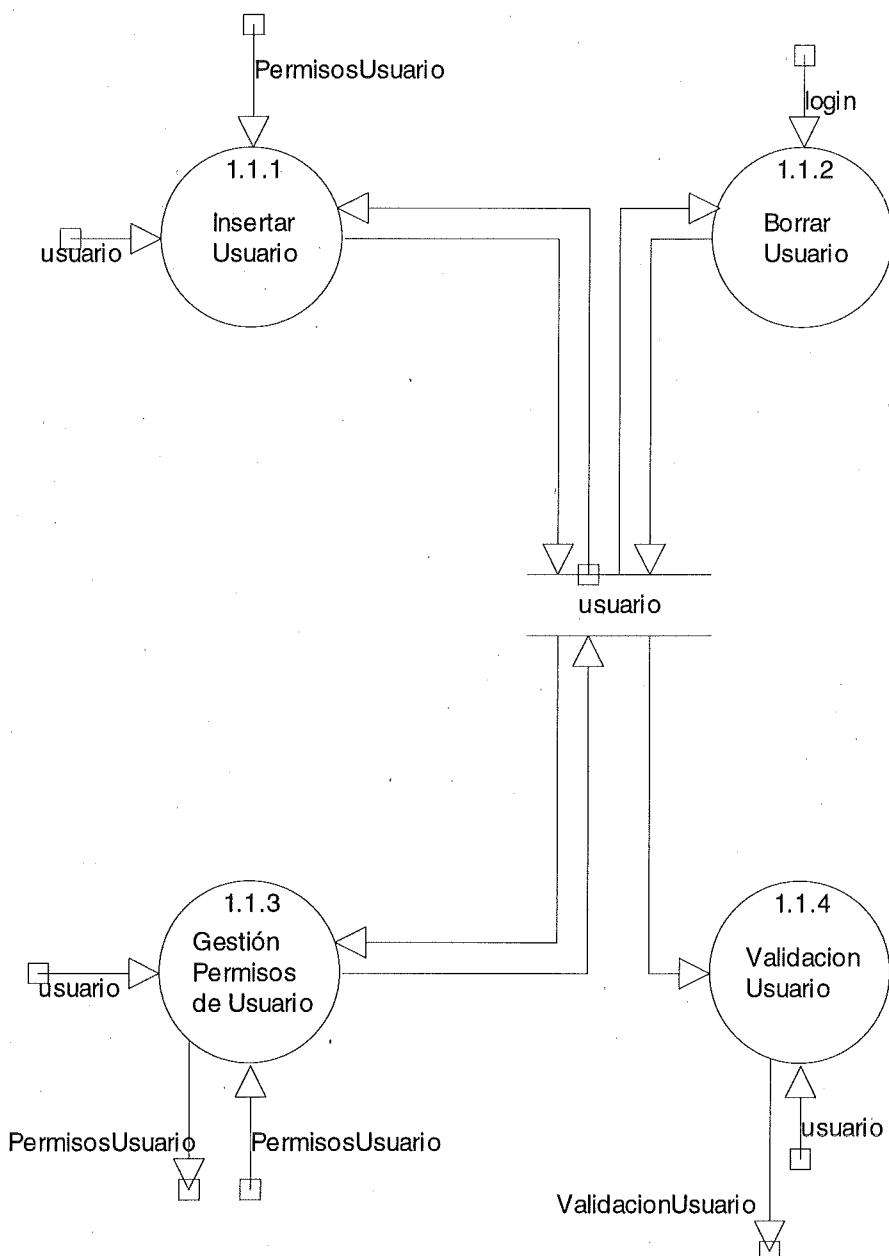
### 3.1.1.1. Administración

Project Name: GesLab 1.0  
 Project Path: c:\bd\dfd  
 Chart File: dfd1dfd  
 Chart Name: Administracion  
 Created On: Mar-22-1999  
 Created By: Alvaro  
 Modified On: Jun-17-1999  
 Modified By: Alvaro



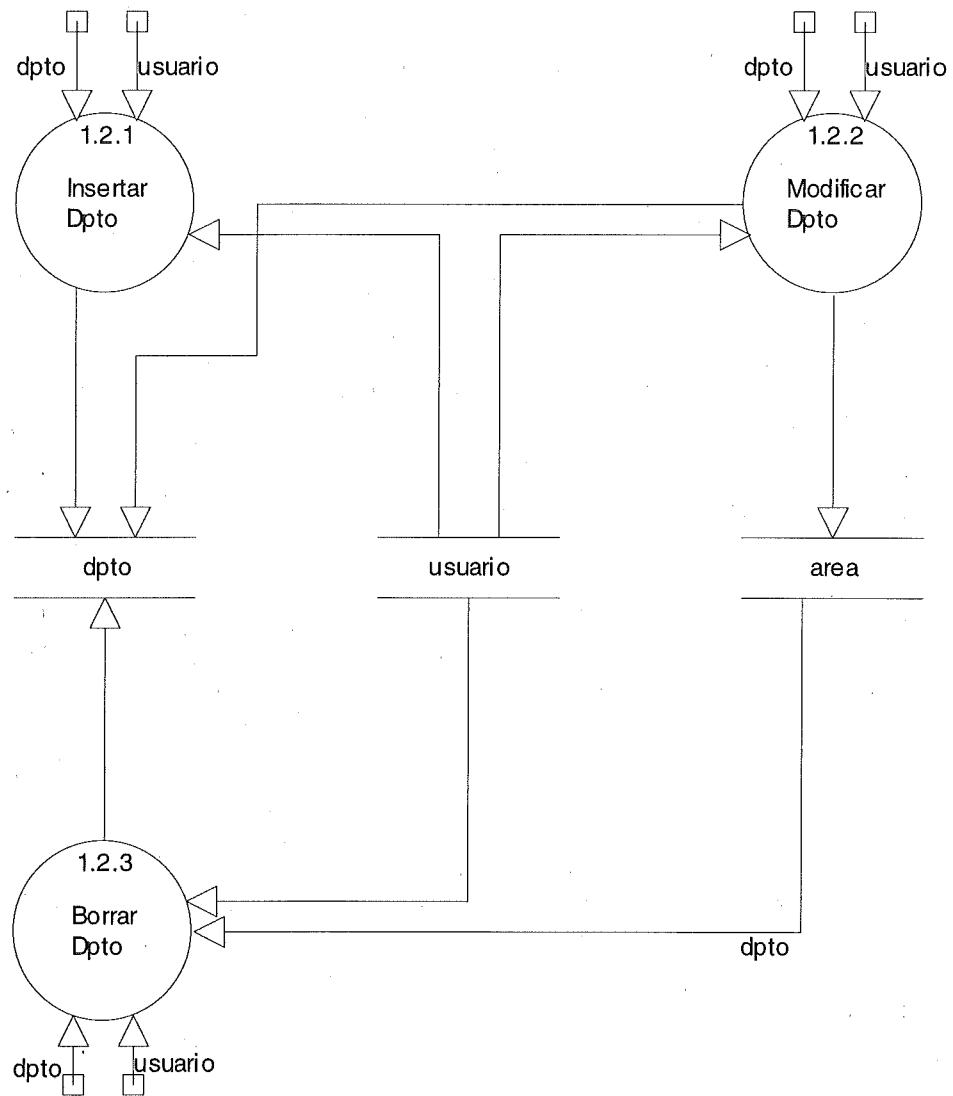
### 3.1.1.1. Gestión Contraseñas

Project Name: Ges Lab 1.0  
Project Path: c:\bd\dfd\  
Chart File: dfd1\_1.dfd  
Chart Name: Gestión Contraseñas  
Created On: Mar-22-1999  
Created By: Alvaro  
Modified On: Jun-14-1999  
Modified By: Alvaro

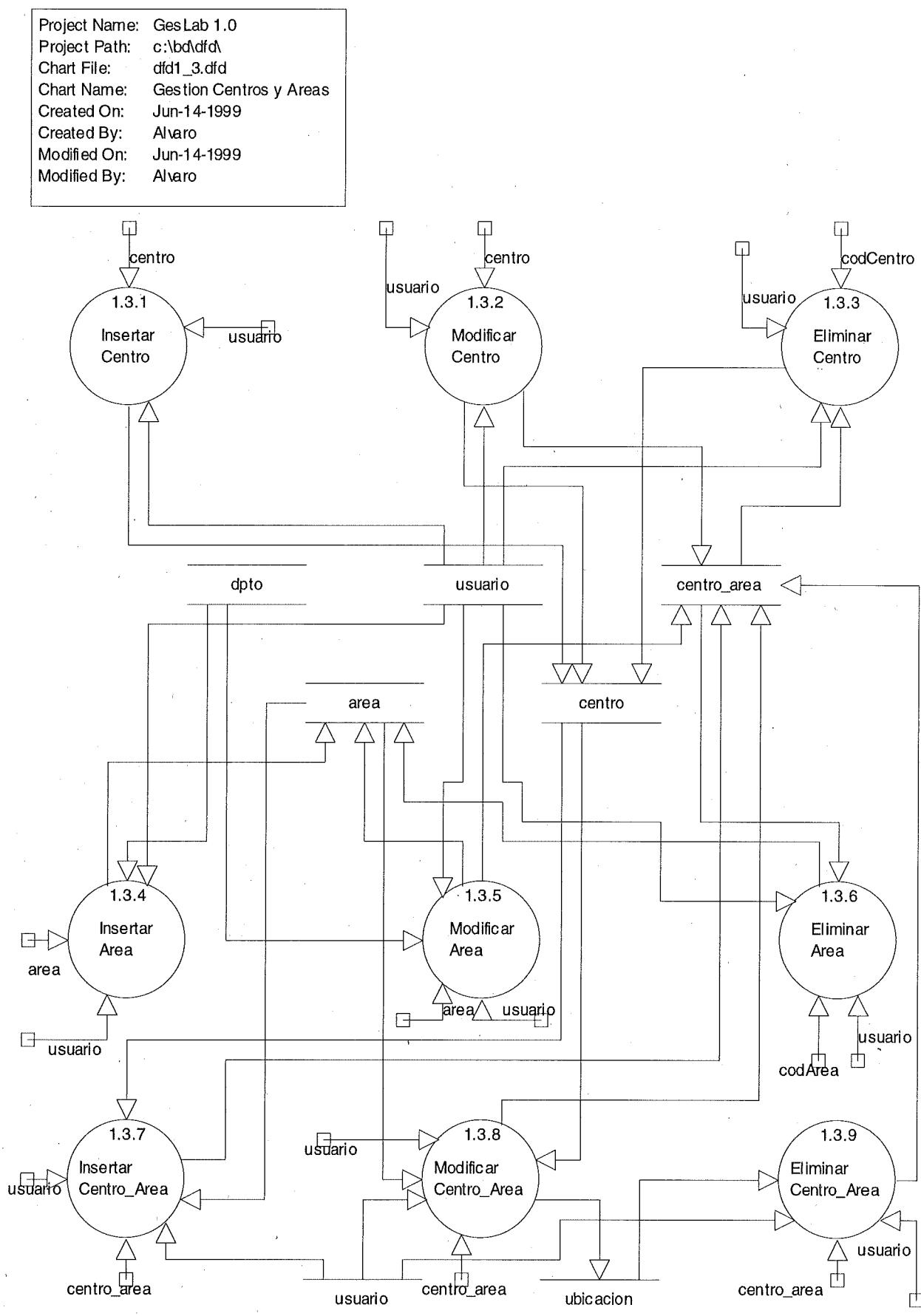


### 3.1.1.1.2. Gestión Dptos

Project Name: GesLab 1.0  
Project Path: c:\bd\dfd  
Chart File: dfd1\_2.dfd  
Chart Name: Gestión Dptos  
Created On: Mar-22-1999  
Created By: Alvaro  
Modified On: Jun-14-1999  
Modified By: Alvaro

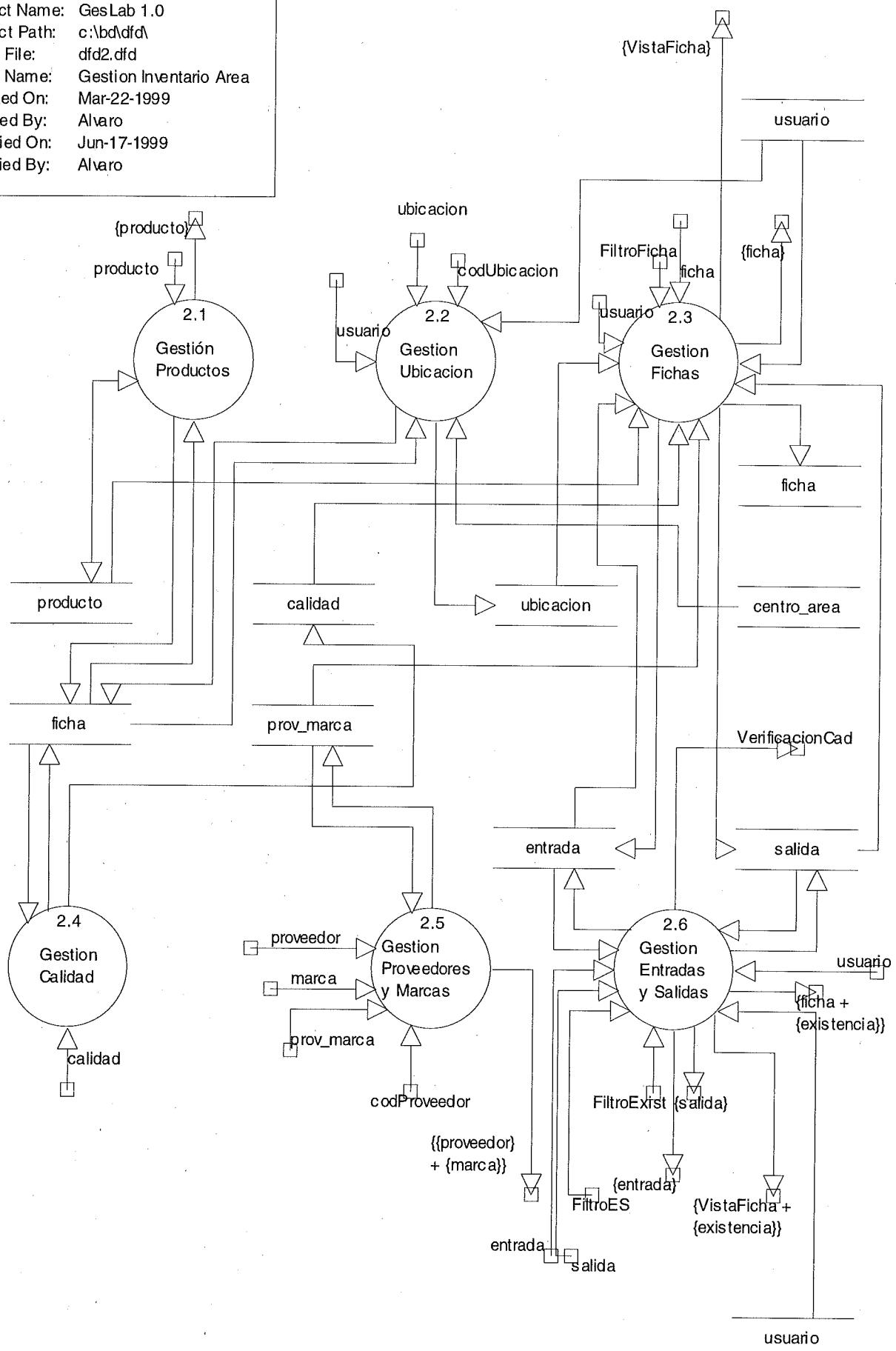


### 3.1.1.1.3. Gestión Centros y Areas



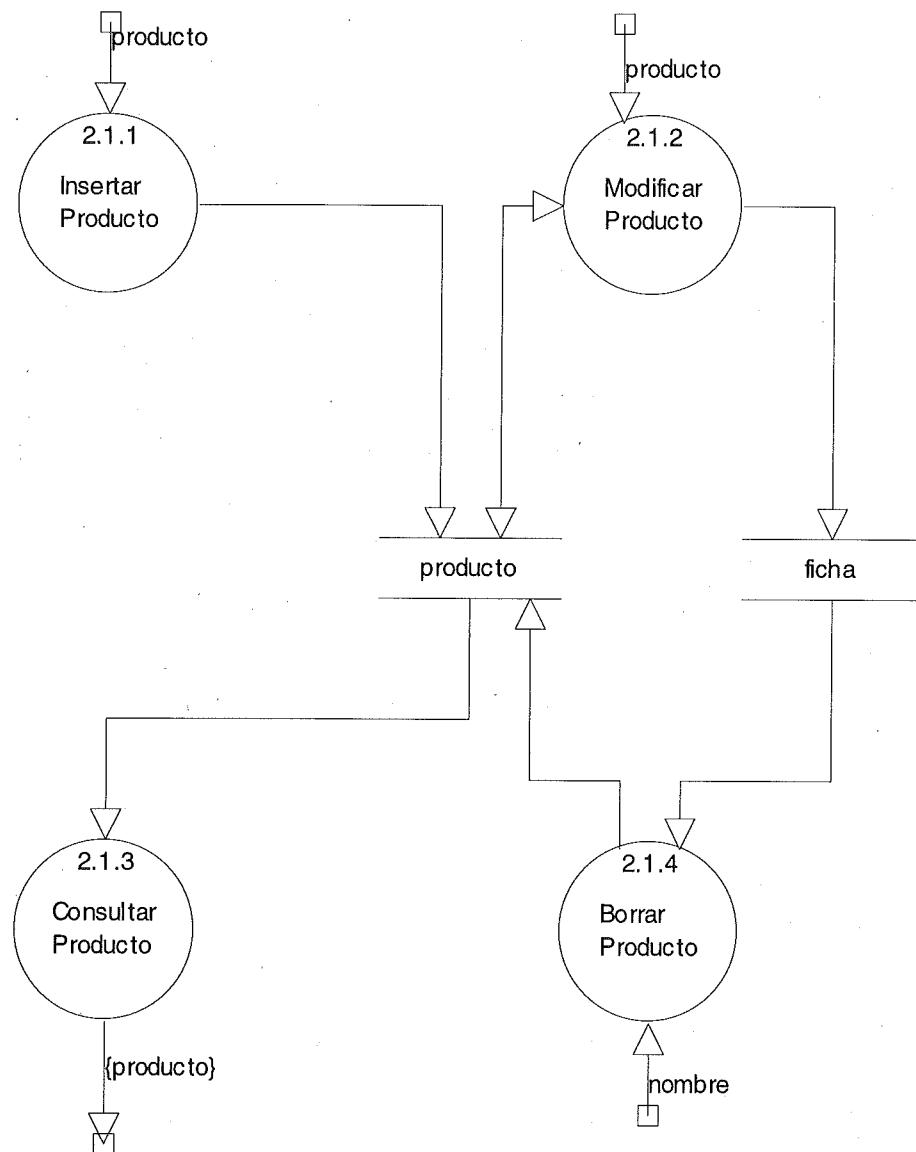
### **3.1.1.2. Gestión Inventario Area**

Project Name:	GesLab 1.0
Project Path:	c:\bd\dfd\
Chart File:	dfd2.rdf
Chart Name:	Gestion Inventario Area
Created On:	Mar-22-1999
Created By:	Alvaro
Modified On:	Jun-17-1999
Modified By:	Alvaro

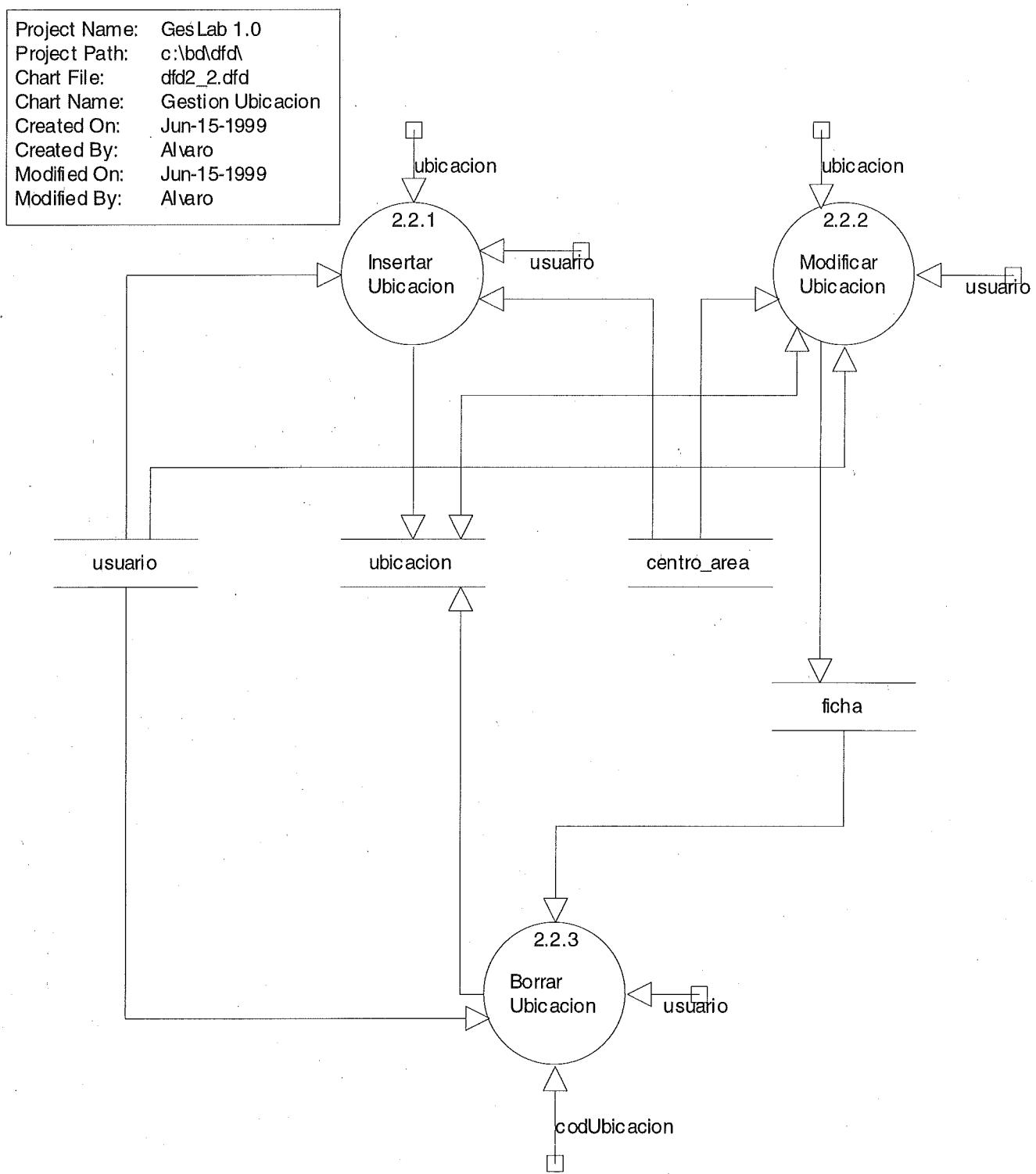


### 3.1.1.2.1. Gestión Productos

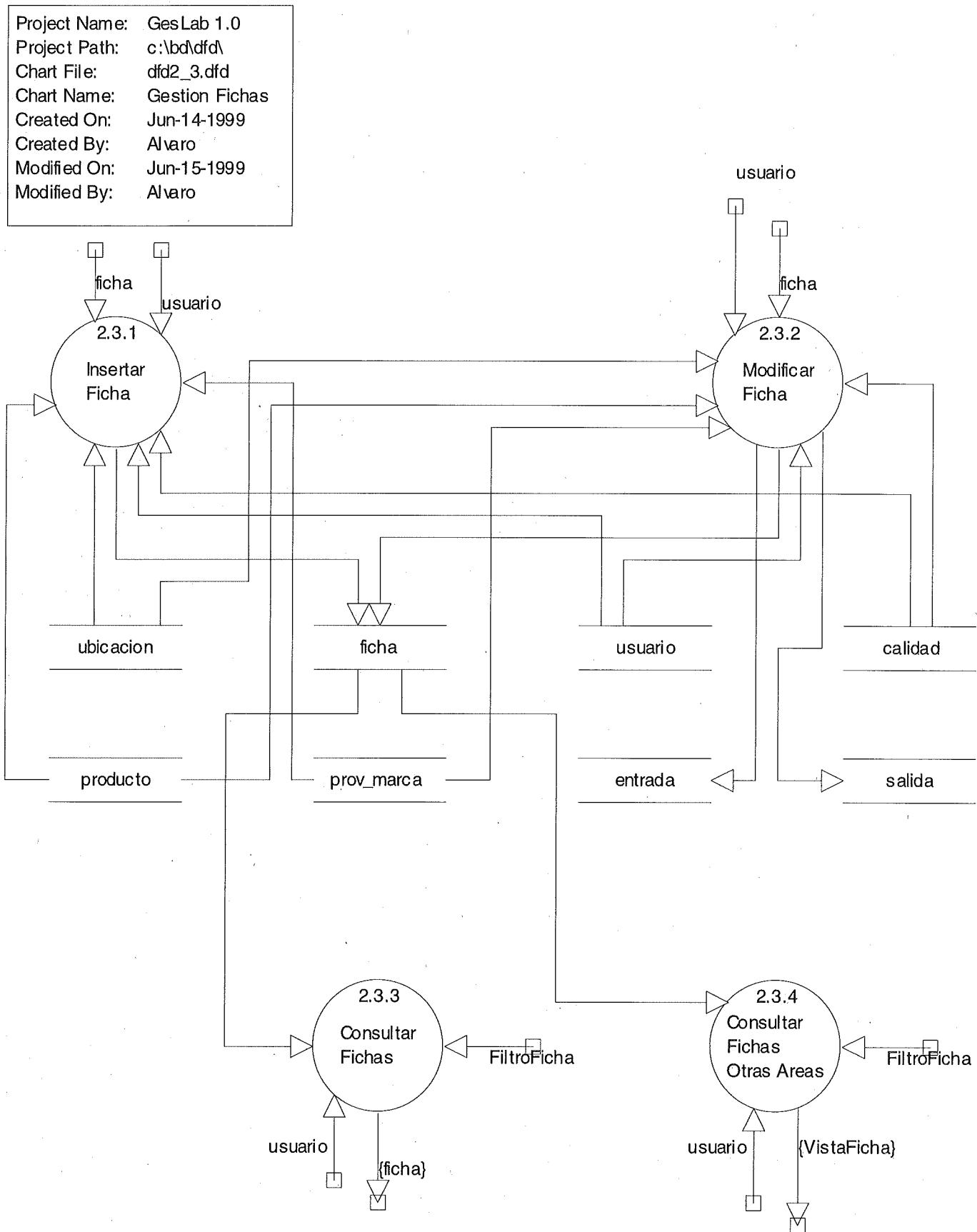
Project Name:	GesLab 1.0
Project Path:	c:\bd\dfd\
Chart File:	dfd2_1dfd
Chart Name:	Gestión Productos
Created On:	Jun-14-1999
Created By:	Alvaro
Modified On:	Jun-14-1999
Modified By:	Alvaro



### 3.1.1.2.2. Gestión Ubicación

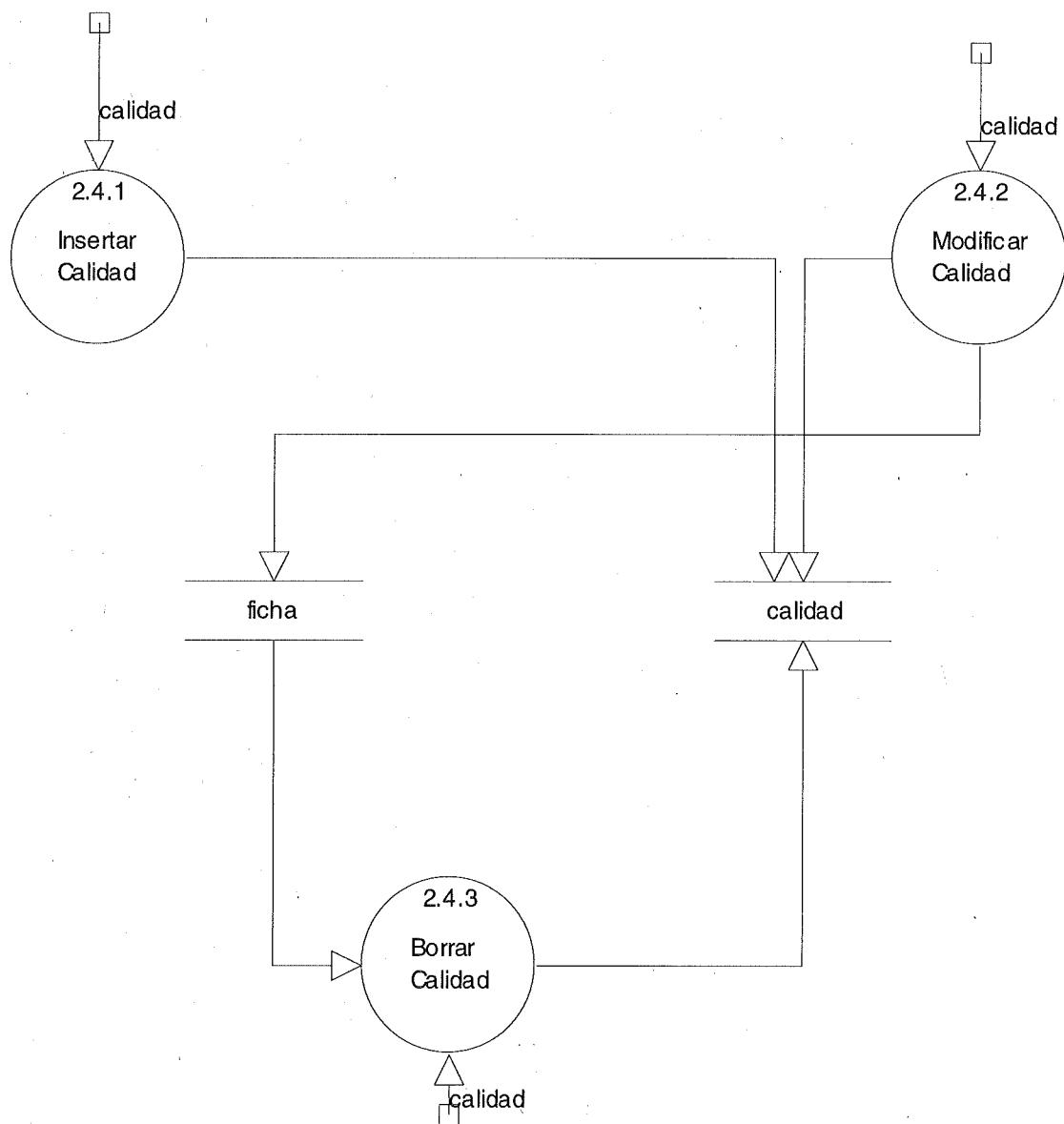


### 3.1.1.2.3. Gestión Fichas

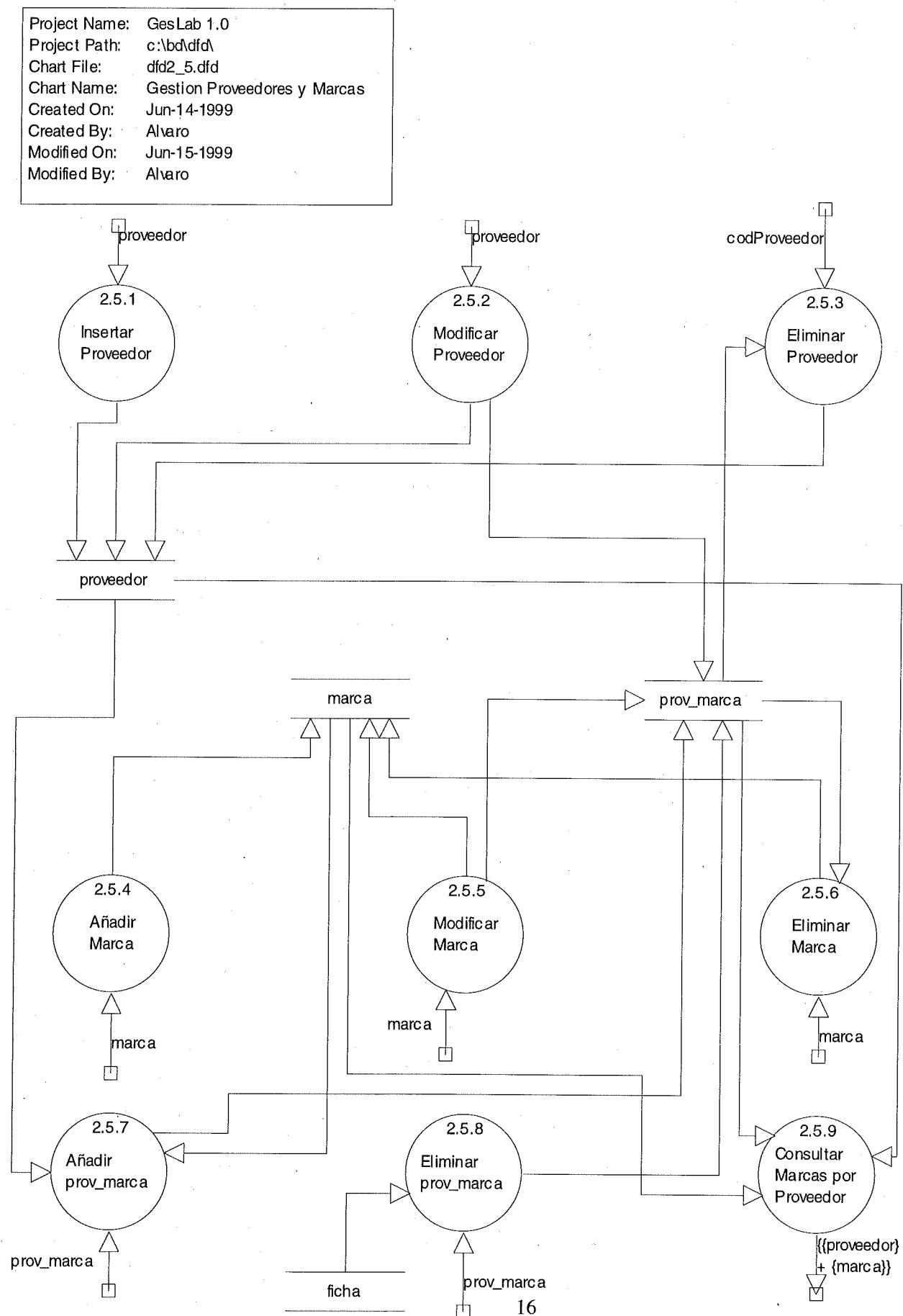


### 3.1.1.2.4. Gestión Calidad

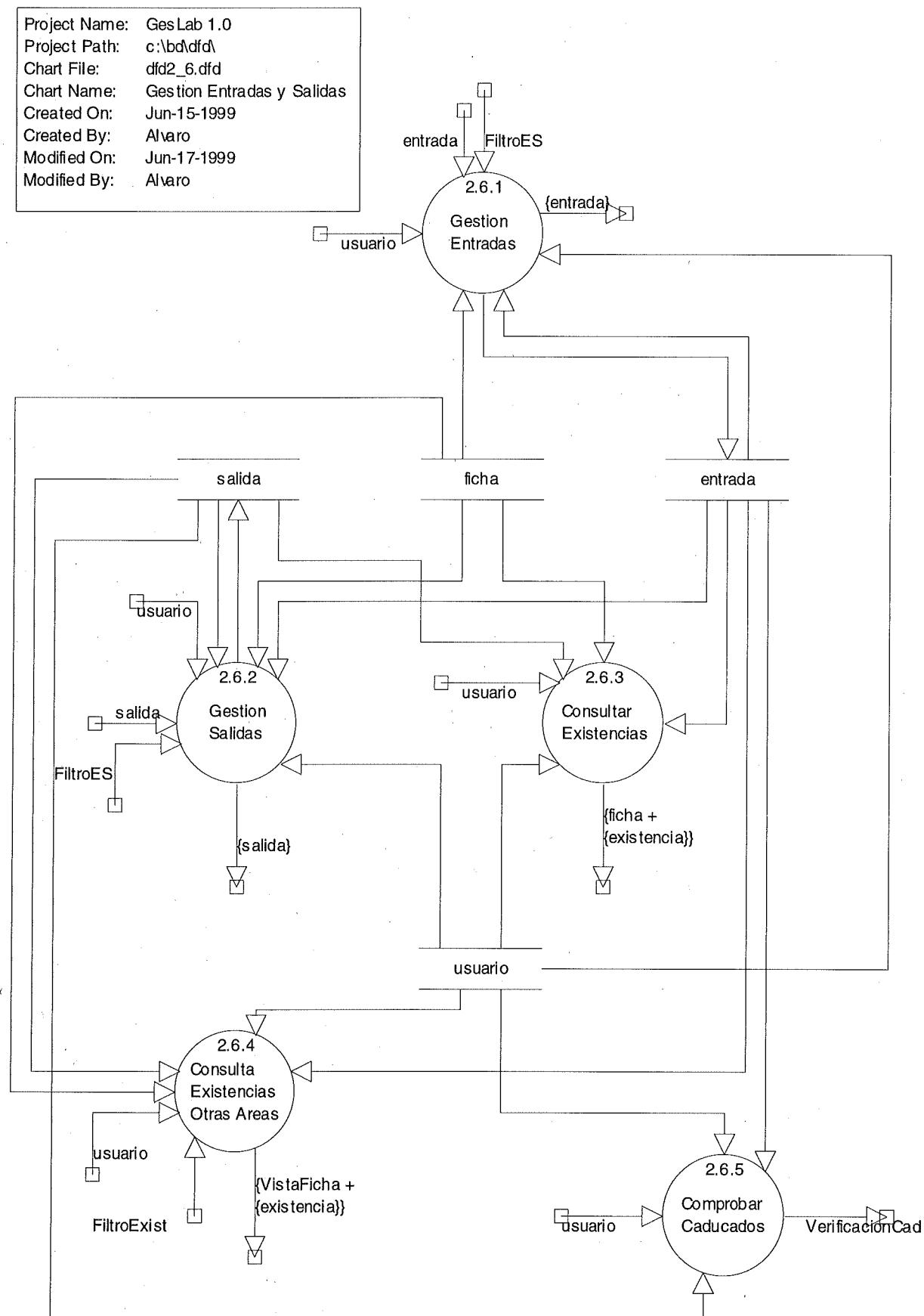
Project Name:	GesLab 1.0
Project Path:	c:\bcd\dfd\
Chart File:	dfd2_4dfd
Chart Name:	Gestion Calidad
Created On:	Jun-14-1999
Created By:	Alvaro
Modified On:	Jun-14-1999
Modified By:	Alvaro



### 3.1.1.2.5. Gestión Proveedores y Marcas

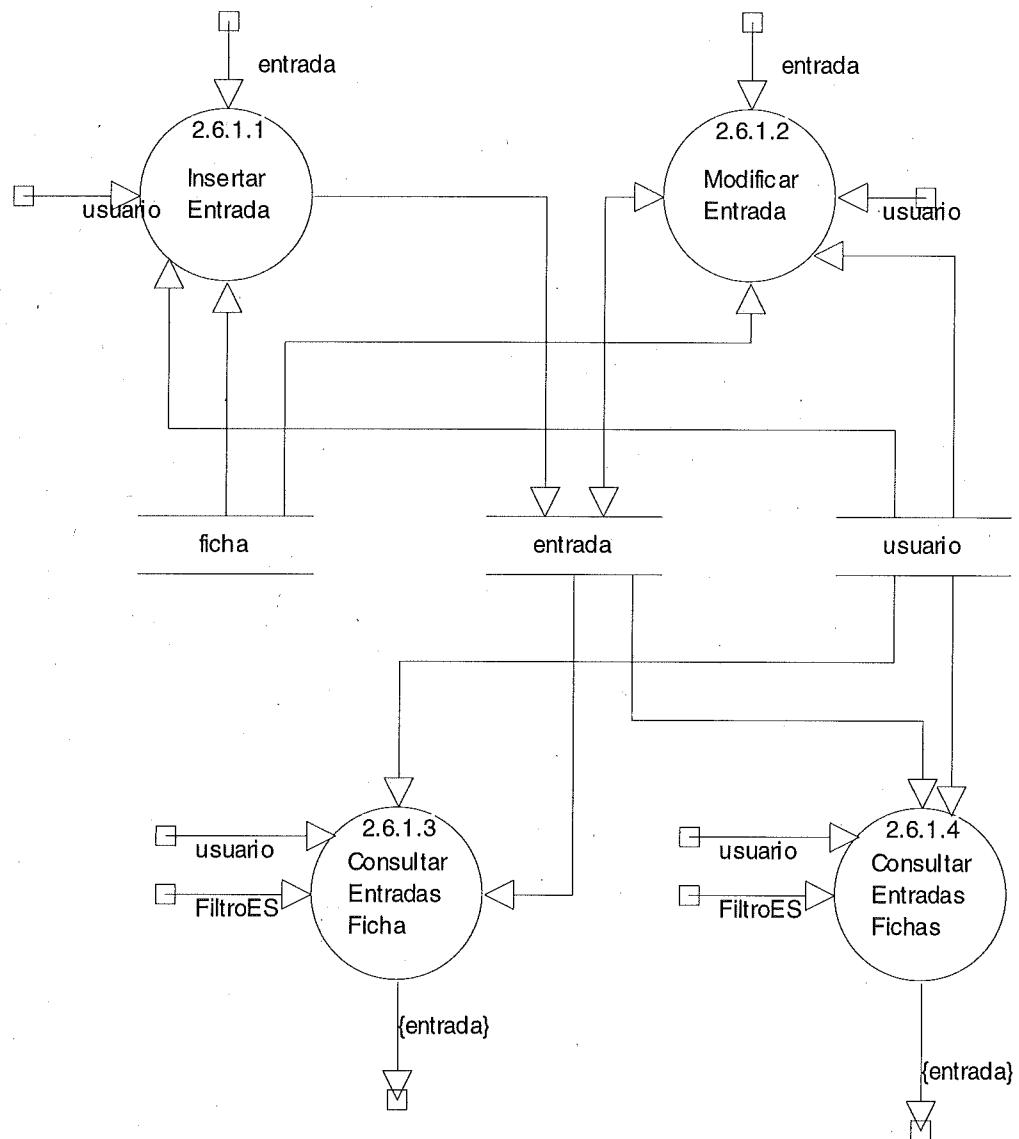


### 3.1.1.2.6. Gestión Entradas y Salidas



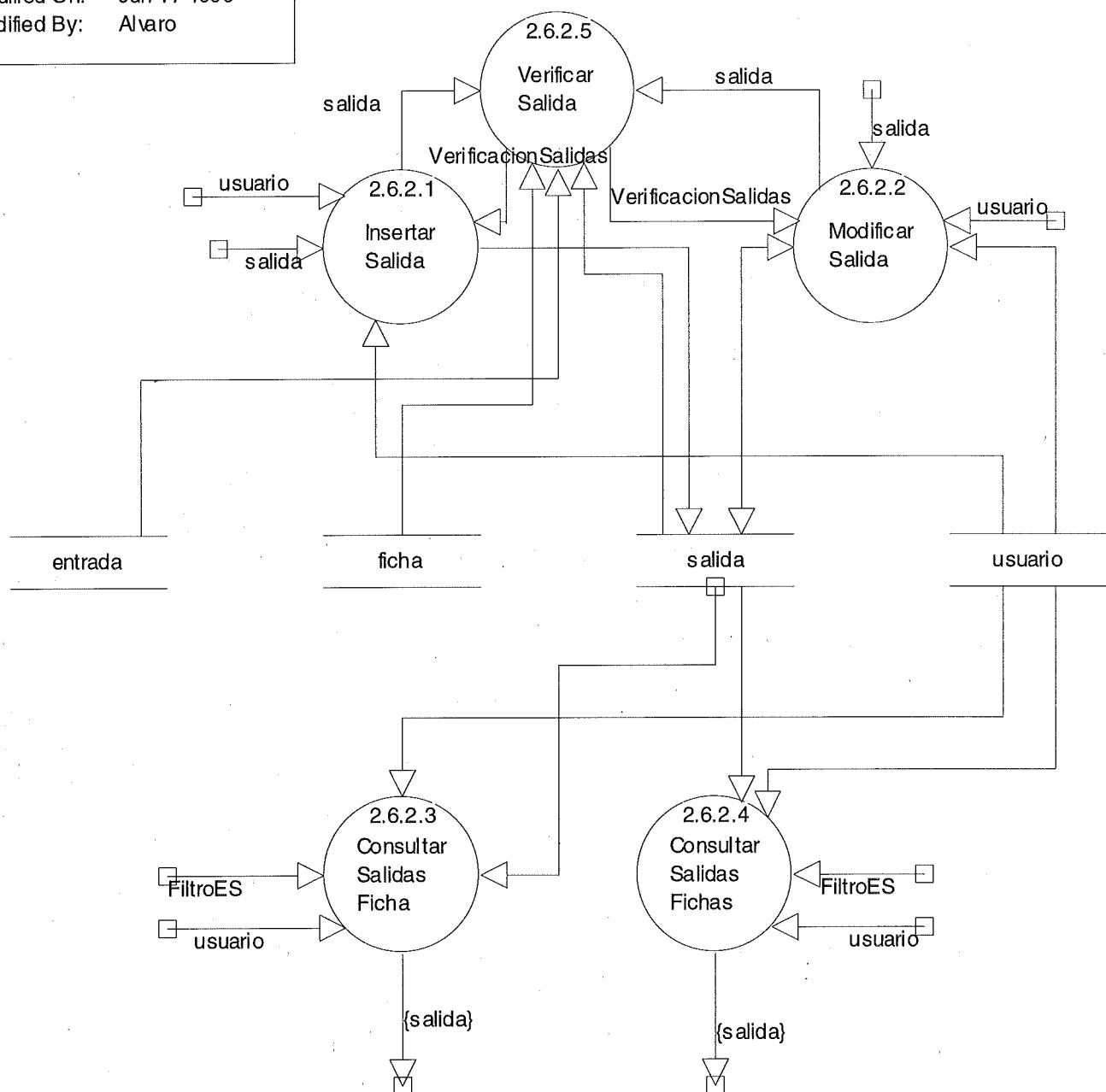
#### **3.1.1.2.6.1. Gestión Entradas**

Project Name: Ges Lab 1.0  
Project Path: c:\bd\dfd\  
Chart File: dfd2\_6\_1.dfd  
Chart Name: Gestion Entradas  
Created On: Jun-15-1999  
Created By: Alvaro  
Modified On: Jun-15-1999  
Modified By: Alvaro



### 3.1.1.2.6.2. Gestión Salidas

Project Name: GesLab 1.0  
 Project Path: c:\bd\dfd  
 Chart File: dfd2\_6\_2.dfd  
 Chart Name: Gestión Salidas  
 Created On: Jun-15-1999  
 Created By: Alvaro  
 Modified On: Jun-17-1999  
 Modified By: Alvaro



### **3.2. Diccionario de datos**

#### **3.2.1. Registros**

- RArea = CodArea + NombreArea + NombreDpto
- RCalidad = @NombreCalidad
- RCentro = \*Centros donde hay algun laboratorio\*  
                  @CodCentro + NombreCentro
- RCentro\_Area = \*Relaciona los centros con las áreas (relación varios a varios)\*  
                  @CodCentro + @CodArea
- RDpto = \*Sólo contiene el nombre del departamento\*  
                  @NombreDpto
- REntrada = CodEntrada + CodFicha + Fecha + FechaCaducidad + Lote + Unidades +  
                  Capacidad + g\_ml + Resíduo
- RFicha = @CodFicha + NombreProducto + NombreCalidad +  
                  CodProveedor + CodMarca + CodUbicacion
- RMarca = @NombreMarca
- PermisosUsuario = usuario + permisos
- RProducto = @NombreProducto + Fórmula + Precauciones
- RProveedor = \*Tabla de Proveedores\*  
                  @CodProveedor + Nombre + Tfno + Fax + e-mail +  
                  Contacto
- RProv\_Marca = @CodProveedor + @NombreMarca
- RSalida = CodSalida + CodFicha + Fecha + FechaCaducidad + Lote + Unidades +  
                  Capacidad + g\_ml + Resíduo

- Rubicacion = @CodUbicacion + NombreUbicacion + Oculto +
   
CodArea + CodCentro

### **3.2.2. Almacenes**

- area = {RArea}
- calidad = {RCalidad}
- centro = {RCentro}
- centro\_area = {Rcentro\_Area}
- dpto = {RDpto}
- entrada = {REntrada}
- ficha = {RFicha}
- marca = {RMarca}
- producto = {RProducto}
- proveedor = {RProveedor}
- prov\_marca = {RProv\_Marca}
- salida = {RSalida}
- ubicacion = {RUbicacion}
- usuario = {RPermisosUsuario}

### **3.2.3. Campos**

- Administrador = \*Usuario que tiene autoridad para reflejar en el sistema  
cambios en la organización de las áreas y departamentos\*
- Capacidad = \*\* number 7
- CodArea = \*Código único de cada Area\* char 3
- CodCentro = \*Código único de cada centro\* char 2

- CodEntrada = \*Código único para cada entrada\* char 20
- CodFicha = \*Código único para cada Ficha\* char 10
- CodProveedor = \*Código único para cada Proveedor\* char 4
- CodSalida = \*Código único para cada salida\* char 20
- CodUbicacion = \*Código único de cada Ubicación\* char 4
- Contacto = \*Persona de contacto para un determinado Proveedor\*
  - varchar 40
- e-mail = \*Dirección de correo electrónico del proveedor\* varchar 40
- Fax = \*\* char 9
- Fecha = \*Fecha de llegada o de salida del producto\* date
- FechaCaducidad = \*\* date
- Fórmula = \*Fórmula del producto\* char 30
- g\_ml = \*La capacidad del envase puede venir dada en gramos o mililitros
  - [g | ml] char 2
- login = \*Nombre del usuario\* [Administrador | UsuarioArea]
- Lote = \*Lote al que pertenece un envase de producto\* char 10
- NombreArea = \*\* char 40
- NombreCalidad = \*Tipo de calidad\* char 10
- NombreCentro = \*\* char 40
- NombreDpto = \*\* char 25
- NombreUbicacion = \*\* varchar 30
- NombreMarca = \*Marca comercial del compuesto\* char 20
- NombreProducto = \*Nombre único para cada compuesto químico\* char 40
- NombreProveedor = \*Nombre de cada Proveedor\* char 40
- Oculto = \*Los productos que se encuentren en una ubicación oculta no serán visibles a los demás usuarios\* [Sí | No] boolean

- password = \*Contraseña única para cada usuario\*
- permisos = \*Permisos de los usuarios de cada área sobre sus registros y los registros de las demás áreas. Lo gestiona el SGBD.\*
- Precauciones = \*Peligros de un producto\* varchar 40
- Residuo = \*El producto puede ser un residuo indeseable\* [Sí | No] boolean
- Tfno = \*\* char 9
- Unidades = \*\* number 6
- UsuarioArea = \*Un nombre por cada una de las distintas áreas que utilizan el sistema\*

### **3.2.4. Otros Flujos de datos**

- EntradaAdm = \*Flujo de datos de entrada al proceso Administración desde Administrador\*
  - usuario + RpermisosUsuario + dpto + centro + area + centro\_area + codArea + codCentro
- EntradaAdministrador = \*Flujo de datos de entrada al sistema desde el Administrador\*
  - EntradaAdm + EntradaInventario
- EntradaInventario = \* Flujo de datos de entrada al proceso Gestión
  - Inventario Area desde ambos usuarios\*
  - Producto + usuario + ubicacion + codUbicacion + ficha
  - + FiltroFicha + calidad + proveedor + marca + prov\_marca + codProveedor + entrada + salida + FiltroES
- EntradaUsuarioArea = \*Flujo de datos de entrada al sistema desde UsuarioArea\*

## Usuario + EntradaInventario

- Existencia = \*Detalle de Listado de existencias\*
  - Capacidad + g\_ml + Unidades
- FechaCadDesde = \*Filtro sobre la fecha de caducidad\*
- FechaCadHasta = \* Filtro sobre la fecha de caducidad\*
- FechaDesde = \*\*
- FechaHasta = \*\*
- FiltroES = \*Filtros para las consultas de las entradas y salidas\*
  - (FechaDesde) + (FechaHasta) +
  - (FechaCadDesde) + (FechaCadHasta) + Residuos +
  - (Residuos\_SI\_NO)
- FiltroFicha = \*Filtros para las consultas de las fichas\*
  - (NombreProducto) + (CodArea) + (CodUbicacion) + (Marca) +
  - (Calidad)
- Residuos = \*Filtrar residuos\* [Sí | No]
- Residuos\_SI\_NO = \*Consultar los que sí sean resíduos o los que no lo sean\* [Sí | No]
- SalidaAdm = \*Flujo de datos de salida desde el proceso Administración a Administrador\*
  - ValidaciónAdministrador + RPermisosUsuario
- SalidaAdministrador = \*Flujo de datos de salida del sistema a Administrador\*
  - SalidaAdm + SalidaInventario
- SalidaInventario = \*Flujo de datos de salidasde el proceso Gestión Inventario Area a ambos usuarios\*
  - {producto} + {VistaFicha} + {ficha} + {{proveedor}} +

- $\{marca\} + \{entrada\} + \{salida\} + \{VistaFicha +$   
 $\{existencia\} + \{ficha + \{existencia\}\} + VerificacionCad$
- SalidaUsuarioArea = \*Flujo de datos de salida del sistema a UsuarioArea\*
    - ValidaciónUsuarioArea + SalidaInventario
  - usuario = \*Nombre y contraseña\* @login + password
  - ValidaciónAdministrador = \*Gestionado por el SGBD\* [OKUA | ERROR]
  - ValidacionUsuario = [ValidaciónAdministrador | ValidaciónUsuarioArea]
  - ValidaciónUsuarioArea = \*Gestionado por el SGBD\* [OKADM | ERROR]
  - VerificaciónCad = \*Comprueba que no hay productos que ya han caducado\*
    - [OK | ERROR]
  - VerificaciónSalidas = \*Comprueba que las salidas de material no son
    - superiores a las existencias\*
  - [OK | ERROR]

### 3.3. Especificación de proceso

La especificación de proceso no se ha actualizado, por razones de tiempo, desde la última reunión. La versión antigua es la siguiente:

#### 3.3.1. Administración

##### 3.3.1.1. Gestión Contraseñas

###### 3.3.1.1.1. Insertar Usuario

SI usuario es Administrador

SI login, password y permisos válidos

Añadir PermisosUsuario a Almacén Usuario

FIN SI

FIN SI

### **Términos del DD**

Almacenes: Almacén Usuario

Registros: PermisosUsuario, Usuario

#### **3.3.1.1.2. Borrar Usuario**

SI usuario es Administrador

BUSCAR registro PermisosUsuario en Almacén Usuario con

    mismo login

SI registro encontrado

    Borrar registro PermisosUsuario de Almacén Usuario

FIN SI

FIN SI

### **Términos del DD**

Almacenes: Almacén Usuario

Registros: PermisosUsuario, Usuario

Campos: login

### **3.3.1.1.3. Gestión Permisos de Usuario**

//Problema del SGBD

SI usuario es Administrador

BUSCAR registro PermisosUsuario en Almacén Usuario con  
mismo login

SI registro encontrado

MOSTRAR PermisosUsuario

SINO

MOSTRAR ERROR

FIN SI

SI PermisosUsuario valido

GUARDAR registro PermisosUsuario en Almacén Usuario

FIN SI

FIN SI

### **Términos del DD**

Almacenes: Almacén Usuario

Registros: PermisosUsuario, Usuario

Campos: login

### **3.3.1.1.4. Validación Usuario**

BUSCAR registro PermisosUsuario en Almacén Usuario con mismo login  
y password que Usuario

SI registro encontrado

    SI permisos = Administrador

        MOSTRAR OKADM

    SINO

        MOSTRAR OKUA

    FIN SI

    SINO

        MOSTRAR ERROR

    FIN SI

### **Términos del DD**

Almacenes: Almacén Usuario

Registros: PermisosUsuario, Usuario

Campos: login, password

### **3.3.1.2. Gestión Dptos**

#### **3.3.1.2.1. Insertar Dpto**

SI usuario es Administrador

INSERTAR registro Dpto en almacén Dpto

FIN SI

### **Términos del DD**

Almacenes: Almacén Dpto

Registros: Dpto

Campos: Nombre

### **3.3.1.2.2. Modificar Dpto**

SI usuario es Administrador

GUARDAR registro Dpto en almacén Dpto

ACTUALIZAR Area.Dpto en registros Area de Almacén Area

FIN SI

### **Términos del DD**

Almacenes: Almacén Dpto, Almacén Area

Registros: Dpto, Area

Campos: Dpto.Nombre, Area.Dpto

### **3.3.1.2.3. Borrar Dpto**

SI usuario es Administrador

BUSCAR registros Area en Almacén Area con Dpto = Nombre

SI no encontrado

BORRAR registro Dpto de almacén Dpto

SINO

MOSTRAR ERROR

FIN SI

FIN SI

### **Términos del DD**

Almacenes: Almacén Dpto, Almacén Area

Registros: Dpto, Area

Campos: Dpto.Nombre, Area.Dpto

### **3.3.1.3. Gestión Areas**

#### **3.3.1.3.1. Insertar Area**

SI usuario es Administrador

BUSCAR registro Dpto con Dpto = Dpto.Nombre

SI no encontrado

MOSTRAR ERROR

SINO

INSERTAR registro Area en Almacén Area

FIN SI

FIN SI

### **Términos del DD**

Almacenes: Almacén Dpto, Almacén Area

Registros: Dpto, Area

Campos: Dpto.Nombre, Area

#### **3.3.1.3.2. Modificar Area**

SI usuario es Administrador

BUSCAR registro Dpto con Dpto = Dpto.Nombre

SI no encontrado

MOSTRAR ERROR

SINO

GUARDAR registro Area en Almacén Area

SI CodArea ha cambiado

ACTUALIZAR registros Centro/Area de Almacén

Centro/Area

ACTUALIZAR registros Ubicacion de Almacén

Ubicacion

FIN SI

**FIN SI**

**FIN SI**

### **Términos del DD**

Almacenes: Almacén Dpto, Almacén Area, Almacén Ubicacion,

Almacén Centro/Area

Registros: Dpto, Area, Ubicacion, Centro/Area

Campos: Dpto.Nombre, Area.Dpto, Area.CodArea,

Ubicacion.CodArea, Centro/Area.CodArea

#### **3.3.1.3.3. Borrar Area**

SI usuario es Administrador

BUSCAR registro Centro/Area o Ubicacion en Almacén

Centro/Area y Almacén Ubicacion con igual CodArea

SI encontrado

MOSTRAR ERROR

SINO

BORRAR registro Area de Almacén Area

**FIN SI**

**FIN SI**

## **Términos del DD**

Almacenes: Almacén Area, Almacén Centro/Area, Almacén Ubicacion

Registros: Area, Centro/Area, Ubicacion

Campos: Area, Ubicacion.CodArea, Centro/Area.CodArea

### **3.3.1.3.4. Gestión Centro/Area**

#### **3.3.1.3.4.1. Insertar Centro/Area**

SI usuario es Administrador

BUSCAR registros Area y Centro en Almacén Area y Almacén

Centro con igual CodArea y CodCentro

SI alguno no encontrado

MOSTRAR ERROR

SINO

INSERTAR registro Centro/Area en Almacén Centro/Area

FIN SI

FIN SI

## **Términos del DD**

Almacenes: Almacén Centro/Area, Almacén Centro, Almacén Area

Registros: Centro/Area, Centro, Area

Campos: CodCentro, CodArea

### **3.3.1.3.4.2. Modificar Centro/Area**

SI usuario es Administrador

BUSCAR registros Area y Centro en Almacén Area y Almacén  
Centro con igual CodArea y CodCentro

SI alguno no encontrado

MOSTRAR ERROR

SINO

GUARDAR registro Centro/Area en Almacén Centro/Area

FIN SI

FIN SI

### **Términos del DD**

Almacenes: Almacén Centro/Area, Almacén Centro, Almacén Area

Registros: Centro/Area, Centro, Area

Campos: CodCentro, CodArea

### **3.3.1.3.4.3. Borrar Centro/Area**

SI usuario es Administrador

BORRAR registro Centro/Area de Almacén Centro/Area

FIN SI

### Términos del DD

Almacenes: Almacén Centro/Area

Registros: Centro/Area

Campos: CodCentro, CodArea

#### 3.3.1.3.4.4. Consultar Areas por Centro

BUSCAR primer registro Centro/Area en Almacén Centro/area con

CodCentro = Centro/Area.CodCentro

MIENTRAS registro <> EOF

    MOSTRAR registro Area en Almacén Area con Area.CodArea =

        Centro/Area.CodArea

    BUSCAR siguiente registro Centro/Area en Almacén Centro/area

        con CodCentro = Centro/Area.CodCentro

FIN MIENTRAS

### Términos del DD

Almacenes: Almacén Centro/Area, Almacén Area

Registros: Centro/Area, Area

Campos: CodCentro, CodArea, Area.Nombre, Dpto.Nombre

#### **3.3.1.3.4.5. Consultar Centros por Área**

BUSCAR primer registro Centro/Area en Almacén Centro/Area con

CodArea = Centro/Area.CodArea

MIENTRAS registro <> EOF

MOSTRAR registro Centro en Almacén Centro con

Centro.CodCentro = Centro/Area.CodCentro

BUSCAR siguiente registro Centro/Area en Almacén Centro/Area

con CodArea = Centro/Area.CodArea

FIN MIENTRAS

#### **Términos del DD**

Almacenes: Almacén Centro/Area, Almacén Centro

Registros: Centro/Area, Centro

Campos: CodCentro, CodArea, Centro.Nombre

#### **3.3.1.4. Gestión Centros**

##### **3.3.1.4.1. Insertar Centro**

SI usuario es Administrador

INSERTAR registro Centro en Almacén Centro

FIN SI

## **Términos del DD**

Almacenes: Almacén Centro

Registros: Centro

Campos: CodCentro, Nombre

### **3.3.1.4.2. Modificar Centro**

SI usuario es Administrador

GUARDAR registro Centro en Almacén Centro

SI CodCentro ha cambiado

ACTUALIZAR registros Centro/Area y Ubicacion de

Almacén Centro/Area y Almacén Ubicacion

FIN SI

FIN SI

## **Términos del DD**

Almacenes: Almacén Centro, Almacén Centro/Area, Almacén  
Ubicacion

Registros: Centro, Centro/Area, Ubicacion

Campos: CodCentro, Nombre

### **3.3.1.4.3. Borrar Centro**

SI usuario es Administrador

BUSCAR registros Centro/Area y Ubicacion de Almacén

Centro/Area y Almacén Ubicacion con mismo CodCentro

SI encontrado

MOSTRAR ERROR

SINO

BORRAR registro Centro de Almacén Centro

FIN SI

FIN SI

### **Términos del DD**

Almacenes: Almacén Centro, Almacén Centro/Area, Almacén

Ubicacion

Registros: Centro, Centro/Area, Ubicacion

Campos: CodCentro, Nombre

### **3.3.1.4. Consultar Areas por Dpto**

BUSCAR primer registro Area en Almacén Area con

Dpto = Area.Dpto

MIENTRAS registro <> EOF

MOSTRAR registro Area en Almacén Area

BUSCAR siguiente registro Area en Almacén Area

con con Dpto = Area.Dpto

FIN MIENTRAS

### Términos del DD

Almacenes: Almacén Area

Registros: Area

Campos: Area.Nombre, Area.Dpto

## 3.3.2. Gestión Fichas

### 3.3.2.1. Gestión Productos

#### 3.3.2.1.1. Insertar Producto

SI existe Producto con igual nombre

MOSTRAR ERROR

SINO

Añadir Producto a Almacén Producto

FIN SI

### **Términos del DD**

Almacenes: Almacén Producto

Registros: Producto

Campos: Nombre

#### **3.3.2.1.2. Modificar Producto**

SI no existe Producto con igual nombre

MOSTRAR ERROR

SINO

SI Producto.Nombre <> Nombre

ACTUALIZAR en cascada los registros de Almacén Ficha

FIN SI

Guardar Producto en Almacén Producto

FIN SI

### **Términos del DD**

Almacenes: Almacén Producto

Registros: Producto

Campos: Nombre

#### **3.3.2.1.3. Consultar Producto**

IR al primer registro de Almacén Producto

MIENTRAS NO Producto <> EOF  
MOSTRAR Producto  
Ir a siguiente registro de Almacén Producto  
FIN MIENTRAS

### Términos del DD

Almacenes: Almacén Producto  
Registros: Producto

#### 3.3.2.2. Gestión Entradas

##### 3.3.2.2.1. Insertar Entrada

BUSCAR Ficha en Almacén Ficha con Entrada.CodFicha =  
Ficha.CodFicha  
SI no encontrado  
MOSTRAR ERROR  
SINO  
AÑADIR Entrada a Almacén Entrada  
FIN SI  
FIN SI

### Términos del DD

Almacenes: Almacén Entrada, Almacén Ficha  
Registros: Entrada, Ficha

Campos: Entrada.CodFicha, Ficha.CodFicha

### **3.3.2.2.2. Modificar Entrada**

BUSCAR Ficha en Almacén Ficha con Entrada.CodFicha =

Ficha.CodFicha

SI no encontrado

MOSTRAR ERROR

SINO

GUARDAR Entrada en Almacén Entrada

FIN SI

FIN SI

### **Términos del DD**

Almacenes: Almacén Entrada, Almacén Ficha

Registros: Entrada, Ficha

Campos: Entrada.CodFicha, Ficha.CodFicha

### **3.3.2.2.3. Consultar Entradas Ficha**

BUSCAR primer registro Entrada de Almacén Entrada con

Entrada.CodFicha = CodFicha

MIENTRAS registro Entrada encontrado

**MOSTRAR Entrada**

Ir a siguiente registro Entrada de Almacén Entrada con  
Entrada.CodFicha = Codficha

**FIN MIENTRAS**

### **Términos del DD**

Almacenes: Almacén Entrada, Almacén Ficha

Registros: Entrada, Ficha

Campos: Entrada.CodFicha, Ficha.CodFicha

#### **3.3.2.2.4. Consultar Entradas Fichas**

IR al primer registro de Almacén Fichas

MIENTRAS fichas <> EOF

BUSCAR primer registro Entrada de Almacén Entrada con

Entrada.CodFicha = Ficha.CodFicha

MIENTRAS registro Entrada encontrado

**MOSTRAR Entrada**

Ir a siguiente registro Entrada en Almacén Entrada con

Entrada.CodFicha = Ficha.Codficha

**FIN MIENTRAS**

**FIN MIENTRAS**

### **Términos del DD**

Almacenes: Almacén Entrada, Almacén Ficha

Registros: Entrada, Ficha

Campos: Entrada.CodFicha, Ficha.CodFicha

### **3.3.2.3. Gestión Salidas**

#### **3.3.2.3.1. Insertar Salida**

Llamar a Verificar Salida (Salida)

SI retorno = OK

    INSERTAR Salida en Almacén Salida

    SINO

        MOSTRAR ERROR

    FIN SI

$$\text{Capacidad\_Salida} = \text{Capacidad} * \text{Unidades}$$

### **Términos del DD**

Almacenes: Almacén Salida

Registros: Salida

### **3.3.2.3.2. Modificar Salida**

Llamar a Verificar Salida (Salida)

SI retorno = OK

    INSERTAR Salida en Almacén Salida

SINO

    MOSTRAR ERROR

FIN SI

Capacidad\_Salida = Capacidad \* Unidades

### **Términos del DD**

Almacenes: Almacén Salida

Registros: Salida

### **3.3.2.3.3. Consultar Salidas Ficha**

BUSCAR primer registro Salida en Almacén Salida con

    Salida.CodFicha = CodFicha

MIENTRAS registro Salida encontrado

    MOSTRAR Salida

Ir a siguiente registro Salida en Almacén Salida con  
Salida.CodFicha = Codficha

FIN MIENTRAS

### Términos del DD

Almacenes: Almacén Salida, Almacén Ficha

Registros: Salida, Ficha

Campos: Salida.CodFicha, Ficha.CodFicha

#### 3.3.2.3.4. Consultar Salidas Fichas

IR al primer registro Salida de Almacén Salida

MIENTRAS fichas <> EOF

BUSCAR primer registro Salida en Almacén Salida con

Salida.CodFicha = Ficha.CodFicha

MIENTRAS registro Salida encontrado

MOSTRAR Salida

Ir a siguiente registro Salida en Almacén Salida con

Salida.CodFicha = Ficha.Codficha

FIN MIENTRAS

FIN MIENTRAS

## **Términos del DD**

Almacenes: Almacén Salida, Almacén Ficha

Registros: Salida, Ficha

Campos: Salida.CodFicha, Ficha.CodFicha

### **3.3.2.3.5. Verificar Salida**

Buscar registro Ficha en Almacén Ficha con CodFicha = Ficha.CodFicha

SI registro Ficha no encontrado

DEVOLVER ERROR

FIN SI

IR al primer registro Entrada en Almacén Entrada con Entrada.CodFicha

= CodFicha y con Entrada.Capacidad = Capacidad y

Entrada.g\_ml = g\_ml

MIENTRAS Entrada <> EOF

BUSCAR primer registro en Exist con Exist.Capacidad =

Entrada.Capacidad Exist.g\_ml = Entrada.g\_ml

SI no encontrado

INSERTAR registro Exist en Almacén Exist con

Entrada.Unidades, Entrada.Capacidad, Entrada.g\_ml

FIN SI

Exist.Unidades += Entrada.Unidades

IR a siguiente registro Entrada en Almacén Entrada con  
Entrada.CodFicha = CodFicha y con Entrada.Capacidad =  
Capacidad y Entrada.g\_ml = g\_ml

FIN MIENTRAS

IR al primer registro Salida en Almacén Salida con Salida.CodFicha  
= CodFicha y con Salida.Capacidad = Capacidad y  
Salida.g\_ml = g\_ml

MIENTRAS Salida <> EOF

BUSCAR primer registro en Exist con Exist.Capacidad =

Salida.Capacidad Exist.g\_ml = Salida.g\_ml

SI no encontrado

DEVOLVER ERROR

SINO

Exist.Unidades -= Salida.Unidades

SI Exist.Unidades < 0

DEVOLVER ERROR

FIN SI

IR a siguiente registro Salida en Almacén Salida con

Salida.CodFicha = CodFicha y con

Salida.Capacidad = Capacidad y Salida.g\_ml = g\_ml

FIN SI

FIN MIENTRAS

### **Términos del DD**

Almacenes: Almacén Entrada, Almacén Salida, Almacén Ficha

Registros: Entrada, Salida, Ficha

Campos: Entrada.CodFicha, Salida.CodFicha, Ficha.CodFicha,

Entrada.g\_ml, Salida.g\_ml, Entrada.Unidades, Salida.Unidades,

Entradas.Capacidad, Salidas.Capacidad, CodFicha, g\_ml

### **Términos Locales**

Almacenes: Almacén Exist

Registros: Exist

Campos: Exist.Unidades, Exist.Capacidad, Exist.g\_ml

#### **3.3.2.4. Insertar Ficha**

BUSCAR registro Ubicacion en Almacén Ubicacion con

Ubicacion.CodUbicacion = CodUbicacion

SI no encontrado

LLAMAR a Insertar Ubicacion

FIN SI

BUSCAR registro Producto en Almacén Producto con Producto.Nombre

= Producto

SI no encontrado

LLAMAR a Insertar Producto

FIN SI

BUSCAR registro Calidad en Almacén Calidad con Calidad.Nombre =

Calidad

SI no encontrado

PREGUNTAR “¿Insertar Calidad(S/N)?”

SI RESPUESTA = SI

AÑADIR Calidad en Almacén Calidad

SINO

DEVOLVER ERROR

FIN SI

FIN SI

BUSCAR registro Marca en Almacén Marca con Marca.Nombre = Marca

SI no encontrado

PREGUNTAR “¿Insertar Marca(S/N)?”

SI RESPUESTA = SI

AÑADIR Marca en Almacén Marca

SINO

DEVOLVER ERROR

FIN SI

FIN SI

SI usuario tiene permiso para insertar datos del área

Ubicacion.CodArea

INSERTA Ficha en Almacén Fichas

SINO

DEVUELVE ERROR\_USUARIO

FIN SI

### Términos del DD

Almacenes: Almacén Ficha, Almacén Ubicacion, Almacén Producto,

Almacén Calidad, Almacén Marca

Registros: Ficha, Ubicacion, Producto, Calidad, Marca

Campos: Ficha.CodUbicacion, Ubicacion.CodUbicacion,

Ubicacion.CodArea, Ficha.Producto, Ficha.Marca,

Ficha.Calidad, Producto.Nombre, Calidad.Nombre,

Marca.Nombre.

#### 3.3.2.5. Modificar Ficha

BUSCAR registro Ubicacion en Almacén Ubicacion con

Ubicacion.CodUbicacion = CodUbicacion

SI no encontrado

LLAMAR a Insertar Ubicacion

FIN SI

BUSCAR registro Producto en Almacén Producto con Producto.Nombre

= Producto

SI no encontrado

LLAMAR a Insertar Producto

FIN SI

BUSCAR registro Calidad en Almacén Calidad con Calidad.Nombre =

Calidad

SI no encontrado

PREGUNTAR “¿Insertar Calidad(S/N)?”

SI RESPUESTA = SI

AÑADIR Calidad en Almacén Calidad

SINO

DEVOLVER ERROR

FIN SI

FIN SI

BUSCAR registro Marca en Almacén Marca con Marca.Nombre = Marca

SI no encontrado

PREGUNTAR “¿Insertar Marca(S/N)?”

SI RESPUESTA = SI

AÑADIR Marca en Almacén Marca

SINO

DEVOLVER ERROR

FIN SI

FIN SI

SI usuario tiene permiso para insertar datos del área

Ubicacion.CodArea  
GUARDAR Ficha en Almacén Fichas  
SINO  
DEVUELVE ERROR\_USUARIO  
FIN SI

### Términos del DD

Almacenes: Almacén Ficha, Almacén Ubicacion, Almacén Producto,  
Almacén Calidad, Almacén Marca  
Registros: Ficha, Ubicacion, Producto, Calidad, Marca  
Campos: Ficha.CodUbicacion, Ubicacion.CodUbicacion,  
Ubicacion.CodArea, Ficha.Producto, Ficha.Marcas,  
Ficha.Calidad, Producto.Nombre, Calidad.Nombre,  
Marca.Nombre.

#### 3.3.2.6. Consultar Fichas

Busqueda = ""  
SI NombreProducto <> NULL  
    Busqueda = Busqueda + "Ficha.Producto = NombreProducto"

FIN SI

SI CodArea <> NULL  
    Busqueda = Busqueda + "Ubicacion.CodArea = CodArea and"

Ficha.Ubicacion = CodUbicacion”

FIN SI

SI CodUbicacion <> NULL

    Busqueda = Busqueda + “Ficha.Ubicacion = CodUbicacion”

FIN SI

IR a primer registro Ficha en Almacén Ficha que satisfaga Consulta

MIENTRAS Ficha <> EOF

    MOSTRAR Ficha

    IR a siguiente registro Ficha en Almacén Ficha que satisfaga

        Consulta

FIN MIENTRAS

### Términos del DD

Almacenes: Almacén Ficha, Almacén Ubicacion

Registros: Ficha, Ubicacion

Campos: Ficha.CodUbicacion, Ubicacion.CodUbicacion,

Ubicacion.CodArea, Ficha.Producto, Ficha.Marcas,

Ficha.Calidad, Producto.Nombre

#### 3.3.2.7. Consultar Existencias

Tabla Temporal Almacén Exist

BUSCAR primer registro Entrada de Almacén Entrada con  
Entrada.CodFicha = CodFicha y que entre en el patrón Consulta  
MIENTRAS Entrada <> EOF

BUSCAR primer registro Exist en Almacén Exist con CodFicha =  
Entrada.CodFicha y misma capacidad y g\_ml  
SI encontrado

Exist.Unidades += Entrada.Unidades

SINO

INSERTAR en Almacén Exist registro Exist = Entrada

FIN SI

BUSCAR siguiente registro Entrada de Almacén Entrada con

Entrada.CodFicha = CodFicha y que entre en el patrón  
Consulta

FIN MIENTRAS

BUSCAR primer registro Salida de Almacén Salida con Salida.CodFicha  
= CodFicha y que entre en el patrón Consulta

MIENTRAS Salida <> EOF

BUSCAR primer registro Exist en Almacén Exist con CodFicha =  
Salida.CodFicha y misma capacidad y g\_ml  
SI encontrado

Exist.Unidades -= Salida.Unidades

SINO

//El proceso Verificar Ficha se encargará de que esto no

//pase nunca

Devolver ERROR

FIN SI

BUSCAR siguiente registro Salida de Almacén Salida con  
Salida.CodFicha = CodFicha y que entre en el patrón  
Consulta

FIN MIENTRAS

MOSTRAR Exist

### Términos del DD

Almacenes: Almacén Entrada, Almacén Salida

Registros: Entrada, Salida

Campos: Entrada.CodFicha, Salida.CodFicha, Entrada.Capacidad,  
Salida.Capacidad, Entrada.g\_ml, Salida.g\_ml.

### Términos locales

Almacenes: Almacén Exist

Registros: Exist

Campos: Exist.Capacidad, Exist.g\_ml, Exist.Unidades

### **3.3.3. Gestión Ubicacion**

#### **3.3.3.1. Insertar Ubicacion**

SI usuario tiene permisos en ese Area

BUSCAR registros Area y Centro en Almacén Area y Almacén

Centro con mismos CodArea y CodCentro

SI encontrados

INSERTAR registro Ubicacion en Almacén Ubicacion

SINO

MOSTRAR ERROR

FIN SI

SINO

MOSTRAR ERROR\_USUARIO

FIN SI

#### **Términos del DD**

Almacenes: Almacén Ubicacion, Almacén Area, Almacén Centro

Registros: Ubicacion, Area, Centro

Campos: CodArea, CodCentro

#### **3.3.3.2. Modificar Ubicacion**

SI usuario tiene permisos en ese Area

BUSCAR registros Area y Centro en Almacén Area y Almacén

Centro con mismos CodArea y CodCentro

SI encontrado

SI CodUbicacion ha cambiado

ACTUALIZAR en cascada registros Ficha en

Almacén Ficha

FIN SI

GUARDAR registro Ubicacion en Almacén Ubicacion

SINO

MOSTRAR ERROR

FIN SI

SINO

MOSTRAR ERROR\_USUARIO

FIN SI

### Términos del DD

Almacenes: Almacén Ubicacion, Almacén Ficha

Registros: Ubicacion, Ficha

Campos: CodUbicacion

#### 3.3.3.3. Borrar Ubicacion

SI usuario tiene permisos en ese Area

BUSCAR primer registro Ubicacion en Almacén Ubicacion

Con Ubicacion.CodUbicacion = CodUbicacion

SI encontrado

BUSCAR primer registro Ficha en Almacén Ficha con

Ficha.Ubicacion = CodUbicacion

SI no encontrado  
    Borrar registro Ubicacion de Almacén  
    Ubicacion  
SINO  
    MOSTRAR ERROR  
    FIN SI  
SINO  
    MOSTRAR ERROR  
    FIN SI  
SINO  
    MOSTRAR ERROR\_USUARIO  
    FIN SI

### Términos del DD

Almacenes: Almacén Ubicacion, Almacén Ficha  
Registros: Ubicacion, Ficha  
Campos: Ubicacion.CodUbicacion, Ficha.Ubicacion

#### 3.3.3.4. Consultar Ubicacion por Área

SI usuario tiene permisos en ese Area  
    BUSCAR primer registro Ubicacion en Almacén Ubicacion  
    con Ubicacion.CodArea = CodArea  
    MIENTRAS Ubicacion <> EOF

**MOSTRAR Ubicacion**

BUSCAR siguiente registro Ubicacion en Almacén

Ubicacion con Ubicacion.CodArea = CodArea

**FIN MIENTRAS**

**SINO**

**MOSTRAR ERROR**

**FIN SI**

### **Términos del DD**

Almacenes: Almacén Ubicacion, Almacén Area

Registros: Ubicacion, Area

Campos: CodArea

#### **3.3.3.5. Consultar Ubicacion por Centro y Area**

SI usuario tiene permisos en ese Area

BUSCAR primer registro Ubicacion en Almacén Ubicacion

con Ubicacion.CodArea = CodArea y con

Ubicacion.CodCentro = CodCentro

MIENTRAS Ubicacion <> EOF

**MOSTRAR Ubicacion**

BUSCAR siguiente registro Ubicacion en Almacén

Ubicacion con Ubicacion.CodArea = CodArea y

con Ubicacion.CodCentro = CodCentro

**FIN MIENTRAS**

**SINO**

**MOSTRAR ERROR**

**FIN SI**

## **Términos del DD**

Almacenes: Almacén Ubicacion, Almacén Area, Almacén Centro

Registros: Ubicacion, Area, Centro

Campos: CodArea, CodCentro

### 3.1. Diagrama Entidad/Relación

El diagrama Entidad/Relación resultante del análisis es el siguiente:

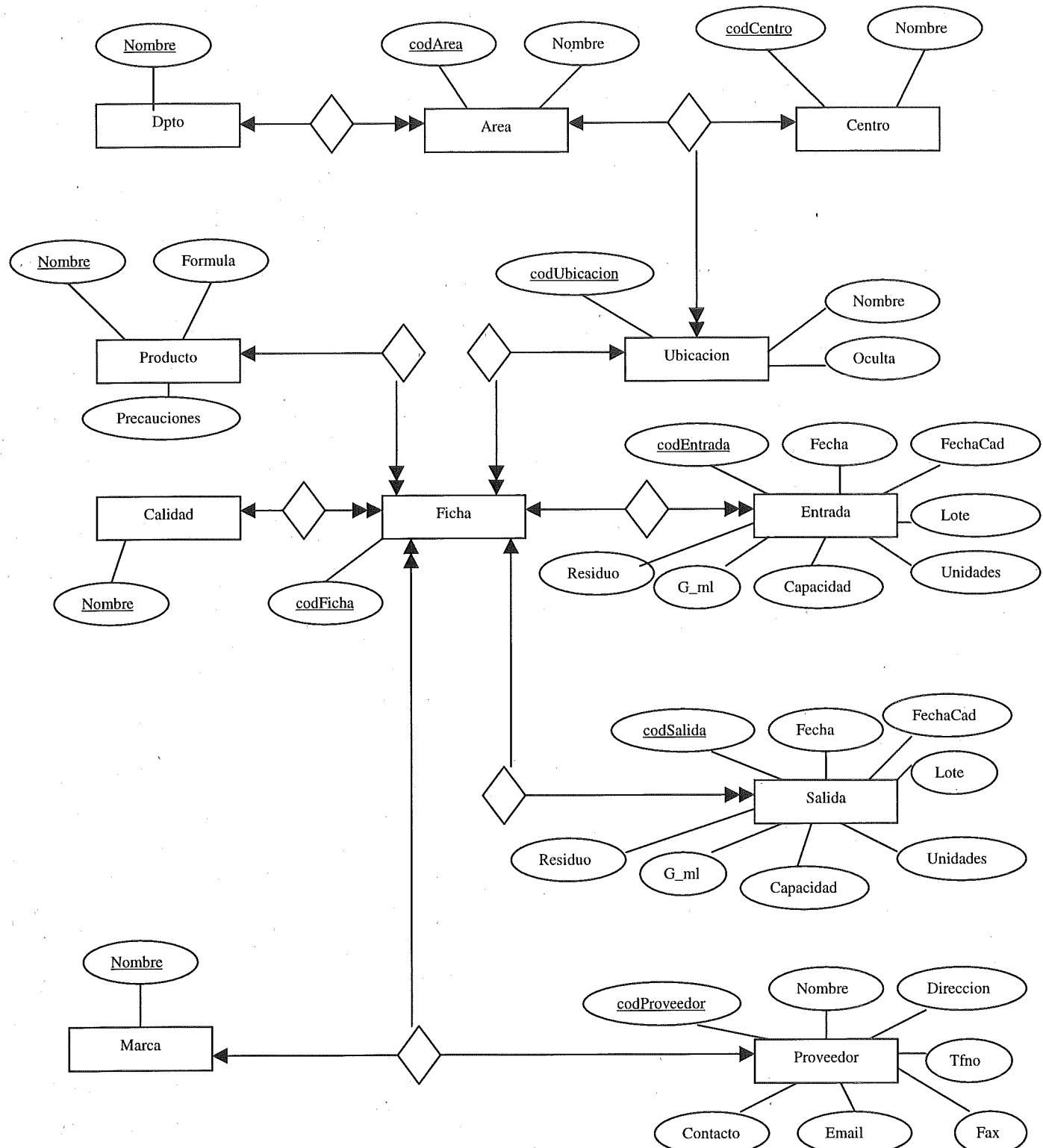


Figura 1. Diagrama Entidad/Relación

## **4. Especificaciones de diseño**

La fase de diseño que se proponía en la metodología de Yourdon tampoco se ha realizado, de nuevo por razones de tiempo. Se elaborará posteriormente siguiendo una metodología orientada a objetos.

En este apartado se detallarán brevemente las decisiones tomadas en la construcción del sistema que cumpla los requisitos especificados anteriormente.

### **4.1. Sistema Operativo y Sistema Gestor de Base de Datos**

Los datos se guardarán en una base de datos única situada en un ordenador central perteneciente a la red de la Universidad de Burgos.

Por razones económicas, y dado que el número de accesos concurrentes será pequeño, el sistema operativo elegido es Windows NT Server versión 4.0 en inglés, y el Sistema Gestor de Base de Datos (de ahora en adelante SGBD) es el SQL Server 6.5 de Microsoft en inglés.

El ordenador se llama *camelot*, y es un servidor dedicado exclusivamente a la base de datos. Es un Pentium II a 400 MHz, con 128 MB de memoria RAM.

### **4.2. Configuración de la Base de Datos**

#### **4.2.1. Usuarios**

El SGBD SQL Server es multiusuario, y permite accesos concurrentes.

Tiene la posibilidad de permitir el acceso a usuarios del dominio de Windows NT, a usuarios creados específicamente desde SQL Server, o a ambos (acceso mixto).

Dado que los usuarios no tienen por qué tener acceso para iniciar una sesión en los ordenadores del dominio, la opción elegida es el acceso a cuentas creadas desde SQL Server.

#### **4.2.2. Tablas**

Siguiendo el diagrama Entidad/Relación, cada entidad corresponde a una tabla. Además se han creado las tablas *centro\_area*, procedente de la relación entre *ubicación*, *centro* y *area*, y *prov\_marca*, procedente de la relación entre *ficha*, *proveedor* y *marca*. Esta

última relación se modificó posteriormente debido a un cambio en los requisitos. Por comodidad los usuarios querían que introducir el proveedor fuese opcional, es decir, el campo *proveedor* de la tabla *ficha* puede ser nulo, por lo que este campo no se relacionó temporalmente con la tabla *prov\_marca*. Más tarde, esta solución se cambió por restablecer la relación inicial y añadir en la tabla *proveedor* el registro con nombre “<Ninguno>” y código ‘0’. Los formularios de gestión de proveedores no podrán ver este registro.

Los scripts de creación de las tablas son los siguientes:

### **dpto**

```
create table dpto (
    nombre char(25),
    constraint PK_dpto primary key clustered (nombre))
```

### **area**

```
create table area (
    codarea char(15),
    nombre varchar(40),
    dpto char(25),
    constraint PK_area primary key clustered (codarea),
    constraint FK_dpto_area foreign key (dpto) references dpto(nombre),
    constraint UNQ_area unique(nombre, dpto))
```

### **centro**

```
create table centro (
    codcentro char(2),
    nombre varchar(40),
    constraint PK_centro primary key clustered (codcentro),
```

constraint UNQ\_centro unique (nombre))

### **centro\_area**

```
create table centro_area (
    centro char(2),
    area char(15),
    constraint PK_centro_area primary key (area, centro),
    constraint FK_centro_ca foreign key (centro)
        references centro (codcentro),
    constraint FK_area_ca foreign key (area)
        references area(codarea))
```

### **ubicacion**

```
create table ubicacion(
    codubicacion char(4),
    nombre varchar(30),
    area char(15),
    centro char(2),
    oculta char(1),
    constraint PK_ubicacion primary key nonclustered (codubicacion),
    constraint FK__centro_area_ub foreign key (area, centro)
        references centro_area(area, centro),
    constraint CK_oculta check ((oculta = 's') or (oculta = 'n')),
    constraint UNQ_ubicacion unique(nombre, centro, area))
```

### **calidad**

```
create table calidad (
    nombre char(10),
    constraint PK_calidad primary key clustered (nombre))
```

### **producto**

```
create table producto (
    nombre char(40),
    formula char(30),
    precauciones varchar(40) null,
    constraint PK_producto primary key clustered (nombre))
```

### **proveedor**

```
create table proveedor (
    codproveedor char(4),
    nombre varchar(40),
    direccion varchar(40) null,
    tfno char(9) null,
    fax char(9) null,
    email varchar(40) null,
    contacto varchar(40) null,
    constraint PK_codproveedor primary key clustered (codproveedor),
    constraint UNQ_proveedor unique(nombre))
```

### **marca**

```
create table marca (
    nombre char(20),
```

```
constraint PK_marca primary key clustered (nombre))
```

### **prov\_marca**

```
create table prov_marca (
    proveedor char(4),
    marca char(20),
    constraint PK_prov_marca primary key clustered (proveedor, marca),
    constraint FK_proveedor_pm foreign key (proveedor)
        references proveedor (codproveedor),
    constraint FK_marca_pm foreign key (marca)
        references marca (nombre))
```

### **ficha**

```
create table ficha (
    codficha char(10),
    calidad char(10),
    ubicacion char(4),
    proveedor char(4),
    marca char(20),
    producto char(40),
    constraint PK_ficha primary key (codficha),
    constraint FK_calidad_ficha foreign key (calidad)
        references calidad (nombre),
    constraint FK_ubicacion_ficha foreign key (ubicacion)
        references ubicacion (codubicacion),
    constraint FK_marca_ficha foreign key (marca)
```

```
    references prov_marca(marca),
constraint FK_proveedor foreign key (proveedor)
    references prov_marca(proveedor),
constraint FK_producto_ficha foreign key (producto)
    references producto (nombre))
```

### **entrada**

```
create table entrada (
codentrada char(20),
ficha char(10),
fecha datetime,
fechacaducidad datetime null,
lote char(10) null,
unidades decimal(6),
capacidad decimal(11, 4),
g_ml char(2),
residuo char(1),
constraint PK_entrada primary key (codentrada),
constraint FK_ficha_entrada foreign key (ficha)
    references ficha (codficha),
constraint CK_s_n_entrada check ((residuo = 's') or (residuo = 'n')))
```

### **salida**

```
create table salida (
codsalida char(20),
ficha char(10),
```

```

fecha datetime,
fechacaducidad datetime null,
lote char(10) null,
unidades decimal(6),
capacidad decimal(11, 4),
g_ml char(2),
residuo char(1),
constraint PK_salida primary key (codsalida),
constraint FK_ficha_salida foreign key (ficha)
    references ficha (codficha),
constraint CK_s_n_salida check ((residuo ='s') or (residuo = 'n'))

```

Para resolver el problema de las claves primarias<sup>1</sup> al añadir un nuevo registro, se crea una tabla llamada *maximo*, sobre la cual nadie tendrá ningún permiso. Tendrá un campo para cada una de las siguientes tablas: *centro*, *ubicación*, *ficha*, *proveedor*, *entrada* y *salida*. En cada uno de estos campos se guarda el código del último centro, ubicación y ficha, respectivamente, que se añadió a la base de datos. Habrá unos procedimientos que aumentarán en 1 el valor de cada uno de estos campos (según la tabla a la que se quiera añadir el registro) y devolverán el nuevo valor de la clave primaria. Los usuarios sí tendrán permiso de ejecución sobre estos procedimientos.

### **maximo**

```

create table maximo(
    centro char(2),
    ubicacion char(4),

```

---

<sup>1</sup> Clave primaria: es un campo de la tabla que tiene un valor único (por ejemplo, cada ficha tiene un código de ficha) y no se puede repetir en dos filas diferentes (no puede haber dos fichas con el mismo código de ficha).

```
proveedor char(4),
```

```
ficha char(10),
```

```
entrada char(20),
```

```
salida char(20))
```

```
insert into maximo values (
```

```
'00',      '0000',      '0000',      '0000000000',      '00000000000000000000000000000000',  
'00000000000000000000000000000000')
```

### **Procedimiento num\_centro**

```
create procedure dbo.num_centro
```

```
as
```

```
declare @aux char(2),
```

```
@salida decimal(2)
```

```
select @aux = centro from maximo
```

```
select @salida = convert(decimal(2), @aux)
```

```
if (select @salida) = 99
```

```
    select @salida = 0
```

```
else
```

```
    select @salida = @salida + 1
```

```
select @aux = convert(char(2), @salida)
```

```
update maximo
```

```
    set centro = @aux
```

```
return @salida
```

go

### **Procedimiento num\_ubicacion**

```
create procedure dbo.num_ubicacion
as
declare @aux char(4),
@salida decimal(4)
select @aux = ubicacion from maximo
select @salida = convert(decimal(4), @aux)
if (select @salida) = 9999
    select @salida = 0
else
    select @salida = @salida + 1

select @aux = convert(char(4), @salida)
update maximo
    set ubicacion = @aux
return @salida
go
```

### **Procedimiento num\_proveedor**

```
create procedure dbo.num_proveedor
as
declare @aux char(4),
@salida decimal(4)
select @aux = proveedor from maximo
```

```
select @salida = convert(decimal(4), @aux)

if (select @salida) = 9999

    select @salida = 0

else

    select @salida = @salida + 1

select @aux = convert(char(4), @salida)

update maximo

set proveedor = @aux

return @salida

go
```

### **Procedimiento num\_ficha**

```
create procedure dbo.num_ficha

as

declare @aux char(10),

@salida decimal(10)

select @aux = ficha from maximo

select @salida = convert(decimal(10), @aux)

if (select @salida) = 9999999999

    select @salida = 0

else

    select @salida = @salida + 1

select @aux = convert(char(10), @salida)

update maximo
```

```
    set ficha = @aux  
    return @salida  
    go
```

### **Procedimiento num\_entrada**

```
create procedure dbo.num_entrada  
as  
declare @aux char(20),  
@salida decimal(20)  
select @aux = entrada from maximo  
select @salida = convert(decimal(10), @aux)  
if (select @salida) = 99999999999999999999  
    select @salida = 0  
else  
    select @salida = @salida + 1  
  
select @aux = convert(char(10), @salida)  
update maximo  
set entrada = @aux  
return @salida  
go
```

### **Procedimiento num\_salida**

```
create procedure dbo.num_salida  
as  
declare @aux char(20),
```

```

@salida decimal(20)

select @aux = salida from maximo

select @salida = convert(decimal(10), @aux)

if (select @salida) = 99999999999999999999

    select @salida = 0

else

    select @salida = @salida + 1


select @aux = convert(char(10), @salida)

update maximo

    set salida = @aux

return @salida

go

```

#### 4.2.3. Vistas

Como cada área sólo puede gestionar sus fichas, habrá una vista de las tablas *ubicacion*, *ficha*, *entrada* y *salida* por cada área. Sobre estas tablas nadie tendrá ningún tipo de permiso, excepto el usuario administrador(sa), que tiene acceso total a la base de datos. Cada vista se llamará <usuario>\_<tabla> Ejemplo: Para el usuario *bioqbm*, tabla *ubicacion*, la vista se llamará *bioqbm\_ubicacion*.

Además habrá una vista existencias, la cual restará las unidades de las salidas a las de las entradas para las filas con valores iguales de los campos *ficha*, *g\_ml* y *capacidad*.

Se crean además las vistas *ficha.todos* y *exist.todos*, que reúnen las fichas visibles (las que se encuentran en una localización no oculta) de todas las áreas, y las existencias de esas fichas, respectivamente.

El script de creación de estas vistas es el siguiente:

**exist**

create view exist as

```
select a.ficha, a.unidades - isnull (b.unidades, 0) total, a.capacidad,  
      a.g_ml  
  from (select ficha, sum(unidades) unidades, capacidad, g_ml  
        from entrada  
       group by ficha, g_ml, capacidad) a left outer join  
  (select ficha, sum (unidades) unidades, capacidad, g_ml  
        from salida  
       group by ficha, g_ml, capacidad) b  
  on a.ficha = b.ficha  
 and a.g_ml = b.g_ml  
 and a.capacidad = b.capacidad
```

go

**ficha.todos**

create view ficha.todos as

```
select f.codficha, f.calidad, f.proveedor, f.marca, f.producto,  
      p.formula, u.centro, u.area  
  from ficha f, ubicacion u, producto p  
 where p.nombre = f.producto  
   and u.codubicacion = f.ubicacion  
   and u.oculta = 'n'
```

go

### **exist.todos**

```
create view exist.todos as  
select e.*  
from exist e, ficha.todos f  
where f.codficha = e.ficha
```

```
go
```

Una vez creadas las tablas, habrá que conceder ciertos permisos a todos los usuarios sobre algunas de ellas:

Permiso de solo lectura (select):

```
grant select on dpto to usuarios  
grant select on area to usuarios  
grant select on centro to usuarios  
grant select on centro_area to usuarios  
grant select on ficha.todos to usuarios  
grant select on exist.todos to usuarios
```

Permiso de lectura/inserción/modificación:

```
grant select, insert, update, delete on marca to usuarios  
grant select, insert, update, delete on proveedor to usuarios  
grant select, insert, update, delete on prov_marca to usuarios  
grant select, insert, update, delete on producto to usuarios
```

grant select, insert, update, delete on calidad to usuarios

Permisos que se deben quitar:

revoke all on ubicacion from public

revoke all on ficha from public

revoke all on entrada from public

revoke all on salida from public

revoke all on exist from public

#### 4.2.4. Crear un nuevo usuario en la base de datos

Se ha creado un procedimiento *crear\_usuario*, que tiene como parámetros el nombre de usuario y la contraseña. Ejemplo: *crear\_usuario* ‘quimicas’, ‘h3i2j1’

Este procedimiento añade el nombre de usuario y un login con el mismo nombre y con dicha password<sup>2</sup> a la base de datos. Añade el usuario al grupo Usuarios. Acto seguido crea las vistas *<usuario>\_ubicacion*, *<usuario>\_ficha*, *<usuario>\_entrada*, *<usuario>\_salida*, *<usuario>\_exist*.

Después asigna los permisos apropiados al usuario sobre las vistas creadas y los procedimientos:

- Permisos de lectura/inserción/modificación sobre *producto*, *<usuario>\_ficha*, *<usuario>\_entrada*, *<usuario>\_salida*, *<usuario>\_exist*.
- Todos los permisos sobre *<usuario>\_ubicacion*.
- Se deniega el permiso de borrado sobre *<usuario>\_ficha*, *<usuario>\_entrada* y *<usuario>\_salida*.

---

<sup>2</sup> En SQL Server un usuario puede tener varios logins, cada uno de éstos con su password (contraseña). Para poder conectarse a la base de datos, un usuario necesita como mínimo un login.

- Permisos de ejecución sobre los procedimientos *num\_centro*, *num\_ubicacion*, *num\_ficha*, *num\_proveedor*, *num\_entrada*, *num\_salida*.

El script de creación del procedimiento es el siguiente:

```
CREATE PROCEDURE crear_usuario
    @usuario varchar(20),
    @password varchar(20)
as
    --Quitar espacios en blanco
    select @usuario = ltrim (rtrim(@usuario))
    select @password = ltrim(rtrim(@password))

    --Crear el usuario en la base de datos
    exec sp_addlogin @usuario, @password, 'DBLab'
    exec sp_adduser @usuario, @usuario, 'Usuarios'

    --Creacion de las vistas para el usuario
    exec ('create view dbo.' + @usuario + '_ubicacion as
        select * from ubicacion
        where area like "' + @usuario + "'"
            with check option')

    exec ('create view dbo.' + @usuario + '_ficha as
        select f.*, p.formula
```

```
from ficha f, dbo.' + @usuario + '_ubicacion u, producto p  
where f.ubicacion = u.codubicacion  
and p.nombre = f.producto  
with check option')
```

```
exec ('create view dbo.' + @usuario + '_entrada as  
select e.*  
from entrada e, dbo.' + @usuario + '_ficha f  
where e.ficha = f.codficha  
with check option')
```

```
exec ('create view dbo.' + @usuario + '_salida as  
select s.*  
from salida s, dbo.' + @usuario + '_ficha f  
where s.ficha = f.codficha  
with check option')
```

```
exec ('create view ' + @usuario + '_exist as  
select e.*  
from ' + @usuario + '_ficha f, exist e  
where f.codficha = e.ficha')
```

--Ahora los permisos

```
exec ('grant select, insert, update on dbo.producto to ' + @usuario)  
exec ('grant all on dbo.' + @usuario + '_ubicacion to ' + @usuario)  
exec ('grant select, insert, update on dbo.' + @usuario + '_ficha to ' + @usuario)
```

```
exec ('grant select, insert, update on dbo.' + @usuario + '_entrada to ' + @usuario)
exec ('grant select, insert, update on dbo.' + @usuario + '_salida to ' + @usuario)
exec ('grant select on dbo.' + @usuario + '_exist to ' + @usuario)
```

--Los que se quitan

```
exec ('revoke delete on dbo.' + @usuario + '_ficha from ' + @usuario)
exec ('revoke delete on dbo.' + @usuario + '_entrada from ' + @usuario)
exec ('revoke delete on dbo.' + @usuario + '_salida from ' + @usuario)
```

--Los procedimientos

```
exec ('grant execute on dbo.num_centro to ' + @usuario)
exec ('grant execute on dbo.num_ubicacion to ' + @usuario)
exec ('grant execute on dbo.num_proveedor to ' + @usuario)
exec ('grant execute on dbo.num_ficha to ' + @usuario)
exec ('grant execute on dbo.num_entrada to ' + @usuario)
exec ('grant execute on dbo.num_salida to ' + @usuario)
```

--Commit

go

Los datos sobre las tablas *area* y *dpto* no se insertan, se deben introducir manualmente en la base de datos, ya que no es normal que cambie la organización en áreas y en departamentos de la Universidad.

Como ampliación de este proyecto se puede implementar la gestión de áreas, departamentos y centro por programa, así como la gestión de usuarios.

#### **4.2.5. Borrar un usuario de la base de datos**

Al igual que para la creación de un usuario, se ha creado un procedimiento para borrar un usuario, que primero borrara todas las vistas de ese usuario y luego quitará el login y el usuario de la base de datos.

Los datos sobre ese usuario(área) no se borraran de la base de datos.

El script de creación del procedimiento es el siguiente:

```
CREATE PROCEDURE borrar_usuario
    @usuario varchar(20)
as
--Quitar espacios en blanco
select @usuario = ltrim (rtrim(@usuario))

--Quitar el usuario de la base de datos
exec sp_dropuser @usuario
exec sp_droplogin @usuario

--Borrar las vistas para el usuario
exec ('drop view dbo.' + @usuario + '_ubicacion')
exec ('drop view dbo.' + @usuario + '_ficha')
exec ('drop view dbo.' + @usuario + '_exist')
exec ('drop view dbo.' + @usuario + '_entrada')
exec ('drop view dbo.' + @usuario + '_salida')

--Commit
go
```

#### 4.2.6. Triggers<sup>3</sup>

En la base de datos se utiliza un trigger en la tabla salidas. Al insertar y modificar, comprueba que el número de unidades es menor o igual al de las existencias. Además comprueba que hay alguna entrada para la misma ficha, con la misma capacidad en gramos o mililitros.

El script de creación del trigger es el siguiente:

```
create trigger dbo.ins_salida
on dbo.salida
for insert, update
as
declare @total int
select @total = count (e.ficha)
from exist e, inserted i
where e.ficha = i.ficha
and e.g_ml = i.g_ml
and e.capacidad = i.capacidad
and e.total < 0

--Si se ha sacado mas contenido del que habia
--se aborta la transaccion
if (@total > 0)
begin
print 'ERROR. No hay tantas existencias'
```

---

<sup>3</sup> Trigger: disparador. Es un conjunto de sentencias que se ejecutan automáticamente al realizar una operación (inserción, modificación, borrado) sobre una tabla.

```
rollback transaction  
end  
  
else  
  
begin  
  
select @total = count (e.ficha)  
from entrada e, inserted i  
where e.ficha = i.ficha  
and e.g_ml = i.g_ml  
and e.capacidad = i.capacidad  
  
  
if (@total = 0)  
begin  
raiserror ('ERROR. No hay tantas existencias', 10, 1)  
rollback transaction  
end  
  
end
```

GO

#### 4.2.7. Backups<sup>4</sup>

Todavía no se ha propuesto ninguna solución para la gestión de las copias de seguridad. Lo más probable es que se hagan copias periódicas de los datos en CD-ROM.

---

<sup>4</sup> Backup: Copia de Seguridad de los datos.

#### 4.2.8. Cliente

El cliente se implementará en el lenguaje Visual Basic 5.0 Enterprise Edition de Microsoft, en español. Se ha elegido este lenguaje por su facilidad (y rapidez) de implementación, por estar diseñado para los sistemas operativos Windows de 32 bits (95/98/NT) y por ser el lenguaje de programación más utilizado en el desarrollo de aplicaciones de gestión.

El acceso remoto a los datos se hará por ODBC<sup>5</sup>, y se utilizará una conexión ODBCDirect<sup>6</sup> a la base de datos para todas las tablas y vistas.

Para las vistas *entrada*, *salida*, *existencia* y *exist.todos* se utilizó en principio otra conexión tipo rdoConnection (además de la conexión ODBCDirect que ya existía), ya que admitía el uso del control RemoteDataControl, el cual facilita las inserciones y las modificaciones de los datos desde un control DBGrid<sup>7</sup>. Debido a los problemas originados por ODBC lo que se hizo finalmente fue realizar las inserciones y modificaciones por medio de formularios, cada uno con todos los campos de las vistas de entradas y salidas. Cada vez que se hace una inserción o una modificación se actualizan los datos del DBGrid, por lo que ya no es necesario seguir utilizando el control RemoteData. El DBGrid también puede estar vinculado a un control DataControl, el cual es de sólo lectura (si las actualizaciones se hacen por formularios, ya no es necesario que el acceso a la tabla sea de lectura/escritura), pero se puede utilizar mediante la conexión ODBCDirect que ya se tenía. El acceso a las vistas de existencias y *exist.todos* es de sólo lectura, por lo que tampoco es necesario aquí el uso de un control RemoteData.

El cliente se instalará en cada uno de los ordenadores de las áreas donde se acceda a la base de datos, bien desde un CD-ROM o desde disquetes de instalación.

---

<sup>5</sup> ODBC: Open DataBase Connectivity. Estándar para el acceso remoto a bases de datos.

<sup>6</sup> ODBCDirect: Conexión que utiliza el estándar ODBC sin pasar por el motor de bases de datos Jet de Microsoft (utilizado en Microsoft Access)

<sup>7</sup> DBGrid: Rejilla que está vinculada a una tabla. Los datos se distribuyen en filas para los distintos registros y en columnas para los campos de dichos registros. Hace más fáciles la inserción y modificación de los datos. El DBGrid se puede vincular a los controles DataControl y RemoteDataControl.

Además, en cada cliente habrá que instalar una conexión ODBC al origen de datos. En este momento, el servidor tiene la dirección IP 193.146.164.8, pero es posible que se cambie.

Pasos para crear y configurar la conexión:

- Pulsar Menú Inicio, Configuración, Panel de Control.
- Hacer click en el icono ODBC de 32 bits o bien Fuentes de datos ODBC. Crear una nueva, bien de usuario o de sistema.
- Agregar una conexión nueva a SQL Server
- Parámetros:

Nombre del origen de datos: dbGesLab

Servidor: 193.146.164.8

Biblioteca de red: dbmssocn<sup>8</sup>

- Pulsando Opciones >>

Nombre de la base de datos: DBLab.

El resto de parámetros no se modifican.

Uno de los requisitos iniciales era la ejecución del propio cliente por red siguiendo el protocolo NET Bios. El programa cliente residiría en el servidor, y los ordenadores terminales, por ejemplo mediante un acceso directo, ejecutarían el programa por red. Conseguir este requisito supondría reorganizar toda la estructura de dominios de la red de la Universidad, por lo que en principio no se podrá cumplir.

---

<sup>8</sup> Dbmssocn: Se refiere a la biblioteca dbmssocn.dll. Esta biblioteca es necesaria para realizar el acceso a SQL Server mediante el protocolo TCP/IP. Si se deja la opción por defecto, solo se permite el acceso por el protocolo NET BIOS. Cabe resaltar que esta información no aparece documentada en los manuales de SQL Server ni en la ayuda sobre ODBC. La resolución de este problema llevó casi una semana.

## **5. Configuración de la base de datos**

Al instalar el software en el servidor, se configuró el sistema operativo, integrando dicho servidor en el dominio NT del CPD<sup>9</sup> de la Universidad de Burgos. También se instaló el software servidor del SQL Server. Esta operación llevó un día.

Una vez hecho esto se procedió a la configuración del SQL Server y al aprendizaje de dicho programa. Para ello se ha utilizado el manual Microsoft SQL Server Training, en inglés, de Microsoft Press, que ya se encontraba en el CPD.

## **6. Implementación en Visual Basic**

Para la programación en el lenguaje Visual Basic 5 Enterprise Edition de Microsoft, se han seguido los libros Visual Basic 5 a fondo, de Sergio Arboles, y Programación de SQL Server 7.0 con Visual Basic 6.0, de William R. Vaughn editado por Microsoft Press. Este último libro ha sido comprado por la Universidad durante el desarrollo de la aplicación.

Al acabar el análisis, a la vez que se iba configurando el SGBD<sup>10</sup> SQL Server, se fue diseñando la interfaz de la aplicación. Tras probar varias veces con distintas configuraciones, se fue comenzando el desarrollo. Después se ha ido realizando algún cambio, y se ha aprovechado la mayoría del código, teniendo que cambiar alguna parte de éste.

Además se ha ido optimizando dicho código, agrupando en funciones aquellas partes que realizaban la misma tarea. De esta forma se mejoran la claridad y la presentación del código fuente del programa.

Según se iba desarrollando la aplicación se han ido añadiendo comentarios que explican el funcionamiento de cada una de sus partes y aquellas particularidades que pueda resultar interesante destacar.

---

<sup>9</sup> CPD: Centro de Proceso de Datos de la Universidad de Burgos.

<sup>10</sup> SGBD: Sistema Gestor de Base de Datos

## 7. Informes

La presentación e impresión de informes se intentó realizar en principio con la herramienta Crystal Reports que incluye Visual Basic 5. Como la funcionalidad de Crystal Reports es bastante limitada, se buscó en vano alguna otra herramienta<sup>11</sup> para la elaboración de dichos informes. Crystal Reports es adecuada para imprimir informes sobre tablas fijas en bases de datos locales, pero no permite cambiar nombres de tablas en tiempo de ejecución, por lo que para cada tabla habría que hacer un informe distinto. En esta aplicación interesa realizar informes sobre las vistas de fichas y existencias que tiene un usuario en un momento dado basándose en un único modelo de informe.

Además sería interesante que pudiera tomar los datos de la aplicación en vez de acceder directamente a la base de datos, ya que cada vez que se imprime un informe se tienen que recibir de nuevo los datos desde el servidor. De esta manera se podrían aprovechar los filtros introducidos en la aplicación (por ejemplo, imprimir un informe únicamente de las existencias de Ácido Sulfúrico en un área).

Otra funcionalidad interesante sería imprimir solamente los campos deseados (por ejemplo, si sólo interesan las existencias no tiene por qué aparecer el proveedor de un producto determinado).

Finalmente el procedimiento utilizado ha sido la impresión de líneas directamente sobre la impresora por defecto. Para la impresión de algunos de los informes se ha utilizado la metodología de Warnier.

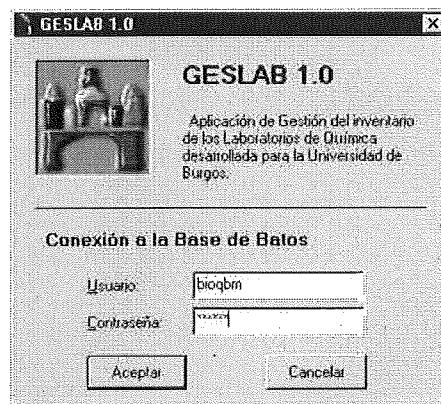
El único inconveniente de esta solución es la imposibilidad de mostrar una presentación preliminar del informe, que puede ser útil para hacerse una idea del número de páginas que ocupará el informe.

---

<sup>11</sup> Se intentó utilizar también el control Data Report que incluye Visual Basic 6.0, pero dicho control tiene bastantes limitaciones; da la impresión de que este control se ha incluido en la versión 6.0 sin estar terminado.

## 8. Interfaz de Usuario

Al arrancar el programa aparecerá un formulario de bienvenida en el cual se introducirá el nombre de usuario y la contraseña para conectarse a la base de datos.



Si los datos introducidos son correctos, se abrirá la conexión anteriormente citada y aparecerá una ventana con un conjunto de pestañas. Inicialmente está activa la pestaña de filtro, donde es posible limitar las consultas de fichas por los siguientes campos:

1. Nombre (completo o no)
2. Fórmula
3. Departamento
4. Área
5. Centro
6. Localización
7. Calidad
8. Marca
9. Proveedor

Las entradas y salidas se pueden filtrar según:

1. Fecha de Entradas
  2. Fecha de Salidas
  3. Fecha de Caducidad
  4. Sólo Residuos
- 4.1. Residuos o no residuos

Como lo más común es hacer una búsqueda por nombre de producto, el programa incorpora la funcionalidad de completar la cadena del producto con el primer producto que encuentra cada vez que se pulsa una tecla (excepto borrar). Por ejemplo, al escribir et, buscará el primer producto que empiece por et (Etanol, por ejemplo) y aparecerá el nombre completo en Producto. Esto es similar al comportamiento de la barra de direcciones de los navegadores de Internet.

Si no se completa el nombre, seguramente ese producto no está almacenado en la tabla de productos de la base de datos.

Una vez establecido el filtro, se pueden consultar las fichas con las entradas, salidas y existencias, o bien las fichas de todas las áreas con sus respectivas existencias.

En las pestañas de entradas y salidas es posible gestionar las fichas, es decir, crear fichas nuevas y modificar las existentes.

The screenshot shows a software application window titled 'Ficha' (Product Card) for managing chemical entries. At the top, there are tabs: 'Conexión', 'Productos', 'Localización', 'Calidad', 'Marca', 'Proveedores', 'Informes', and 'Ayuda'. Below the tabs, there are four main sections:

- Producto:** Acido Sulfúrico (H<sub>2</sub>SO<sub>4</sub>)
- Fórmula:** H<sub>2</sub>SO<sub>4</sub>
- Departamento:** Biotecnología y C. Alim.
- Área:** Biocatánica y Biología Molecular

On the right side of the card, there are three small icons: an upward arrow, a downward arrow, and a square with a diagonal line.

Below the card, there are several buttons and input fields:

- Centro:** Facultad de Ciencias
- Localización:** Nevera
- Proveedor:** John Pazos
- Marca:** Meck
- Ver Entradas**
- Nueva Ficha**
- Modificar Ficha**
- Validar Operación**
- Corregir**
- Calidad:** PA

At the bottom left, there is a table titled 'Entradas' (Entries) with the following data:

Fecha	Fecha Caducidad	Lote	Unidades	Capacidad	g/ml	Residuo
22/10/1999			10	1000	g	n

On the right side of the table, there are two buttons: 'Nueva Entrada' and 'Modificar Entrada'.

At the very bottom, there are six buttons: 'Filtrar', 'Entradas', 'Salidas', 'Existencias en mi Área', 'Existencias en Todas las Áreas', and 'Salir'.

[x] [e]

Ficha

<b>Producto</b>	Ácido Sulfúrico	<input type="checkbox"/> Oculta a los demás áreas																				
<b>Fórmula</b>	H <sub>2</sub> SO <sub>4</sub>																					
<b>Departamento</b>	Biotecnología y C. Alim.																					
<b>Centro</b>	Facultad de Ciencias																					
<b>Proveedor</b>	Jofra Pacos																					
<input type="button" value="Ver Salidas"/>	<input type="button" value="Nueva Ficha"/>	<input type="button" value="Modificar Ficha"/>	<input type="button" value="Validar Entrada"/>	<input type="button" value="Cancelar"/>	<b>Localización</b> Nevera	<b>Marca</b> Merck	<b>Calidad</b> PA															
<table border="1"> <thead> <tr> <th>Salidas</th> <th>Fecha</th> <th>Fecha Caducidad</th> <th>Lote</th> <th>Unidades</th> <th>Capacidad g/ml</th> <th>Residuo</th> </tr> </thead> <tbody> <tr> <td></td> <td>► 25/10/1999</td> <td></td> <td></td> <td>1</td> <td>1000</td> <td>g</td> <td>n</td> </tr> </tbody> </table>								Salidas	Fecha	Fecha Caducidad	Lote	Unidades	Capacidad g/ml	Residuo		► 25/10/1999			1	1000	g	n
Salidas	Fecha	Fecha Caducidad	Lote	Unidades	Capacidad g/ml	Residuo																
	► 25/10/1999			1	1000	g	n															
				<input type="button" value="Nueva Salida"/> <input type="button" value="Modificar Salida"/>																		
<input type="button" value="Filtrar"/> <input type="button" value="Entradas"/> <input type="button" value="Salidas"/> <input type="button" value="Existencias en mi Área"/> <input type="button" value="Existencias en Todas las Áreas"/> <input type="button" value="Salir"/>																						

[x] [e]

Ficha

<b>Producto</b>	Ácido Sulfúrico	<input type="checkbox"/> Oculta a los demás áreas										
<b>Fórmula</b>	H <sub>2</sub> SO <sub>4</sub>											
<b>Departamento</b>	Biotecnología y C. Alim.											
<b>Centro</b>	Facultad de Ciencias											
<b>Proveedor</b>	Jofra Pacos											
<input type="button" value="Ver Existencias"/>	<input type="button" value="Crear Informe"/>	<b>Localización</b> Nevera	<b>Marca</b> Merck	<b>Calidad</b> PA								
<table border="1"> <thead> <tr> <th>Existencias</th> <th>Unidades</th> <th>Capacidad</th> <th>g/ml</th> </tr> </thead> <tbody> <tr> <td></td> <td>► 9</td> <td>1000</td> <td>g</td> </tr> </tbody> </table>					Existencias	Unidades	Capacidad	g/ml		► 9	1000	g
Existencias	Unidades	Capacidad	g/ml									
	► 9	1000	g									
<input type="button" value="Filtrar"/> <input type="button" value="Entradas"/> <input type="button" value="Salidas"/> <input type="button" value="Existencias en mi Área"/> <input type="button" value="Existencias en Todas las Áreas"/> <input type="button" value="Salir"/>												

Conexión Productos Localización Calidad Marca Proveedores Informes Ayuda

Filtrar Consultas Entradas de Compuestos Salidas de Compuestos Existencias en mi Área Existencias en Todas las Áreas

**Ficha**

**Producto:** Ácido Sulfúrico  
**Fórmula:** H<sub>2</sub>SO<sub>4</sub>

**Departamento:** Biotecnología y C. Alim.  
**Área:** Bioquímica y Biología Molecular

**Centro:** Facultad de Ciencias  
**Marca:** Merck

**Proveedor:** John Pascos  
**Calidad:** PA

**Ver Existencias** **Crear Informe**

Existencias en Todas las Áreas		
Unidades	Capacidad	[g/ml]
▶ 9	1000	g

**Filtrar** **Entradas** **Salidas** **Existencias en mi Área** **Existencias en Todas las Áreas** **Sair**

Se han añadido además formularios para la gestión de productos, localizaciones, marcas, calidades y proveedores.

En algunos de estos formularios se ha introducido la posibilidad de Eliminar algún elemento de la base de datos (por ejemplo, eliminar una localización), siempre que no haya ningún otro elemento que dependa del primero (si se elimina un armario, no debe haber ningún producto dentro de ese armario; si se elimina una marca, no debe haber ningún producto de esa marca).

## 9. Posibles ampliaciones de esta beca

- Gestión de productos caducados. En la última reunión se expuso que la mayoría de los productos no tenían fecha de caducidad, o en caso contrario era muy lejana, por lo que se restó importancia a esta funcionalidad. Esta funcionalidad se puede implementar mediante un sencillo procedimiento en la base de datos o bien creando una vista de productos caducados.

- Gestión de departamentos, áreas y centros. Esta opción es interesante en caso de que haya cambios en la organización de la Universidad. Con esta funcionalidad, el usuario administrador puede introducir estos cambios en la base de datos desde el programa, sin necesidad de acceder directamente a la base de datos.



# **Geslab 1.0**

***Memoria del  
desarrollo de la  
aplicación y  
manuales***

*Autor: Álvaro de Luis de Miguel*

# **Memoria del desarrollo de una aplicación realizada para la Universidad de Burgos**

En esta memoria se van a detallar las fases del desarrollo de una aplicación para la gestión del inventario de los laboratorios de la Universidad de Burgos.

## **1. Entrevistas con los futuros usuarios**

Al comienzo de la beca, en Febrero, tuvo lugar una primera entrevista (muy larga) con un usuario en la que se especificaron los requisitos fundamentales del sistema.

Durante las dos semanas siguientes se realizaron entrevistas puntuales con ese mismo usuario para detallar alguno de los requisitos que no habían quedado claros.

Más tarde, en Junio, la coordinadora de Seguridad y Salud convocó una reunión llamando a todas las áreas interesadas y se contó con la asistencia de unos diez representantes. En esta reunión se introdujeron cambios importantes que obligaron a modificar en profundidad el sistema, desde el análisis hasta la implementación, que ya estaba bastante avanzada, en especial la interfaz de usuario.

## **2. Especificación de requisitos del sistema**

Durante la primera entrevista se elaboró una especificación de requisitos del sistema inicial, pero en la siguiente reunión con los usuarios se añadieron varios requisitos, como la gestión de proveedores, y algunos cambios en los campos de las tablas.

A continuación se incluye una especificación de requisitos del sistema. Esta especificación es la que se definió en la segunda entrevista.

### **2.1. Descripción general del sistema**

El sistema deberá ser capaz de gestionar el inventario de todos los laboratorios de química de la Universidad de Burgos, controlando la fecha en que entran los productos y la fecha en que salen, y a partir de estos datos se calculan las existencias. Debe permitir

consultar las existencias de los otros laboratorios, de tal forma que unos laboratorios puedan pedir a otros productos que les faltan, y así aprovechar mejor los recursos de la Universidad.

## **2.2. Descripción detallada de los requisitos del sistema**

1. El sistema deberá tener terminales en cada uno de los laboratorios interesados en gestionar su inventario de productos.
2. Cada área deberá gestionar solamente sus propios productos. En la Universidad hay varios departamentos y varias áreas por cada departamento. Un área puede encontrarse repartida por distintos centros, y lógicamente en un centro hay varias áreas.
3. En cada área y centro hay varias localizaciones, es decir, ubicaciones donde se pueden guardar los productos (tales como armarios, estanterías...).
4. Cada área debe tener la posibilidad de ocultar una localización a las demás áreas, de tal forma que éstas no puedan conocer la existencia de los productos almacenados en dichas localizaciones (véase más adelante). También será posible ocultar cada ficha individualmente.
5. Se almacenarán los productos con su nombre, fórmula y precauciones. Este último campo será de texto libre.
6. Los productos deberán estar organizados en fichas. Cada ficha se corresponde con un producto guardado en una localización (no es lo mismo un bote de NaCl guardado en el armario que un saco de NaCl guardado en el almacén), con una calidad determinada, de una marca determinada y proporcionado por un proveedor determinado. Este último campo se podrá dejar en blanco.
7. Se almacenarán datos de los proveedores, como su nombre, dirección, teléfono, fax, e-mail y una persona de contacto.
8. Se guardarán además los nombres de las marcas y calidades de los productos. Se deberá permitir su gestión por parte de cualquier usuario.
9. También se gestionará la relación entre marcas de productos y proveedores. Un proveedor llevará varias marcas, y una marca puede ser llevada por varios proveedores.
10. En los envoltorios de los productos (frascos...) suele venir un número de lote. Este se guardará también, opcionalmente.

11. En caso de que el producto tenga fecha de caducidad, se deberá permitir almacenar ésta.
12. En un mismo pedido suelen llegar varias unidades de un producto que corresponde a una determinada ficha.
13. Se almacenará la capacidad de cada frasco, bote... Dicha capacidad puede venir dada en gramos(g) o en mililitros(ml).
14. Se podrá especificar si un frasco es un residuo o no, es decir, si el área quiere deshacerse de ese producto o no.
15. Se guardarán las entradas y salidas de los productos, de tal forma que cada ficha tendrá varias entradas y salidas. De cada entrada y salida se guardan los mismos datos: fecha, fecha de caducidad, lote, unidades, capacidad, en gramos o en mililitros, residuo.
16. Basándose en las entradas y salidas de los productos, se calcularán las existencias de los productos del área. En este cálculo no se tendrá en cuenta si un producto es residuo o no, ni la fecha de caducidad. Se calculará el número de unidades de una determinada capacidad (Ejemplo: 10 frascos de 200 g, 25 de 500 g, 40 de 1000 ml)
17. Deberá ser posible consultar las existencias de otras áreas. Esto es útil para aprovechar mejor el material que hay en la Universidad, de tal forma que un laboratorio pueda pedir a algún otro un producto que le falta al primero y le sobra al último. Será posible ocultar productos a las demás áreas (véase requisito 4). Se podrán consultar todas las fichas no ocultas y situadas en localizaciones no ocultas. Se sabrá el área y el centro donde están las existencias, pero no la localización exacta, por seguridad.
18. Se podrán generar informes a partir de las existencias de cada área así como de todas las áreas, pudiendo seleccionar los campos que aparecerán en dichos informes.
19. Los datos de departamentos, áreas y centros sólo podrán ser modificados por un usuario administrador, que tendrá el control total sobre el sistema.
20. Cada área podrá gestionar sus propias localizaciones, pero no las de las demás áreas. Se puede añadir, modificar y borrar una localización.
21. Cada área también podrá gestionar solamente sus fichas, entradas y salidas. Para estos datos sólo se permitirán las opciones de añadir y modificar; no será posible borrar, por seguridad.

22. Los productos, calidades, proveedores y marcas serán comunes para todas las áreas y los podrán gestionar cualquiera de éstas. Se supone que cada usuario sabe lo que hace. Las operaciones permitidas para estos datos serán añadir y modificar. Se podrán borrar aquellos que no estén incluidos en ninguna ficha.
23. Cada producto tiene una hoja de seguridad. El programa deberá ser capaz de visualizar la hoja de seguridad de cada producto.
24. Se elaborará documentación de ayuda al usuario, así como ayuda a los operadores para las tareas de mantenimiento y administración del sistema.

### **3. Análisis**

Una vez realizadas la primera entrevista y la especificación de requisitos, se procedió a la elaboración del análisis, que duró casi todo el primer mes de la beca. Se ha utilizado la metodología estructurada de Yourdon. La realización del análisis no ha sido muy detallada, ya que posteriormente se realizará un análisis y un diseño siguiendo una metodología orientada a objetos, para ser presentado el trabajo como proyecto de fin de carrera.

La segunda reunión obligó a cambiar el análisis, lo cual llevó casi otras dos semanas. La especificación de proceso no se actualizó por razones de tiempo.

Para la realización de los DFD se ha utilizado la herramienta EasyCase Professional  
4.23.

### 3.1. Diagrama Entidad/Relación

El diagrama Entidad/Relación resultante del análisis es el siguiente:

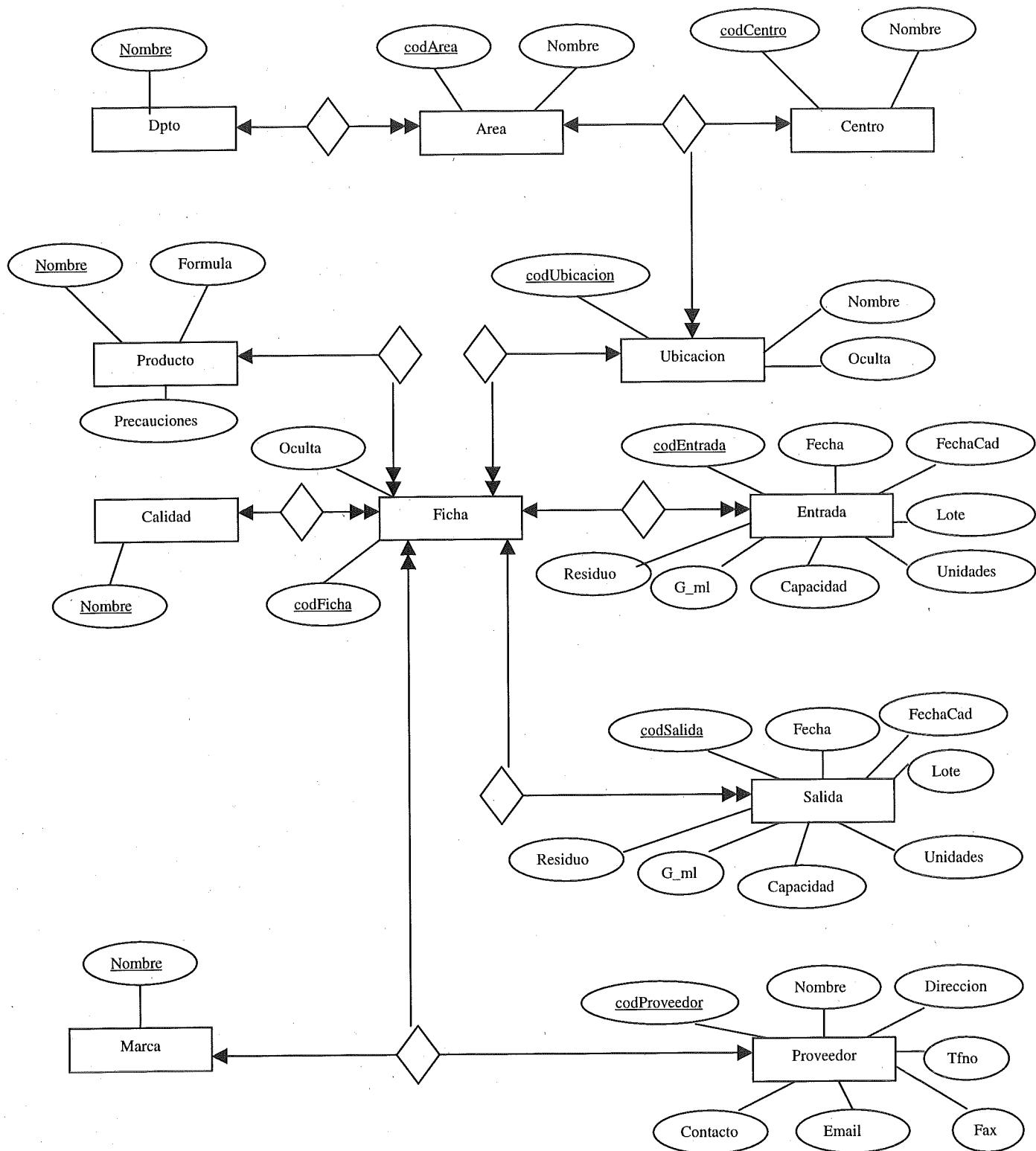


Figura 1. Diagrama Entidad/Relación

## **4. Especificaciones de diseño**

La fase de diseño que se proponía en la metodología de Yourdon tampoco se ha realizado, de nuevo por razones de tiempo. Se elaborará posteriormente siguiendo una metodología orientada a objetos.

En este apartado se detallarán brevemente las decisiones tomadas en la construcción del sistema que cumpla los requisitos especificados anteriormente.

### **4.1. Sistema Operativo y Sistema Gestor de Base de Datos**

Los datos se guardarán en una base de datos única situada en un ordenador central perteneciente a la red de la Universidad de Burgos.

Por razones económicas, y dado que el número de accesos concurrentes será pequeño, el sistema operativo elegido es Windows NT Server versión 4.0 en inglés, y el Sistema Gestor de Base de Datos (de ahora en adelante SGBD) es el SQL Server 6.5 de Microsoft en inglés.

El ordenador se llama *camelot*, y es un servidor dedicado exclusivamente a la base de datos. Es un Pentium II a 400 MHz, con 128 MB de memoria RAM.

### **4.2. Configuración de la Base de Datos**

#### **4.2.1. Usuarios**

El SGBD SQL Server es multiusuario, y permite accesos concurrentes.

Tiene la posibilidad de permitir el acceso a usuarios del dominio de Windows NT, a usuarios creados específicamente desde SQL Server, o a ambos (acceso mixto).

Dado que los usuarios no tienen por qué tener acceso para iniciar una sesión en los ordenadores del dominio, la opción elegida es el acceso a cuentas creadas desde SQL Server.

#### **4.2.2. Tablas**

Siguiendo el diagrama Entidad/Relación, cada entidad corresponde a una tabla. Además se han creado las tablas *centro\_area*, procedente de la relación entre *ubicación*, *centro* y *area*, y *prov\_marca*, procedente de la relación entre *ficha*, *proveedor* y *marca*. Esta

última relación se modificó posteriormente debido a un cambio en los requisitos. Por comodidad los usuarios querían que introducir el proveedor fuese opcional, es decir, el campo *proveedor* de la tabla *ficha* puede ser nulo, por lo que este campo no se relacionó temporalmente con la tabla *prov\_marca*. Más tarde, esta solución se cambió por restablecer la relación inicial y añadir en la tabla *proveedor* el registro con nombre “<Ninguno>” y código ‘0’. Los formularios de gestión de proveedores no podrán ver este registro.

Para resolver el problema de las claves primarias<sup>1</sup> al añadir un nuevo registro, se crea una tabla llamada *maximo*, sobre la cual nadie tendrá ningún permiso. Tendrá un campo para cada una de las siguientes tablas: *centro*, *ubicación*, *ficha*, *proveedor*, *entrada* y *salida*. En cada uno de estos campos se guarda el código del último centro, ubicación y ficha, respectivamente, que se añadió a la base de datos. Habrá unos procedimientos que aumentarán en 1 el valor de cada uno de estos campos (según la tabla a la que se quiera añadir el registro) y devolverán el nuevo valor de la clave primaria. Los usuarios sí tendrán permiso de ejecución sobre estos procedimientos.

#### 4.2.3. Vistas

Como cada área sólo puede gestionar sus fichas, habrá una vista de las tablas *ubicacion*, *ficha*, *entrada* y *salida* por cada área. Sobre estas tablas nadie tendrá ningún tipo de permiso, excepto el usuario administrador(sa), que tiene acceso total a la base de datos. Cada vista se llamará <usuario>\_<tabla> Ejemplo: Para el usuario *bioqbm*, tabla *ubicacion*, la vista se llamará *bioqbm\_ubicacion*.

Además habrá una vista existencias, la cual restará las unidades de las salidas a las de las entradas para las filas con valores iguales de los campos *ficha*, *g\_ml* y *capacidad*.

Se crean además las vistas *ficha.todos* y *exist.todos*, que reúnen las fichas visibles (las que se encuentran en una localización no oculta) de todas las áreas, y las existencias de esas fichas, respectivamente.

---

<sup>1</sup> Clave primaria: es un campo de la tabla que tiene un valor único (por ejemplo, cada ficha tiene un código de ficha) y no se puede repetir en dos filas diferentes (no puede haber dos fichas con el mismo código de ficha).

Una vez creadas las tablas y las vistas habrá que conceder ciertos permisos a todos los usuarios sobre algunas de ellas:

- Permiso de solo lectura (*select*) sobre *dpto*, *area*, *centro*, *centro\_area*, *ficha.todos* y *exist.todos*.
- Permiso de lectura/inserción/modificación/borrado sobre *marca*, *proveedor*, *prov\_marca*, *producto* y *calidad*.
- Tablas y vistas sin ningún permiso para los usuarios: *ubicacion*, *ficha*, *entrada*, *salida*, *exist*.

#### 4.2.4. Crear un nuevo usuario en la base de datos

Se ha creado un procedimiento *crear\_usuario*, que tiene como parámetros el nombre de usuario y la contraseña. Ejemplo: *crear\_usuario* ‘quimicas’, ‘h3i2j1’

Este procedimiento añade el nombre de usuario y un login con el mismo nombre y con dicha password<sup>2</sup> a la base de datos. Añade el usuario al grupo Usuarios. Acto seguido crea las vistas *<usuario>\_ubicacion*, *<usuario>\_ficha*, *<usuario>\_entrada*, *<usuario>\_salida*, *<usuario>\_exist*.

Después asigna los permisos apropiados al usuario sobre las vistas creadas y los procedimientos:

- Permisos de lectura/inserción/modificación sobre *producto*, *<usuario>\_ficha*, *<usuario>\_entrada*, *<usuario>\_salida*, *<usuario>\_exist*.
- Todos los permisos sobre *<usuario>\_ubicacion*.
- Se deniega el permiso de borrado sobre *<usuario>\_ficha*, *<usuario>\_entrada* y *<usuario>\_salida*.

---

<sup>2</sup> En SQL Server un usuario puede tener varios logins, cada uno de éstos con su password (contraseña). Para poder conectarse a la base de datos, un usuario necesita como mínimo un login.

- Permisos de ejecución sobre los procedimientos *num\_centro*, *num\_ubicacion*, *num\_ficha*, *num\_proveedor*, *num\_entrada*, *num\_salida*.

Los datos sobre las tablas *area* y *dpto* no se insertan, se deben introducir manualmente en la base de datos, ya que no es normal que cambie la organización en áreas y en departamentos de la Universidad.

Como ampliación de este proyecto se puede implementar la gestión de áreas, departamentos y centro por programa, así como la gestión de usuarios.

#### **4.2.5. Borrar un usuario de la base de datos**

Al igual que para la creación de un usuario, se ha creado un procedimiento para borrar un usuario, que primero borrara todas las vistas de ese usuario y luego quitará el login y el usuario de la base de datos.

Los datos sobre ese usuario(área) no se borrarán de la base de datos.

#### **4.2.6. Triggers<sup>3</sup>**

En la base de datos se utiliza un trigger en la tabla salidas. Al insertar y modificar, comprueba que el número de unidades es menor o igual al de las existencias. Además comprueba que hay alguna entrada para la misma ficha, con la misma capacidad en gramos o mililitros.

#### **4.2.7. Backups<sup>4</sup>**

Todavía no se ha propuesto ninguna solución para la gestión de las copias de seguridad. Lo más probable es que se hagan copias periódicas de los datos en CD-ROM.

---

<sup>3</sup> Trigger: disparador. Es un conjunto de sentencias que se ejecutan automáticamente al realizar una operación ( inserción, modificación, borrado) sobre una tabla.

<sup>4</sup> Backup: Copia de Seguridad de los datos.

#### 4.2.8. Cliente

El cliente se implementará en el lenguaje Visual Basic 5.0 Enterprise Edition de Microsoft, en español. Se ha elegido este lenguaje por su facilidad (y rapidez) de implementación, por estar diseñado para los sistemas operativos Windows de 32 bits (95/98/NT) y por ser el lenguaje de programación más utilizado en el desarrollo de aplicaciones de gestión.

El acceso remoto a los datos se hará por ODBC<sup>5</sup>, y se utilizará una conexión ODBCDirect<sup>6</sup> a la base de datos para todas las tablas y vistas.

Para las vistas *entrada*, *salida*, *existencia* y *exist.todos* se utilizó en principio otra conexión tipo rdoConnection (además de la conexión ODBCDirect que ya existía), ya que admitía el uso del control RemoteDataControl, el cual facilita las inserciones y las modificaciones de los datos desde un control DBGrid<sup>7</sup>. Debido a los problemas originados por ODBC lo que se hizo finalmente fue realizar las inserciones y modificaciones por medio de formularios, cada uno con todos los campos de las vistas de entradas y salidas. Cada vez que se hace una inserción o una modificación se actualizan los datos del DBGrid, por lo que ya no es necesario seguir utilizando el control RemoteData. El DBGrid también puede estar vinculado a un control DataControl, el cual es de sólo lectura (si las actualizaciones se hacen por formularios, ya no es necesario que el acceso a la tabla sea de lectura/escritura), pero se puede utilizar mediante la conexión ODBCDirect que ya se tenía. El acceso a las vistas de existencias y *exist.todos* es de sólo lectura, por lo que tampoco es necesario aquí el uso de un control RemoteData.

El cliente se instalará en cada uno de los ordenadores de las áreas donde se acceda a la base de datos, bien desde un CD-ROM o desde disquetes de instalación.

Además, en cada cliente habrá que instalar una conexión ODBC al origen de datos. En este momento, el servidor tiene la dirección IP 193.146.164.8, pero es posible que se cambie.

---

<sup>5</sup> ODBC: Open DataBase Connectivity. Estándar para el acceso remoto a bases de datos.

<sup>6</sup> ODBCDirect: Conexión que utiliza el estándar ODBC sin pasar por el motor de bases de datos Jet de Microsoft (utilizado en Microsoft Access)

<sup>7</sup> DBGrid: Rejilla que está vinculada a una tabla. Los datos se distribuyen en filas para los distintos registros y en columnas para los campos de dichos registros. Hace más fáciles la inserción y modificación de los datos. El DBGrid se puede vincular a los controles DataControl y RemoteDataControl.

Pasos para crear y configurar la conexión:

- Pulsar Menú Inicio, Configuración, Panel de Control.
- Hacer click en el ícono ODBC de 32 bits o bien Fuentes de datos ODBC. Crear una nueva, bien de usuario o de sistema.
- Agregar una conexión nueva a SQL Server
- Parámetros:

Nombre del origen de datos: dbGesLab

Servidor: 193.146.164.8

Biblioteca de red: dbmssocn<sup>8</sup>

- Pulsando Opciones >>

Nombre de la base de datos: DBLab.

El resto de parámetros no se modifican.

Uno de los requisitos iniciales era la ejecución del propio cliente por red siguiendo el protocolo NET Bios. El programa cliente residiría en el servidor, y los ordenadores terminales, por ejemplo mediante un acceso directo, ejecutarían el programa por red. Conseguir este requisito supondría reorganizar toda la estructura de dominios de la red de la Universidad, por lo que en principio no se podrá cumplir.

---

<sup>8</sup> Dbmssocn: Se refiere a la biblioteca dbmssocn.dll. Esta biblioteca es necesaria para realizar el acceso a SQL Server mediante el protocolo de internet, TCP/IP. Si se deja la opción por defecto, solo se permite el acceso por el protocolo NET BIOS (diseñado por Microsoft). Cabe resaltar que esta información no aparece documentada en los manuales de SQL Server ni en la ayuda sobre ODBC. La resolución de este problema llevó casi una semana.

## **5. Configuración de la base de datos**

Al instalar el software en el servidor, se configuró el sistema operativo, integrando dicho servidor en el dominio NT del CPD<sup>9</sup> de la Universidad de Burgos. También se instaló el software servidor del SQL Server. Esta operación llevó un día.

Una vez hecho esto se procedió a la configuración del SQL Server y al aprendizaje de dicho programa. Para ello se ha utilizado el manual Microsoft SQL Server Training, en inglés, de Microsoft Press, que ya se encontraba en el CPD.

## **6. Implementación en Visual Basic**

Para la programación en el lenguaje Visual Basic 5 Enterprise Edition de Microsoft, se han seguido los libros Visual Basic 5 a fondo, de Sergio Arboles, y Programación de SQL Server 7.0 con Visual Basic 6.0, de William R. Vaughn editado por Microsoft Press. Este último libro ha sido comprado por la Universidad durante el desarrollo de la aplicación.

Al acabar el análisis, a la vez que se iba configurando el SGBD<sup>10</sup> SQL Server, se fue diseñando la interfaz de la aplicación. Tras probar varias veces con distintas configuraciones, se fue comenzando el desarrollo. Después se ha ido realizando algún cambio, y se ha aprovechado la mayoría del código, teniendo que cambiar alguna parte de éste.

Además se ha ido optimizando dicho código, agrupando en funciones aquellas partes que realizaban la misma tarea. De esta forma se mejoran la claridad y la presentación del código fuente del programa.

Según se iba desarrollando la aplicación se han ido añadiendo comentarios que explican el funcionamiento de cada una de sus partes y aquellas particularidades que pueda resultar interesante destacar.

---

<sup>9</sup> CPD: Centro de Proceso de Datos de la Universidad de Burgos.

<sup>10</sup> SGBD: Sistema Gestor de Base de Datos

## 7. Informes

La presentación e impresión de informes se intentó realizar en principio con la herramienta Crystal Reports que incluye Visual Basic 5. Como la funcionalidad de Crystal Reports es bastante limitada, se buscó en vano alguna otra herramienta<sup>11</sup> para la elaboración de dichos informes. Crystal Reports es adecuada para imprimir informes sobre tablas fijas en bases de datos locales, pero no permite cambiar nombres de tablas en tiempo de ejecución, por lo que para cada tabla habría que hacer un informe distinto. En esta aplicación interesa realizar informes sobre las vistas de fichas y existencias que tiene un usuario en un momento dado basándose en un único modelo de informe.

Además sería interesante que pudiera tomar los datos de la aplicación en vez de acceder directamente a la base de datos, ya que cada vez que se imprime un informe se tienen que recibir de nuevo los datos desde el servidor. De esta manera se podrían aprovechar los filtros introducidos en la aplicación (por ejemplo, imprimir un informe únicamente de las existencias de Ácido Sulfúrico en un área).

Otra funcionalidad interesante sería imprimir solamente los campos deseados (por ejemplo, si sólo interesan las existencias no tiene por qué aparecer el proveedor de un producto determinado).

Finalmente el procedimiento utilizado ha sido la impresión de líneas directamente sobre la impresora por defecto. Para la impresión de algunos de los informes se ha utilizado la metodología de Warnier.

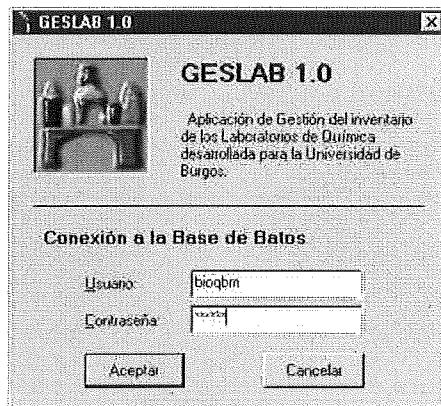
El único inconveniente de esta solución es la imposibilidad de mostrar una presentación preliminar del informe, que puede ser útil para hacerse una idea del número de páginas que ocupará el informe.

---

<sup>11</sup> Se intentó utilizar también el control Data Report que incluye Visual Basic 6.0, pero dicho control tiene bastantes limitaciones; da la impresión de que este control se ha incluido en la versión 6.0 sin estar terminado.

## 8. Interfaz de Usuario

Al arrancar el programa aparecerá un formulario de bienvenida en el cual se introducirá el nombre de usuario y la contraseña para conectarse a la base de datos.



Si los datos introducidos son correctos, se abrirá la conexión anteriormente citada y aparecerá una ventana con un conjunto de pestañas. Inicialmente está activa la pestaña de filtro, donde es posible limitar las consultas de fichas por los siguientes campos:

1. Nombre (completo o no)
2. Fórmula
3. Departamento
4. Área
5. Centro
6. Localización
7. Calidad
8. Marca
9. Proveedor

Las entradas y salidas se pueden filtrar según:

1. Fecha de Entradas
  2. Fecha de Salidas
  3. Fecha de Caducidad
  4. Sólo Residuos
- 4.1. Residuos o no residuos

Como lo más común es hacer una búsqueda por nombre de producto, el programa incorpora la funcionalidad de completar la cadena del producto con el primer producto que encuentra cada vez que se pulsa una tecla (excepto borrar). Por ejemplo, al escribir et, buscará el primer producto que empiece por et (Etanol, por ejemplo) y aparecerá el nombre completo en Producto. Esto es similar al comportamiento de la barra de direcciones de los navegadores de Internet.

Si no se completa el nombre, seguramente ese producto no está almacenado en la tabla de productos de la base de datos.

Una vez establecido el filtro, se pueden consultar las fichas con las entradas, salidas y existencias, o bien las fichas de todas las áreas con sus respectivas existencias.

En las pestañas de entradas y salidas es posible gestionar las fichas, es decir, crear fichas nuevas y modificar las existentes.

**Ficha**

**Producto:** Ácido Sulfúrico

**Fórmula:** H<sub>2</sub>SO<sub>4</sub>

Oculta a los demás áreas

**Departamento:** Biotecnología y C. Alim.

**Área:** Bioquímica y Biología Molecular

**Centro:** Facultad de Ciencias

**Localización:** Neváia

**Proveedor:** John Parez

**Marca:** Msick

**Ver Entradas** **Nueva Ficha** **Modificar Ficha** **Validar Entrada** **Cancelar** **Calidad:** PA

Entradas						
Fecha	Fecha Caducidad	Lote	Unidades	Capacidad	g/ml	Residuo
► 22/10/1999			10	1000	g	n

**Nueva Entrada**  
**Modificar Entrada**

**Filtrar** **Entradas** **Salidas** **Existencias en mi Área** **Existencias en Todas las Áreas** **Sair**

[Minimizar] [Maximizar] [Cerrar]

**Ficha**

<b>Producto</b>	Ácido Sulfúrico	<input type="checkbox"/> Oculta a los demás áreas			
<b>Fórmula</b>	H <sub>2</sub> SO <sub>4</sub>				
<b>Departamento</b>	Biología y C. Alim.	<input type="checkbox"/> <b>Área</b> Bioquímica y Biología Molecular			
<b>Centro</b>	Facultad de Ciencias	<input type="checkbox"/> <b>Localización</b> Nevera			
<b>Proveedor</b>	John Pazos	<input type="checkbox"/> <b>Marca</b> Merck			
<input type="button" value="Ver Salidas"/>	<input type="button" value="Nueva Ficha"/>	<input type="button" value="Modificar Ficha"/>	<input type="button" value="Validar Operación"/>	<input type="button" value="Cancelar"/>	<b>Calidad</b> PA

Salidas						
Fecha	Fecha Caducidad	Lote	Unidades	Capacidad	g/ml	Residuo
► 25/10/1999			1	1000	g	n

[Minimizar] [Maximizar] [Cerrar]

**Ficha**

<b>Producto</b>	Ácido Sulfúrico	<input type="checkbox"/> Oculta a los demás áreas
<b>Fórmula</b>	H <sub>2</sub> SO <sub>4</sub>	
<b>Departamento</b>	Biología y C. Alim.	<input type="checkbox"/> <b>Área</b> Bioquímica y Biología Molecular
<b>Centro</b>	Facultad de Ciencias	<input type="checkbox"/> <b>Localización</b> Nevera
<b>Proveedor</b>	John Pazos	<input type="checkbox"/> <b>Marca</b> Merck
<input type="button" value="Ver Existencias"/>	<input type="button" value="Crear Informe"/>	<b>Calidad</b> PA

Existencias		
Unidades	Capacidad	g/ml
► 9	1000	g

Conexión Productos Localización Calidad Marca Proveedores Informes Ayuda

Filtrar Consultas Entradas de Compuestos Salidas de Compuestos Existencias en mi Área Existencias en Todas las Áreas

**Ficha**

**Producto:** Ácido Sulfúrico  
**Fórmula:** H<sub>2</sub>SO<sub>4</sub>

**Departamento:** Biología Celular y Bioquímica  
**Área:** Bioquímica y Biología Molecular

**Centro:** Facultad de Ciencias  
**Marca:** Merck

**Proveedor:** John Pazos  
**Calidad:** PA

**Ver Existencias** **Crear Informe**

**Existencias en Todas las Áreas**

Unidades	Capacidad	g/ml
9	1000	g

**Filtrar** **Entradas** **Salidas** **Existencias en mi Área** **Existencias en Todas las Áreas** **Sair**

Se han añadido además formularios para la gestión de productos, localizaciones, marcas, calidades y proveedores.

En algunos de estos formularios se ha introducido la posibilidad de Eliminar algún elemento de la base de datos (por ejemplo, eliminar una localización), siempre que no haya ningún otro elemento que dependa del primero (si se elimina un armario, no debe haber ningún producto dentro de ese armario; si se elimina una marca, no debe haber ningún producto de esa marca).

## 9. Posibles ampliaciones de esta beca

- Gestión de productos caducados. En la última reunión se expuso que la mayoría de los productos no tenían fecha de caducidad, o en caso contrario era muy lejana, por lo que se restó importancia a esta funcionalidad. Esta funcionalidad se puede implementar mediante un sencillo procedimiento en la base de datos o bien creando una vista de productos caducados.

- Gestión de departamentos, áreas y centros. Esta opción es interesante en caso de que haya cambios en la organización de la Universidad. Con esta funcionalidad, el usuario administrador puede introducir estos cambios en la base de datos desde el programa, sin necesidad de acceder directamente a la base de datos.

# **Geslab 1.0**

***Manual de Usuario***

## **Contenido**

<b>1- INTRODUCCIÓN .....</b>	<b>1</b>
<b>1.1- CONSTITUCIÓN DEL SISTEMA GESLAB.....</b>	<b>1</b>
<b>1.2- INSTALACIÓN DEL CLIENTE.....</b>	<b>3</b>
<b>1.3- DARSE DE ALTA EN LA BASE DE DATOS .....</b>	<b>5</b>
<b>2- PRIMEROS PASOS.....</b>	<b>6</b>
<b>2.1- EJECUTAR LA APLICACIÓN.....</b>	<b>6</b>
<b>2.2- CREAR UNA FICHA.....</b>	<b>7</b>
<b>2.3- MODIFICAR LOS DATOS DE UNA FICHA .....</b>	<b>10</b>
<b>2.4- FILTRAR CONSULTAS DE FICHAS.....</b>	<b>13</b>
<b>2.5- INTRODUCIR ENTRADAS Y SALIDAS .....</b>	<b>15</b>
<b>2.6- MODIFICAR DATOS DE ENTRADAS Y SALIDAS.....</b>	<b>17</b>
<b>2.7- CONSULTAR EXISTENCIAS EN MI AREA .....</b>	<b>19</b>
<b>2.8- CONSULTAR EXISTENCIAS EN TODAS LAS AREAS.....</b>	<b>20</b>
<b>3- AVANZADO .....</b>	<b>21</b>
<b>3.1- GESTIONAR PRODUCTOS.....</b>	<b>21</b>
<b>3.2- GESTIONAR LOCALIZACIONES.....</b>	<b>23</b>
<b>3.3- GESTIONAR PROVEORES.....</b>	<b>25</b>
<b>3.4- GESTIONAR MARCAS.....</b>	<b>27</b>
<b>3.5- GESTIONAR CALIDADES.....</b>	<b>28</b>
<b>3.6- IMPRIMIR INFORMES DE EXISTENCIAS EN MI AREA .....</b>	<b>29</b>
<b>3.7- IMPRIMIR INFORMES DE EXISTENCIAS EN TODAS LAS AREAS.....</b>	<b>30</b>

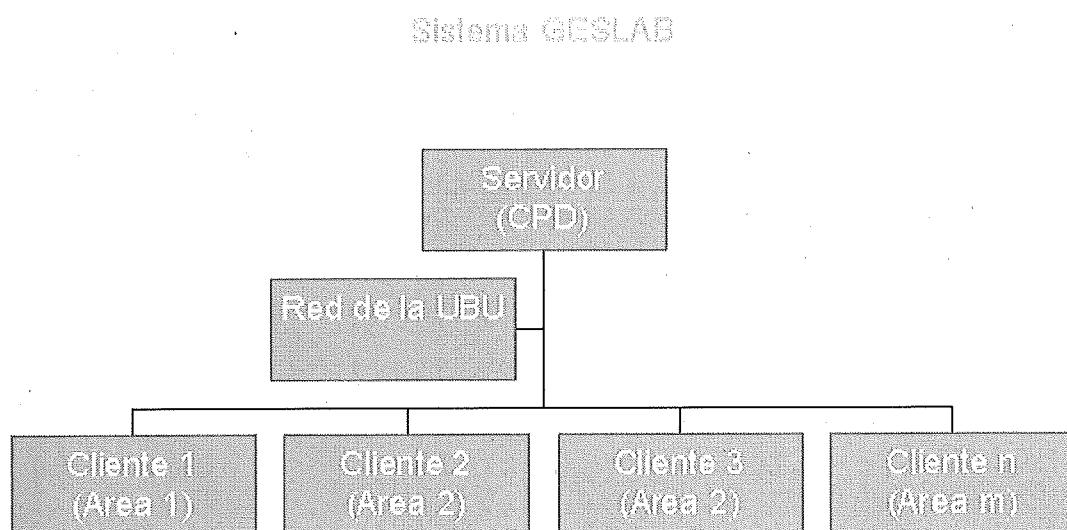
## 1- Introducción

### 1.1- Descripción y Constitución del Sistema GESLAB

#### Descripción

GESLAB es un sistema con arquitectura cliente/servidor<sup>1</sup> que gestiona el inventario de los laboratorios de química de la Universidad de Burgos. Está organizado por departamentos y áreas, cada una de las cuales puede gestionar su propio inventario, pero no el de las demás.

El sistema organiza el inventario de cada área en fichas, según el producto, su calidad, su localización, etc. Controla las fechas en que entran y salen los envases de cada ficha, y a partir de estos datos calcula las existencias de cada ficha que hay actualmente en el área. Además permite consultar las existencias de los otros laboratorios, de tal forma que unos laboratorios puedan pedir a otros productos que les faltan, y así aprovechar mejor los recursos de la Universidad.



<sup>1</sup> **Cliente/Servidor.** Arquitectura compuesta por un ordenador que pide datos y los maneja (cliente) y otro ordenador que los proporciona (servidor).

## **Constitución**

El sistema está constituido por un servidor en el que se almacenan los datos y varios clientes que realizan operaciones sobre dichos datos.

El servidor está formado por un PC que se encuentra en el CPD de la Universidad, y los clientes son varios ordenadores PC situados en cualquiera de los laboratorios.

### **Requisitos mínimos de un cliente:**

- Procesador compatible x86, preferiblemente Intel Pentium II y superiores para poder ejecutar el programa con rapidez.
- 32 MB o más de Memoria RAM (recomendado 64 MB)
- Sistema operativo Microsoft Windows de 32 bits (95, 98, NT)
- 15 MB de espacio libre en el disco duro aproximadamente
- Conexión a la red de la Universidad de Burgos

## **1.2- Instalación de la Aplicación**

---

La aplicación puede instalarse desde un CD-ROM, varios discos o bien desde un directorio compartido.

Los pasos para instalar la aplicación son los siguientes:

### **1. Ejecución del Programa de Instalación**

- Desde CD-ROM:
  - Desde el escritorio de Windows, hacer click en Mi PC o bien arrancar el Explorador de Windows.
  - Pinchar en la Unidad del CD-ROM y ejecutar el fichero Setup.exe
- Desde disquetes:
  - Desde el escritorio de Windows, hacer click en Mi PC o bien arrancar el Explorador de Windows.
  - Pinchar en la Unidad de disco (A:) y ejecutar el fichero Setup.exe
- Desde un directorio compartido en un ordenador de la red:
  - Desde el escritorio de Windows, hacer click en Entorno de Red o bien arrancar el Explorador de Windows.
  - Seleccionar el ordenador y el directorio compartido donde se encuentra el programa
  - Ejecutar el fichero Setup.exe

## 2. Instalación del origen de datos ODBC

El programa requiere una conexión ODBC<sup>2</sup> al servidor para poder acceder a los datos que es necesario crear manualmente.

Los pasos para crear la conexión son los siguientes:

- Abrir el Panel de Control desde el Menú Inicio / Configuración.
- Hacer doble click en el ícono ODBC.
- Agregar un origen de datos DSN de sistema<sup>3</sup> o bien DSN de usuario<sup>4</sup>. Para ello seleccionar la pestaña deseada y pinchar en el botón Agregar...
- Aparecerá una pantalla en la cual se deberá seleccionar un origen de datos de SQL Server. Si no existe, consulte al operador de su área.
- En las pantallas siguientes se llenarán los siguientes datos:
  - Nombre del origen de datos: dbGesLab
  - Servidor: 193.146.164.8
  - Biblioteca de red: dbmssocn (o bien Configuración del cliente: TCP/IP)
  - Pulsando Opciones >>
  - Nombre de la base de datos: DBLab.

El resto de parámetros no se modifican.

---

<sup>2</sup> **ODBC.** Es un estándar de Microsoft para acceder a las bases de datos remotamente. Tiene la ventaja de que no importa la base de datos a la que se accede, es decir; si se cambia la base de datos del servidor por otra de marca diferente, el programa funcionará sin modificarlo (modificando sólo la conexión). GESLAB accede a la base de datos mediante ODBC, y necesita saber a qué base de datos se accede y en qué ordenador está situada. En una conexión se especifican estos parámetros.

<sup>3</sup> **DSN de sistema.** Una ficha DSN incluye los parámetros de configuración de una conexión, como el nombre del servidor, marca de la base de datos, etc. En caso de que el ordenador local (cliente) tenga varios usuarios y cada usuario entre con un nombre de usuario y una contraseña diferente, un DSN de sistema permite que una conexión la puedan utilizar todos los usuarios de dicho ordenador, es decir, cualquier usuario del ordenador local podrá ejecutar GESLAB.

<sup>4</sup> **DSN de usuario.** Una ficha DSN incluye los parámetros de configuración de una conexión, como el nombre del servidor, marca de la base de datos, etc. En caso de que el ordenador local (cliente) tenga varios usuarios y cada usuario entre con un nombre de usuario y una contraseña diferente, un DSN de sistema permite que una conexión la pueda utilizar únicamente el usuario actual de dicho ordenador, es decir, ningún otro usuario del ordenador local podrá ejecutar GESLAB.

### **1.3- Darse de Alta en la Base de Datos**

---

Para poder utilizar el programa GESLAB en un área lo primero que se debe hacer es dar de alta dicha área en la base de datos. A cada área se le asignará un nombre de usuario y una contraseña.

Por seguridad es recomendable elegir una contraseña larga, a ser posible con letras y números combinados; no debe ser una palabra fácil de adivinar, y por supuesto distinta al nombre de usuario.

Deberá comunicarse al personal del CPD de la Universidad la intención de darse de alta en el sistema.

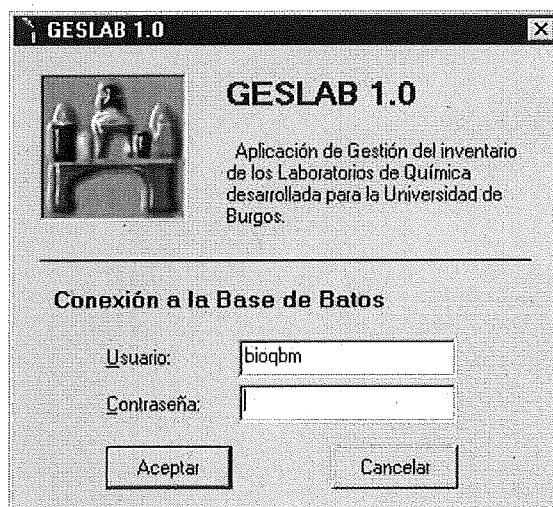
## 2- Primeros pasos

### 2.1- Ejecutar el programa

---

Una vez instalado (véase *Instalación de la aplicación*), el programa para acceder a la base de datos se ejecutará desde el Menú Inicio, en la carpeta Programas / Geslab. Al pinchar en el acceso directo a Geslab, aparecerá un formulario donde se pide un nombre de usuario y una contraseña. Si el área todavía no tiene asignado un nombre de usuario y una contraseña, deberá darse de alta en la base de datos.

Después de haber introducido el nombre de usuario y la contraseña se pinchará en el botón *Aceptar*. Si el nombre de usuario o la contraseña son erróneos, el programa no arrancará; permitirá tres intentos para identificarse en el sistema y al tercer fallo se cerrará la aplicación.



## 2.2- Crear una ficha

Para añadir una ficha nueva hay que colocarse en el formulario principal, y luego en la pestaña *Entradas de Compuestos* o bien en *Salidas de Compuestos*.

The screenshot shows a software interface for managing compound inputs. At the top, there's a menu bar with 'Ficha', 'Conexión', 'Productos', 'Localización', 'Calidad', 'Marca', 'Proveedores', 'Informes', and 'Ayuda'. Below the menu is a toolbar with buttons for 'Filtrar Consultas' (Filter Consultations), 'Entradas de Compuestos' (Inputs of Compounds), 'Salidas de Compuestos' (Outputs of Compounds), 'Existencias en mi Área' (Stocks in my Area), and 'Existencias en Todas las Áreas' (Stocks in All Areas). The main area is titled 'Ficha' and contains fields for 'Producto' (Acido Sulfúrico), 'Fórmula' (H<sub>2</sub>SO<sub>4</sub>), 'Departamento' (Biotecnología y Bioinformática), 'Área' (Biotecnología y Biología Molecular), 'Centro' (Facultad de Ciencias), 'Localización' (Nevadas), 'Proveedor' (Iberia Pastes), and 'Marca' (Merck). There are also buttons for 'Ver Entradas' (View Inputs), 'Nueva Ficha' (New Record), 'Modificar Ficha' (Modify Record), 'Validar Operación' (Validate Operation), 'Cancelar' (Cancel), and 'Calidad' (Quality). To the right of the input fields are three small buttons with arrows pointing up, down, and left. Below the input fields is a table titled 'Entradas' (Inputs) with the following data:

Fecha	Fecha Caducidad	Lote	Unidades	Capacidad	g/ml	Residuo
22/10/1999			10	1000	g	n

Buttons for 'Nuevo' (New), 'Modificar Entrada' (Modify Input), 'Filtrar' (Filter), 'Entradas' (Inputs), 'Salidas' (Outputs), 'Existencias en mi Área' (Stocks in my Area), 'Existencias en Todas las Áreas' (Stocks in All Areas), and 'Sala' (Sala) are located at the bottom.

Después se pulsará el botón *Nueva Ficha*, que hará que se borren todos los campos del formulario y permitirá escribir en ellos. Si en la pestaña *Filtrar Consultas* se filtró por algún campo, aparecerán automáticamente estos valores en el Formulario.

Por ejemplo, si en el campo *Producto* de la pestaña *Filtrar Consultas* se tecleó "Etanol", al pulsar *Nueva Ficha* aparecerá "Etanol" en el campo *Producto* del formulario de ficha. Acto seguido se debe llenar los campos con un valor adecuado. Todos los campos, excepto producto, se llenarán con un valor que se encuentre en la lista desplegable. El único campo que puede quedar vacío será *Proveedor*, que se considerará como Ninguno. Los demás campos deben llenarse obligatoriamente excepto *Fórmula*, que se actualiza automáticamente al cambiar de producto.

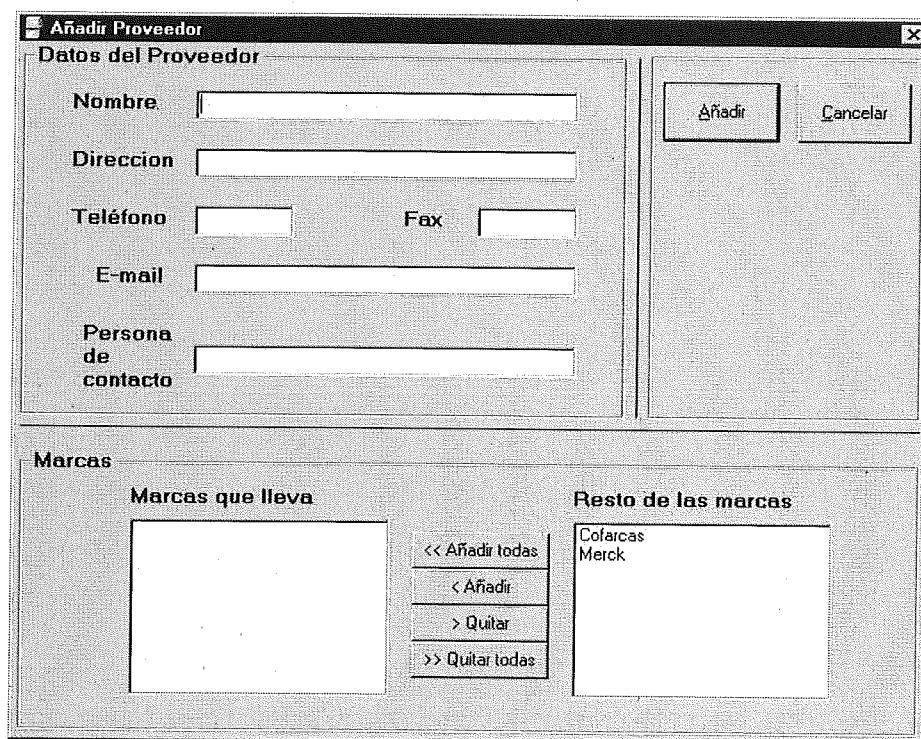
Una vez hecho esto, para guardar la nueva ficha en la base de datos se pulsará el botón *Validar Operación*, o bien se pulsará *Cancelar*.

## Añadir valores nuevos en los campos localización, proveedor, marca y calidad

En el caso de los campos *marca* y *calidad*, si se escribe en estos campos un valor que no aparece en sus respectivas listas desplegables, se preguntará si se quiere añadir a la base de datos.

Por ejemplo, si la marca del producto no aparece en la lista desplegable (no está almacenada en la base de datos), se escribe dicha marca en el campo *Marca* y al hacer click con el ratón en otra parte aparecerá un mensaje que pregunta si se quiere añadir la nueva marca a la base de datos. Si ha habido una equivocación al escribir el nombre de la marca, se puede rectificar pulsando “No”. En caso de que se pulse “Sí”, la marca se añadirá a la base de datos.

Si lo que se añade es un nuevo *proveedor* o *localización*, aparecerá otro formulario en el que se llenarán todos los campos del nuevo proveedor o de la nueva localización.



Nueva Localización

Localización	<input type="text"/>
Departamento	Biotecnología y C. Alim.
Área	Bioquímica y Biología Molecular
Centro	Facultad de Ciencias
Ocultar a las demás áreas	No

Añadir      Cancelar

## 2.3- Modificar los datos de una ficha

Para cambiar los datos de una ficha hay que posicionarse en el formulario principal, y luego en la pestaña *Entradas de Compuestos* o bien en *Salidas de Compuestos*.

The screenshot shows a software application window titled "Ficha". At the top, there's a menu bar with options like "Conexión", "Productos", "Localización", "Calidad", "Marca", "Proveedores", "Informes", and "Ayuda". Below the menu, there are four tabs: "Entradas de Compuestos" (which is selected and highlighted in blue), "Salidas de Compuestos", "Existencias en mi Área", and "Existencias en Todas las Áreas". The main working area has several input fields and dropdown menus. The "Producto" field contains "Acido Sulfúrico", and the "Fórmula" field contains "H<sub>2</sub>SO<sub>4</sub>". There are dropdowns for "Departamento" (set to "Bioquímica y C. Aéreos"), "Área" (set to "Bioquímica y Biología Molecular"), "Centro" (set to "Facultad de Ciencias"), "Localización" (set to "Planta"), "Proveedor" (set to "None"), "Marca" (set to "Merck"), and "Calidad" (set to "PA"). Below these fields are several buttons: "Ver Entradas", "Nueva Ficha", "Modificar Ficha", "Validar Operación", "Cancelar", "Entradas" (with a small table showing a single row for "22/10/1999"), "Nueva Entrada", and "Modificar Entrada". At the bottom of the window are more buttons: "Ficha", "Entradas", "Salidas", "Existencias en mi Área", "Existencias en Todas las Áreas", and "Salir".

Después se pulsará el botón *Modificar Ficha*, que hará que se pueda escribir en los campos del formulario.

Acto seguido se debe modificar los campos con un valor adecuado. Todos los campos, excepto producto, se rellenarán con un valor que se encuentre en la lista desplegable. El único campo que puede quedar vacío será Proveedor, que se considerará como Ninguno. Los demás campos deben rellenarse obligatoriamente excepto *Fórmula*, que se actualiza automáticamente al cambiar de producto.

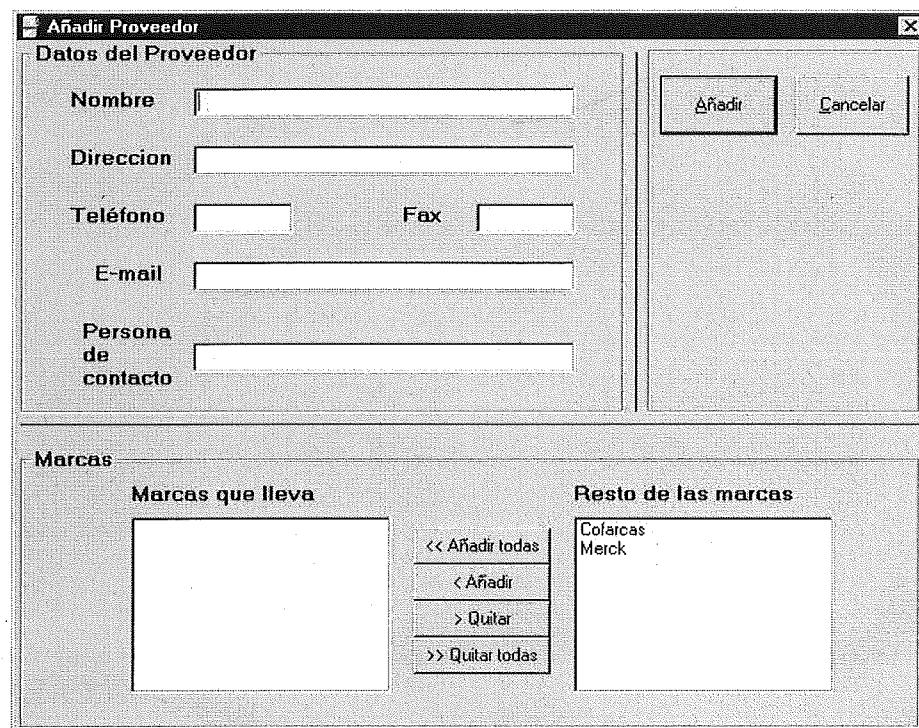
Una vez hecho esto, para guardar los cambios en la base de datos se pulsará el botón *Validar Operación*, o bien se pulsará *Cancelar*.

## Añadir valores nuevos en los campos localización, proveedor, marca y calidad

En el caso de los campos *marca* y *calidad*, si se escribe en estos campos un valor que no aparece en sus respectivas listas desplegables, se preguntará si se quiere añadir a la base de datos.

Por ejemplo, si la marca del producto no aparece en la lista desplegable (no está almacenada en la base de datos), se escribe dicha marca en el campo *Marca* y al hacer click con el ratón en otra parte aparecerá un mensaje que pregunta si se quiere añadir la nueva marca a la base de datos. Si ha habido una equivocación al escribir el nombre de la marca, se puede rectificar pulsando “No”. En caso de que se pulse “Sí”, la marca se añadirá a la base de datos.

Si lo que se añade es un nuevo *proveedor* o *localización*, aparecerá otro formulario en el que se llenarán todos los campos del nuevo proveedor o de la nueva localización.



Nueva Localización

Localización	<input type="text"/>
Departamento	Biotecnología y C. Alim.
Área	Bioquímica y Biología Molecular
Centro	Facultad de Ciencias
Ocultar a las demás áreas	<input checked="" type="checkbox"/> No

Añadir Cancelar

## 2.4- Filtrar Consultas

En un área suele haber un gran número de fichas almacenadas, por lo que la búsqueda de una ficha puede ser un proceso largo y aburrido.

Para evitar esto se utiliza el filtrado de consultas, en el que sólo aparecerán por pantalla las fichas deseadas, y afectará tanto a las fichas de nuestra área como a las fichas de todas las áreas.

The screenshot shows the 'Filtrar Consultas' (Filter Queries) window of the GESLAB software. At the top, there are several tabs: 'Conexión', 'Productos', 'Localización', 'Calidad', 'Marca', 'Proveedores', 'Informes', and 'Ayuda'. Below the tabs, there is a search bar labeled 'Consultar por...' and two input fields: 'Producto' and 'Fórmula'. Underneath these are dropdown menus for 'Departamento' (set to 'Biotecnología y C. Alm.'), 'Área' (set to 'Bioquímica y Biología Molecular'), 'Centro' (set to 'Todos'), 'Localización' (set to 'Todas'), 'Proveedor' (set to 'Todos'), 'Marca' (set to 'Todas'), and 'Calidad' (set to 'Todas'). There is also a checkbox 'Sólo Ocultas a las Demás Áreas' which is unchecked. Below these filters is a section titled 'Entradas y Salidas' with a checkbox 'Sólo Residuos' which is checked. It includes date range inputs for 'Fecha Entradas' (From / /, To / /), 'Fecha Salidas' (From / /, To / /), and 'Fecha Caducidad' (From / /, To / /). At the bottom of the window are five buttons: 'Filtrar', 'Entradas', 'Salidas', 'Existencias en mi Área', and 'Existencias en Todas las Áreas'.

En el sistema GESLAB es posible filtrar consultas por:

- Producto
- Fórmula
- Departamento
- Área
- Centro
- Localización
- Proveedor
- Marca
- Calidad

- Residuo (Sí o No)
- Fecha de Entrada
- Fecha de Salida
- Fecha de Caducidad

El filtro no distingue mayúsculas de minúsculas ni caracteres acentuados en ningún campo.

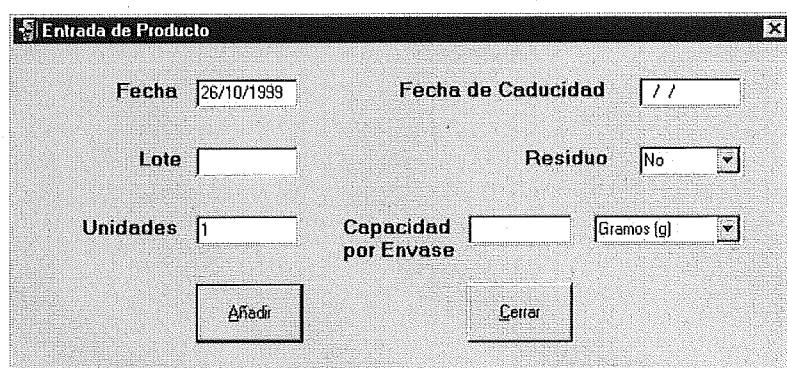
Si no hay ninguna ficha que cumpla todas las condiciones del filtro, al cambiar de pestaña aparecerá un mensaje “No hay Registros”. Por ejemplo, al seleccionar productos de un área que no es la nuestra, en las pestañas *Entradas de Compuestos*, *Salidas de Compuestos* y *Existencias en mi Área* aparecerá “No hay Registros”. En *Existencias en Todas las Áreas* aparecerán los registros que esa área haya dejado visibles a los demás.

## 2.5- Introducir Entradas y Salidas

Cuando llega un pedido al laboratorio, hay que registrar en el ordenador una entrada por cada producto, y si se acaba algo (un frasco con una disolución, o cualquier tipo de envase), se registrará una salida.

El procedimiento para introducir una entrada o una salida es:

- Seleccionar un filtro lo más preciso posible para que la búsqueda de la ficha adecuada sea rápida (véase *Filtrar Consultas*).
- Colocarse en la pestaña *Entradas de Compuestos* o bien en *Salidas de Compuestos* y buscar la ficha mediante los botones Primero, Anterior, Siguiente, Último.
- En caso de que no haya una ficha en la base de datos para ese producto, habrá que crear una nueva(véase *Crear una ficha*).
- Una vez que aparece por pantalla la ficha deseada, pinchar en el botón *Ver Entradas* o *Ver Salidas*.
- Pinchar en el botón *Nueva Entrada* o *Nueva Salida*.
- Aparecerá uno de los siguientes formularios (los campos son idénticos):



**Salida de Producto**

<b>Fecha:</b> <input type="text" value="26/10/1999"/>	<b>Fecha de Caducidad:</b> <input type="text" value="//"/>	
<b>Lote:</b> <input type="text"/>	<b>Residuo:</b> <input type="text" value="No"/> <input type="button" value="▼"/>	
<b>Unidades:</b> <input type="text" value="1"/>	<b>Capacidad por Envase:</b> <input type="text"/>	<b>Gramos (g):</b> <input type="text"/> <input type="button" value="▼"/>
<input type="button" value="Añadir"/>	<input type="button" value="Cerrar"/>	

## 2.6- Modificar Entradas y Salidas

---

Si ha habido un error al introducir una entrada o una salida, se pueden corregir estos datos.

El procedimiento para modificar una entrada o una salida es:

- Seleccionar un filtro lo más preciso posible para que la búsqueda de la ficha adecuada sea rápida (véase *Filtrar Consultas*).
- Colocarse en la pestaña *Entradas de Compuestos* o bien en *Salidas de Compuestos* y buscar la ficha mediante los botones Primero, Anterior, Siguiente, Último.
- Una vez que aparece por pantalla la ficha deseada, pinchar en el botón *Ver Entradas* o *Ver Salidas*.
- Seleccionar la fila a modificar y pinchar el botón *Modificar Entrada* o *Modificar Salida*.
- Aparecerá uno de los siguientes formularios (los campos son idénticos):

The dialog box is titled "Entrada de Producto". It has several input fields:

- "Fecha": 22/10/1999
- "Fecha de Caducidad": 77
- "Lote": (empty)
- "Residuo": No
- "Unidades": 10
- "Capacidad por Envase": 1000
- "Gramos (g)": (empty)

At the bottom are two buttons: "Modificar" (Modify) and "Cancelar" (Cancel).

 Salida de Producto

Fecha	25/10/1999	Fecha de Caducidad	/ /
Lote	[ ]	Residuo	No <input type="button" value="▼"/>
Unidades	1 <input type="button" value="▼"/>	Capacidad por Envase	1000 <input type="button" value="▼"/> Gramos (g)
<input type="button" value="Modificar"/>		<input type="button" value="Cancelar"/>	

## 2.7- Consultar Existencias en mi Área

Los pasos para conocer las existencias de un producto que hay en mi área son los siguientes:

- Seleccionar un filtro lo más preciso posible para que la búsqueda de la ficha adecuada sea rápida (véase *Filtrar Consultas*).
- Colocarse en la pestaña *Existencias en mi Área* y buscar la ficha mediante los botones Primero, Anterior, Siguiente, Último.
- Una vez que aparece por pantalla la ficha deseada, pinchar en el botón *Ver Existencias*.
- Aparecerán las cantidades de envases que posee el área agrupadas por capacidad. Para obtener estas cantidades se restan las entradas menos las salidas de esa ficha.
- Para imprimir un informe de todas las fichas (filtradas), pinchar el botón *Crear Informe* (véase *Imprimir Informes de un Área*).

## 2.8- Consultar Existencias en Todas las Áreas

Los pasos para conocer las existencias de un producto que hay en mi área son los siguientes:

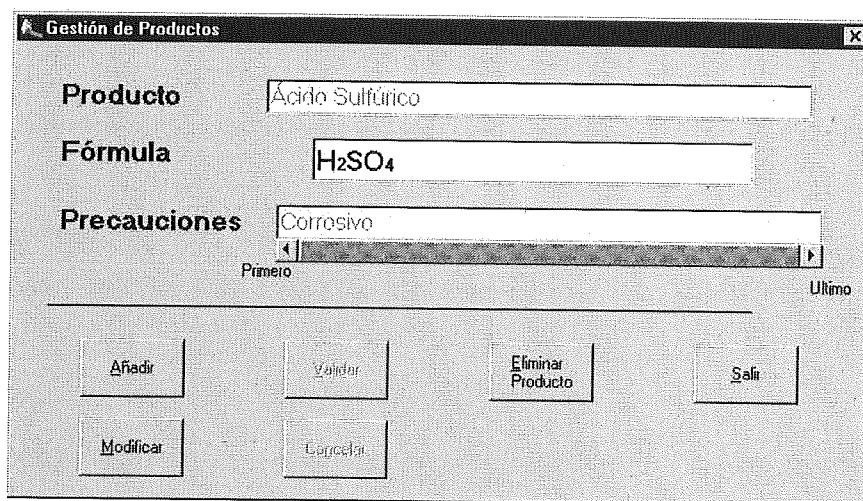
- Seleccionar un filtro lo más preciso posible para que la búsqueda de la ficha adecuada sea rápida (véase *Filtrar Consultas*).
- Colocarse en la pestaña *Existencias en Todas las Áreas* y buscar la ficha mediante los botones Primero, Anterior, Siguiente, Último.
- Una vez que aparece por pantalla la ficha deseada, pinchar en el botón *Ver Existencias*.
- Aparecerán las cantidades de envases que posee el área agrupadas por capacidad. Para obtener estas cantidades se restan las entradas menos las salidas de esa ficha.
- Para imprimir un informe de todas las fichas (filtradas), pinchar el botón *Crear Informe* (véase *Imprimir Informes de Todas las Áreas*).

Unidades	Capacidad	q_ml
9	1000	g

### 3- Avanzado

#### 3.1- Gestionar Productos

Para abrir el formulario de Gestión de Productos se selecciona el menú *Productos* y la opción *Gestión Completa de Productos*.



##### Añadir un producto nuevo

Para añadir un producto nuevo, lo primero que se debe hacer es pulsar el botón *Añadir*. Entonces se vaciarán todos los campos y se deberá escribir en ellos el nombre del producto, su fórmula y las precauciones que hay que tomar al tratar dicho producto. Una vez que los datos son correctos se pulsará el botón *Validar*; en caso de equivocación se pulsará *Cancelar* y el nuevo producto no se almacenará en la base de datos.

##### Modificar los datos de un producto

Para modificar los datos de un producto existente, lo primero que se debe hacer es dirigirse al producto a modificar mediante el *scroll*. Una vez que el producto aparece por pantalla, se pulsará el botón *Modificar*. Entonces se podrá cambiar el contenido de los campos *Fórmula* y *Precauciones*. Una vez que los datos son correctos se pulsará el

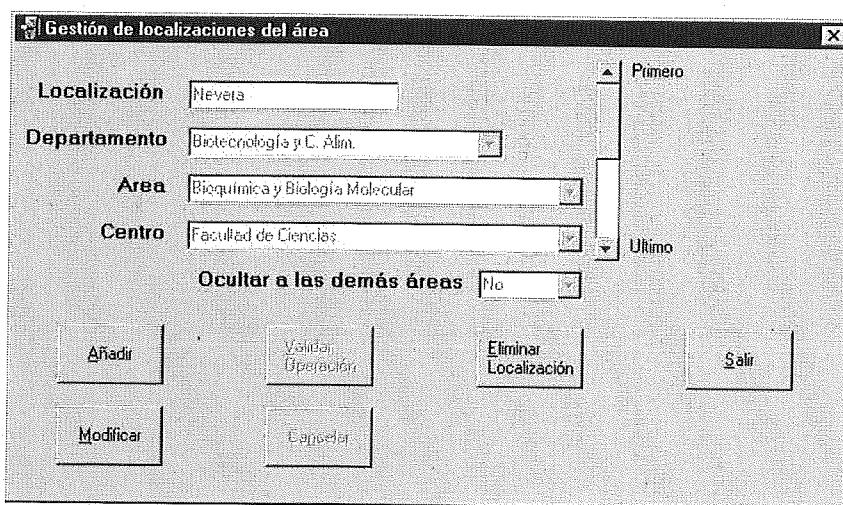
botón *Validar*; en caso de equivocación se pulsará *Cancelar* y los cambios no se almacenarán en la base de datos.

### **Eliminar un producto**

Esta opción permite borrar de la base de datos el producto actual. Para que sea posible eliminarlo no debe haber ninguna ficha de dicho producto en la base de datos.

### 3.2- Gestionar Localizaciones

Para abrir el formulario de Gestión de Localizaciones se selecciona el menú *Localización* y la opción *Gestión de Localizaciones*.



#### Añadir una localización nueva

Para añadir una localización nueva, lo primero que se debe hacer es pulsar el botón *Añadir*. Entonces se vaciará el campo *Nombre* y se deberá escribir en él el nombre de la localización; el área, departamento y centro normalmente no será necesario modificarlos. Una vez que los datos son correctos se pulsará el botón *Validar*; en caso de equivocación se pulsará *Cancelar* y la nueva localización no se almacenará en la base de datos.

#### Modificar los datos de una localización

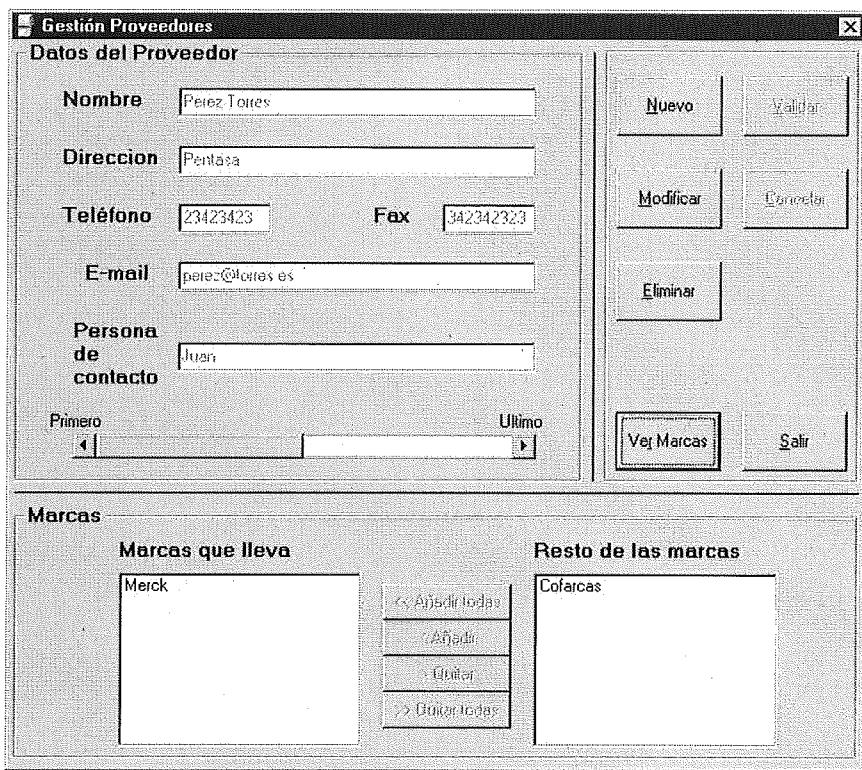
Para modificar los datos de una localización existente, lo primero que se debe hacer es dirigirse a la localización a modificar mediante el *scroll*. Una vez que la localización aparece por pantalla, se pulsará el botón *Modificar*. Entonces se podrá cambiar el contenido de todos los campos, incluido el nombre. Una vez que los datos son correctos se pulsará el botón *Validar*; en caso de equivocación se pulsará *Cancelar* y los cambios no se almacenarán en la base de datos.

## **Eliminar una localización**

Esta opción permite borrar de la base de datos la localización actual. Para que sea posible eliminarla no debe haber ninguna ficha situada en dicha localización en la base de datos.

### 3.3- Gestionar Proveedores

Para abrir el formulario de Gestión de Proveedores se selecciona el menú *Proveedores* y la opción *Gestión de Proveedores*.



#### Añadir un proveedor nuevo

Para añadir un proveedor nuevo, lo primero que se debe hacer es pulsar el botón *Añadir*. Entonces se vaciarán todos los campos y se deberá escribir en ellos el nombre del proveedor, su dirección, teléfono, fax, dirección de correo electrónico y una persona de contacto. Si las marcas están visibles, se deberán añadir las marcas que dicho proveedor lleva. En caso de no estar seguro es conveniente añadir todas las marcas para que al introducir una nueva ficha se pueda establecer una relación entre el nuevo proveedor y cualquier marca. Una vez que los datos son correctos se pulsará el botón *Validar*; en caso de equivocación se pulsará *Cancelar* y el nuevo proveedor no se almacenará en la base de datos.

### **Modificar los datos de un proveedor**

Para modificar los datos de un proveedor existente, lo primero que se debe hacer es dirigirse al proveedor a modificar mediante el *scroll*. Una vez que el proveedor aparece por pantalla, se pulsará el botón *Modificar*. Entonces se podrá cambiar el contenido de los campos, incluidas las marcas que lleva y las que no lleva. Una vez que los datos son correctos se pulsará el botón *Validar*; en caso de equivocación se pulsará *Cancelar* y los cambios no se almacenarán en la base de datos.

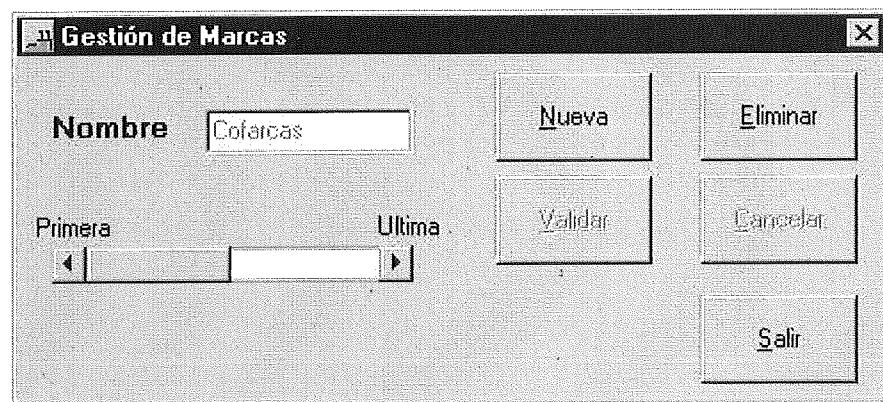
### **Eliminar un proveedor**

Esta opción permite borrar de la base de datos el proveedor actual. Para que sea posible eliminarlo no debe haber ninguna ficha con dicho proveedor en la base de datos.

### 3.4- Gestionar Marcas

---

Para abrir el formulario de Gestión de Marcas se selecciona el menú *Marca* y la opción *Gestión de Marcas*.



#### Añadir una marca nueva

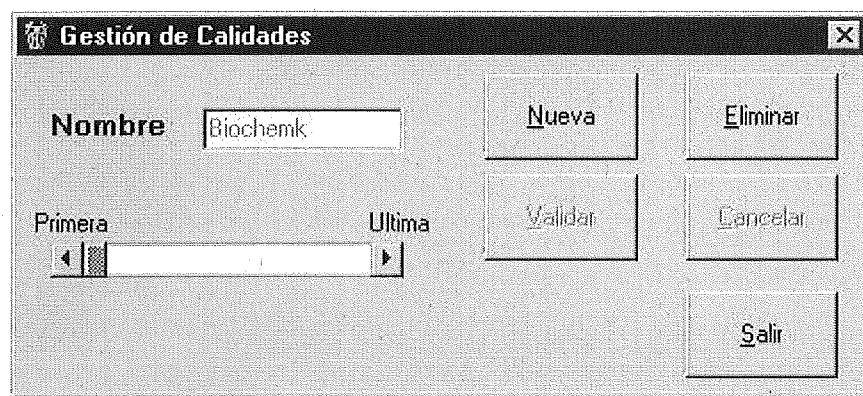
Para añadir una marca nueva, lo primero que se debe hacer es pulsar el botón *Añadir*. Entonces se vaciará el campo *Nombre* y se deberá escribir en él el nombre de la nueva marca. Una vez que los datos son correctos se pulsará el botón *Validar*; en caso de equivocación se pulsará *Cancelar* y la nueva marca no se almacenará en la base de datos.

#### Eliminar una marca

Esta opción permite borrar de la base de datos la marca actual. Para que sea posible eliminarla no debe haber ninguna ficha de dicha marca en la base de datos.

### 3.5- Gestionar Calidades

Para abrir el formulario de Gestión de Calidades se selecciona el menú *Calidad* y la opción *Gestión de Calidades*.



#### Añadir una calidad nueva

Para añadir una calidad nueva, lo primero que se debe hacer es pulsar el botón *Añadir*. Entonces se vaciará el campo *Nombre* y se deberá escribir en él el nombre de la nueva calidad. Una vez que los datos son correctos se pulsará el botón *Validar*; en caso de equivocación se pulsará *Cancelar* y la nueva calidad no se almacenará en la base de datos.

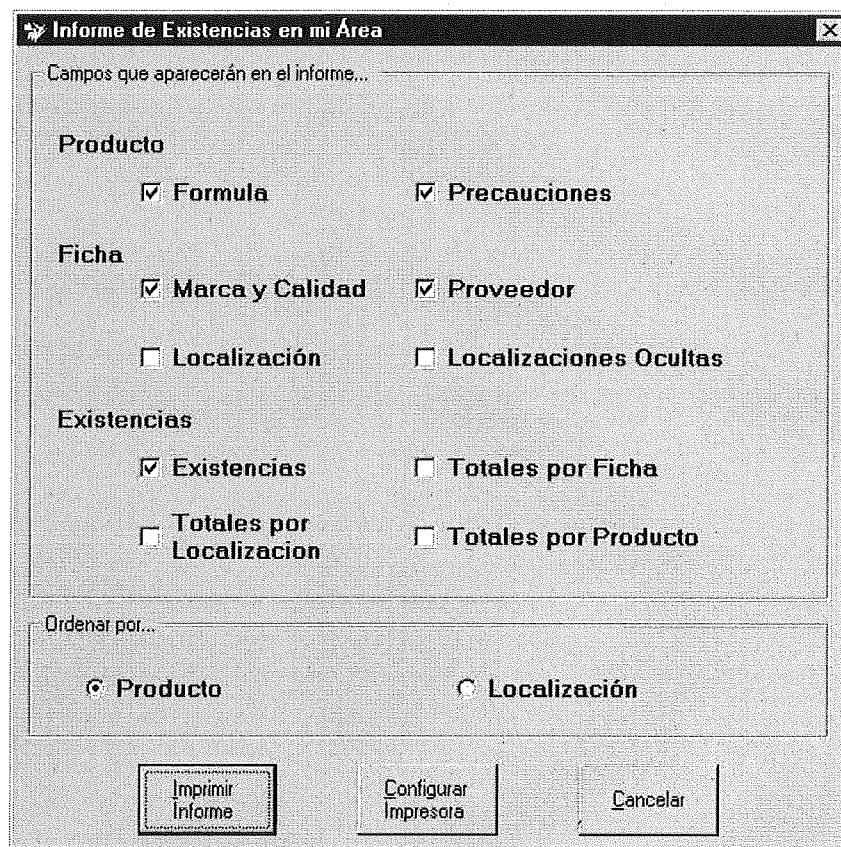
#### Eliminar una calidad

Esta opción permite borrar de la base de datos la calidad actual. Para que sea posible eliminarla no debe haber ninguna ficha de dicha calidad en la base de datos.

### 3.6- Imprimir Informes de Existencias en mi Área

Al obtener un informe de existencias en mi área se imprime en papel todas las fichas que aparecen en la pestaña *Existencias en mi Área*. Los filtros establecidos en la pestaña de filtro también son válidos para el informe.

Para imprimir un informe de mi área se debe abrir el formulario Informe de Existencias en mi Área. Para abrir este formulario se selecciona el menú *Informes* y la opción *Existencias en mi Área* o bien desde la pestaña *Existencias en mi Área* se pulsa el botón *Crear Informe*.



### 3.7- Imprimir Informes de Existencias en Todas las Áreas

Al obtener un informe de existencias en todas las áreas se imprime en papel todas las fichas que aparecen en la pestaña *Existencias en Todas las Áreas*. Los filtros establecidos en la pestaña de filtro también son válidos para el informe.

Para imprimir un informe de todas las áreas se debe abrir el formulario Informe de Existencias en Todas las Áreas. Para abrir este formulario se selecciona el menú *Informes* y la opción *Existencias en Todas las Áreas* o bien desde la pestaña *Existencias en Todas las Áreas* se pulsa el botón *Crear Informe*.

