



**BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA**

---

---

**Facultad De Ciencias De La Computación**

**Sistema de base de datos para el manejo  
Y control de laboratorios de química  
MACOLAQUI**

**TESIS**

**Que para obtener el grado de:  
Licenciada En Ciencias De La Computación**

**Presenta:**

Ma. Isabel Gutiérrez Luna

**Asesor:**

M. C. Mireya Tovar Vidal



Puebla, Puebla.

Junio 2012

---

---

## Agradecimientos

Primero que nada doy gracias a dios porque sé, que es él quien me puso en este camino a seguir y luchar día a día para ser mejor, desde el momento que decidí tomar esta carrera tan hermosa que me ha llenado en todos los aspectos, el ha estado en todos mis momentos buenos y malos sin desviar mi atención en un objetivo, llegar a la conclusión de este proyecto.

Deseo expresar mi más sincero agradecimiento a los profesores de la Facultad Ciencias de la Computación de la Benemérita Universidad Autónoma de Puebla por su enseñanza tan eficaz y avanzada que me ha ayudado para resolver proyectos profesionales como ha sido el presente documento.

Además, manifiesto mi agradecimiento especial a la M. C. Mireya Tovar Vidal por su apoyo en la revisión de este manuscrito, por su tiempo, por su disponibilidad, su forma tan profesional y sus comentarios tan puntuales, ha sido un ejemplo a seguir.

Gracias también a mis padres Alberto Gutiérrez Morales y Rosa Luna Mier por su apoyo tan constante que me han dado desde el primer momento que inicie esta hermosa aventura, gracias papa por ser un ejemplo y enseñarme a ser responsable en todos los aspectos, gracias mama por apoyarme día a día como madre y como abuela, que sin su ayuda muchas de mis actividades no hubiera logrado llevar a cabo. Sé que este proyecto no hubiera podido haberlo logrado sin su valioso apoyo constante.

Gracias a mis hermanos en especial a Raúl Gutiérrez Luna por sus consejos incansables de seguir adelante y terminar lo iniciado, por apoyarme desde los inicios de este importante proyecto de mi vida tanto familiarmente como económicamente.

Quiero dedicar este documento especialmente a mi hijo Jesús Alberto Gutiérrez Luna ya que será una valiosa herramienta para él en su futuro inmediato. Así como también hijo, demostrarte que si se puede lograr a obtener algo tan deseado siempre y cuando te lo plantees como una gran meta en tu vida. Para poder cerrar e iniciar otras como te las propongas.

---

## Resumen

Tesis presentada sobre como controlar de una forma automatizada laboratorios de química en una universidad, comenzando por asignar permisos a usuarios de tipo Director, Almacenista, Secretario Académico, Secretario Administrativo, Profesores. De acuerdo a tipo de usuario se asignan permisos a ciertas áreas del sistema, de esta forma se tiene mejor control de información. El sistema puede controlar desde registrar usuarios, validar perfiles, así como administrar cristalería, instrumental, sustancias químicas, otros materiales, laboratorios, personal, asignaturas, deudores, prácticas programadas y prácticas de profesores de un laboratorio de química.

Para desarrollar un sistema de calidad se usaron herramientas, métodos y procedimientos de la Ingeniería de Software, como es el modelo de cascada, el cual me describe el ciclo de vida de mi sistema, mostrando sus fases de desarrollo.

De acuerdo al plan del desarrollo del software, independientemente del ciclo de vida elegido, se utilizó el proceso de desarrollo de software que contiene tres fases, el cual en cada una de ellas se aplicaron metodologías, como son:

- *Fase de definición del software:* Análisis de requerimientos, análisis del sistema y planificación del proyecto del software.
- *Fase de desarrollo del software:* Diseño de software, codificación y prueba de software.
- *Fase de mantenimiento del software:* Corrección, adaptación y mejora.

En las unidades de este documento se ven reflejadas y aplicadas cada una de dichas metodologías de la Ingeniería de Software.

El lenguaje de programación Progress 9 se usó para la codificación del sistema.

---

---

## Introducción

El soporte computacional para el manejo y control de un laboratorio de química es una herramienta de primera importancia para mejorar el uso de ciertos compuestos químicos, así como, los tipos de laboratorios en que se utilizan cada uno, los profesores que controlan esas sustancias y que imparten las diferentes asignaturas, un control exhaustivo de la existencia de cada uno de los reactivos, debe brindar las asignaturas que mayor uso hacen del laboratorio y del material de trabajo. Así como del personal encargado del laboratorio. Actualmente no se cuenta con la necesaria infraestructura para controlarlos. Anteriormente se había considerado manejar las sustancias químicas en programas aplicativos sin tener la suficiente capacidad de control.

En la actualidad existe un control en papel para cada sustancia, este trabajo es sencillo cuando el número de sustancias es pequeño, pero cuando existe una gran demanda y el número de sustancias es grande, sale del control humano. De esta forma se ha sugerido que para una mejor administración de las sustancias se manejen bases de datos ya que una de sus ventajas o atributos es que controle una gran cantidad de información, permitiendo así mejor manejo en las existencias de las sustancias. Un sistema de base de datos es básicamente un sistema computarizado para llevar registros. Es posible considerar a la propia base de datos como una especie de armario electrónico para archivar, es decir, es un depósito o contenedor de una colección de archivos de datos computarizados.

Los usuarios del sistema pueden realizar una variedad de operaciones sobre dichos archivos, por ejemplo: agregar nuevos archivos vacíos a la base de datos, insertar datos dentro de los archivos existentes, recuperar datos de los archivos existentes, modificar datos en archivos existentes, eliminar datos de los archivos existentes y eliminar archivos existentes de las bases de datos.

Los sistemas de bases de datos presentan numerosas ventajas que se pueden dividir en dos grupos: las que se deben a la integración de datos y las que se deben a la interface común que proporciona el Sistema Manejador de Bases de Datos (SMBD).

Ventajas por la integración de datos: control sobre la redundancia de datos, consistencia de datos, más información sobre la misma cantidad de datos, compartición de datos, mantenimiento de estándares.

Ventajas por la existencia del SMBD: mejora en la integridad de datos, mejora en la seguridad, mejora en la accesibilidad a los datos, mejora en la productividad, mejora en el mantenimiento gracias a la independencia de datos, aumento de la concurrencia, mejora en los servicios de copias de seguridad y de recuperación ante fallos.

Inconvenientes de los sistemas de bases de datos: complejidad, tamaño, costo económico del SMBD, costo del equipamiento adicional, costo de la conversión, prestaciones y vulnerable a los fallos.

Para la realización de este sistema es necesario incluir las etapas más importantes de la ingeniería del software, indicar los modelos del ciclo de vida, identificar requerimientos y el diagrama de flujo de datos. También, es importante estudiar los conceptos de las bases de datos que se verán en el capítulo 1, con la finalidad de llegar al diseño del sistema utilizando el modelo entidad-relación que se verá en el capítulo 2, aplicando las técnicas del modelo conceptual, lógico y físico y la normalización del mismo que se verá en el capítulo 3.

---

Al terminar la investigación, se habrá codificado un Software, el cual permitirá identificar las sustancias más utilizadas e importantes de un laboratorio químico, con esto se dará a conocer uno de los múltiples usos que pueden tener las bases de datos relacionales dentro de la computación, en especial para obtener estudios estadísticos en áreas tan importantes como la química. Finalmente se obtiene la implantación del software el cual tiene una interfaz amigable con el cliente mostrándole los resultados esperados.

---

---

# Índice

Agradecimientos .....	2
Resumen .....	3
Introducción .....	4
Capítulo I Marco teórico	
1.1 Introducción .....	7
1.1.1 Definición de Sistema o software .....	7
1.2 Ingeniería como opción de poder de cómputo .....	7
1.2.1 Calidad en el software .....	8
1.2.2 Ciclo de vida del software .....	8
1.3 Planificación gestión de proyectos .....	9
1.4. Determinación de requerimientos .....	9
1.5 Modelo de análisis .....	9
1.6 Administración de bases de datos .....	9
1.6.1 EL Sistema Manejador de bases de datos (SMBD) .....	10
1.6.2 Los tres niveles de la arquitectura de los sistemas de base de datos .....	11
1.6.3 Modelos de SMBD .....	11
1.7 Progress .....	15
1.8 Documentación de la etapa de ingeniería de sistemas .....	16
1.8.1 Tecnología Estratificada .....	16
Capítulo II Análisis del sistema	
2.1 Introducción .....	19
2.1.1 Planificación del proyecto de software .....	19
2.2 Plan de proyecto .....	19
2.2.1 Descripción del proyecto .....	19
2.2.2 Documentos relacionados .....	19
2.2.3 Recursos necesarios .....	19
2.2.4 Análisis de riesgos .....	19
2.3 Análisis del sistema .....	19
2.3.1 Principios del análisis .....	20
2.3.2 Modelo de análisis de un sistema .....	21
2.3.3 Herramientas del análisis estructurado .....	22
2.4 Especificación de requisitos del software .....	22
2.4.1 Introducción .....	23
2.4.2 Escenarios de uso .....	24
2.4.3 Modelo de datos .....	27
2.4.4 Modelo funcional .....	30
2.4.5 Modelo de comportamiento .....	35
2.4.6 Limitaciones y restricciones .....	36
2.4.7 Criterios de validación .....	36
Capítulo III Diseño del sistema	
3.1 Introducción .....	37
3.1.1 El diseño de base de datos .....	37
3.2. Especificación del diseño del software .....	38
3.2.1 Modelo conceptual .....	38
3.2.2 Diagrama entidad-relación .....	40
3.2.3 El documento del diseño lógico .....	40
3.2.4 Modelo físico .....	45
Capítulo IV Implantación del sistema y resultados	
4.1 Introducción .....	53
4.2 Análisis de la herramienta Progress .....	53
4.3 Implantación del sistema.....	53
Conclusiones .....	60
Bibliografía .....	62
Anexos .....	63
Anexo 1.1. Identificar los atributos y asociarlos a entidades y relaciones .....	63
Anexo 1.2. Determinar los dominios de los atributos .....	65

## Capítulo I. Marco teórico

### 1.1 Introducción

Para la realización de este sistema es necesario estudiar las etapas más importantes de la ingeniería del software, de esta forma en este capítulo hablaremos de la ingeniería como opción de cómputo, para llevar el control de la calidad de nuestro sistema siguiendo su metodología.

#### 1.1.1 Definición de Sistema o software.

Un proyecto de sistema o software es el proceso de gestión para la creación de un sistema o software, el cual encierra un conjunto de actividades, una de las cuales es la estimación; estimar es echar un vistazo al futuro y aceptamos resignados cierto grado de incertidumbre. Al estimar tomamos en cuenta no solo el procedimiento técnico a utilizar en el proyecto, sino que se toma en cuenta los recursos, costos y planificación. El tamaño del proyecto es otro factor importante que puede afectar la precisión de las estimaciones. A medida que el tamaño aumenta, crece rápidamente la interdependencia entre varios elementos del Software. La disponibilidad de información histórica es otro elemento que determina el riesgo de la estimación [2].

#### 1.2 Ingeniería como opción de poder de cómputo.

Durante las tres primeras décadas de la Informática, el principal desafío era el desarrollo del hardware de las computadoras, de forma que se redujera el costo de procesamiento y almacenamiento de datos.

La necesidad de enfoques sistemáticos para el desarrollo y mantenimiento de productos de software se patentizó en la década de 1960. En ésta década aparecieron las computadoras de la tercera generación y se desarrollaron técnicas de programación como la multiprogramación y el tiempo compartido. Mientras las computadoras estaban haciéndose más complejas, resultó obvio que la demanda por los productos de software creció en mayor cantidad que la capacidad de producir y mantener dicho software. Estas nuevas capacidades aportaron la tecnología necesaria para el establecimiento de sistemas computacionales interactivos, de multiusuario, en línea y en tiempo real; surgiendo nuevas aplicaciones para la computación, como las reservaciones aéreas, bancos de información médica, etc. [5].

Fue hasta el año 1968 que se convocó una reunión en Garmisch, Alemania Oriental estimulándose el interés hacia los aspectos técnicos y administrativos utilizados en el desarrollo y mantenimiento de software, y fue entonces donde se utilizó el término "Ingeniería de Software".

A lo largo de la década de los ochenta, los avances en microelectrónica han dado como resultado una mayor potencia de cálculo a la vez que una reducción de costo. Hoy el problema es diferente, el principal desafío es mejorar la calidad y reducir el costo.

La Ingeniería de Software es la ciencia que estudia y documenta un conjunto de principios, metodologías, métodos y técnicas que permiten construir software o sistemas de información de manera repetible y confiable [5]. Así que si se siguen las metodologías que la ingeniería de software propone, se tendrá la seguridad de que cada vez que se construya un sistema de software este cumplirá con las especificaciones y requisitos que se plantearon para su elaboración, es decir, proporcionará adecuadamente las funciones y servicios para los que fue creado. Además, se garantizará que el sistema es robusto, confiable y eficiente.

En otras palabras, al aplicar las metodologías de la ingeniería de software, se construirá un sistema de software de calidad.

A finales de los 70, inicio de los 80's (del siglo XX), surgió toda una revolución en el mundo del desarrollo de software. En aquellos tiempos, hacer cualquier aplicación requería mucho esfuerzo, sin herramientas visuales, mucha gente, muchas pruebas, muchos medios, lo cual no quiere decir que ahora sea fácil, pero lo que yo puedo hacer ahora solo, necesitaba hace 10 años de unas 3 personas para hacerlo. Las empresas, los desarrolladores, se dieron cuenta de que mantener ese ritmo, con aplicaciones monstruosas, con muchísimo código, tanto que al cabo de unos meses ya no se sabía ni como funcionaba, necesitaba una solución.

Los desarrollos de ingeniería de Software comenzaron con la técnica de programación y después fueron utilizados en otras fases del ciclo de vida del software. La programación estructurada fue seguida por otros métodos estructurados de análisis y también métodos estructurados de diseño. Además, comenzaron tecnologías orientadas a objeto. *En épocas tempranas la programación era la tarea de oro de ingeniería del software pero ahora la ingeniería y el diseño de requisito son más populares.* En los años 90s la gerencia de proyecto ganó interés y llegó a ser un componente importante en ingeniería del software. En la década pasada, los estándares de la ingeniería de Software y la madurez de proceso han caracterizado la industria del software como una disciplina madura.

### **1.2.1 Calidad en el software.**

En la actualidad los desarrolladores de software, coinciden en que el aplicar los principios de calidad al proceso de desarrollo de software, es una condición necesaria para garantizar el éxito de un proyecto [5].

### **1.2.2 Ciclo de vida del software**

Como todo producto de ingeniería, el software también tiene su ciclo de vida. Este corresponde al periodo desde que el sistema se concibe hasta que se deja de usar, pasando por su especificación, desarrollo, transferencia, y explotación [5]. En definitiva, el ciclo de vida del software corresponde a las fases involucradas en todo el periodo del sistema de software.

Existen distintos modelos para enfrentar el desarrollo de un sistema de software a lo largo de su ciclo de vida. Cada uno de estos tiene sus ventajas y desventajas de acuerdo a las condiciones del problema a desarrollar y la disponibilidad y madurez del cliente. Esto significa que no hay un único modelo de desarrollo, y por lo tanto, es responsabilidad del equipo de desarrollo decidir cual modelo elegir para las condiciones del problema en vista. Esta decisión es muy importante, ya que una elección equivocada puede significar el fracaso del proyecto. El modelo que se usa para la realización de este proyecto es:

\*Modelo de cascada: El modelo de desarrollo en cascada que se muestra en la Figura 1.1 se caracteriza debido a que sus fases se distribuyen en forma secuencial, una después de otra. Primero se realiza la fase de análisis. Una vez completada dicha fase, se desarrolla la fase de diseño. Luego, la fase de producción, que incluye el diseño detallado, programación y prueba del producto. Una vez finalizada dicha fase, se sigue a la fase de transferencia del sistema construido al ambiente de producción. Por último, se realiza la fase de operación y mantenimiento del sistema de software. El modelo de desarrollo se llama cascada, en donde el producto de una fase va cayendo a la siguiente para su procesamiento.

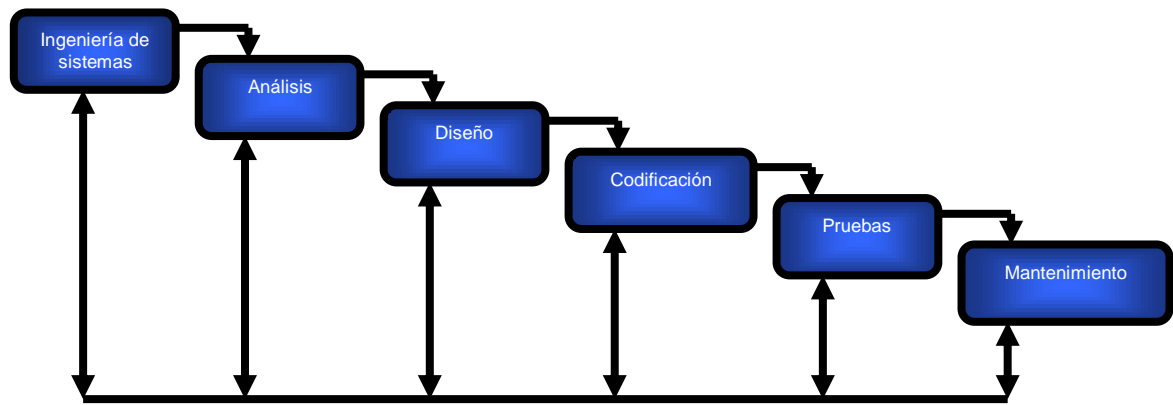


Figura 1.1. Modelo de cascada.

### 1.3 Planificación gestión de proyectos.

El ciclo de desarrollo del software incluye varias etapas, algunas de las cuales se repiten hasta que el sistema este completo y los clientes y usuarios satisfechos. Sin embargo, antes de comprometer los fondos para un desarrollo de software o el mantenimiento de un proyecto, por lo general el cliente quiere una estimación de cuanto costara el proyecto y cuanto durará.

La necesidad de administrar es una distinción importante entre un desarrollo profesional de software y la programación no profesional [5]. Se introduce el tema y se describen tres actividades importantes de administración: planeación, calendarización de proyectos y administración de riesgos.

### 1.4 Determinación de requerimientos.

Cuando un cliente solicita que se construya un nuevo sistema, tiene una noción de lo que el sistema debe hacer. El aspecto fundamental del análisis de sistemas es comprender las facetas importantes de la primera parte de la empresa que se encuentra bajo estudio. El Análisis de requerimientos de software: “Es el proceso de establecer con todo detalle los servicios que debe proporcionar el sistema y las restricciones con las cuales debe operar”[5].

### 1.5 Modelo de análisis.

Después de la planificación del proyecto, se inicia propiamente dicho, la primera etapa del desarrollo del sistema que consiste en elaborar el modelo del análisis del sistema.

Partiremos de algunas definiciones y posteriormente veremos como se aplican los principios de la teoría de sistemas al caso concreto de sistemas de información, lo cual nos permitirá facilitar el proceso de modelo del sistema [5].

Existen diferentes enfoques (metodologías) para modelar un sistema, pero tenemos dos grandes ramas:

1. Metodologías de análisis y diseño estructurado.
2. Metodologías de análisis y diseño orientado a objetos.

En este proyecto de se desarrolla la metodología de análisis y diseñó estructurado [5].

### 1.6 Administración de bases de datos

El administrador de datos y el administrador de la base de datos son las personas o grupos de personas encargadas de gestionar y controlar todas las actividades que tienen que ver con los datos de la empresa y con la base de datos, respectivamente.

**El administrador de datos** es quien entiende los datos y las necesidades de la empresa con respecto a los datos. Su trabajo es decidir qué datos deben almacenarse en la base de datos, establecer políticas para mantener y gestionar los datos una vez que hayan sido almacenados [3]. Un ejemplo de tal política sería la que estable quién puede realizar: qué operaciones sobre qué datos y en qué circunstancias.

La persona (o personas) que se encarga de implementar las decisiones del administrador de datos es el **administrador de la base de datos**. Su trabajo es crear la base de datos e implementar los controles necesarios para que se respeten las políticas establecidas por el administrador de datos. El administrador de la base de datos es el responsable de garantizar que el sistema obtenga las prestaciones deseadas, además de prestar otros servicios técnicos.

El administrador de datos dentro de sus funciones en las etapas del ciclo de vida está: la planificación de la base de datos, la definición del sistema, recolección y análisis de los requisitos, diseño conceptual y diseño lógico de la base de datos. El administrador de la base de datos maneja: selección de Sistema Manejador de Bases de Datos (SMBD), diseño de las aplicaciones, diseño físico, implementación, conversión y carga de datos, prueba y mantenimiento.

### **1.6.1 EL Sistema Manejador de bases de datos (SMBD).**

Se dice que los datos de la base de datos “persisten” debido a que una vez aceptados por el SMBD para entrar a la base de datos, en lo sucesivo solo pueden ser removidos de la base de datos por alguna solicitud explícita al SMBD, no como efecto lateral de algún programa que termina su ejecución [3]. Por lo tanto, esta noción de persistencia nos permite dar una definición más precisa del término “base de datos”:

“Una base de datos es un conjunto de datos persistentes que es utilizado por los sistemas de aplicación de alguna empresa dada”.

Una vez entendida la descripción de base de datos, daremos la descripción de un SMBD:

“Un sistema de base de datos es básicamente un sistema computarizado para llevar registros”.

A continuación se presentan los criterios para la aceptación de un SMBD:

- El modelo lógico en que esta basado son el relacional, el de red, el jerárquico y modelos orientados a objetos.
- El número de usuarios a los que da servicio el sistema. Los sistemas monousuario sólo atienden a un usuario a la vez, y su principal uso se da en los ordenadores personales. Los sistemas multiusuario, entre los que se encuentran la mayor parte de los SMBD, atienden a varios usuarios al mismo tiempo.
- El número de sitios en los que está distribuida la base de datos. Casi todos los SMBD son centralizados: sus datos se almacenan en un solo computador. Los SMBD centralizados pueden atender a varios usuarios, pero el SMBD y la base de datos en sí residen por completo en una sola máquina. En los SMBD distribuidos, la base de datos real y el propio software del SMBD pueden estar distribuidos en varios sitios conectados por una red. Los SMBD distribuidos homogéneos utilizan el mismo SMBD en múltiples sitios. Una tendencia reciente consiste en crear software para tener acceso a varias bases de datos autónomas preexistentes almacenadas en SMBD distribuidos

heterogéneos. Esto da lugar a los SMBD federados o sistemas multibase de datos en los que los SMBD participantes tienen cierto grado de autonomía local. Muchos SMBD distribuidos emplean una arquitectura cliente-servidor.

- El costo del SMBD. La mayor parte de los paquetes de SMBD cuestan entre 10.000 y 100.000 euros. Los sistemas monousuario más económicos para microcomputadores cuestan entre 100 y 3.000 euros. En el otro extremo, los paquetes más completos cuestan más de 100.000 euros.
- Los SMBD pueden ser de propósito general o de propósito específico. Cuando el rendimiento es fundamental, se puede diseñar y construir un SMBD de propósito especial para una aplicación específica, y este sistema no sirve para otras aplicaciones.

### 1.6.2 Los tres niveles de la arquitectura de los sistemas de base de datos.

Hay tres características importantes inherentes a los sistemas de bases de datos: la separación entre los programas de aplicación y los datos, el manejo de múltiples vistas por parte de los usuarios y el uso de un catálogo para almacenar el esquema de la base de datos [3]. En 1975, el comité ANSI-SPARC (American National Standard Institute - Standards Planning and Requirements Committee) propuso una arquitectura de tres niveles para los sistemas de bases de datos, que resulta muy útil a la hora de conseguir estas tres características.

El objetivo de la arquitectura de tres niveles es el de separar los programas de aplicación de la base de datos física. En esta arquitectura, el esquema de una base de datos se define en tres niveles de abstracción distintos:

- En el **nivel interno** se describe la estructura física de la base de datos mediante un esquema interno. Este esquema se especifica mediante un modelo físico y describe todos los detalles para el almacenamiento de la base de datos, así como los métodos de acceso.
- En el **nivel conceptual** se describe la estructura de toda la base de datos para una comunidad de usuarios (todos los de una empresa u organización), mediante un esquema conceptual. Este esquema oculta los detalles de las estructuras de almacenamiento y se concentra en describir entidades, atributos, relaciones, operaciones de los usuarios y restricciones. En este nivel se puede utilizar un modelo conceptual o un modelo lógico para especificar el esquema.
- En el **nivel externo** se describen varios esquemas externos o vistas de usuario. Cada esquema externo describe la parte de la base de datos que interesa a un grupo de usuarios determinado y oculta a ese grupo el resto de la base de datos. En este nivel se puede utilizar un modelo conceptual o un modelo lógico para especificar los esquemas.

### 1.6.3 Modelos de SMBD

Describiré a continuación los diferentes modelos empleados en los SMBD:

**En el modelo relacional** la base de datos es percibida por el usuario como un conjunto de tablas. Esta percepción es sólo a nivel lógico (en los niveles externo y conceptual de la arquitectura de tres niveles), ya que a nivel físico puede estar implementada mediante distintas estructuras de almacenamiento. La mayoría de los SMBD comerciales actuales están basados en el modelo relacional, que proporciona una mayor independencia de datos. Se dice es declarativo (se especifica qué datos se han de obtener).

**El modelo entidad-relación** se basa en 2 conceptos: entidad y relación. Para darnos una idea intuitiva, daremos las siguientes definiciones: Las entidades son objetos sobre los cuales queremos guardar información y la característica es que tengan existencia por sí mismos. Las relaciones son asociaciones entre entidades. Dentro del modelo entidad relación se identifican:

- Entidades.
- Atributos.
- Dominios.
- Claves.
- Interrelaciones.

La Figura 1.2 presenta los elementos del modelo entidad-relación. A continuación se define cada uno.

### Entidades

Se puede definir como entidad a cualquier objeto, real o abstracto, que existe en un contexto determinado o puede llegar a existir y del cual deseamos guardar información, por ejemplo: "PROFESOR", "CURSO", "ALUMNO" [6]. Las entidades las podemos clasificar en:

**Regulares:** aquellas que existen por sí mismas y que la existencia de un ejemplar en la entidad no depende de la existencia de otros ejemplares en otra entidad. Como lo muestra la Figura 1.2 A).

**Débiles:** son aquellas entidades en las que se hace necesaria la existencia de ejemplares de otras entidades distintas para que puedan existir ejemplares en esta entidad. Como lo muestra la Figura 1.2 B).

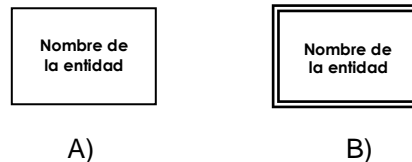


Figura 1.2 A) Entidad regular, B) Entidad débil.

### Atributos

Las entidades se componen de atributos que son las propiedades o características que tienen las entidades. Cada ejemplar de una misma entidad posee los mismos atributos, tanto en nombre como en número, diferenciándose cada uno de los ejemplares por los valores que toman dichos atributos.

Existen cuatro tipos de atributos:

**Obligatorios:** aquellos que deben tomar un valor y no se permite ningún ejemplar que no tenga un valor determinado en el atributo.

**Opcional:** aquellos atributos que pueden tener valores o no tenerlo.

**Monoevaluado:** aquel atributo que sólo puede tener un único valor.

**Multievaluado:** aquellos atributos que pueden tener varios valores.

Existen atributos, llamados derivados, cuyo valor se obtiene a partir de los valores de otros atributos. En determinadas ocasiones es necesaria la descomposición de un atributo para definirlos en más de un dominio.

### Dominios

Se define dominio como un conjunto de valores que puede tomar un determinado atributo dentro de una entidad. De forma casi inherente al término dominio aparece el concepto restricción para un atributo. Cada atributo puede adoptar una serie de valores de un dominio restringiendo determinados valores.

### Claves

El modelo entidad-relación exige que cada entidad tenga un identificador, se trata de un atributo o conjunto de atributos que identifican de forma única a cada uno de los ejemplares de la entidad. De tal forma que ningún par de ejemplares de la entidad puedan tener el mismo valor en ese identificador. Estos identificadores reciben el nombre de **identificador principal (IP) o clave primaria (PK -Primary Key-)**. Se puede dar el caso de existir algún

identificador más en la entidad, a estos identificadores se les denomina **identificadores candidatos (IC)**.

**Interrelaciones**

Se entiende por interrelación a la asociación, vinculación o correspondencia entre entidades. Al igual que las entidades, las interrelaciones se pueden clasificar en regulares y débiles, según estén asociando dos tipos de entidades regulares o una entidad débil con otra de cualquier tipo. Las interrelaciones débiles se subdividen en dos grupos:

**En existencia:** cuando los ejemplares de la entidad débil no pueden existir si desaparece el ejemplar de la entidad regular del cual dependen.

**En identificación:** cuando, además de ser una relación en existencia, los ejemplares de la entidad débil no se pueden identificar por sí mismos y exigen añadir el identificador principal de la entidad regular del cual dependen para ser identificados.

En cada interrelación se debe establecer el número máximo y mínimo de ejemplares de un tipo de entidad que pueden estar asociadas, mediante una determinada relación, con un ejemplar de otra entidad. Este valor máximo y mínimo se conoce como cardinalidad y, según corresponda, se representa de la siguiente forma: (0,n), (n,0), (1,n), (n,1), (0,1), (1,0), (0,0) ó (n,n). La cardinalidad se representa en la Figura 1.3:

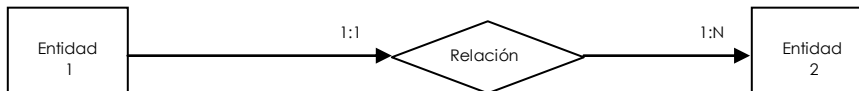


Figura 1.3 cardinalidad.

En el diagrama anterior la cardinalidad "Entidad 1" - "Entidad 2" es 1:1 porque al formularnos la pregunta ¿cuántas entidades 1 se pueden relacionar con una entidad 2? La respuesta es, uno como mínimo y uno como máximo, ya que una entidad 2 es realizada por una única entidad 1 y no cabe la posibilidad que la misma entidad 2 esté formulado por dos entidades 1 distinta. La cardinalidad "Entidad 2" - "Entidad 1" es 1:N por que al formularnos la pregunta ¿cuántas entidades 2 se pueden relacionar con una entidad 1? la respuesta es, como mínimo una entidad 2 pertenece a una entidad 1, pero varias entidad 2 pueden estar relacionados con la mismo entidad 1.

Existen ocasiones concretas en que las relaciones tienen atributos, es el caso de la Figura 1.4 en donde la entidad 1 recibe la entidad 2, y la interrelación posee los atributos.

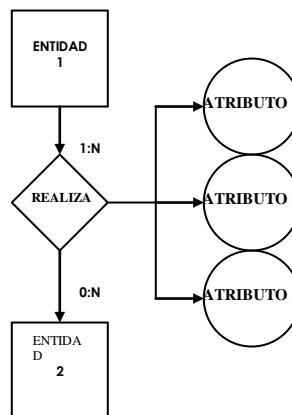


Figura 1.4 Relaciones con

A medida que se van estableciendo las interrelaciones hay que prestar especial atención a las interrelaciones cíclicas o redundantes, que son aquellas que su eliminación no

implica la pérdida de información. Como ejemplo la Figura 1.5 describe un modelo de entidad - relación:

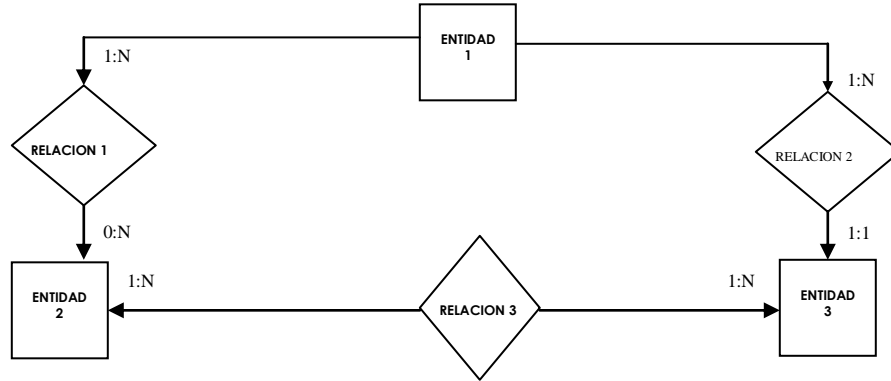


Figura 1.5 Modelo entidad-relación.

En determinadas ocasiones aparecen relaciones que asocian a más de dos entidades, se trata de las relaciones de grado superior. Un ejemplo de este tipo de relación es la Figura 1.6:

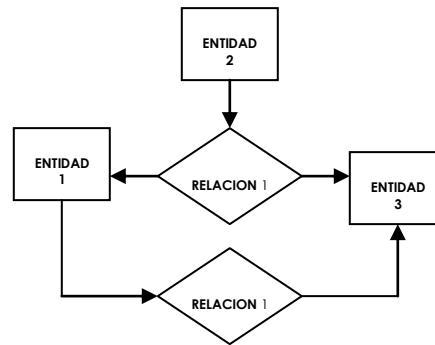


Figura 1.6 Relaciones de grado superior.

**Modelo relacional.**

En 1970, el modo en que se veían las bases de datos cambió por completo cuando E. F. Codd introdujo el modelo relacional. En aquellos momentos, el enfoque existente para la estructura de las bases de datos utilizaba punteros físicos (direcciones de disco) para relacionar registros de distintos archivos. Si, por ejemplo, se quería relacionar un registro **A** con un registro **B**, se debía añadir al registro **A** un campo conteniendo la dirección en disco del registro **B**. Este campo añadido, un puntero físico, siempre señalaría desde el registro **A** al registro **B** [6]. Codd demostró que estas bases de datos limitaban en gran medida los tipos de operaciones que los usuarios podían realizar sobre los datos. Además, estas bases de datos eran muy vulnerables a cambios en el entorno físico. Si se añadían los controladores de un nuevo disco al sistema y los datos se movían de una localización física a otra, se requería una conversión de los ficheros de datos [5]. Estos sistemas se basaban en el modelo de red y el modelo jerárquico, los dos modelos lógicos que constituyeron la primera generación de los SMBD.

El modelo relacional representa la segunda generación de los SMBD. En él, todos los datos están estructurados a nivel lógico como tablas formadas por filas y columnas, aunque a nivel físico pueden tener una estructura completamente distinta. Un punto fuerte del modelo relacional es la sencillez de su estructura lógica. Pero detrás de esa simple estructura hay un fundamento teórico importante del que carecen los SMBD de la primera generación, lo que constituye otro punto a su favor.

En los últimos años, se han propuesto algunas extensiones al modelo relacional para capturar mejor el significado de los datos, para disponer de los conceptos de la orientación a objetos y para disponer de capacidad deductiva. El modelo relacional, como todo modelo de datos, tiene que ver con tres aspectos de los datos:

- Estructura de datos.
- Integridad de datos.
- Manejo de datos.

El modelo relacional se basa en dos ramas de las matemáticas: la teoría de conjuntos y la lógica de predicados de primer orden. El hecho de que el modelo relacional esté basado en la teoría de las matemáticas es lo que lo hace tan seguro y robusto. Al mismo tiempo, estas ramas de las matemáticas proporcionan los elementos básicos necesarios para crear una base de datos relacional con una buena estructura, y proporcionan las líneas que se utilizan para formular buenas metodologías de diseño.

### 1.7 Progress

Progress es un proveedor líder de productos de software para desarrollar, implementar y administrar aplicaciones empresariales entrando al Internet. Ofrece servidores de mensajería de nivel superior, servidores de aplicación, bases de datos y productos para el desarrollo y administración de aplicaciones.

El lenguaje Progress maneja una base de datos relacional. Comenzó en 1985 100% carácter para usarse sobre plataforma Unix o Dos. Comenzó con la versión 6. El lenguaje que usa es el 4GL de Progress, un lenguaje de cuarta generación, enfocado a un lenguaje común con ayudas para generar las bases de datos. Considera archivos incluidos para insertar código.

Después de la llegada de Windows, nace la versión 7 character y la widget, se puede instalar en algunas redes y a parte de los archivos incluidos nacieron los procedimientos internos, es decir, las subrutinas. Dentro del widget (button, radio, query, select, browse). Agrego, además, las directivas del PRE-procesador y los templates.

Ya incluyendo todo esto surgen los llamados SMART-OBJECT, que consisten en la generación de librerías que manejan objetos inteligentes (herencia, polimorfismo, etc.). Se genera un solo SMART que se llama SDO, es un objeto que contiene datos (SDO contiene tratamiento visual, VIEWER y browse).

- La versión 9 es multivolumen y compile con ORACLE, llego a México en 1992.
- Progress se instala desde el registry. Progress.exe, ambiente carácter.
- Prowin32.exe ejecutable de Progress pero para Windows.
- Progress acepta single user y multiuser.
- Single user: Base de datos local, es un ambiente cliente servidor, no es necesario atravesar un server. En este estado nadie mas tiene acceso a la base de datos.

Progress Enterprise RDBMS esta diseñado para grandes ambientes empresariales en el procesamiento de transacciones a través de las aplicaciones mas demandantes de hoy día, basadas en procesamiento de transacciones en línea (OLTP) utilizando SQL y Progress 4GL. Cimentando en una arquitectura flexible, de multienlaces y multiservidor. Progress Enterprise RDBMS es una base de datos empresarial a gran escala, poderosa y abierta, que puede ejecutarse a través de múltiples plataformas de hardware y redes. Progress Enterprise RDBMS incluye toda la funcionalidad necesaria para satisfacer los requerimientos OLTP más demandantes. Estas capacidades incluyen bloqueo a nivel de registro, recuperación "roll-back" y "roll-forward", recuperación "point-in-time", administración de la base de datos distribuida "two-phase commint", un juego completo de utilidades en línea y soporte

completo para ANSI estándar SQL-92. Una combinación de poder, flexibilidad y fácil operación hacen de Progress Enterprise RDBMS un sistema ideal para un amplio rango de aplicaciones comerciales y de procesamiento de datos. Sofisticadas capacidades auto ajustables e interfaces graficas simples para la administración del sistema, hacen de Progress Enterprise RDMBS un sistema mas sencillo de instalar, afinar y administrar que otros productos. Con bajos costos de administración, un bajo costo inicial por licencia y costos mínimos por actualización, progress Enterprise RDBMS proporciona una ventaja significativa en el costo de propiedad -aproximadamente de 4 a 1- sobre los productos de bases de datos de la competencia.

### 1.8 Documentación de la etapa de ingeniería de sistemas

Después de documentar toda la teoría que necesitaremos en este proyecto, finalizare este desarrollando los puntos del documento de la ingeniería de software que es necesario cubrirlos en este capítulo, siguiendo nuestra metodología de la investigación.

Considerando el ciclo de vida elegido "Modelo de cascada", en este capítulo se documenta la primera etapa, "Ingeniería de sistemas".

#### Definición de ingeniería de software.

Es una disciplina que integra métodos, herramientas y procedimientos para el desarrollo de software de computadora bajo un enfoque de calidad.

#### 1.8.1 Tecnología Estratificada

El plan de desarrollo del software se seguirá de acuerdo al esquema mostrado en la Figura 1.7.

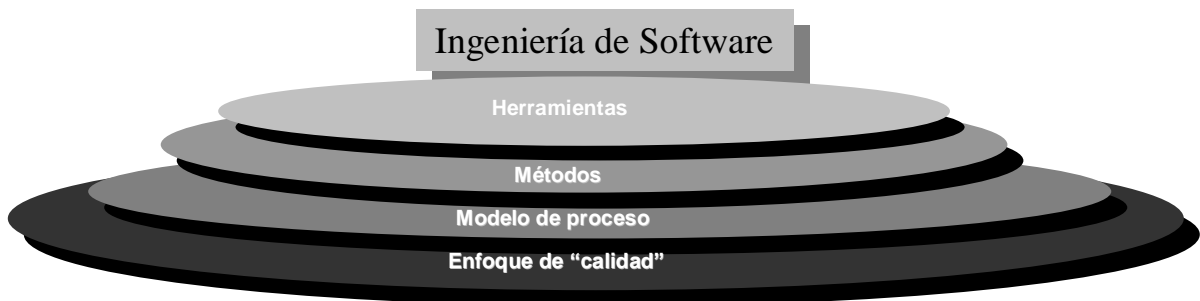


Figura 1.7 Plan de desarrollo del software.

#### Proceso

El proceso de desarrollo de software contiene tres fases genéricas, independientemente del ciclo de vida elegido. Las tres fases, definición, desarrollo, y mantenimiento, se encuentran en todos los desarrollos de software independientemente del área de aplicación, del tamaño del proyecto o de la complejidad [7].

#### El proceso como solución del problema.

##### Fase de definición del Software.

La fase de definición se centra sobre el qué. Esto es, durante la definición, el que desarrolla el software intenta identificar qué información debe ser proporcionada, qué función y rendimiento se desea, qué interfaces deben establecerse, qué restricciones de diseño existen y qué criterios de validación se necesitan para definir un sistema correcto. Aunque los métodos aplicados durante la fase de definición variarán dependiendo del ciclo de vida aplicado, de alguna forma se producirán tres pasos específicos: [7].

- Análisis de Requerimientos.** El ámbito establecido para el software proporciona la dirección a seguir, pero antes de comenzar a trabajar es necesario disponer de una información mas detallada del ámbito de información y de función del software.
- Análisis del sistema.** Define el papel de cada elemento del sistema informático, asignando finalmente al software el papel que va a desempeñar.
- Planificación del proyecto de software.** Una vez establecido el ámbito del software, se analizan los riesgos, se asigna los recursos, se estiman los costos, se definen las tareas y se planifica el trabajo.

#### **Fase de desarrollo del Software.**

La fase de desarrollo se centra en el cómo. Esto es, durante esta fase. El que desarrolla el software intenta descubrir cómo han de diseñarse las estructuras de datos y la arquitectura del software, cómo han de implementarse los detalles procedimentales, cómo ha de traducirse el diseño a un lenguaje de programación y cómo ha de realizarse la prueba. Los métodos aplicados durante la fase de desarrollo varían, pero de alguna forma se aplicarán tres pasos concretos:

- Diseño de software.** El diseño traduce los requisitos de software a un conjunto de representaciones (algunas gráficas y otras tabulares o basadas en lenguajes) que describen las estructuras de bases de datos, la arquitectura, el procedimiento algorítmico y las características de la interfaz.
- Codificación.** Las representaciones del diseño deberán ser traducidas a un lenguaje artificial (un lenguaje de programación convencional o un lenguaje no procedimental T4G), dando como resultado unas instrucciones ejecutables en la computadora.
- Prueba del software.** Una vez que el software ha sido implementado en una forma ejecutable por la computadora, debe ser probado para descubrir los defectos que puedan existir, en la función, en la lógica y en la implementación.

#### **Fase de mantenimiento del Software.**

La fase de mantenimiento se centra en el cambio que va asociado a la corrección de errores, a las adaptaciones requeridas por la evolución del entorno del software y a las modificaciones debidas a los cambios de requisitos del usuario dirigidos a reforzar o ampliar el sistema. La fase de mantenimiento vuelve a aplicar las fases de definición y de desarrollo, pero en el contexto del software ya existente. Durante la fase de desarrollo se encuentran tres tipos de cambio:

- Corrección.** Incluso llevando a cabo las mejores actividades de garantía de calidad, es muy probable que el cliente descubra defectos en el software. El mantenimiento correctivo cambia el software para corregir los defectos.
- Adaptación.** Con el paso del tiempo es probable que cambie el entorno original (sistemas operativos, equipos periféricos, etc.) para los que se desarrollo el software. El mantenimiento adaptivo consiste en modificar el software para acomodarlo a los cambios de su entorno externo.
- Mejora.** Conforme utilice el software, el usuario puede descubrir funciones adicionales que podrían interesar que estuvieran incorporadas en el software. El mantenimiento perfectivo amplía el software mas allá de sus requisitos funcionales originales.

#### **Herramientas.**

Las herramientas son el soporte automático o semiautomático de aplicaciones de los métodos. Existen herramientas para cada uno de los métodos anteriores. Cuando la información creada por una herramienta puede ser utilizada por otra se establece un sistema para el desarrollo del software llamado ingeniería del software asistida por ordenador (CASE: Computer Aided Software Engineering). Cada método tiene sus propias herramientas [7].

**Consideraciones de Software y Hardware mínimas.**

Hardware	Software
Procesador Intel Pentium IV a 1 GHZ 512 MB en RAM 40 Gb en D.D.	Office 2007 Internet Explore Sistema Operativo Windows XP Progress Appbuilder 9.1B MySql PHP

**Métodos**

Los métodos nos enseñan como construir técnicamente el software. Estos métodos abarcan, planificación y estimación, análisis de requisitos del sistema y del software, diseño de estructura de datos, diseño de programa y procedimientos algorítmicos, codificación, prueba y mantenimiento.

**Enfoque de calidad**

Tamaño, tiempo de entrega, presupuestos y costos, dominio de la aplicación, tecnología a ser implantada, restricciones del sistema, requerimientos del usuario, recursos disponibles.

|

## Capítulo II Análisis del sistema

### 2.1 Introducción

Después de la ingeniería de sistemas, se inicia, la primera etapa del desarrollo del sistema que consiste en elaborar el modelo de análisis del sistema.

#### 2.1.1 Planificación del proyecto de software.

Para llevar el control de calidad de nuestro sistema se investigo acerca de los diferentes formatos propuestos por estándares de calidad y otra propuesta de esquema, plantilla "Plan del proyecto" [7]. De dichas propuestas la que sé eligió fue la plantilla. Por consiguiente, para el documento del plan del proyecto de software, seguiremos los puntos de esta plantilla.

### 2.2 Plan de proyecto

La administración efectiva de un proyecto de software depende de la planeación del progreso del proyecto. Un plan preparado al inicio de un proyecto, debe utilizarse como un conductor para el proyecto. Este debe ser el mejor posible de acuerdo con la información disponible. Está evolucionará conforme el proyecto progrese y la información disponible será mejor. Se introduce el tema y se describen tres actividades importantes de administración; planeación, calendarización de proyectos y administración de riesgos.

#### 2.2.1 Descripción del proyecto

Sistema de base de datos para el manejo y control de laboratorios de química.

#### 2.2.2 Documentos relacionados

El documento de ingeniería de software y el documento de análisis de requerimientos.

#### 2.2.3 Recursos necesarios

Hardware y software:  
 Un equipo portátil  
 Un cañón para presentaciones  
 Herramienta de Trabajo

En la Tabla 2.1 se presentan los recursos necesarios mínimos para la realización del proyecto.

Hardware	Software
Procesador Intel Pentium IV a 1 GHZ 512 MB en RAM 40 Gb en D.D.	Office 2007 Internet Explore Sistema Operativo Windows XP Progress Appbuilder 9.1B MySql PHP

Tabla 2.1 Recursos necesarios del hardware y el software.

#### 2.2.4 Análisis de riesgos

- Uno de los riesgos del sistema es terminar fuera de tiempo el sistema.
- Que el cliente proporcione nuevos requerimientos.

### 2.3 Análisis del sistema.

Después de la planificación del proyecto, se inicia, la primera etapa del desarrollo del sistema que consiste en elaborar el modelo de análisis del sistema.

El objetivo de esta etapa es determinar el contexto en el cual se va a crear la aplicación y derivar de allí los requerimientos que deberá atender la solución interactiva, como complemento a otras soluciones basadas en uso de otros medios (personales, impresos, audio-visuales), teniendo claro el rol de cada uno de los medios educativos seleccionados y la viabilidad de usarlos. El análisis es el estudio de un problema antes de realizar alguna acción. Partiremos de algunas definiciones y posteriormente veremos cómo se aplican los principios de la teoría de sistemas al caso concreto de sistemas de información [5], lo cual nos permitirá facilitar el proceso de Modelado de sistemas.

#### Definiciones de sistema de información.

Definimos a un sistema como un conjunto de dispositivos o de objetos relacionados de manera que forman una unidad o todo un organismo. Un sistema basado en computadora es un conjunto o arreglo de elementos que están organizados para realizar un objetivo predefinido procesando información. Un sistema abierto interactúa (se comunica) con otros sistemas que conforman su entorno o medio ambiente. Esta interacción con el medio se da recibiendo información (entrada) y proporciona información (salida), esto se muestra en la Figura 2.1

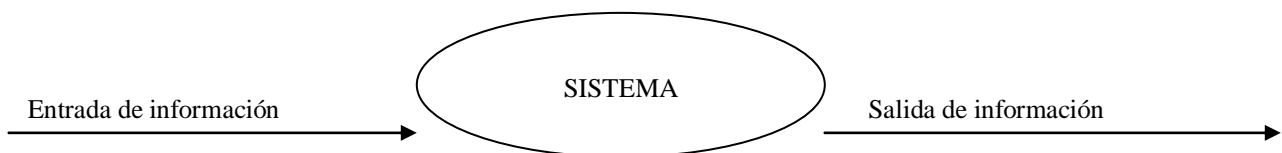


Figura 2.1 sistema abierto.

Se diseña con tal de satisfacer las necesidades de información de una organización (se entiende por organización empresas o instituciones públicas o privadas) y, además, está inmerso en ella. El sistema de información debe extraer los datos del entorno (la propia organización) o externas y los resultados deben ser la información que la organización necesita para la gestión y toma de decisiones. Se compone de diferentes partes u órganos:

- *Datos*
- *Hardware*
- *Software*
- *Sistema operativo*
- *Programas específicos*
- *Facturación*
- *la base de datos*
- *Administrador de datos*
- *Usuarios*
- *Sistema gestor de bases de datos*

#### 2.3.1 Principios del análisis

El análisis de sistemas debe de cumplir con los siguientes principios para realizar un buen análisis:

- Se debe entender y representar adecuadamente el dominio de la información del problema.
- Se deben desarrollar modelos que representen la información, la función y el comportamiento del sistema.
- Los modelos se deben dividir de manera que describan el detalle del sistema de manera jerárquica.
- El proceso del análisis se debe mover de información esencial a detalles de implementación.

También, se deben contemplar algunos inconvenientes en el análisis de un sistema como:

- Comunicación entre individuos
- La naturaleza cambiante de los requerimientos del sistema
- La carencia de herramientas para llevar a cabo el análisis
- Problemas de localización del trabajo en la organización
- Políticas no escritas en el trabajo
- Dificultades al interpretar documentos escritos
- Relaciones entre el equipo del análisis

La Figura 2.2 muestra la relación que debe tener el analista para obtener el análisis del sistema.

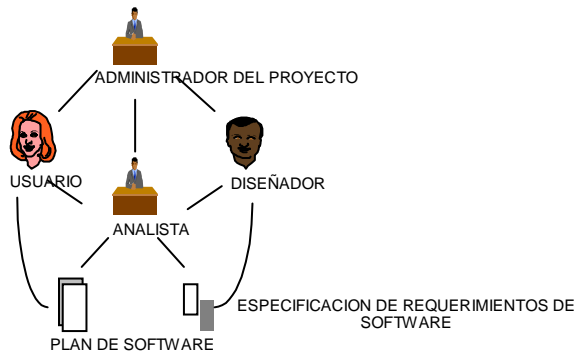


Figura 2.2 Comunicación entre individuos.

### 2.3.2 Modelo de análisis de un sistema

Los modelos de análisis se crean para:

- Facilitar la comprensión del sistema.
- Definir los procesos a cualquier nivel de detalle.
- Representar el comportamiento de los procesos.
- Definir las entradas externas e internas del sistema, y sus salidas.

El modelo de análisis de la metodología de análisis y diseño estructurado comprende tres puntos importantes. A continuación se describen cada uno de ellos:

**Modelo de datos:** se compone de tres piezas de información interrelacionadas: El objeto de datos, los atributos que describen el objeto de datos y las relaciones que conectan las relaciones entre si.

El modelo de datos es descrito con los objetos de datos y expresado gráficamente mediante el diagrama Entidad-Relación. Para este modelo se hace uso de los diagramas Entidad-Relación (DER) los cuales permiten identificar los objetos y sus relaciones mediante una notación gráfica.

**Modelo funcional:** La información se transforma a medida que fluye por un sistema. Aquí se modela el flujo y el contenido de la información. El modelo funcional es descrito por la especificación de procesos y expresado gráficamente por el diagrama de flujo de datos. Para este modelo se emplea los diagramas de flujo de datos (DFD), los DFD's son una técnica en la que se representa el flujo de la información y las transformaciones que se aplicaran a los datos al moverse de la entrada a la salida del sistema de una forma grafica.

**Modelo de comportamiento:** Se representa la forma en que se comporta el sistema, es decir, como se efectúan en el tiempo los distintos procesos y como va cambiando el estado del sistema.

El modelo de comportamiento se describe por la especificación de control y expresado gráficamente por el diagrama de **transición de estados** (DTE). Un DTE muestra los estados y los sucesos que hacen que un sistema cambie de estado.

En el centro del modelo de análisis se encuentra el *diccionario de datos* que proporciona una definición y descripción de todos los objetos de datos tanto de entrada como de salida. Todas las definiciones de los elementos en el sistema (flujo de datos, procesos y almacenes de datos) están descritas en forma detallada en el diccionario de datos. Si algún miembro del equipo encargado del proyecto desea saber alguna definición del nombre de un dato o el contenido particular de un flujo de datos, esta información debe encontrarse disponible en el diccionario de datos. El español estructurado es una forma de representar métodos del prototipo de sistemas.

### Definición de prototipo.

El prototipo es un sistema que funciona, no solo una idea en el papel, desarrollado con la finalidad de probar ideas y suposiciones relacionadas con el nuevo sistema. Al igual que cualquier sistema basado en computadora, está constituido por software que acepta entradas, realiza cálculos, produce información ya sea impreso o presentada en una pantalla, o que lleva a cabo otras actividades significativas. Es la primera versión, o iteración, de un sistema de información. La Figura 2.3 muestra la estructura del modelo de análisis.

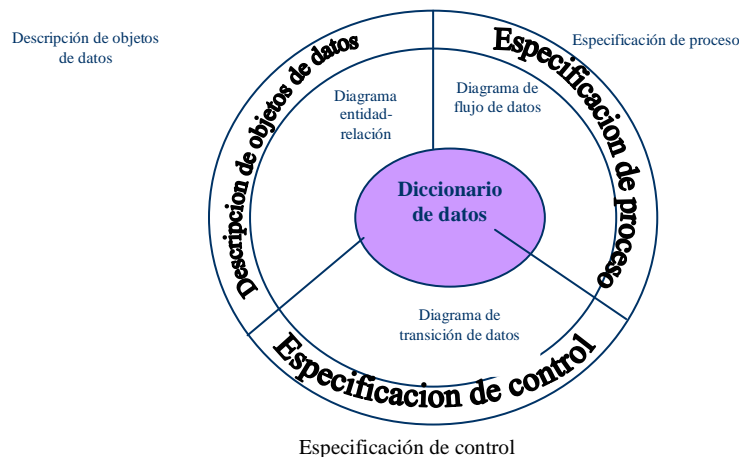


Figura 2.3 Estructura del modelo de análisis.

Los elementos esenciales son símbolos gráficos, diagramas de flujo de datos y diccionario centralizado de datos.

### 2.3.3 Herramientas del análisis estructurado

Son herramientas que ayudan a los especialistas en sistemas a documentar un sistema existente, ya sea éste manual o automatizado, y a determinar los requerimientos de una nueva aplicación. Estas herramientas incluyen:

- Diagrama entidad-relación (desarrollado en el documento de análisis de requisitos, en el modelo de datos).
- Diagramas de flujo de datos (DFD) (desarrollado en el documento de análisis de requisitos, en el modelo funcional).
- La especificación de procesos. (Desarrollado en el documento de análisis de requisitos, en el modelo funcional).
- El diagrama de transición de estados. (Desarrollado en el documento de análisis de requisitos, en el modelo de comportamiento).
- Español estructurado
- Diccionario de datos (DD)

## 2.4 Especificación de requisitos del software.

Para llevar el control de calidad de nuestro sistema se investigo acerca de los diferentes formatos propuestos por estándares de calidad, como el esquema para la especificación de requisitos (IEEE/ANSI 830, 1984) [5], El esquema para la especificación de requisitos (IEEE/ANSI 830, 1998) [5] y otra propuesta de esquema, plantilla “Especificación de requisitos de Software” [7]. De dichas propuestas la que sé eligió fue la tercera opción debido a que esta mas completa. Por consiguiente, para el documento de requerimientos del software, seguiremos los puntos de esta plantilla.

### 2.4.1 Introducción

Como síntesis de la etapa de análisis se deben formular los requerimientos que deberá atender el material interactivo que se desea obtener. Este documento contiene la declaración oficial de que es lo que requieren los desarrolladores del sistema. Incluye tanto los requerimientos del usuario para el sistema como una especificación detallada de los requerimientos del sistema. El documento de requerimientos de software satisface seis requisitos:

- Especificación de la conducta externa del sistema.
- Especifica los límites de la implementación.
- Fácil de cambiar.
- Sirve como una herramienta de referencia para mantenimiento.
- Recuerda el ciclo de vida del sistema, esto es, predice cambios.
- Proporciona respuestas características a un evento no esperado.

### Objetivos

Realizar un sistema que muestre un control de los diferentes elementos que manipula los laboratorios de la facultad de química de la Benemérita Universidad Autónoma de Puebla, con el fin de facilitar la obtención de las asignaturas que mayor uso hacen de ellos y se indique en que periodos han quedado fuera de los limites de existencia los elementos que manipulan, así como del personal encargado para su uso y administración.

### Alcance

Elaborar un sistema que lleve el control de los diversos elementos que manipula la facultad de Ingeniería química, el sistema deberá controlar la cristalería manejando sus datos como nombre, capacidad, existencia; instrumental manejando sus datos como: nombre, existencia, especificación; sustancias manejando sus datos como tipo de medida, cantidad, fecha de compra, código, nombre sistemático, nombre común, formula, ubicación, motivo, proveedor, fecha de adquisición, observaciones, lote del producto, fecha de caducidad, grado de reactividad, propiedades; otros materiales manejando sus datos como nombre, existencia, especificación; un registro de los diferentes tipos de sustancias utilizadas en cada asignatura manejando los datos como nombre, código, periodo, laboratorio, sustancias; los tipos de laboratorio en que se utilizan cada una de las sustancias químicas manejando sus datos como nombre, sustancia; información general personal encargado manejando sus datos como identificación, nombre, edad, dirección, lugar de nacimiento, ciudad, estado, teléfono, especialidad, grado, experiencia, departamento, fecha, comentarios; deudores manejando sus datos como identificación, nombre, edad, dirección, lugar de nacimiento, ciudad, estado, teléfono, especialidad, grado, experiencia, departamento, fecha, comentarios; Prácticas programadas manejando datos como no. De práctica, nombre de cristalería, cantidad/ubc,

nombre de instrumental, cantidad/ubi, nombre de sustancias, cantidad/ubs, materia, nrc, entrega; Prácticas de profesores manejando datos como código, profesor, número de práctica, fecha, laboratorio, no. De equipos, materia, nrc); E informes generales de cristalería, instrumental, sustancias, otros materiales, deudores.

El sistema deberá ser capaz de indicar al personal cual sustancia ha quedado fuera de los límites de existencia y también cual es la más utilizada. El sistema será controlado por cinco tipos de usuarios: director (usuario líder), almacenista, secretario administrativo, secretario académico y profesores los cuales tendrán su propio password de entrada al sistema. Estos requerimientos se deben controlar de la siguiente manera: el director podrá controlar todo el sistema; el almacenista controlará cristalería, instrumental, sustancias, deudores, prácticas programadas sólo podrá modificar y consultar, laboratorio, profesor, asignatura, prácticas de profesores sólo podrá consultar; el secretario académico manejará profesores, asignatura y prácticas programadas, prácticas de profesores y deudores sólo podrá consultar; el secretario administrativo controlará cristalería, instrumental, sustancias, laboratorios y sólo podrá consultar deudores e informes; los profesores podrán controlar prácticas de profesores y todas las áreas restantes del sistema solo consultando restringiéndolo al área de herramientas. El área de herramientas manejará, alta de usuario, eliminación de usuario, los cuales manejarán los datos siguientes, nombre de usuario, contraseña, confirmar contraseña, nivel de usuario.

### Contexto

El software se encuadra en la línea de productos de sistemas de bases de datos y por lo tanto es necesario discutir los aspectos estratégicos de esta línea.

El soporte computacional para el manejo y control de sustancias químicas en un laboratorio es una herramienta de primera importancia para mejorar el uso de ciertos compuestos químicos, así como, los tipos de laboratorios en que se utiliza cada uno, los profesores que controlan esas sustancias y que imparten las diferentes asignaturas, un control exhaustivo de la existencia de cada uno de los reactivos, pues debe avisar con cierto periodo de anticipación, qué material se ha quedado fuera de los límites de existencia, debe brindar las asignaturas que mayor uso hace del laboratorio y del material de trabajo. Así como el personal encargado del laboratorio. Actualmente no se cuenta con la necesaria infraestructura para controlarlos. Anteriormente se había considerado manejar las sustancias químicas en programas aplicativos no teniendo la suficiente capacidad de control. En la actualidad existe un control en papel para cada sustancia, este trabajo es sencillo cuando el número de sustancias es pequeño, pero cuando existe una gran demanda y el número de sustancias es grande, sale del control humano. De esta forma se ha sugerido que para una mejor administración de las sustancias se manejen bases de datos ya que una de sus ventajas o atributos es que maneja una gran cantidad de información permitiendo un mejor control en las existencias de las sustancias. Un sistema de base de datos es básicamente un sistema computarizado para llevar registros. Es posible considerar a la propia base de datos como una especie de armario electrónico para archivar, es decir, es un depósito o contenedor de una colección de archivos de datos computarizados.

### Restricciones principales

- El sistema será controlado por cinco tipos de usuarios, 1 usuario líder, que representaría el director de la facultad de química y 4 usuarios secundarios que tendrán restringidas las áreas, almacenista, secretario administrativo, secretario académico y profesores los cuales tendrán su propio password de entrada al sistema.
- Al sistema no se le podrá introducir información sino lo especifica el programa por sí mismo.

### 2.4.2 Escenarios de uso

Esta sección organiza la información de requisitos que se recoge del cliente. Un escenario es una interacción entre el sistema y los actores.

### Perfiles de usuario

Un actor es una agrupación uniforme de personas, sistemas o máquinas que interactúan con el sistema que estamos construyendo de la misma forma. Basándose en las necesidades anteriores los actores del sistema son: un usuario líder que es el director y cuatro usuarios secundarios que son almacenista, secretario académico, secretario administrativo, profesores.

Los actores se representan con dibujos simplificados de personas, llamados en inglés "stick man" (hombres de palo), como se muestra en la Figura 2.4.

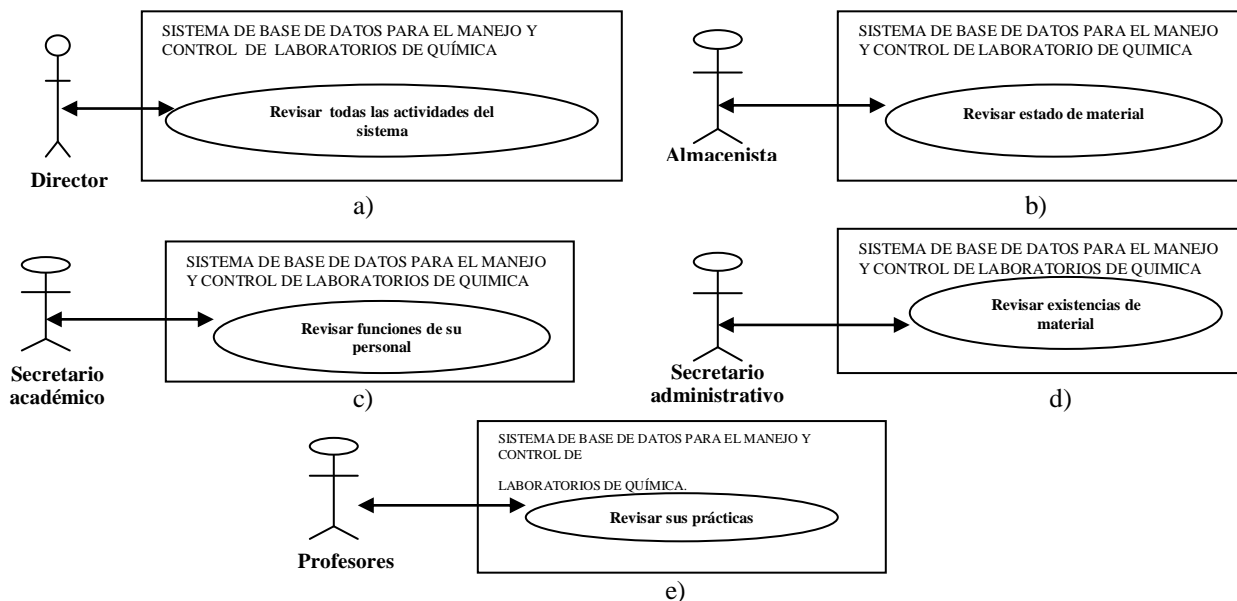


Figura. 2.4 Las figuras muestran cómo interactúan los actores con el sistema a) Actor director, b) Actor almacenista, c) Actor secretario académico, d) Actor secretario administrativo, e) Actor Profesores.

### Casos de uso

Un caso de uso es una secuencia de interacciones entre un sistema y alguien o algo que usa alguno de sus servicios. Un caso de uso es iniciado por un actor. A partir de ese momento, ese actor, junto con otros actores, intercambia datos o control con el sistema, participando de ese caso de uso. Las necesidades siguientes deben satisfacerse por el sistema:

- El director entra al sistema para verificar el estado de todas las actividades del sistema, como los laboratorios el saber cómo están trabajando, saber los datos de su personal, saber las asignaturas que llevarán laboratorio y que material necesitaran, quienes adeudan material, que prácticas están programadas en el presente cuatrimestre, que material necesitan los profesores para sus prácticas, el podrá rápidamente revisar sus existencias en cristalería, instrumental y sustancias revisando informes.
- El director da de alta a un nuevo usuario del sistema.
- El almacenista da de alta nuevo material de cristalería, instrumental y sustancias.
- El almacenista consulta que laboratorio necesita material.
- El almacenista consulta sus datos en el sistema.
- El almacenista consulta en el sistema que materias necesitan material para preparar dicho material.
- El almacenista da de alta, modifica datos, elimina o consulta deudores de material.
- El almacenista da de alta, modifica datos, elimina o consulta datos de prácticas programadas.

- El almacenista puede consultar los datos de las prácticas de los profesores para saber cual es el material que necesitan y poder prepararlo anticipadamente.
- El secretario académico puede dar de alta, modifica, elimina o consulta datos del personal.
- El secretario académico puede dar de alta, modifica, elimina o consulta datos de las asignaturas asignadas en un cuatrimestre.
- El secretario académico puede consultar a los deudores de material para poder buscar a dicha persona y retener papeles.
- El secretario académico puede consultar que prácticas programadas se han cumplido para saber si el personal esta trabajando.
- El secretario académico puede consultar que prácticas de profesores están completas.
- El secretario administrativo puede revisar el estado de cristalería, instrumental, sustancias, así como de sus existencias.
- El secretario administrativo puede dar de alta, modificar, eliminar, consultar material de los laboratorios.
- El secretario administrativo puede consultar quienes son deudores y saber cual es el material que falta.
- El secretario académico puede rápidamente consultar las existencias de material revisando los informes.
- Los profesores pueden dar de alta, modificar, eliminar, consultar sus prácticas al sistema.
- Los profesores podrán consultar todo el sistema a través de informes.

Basándose en estas necesidades, los casos de uso siguientes han sido identificados:

1. Mantenimiento de información de cristalería.
2. Mantenimiento de información de instrumental.
3. Mantenimiento de información de sustancias.
4. Mantenimiento de información de otros materiales.
5. Mantenimiento de información de laboratorios.
6. Mantenimiento de información de personal.
7. Mantenimiento de información de asignaturas.
8. Mantenimiento de información de deudores.
9. Mantenimiento de información de prácticas programadas
10. Mantenimiento de información de prácticas de profesores.
11. Proporcionar lista de datos en cristalería.
12. Proporcionar lista de datos en instrumental.
13. Proporcionar lista de datos en sustancias.
14. Proporcionar lista de datos en otros materiales.
15. Proporcionar lista de datos en laboratorios.
16. Proporcionar lista de datos en personal.
17. Proporcionar lista de datos en asignaturas.
18. Proporcionar lista de datos en deudores.
19. Proporcionar lista de datos en prácticas programadas.
20. Proporcionar lista de datos en prácticas de profesores.
21. Registro de usuarios.
22. Validar perfil.

De acuerdo a los casos de uso identificados se obtuvo el diagrama principal que se presenta en la Figura 2.5.

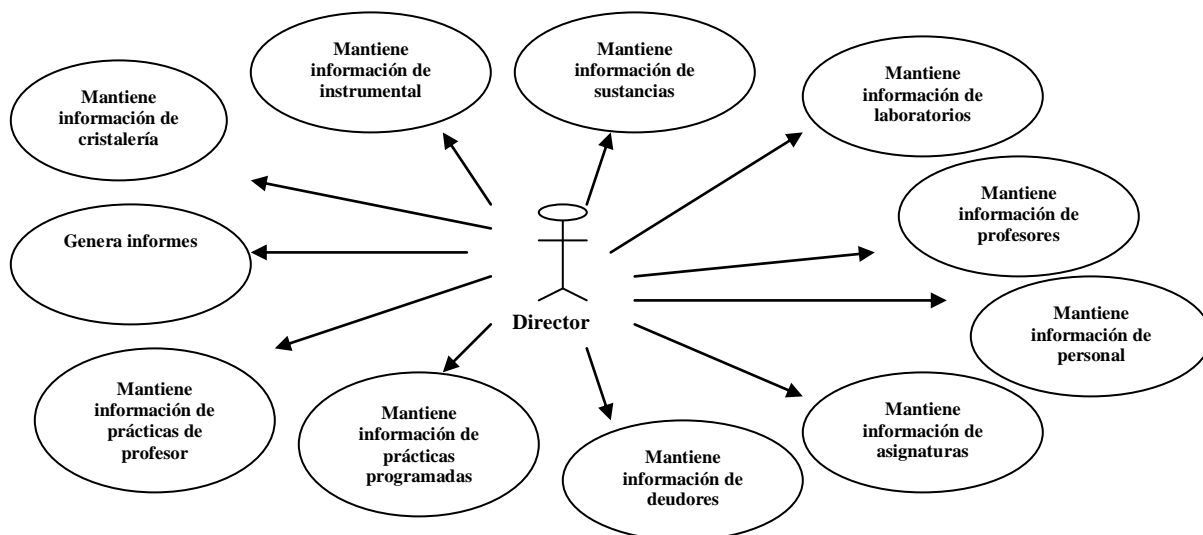


Figura 2.5 Diagramas de caso de uso principal para el sistema Manejo y Control de Laboratorios de Química (MACOLAQUI).

### Consideraciones especiales

El sistema será monousuario.

#### 2.4.3 Modelo de datos

El modelo de datos estudia los datos independientemente del procesamiento que los transforma. El modelo de datos se compone de tres piezas de información interrelacionadas: el objeto de datos, los atributos que describen el objeto de datos, y la relación que conecta objetos de datos entre si.

**Objetos de datos:** Es una representación de cualquier composición de información compuesta que debe comprender el software, por consiguiente, de acuerdo al problema los objetos de datos son los siguientes: cristalería, instrumental, sustancias, otros materiales, laboratorios, personal, asignaturas, deudores, prácticas programadas, prácticas de profesores y usuario.

**Atributos:** El conjunto de atributos apropiado para un objeto de datos dado se determina mediante el entendimiento del contexto del problema, por lo tanto los atributos son:

- Cristalería (idc, nombre, capacidad, existencia, especificación, ubc).
- Instrumental (idi, nombre, capacidad, existencia, especificación, ubi).
- Sustancias (ids, tipo de medida, cantidad, fecha de compra, nombre sistemático, nombre común, formula, ubicación, motivo, proveedor, fecha de adquisición, observaciones, lote del producto, fecha de caducidad, grado de reactividad, propiedades, ubs)
- Otros materiales(idom, nombre, capacidad, existencia, especificación, ubom)
- Laboratorios (idl, nombre, ids)
- Personal (idp, nombre, edad, dirección, lugar de nacimiento, ciudad, estado, teléfono, especialidad, grado, experiencia, departamento, fecha, comentarios)
- Asignaturas (nrc, materia, periodo, idl, ids, idc, idom, idi)
- Deudores (matricula, nombre, sección, fecha, asignación, idc, idi, idom, ids, observaciones)
- Prácticas programadas(idpprog, idl, idc, ubc, idi, ubi, ids, ubs, idom, ubom, materia, nrc, entrega)
- Prácticas de profesores (idpprof, idp, idpprog, fecha, idl, no. De equipos, materia, nrc).

- Informes (cristalería, instrumental, sustancias, otros materiales, deudores).
- Usuario (nombre de usuario, contraseña, confirmar contraseña, nivel de usuario).

**Relaciones:** Se define como un conjunto de parejas objeto-relación que define las relaciones relevantes. De acuerdo a lo anterior se obtuvieron las siguientes relaciones:

Un laboratorio debe tener sustancias.

Una asignatura puede tener laboratorio.

Una asignatura puede usar sustancias.

Una asignatura puede usar cristalería.

Una asignatura puede usar instrumental.

Una asignatura puede usar otros materiales.

Un deudor debe tener prestado sustancias.

Un deudor debe tener prestado cristalería.

Un deudor debe tener prestado instrumental.

Un deudor debe tener prestado otros materiales.

Las prácticas programadas deben usar laboratorio.

Las prácticas programadas pueden usar sustancias.

Las prácticas programadas pueden usar cristalería.

Las prácticas programadas pueden usar instrumental.

Las prácticas programadas pueden usar otros materiales.

Las prácticas programadas deben tener información general de la asignatura.

Las prácticas de profesores deben tener información general del profesor.

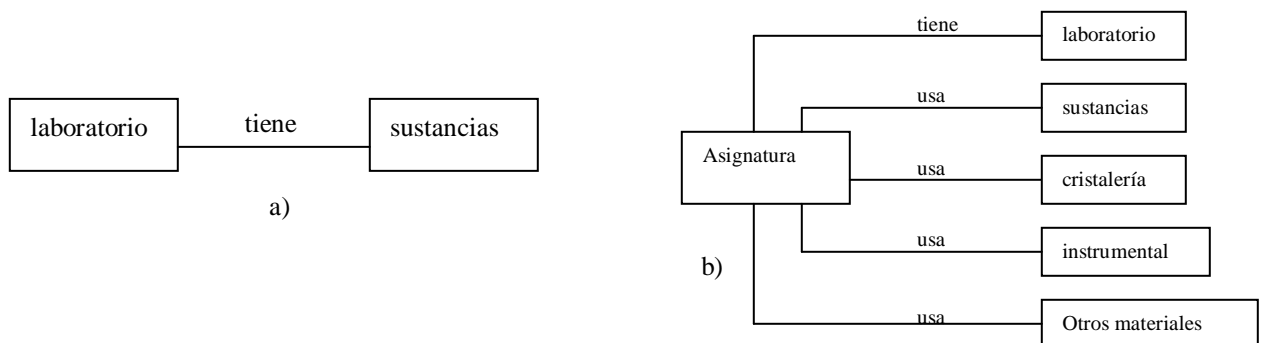
Las prácticas de profesores deben tener información de prácticas programadas.

Las prácticas de profesores deben usar laboratorio.

Las prácticas de profesores deben tener información general de la asignatura.

**Modelo de objetos**

Se describe el modelo de objetos esencial (el derivado directamente del dominio del problema o del modelo de datos. Para este modelo se hace uso de los diagramas Entidad-Relación (DER) los cuales permiten identificar los objetos y sus relaciones mediante una notación grafica). Se describen tanto objetos como relaciones, en este apartado daremos comienzo a la descripción gráfica de los objetos y las relaciones, desarrollándolo más a profundidad en el documento de diseño del software, de acuerdo a lo anterior tenemos la Figura 2.6



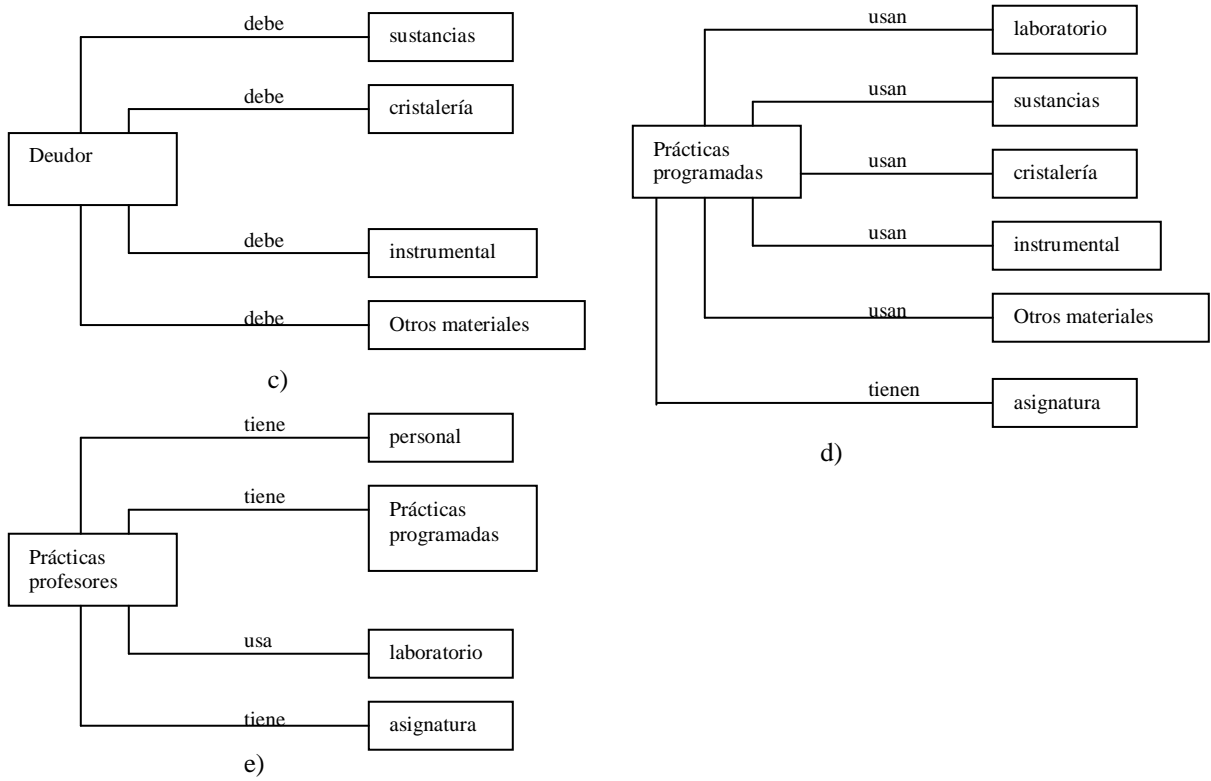


Figura 2.6 Relaciones entre objetos a) relaciones del objeto laboratorio, b) relaciones del objeto asignatura, c) relaciones del objeto deudor, d) relaciones del objeto prácticas programadas, e) relaciones del objeto prácticas profesores.

Por lo tanto el modelo de objetos queda de la siguiente manera, como lo muestra la Figura 2.7:

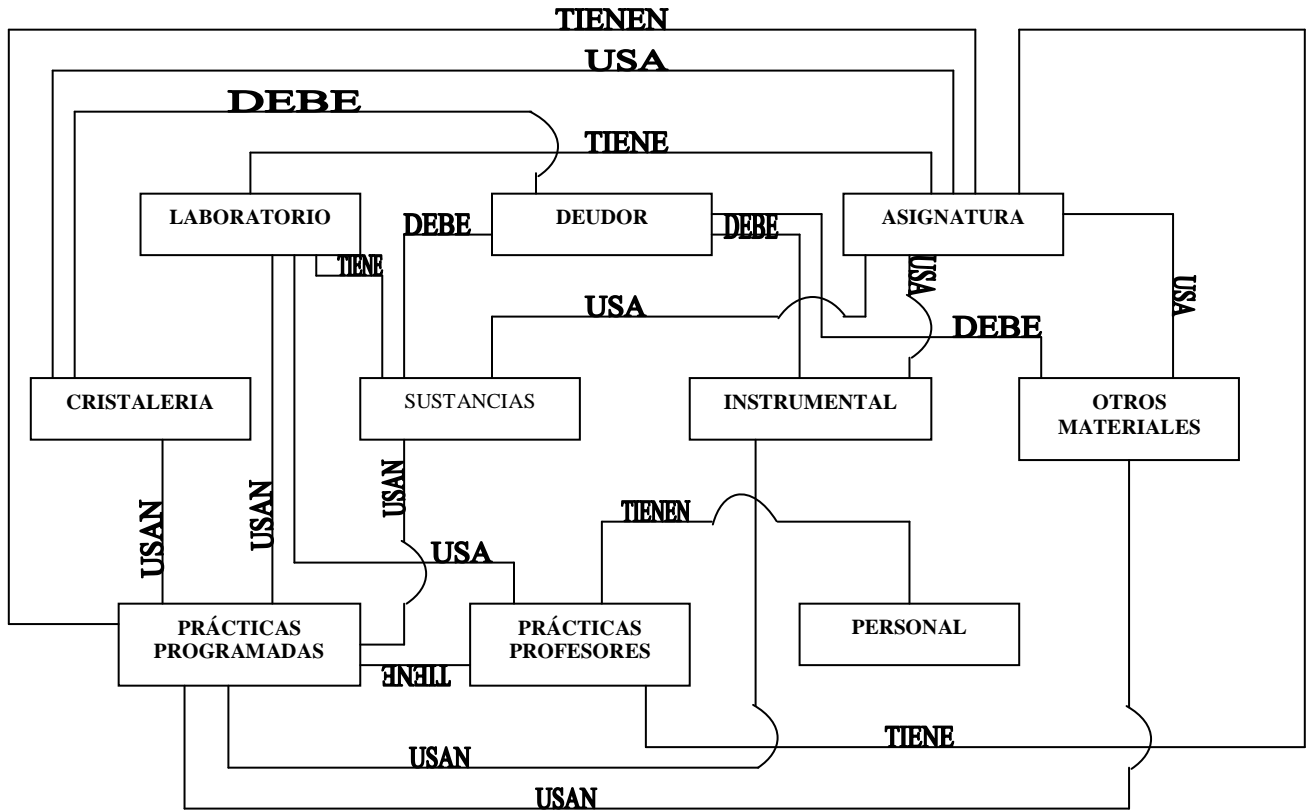


Figura 2.7 Modelo de objetos del sistema MACOLAQUI.

#### 2.4.4 Modelo funcional

Esta sección describe cada una de las funcionalidades principales del software.

- El sistema será controlado por cinco tipos de usuarios: director (usuario líder), almacenista, secretario administrativo, secretario académico y profesores los cuales tendrán su propia contraseña de entrada al sistema.
- El sistema deberá de controlar y manejar cristalería, instrumental, sustancias, otros materiales, laboratorio, personal, asignaturas, deudores, prácticas programadas, prácticas de profesores, informes, herramientas.

A continuación veremos el modelo funcional del sistema MACOLAQUI.

##### **Descripción de funciones 1**

Entrada al sistema.

##### **Especificación de proceso**

Al ingresar al sistema cada uno de los usuarios según su perfil de usuarios deberá ingresar al sistema mediante una contraseña. El sistema analizará los datos y tomara una decisión, si la contraseña es válida, ingresara al usuario al área de acuerdo al perfil detectado, de lo contrario el sistema avisara que no existe usuario y preguntará si desea continuar, si el usuario responde afirmativamente, el sistema volverá a pedir la contraseña de lo contrario terminará el sistema.

##### **Rendimiento**

Se estima que aproximadamente debe durar este proceso medio minuto.

##### **Restricciones de diseño**

Para verificar la validez de la contraseña se debe realizar una toma de decisión.

Una vez que el sistema acepto la contraseña, que el sistema tome la decisión de colocar al usuario en el área adecuada según el perfil. Si el sistema no acepto la contraseña, que el sistema tome la decisión de poder introducir de nuevo la contraseña o que finalicé el sistema.

##### **Descripción de funciones 2**

Control de datos por el usuario líder.

##### **Especificación de proceso**

Al ingresar al usuario al área de acuerdo a su perfil detectado el sistema le proporcionara los datos, si es el usuario líder podrá ingresar a todos las áreas como: cristalería, instrumental, sustancias, otros materiales, laboratorio, personal, asignaturas, deudores, prácticas programadas, prácticas de profesores, informes ó herramientas.

##### **Rendimiento**

Se estima que aproximadamente debe durar este proceso medio minuto.

##### **Restricciones de diseño**

Todas las áreas serán controladas por altas, bajas, modificaciones y consultas, excepto informes y herramientas.

##### **Descripción de funciones 3**

Control de datos por el usuario almacenista.

##### **Especificación de proceso**

Al ingresar al usuario al área de acuerdo a su perfil detectado el sistema le proporcionara los datos, si es el usuario almacenista controlara cristalería, instrumental, sustancias, deudores, prácticas programadas solo podrá modificar y consultar, laboratorio, profesor, asignatura, prácticas de profesores solo podrá consultar.

##### **Rendimiento**

Se estima que aproximadamente debe durar este proceso medio minuto.

##### **Restricciones de diseño**

Todas las áreas serán controladas por altas, bajas, modificaciones y consultas, excepto informes y herramientas.

#### **Descripción de funciones 4**

Control de datos por el usuario secretario académico.

##### **Especificación de proceso**

Al ingresar al usuario al área de acuerdo a su perfil detectado el sistema le proporcionara los datos, si es el usuario el secretario académico manejará profesores, asignatura y prácticas programadas, prácticas de profesores y deudores solo podrá consultar.

##### **Rendimiento**

Se estima que aproximadamente debe durar este proceso medio minuto.

##### **Restricciones de diseño**

Todas las áreas serán controladas por altas, bajas, modificaciones y consultas, excepto informes y herramientas.

#### **Descripción de funciones 5**

Control de datos por el usuario secretario administrativo.

##### **Especificación de proceso**

Al ingresar al usuario al área de acuerdo a su perfil detectado el sistema le proporcionara los datos, si es el usuario secretario administrativo controlara cristalería, instrumental, sustancias, laboratorios y solo podrá consultar deudores e informes.

##### **Rendimiento**

Se estima que aproximadamente debe durar este proceso medio minuto.

##### **Restricciones de diseño**

Todas las áreas serán controladas por altas, bajas, modificaciones y consultas, excepto informes y herramientas.

#### **Descripción de funciones 6**

Control de datos por el usuario profesores.

##### **Especificación de proceso**

Al ingresar al usuario al área de acuerdo a su perfil detectado el sistema le proporcionará los datos, si es el usuario profesores podrán controlar prácticas de profesores y todas las áreas restantes del sistema solo consultando, restringiéndolo al área de herramientas.

##### **Rendimiento**

Se estima que aproximadamente debe durar este proceso medio minuto.

##### **Restricciones de diseño**

Todas las áreas serán controladas por altas, bajas, modificaciones y consultas, excepto informes y herramientas.

#### **Descripción de funciones 7**

Manejo de datos de informes.

##### **Especificación de proceso**

Al ingresar al usuario al área de acuerdo a su perfil detectado, si el usuario revisa informes le proporcionará el sistema una lista de datos según el informe elegido.

##### **Rendimiento**

Se estima que aproximadamente debe durar este proceso medio minuto.

##### **Restricciones de diseño**

Solo proporcionara listado de datos.

#### **Descripción de funciones 8**

Manejo de herramientas.

##### **Especificación de proceso**

Al ingresar al usuario al área de acuerdo su perfil detectado, si el usuario revisa herramientas le proporcionara el sistema dos entradas: altas de usuario y eliminación de usuario. La primera opción, altas de usuario, maneja los datos siguientes: nombre de usuario, contraseña, confirmar contraseña y nivel de usuario, la segunda opción, eliminación de usuario, pide nombre del usuario a eliminar, si existe lo elimina de lo contrario manda un letrero avisando que no existe.

##### **Rendimiento**

Se estima que aproximadamente debe durar este proceso medio minuto.

**Restricciones de diseño**

El área de herramientas maneja, alta de usuario, eliminación de usuario. Si deseamos eliminar un usuario y no existe no lo eliminara el sistema.

**Descripción interfaces**

Esta sección detalla las interfaces del software con el "mundo exterior".

**Interfaces hardware.**

El sistema será monousuario.

**Interfaces software.**

Se necesita tener instalado Office 2000, Internet Explore, Sistema Operativo Windows 98, Manejador de bases de datos de progress.

**Interfaces humanas**

Las personas que tendrán acceso al sistema, serán:, director, secretario académico, secretario administrativo, profesores y almacenistas.

**Diagrama de flujo de datos (DFD)**

El DFD es también conocido como grafo de flujo de datos o como diagrama de burbujas.

Se puede usar el diagrama de flujo de datos para representar un sistema o software o cualquier nivel de abstracción. De hecho los DFD pueden ser divididos en niveles que representen un mayor flujo de información y un mayor detalle funcional. Por consiguiente, el DFD proporciona un mecanismo para el modelado funcional. Al hacer esto se cumple el segundo principio de análisis operacional.

**El DFD de nivel 0**

Un DFD de nivel 0 también es denominado modelo fundamental del sistema o modelo de contexto y representa al elemento de software completo como una sola burbuja con datos de entrada y de salida, respectivamente, como lo muestra la Figura 2.8.

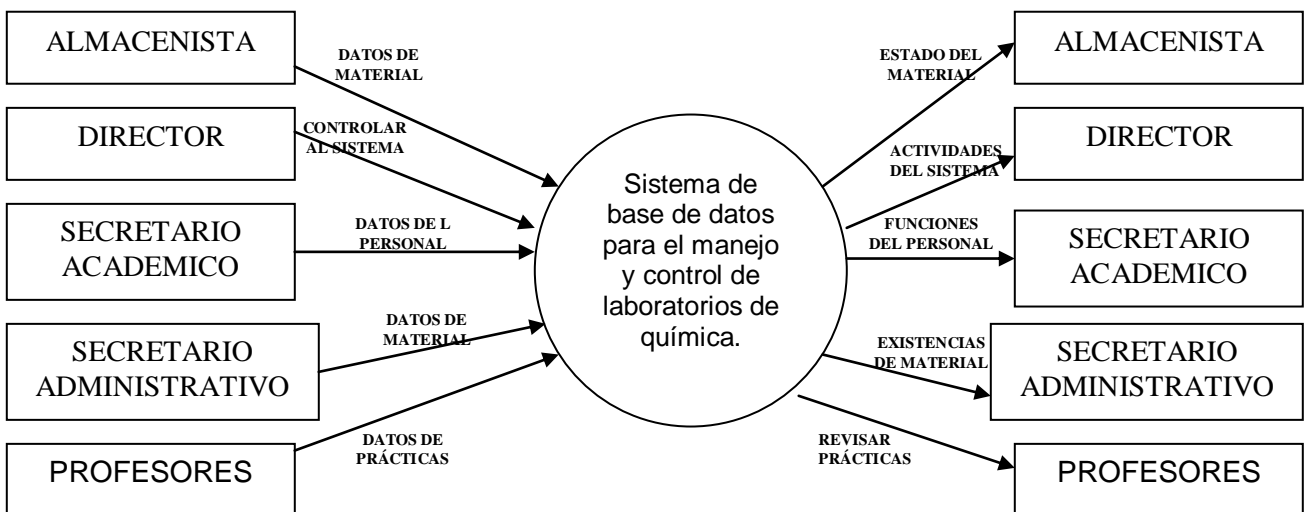


Figura 2.8 Diagrama de flujo de datos de nivel 0 del sistema MACOLAQUI.

**El DFD de nivel 1**

Un DFD de nivel 1 se utiliza para mostrar más detalle de un DFD de nivel 0, aparece representados procesos (burbujas) y caminos de flujo de información adicionales. Por ejemplo, puede contener cinco o seis burbujas con flechas interconectadas. Cada uno de los

procesos representados en el nivel 1 es una subfunción del sistema general en el modelo de contexto.

Para elaborar el DFD de nivel 1 de una forma sencilla consiste en realizar un <<análisis gramatical>>. Es decir, aislamos todos los nombre y todos los verbos de la narrativa presentada en la pagina 14. Para ilustrarlo, reproducimos de nuevo la narrativa de procesamiento, subrayando las primeras ocurrencias de los nombres y las primeras ocurrencias de los verbos.

#### ANÁLISIS GRAMATICAL:

Al *ingresar* al sistema cada uno de los usuarios según su perfil de usuarios *deberá* ingresar al sistema mediante una contraseña.

El sistema *analizará* los datos y *tomará* una decisión, si la contraseña es valida, ingresara al usuario al área de acuerdo al perfil *detectado*, de lo contrario el sistema *avisará* que no existe usuario y *preguntará* si desea continuar, si el usuario *responde* afirmativamente, el sistema *volverá a pedir* la contraseña de lo contrario *terminara* el sistema.

Al *ingresar* al usuario al área de acuerdo su perfil detectado el sistema le *proporcionará* los datos, si es el usuario líder (Director) podrá ingresar a todos las áreas como a cristería, instrumental, sustancias, otros materiales, laboratorio, personal, asignaturas, deudores, prácticas programadas, prácticas de profesores, informes, herramientas, así como configuración de registro de nuevos usuarios.

Al *ingresar* al usuario al área de acuerdo a su perfil detectado el sistema le *proporcionará* los datos, si es el usuario almacenista *controlara* cristería, instrumental, sustancias, otros materiales, deudores, prácticas programadas solo *podrá modificar y consultar*, laboratorio, personal, material, prácticas de profesores solo *podrá consultar*.

Al **ingresar** al usuario al área de acuerdo a su perfil detectado el sistema le *proporcionara* los datos, si es el usuario el secretario académico *manejara* personal, asignatura y prácticas programadas, prácticas de profesores y deudores solo *podrá consultar*.

Al *ingresar* al usuario al área de acuerdo a su perfil *detectado* el sistema le *proporcionara* los datos, si es el usuario secretario administrativo controlara cristería, instrumental, sustancias, laboratorios, otros materiales y solo *podrá consultar* deudores e informes.

Al *ingresar* al usuario al área de acuerdo a su perfil detectado el sistema le *proporcionara* los datos, si es el usuario profesores *podrán controlar* prácticas de profesores y todas las áreas restantes del sistema solo *consultando* restringiéndolo al área de herramientas.

Al *ingresar* al usuario al área de acuerdo a su perfil *detectado*, si el usuario *revisa* informes le *proporcionara* el sistema una lista de datos según el informe *elegido*.

Al *ingresar* al usuario al área de acuerdo a su perfil *detectado*, si el usuario *revisa* herramientas le *proporcionara* el sistema dos entradas altas de usuario el cual *maneja* los datos siguientes, nombre de usuario, contraseña, confirmar contraseña, nivel de usuario. Eliminación de usuario, el cual le *pide* nombre si *existe* lo *elimina* de lo contrario *manda* un letrero avisando que no *existe*.

De acuerdo con el análisis gramatical, comienza a aparecer un patron. Todos los verbos son procesos del “Sistema de base de datos para el manejo y control de sustancias químicas de un laboratorio”, es decir, deben de estar en última instancia representados, como burbujas en el consiguiente DFD. Todos los nombres son, o bien entidades externas (cuadrados), objetos de datos o de control (flechas) o almacenes de datos (líneas dobles). Usando esta información obtenemos el DFD de nivel 1 que se muestra a continuación en la Figura 2.9.

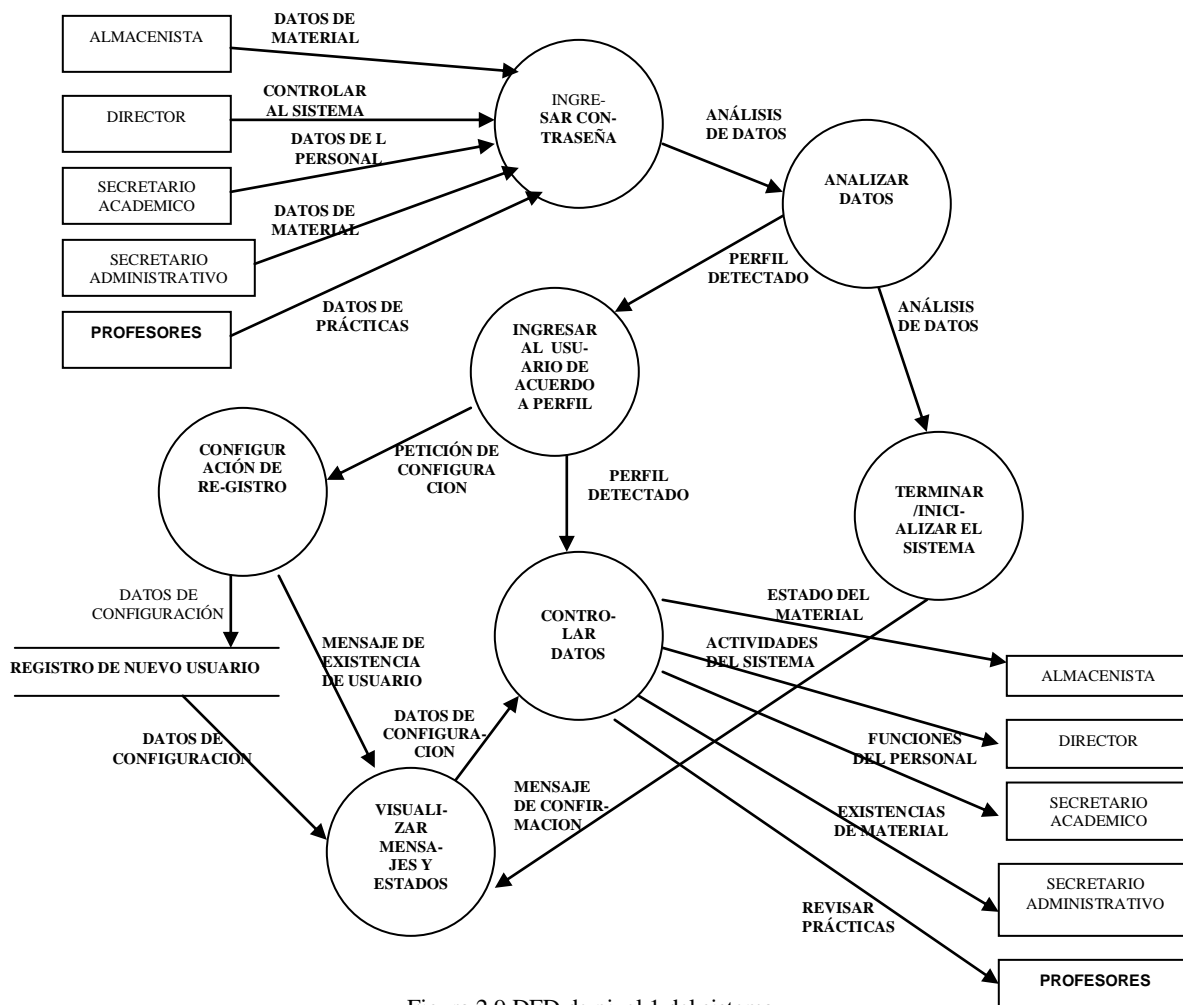


Figura 2.9 DFD de nivel 1 del sistema MACOLAQUI.

## 2.4.5 Modelo de comportamiento

El modelo de comportamiento es uno de los principios fundamentales de todos los métodos de análisis de requisitos.

### Comportamiento software

Se describen los eventos y estados principales.

El caso de uso representa una secuencia de actividades que incluye a actores y el sistema. En general, un evento ocurre cada vez que un sistema y un actor intercambian información. Un evento no es la información que se intercambia; es el hecho de que la información ha sido intercambiada. En el contexto de sistemas, deben considerarse dos caracterizaciones de estados:

- Estados de cada objeto cuando el sistema ejecuta su función.
- El estado del sistema observado desde el exterior cuando este ejecuta su función.

Para forzar la transición de un objeto de un estado activo a otro debe ocurrir un evento.

### Eventos

Una muestra de eventos para el caso de uso **registro de usuario** sería la siguiente, de acuerdo a una plantilla. El caso a estudiar es, **FLUJO DE EVENTOS PARA EL CASO DE USO REGISTRO DE USUARIO**

#### Precondiciones:

El caso de uso validar perfil se debe ejecutar antes de que el caso de uso registro de usuario comience.

#### Flujo principal:

Este caso de uso comienza cuando el director acceda al sistema y escribe su contraseña. El sistema verifica que la contraseña sea válida (E1) e indica al director que puede entrar a todas las áreas del sistema. El director selecciona herramientas (E2), el sistema que seleccione permite al director seleccionar la opción deseada: ALTA DE USUARIOS, BAJA DE USUARIOS.

- Si la opción seleccionada es ALTA DE USUARIOS, se realizara el subflujo S1: Dar de alta un nuevo usuario.
- Si la opción seleccionada es BAJA DE USUARIOS, se realizara el subflujo S2: Dar de baja un usuario existente.

El sistema analizara los datos y tomara una decisión, si la contraseña es válida, ingresara al usuario al área de acuerdo al perfil detectado, de lo contrario el sistema avisara que no existe usuario y preguntara si desea continuar, si el usuario responde afirmativamente, el sistema volverá a pedir la contraseña de lo contrario terminara el sistema.

#### Lista de subflujos eventos principales:

S1: Dar de alta un nuevo usuario.

El sistema despliega una pantalla de registro que contiene los campos nombre de usuario, contraseña, confirmar contraseña, nivel de usuario. El director introduce datos en los campos (E3). El sistema ingresa los datos del nuevo usuario (E4). Entonces el caso de uso comienza de nuevo.

S2: Dar de baja un usuario existente.

El sistema despliega una pantalla que contiene el campo nombre de usuario. El director introduce el nombre de usuario (E5). El sistema elimina los datos del usuario (E6). Entonces el caso de uso comienza de nuevo.

**Flujos de eventos alternativos o de excepción.**

E1: Se introduce la contraseña del director no valido. El director puede rescribir su contraseña o terminar el caso de uso.

E2: Se elige una opción diferente a herramientas. El director se puede desplazar por su menú de opciones y elegir la opción adecuada o terminar el caso de uso.

E3: Se introducen mal los datos de los campos. El director puede rescribir los datos válidos o terminar el caso de uso.

E4: El sistema no ingresa los datos. La información es guardada y el sistema incorpora los datos en un momento posterior. El caso de uso continúa.

E5: Se introduce un nombre de usuario no válido: El director puede rescribir el nombre válido o terminar el caso de uso.

E6: No puede eliminar al usuario. La información es guardada y el sistema elimina al usuario en un momento posterior. El caso de uso continúa.

**Estados**

Un estado es un modo observable del comportamiento. A continuación se mencionan las listas de estados principales: El objeto director tendrá los siguientes estados: ingresar, reingresar, seleccionar, introduce, elimina, desplaza, rescribe, comparando, esperando.

**Diagramas de transición de estados**

En un diagrama de transición estados los rectángulos representan estados del sistema y las flechas representan transiciones entre estados, es decir, los eventos. Cada flecha esta etiquetada con una expresión en forma de fracción. La parte superior indica el suceso que hace que se produzca la transición. La parte inferior indica la acción que se produce como consecuencia del suceso, la Figura 2.10 muestra el diagrama de transición de estados del sistema MACOLAQUI.

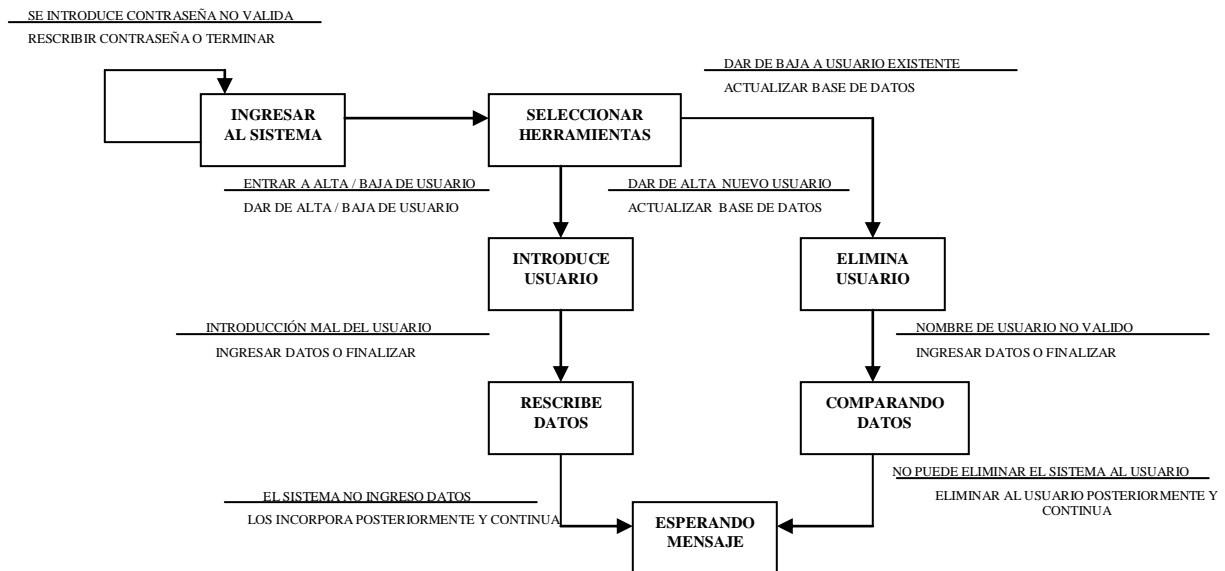


Fig. 2.10 Diagrama de transición de estados.

**2.4.6 Limitaciones y restricciones**

El sistema será controlado por perfiles de usuarios. Debido a que nuestro sistema esta enfocado exclusivamente a la facultad de química, no se podrá obtener información de otras facultades de la BUAP, otra información derivada u otros datos deseados por el usuario que no sean relacionados con el sistema. Una de las restricciones que pueden afectar a la especificación, diseño, implementación, o prueba del software, es que el sistema es monousuario y que el cliente quede insatisfecho y se tenga que implementar a multiusuario.

**2.4.7 Criterios de validación**

El sistema será validado y aceptado, cuando haya cumplido todos los requerimientos del usuario, cuando se haya instalado completamente. Cuando este en funcionamiento por lo menos un mes y no haya problemas en un 100%.

## Capítulo III Diseño del sistema.

### 3.1 Introducción

Después del análisis del sistema, se inicia, la segunda etapa del desarrollo del sistema que consiste en elaborar el modelo de diseño del sistema. Siguiendo el proceso como solución del problema, de la etapa de ingeniería, se desarrolla la segunda fase, la fase de desarrollo del software<sup>1</sup>, la cual contempla tres puntos fuertes: diseño del software, codificación y pruebas del software. En este capítulo se trata exclusivamente el punto de diseño del software. El diseño de software se divide en tres partes importantes: diseño conceptual, diseño lógico y diseño físico.

#### 3.1.1 El diseño de base de datos.

En la Figura 3.1 se puede apreciar el proceso de diseño de base de datos. Los requisitos de datos constituyen un componente de los requisitos de un producto y son una entrada al diseño conceptual.

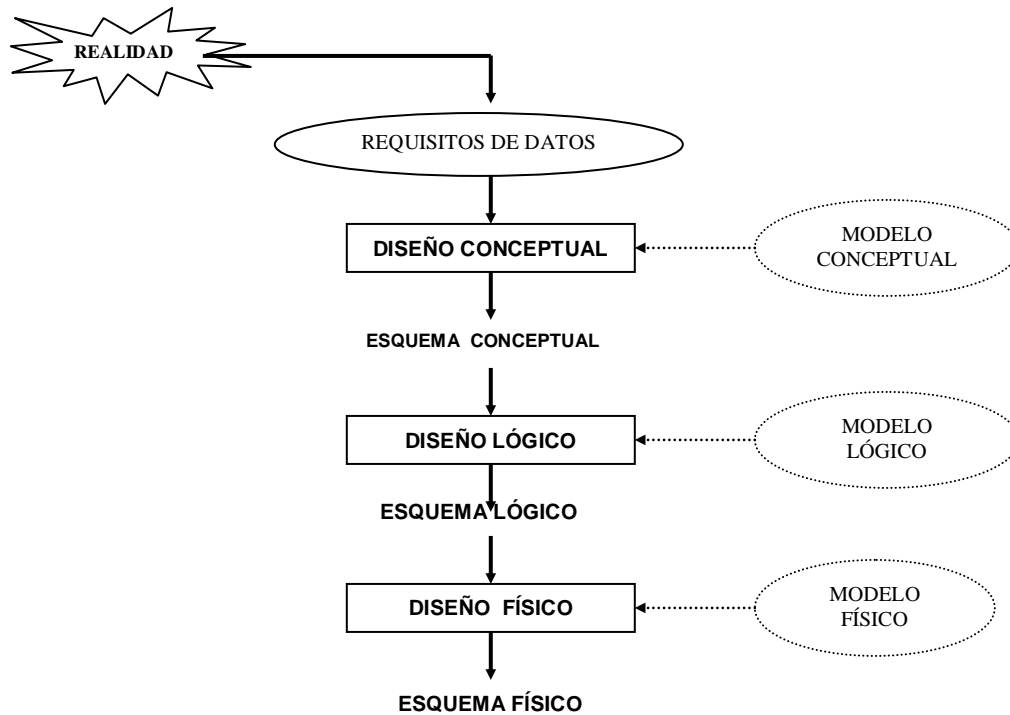


Figura. 3.1 Proceso de diseño de una base de datos.

<sup>1</sup> Tema visto en el capítulo 1 en el punto 1.6.2.1.1

### Diseño conceptual

Recibe como entrada la especificación de requisitos y su resultado es el esquema conceptual de la base de datos, que es una descripción de alto nivel de la estructura de la base de datos, independiente del software que se use para manipularla. Existen varios modelos conceptuales tales como el modelo entidad relación (MER), modelo orientado a objetos (MOO) y Formalismo individual.

### Diseño lógico

Recibe como entrada el esquema conceptual y da como resultado un esquema lógico, que es una descripción de la estructura de la base de datos que puede procesar el software DBMS. Dentro de los modelos lógicos están el modelo relacional, de redes y jerárquico.

### Diseño físico.

Recibe como entrada al esquema lógico y da como resultado un esquema físico, que es una descripción de la implementación de una base de datos en la memoria secundaria, describe las estructuras de almacenamiento y los métodos usados para obtener un acceso efectivo a los datos.

Algunos modelos físicos son: Modelo unificador, memoria de elementos.

## 3.2 Especificación del diseño del software.

Para llevar el control de calidad de nuestro sistema se investigo acerca de los diferentes formatos propuestos por estándares de calidad. En esta sección se incluye el desarrollo del diseño abarcando el modelo conceptual, el modelo lógico y el modelo físico del sistema.

### 3.2.1 Modelo conceptual

El primer paso en el diseño de una base de datos es la producción del esquema conceptual. En los esquemas conceptuales correspondientes a cada vista de usuario se les denomina *esquemas conceptuales locales*. Cada uno de estos esquemas se compone de entidades, relaciones, atributos, dominios de atributos e identificadores. El esquema conceptual también tendrá una documentación, que se irá produciendo durante su desarrollo.

#### Diseño conceptual

Básicamente tiene dos etapas:

- Etapa de análisis de requisitos: Desarrollado en el documento de análisis de requisitos.
- Etapa de conceptualización: Se desarrolla en esta etapa.

Las tareas a realizar en el diseño conceptual son las siguientes:

1. Identificar las entidades.
2. Identificar las relaciones.
3. Identificar los atributos y asociarlos a entidades y relaciones.
4. Determinar los dominios de los atributos.
5. Determinar los identificadores.
6. Determinar las jerarquías de generalización (si las hay).
7. Dibujar el diagrama entidad-relación.
8. Revisar el esquema conceptual local con el usuario.

#### Identificar las entidades.

Las entidades que se identificaron para la solución del sistema se muestran en la Tabla 3.1:

Tipo de entidad
Cristalería
Instrumental

Sustancia
Otros materiales
Laboratorio
Personal
Asignatura
Deudores
Practicas programadas
Practicas de profesores
usuario

Tabla 3.1 Identificación de entidades

Identificar las relaciones.

Las relaciones que se identificaron para la solución del sistema son:

Un laboratorio debe tener sustancias.

Una asignatura puede tener laboratorio.

Una asignatura puede usar sustancias.

Una asignatura puede usar cristalería.

Una asignatura puede usar instrumental.

Una asignatura puede usar otros materiales.

Un deudor debe tener prestado sustancias.

Un deudor debe tener prestado cristalería.

Un deudor debe tener prestado instrumental.

Un deudor debe tener prestado otros materiales.

Las prácticas programadas deben usar laboratorio.

Las prácticas programadas pueden usar sustancias.

Las prácticas programadas pueden usar cristalería.

Las prácticas programadas pueden usar instrumental.

Las prácticas programadas pueden usar otros materiales.

Las prácticas programadas deben tener información general de la asignatura.

Las prácticas de profesores deben tener información general del profesor.

Las prácticas de profesores deben tener información de prácticas programadas.

Las prácticas de profesores deben usar laboratorio.

Las prácticas de profesores deben tener información general de la asignatura.

Identificar los atributos y asociarlos a entidades y relaciones.

Los atributos y sus respectivas entidades que se identificaron para la solución del sistema se presentan en las Apéndice A:

Determinar los identificadores.

Los identificadores de las entidades para la solución del sistema se presentan en la tabla 3.23.

Entidad	identificador
Cristalería	Idc
Instrumental	Idi
Sustancia	Ids
Otros materiales	Idom
Laboratorio	Idl
Personal	Idp
Asignatura	Nrc
Deudores	Matricula
Prácticas programadas	Idpprog
Prácticas de profesores	Idpprof
usuario	Contraseña y cc

Tabla 3.23 identificadores de las entidades.



La metodología que se va a seguir para el diseño lógico en el modelo relacional consta de dos fases, cada una de ellas compuesta por varios pasos que se detallan a continuación:

1. Construir y validar los esquemas lógicos locales para cada vista de usuario.
  2. Convertir los esquemas conceptuales locales en esquemas lógicos locales.
  3. Derivar un conjunto de relaciones (tablas) para cada esquema lógico local.
  4. Validar cada esquema mediante la normalización.
  5. Validar cada esquema frente a las transacciones del usuario.
  6. Dibujar el diagrama entidad-relación.
  7. Definir las restricciones de integridad.
  8. Revisar cada esquema lógico local con el usuario correspondiente.
  9. Construir y validar el esquema lógico global.
  10. Mezclar los esquemas lógicos locales en un esquema lógico global.
  11. Validar el esquema lógico global.
  12. Estudiar el crecimiento futuro.
  13. Dibujar el diagrama entidad-relación final.
  14. Revisar el esquema lógico global con los usuarios
1. Construir y validar los esquemas lógicos locales para cada vista de usuario. A continuación se detallan las entidades y sus atributos correspondientes.
    - Cristalería (idc, nombre, capacidad, existencia, especificación, ubc).
    - Instrumental(idi, nombre, capacidad, existencia, especificación, ubi).
    - Sustancias (ids, tipo de medida, cantidad, fecha de compra, nombre sistemático, nombre común, fórmula, ubicación, motivo, proveedor, fecha de adquisición, observaciones, lote del producto, fecha de caducidad, grado de reactividad, propiedades, ubs)
    - Otros materiales(idom, nombre, capacidad, existencia, especificación, ubom)
    - Laboratorios (idl, nombre, ids)
    - Personal(idp, nombre, edad, dirección, lugar de nacimiento, ciudad, estado, teléfono, especialidad, grado, experiencia, departamento, fecha, comentarios)
    - Asignaturas (nrc, materia, periodo, idl, ids, idc, idom, idi)
    - Deudores (matricula, nombre, sección, fecha, asignación, idc, idi, idom, ids, observaciones)
    - Practicas programadas(idpprog, idl, idc, ubc, idi, ubi, ids, ubs, idom, ubom, materia, nrc, entrega)
    - Practicas de profesores(idpprof, idp, idpprog, fecha, idl, no. De equipos, materia, nrc)
    - Informes (cristalería, instrumental, sustancias, otros materiales, deudores)
    - Usuario (nombre de usuario, contraseña, confirmar contraseña, nivel de usuario)

2. Convertir los esquemas conceptuales locales en esquemas lógicos locales, como lo muestra la figura 3.4. a), b), c), d) y e).

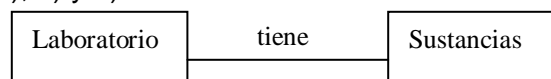


Fig. 3.4 a) Esquema conceptual local del objeto laboratorio.

### Esquema lógico local del objeto laboratorio.

Un esquema lógico local de un objeto nos permite identificar rápidamente los atributos que contienen las entidades.

laboratorios (idl, nombre, ids)

sustancias (ids, tipo de medida, cantidad, fecha de compra, nombre sistemático, nombre común, fórmula, ubicación, motivo, proveedor, fecha de adquisición, observaciones, lote del producto, fecha de caducidad, grado de reactividad, propiedades, ubs)

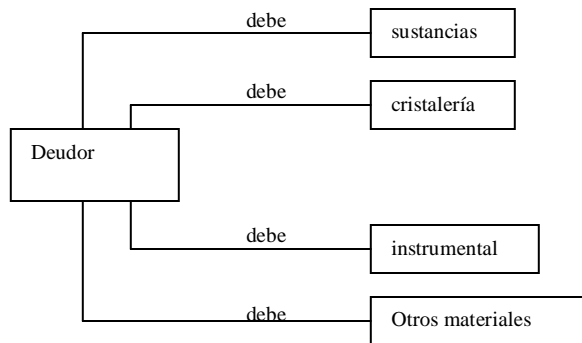


Figura 3.4 b) Esquema conceptual local del objeto deudor.

**Esquema lógico local del objeto deudor.**

Deudores (matricula, nombre, sección, fecha, asignación, idc, idi, idom, ids, observaciones)

Cristalería (idc, nombre, capacidad, existencia, especificación, ubc).

Instrumental (idi, nombre, capacidad, existencia, especificación, ubi).

sustancias (ids, tipo de medida, cantidad, fecha de compra, nombre sistemático, nombre común, fórmula, ubicación, motivo, proveedor, fecha de adquisición, observaciones, lote del producto, fecha de caducidad, grado de reactividad, propiedades, ubs)

otros materiales(idom, nombre, capacidad, existencia, especificación, ubom)

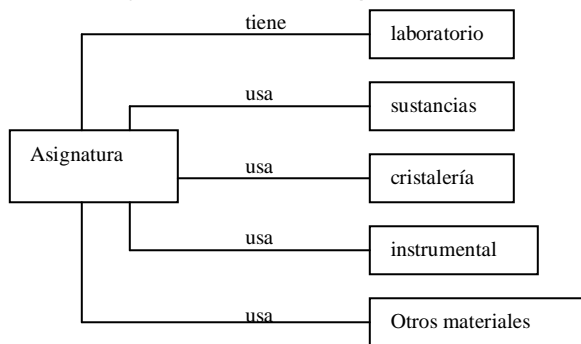


Figura. 3.4 c) Esquema conceptual local del objeto asignatura.

**Esquema lógico local del objeto asignatura.**

asignaturas (nrc, materia, periodo, idl, ids, idc, idom, idi)

Cristalería (idc, nombre, capacidad, existencia, especificación, ubc).

Instrumental(idi, nombre, capacidad, existencia, especificación, ubi).

sustancias (ids, tipo de medida, cantidad, fecha de compra, nombre sistemático, nombre común, fórmula, ubicación, motivo, proveedor, fecha de adquisición, observaciones, lote del producto, fecha de caducidad, grado de reactividad, propiedades, ubs)

otros materiales(idom, nombre, capacidad, existencia, especificación, ubom)

laboratorios (idl, nombre, ids)

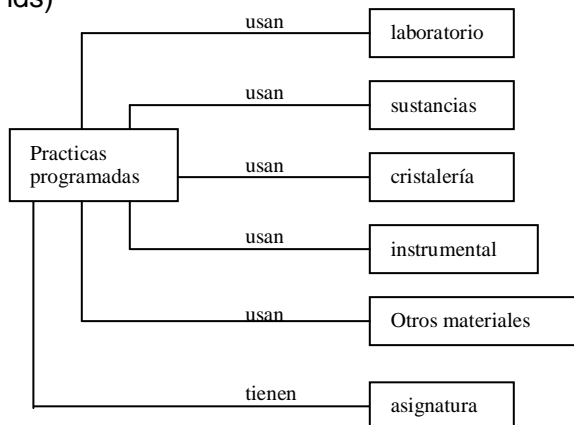


Figura 3.4 d) Esquema conceptual local del objeto prácticas programadas.

**Esquema lógico local del objeto practicas programadas.**

Prácticas programadas(idpprog, idl, idc, ubc, idi, ubi, ids, ubs, idom, ubom, materia, nrc, entrega)

Cristalería (idc, nombre, capacidad, existencia, especificación, ubc).

Instrumental(idi, nombre, capacidad, existencia, especificación, ubi).

sustancias (ids, tipo de medida, cantidad, fecha de compra, nombre sistemático, nombre común, fórmula, ubicación, motivo, proveedor, fecha de adquisición, observaciones, lote del producto, fecha de caducidad, grado de reactividad, propiedades, ubs)

otros materiales(idom, nombre, capacidad, existencia, especificación, ubom)

laboratorios (idl, nombre, ids)

asignaturas (nrc, materia, periodo, idl, ids, idc, idom, idi)

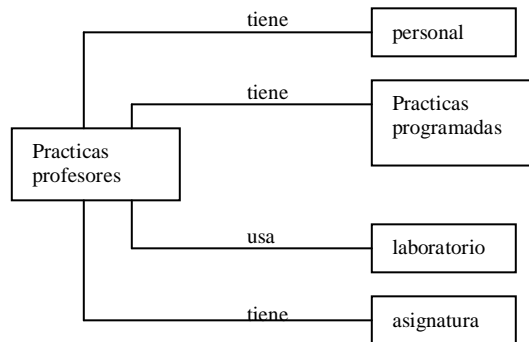


Figura 3.4 e) Esquema conceptual local del objeto practicas profesores.

**Esquema lógico local del objeto prácticas profesores.**

Prácticas de profesores(idpprof, idp, idpprog, fecha, idl, no. De equipos, materia, nrc)

Personal (idp, nombre, edad, dirección, lugar de nacimiento, ciudad, estado, teléfono, especialidad, grado, experiencia, departamento, fecha, comentarios)

Prácticas programadas(idpprog, idl, idc, ubc, idi, ubi, ids, ubs, idom, ubom, materia, nrc, entrega)

Laboratorios (idl, nombre, ids)

Asignaturas (nrc, materia, periodo, idl, ids, idc, idom, idi)

3. Derivar de un conjunto de relaciones (tablas) para cada esquema lógico local.

En esta sección se mostrara la derivación de tablas obtenidas del esquema lógico local del objeto laboratorio, como lo muestra la Figura 3.5.

**Esquema lógico local del objeto laboratorio.**

laboratorios (idl, nombre, ids)

sustancias (ids, tipo de medida, cantidad, fecha de compra, nombre sistemático, nombre común, fórmula, ubicación, motivo, proveedor, fecha de adquisición, observaciones, lote del producto, fecha de caducidad, grado de reactividad, propiedades, ubs)

Idl	Nombre	ids																
ids	Tipo de medida	cantidad	Fecha de compra	Fecha sistemático	Nombre común	fórmula	ubicación	motivo	proveedor	Fecha de adquisición	observaciones	Lote del producto	Fecha de caducidad	Grado de reactividad	Propiedades	ubs		
							Capítulo III											

Figura 3.5 Relación de tablas.

La normalización es un proceso que permite decidir a qué entidad pertenece cada atributo.

4. Construir y validar el esquema lógico global.

Laboratorios (idl, nombre, ids)

Sustancias (ids, tipo de medida, cantidad, fecha de compra, nombre sistemático, nombre común, fórmula, ubicación, motivo, proveedor, fecha de adquisición, observaciones, lote del producto, fecha de caducidad, grado de reactividad, propiedades, ubs)

Deudores (matricula, nombre, sección, fecha, asignación, idc, idi, idom, ids, observaciones)

Cristalería (idc, nombre, capacidad, existencia, especificación, ubc).

Instrumental (idi, nombre, capacidad, existencia, especificación, ubi).

Sustancias (ids, tipo de medida, cantidad, fecha de compra, nombre sistemático, nombre común, fórmula, ubicación, motivo, proveedor, fecha de adquisición, observaciones, lote del producto, fecha de caducidad, grado de reactividad, propiedades, ubs)

Otros materiales (idom, nombre, capacidad, existencia, especificación, ubom)

Asignaturas (nrc, materia, periodo, idl, ids, idc, idom, idi)

Cristalería (idc, nombre, capacidad, existencia, especificación, ubc).

Instrumental (idi, nombre, capacidad, existencia, especificación, ubi).

Sustancias (ids, tipo de medida, cantidad, fecha de compra, nombre sistemático, nombre común, fórmula, ubicación, motivo, proveedor, fecha de adquisición, observaciones, lote del producto, fecha de caducidad, grado de reactividad, propiedades, ubs)

Otros materiales (idom, nombre, capacidad, existencia, especificación, ubom)

Laboratorios (idl, nombre, ids)

Prácticas programadas(idpprog, idl, idc, ubc, idi, ubi, ids, ubs, idom, ubom, materia, nrc, entrega)

Cristalería (idc, nombre, capacidad, existencia, especificación, ubc).

Instrumental (idi, nombre, capacidad, existencia, especificación, ubi).

Sustancias (ids, tipo de medida, cantidad, fecha de compra, nombre sistemático, nombre común, fórmula, ubicación, motivo, proveedor, fecha de adquisición, observaciones, lote del producto, fecha de caducidad, grado de reactividad, propiedades, ubs)

Otros materiales (idom, nombre, capacidad, existencia, especificación, ubom)

Laboratorios (idl, nombre, ids)

Sustancias (ids, tipo de medida, cantidad, fecha de compra, nombre sistemático, nombre común, fórmula, ubicación, motivo, proveedor, fecha de adquisición, observaciones, lote del producto, fecha de caducidad, grado de reactividad, propiedades, ubs)

Asignaturas (nrc, materia, periodo, idl, ids, idc, idom, idi)

Cristalería (idc, nombre, capacidad, existencia, especificación, ubc).

Instrumental (idi, nombre, capacidad, existencia, especificación, ubi).

Otros materiales (idom, nombre, capacidad, existencia, especificación, ubom)

Laboratorios (idl, nombre, ids)

Prácticas de profesores (idpprof, idp, idpprog, fecha, idl, no. De equipos, materia, nrc)

Personal (idp, nombre, edad, dirección, lugar de nacimiento, ciudad, estado, teléfono, especialidad, grado, experiencia, departamento, fecha, comentarios)

Prácticas programadas(idpprog, idl, idc, ubc, idi, ubi, ids, ubs, idom, ubom, materia, nrc, entrega)

Cristalería (idc, nombre, capacidad, existencia, especificación, ubc).

Instrumental (idi, nombre, capacidad, existencia, especificación, ubi).

Sustancias (ids, tipo de medida, cantidad, fecha de compra, nombre sistemático, nombre común, fórmula, ubicación, motivo, proveedor, fecha de adquisición, observaciones, lote del producto, fecha de caducidad, grado de reactividad, propiedades, ubs)

Otros materiales (idom, nombre, capacidad, existencia, especificación, ubom)

Laboratorios (idl, nombre, ids)

Sustancias (ids, tipo de medida, cantidad, fecha de compra, nombre sistemático, nombre común, formula, ubicación, motivo, proveedor, fecha de adquisición, observaciones, lote del producto, fecha de caducidad, grado de reactividad, propiedades, ubs)

Asignaturas (nrc, materia, periodo, idl, ids, idc, idom, idi)

Cristalería (idc, nombre, capacidad, existencia, especificación, ubc).

Instrumental (idi, nombre, capacidad, existencia, especificación, ubi).

Sustancias (ids, tipo de medida, cantidad, fecha de compra, nombre sistemático, nombre común, formula, ubicación, motivo, proveedor, fecha de adquisición, observaciones, lote del producto, fecha de caducidad, grado de reactividad, propiedades, ubs)

Otros materiales (idom, nombre, capacidad, existencia, especificación, ubom)

Laboratorios (idl, nombre, ids)

Laboratorios (idl, nombre, ids)

Sustancias (ids, tipo de medida, cantidad, fecha de compra, nombre sistemático, nombre común, formula, ubicación, motivo, proveedor, fecha de adquisición, observaciones, lote del producto, fecha de caducidad, grado de reactividad, propiedades, ubs)

Asignaturas (nrc, materia, periodo, idl, ids, idc, idom, idi)

Cristalería (idc, nombre, capacidad, existencia, especificación, ubc).

Instrumental (idi, nombre, capacidad, existencia, especificación, ubi).

Sustancias (ids, tipo de medida, cantidad, fecha de compra, nombre sistemático, nombre común, formula, ubicación, motivo, proveedor, fecha de adquisición, observaciones, lote del producto, fecha de caducidad, grado de reactividad, propiedades, ubs)

Otros materiales (idom, nombre, capacidad, existencia, especificación, ubom)

Laboratorios (idl, nombre, ids)

### 3.2.4 Modelo físico.

**Algunos elementos del modelo físico son:**

**Vistas:** Son visiones lógicas de tablas, que permiten entregar a los usuarios solo la información que a estos les interesa. Facilitan el control de la seguridad de la base de datos. Las Figuras 3.6 – 22, presentan las vistas del sistema MACOLAQUI.

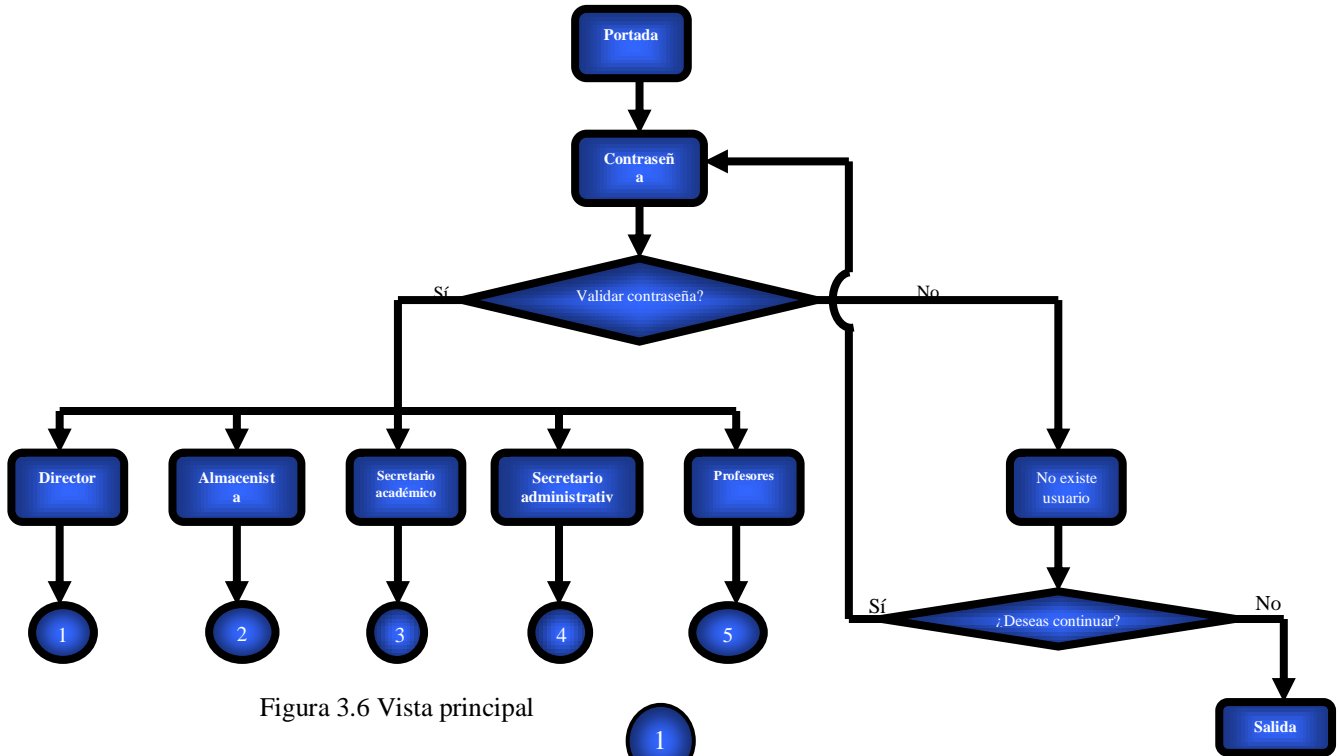


Figura 3.6 Vista principal

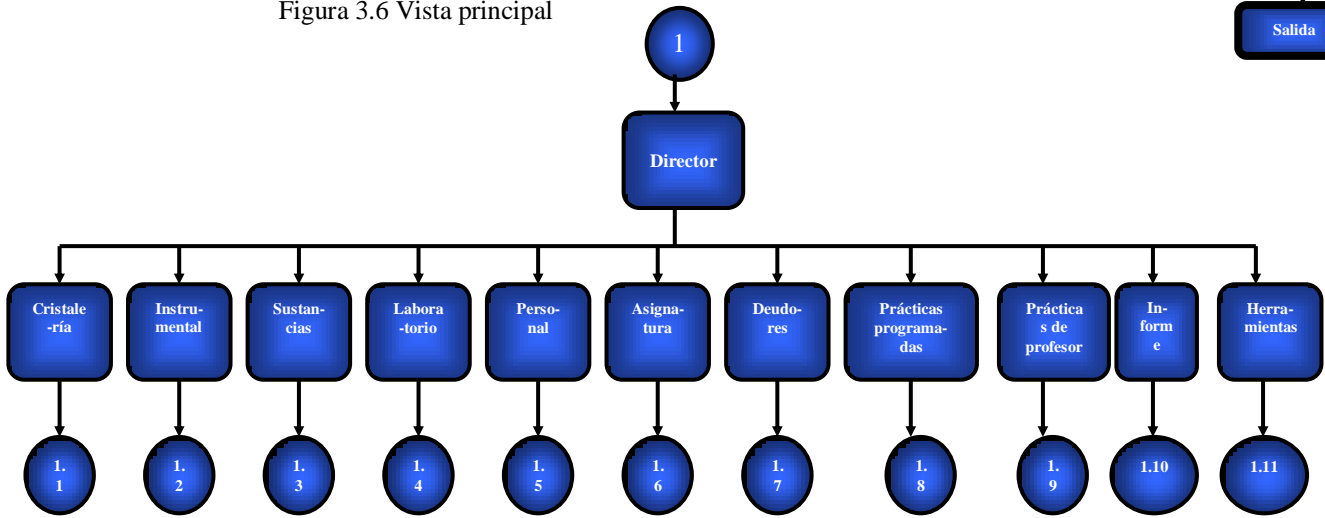


Figura 3.7 Vista del actor director

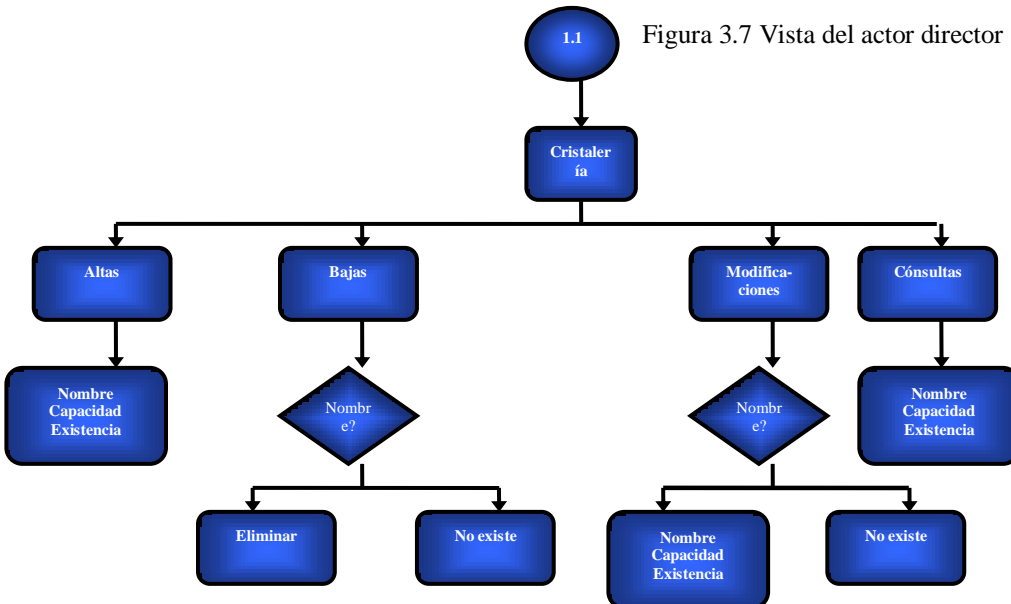


Figura 3.8 Vista de la entidad cristalería.

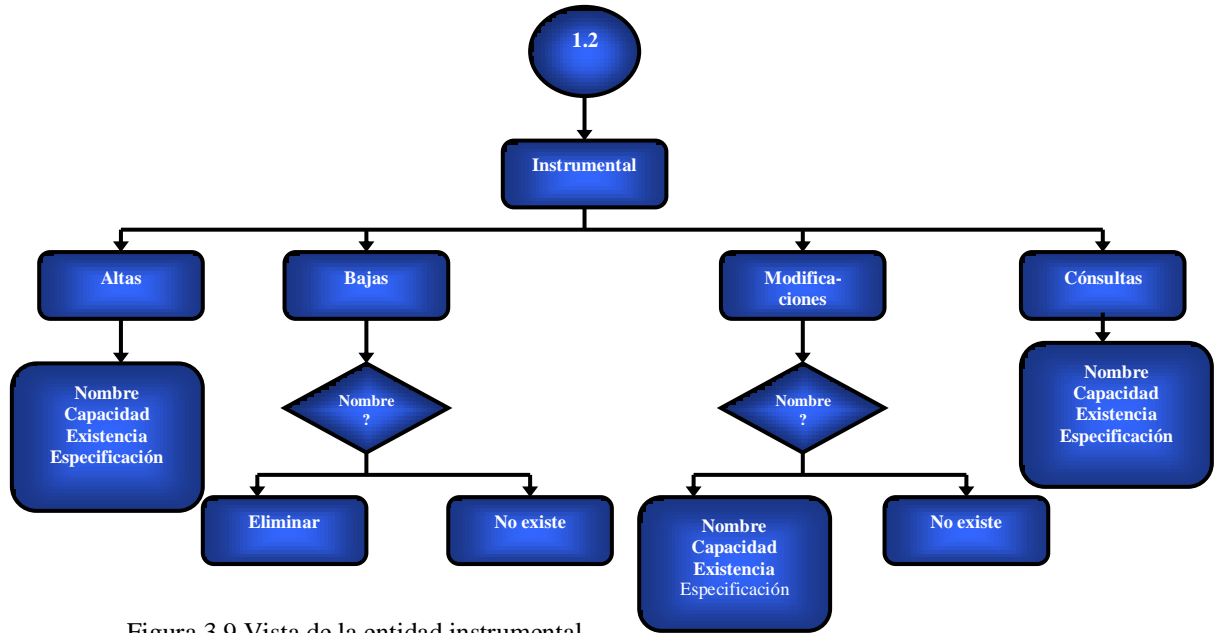


Figura 3.9 Vista de la entidad instrumental

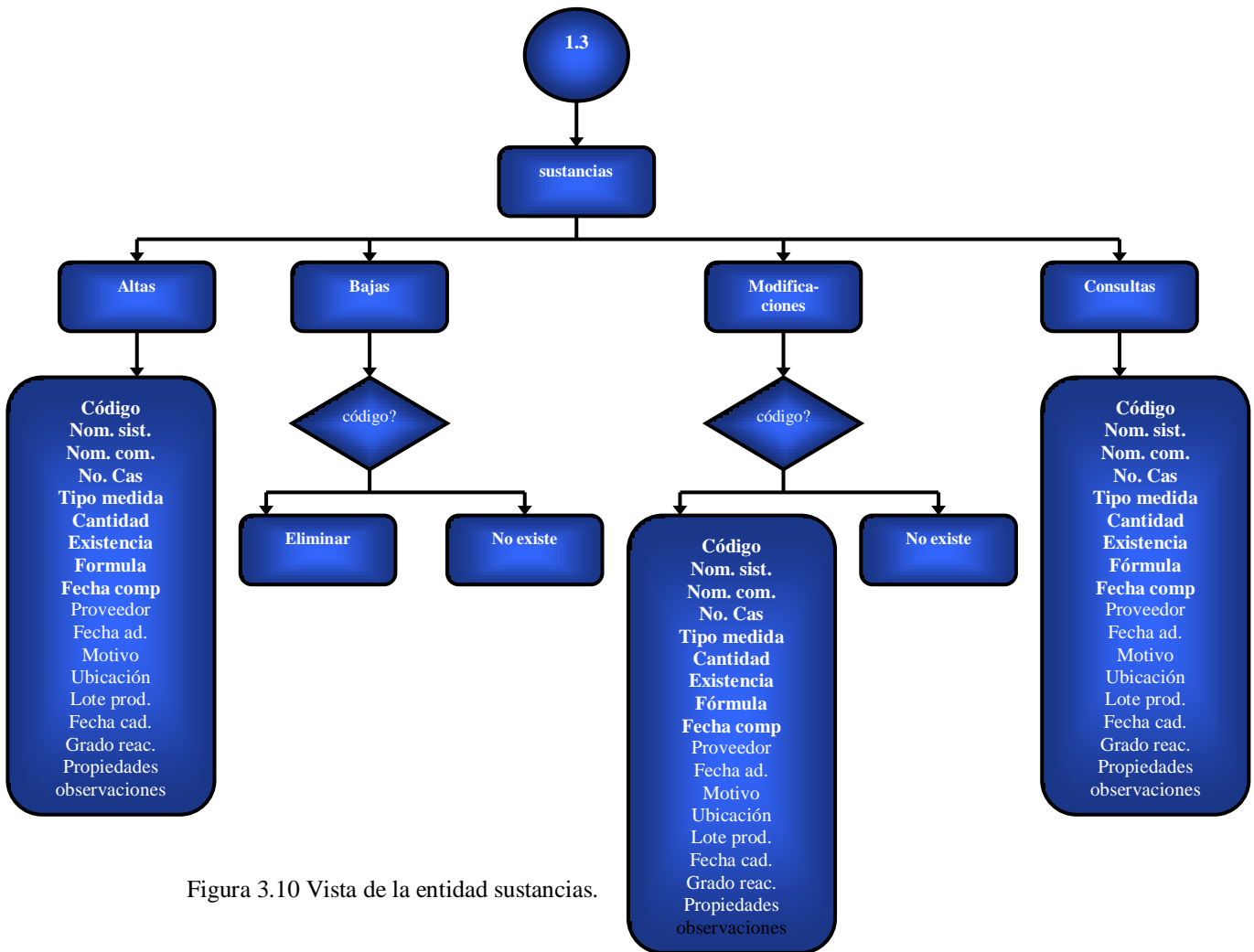


Figura 3.10 Vista de la entidad sustancias.

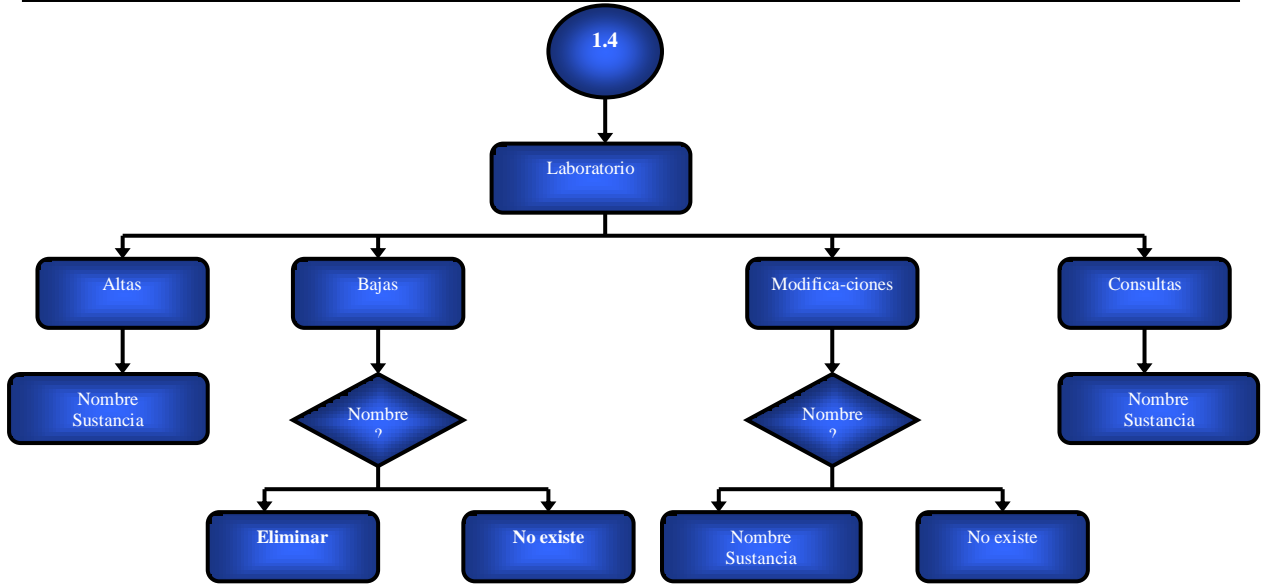


Figura 3.11 Vista de la entidad laboratorio.

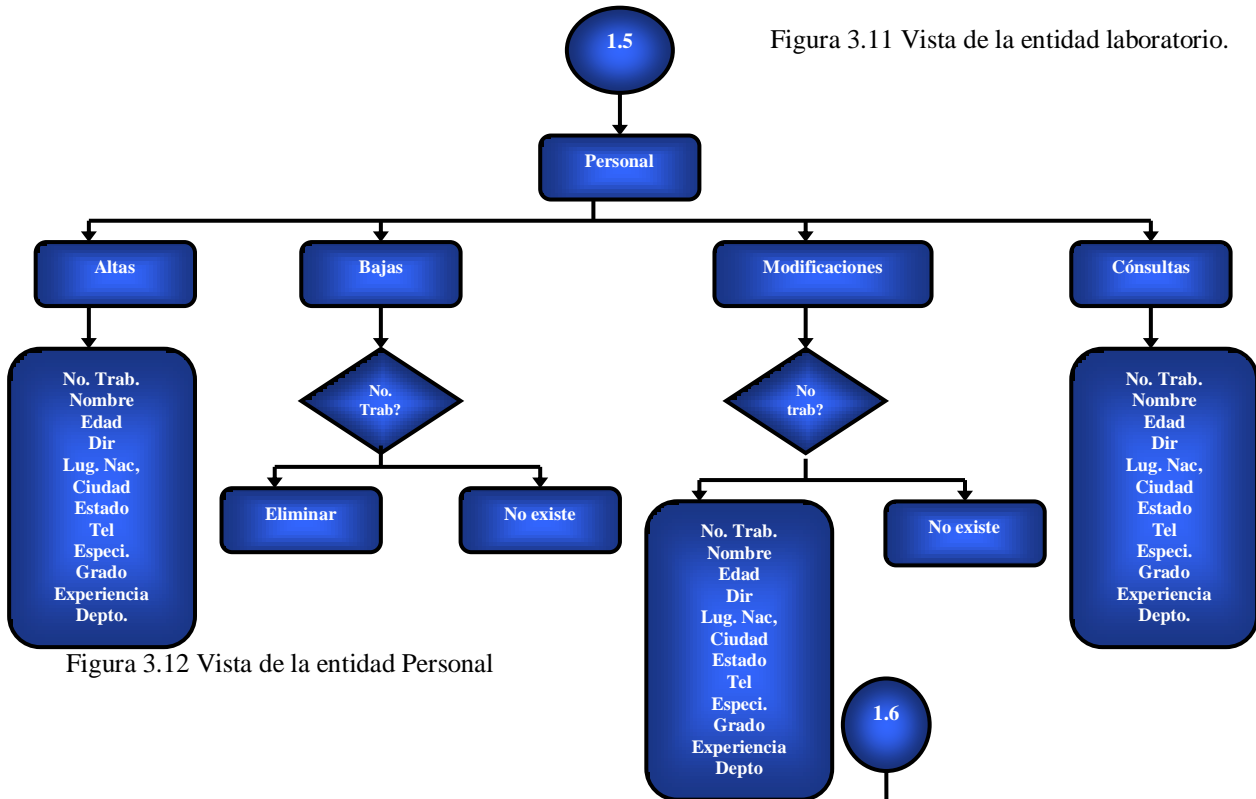


Figura 3.12 Vista de la entidad Personal

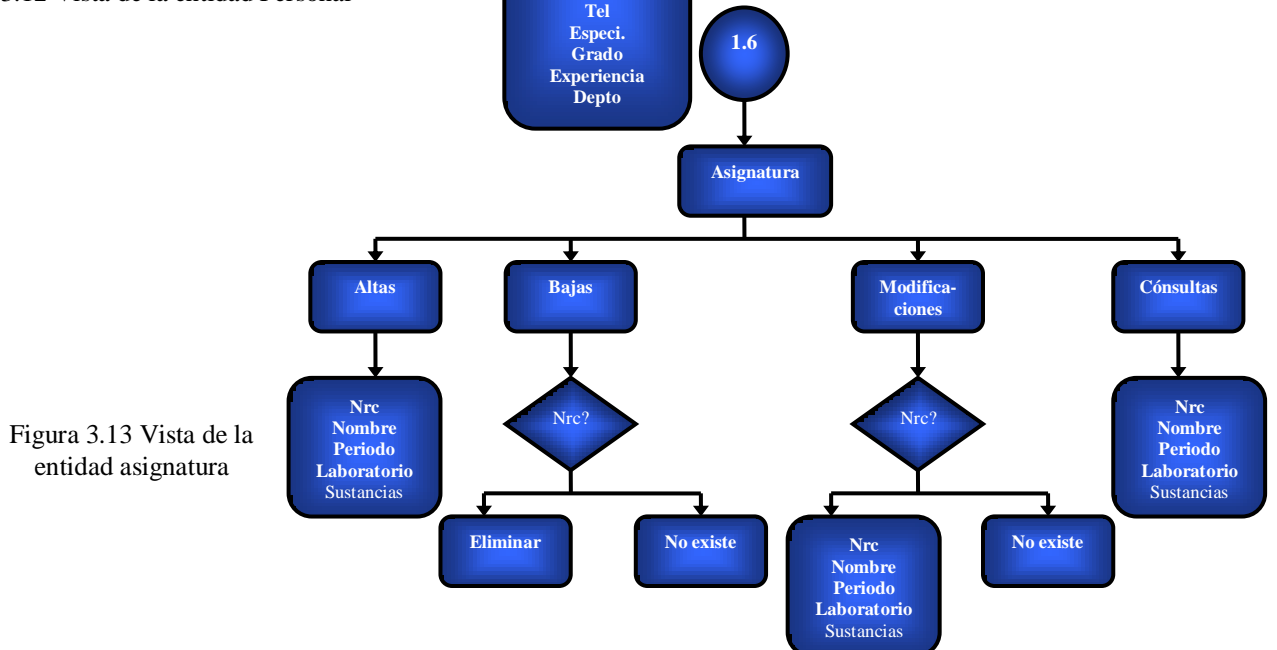


Figura 3.13 Vista de la entidad asignatura

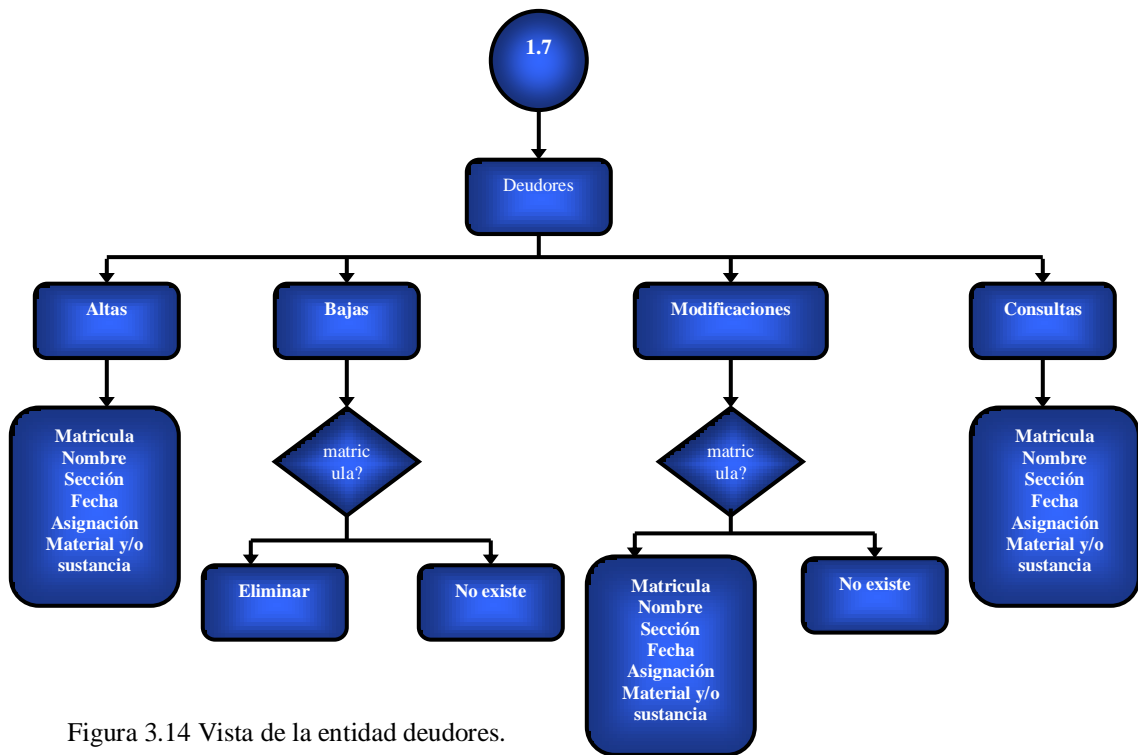


Figura 3.14 Vista de la entidad deudores.

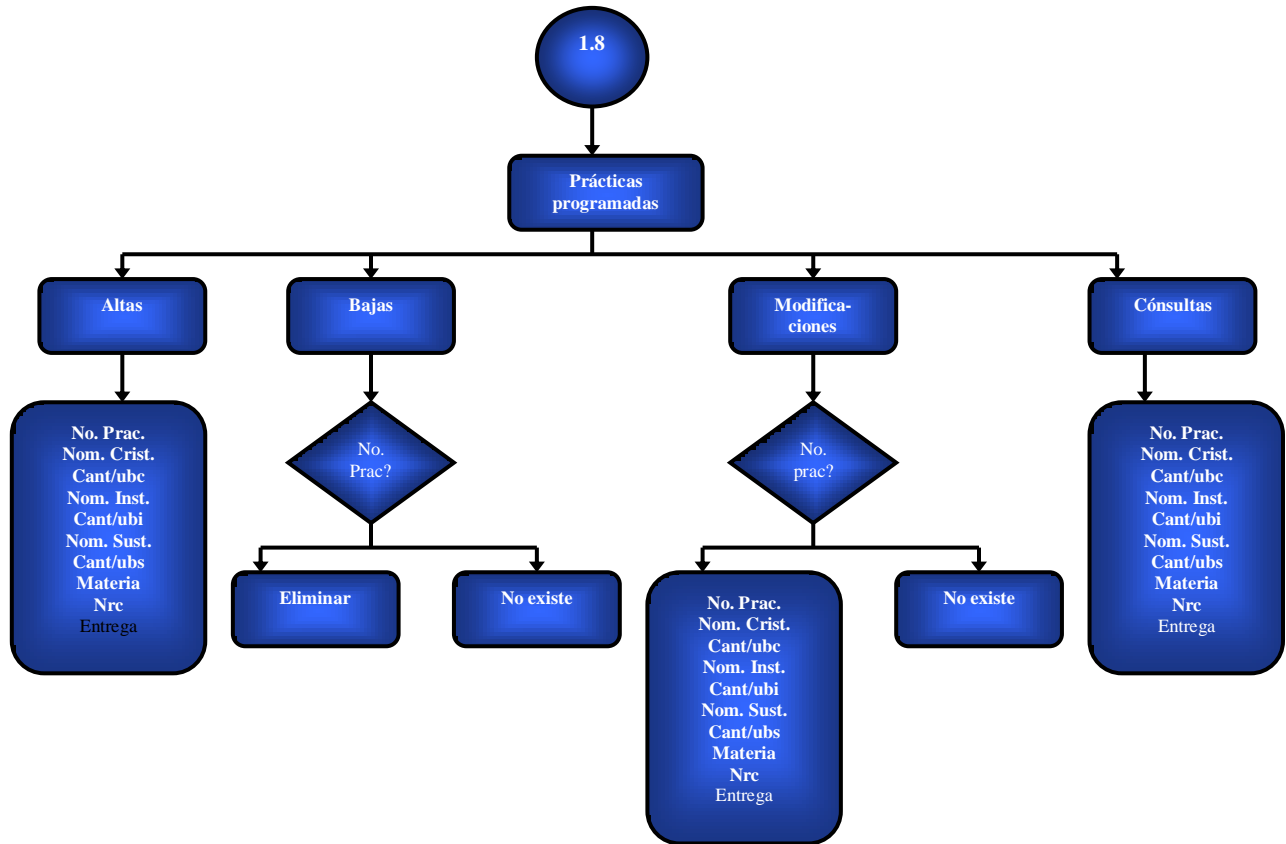


Figura 3.15 Vista de la entidad prácticas programadas.

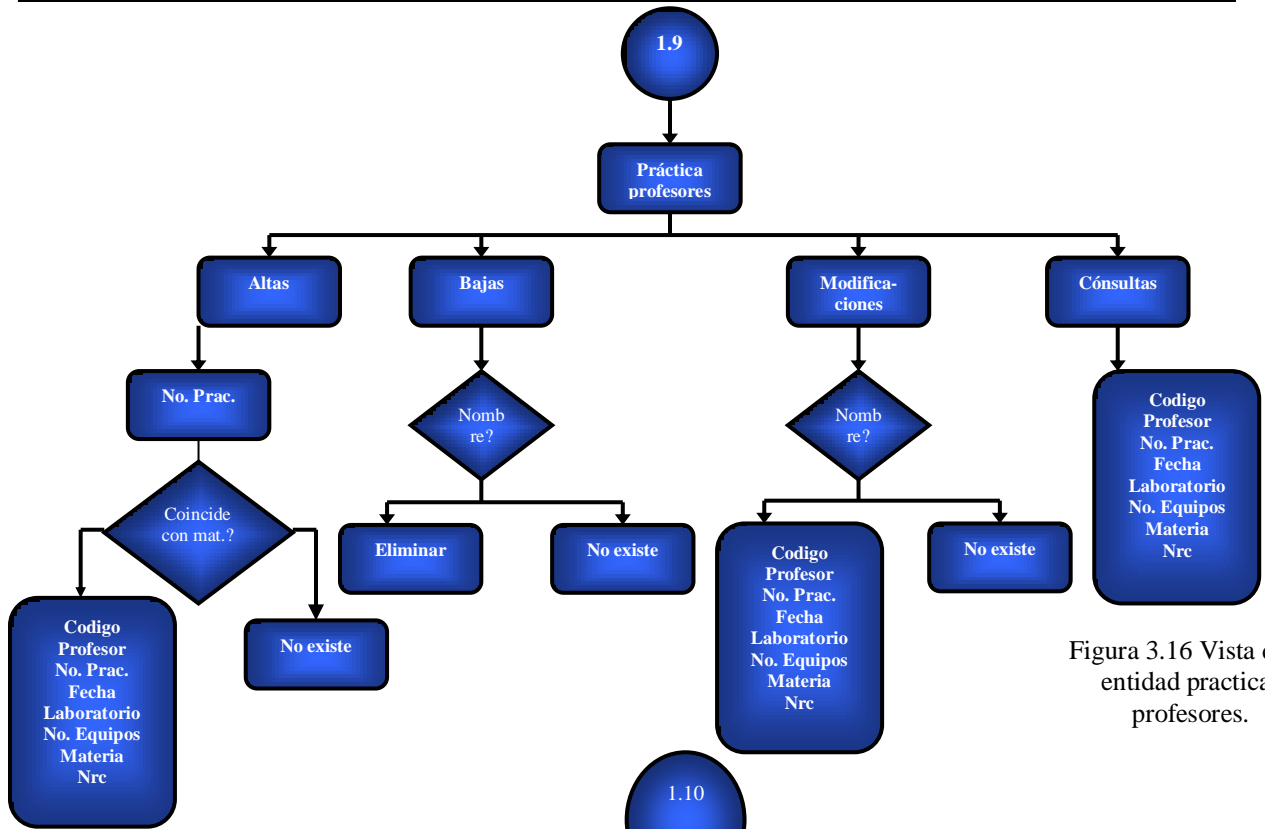


Figura 3.16 Vista de la entidad practicas profesores.

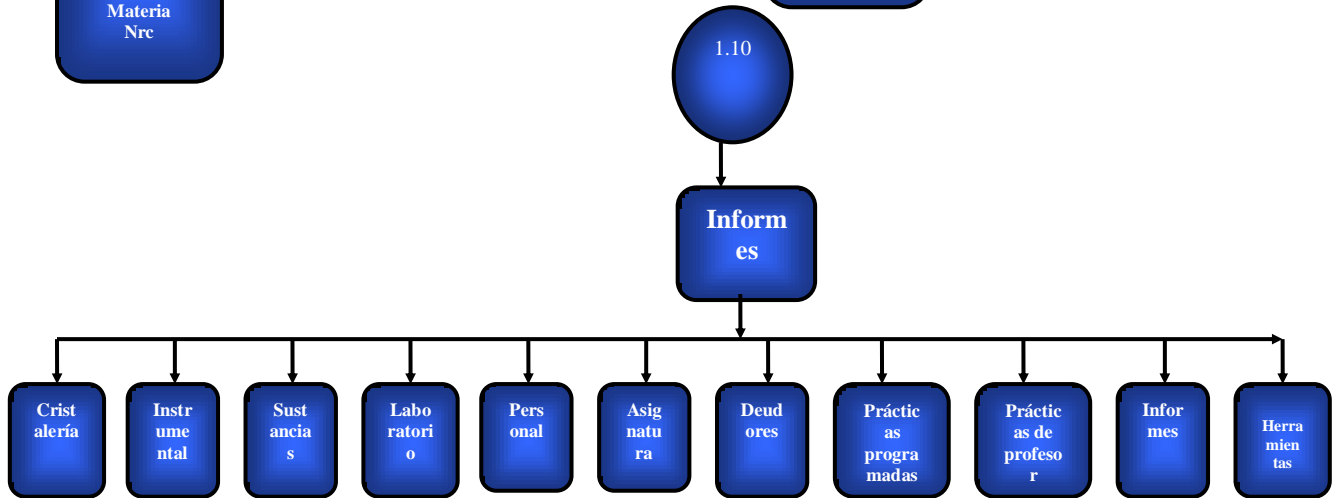


Figura 3.17 Vista de los informes de las entidades.

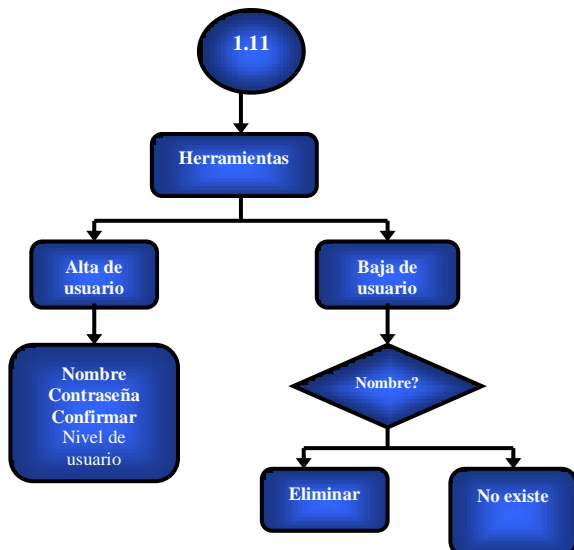


Figura 3.18 Vista de herramientas.

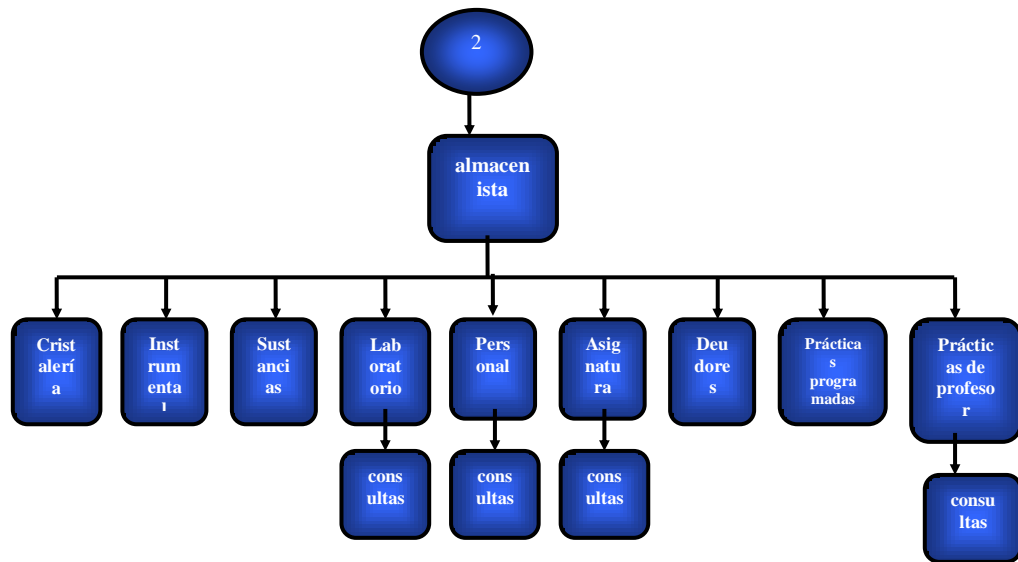


Figura 3.19 Vista del actor almacenista.

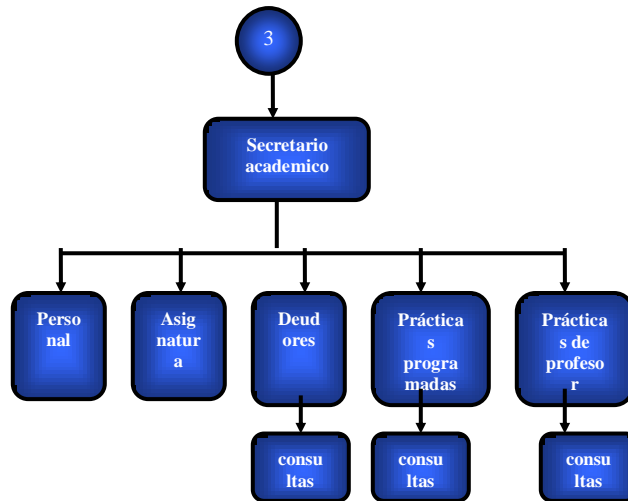


Figura 3.20 Vista del actor secretario académico.

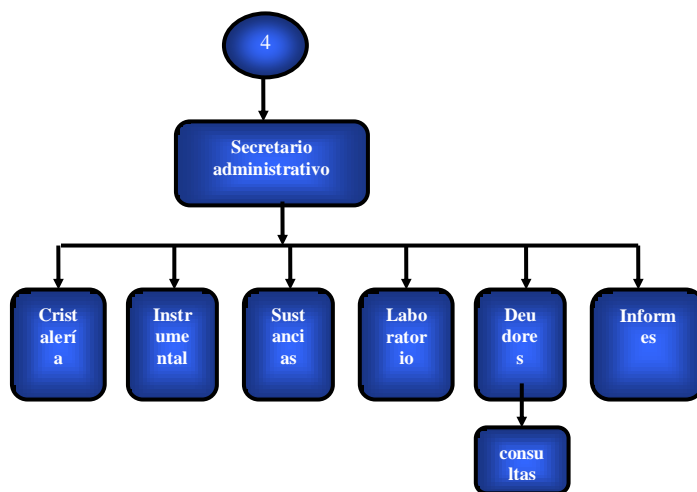


Figura 3.21 Vista del actor secretario administrativo

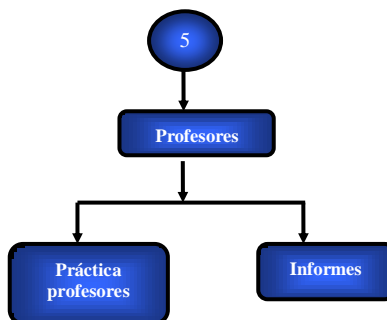


Figura 3.22 Vista de actor profesores.

## Capítulo IV. Implantación del sistema y resultados

### 4.1 Introducción

Después del diseño, se inicia, la implantación del sistema y conclusiones. Siguiendo el proceso como solución del problema, de la etapa de ingeniería, en este capítulo se abarca la fase de desarrollo del software<sup>2</sup>, la cual contempla los últimos dos puntos fuertes, la codificación y pruebas del software.

### 4.2 Análisis de la herramienta Progress

**Parámetros máximos, de la base de datos.** Progress Enterprise RDBMS, presenta los siguientes datos:

Tablas: 32,000

Índices: 32,000

Campos por tabla: 32,000

Usuarios concurrentes por base de datos: Hasta 10,000

Tamaño del registro: 32,000

Conjunto de "Buffers" (32 bits) : 125,000,000 buffers/128 gigabytes (64 bits).

500,000 buffers/4 Gigabytes (32 bits)

#### Los tipos de datos soportados por Progress son:

Carácter: 32,000 bytes (máximo)

Fecha: desde 1/1/32768 A. C. Hasta 1/1/32767 D. C.

Decimal: Un total de 50 dígitos, de 1 – 10 puntos decimales.

Número entero: de –2,1467, 483,648 a 2,167,483,648

Lógica: Verdadero / falso, si/no

Soporte de red

TCP/IP

### 4.3 Implantación del sistema implementación

De acuerdo a lo anterior se tienen las siguientes pantallas presentadas en las Figuras 4.1-4.12. Cuando inicialice el sistema se visualiza la pantalla de la Figura 4.1.



Cuando se le dé clic en el botón cerrar, la presentación se cierra y se visualiza el cuadro de dialogo de la Figura 4.2 que nos pide una contraseña de entrada. Los usuarios podrán

<sup>2</sup> Tema visto en el capítulo 1.

adquirir la contraseña con el director de la facultad de química de la BUAP, debido a que él controla al sistema y solo el podrá dar de alta a los usuarios que lo necesiten, de acuerdo al nivel de jerarquía de los usuarios, así como, darlos de baja del sistema.

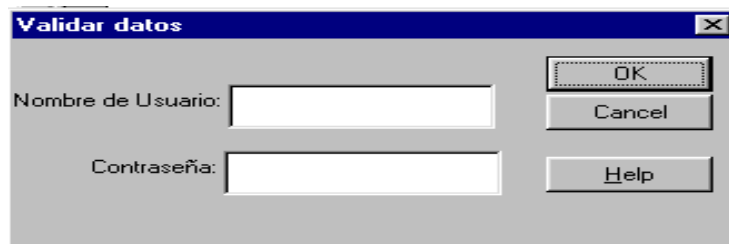


Figura 4.2 Cuadro de dialogo de validación de contraseña.

Después de introducir el nombre y la contraseña, el sistema visualizara la pantalla de acuerdo al tipo de usuario que sea. Por ejemplo, si es el director se visualizará la pantalla de la Figura 4.3:



Fig. 4.3 Pantalla principal del sistema MACOLAQUI nivel de usuario director.

Si es el almacenista se visualizara la pantalla de la Figura 4.4, restringiéndolo a las operaciones de: informes y herramientas:



Figura. 4.4 Pantalla principal de almacenista.

Si es el Secretario académico, visualizara pantalla de la Figura 4.5, el solo tiene acceso a: personal, materia, prácticas programadas, prácticas profesores y deudores.



Figura 4.5 Pantalla principal de secretario académico.

Si es el Secretario administrativo, visualizara pantalla de la Figura 4.6, en la que solo se le permite: cristalería, instrumental, sustancias, otros materiales, laboratorio e informes.



Figura 4.6 Pantalla principal de secretario administrativo.

Si es profesor, visualizara la pantalla de la Figura 4.7, restringiéndolo a prácticas de profesores e informes.

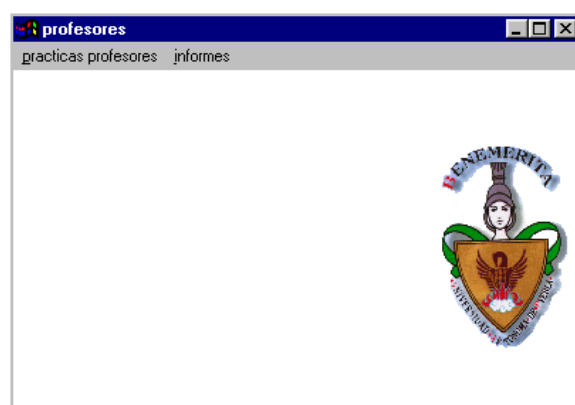


Figura. 4.7 Pantalla principal de profesores.

Si es el director y elige unos de los menús se visualiza la Figura 4.8, la interfaz de la Figura, en la que se le permite dar de alta, baja, modificar y consultar la cristalería.

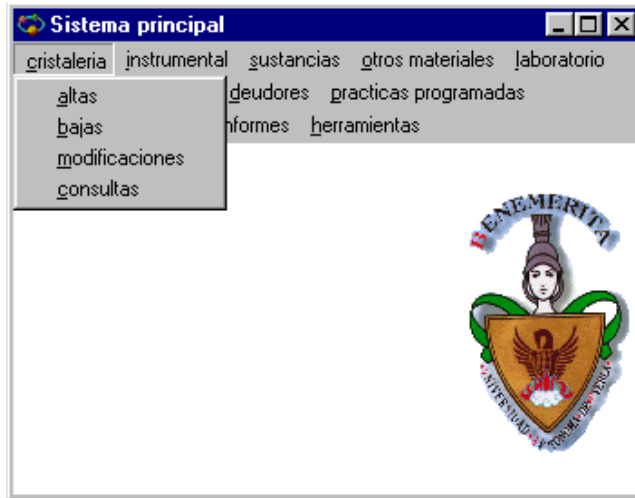
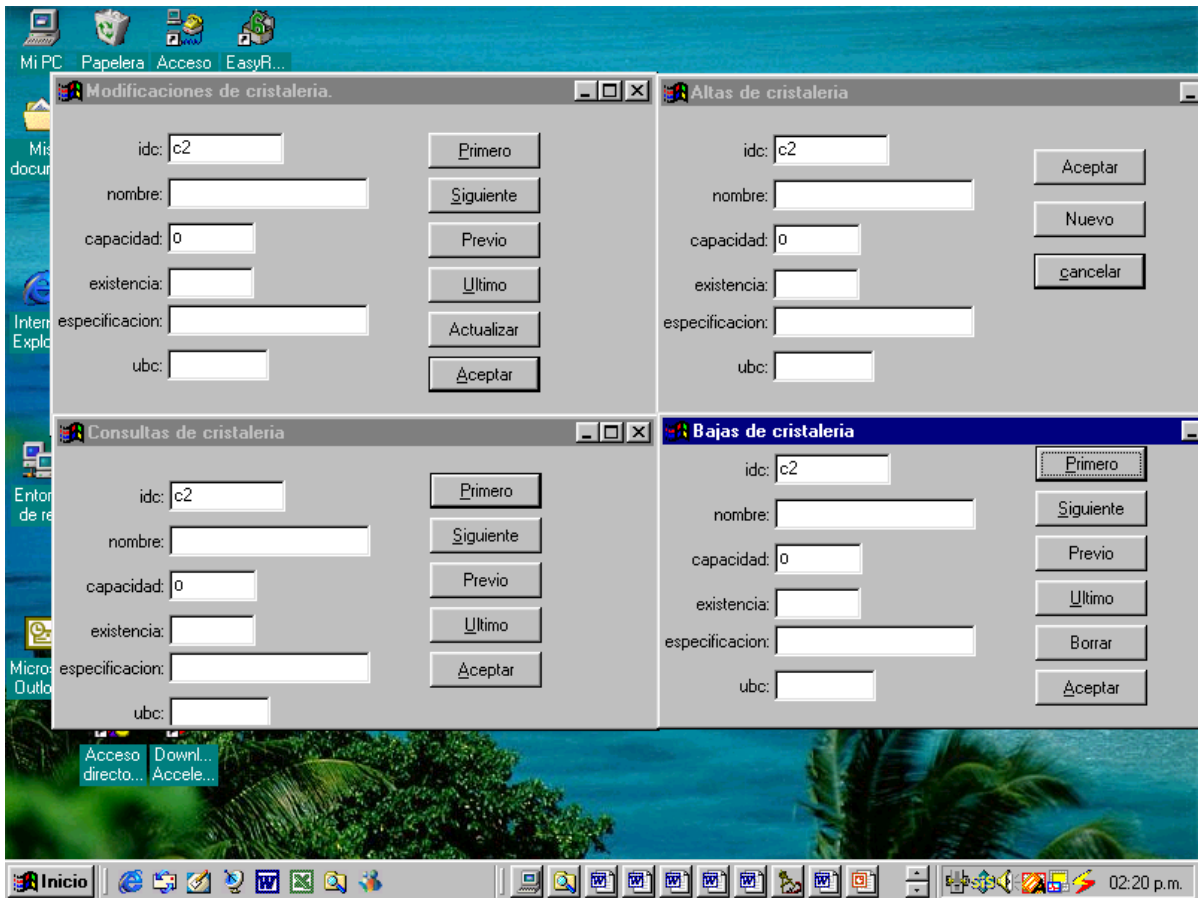


Figura 4.8 comandos de los menús del sistema principal.

La Figura 4.9 presenta la interfaz de cada opción del menú cristalería.



Una de las ventanas importantes es la alta y baja de usuarios que se encuentra en Herramientas, como lo muestra la Figura 4.10.

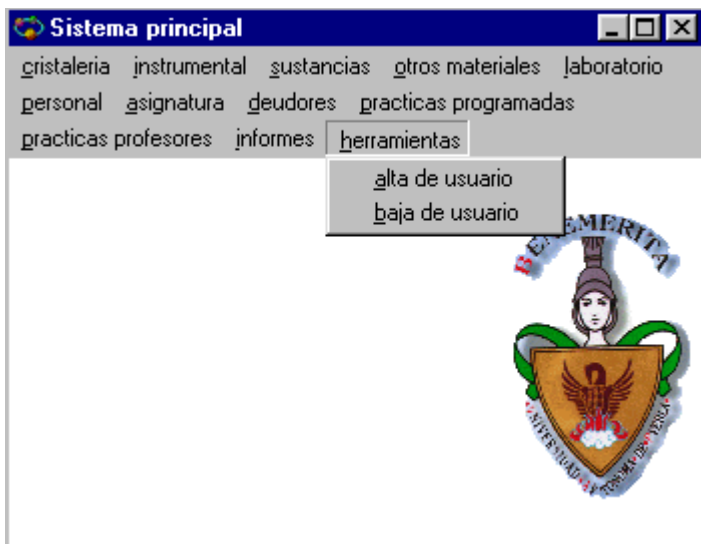


Figura. 4.10 Comandos del menú herramientas del sistema principal.

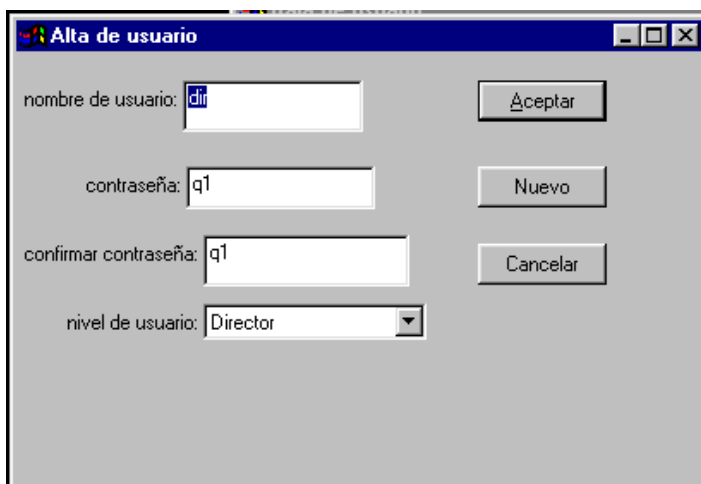


Figura 4.11 Ventana alta de usuario.

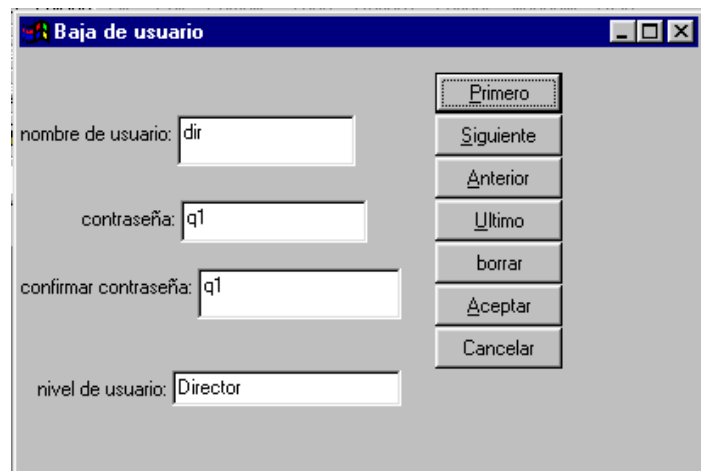


Figura 4.12 Ventana baja de usuario.

De acuerdo a todo lo analizado y planeado se tienen las siguientes pantallas de prueba mostradas en las Figuras 4.13 y 4.14, al sistema real. La Figura 4.13 es la pantalla principal. La Figura 4.14 muestra la misma pantalla principal, pero, con un menú desglosándose.

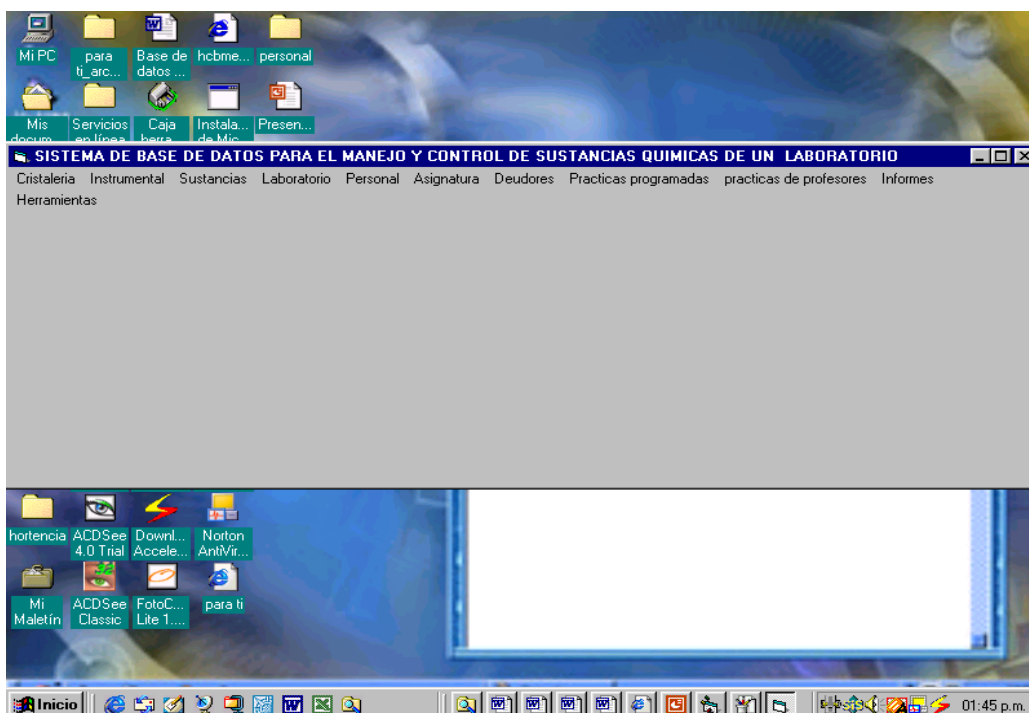


Figura. 4.13 Pantalla de prueba, visualizando como quedara el sistema.

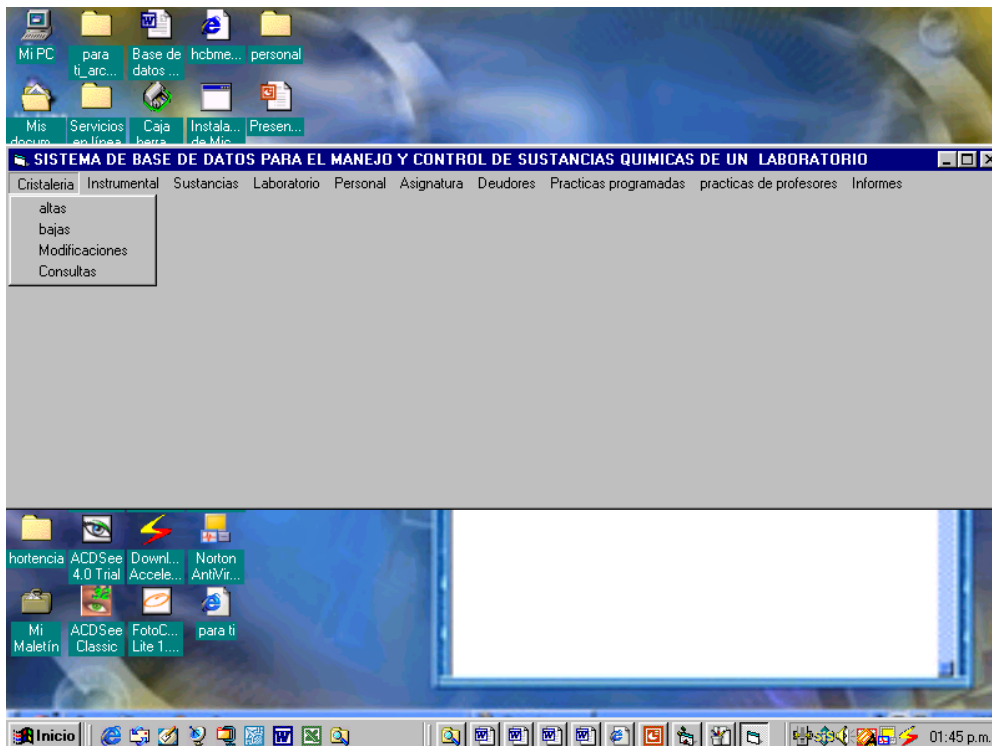


Figura 4.14 Pantalla de prueba principal.

La estructura final del sistema (MACOLAQUI) se presenta en la Figura 4.15.

### Estructura final del sistema MACOLAQUI

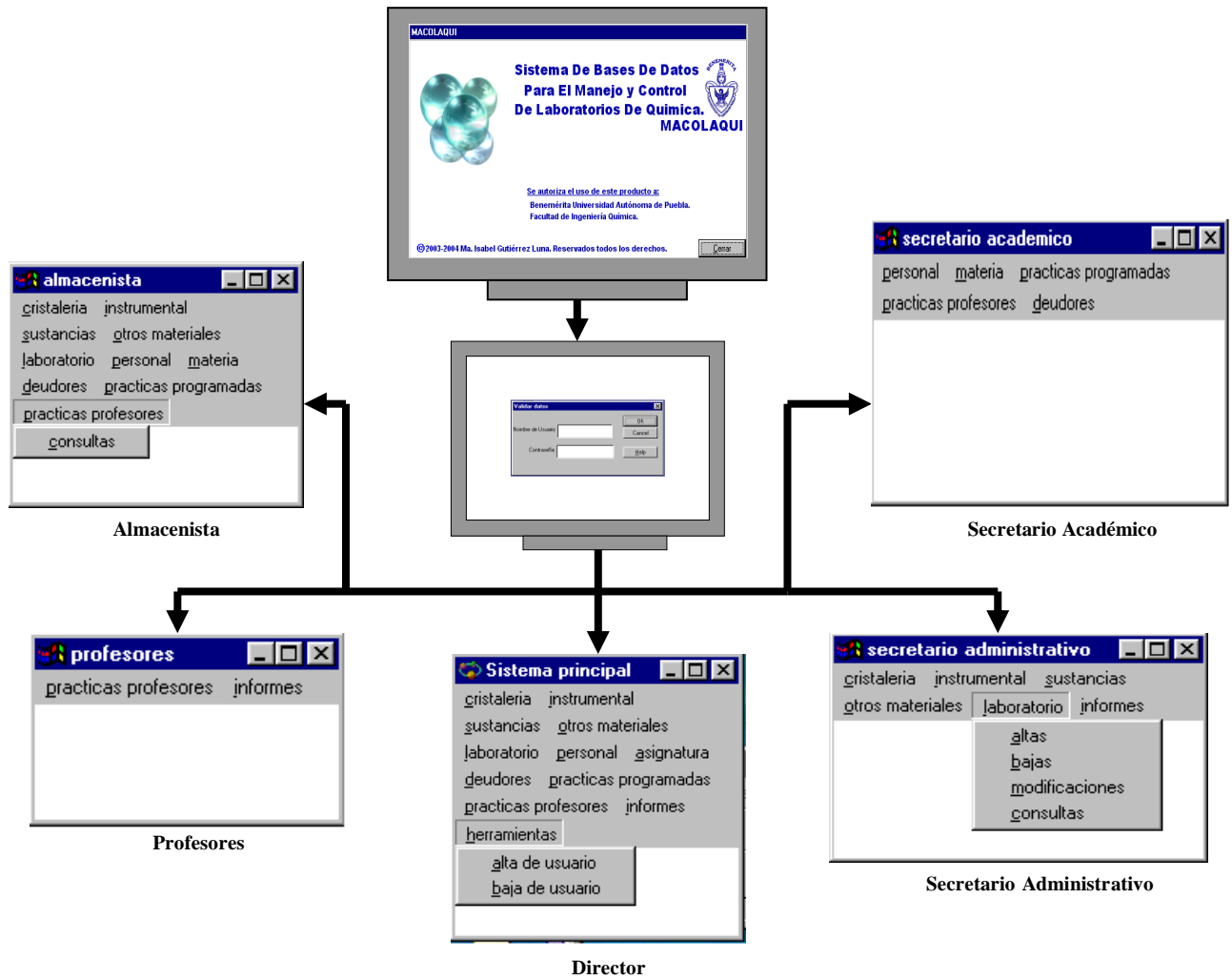


Figura 4.15 Estructura final del sistema.

---

---

## Conclusiones

### Objetivos Generales Y Específicos Del Proyecto.

Los objetivos generales del sistema se cumplieron, en un principio se pensó en realizar un sistema que mostrara un control de las diferentes sustancias químicas de un laboratorio, debido a las necesidades del cliente se incrementaron mas aplicaciones y se planeo en realizar un sistema que mostrara un control de los diferentes elementos\* que manipula laboratorios de química, con el fin de facilitar la obtención de las asignaturas que mayor uso hace de ellos, así como del personal encargado para su uso y administración.

\*Los elementos a manipular son:

- Cristalería.
- Instrumental.
- Sustancias.
- Materiales.
- Laboratorios.
- Personal.
- Asignaturas.
- Deudores.
- Practicas programadas.
- Practicas profesores.
- Herramientas del sistema.

### Propuestas abiertas

#### Arquitectura cliente servidor

La arquitectura cliente servidor es un modelo para el desarrollo de sistemas de información, en el que las transacciones se dividen en procesos independientes que cooperan entre si para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el dialogo o solicita los recursos y servicios, al proceso que responde a las solicitudes.

Se usara el nivel 4, como lo muestra la Figura 4.16.

Base de datos remota

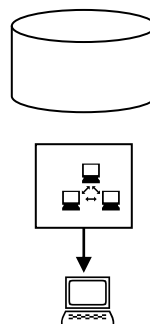


Figura 4.16 Nivel 4 de la arquitectura cliente – servidor.

En el cuarto nivel el cliente realiza tanto las funciones de presentación como los procesos. Por su parte, el servidor almacena y gestiona los datos que permanecen en una base de datos centralizada. En esta situación se dice que hay una gestión de datos remota.

La Figura 4.17 muestra las Funciones básicas de la arquitectura cliente servidor para mi sistema:

---

## Almacenista

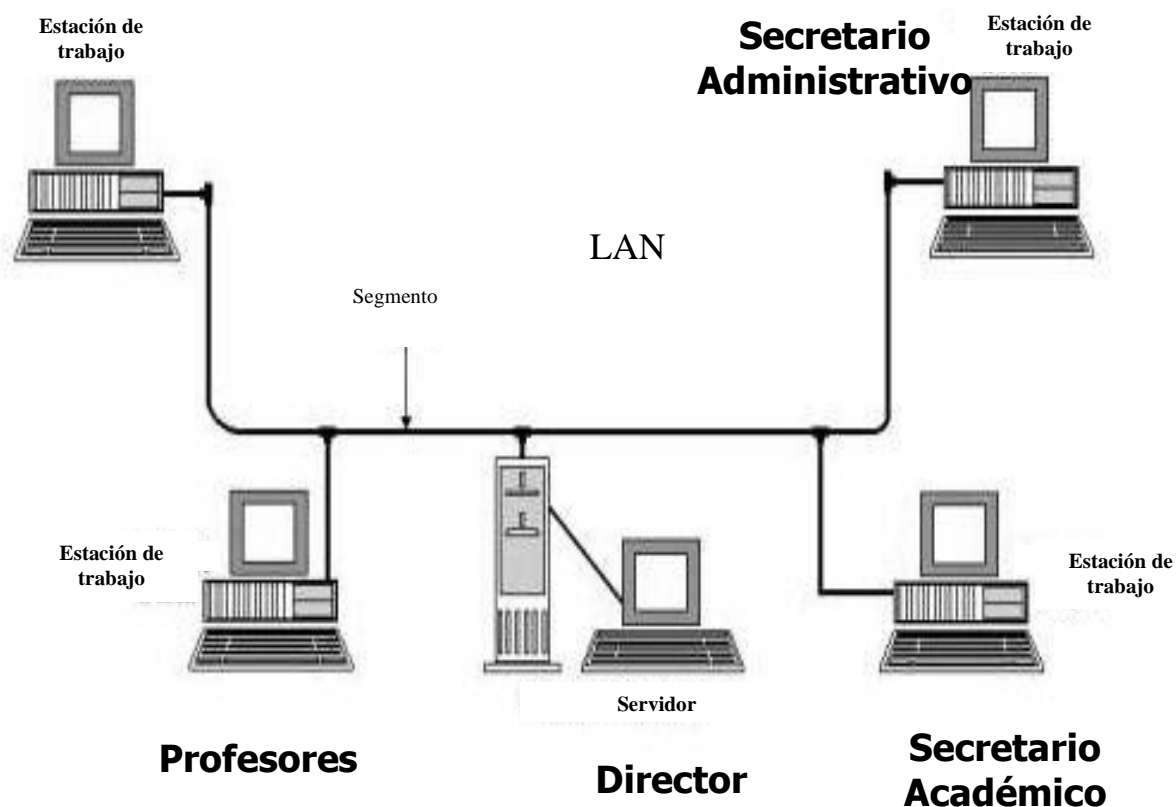


Figura 4.17 Funciones básicas con la arquitectura cliente – servidor para el sistema MACOLAQUI.

Otra propuesta más es con respecto al software, elegir software libre como PHP y MySQL.

Se puede implementar el sistema bajo el ambiente web utilizando software de licenciamiento libre, ya que a la empresa le trae muchos beneficios económicos, así como, el obtener una interfaz más agradable al usuario.

---

## Bibliografía

1. Peter Rob, Carlos Coronel. Database Systems Navathe, Cengage Learning, 2009.
2. Salvador Sánchez Alonso, Miguel Ángel Sicilia Urbán, Daniel Rodríguez García. Ingeniería del software: Un enfoque desde la guía SWEBOK. Ibergarceta Publicaciones, S.L., 2011
3. Ma. Victoria Nevado Cabello. Introducción a Las Bases de Datos Relacionales. Vision Libros 2010.
4. Andrew J. Oppel. Fundamentos de Bases de Datos. McGraw-Hill Interamericana de España S.L. 2010.
5. Lic. Jorge Jiménez González. Notas de ingeniería de software. Diplomado de bases de datos. FCC-BUAP.
6. Jorge Del Río San José. Introducción al tratamiento de datos espaciales en hidrología. Ed. Bubok 2010.
7. Vicente Fernández Alarcón. Desarrollo de sistemas de información: Una metodología basada en el modelado. Ed. Upc 2010.
8. Olga Pons. Introducción a los Sistemas de Bases de Datos. Ed. Paraninfo. 2008.
9. Antonio Iturbe, Lourdes S Nchez, Luis Ch. Consideraciones Conceptuales Sobre Los Sistemas de Información 2011.
10. Emma Rico. Diseño y Administración de Bases de Datos. Ed. UPC 2009. Prentice Hall
11. Grady Booch, Robert A. Maksimchuk, Michael W. Engle. Object-oriented analysis and design with applications. Addison-Wesley, 2007

## Anexos

### Anexo 1.1. Identificar los atributos y asociarlos a entidades y relaciones

Identificación de atributos									
Entidad	Atributo	Alias	Tipo de dato	longitud	Valores por defecto	Siempre tendrá valor	Compuesto/Simple	Derivado	Multievaluado
Cristalería	Idc	No	Carácter	10	No	Si	S	No	Si
	nombre	No	Carácter	25	No	Si	S	No	Si
	Capacidad	No	Numero	6	No	No	C	No	Si
	Existencia	No	Carácter	6	No	No	C	Si	Si
	Especificación	No	Carácter	25	No	No	S	No	Si
	Ubc	No	Carácter	8	No	No	C	No	Si

Tabla 1.1.1. Atributos de la entidad cristalería.

Identificación de atributos									
Entidad	Atributo	Alias	Tipo de dato	longitud	Valores por defecto	Siempre tendrá valor	Compuesto/Simple	Derivado	Multievaluado
Instrumental	idi	No	Carácter	10	No	Si	S	No	Si
	nombre	No	Carácter	25	No	Si	S	No	Si
	capacidad	No	Numero	6	No	No	C	No	Si
	Existencia	No	Carácter	6	No	No	C	Si	Si
	Especificación	No	Carácter	25	No	No	S	No	Si
		ubi	No	Carácter	8	No	No	C	No

Tabla 1.1.2 Atributos de la entidad instrumental.

Identificación de atributos									
Entidad	Atributo	Alias	Tipo de dato	longitud	Valores por defecto	Siempre tendrá valor	Compuesto/Simple	Derivado	Multievaluado
Sustancia	ids	No	Carácter	10	No	Si	S	No	Si
	Tipo de medida	No	carácter	15	No	No	S	No	Si
	Cantidad	No	Carácter	15	No	No	S	No	Si
	fecha de compra	No	Carácter	10	No	No	S	No	Si
	nombre sistemático	No	Carácter	25	No	Si	S	No	Si
	nombre común	No	Carácter	25	No	Si	S	No	Si
	Formula	No	Carácter	25	No	No	S	No	Si
	Ubicación	No	Carácter	25	No	No	S	No	Si
	Motivo	No	Carácter	15	No	No	S	No	Si
	Proveedor	No	Carácter	25	No	No	S	No	Si
	Fecha de adquisición	No	Carácter	10	No	Si	S	No	Si
	Observaciones	No	memo	4	No	No	S	No	Si
	Lote de producto	No	Carácter	10	No	Si	S	No	Si
	Fecha de caducidad	No	Carácter	10	No	Si	S	No	Si
	Grado de reactividad	No	Carácter	10	Si	Si	S	No	Si
	propiedades	No	memo	4	no	Si	S	No	Si
	ubs	No							

Tabla 1.1.3 Atributos de la entidad sustancia.

Identificación de atributos									
Entidad	Atributo	Alias	Tipo de dato	longitud	Valores por defecto	Siempre tendrá valor	Compuesto/Simple	Derivado	Multievaluado
Otros	idom	No	Carácter	10	No	Si	S	No	Si

materiales	nombre	No	Carácter	25	No	Si	S	No	Si
	capacidad	No	Número	6	No	No	C	No	Si
	Existencia	No	Carácter	6	No	No	C	Si	Si
	Especificación	No	Carácter	25	No	No	S	No	Si
	ubom	No	Carácter	8	No	No	C	No	Si

Tabla 1.1.4 Atributos de la entidad otros materiales.

Identificación de atributos									
Entidad	Atributo	Alias	Tipo de dato	longitud	Valores por defecto	Siempre tendrá valor	Compuesto/Simple	Derivado	Multievaluado
<b>Laboratorio</b>	<b>idl</b>	No	Carácter	10	No	Si	S	No	Si
	Nombre de laboratorio	Si nombre	carácter	20	No	Si	S	No	Si
	ids	No	Carácter	10	No	No	S	Si	Si

Tabla 1.1.5 Atributos de la entidad laboratorio.

Identificación de atributos									
Entidad	Atributo	Alias	Tipo de dato	longitud	Valores por defecto	Siempre tendrá valor	Compuesto/Simple	Derivado	Multievaluado
Personalr	IDP	No	carácter	10	No	Si	S	No	Sí
	Nombre	No	Carácter	15	No	Si	S	No	Si
	Edad	No	Numérico	3	No	No	S	No	Sí
	Dirección	No	Carácter	35	No	Si	S	No	Si
	Lugar de nacimiento	No	Carácter	25	No	No	S	No	Si
	Ciudad	No	Carácter	25	No	No	S	No	Si
	Estado	No	Carácter	25	No	No	S	No	Si
	Teléfono	No	Carácter	25	No	No	S	No	Si
	Especialidad	No	Carácter	25	No	No	S	No	Si
	Grado	No	Carácter	15	No	No	S	No	Si
	Experiencia	No	memo	4	No	No	S	No	Si
	Área	No	Carácter	15	No	No	S	No	Si
	Fecha	No	Carácter	10	No	Si	S	No	Si
Comentarios	No	memo	4	No	Si	S	No	Si	

Tabla 1.1.6 Atributos de la entidad personal.

Identificación de atributos									
Entidad	Atributo	Alias	Tipo de dato	longitud	Valores por defecto	Siempre tendrá valor	Compuesto/Simple	Derivado	Multievaluado
Asignatura	Nrc	No	carácter	10	No	Si	S	No	Sí
	Materia	No	Carácter	20	No	Si	S	No	Si
	Periodo	No	Carácter	10	No	No	S	No	Sí
	idl	No	Carácter	10	No	No	S	Si	Si
	ids	No	Carácter	10	No	No	S	Si	Si
	idc	No	Carácter	10	No	No	S	Si	Si
	idom	No	Carácter	10	No	No	S	Si	Si
	idi	No	Carácter	10	No	No	S	Si	Si

Tabla 1.1.7 Atributos de la entidad asignatura.

Identificación de atributos									
Entidad	Atributo	Alias	Tipo de dato	longitud	Valores por defecto	Siempre tendrá valor	Compuesto/Simple	Derivado	Multievaluado
Deudores	Matricula	No	carácter	10	No	Si	S	No	Sí
	Nombre	No	Carácter	15	No	Si	S	No	Si
	Sección	No	Numérico	3	No	No	S	No	Sí
	Fecha	No	Carácter	10	No	Si	S	No	Si
	ids	No	Carácter	10	No	No	S	Si	Si

	idc	No	Carácter	10	No	No	S	Si	Si
	idom	No	Carácter	10	No	No	S	Si	Si
	idi	No	Carácter	10	No	No	S	Si	Si
	Asignación	No	Carácter	35	No	No	S	Si	Si
	observaciones	No	Memo	4	No	No	S	No	Si

Tabla 1.1.8 Atributos de la entidad deudores.

Entidad	Atributo	Alias	Tipo de dato	longitud	Valores por defecto	Siempre tendrá valor	Compuesto/Simpl e	Derivado	Multievaluado
Prácticas programadas	idpprog	No	carácter	10	No	Si	S	No	Sí
	Idc	No	Carácter	10	No	No	S	Si	Si
	Ubc	No	Carácter	8	No	No	C	Si	Si
	Idi	No	Carácter	10	No	No	S	Si	Si
	Ubi	No	Carácter	8	No	No	C	Si	Si
	Idom	No	Carácter	10	No	No	S	Si	Si
	Ubom	No	Carácter	8	No	No	C	Si	Si
	Idl x	No	Carácter	10	No	No	S	Si	Si
	Ubl x	No	Carácter	8	No	No	C	Si	Si
	Materia	No	Carácter	20	No	No	S	Si	Si
	Nrc	No	carácter	10	No	No	S	Si	Sí
Entrega	No	Carácter	8	No	No	S	No	Si	

Tabla 1.1.9 Atributos de la entidad practicas programadas.

Entidad	Atributo	Alias	Tipo de dato	longitud	Valores por defecto	Siempre tendrá valor	Compuesto/Simpl e	Derivado	Multievaluado
Prácticas profesores	idpprof	No	carácter	10	No	Si	S	No	Sí
	Idp	No	Carácter	10	No	No	S	Si	Si
	Idpprog	No	Carácter	10	No	No	S	Si	Si
	Idl	No	Carácter	10	No	No	S	Si	Si
	No. De equipos	Si Ne	Numero	2	No	No	S	No	Si
	Materia	No	Carácter	20	No	No	S	Si	Si
	Nrc	No	carácter	10	No	No	S	Si	Sí
Fecha	No	Carácter	8	No	No	S	No	Si	

Tabla 1.1.10 Atributos de la entidad practica profesor.

Entidad	Atributo	Alias	Tipo de dato	longitud	Valores por defecto	Siempre tendra valor	Compuesto/Simpl e	Derivado	Multievaluado
Usuario	Nombre de usuario	Si nomb re	carácter	25	No	Si	S	No	Sí
	Contraseña	No	Carácter	10	No	Si	S	No	Si
	Confirmar contraseña	Si cc	Carácter	10	No	Si	S	No	Si
	Nivel de usuario	Si Nu	Carácter	25	Si	Si	S	No	Si

Tabla 1.1.11 Atributos de la entidad usuario.

## Anexo 1.2. Determinar los dominios de los atributos.

Los dominios de los atributos de las entidades que se identificaron para la solución del sistema se presentan en las tablas 1.2.12-1.2.22, donde C identifica un carácter y 9 un número.

Atributo	Tipo de dato	Longitud	Dominio	Operaciones
Idc	Carácter	10	{CCCCCCCCCC}	Identificación de cristalería
Nombre	Carácter	25	{CCCCCCCCCCCCCCCCCCCC}	Tipo de la medida de la sustancia

Capacidad	Numero	4	{9999}	Cuantos se dieron de alta.
Existencia	Número	4	{9999}	Cuantos existen actualmente.
Especificación	Carácter	25	{CCCCCCCCCCCCCCCCCCCCCCCC}	Detalles de cristalería.
Ubc	Carácter	8	{CCCCCCCC}	Unidades con la que se mide la cristalería.

Tabla 1.2.12 Dominio de los atributos de cristalería

Atributo	Tipo de dato	Longitud	Dominio	Operaciones
Idi	Carácter	10	{CCCCCCCCCC}	Identificación de cristalería
Nombre	Carácter	25	{CCCCCCCCCCCCCCCCCCCCCCCC}	Tipo de la medida de la sustancia
Capacidad	Numero	4	{9999}	Cuantos se dieron de alta.
Existencia	Número	4	{9999}	Cuantos existen actualmente.
Especificación	Carácter	25	{CCCCCCCCCCCCCCCCCCCCCCCC}	Detalles de cristalería.
Ubi	Carácter	8	{CCCCCCCC}	Unidades con la que se mide la cristalería.

Tabla 1.2.13 Dominio de los atributos de instrumental

Atributo	Tipo de dato	Longitud	Dominio	Operaciones
Ids	Carácter	10	{CCCCCCCCCC}	Identificación de sustancia.
Tipo de medida	Carácter	15	{CCCCCCCCCCCCCCCC}	Tipo de la medida de la sustancia
Cantidad	Carácter	15	{CCCCCCCCCCCCCCCC}	Cantidad de la medida
fecha de compra	Carácter	10	{99/99/9999}	Fecha en que se compro
nombre sistemático	Carácter	25	{CCCCCCCCCCCCCCCCCCCCCCCC}	Nombre de origen
nombre común	Carácter	25	{CCCCCCCCCCCCCCCCCCCCCCCC}	Nombre como se conoce
Formula	Carácter	25	{CCCCCCCCCCCCCCCCCCCCCCCC}	Formula de la sustancia
Ubicación	Carácter	25	{CCCCCCCCCCCCCCCCCCCCCCCC}	Donde se encuentra ubicado dentro del almacén
Motivo	Carácter	15	{CCCCCCCCCCCCCCCC}	Como se adquirió
Proveedor	Carácter	25	{CCCCCCCCCCCCCCCCCCCCCCCC}	Quien lo vendió
Fecha de adquisición	Carácter	10	{99/99/9999}	Cuando se adquiere
Observaciones	memo	4	{CCCC}	Algún punto a considerar
Lote de producto	Carácter	10	{CC99999999}	No. De lote
Fecha de caducidad	Carácter	10	{99/99/9999}	Fecha en que caduca la sustancia
Grado de reactividad	Carácter	10	{CCCCCCCCCC}	Peligrosidad
propiedades	memo	4	{CCCC}	Propiedad de la sustancia

Tabla 1.2.14 Dominio de los atributos de las sustancias

Atributo	Tipo de dato	Longitud	Dominio	Operaciones
Idom	Carácter	10	{CCCCCCCCCC}	Identificación de cristalería
Nombre	Carácter	25	{CCCCCCCCCCCCCCCCCCCCCCCC}	Tipo de la medida de la sustancia
Capacidad	Número	4	{9999}	Cuantos se dieron de alta.
Existencia	Número	4	{9999}	Cuantos existen actualmente.
Especificación	Carácter	25	{CCCCCCCCCCCCCCCCCCCCCCCC}	Detalles de cristalería.
Ubom	Carácter	8	{CCCCCCCC}	Unidades con la que se mide la cristalería.

Tabla 1.2.15 Dominio de los atributos de otros materiales

Atributo	Tipo de dato	Longitud	Dominio	Operaciones
Idl	Carácter	10	{CCCCCCCCCC}	Identificación de laboratorio
Nombre de laboratorio	Carácter	20	{CCCCCCCCCCCCCCCCCCCC}	Nombre del laboratorio
Ids	Carácter	10	{CCCCCCCCCC}	Tipo de sustancia que maneja El laboratorio

Tabla 1.2.16 Dominio de los atributos de laboratorio

Atributo	Tipo de dato	Longitud	Dominio	Operaciones
IDP	carácter	10	{CCCCCCCCCC}	ID del profesor
Nombre	Carácter	15	{CCCCCCCCCCCCCCCC}	Nombre completo del profesor con apellidos
Edad	Numérico	3	{999}	Edad
Dirección	Carácter	35	{CCCCCCCCCCCCCCCCCCCCCCCCCCCC}	Dirección completa del profesor
Lugar de nacimiento	Carácter	25	{CCCCCCCCCCCCCCCCCCCCCCCC}	Lugar de nacimiento el profesor

Ciudad	Carácter	25	{CCCCCCCCCCCCCCCCCCCCCCCC}	Ciudad en donde vive
Estado	Carácter	25	{CCCCCCCCCCCCCCCCCCCCCCCC}	Estado donde vive
Telefono	Carácter	25	{CCCCCCCCCCCCCCCCCCCCCCCC}	Telefono donde se pueda localizar
Especialidad	Carácter	25	{CCCCCCCCCCCCCCCCCCCCCCCC}	Especialidad con la que se graduo
Grado	Carácter	15	{CCCCCCCCCCCCCCCC}	Grado de estudios
Experiencia	memo	4	{CCCC}	Experiencia de su area
Area	Carácter	15	{CCCCCCCCCCCCCCCC}	Area la que se especializa
Fecha	Carácter	10	{99/99/9999}	Fecha en cuando fue dado de alta
Comentarios	carácter	4	{CCCC}	Comentarios

Tabla 1.2.17 Dominio de los atributos de personal

Atributo	Tipo de dato	Longitud	Dominio	Operaciones
Nrc	Carácter	10	{CCCCCCCC}	Nombre de la materia
Materia	Carácter	20	{CCCCCCCCCCCCCCCCCCCC}	Código del contenido de la materia
Periodo	Carácter	10	{CCCCCCCC}	Periodo de trabajo
Idl	Carácter	10	{CCCCCCCC}	Que laboratorio debe aplicarse
Ids	Carácter	10	{CCCCCCCC}	Que sustancia debe aplicarse
Idc	Carácter	10	{CCCCCCCC}	Que cristaleria debe aplicarse
Idi	Carácter	10	{CCCCCCCC}	Que instrumental debe aplicarse
Idom	Carácter	10	{CCCCCCCC}	Que otros materiales debe aplicarse

Tabla 1.2.18 Dominio de los atributos asignatura

Atributo	Tipo de dato	Longitud	Dominio	Operaciones
Matricula	Número	10	{9999999999}	Control del alumno
Nombre	Carácter	25	{CCCCCCCCCCCCCCCCCCCC}	Nombre del alumno
Sección	Número	4	{9999}	Seccion en la que esta inscrito.
Fecha	Carácter	10	{99/99/9999}	Fecha de prestamo
Asignación	Carácter	15	{CCCCCCCCCCCCCCCC}	Quien asigno
Ids	Carácter	10	{CCCCCCCC}	Que sustancia debe aplicarse
Idc	Carácter	10	{CCCCCCCC}	Que cristaleria debe aplicarse
Idi	Carácter	10	{CCCCCCCC}	Que instrumental debe aplicarse
Idom	Carácter	10	{CCCCCCCC}	Que otros materiales debe aplicarse
observaciones	Carácter	4	{CCCC}	Escribir todo lo referente al alumno.

Tabla 1.2.19 Dominio de los atributos de deudores

Atributo	Tipo de dato	Longitud	Dominio	Operaciones
idpprog	carácter	10	{CCCCCCCC}	Identificación de practicas programadas
Idc	Carácter	10	{CCCCCCCC}	Identificación de cristalería
Ubc	Carácter	8	{CCCCCCC}	Unidades básicas de cristaleria
Idi	Carácter	10	{CCCCCCCC}	Identificación de instrumental
Ubi	Carácter	8	{CCCCCCC}	Unidades básicas de instrumental
Idom	Carácter	10	{CCCCCCCC}	Identificación de otros materiales
Udom	Carácter	8	{CCCCCCC}	Unidades básicas de otros materiales
Ids	Carácter	10	{CCCCCCCC}	Identificación de sustancias
Ub	Carácter	8	{CCCCCCC}	Unidades básicas de sustancias
Materia	Carácter	20	{CCCCCCCCCCCCCCCCCCCC}	Nombre de materia
Nrc	carácter	10	{CCCCCCCC}	Identificación de materia
Entrega	Carácter	8	{CCCCCCC}	Se hizo la practica o no

Tabla 1.2.20 Dominio de los atributos de practicas programadas.

Atributo	Tipo de dato	longitud	Dominio	Operaciones
Idpprof	Carácter	10	{CCCCCCCC}	Identificación de practicas profesores
Idp	Carácter	10	{CCCCCCCC}	Identificación del profesor
Idpprog	Carácter	10	{CCCCCCCC}	Identificación de practicas programadas
Idl	Carácter	10	{CCCCCCCC}	Identificación de laboratorio
No. De equipos	Número	2	{99}	No de equipos que aran la practica
Materia	Carácter	20	{CCCCCCCCCCCCCCCCCCCC}	Nombre de la Materia de la practica
Nrc	carácter	10	{CCCCCCCC}	Identificación de la materia

Fecha	Carácter	8	{CCCCCCC}	Fecha de programación de la materia
-------	----------	---	-----------	-------------------------------------

Tabla 1.2.21 Dominio de los atributos de practicas profesores

Atributo	Tipo de dato	longitud	Dominio	Operaciones
Nombre de usuario	carácter	35	{CCCCCCCCCCCCCCCCCCCCCCCC}	Nombre del usuario
Contraseña	Carácter	10	{CCCCCCCC}	Contraseña del usuario
Confirmar contraseña	Carácter	10	{CCCCCCCC}	Repetir la contraseña del usuario
Nivel de usuario	Carácter	25	{CCCCCCCCCCCCCCCCCCCCCCCC}	Elegir el nivel de usuario

Tabla 1.2.22 Dominio de los atributos de usuario