

**ESCUELA POLITÉCNICA DEL EJÉRCITO**

**DPTO. DE CIENCIAS DE LA COMPUTACIÓN**

**CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

**TÍTULO DEL PROYECTO**

**DESARROLLO DEL CURSO B-LEARNING DE**

**BASE DE DATOS I**

**Previa a la obtención del Título de:**

**INGENIERO DE SISTEMAS E INFORMÁTICA**

**POR: PAOLA ALEXANDRA COELLO BAQUERO**

**SANGOLQUÍ, Julio 2012**

## **CERTIFICACIÓN**

Certifico que el presente trabajo fue realizado en su totalidad por la Srta. PAOLA ALEXANDRA COELLO BAQUERO como requerimiento parcial a la obtención del título de INGENIERO EN SISTEMAS E INFORMÁTICA.

Sangolquí, Julio del 2012

---

ING. PAÚL DÍAZ  
DIRECTOR

## **DEDICATORIA**

Mi tesis la dedico:

A DIOS por concederme la vida, por mi familia, por la fortaleza que me ha brindado para cumplir mis objetivos.

A mis padres por su amor, comprensión por estar junto a mí en todo momento. Por enseñarme que en la vida no hay que darse por vencido ante cualquier obstáculo, por su cariño y apoyo incondicional. Además por confiar en mí siempre.

**PAOLA ALEXANDRA COELLO BAQUERO**

## **AGRADECIMIENTO**

Gracias a Dios por la vida y por concederme todo lo que he logrado.

A mi familia por su cariño y comprensión, por su total apoyo, a mis padres Freddy y Fanny, al amor de mi vida Carlos y mi hermana Dianita.

A mis queridos profesores Director y Codirector de tesis quienes me ayudaron a terminar este proyecto con éxito.

A mis compañeros de carrera que me han brindado su apoyo para lograr este objetivo tan importante en la vida universitaria.

**PAOLA ALEXANDRA COELLO BAQUERO**

## ÍNDICE DE CONTENIDOS

CAPÍTULO 1 .....	2
GENERALIDADES .....	2
1.1. PRESENTACIÓN .....	2
1.2. ANTECEDENTES .....	2
1.4. INTRODUCCIÓN .....	4
1.5. OBJETIVOS .....	5
1.5.1. Objetivo General .....	5
1.5.2. Objetivos Específicos .....	5
1.6. ALCANCE .....	5
1.7. LIMITACIONES .....	6
CAPÍTULO 2 .....	7
MARCO TEÓRICO .....	7
2.1. INTRODUCCIÓN .....	7
2.2. E-Learning .....	7
2.3. B-Learning .....	8
2.3.1. Diferencias entre B-Learning y E-Learning .....	8
2.4. Tipos de E-Learning .....	8
2.4.1. CBT (Computer – Based Training) .....	9
2.4.2. WBT (Web – Based Training) .....	9
2.5. Sistemas de Gestión de Cursos MOODLE .....	9
2.6. Modelo de Aprendizaje .....	12
2.6.1. Conductivismo .....	12
2.6.1.2. Factores que influyen en el aprendizaje .....	13
2.6.1.3. El rol de la memoria .....	13
2.6.1.4. Cómo ocurre la transferencia .....	13
2.6.2. Cognitivismo .....	13
2.6.2.1. Cómo ocurre el aprendizaje .....	13
2.6.2.2. Cuáles factores influyen en el aprendizaje .....	14
2.6.2.3. Cuál es el papel de la memoria .....	14
2.6.2.4. Cómo ocurre la transferencia .....	14
2.6.3. Constructivismo .....	14
2.6.3.1. Cómo ocurre el aprendizaje .....	15
2.6.3.2. Factores influyen en el aprendizaje? .....	15

2.6.3.3.	Papel de la memoria.....	15
2.6.3.4.	Cómo ocurre la transferencia.....	15
2.7.	Metodologías de Desarrollo de Objetos de Aprendizaje Basado en Competencias .....	16
2.7.1.	Competencias.....	16
2.7.2.	Objetos de Aprendizaje .....	17
2.8.	Categorización o niveles de los objetos de aprendizaje .....	19
2.9.	Metodologías para el Diseño del Curso.....	20
2.10.	Metodología OOHDM .....	20
2.11.	Metodologías para el Desarrollo de Contenidos .....	21
2.11.1.	Método Ergoglífico.....	21
2.11.2.	Especificación de Objetos de Aprendizaje.....	23
2.11.3.	Especificación de Framework .....	23
2.11.4.	Lista de Requerimientos Tecnológicos .....	23
2.11.4.1.	Diseño del Flujo de Trabajo .....	23
2.11.4.2.	Establecimiento de Estándares Internos de Interfaces.....	24
CAPÍTULO 3 .....		25
DESARROLLO E IMPLEMENTACION .....		25
3.1.	Competencias.....	25
3.2.	Objetos de Aprendizaje .....	27
3.3.	Diseño Instruccional (Aplicación del Modelo ADDIE) .....	38
3.3.1.	Análisis Instruccional .....	39
3.3.1.1.	Características de la audiencia.....	39
3.3.1.2.	Identificación de las necesidades de conocimientos .....	39
3.3.1.3.	El presupuesto requerido y disponible.....	40
3.3.1.4.	Los medios de difusión .....	41
3.3.1.5.	Limitaciones.....	41
3.3.1.6.	Actividades de aprendizaje que necesitan hacer los estudiantes para el logro de competencias.....	41
3.3.2.	Diseño Instruccional .....	42
3.3.2.1.	Selección del mejor ambiente.....	42
3.3.2.2.	Señalamiento de los objetivos instruccionales.....	42
3.3.2.3.	Selección de estrategias pedagógica .....	42
3.3.2.4.	Bosquejo de unidades, lecciones y módulos .....	42

3.3.2.5.	Diseño del contenido del curso, teniendo en cuenta los medios interactivos electrónicos .....	43
3.3.3.	Desarrollo Instruccional .....	43
3.3.3.1.	Se selecciona, obtiene o crea el medio requerido .....	43
3.3.3.2.	Se determinan las interacciones apropiadas .....	43
3.3.3.3.	Planificación de actividades.....	43
3.3.4.	Implantación e Implementación Instruccional .....	44
3.3.4.1.	Duplican y distribuyen los materiales.....	44
3.3.4.2.	Implanta e implementa el curso .....	44
3.3.4.3.	Resuelve problemas técnicos y se discuten planes alternos .....	44
3.3.5.	Evaluación Instruccional .....	45
3.4.	Metodología OOHDM .....	45
3.4.1.	Diseño Conceptual .....	45
3.4.2.	Mapas de Navegación .....	45
3.4.3.	Diseño de interfaces abstractas.....	46
3.4.4.	Implementación .....	47
3.4.5.	Casos de Uso .....	47
3.4.6.	Diagramas de Secuencia.....	77
3.4.7.	Diagramas de Clases Navegacionales .....	77
3.4.8.	Desarrollo del Curso B-Learning de Base de Datos I con el Método Ergoglífico 80	
3.4.8.1.	Análisis Ergoglífico .....	80
3.4.8.2.	Conocimiento de los Usuarios .....	81
3.4.8.3.	Restricciones Generales.....	82
3.4.8.4.	Etapa de Diseño .....	83
CAPÍTULO 4 .....		95
CONTENIDO DEL CURSO DE BASE DE DATOS 1 .....		95
Estructura del Curso.....		96
DESARROLLO DEL CURSO .....		97
UNIDAD 1: GENERALIDADES DE LAS BASES DE DATOS .....		97
LECCIÓN 1.1: Base de Datos .....		97
LECCIÓN 1.2: Sistemas de Gestión de Base de Datos .....		106
LECCIÓN 1.3: Modelos de Datos.....		122
LECCIÓN 1.4: Herramientas Case.....		136

UNIDAD 2: PROCESO DE DESARROLLO DE LA BASE DE DATOS .....	145
LECCIÓN 2.1: Modelo Entidad - Relación.....	145
LECCIÓN 2.2: Modelo Relacional .....	171
LECCIÓN 2.3: Lenguajes Relaciones .....	190
LECCION 2.4: Creación y Desarrollo de Bases de Datos .....	205
LECCIÓN 2.5: Diseño Conceptual, Diseño Lógico y Diseño Físico .....	217
LECCION 2.6: Aplicación de una Herramienta Case .....	233
LECCIÓN 2.7: Normalización.....	249
LECCIÓN 2.8: Diccionario de Datos .....	266
LECCIÓN 2.9: Implementación de la base de datos utilizando SQL – DDL...	276
UNIDAD 3: SQL LENGUAJE DE CONSULTA RELACIONAL.....	288
LECCIÓN 3.1: SQL-DML.....	288
LECCIÓN 3.2: Consultas sobre tablas y tablas relacionadas.....	298
LECCIÓN 3.3: Funciones de Agregación .....	314
CAPÍTULO 5 .....	324
CONCLUSIONES.....	324
RECOMENDACIONES.....	325
BIBLIOGRAFÍA.....	326

## LISTADO DE TABLAS

<b>TABLA 2.1:</b> CUADRO COMPARATIVO DE PLATAFORMAS.....	11
<b>TABLA 3.1:</b> ESPECIFICACIÓN CASO DE USO – INGRESAR A LA PLATAFORMA.....	48
<b>TABLA 3.2:</b> ESPECIFICACIÓN CASO DE USO – AGREGAR FORO.....	49
<b>TABLA 3.3:</b> ESPECIFICACIÓN CASO DE USO – ACTUALIZAR FORO.....	50
<b>TABLA 3.4:</b> ESPECIFICACIÓN CASO DE USO – DESARROLLAR FORO.....	51
<b>TABLA 3.5:</b> ESPECIFICACIÓN CASO DE USO – ACCIÓN EDITAR FORO.....	51
<b>TABLA 3.6:</b> ESPECIFICACIÓN CASO DE USO – ESCOGER CURSO.....	52
<b>TABLA 3.7:</b> ESPECIFICACIÓN CASO DE USO – CREAR CLASE O TIPO DE TAREA.....	53
<b>TABLA 3.8:</b> ESPECIFICACIÓN CASO DE USO – ACTUALIZAR TAREA.....	53
<b>TABLA 3.9:</b> ESPECIFICACIÓN CASO DE USO – CALIFICAR TAREA.....	54
<b>TABLA 3.10:</b> ESPECIFICACIÓN CASO DE USO – AGREGAR GLOSARIO.....	55
<b>TABLA 3.11:</b> ESPECIFICACIÓN CASO DE USO – AGREGAR ENTRADA.....	56
<b>TABLA 3.12:</b> INCORPORACIÓN DE IMÁGENES EN GLOSARIO.....	57
<b>TABLA 3.13:</b> ESPECIFICACIÓN CASO DE USO – ACTUALIZAR GLOSARIO.....	58
<b>TABLA 3.14:</b> ESPECIFICACIÓN CASO DE USO – ELIMINAR GLOSARIO.....	58
<b>TABLA 3.15:</b> ESPECIFICACIÓN CASO DE USO – CONSULTAR GLOSARIO.....	59
<b>TABLA 3.16:</b> ESPECIFICACIÓN CASO DE USO – AGREGAR LECCIÓN.....	60
<b>TABLA 3.17:</b> ESPECIFICACIÓN CASO DE USO – MODIFICAR LECCIÓN.....	61
<b>TABLA 3.18:</b> ESPECIFICACIÓN CASO DE USO – ELIMINAR LECCIÓN.....	61

<b>TABLA 3.19:</b> ESPECIFICACIÓN CASO DE USO – INGRESAR A LECCIÓN.....	62
<b>TABLA 3.20:</b> ESPECIFICACIÓN CASO DE USO – AGREGAR TALLER.....	63
<b>TABLA 3.21:</b> ESPECIFICACIÓN CASO DE USO – ACTUALIZAR TALLER.....	64
<b>TABLA 3.22:</b> ESPECIFICACIÓN CASO DE USO – MOVER DERECHA / IZQUIERDA TALLER.....	65
<b>TABLA 3.23:</b> ESPECIFICACIÓN CASO DE USO – MOVER TALLER.....	65
<b>TABLA 3.24:</b> ESPECIFICACIÓN CASO DE USO – BORRAR TALLER.....	66
<b>TABLA 3.25:</b> ESPECIFICACIÓN CASO DE USO - OCULTAR TALLER.....	66
<b>TABLA 3.26:</b> ESPECIFICACIÓN CASO DE USO - MOSTRAR TALLER.....	67
<b>TABLA 3.27:</b> ESPECIFICACIÓN CASO DE USO – DESARROLLAR TALLER.....	68
<b>TABLA 3.28:</b> ESPECIFICACIÓN CASO DE USO – EDITAR TALLER.....	68
<b>TABLA 3.29:</b> ESPECIFICACIÓN CASO DE USO – BORRAR TALLER.....	69
<b>TABLA 3.30:</b> ESPECIFICACIÓN CASO DE USO – CREAR PREGUNTA.....	70
<b>TABLA 3.31:</b> ESPECIFICACIÓN CASO DE USO – EDITAR PREGUNTA.....	70
<b>TABLA 3.32:</b> ESPECIFICACIÓN CASO DE USO – BORRAR PREGUNTA.....	71
<b>TABLA 3.33:</b> ESPECIFICACIÓN CASO DE USO – AÑADIR PREGUNTA AL CUESTIONARIO.....	72
<b>TABLA 3.34:</b> ESPECIFICACIÓN CASO DE USO – AGREGAR CUESTIONARIO.....	72
<b>TABLA 3.35:</b> ESPECIFICACIÓN CASO DE USO – MODIFICAR CUESTIONARIO.....	73

<b>TABLA 3.36:</b> ESPECIFICACIÓN CASO DE USO – BORRAR CUESTIONARIO.....	73
<b>TABLA 3.37:</b> ESPECIFICACIÓN CASO DE USO – MOVER CUESTIONARIO.....	74
<b>TABLA 3.38:</b> ESPECIFICACIÓN CASO DE USO – OCULTAR CUESTIONARIO.....	74
<b>TABLA 3.39:</b> ESPECIFICACIÓN CASO DE USO – MOSTRAR CUESTIONARIO.....	75
<b>TABLA 3.40:</b> ESPECIFICACIÓN CASO DE USO – RESPONDER CUESTIONARIO.....	76
<b>TABLA 3.41:</b> ESPECIFICACIÓN CASO DE USO – CALIFICAR CUESTIONARIO.....	76
<b>TABLA 3.42:</b> ESPECIFICACIÓN DEL FRAMEWORK.....	85
<b>TABLA 3.43:</b> REQUISITOS DE HARDWARE Y SOFTWARE.....	86
<b>TABLA 3.44:</b> NECESIDADES DE LOS PARTICIPANTES.....	87
<b>TABLA 3.45:</b> PALETA DE COLORES.....	89
<b>TABLA 3.46:</b> TIPOGRAFÍA.....	90

## LISTADO DE FIGURAS

<b>FIGURA 2.1:</b> CONDUCTIVISMO.....	12
<b>FIGURA 2.2:</b> CONGNITIVISMO.....	14
<b>FIGURA 2.3:</b> CONSTRUCTIVISMO.....	16
<b>FIGURA 2.4:</b> ESTRUCTURA DE INTEGRACION DE OBJETOS DE APRENDIZAJE.....	17
<b>FIGURA 2.5:</b> ESTRUCTURA DE OBJETOS DE APRENDIZAJE.....	18
<b>FIGURA: 3.1:</b> MAPA NAVEGACIONAL DEL CURSO.....	46
<b>FIGURA 3.2:</b> CASO DE USO: INGRESAR A LA PLATAFORMA.....	47
<b>FIGURA 3.3:</b> CASO DE USO: AGREGAR FORO.....	48
<b>FIGURA 3.4:</b> CASO DE USO: ACTUALIZAR FORO.....	49
<b>FIGURA 3.5:</b> CASO DE USO: DESARROLLAR EL FORO.....	50
<b>FIGURA 3.6:</b> CASOS DE USO TAREA.....	52
<b>FIGURA 3.7:</b> CASO DE USO: AGREGAR GLOSARIO.....	54
<b>FIGURA 3.8:</b> CASO DE USO: AGREGAR ENTRADA.....	56
<b>FIGURA 3.9:</b> CASO DE USO: ACTUALIZAR GLOSARIO.....	57
<b>FIGURA 3.10:</b> CASO DE USO: AGREGAR LECCIÓN.....	59
<b>FIGURA 3.11:</b> CASO DE USO: MODIFICAR LECCIÓN.....	60
<b>FIGURA 3.12:</b> CASO DE USO: INGRESAR A LECCIÓN.....	62
<b>FIGURA 3.13:</b> CASO DE USO: AGREGAR TALLER.....	63
<b>FIGURA 3.14:</b> CASO DE USO: ACTUALIZAR TALLER.....	64
<b>FIGURA 3.15:</b> CASO DE USO: DESARROLLAR TALLER.....	67
<b>FIGURA 3.16:</b> CASO DE USO: AÑADIR PREGUNTAS AL CUESTIONARIO.....	69
<b>FIGURA 3.17:</b> CASO DE USO: CUESTIONARIO (EVALUACIÓN.....	71

<b>FIGURA 3.18:</b> CASO DE USO: RESPONDER CUESTIONARIO.....	75
<b>FIGURA 3.19:</b> EJEMPLO DIAGRAMAS DE SECUENCIA UNIDAD I.....	77
<b>FIGURA 3.20:</b> DIAGRAMA DE CLASES NAVEGACIONALES.....	78
<b>FIGURA 3.21:</b> DIAGRAMA NAVEGACIONAL DEL DESARROLLO DE LA LECCIÓN.....	79
<b>FIGURA 3.22:</b> DIAGRAMA DE LA FASE DE ANÁLISIS.....	81
<b>FIGURA 3.23:</b> DIAGRAMA DE DISEÑO.....	83
<b>FIGURA 3.24:</b> DIAGRAMA DE DISEÑO.....	91
<b>FIGURA 3.25:</b> PLANTILLA DE CONTENIDO DE CURSO.....	92



## RESUMEN

El presente proyecto consiste en el análisis, diseño e implementación de un curso B-Learning de “Base de Datos I”. El objetivo del curso es instruir a los aprendientes en los conceptos y prácticas que contribuyan en el modelamiento de la Base de Datos de una organización. Se parte definiendo las competencias específicas y generales de un Ingeniero de Sistemas e Informática. Las competencias son posteriormente traducidas en Objetos de Aprendizaje. La gestión de contenidos fueron organizados de acuerdo a metodologías basadas en el Diseño Instrucciones y Paradigmas de Enseñanza como el conductismo, cognitivismo, constructivismo.

Para el diseño e implementación del curso se recurrió a la metodología OOHDM y el método Ergoglífico del cual se consideró lo concerniente al diseño de contenidos para productos educativos, mas no la parte de desarrollo de software, porque ya se encuentra definido en la plataforma Moodle. El modelo ADDIE para el Análisis, Diseño, Desarrollo, Implementación y Evaluación.

El curso está formado por tres unidades: 1. Generalidades de Bases de Datos 2. Proceso de Desarrollo de la Base de Datos, 3. SQL Lenguaje de Consulta Relacional Estructurado, subdivididas en lecciones, donde cada unidad contiene una breve introducción, una parte teórica y las correspondientes actividades de aprendizaje para reforzar los conocimientos. En la plataforma Moodle del Departamento de Ciencias de la Computación se encuentra levantado el curso para su explotación.

# **CAPÍTULO 1**

## **GENERALIDADES**

### **1.1. PRESENTACIÓN**

La educación virtual representa para las empresas la respuesta a las necesidades de educación continua, al ser un método flexible, donde la formación de sus empleados no interfiere en el proceso de producción, en este contexto como respuesta a esas necesidades, el Departamento de Ciencias de la Computación de la Escuela Politécnica del Ejército, plantea el desarrollo de cursos virtuales, entre ellos para la asignatura de Base de Datos I.

El curso virtual de Base de Datos I está basado en la aplicación de metodologías como: Diseño Instruccional, Ergoglífico, OOHDM y objetos de aprendizaje basado en las competencias que el Ingeniero debe adquirir en su carrera.

Mediante este curso los aprendientes van a analizar y revisar los temas de la materia a través de sus conocimientos y profundizar con actividades de aprendizaje como: tareas, cuestionarios, autoevaluaciones, foros; a éstos se puede ingresar mediante la plataforma virtual Moodle del Departamento de Ciencias de la Computación o acudir al material que se proporcionará en el CD.

### **1.2. ANTECEDENTES**

Las ocupaciones de los estudiantes universitarios determinan la necesidad de contar con cursos virtuales que optimicen el tiempo y el espacio para su formación. Por otro lado varias de las universidades y de hecho también la ESPE, han adoptado el paradigma de las competencias del futuro profesional para el diseño curricular.

En el Departamento de Ciencias de la Computación, se han desarrollado varias tesis que han permitido definir las plataformas, metodologías y estándares para los cursos virtuales tomado en cuenta fundamentalmente las características de las asignaturas que componen la currícula, destacándose:

La constante actualización de los contenidos debido al acelerado desarrollo tecnológico y el alto contenido práctico, por lo que se está desarrollando software educativo sobre las asignaturas de la Carrera de Ingeniería en Sistemas e Informática, se han levantado estos cursos en la plataforma Moodle que permitirá el acceso al material educativo tanto para la modalidad virtual como apoyo a la modalidad presencial.

La asignatura de Base de Datos I es una de las materias que se desarrolla para cumplir el plan operativo anual del Departamento de Ciencias de la Computación con miras a implementar la modalidad virtual en la carrera de Ingeniería de Sistema e Informática.

### **1.3. JUSTIFICACIÓN E IMPORTANCIA**

El Departamento de Ciencias de Computación de la Escuela Politécnica del Ejército, como parte de la formación a los estudiantes, imparte la asignatura de Base de Datos I en la Carrera de Ingeniería de Sistemas e Informática. Esta materia se dicta en la modalidad presencial debido a la complejidad de los contenidos y cuando no existe el material de apoyo suficiente que permita la formación en línea de los estudiantes.

El desarrollo de este proyecto, permitirá obtener los conocimientos necesarios sobre la asignatura bajo una modalidad no presencial y además podrá utilizarse como apoyo por los maestros y alumnos en la modalidad presencial. Este curso lo desarrollo con el auspicio del Departamento de Ciencias de la Computación y los medios tecnológicos necesarios disponibles en el salón tecnológico.

#### 1.4. INTRODUCCIÓN

En los últimos años se ha visto la transformación de las tecnologías Web y el uso de Internet, lo que permite mejoras en el campo educativo. Se conoce como *E-Learning* o *Aprendizaje Electrónico* a la herramienta formativa desde el punto de vista tecnológico que permite tener mejores oportunidades en el campo educativo. Existiendo también la modalidad *B-Learning* que consiste en el proceso de educación semipresencial; que incluye actividades de E-Learning.

La tecnología adecuadamente aplicada fomenta procesos de aprendizaje en los cuales se utilizan elementos tales como el diseño de aprendizaje y el análisis de factores sociales y culturales. De ahí que, si no tenemos muy clara esta visión de los sistemas de aprendizaje, no se aprovechará todo el potencial de la tecnología.

En la actualidad innovar en el área educativa es esencial y fundamental, por esta razón el B-Learning o formación mixta es una propuesta de aprendizaje basada en la utilización de recursos pedagógicos desarrollados a partir de nuevas tecnologías.

El presente proyecto se enfoca en el análisis y utilización de estas metodologías como un medio de enseñanza y pretende proporcionar a los participantes un intercambio dinámico que conjuntamente con la voluntad y compromiso que exige este tipo de educación, permita obtener un aprendizaje de calidad.

El presente proyecto de tesis plantea el diseño y elaboración de un CURSO B-LEARNING DE BASE DE DATOS I, asignatura impartida actualmente en la carrera de Ingeniería en Sistemas e Informática de la Escuela Politécnica del Ejército. El curso es E-Learning ya que el alumno accede a la información del mismo mediante Internet y realiza actividades de aprendizaje (tareas, cuestionario, foros) y es B-Learning ya que las prácticas e instalaciones de Bases de Datos, Herramientas Case se las deberá realizar en forma presencial.

## **1.5. OBJETIVOS**

### **1.5.1. Objetivo General**

Desarrollar e implementar el curso virtual de Base de Datos I en la plataforma Moodle del Departamento de Ciencias de la Computación.

### **1.5.2. Objetivos Específicos**

- Crear formatos para el diseño del curso basados en el método Ergoglífico y la metodología OOHDM.
- Revisar metodologías que se basan en el diseño instruccional como el Modelo Addie.
- Definir las competencias generales y específicas para Ingenieros.
- Desarrollar el Curso de Base de Datos basándose en objetos de aprendizaje.
- Generar un archivo magnético el cual contendrá el material del curso para los aprendientes que por no disponer un adecuado servicio de Internet no acceden en línea al curso.
- Levantar el curso en la plataforma Moodle del Departamento de Ciencias de la Computación.

## **1.6. ALCANCE**

El proyecto se logró siguiendo los métodos pedagógicos para E-Learning, tales como las actividades de aprendizaje. Las lecciones están conformadas por módulos de contenido, incluyen bloques de texto y gráficas.

Se terminará con la elaboración del curso en modalidad E-Learning y B-Learning en cuanto sea necesario. Los contenidos del curso corresponden al plan analítico de la materia dictada en la actualidad en la Carrera de Ingeniería de Sistemas e Informática de la Escuela Politécnica del Ejército.

Los contenidos corresponden al siguiente temario:

1. Generalidades de Bases de Datos
  - 1.1. Base de datos: Dato, Información, Conocimiento, definición de base de datos
  - 1.2. Sistemas de Gestión de Bases de Datos
  - 1.3. Modelos de Datos
  - 1.4. Herramientas Case
  
2. Proceso de desarrollo de la Base de Datos
  - 2.1. Modelo Entidad – Relación
  - 2.2. Modelo Relacional
  - 2.3. Lenguajes Relacionales
  - 2.4. Creación y Desarrollo de una Base de Datos
  - 2.5. Diseño Conceptual, Lógico y Físico
  - 2.6. Aplicación de una Herramienta Case
  - 2.7. Normalización
  - 2.8. Diccionario de Datos
  - 2.9. Implementación de una Base de Datos utilizando SQL – DDL
  
3. SQL Lenguaje de Consulta Relacional Estructurado
  - 3.1. SQL-DML
  - 3.2. Consultas sobre tablas y tablas relacionadas
  - 3.3. Funciones de Agregación

#### **1.7. LIMITACIONES**

- La responsabilidad de la administración de la asignatura de Base de Datos a nivel del sitio es del profesor titular.
- El material del curso se encuentra en un CD que no constituye un curso interactivo. La finalidad es que el estudiante disponga del material cuando no disponga de un adecuado servicio de Internet.

## CAPÍTULO 2

### MARCO TEÓRICO

#### 2.1. INTRODUCCIÓN

El Curso E-Learning de Base de Datos I, se enmarca en un proyecto más amplio de aplicación de las Tecnologías de Información y Comunicaciones, aplicando Tecnología Educativa, que se lleva a cabo en el Salón Tecnológico del Departamento de Ciencias de la Computación. Con la finalidad de no duplicar esfuerzos aprovechando las tesis anteriores, dando énfasis en las nuevas tesis a los contenidos de las asignaturas.

La presente tesis por ser parte de un gran proyecto se implementará en la plataforma Moodle, la misma que en un exhaustivo estudio previo fue definida para el Departamento de Ciencias de la Computación como la plataforma adecuada. (Tesis por Cecilia Rivera y Elena Garcés – año 2006).

La metodología OOHDM y como referencia el método Ergoglífico, estos temas fueron desarrollados anteriormente en una tesis (por Andrea Hermosa y Fernanda Chica – año 2008), en lo referente a los formatos, Plantillas, Objetos de Aprendizaje basados en competencias y la metodología de desarrollo del curso virtual, el presente proyecto se fundamenta en la tesis de Magíster en Sistemas Informáticos Educativos del Ing. Edgar Hermosa Mena, de manera que únicamente se hará un estudio rápido para comprender la utilización de las metodologías y métodos (año 2009). Para el caso del Diseño Instruccional el sistema se basa en la recopilación y síntesis de información de tesis anteriores.

#### 2.2. E-Learning

El E-Learning o Aprendizaje Electrónico se define de acuerdo a tres aspectos: *técnico*, *educativo* y *descriptivo*. La provisión de material educativo a través de medios electrónicos, el uso para los educadores de la tecnología para facilitar el aprendizaje y uso de elementos como audio, video, ofimática por medio de Internet, indican para que sirven cada uno de los aspectos respectivamente.

Posee ventajas como: mayor productividad, entrega oportuna, capacitación flexible, ahorros en costos por cada participante.

### **2.3. B-Learning**

El término B-Learning es la abreviación de Blended Learning que dentro de la enseñanza virtual es la Formación Combinada. Se refiere a la modalidad semipresencial de estudios que contiene orden presencial y no presencial. Este modelo de aprendizaje ya se lo usa por poseer ventajas (aulas virtuales, Internet) y permite tener un instructor tutor por cada curso. De igual manera es necesario cuando se requieren equipos de laboratorio que individualmente el estudiante no está en condiciones de disponer.

#### **2.3.1. Diferencias entre B-Learning y E-Learning**

En el aprendizaje a distancia tiene mucha importancia el E-Learning, a diferencia del sistema tradicional que usa libros, éste modelo aprovecha recursos de la informática y el Internet, para que el participante obtenga herramientas didácticas que permiten que los cursos sean dinámicos.

El instructor en E-Learning es un tutor on-line, la diferencia radica en que para realizar todas las actividades se lo hace por a través de Internet, por mensajería instantánea, correo electrónico o videoconferencias. Mientras que en B-Learning el instructor asume la enseñanza tradicional pero utiliza material didáctico que se encuentra en Internet para desarrollar sus labores en dos pasos: tutor on-line (tutorías a distancia) y como formador tradicional (enseñanza presencial). El tutor lo que hace es combinar los dos formas según los requerimientos de cada curso.

### **2.4. Tipos de E-Learning**

Los tipos de E-Learning son los siguientes:

#### **2.4.1. CBT (Computer – Based Training)**

Es un prototipo de la educación muy efectiva, ya que los aprendientes obtienen información en el momento que se lo requiera, sin que interese el lugar donde el alumno se encuentre y pueda estudiar a su propio ritmo.

Posee una interfaz formada por imágenes, animaciones, texto, audio y video; con lo que el alumno se educa por medio de un contexto interactivo. Cada alumno realiza tareas a través del computador y en el caso de fallar en respuestas tendrá una retroalimentación. La evaluación del aprendizaje se lo hace por medio de preguntas de opción múltiple.

#### **2.4.2. WBT (Web – Based Training)**

Este tipo de E-Learning admite entregar la información a los alumnos por medio de sistemas de comunicación como Internet o una Intranet. Los dos modelos para proporcionar la instrucción basada en la web son: síncrono, que es dada por el facilitador y asíncrona que es auto dirigido. Estos modelos a la vez son formulado por la combinación de métodos estáticos (portales, audio, video) y métodos interactivos (foros, chats, etc).

### **2.5. Sistemas de Gestión de Cursos MOODLE**

La plataforma MOODLE fue analizada en un estudio realizado en el año 2006 mediante la Tesis de Grado “Evaluación de plataformas tecnológicas para la teleformación o E-Learning para el ámbito universitario”, el resultado obtenido fue que esta plataforma es la mejor y más conveniente para el Departamento de Ciencias de la Computación de la Escuela Politécnica del Ejército. MOODLE fue diseñado para facilitar a los educadores que desean implantar cursos online de calidad y como soporte a la modalidad de estudios presencial, Moodle es el sistema de gestión de aprendizaje LMS (Learning Management System) o contexto visual de enseñanza – aprendizaje, más utilizado a nivel universitario.

La plataforma contiene características como el constructivismo y una perspectiva construccionista social, en la que el alumno aporta a la experiencia educativa con lo que se analiza, investiga, construye y colabora.

Además admite administrar el sitio de usuarios y cursos, tiene un diseño modular que permite agregar actividades al proceso para el avance de un curso, las más utilizadas son: foros, tareas, cuestionarios, chats y glosario. El estudio que se realizó en la Tesis de Grado “Evaluación de plataformas tecnológicas para la teleformación o E-Learning para el ámbito universitario”, proporcionó el siguiente resultado:

“En comparación con E-educativa y analizando la viabilidad técnica y el cumplimiento de funcionalidades básicas, las plataformas Blackboard, WebCt, DotLRN y Moodle tienen la misma posibilidad de ser implementadas en la ESPE, sin embargo, debido a los costos, capacidad de personalización de código y a la aceptación en instituciones de educación superior a nivel mundial, se ha sentado en este estudio la superioridad de Moodle. Alto nivel de aceptación a nivel universitario, de una muestra a nivel mundial de más de 1200 universidades que utilizan una plataforma E-Learning, se observa que el 32.6% utilizan Moodle. Representa las ventajas de ser Open Source: bajos costos de implementación y personalización del código según las necesidades del centro educativo.

Se determina que la instalación y configuración de Moodle tanto en Windows como en distribuciones basadas en Linux tal como Red Hat 9 es simple y rápida, sin embargo, si se necesita una configuración más específica en cuanto a seguridades e integración con otros sistemas, se requiere un mayor nivel de conocimientos.”<sup>1</sup>. El siguiente es el cuadro comparativo de las plataformas virtuales analizadas, donde se destaca MOODLE:

---

<sup>1</sup> Garcés Elena y Rivera Enríquez Cecilia. (2006). *Evaluación de plataformas tecnológicas para la teleformación o e-learning para el ámbito universitario, tomando como caso de estudio e-educativa*. Quito: Escuela Politécnica del Ejército.

Tabla 2.1: Cuadro comparativo de las plataformas<sup>2</sup>

		MOODLE	BLACKBOARD	WEBCT	E-DUCATIVA	DOTLRN
Costo de Licencia		NO	67700	27300	5700 (licencia perpetua)	NO
Incluye Soporte Técnico		NO	24X7X365 vía telefónica y vía web	9 horas vía telefónica y vía Web	24X7X365 vía telefónica y vía web	X
CAPACITACIÓN	Total Capacitación	800	10000	NO	2800	X
	Tiempo	30h	3 días	NO	SI	X
	Nro.Doc	30	15	NO	SI	X
	Para Docentes	SI	SI	NO	SI	X
	Tiempo	20h	2 días	NO	SI	X
	No. Adm.	3	15	NO	SI	X
	Para Administradores	SI	SI	NO	SI	X
TOTAL \$		1000	77700	27300	20400	6000
El total incluye / no incluye		Incluye costo de instalación y personalización \$200. No incluye traslado ni alojamiento del instructor	No incluye viáticos del personal, hospedaje, alimentación, transporte local y aéreo. Costo de licencias de cualquier otro software.	Este valor no incluye servicios adicionales como instalación y configuración, capacitación de administrador, capacitación de docentes, diseño de contenidos virtuales, etc. Estos servicios pueden ser prestados por Latined según requerimiento.	Este valor incluye gastos de instalación y configuración \$4500, soporte y actualizaciones por un año \$5900, movilización y manutención del instructor por 5 días \$1500	Incluye costo de instalación y personalización \$200. No incluye traslado ni alojamiento del instructor
OPCIONAL	Hosting	\$369 /mensual	\$33375 /anual	\$75000 /anual	\$ 60 mensuales con 100MB para contenidos	\$100 /mensual
	Soporte Técnico	\$150 /mensual	X	X	\$5900 /anual	X

<sup>2</sup> Garcés Elena y Rivera Enríquez Cecilia. (2006). *Evaluación de plataformas tecnológicas para la teleformación o E-Learning para el ámbito universitario, tomando como caso de estudio e-educativa*. Quito: Escuela Politécnica del Ejército.

## 2.6. Modelo de Aprendizaje

Los modelos de aprendizaje ofrecen a los profesionales del área educativa estrategias, formas y técnicas de instrucción que hacen más fácil el aprendizaje. Las perspectivas del proceso de aprendizaje son: conductivismo, cognitvismo y constructivismo.

### 2.6.1. Conductivismo

Define el aprendizaje con los cambios de conducta visible, el aprendizaje en esta perspectiva se logra cuando el alumno demuestra una respuesta acertada frente a un estímulo. Se da importancia a las consecuencias de las conductas y las respuestas que se da con refuerzo son más probables a que ocurran posteriormente. Este método estudia los cambios en la conducta de una persona.

#### 2.6.1.1. Cómo ocurre el aprendizaje

El aprendizaje se alcanza cuando se presenta una respuesta adecuada luego de presentar un elemento específico. Esta teoría es importante por los efectos de las conductas y plantea que las respuestas en las que se continúa con refuerzo son probables que ocurran nuevamente a futuro. Además no se hace esfuerzo en conocer los procedimientos mentales que el estudiante necesita.

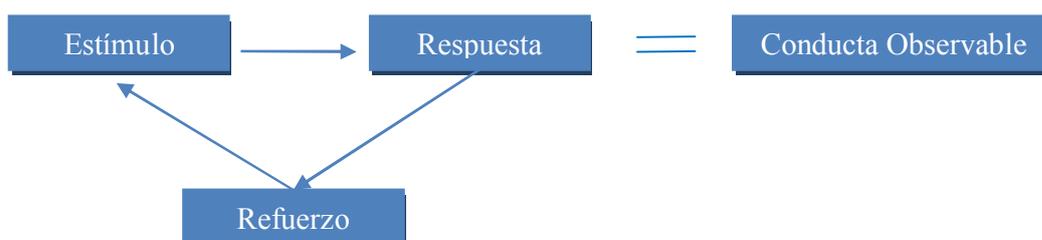


Figura 2.1: Conductivismo<sup>3</sup>

<sup>3</sup> [www.slideshare.net/wiesco/teorias-del-aprendizaje-312461](http://www.slideshare.net/wiesco/teorias-del-aprendizaje-312461)

### **2.6.1.2. Factores que influyen en el aprendizaje**

Se inicia con la evaluación a los estudiantes para verificar el inicio de la instrucción y para establecer los puntos en los que se debe reforzar.

### **2.6.1.3. El rol de la memoria**

El conductivismo no toma en cuenta a la memoria aunque se debe analizar el alcance de los “hábitos” y el olvido que se considera como la “falta de uso”. La práctica constante ayuda al estudiante a dar mejores respuestas.

### **2.6.1.4. Cómo ocurre la transferencia**

En el conductivismo la transferencia es el producto de generalizar las conductas se puede transferir por medio de mecanismos comunes por situaciones con características semejante o iguales.

## **2.6.2. Cognitivismo**

El cognitivismo aparece en la década de los años sesenta y se encarga de estudiar la mente humana como una técnica de conocimiento. Se fundamenta en el estudio del aprendizaje según la lógica de los procesos mentales, lo que significa cómo distinguirlos y cómo se organizan para lograr una conclusión.

“La teoría cognitivista enfatiza la adquisición de conocimientos y estructuras mentales internas; se dedican a la conceptualización de los procesos del estudiante y se ocupa de cómo la información es recibida, organizada y almacenada.”<sup>4</sup>

### **2.6.2.1. Cómo ocurre el aprendizaje**

En el cognitivismo se muestran los procesos de aprendizaje del estudiante y su objetivo es verificar cómo se recibe, organiza, almacena y se localiza la información. El aprendizaje se relaciona con lo que el estudiante sabe y cómo *adquiere el conocimiento* (se considera como una actividad mental).

---

<sup>4</sup> [http://www.slideshare.net/plazaa\\_10\\_10/cognitivismo](http://www.slideshare.net/plazaa_10_10/cognitivismo)

### 2.6.2.2. Cuáles factores influyen en el aprendizaje

El enfoque cognitivo se concentra en las actividades mentales del estudiante que conduce a una respuesta y reconocen los procesos de planificación mental. Se considera como principio básico del cognitismo a la retroalimentación que juega un papel importante, se la utiliza para guiar y apoyar las conexiones mentales exactas, entre los principios cognitivistas se incluye lo siguiente:

- Énfasis en la participación del estudiante en el proceso de aprendizaje.
- Creación de ambiente de aprendizaje que permita y estimule a los estudiantes hacer conexiones con material aprendido.

### 2.6.2.3. Cuál es el papel de la memoria

La información necesita ser organizada adecuadamente en la memoria para que el aprendizaje sea positivo.

### 2.6.2.4. Cómo ocurre la transferencia

“Cuando un estudiante entiende como aplicar el conocimiento en diferentes contextos, entonces ha ocurrido la transferencia.”



Figura 2.2: Cognitismo<sup>5</sup>

### 2.6.3. Constructivismo

“El constructivismo es una teoría que equipara al aprendizaje con la creación de significados a partir de experiencias.”<sup>6</sup>

---

<sup>5</sup>TeoríasdeAprendizaje.ppt/ Dilayla Terraza

<sup>6</sup>[http://ares.unimet.edu.ve/programacion/psfase3/modII/biblio/CONDUCTISMO\\_%20COGNITIVISMO\\_%20CONSTRUCTIVISMO.pdf](http://ares.unimet.edu.ve/programacion/psfase3/modII/biblio/CONDUCTISMO_%20COGNITIVISMO_%20CONSTRUCTIVISMO.pdf)

### **2.6.3.1. Cómo ocurre el aprendizaje**

El constructivismo es parte del cognitivismo su diferencia es que en el cognitivismo “la mente es una herramienta de referencia para el mundo real; los constructivistas creen que la mente filtra lo que nos llega del mundo para producir su propia y única realidad”.<sup>7</sup>

Los estudiantes no transmiten el conocimiento del exterior a su memoria sino establecen interpretaciones que se basan en experiencias anteriores.

### **2.6.3.2. Factores influyen en el aprendizaje?**

Igualmente los factores ambientales son importantes en esta teoría además la interacción entre variables del conocimiento. Es importante que el aprendizaje se lo realice con actividades que estén relacionadas con los temas que se desarrollen.

### **2.6.3.3. Papel de la memoria**

El objetivo es que el estudiante pueda construir e interpretar su conocimiento. Una definición o concepto se va actualizando de acuerdo a su utilización, por lo que la memoria estará en construcción.

### **2.6.3.4. Cómo ocurre la transferencia**

“La transferencia puede facilitarse envolviendo a la persona en tareas auténticas ancladas en contextos significativos.”<sup>8</sup>

Lo importante en esta teoría de aprendizaje es que el “aprendizaje siempre toma lugar en un contexto y que el contexto forma un vínculo inexorable con el conocimiento inmerso en él. Por lo tanto, la meta de la instrucción es describir las tareas con precisión y no es definir la estructura del aprendizaje requerido para lograr una tarea. Si el aprendizaje se descontextualiza, hay poca esperanza de que la transferencia ocurra”. La plataforma Moodle estableció su diseño mediante las ideas del constructivismo el cual afirma que en la mente del estudiante es donde se construye el conocimiento.

---

<sup>7</sup>[http://ares.unimet.edu.ve/programacion/psfase3/modII/biblio/CONDUCTISMO\\_%20COGNITIVISMO\\_%20CONSTRUCTIVISMO.pdf](http://ares.unimet.edu.ve/programacion/psfase3/modII/biblio/CONDUCTISMO_%20COGNITIVISMO_%20CONSTRUCTIVISMO.pdf)

<sup>8</sup>[http://ares.unimet.edu.ve/programacion/psfase3/modII/biblio/CONDUCTISMO\\_%20COGNITIVISMO\\_%20CONSTRUCTIVISMO.pdf](http://ares.unimet.edu.ve/programacion/psfase3/modII/biblio/CONDUCTISMO_%20COGNITIVISMO_%20CONSTRUCTIVISMO.pdf)



Figura 2.3: Constructivismo<sup>9</sup>

## 2.7. Metodologías de Desarrollo de Objetos de Aprendizaje Basado en Competencias

Actualmente usar las tecnologías de información y las comunicaciones, requieren que los alumnos conozcan los elementos básicos para enfrentar cambios laborales y de la vida diaria. Usar las metodologías origina una formación, integran el proceso de enseñanza – aprendizaje utilizando el contexto de formación basada en competencias en todo lo que se refiere a técnicas y orientación pedagógica que son el apoyo para desarrollar los cursos. Al utilizar el modelo de OA (Objetos de Aprendizaje) en la asignatura de Base de Datos I, permite al alumno que luego de cumplir el aprendizaje, esté en capacidad de desempeñar una determinada competencia. El contexto basado en competencias a nivel educativo, apoya a mejorar la calidad, con lo que se obtiene la sistematización de procesos referentes a talento humano con los instructores, destrezas didácticas y de valoración más ajustadas a los procesos de desempeño.

### 2.7.1. Competencias

Las competencias son las tácticas, habilidades y actitudes perceptibles que la persona contribuye a un entorno particular para desempeñar eficientemente. Además las competencias son procedimientos a través de los cuales los individuos ejecutan actividades o solucionan problemas, en base al conocimiento, destrezas, valores, en el entorno laboral-profesional, mediante tres tipos de saberes: *saber hacer*, *saber ser*, *saber conocer*.

---

<sup>9</sup> [www.slideshare.net/edgar\\_rg/diseo-instruccion-1737789](http://www.slideshare.net/edgar_rg/diseo-instruccion-1737789)

### 2.7.2. Objetos de Aprendizaje

Se conocen como objetos de aprendizaje a los elementos que apoyan al proceso de aprendizaje por medio de la tecnología. Pueden ser desde una página HTML, hasta un modelo de preparación con muchas páginas e imágenes. Los objetos de aprendizaje deben facilitar su reutilización, interoperabilidad, accesibilidad y permanencia en el tiempo. Cada objeto de aprendizaje debe ser usado como un componente independiente, y no debe hacer referencia a otros objetos.

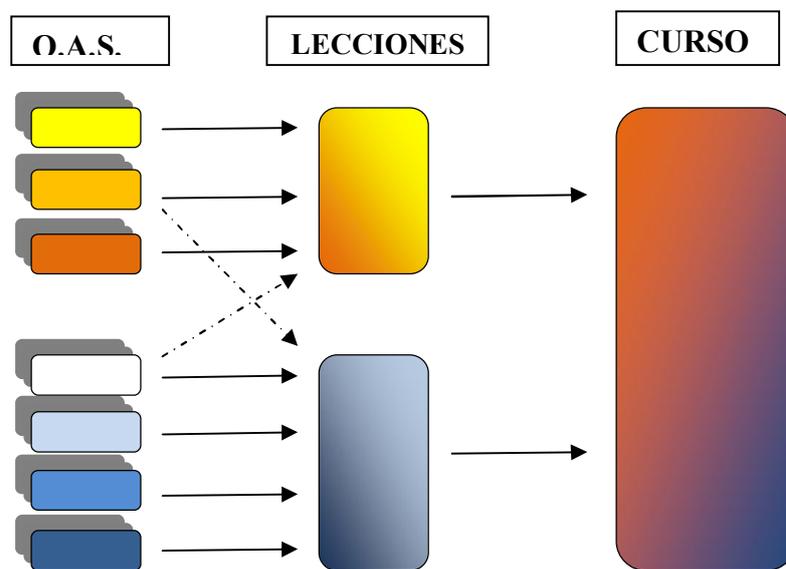


Figura 2.4: Estructura de Integración de Objetos de Aprendizaje<sup>10</sup>

<sup>10</sup> [http://nuria.worldhostsoft.com/demos/MII-L1\\_.html](http://nuria.worldhostsoft.com/demos/MII-L1_.html)

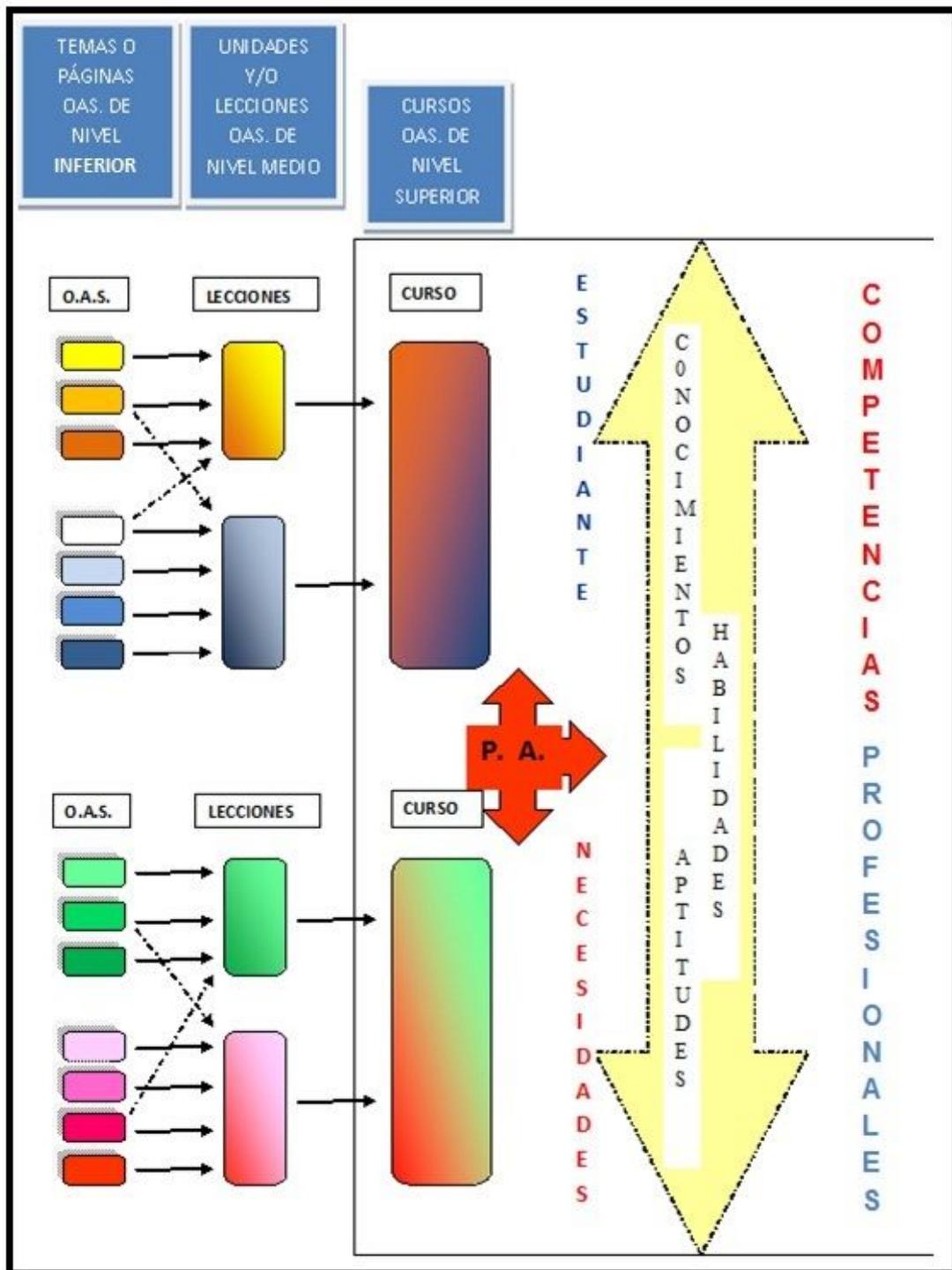


Figura 2.5: Estructura de Objetos de Aprendizaje<sup>11</sup>

<sup>11</sup> Hermosa Edgar (2009). *Introducción a la Metodología para Diseñar Cursos Virtuales*. Quito: Escuela Politécnica del Ejército.

## **2.8. Categorización o niveles de los objetos de aprendizaje**

Para el presente curso, los objetos de *aprendizaje de nivel superior se refieren a los cursos*, los mismos que están conformados por elementos (unidades académicas), que se establecen para lograr uno de los propósitos del total del objeto.

Los *objetos de nivel medio lo conforman las unidades académicas*, las que se organizan por objetos más pequeños llamados lecciones (también objetos de aprendizaje de nivel medio).

Los *objetos de aprendizaje de nivel inferior se refieren a los temas o páginas* que cumplen un objetivo o tienen una respuesta para una pregunta. En este nivel inferior están los tipos de contenido y evaluación (texto, imágenes, animaciones, videos). Se conoce al objeto de aprendizaje como la estructura mínima independiente, una entidad reutilizable.

A continuación se presenta los objetos de aprendizaje del curso de Base de Datos I.

- **Objeto de Aprendizaje Nivel Superior**

Curso B-Learning de Base de Datos

- **Objetos de Aprendizaje Nivel Medio**

**UNIDADES**

1. Generalidades de las Bases de Datos
2. Proceso de Desarrollo de la Base de Datos
3. SQL Lenguaje de Consulta Relacional Estructurado

**LECCIONES (correspondientes a la unidad 1).**

- 1.1. Bases de Datos
- 1.2. Sistemas de Gestión de Base de Datos
- 1.3. Modelos de Datos
- 1.4. Herramientas Case

## **LECCIONES (correspondientes a la unidad 2).**

- 2.1. Modelo Entidad – Relación
- 2.2. Modelo Relacional
- 2.3. Lenguajes Relacionales
- 2.4. Creación y Desarrollo de una Base de Datos
- 2.5. Diseño Conceptual, Lógico y Físico
- 2.6. Aplicación de una Herramienta Case
- 2.7. Normalización
- 2.8. Diccionario de Datos
- 2.9. Implementación de una Base de Datos utilizando SQL – DDL

## **LECCIONES (correspondientes a la unidad 3).**

- 3.1. SQL-DML
- 3.2. Consultas sobre tablas y tablas relacionadas
- 3.3. Funciones de Agregación

### **2.9. Metodologías para el Diseño del Curso**

Existen varias metodologías para el diseño de cursos, una de ellas es OOHDM mediante la cual se han realizado aplicaciones que se basan en sitios web, CD-ROM, usada en cursos anteriores, se aplicará también en éste, en las partes pertinentes al diseño de contenidos, puesto que en relación al software se utilizará la plataforma MOODLE previamente seleccionada.

### **2.10. Metodología OOHDM**

Método de Diseño Hipermedia Orientado a Objetos (OOHDM), admite una descripción concisa y aplicación de hipermedia; se lo realiza sobre el fundamento de modelos que permiten la implementación y que detallan la información, navegación, interfaces de aplicaciones.

OOHDM usa módulos de estructuración de un contexto orientado a objetos que permiten un breve detalle de los componentes de información y además describe los patrones de navegación y los cambios de una interfaz complicada. La aplicación hipermedia en esta metodología se desarrolla en cuatro etapas que se ajustan a formas de desarrollo. Las cuatro etapas son:

- El Diseño Conceptual constituyen los objetos y sus relaciones, similar al modelo entidad-relación en Bases de Datos, realizado mediante UML.
- En el Diseño Navegacional se realiza la estructura de navegación hipermedia en el que los vínculos provienen de las relaciones conceptuales. Los términos de nodos y enlaces describen a la semántica de navegación.
- En el Diseño de la Interfaz Abstracta lo más importante es definir los objetos en representaciones de interfaz del que resulta un aspecto visible al usuario.
- La siguiente etapa es la Implementación que describe el código de aplicación que se obtiene del diseño, esto es un proceso iterativo, el cual debe repetirse las veces necesarias hasta lograr el detalle requerido.

## **2.11. Metodologías para el Desarrollo de Contenidos**

### **2.11.1. Método Ergoglífico**

Se tomó el Método Ergoglífico como referencia para el desarrollo de contenidos en el Curso B-Learning de Base de Datos I, el que está conformado por las siguientes etapas:

**Análisis** se identifica los requerimientos, objetivos generales y específicos del curso, se describe el grupo de personas que van a usar el curso o utilizar el proyecto.

**Diseño** se construye la estrategia de trabajo, el diseño de contenidos como: lecciones, páginas que se va a mostrar en la plataforma, lo que conforma el framework (perfil del proyecto).

- **Selección de la Estrategia**

El diseño Ergoglífico tiene las siguientes estrategias:

- *Estrategia Informativa*: El alumno aprende a buscar información sin que le enseñen.
- *Estrategia Conductiva*: Instruye la parte conceptual o denominado contenido.
- *Estrategia Congnoscitiva*: Enseña la psicología del aprendizaje, ayuda al aprendiente por medio del descubrimiento a construir su propio conocimiento.
- *Estrategia Constructivista*: Se orienta al estudio del aprendizaje del alumno desarrollando habilidades para que se instruya por sí mismo.

- **Seleccionar el tipo de producto conocimiento**

Se debe seleccionar el tipo de producto - conocimiento en el que se verifique quien guía la forma de aprendizaje si el productor o el usuario. Por lo que antes de escoger el tipo de curso de debe considerar algunos puntos:

- Líder - Productor, que tiene el control sobre el contenido del curso, mientras que el usuario es el destinatario de la información.
- Líder - Consumidor, es quien tiene el control total de cómo se toma el curso.
  - Síncrono, es cuando un alumno efectúa una actividad al mismo tiempo o en tiempo real.
  - Asíncrono, los alumnos desarrollan las actividades solicitadas en el curso cuando lo requieran o mediante sus técnicas de estudio.

**Construcción** es equivalente al diseño, se detalla el diseño de flujo de trabajo y se crean normas a usar para la descripción del proyecto.

**Evaluación** es ejecutada al terminar el proyecto, efectuando evaluaciones en niveles como: aprendizaje logrado, aceptación, facilidad de uso y aplicación en el mundo real.

### **2.11.2. Especificación de Objetos de Aprendizaje**

Los Objetos de Aprendizaje constituyen parte importante del proceso de aprendizaje, formando una estructura de recursos que son independientes, se puede reutilizarlos. Hay varios tipos de Objetos de Aprendizaje:

- Colecciones. Ejemplo: Carrera
- Objetos de Aprendizaje de Nivel Superior. Ejemplo: Curso
- Objetos de Aprendizaje de Nivel Medio. Ejemplo: Unidades y Lecciones
- Objetos de Aprendizaje de Nivel Inferior. Ejemplo: Temas y Páginas

### **2.11.3. Especificación de Framework**

Se denomina framework a todo lo que no es contenido y suministra el ambiente externo del curso. Está formado de recursos y servicios, elementos de acceso como el menú principal, menú secundario, mapa del curso.

### **2.11.4. Lista de Requerimientos Tecnológicos**

Se definirá las herramientas y tecnologías que se usarán en el curso.

#### **2.11.4.1. Diseño del Flujo de Trabajo**

##### **▪ Participantes y proceso**

Es el detalle de los Participantes y los procesos que han intervenido en el desarrollo del curso:

Contenido de la Materia

Digitación de

Diseño Grafico

Gráficos

Implementación en el Moodle

- **Herramientas para el curso**

Son las herramientas a utilizar para el desarrollo del curso.

#### **2.11.4.2. Establecimiento de Estándares Internos de Interfaces**

- **Selección de colores**

El uso de colores para las páginas del curso se lo realiza mediante un formulario que se denomina paleta de colores. Los colores se detallan en valores hexadecimales, describiendo dónde se los usará, esto permite al programador que es quien realiza las plantillas a saber los colores que se usará en las diferentes definiciones de estilo.

- **Selección del tipo de letra**

Se debe especificar la tipografía del proyecto, los tipos de letra a usarse en textos de ilustraciones, módulos de contenido, pueden ser marcados sobre el cuerpo del texto de cada página. Estos detalles forman parte de las plantillas básicamente.

- **Disposición de Ventanas y Frames**

Se debe definir la disposición de las páginas y otros componentes, las capas deben detallar el tamaño de la ventana y proporciones, como se encuentra la división en el diseño, el lugar de los botones y controles más elementales.

## CAPÍTULO 3

### DESARROLLO E IMPLEMENTACIÓN

#### 3.1. Competencias

A continuación se determinan, las *competencias genéricas* que alcanzará el estudiante que aprueba el curso de Base de Datos I:

- A) Conceptualiza y compara los principios de base de datos y lenguajes de manejo de base de datos.
- B) Diseña e implementa bases de datos conforme a las características del sistema de información al cual va a soportar dicha base de datos, según los requerimientos establecidos, definidos o proporcionados, asegurándose de que exista integridad de los datos, utilizando el lenguaje SQL y herramientas CASE de apoyo, observando estándares y metodologías.
- C) Estructura y ejecuta consultas sobre los datos para responder las necesidades de información de los usuarios, utilizando el lenguaje SQL, asegurándose que el resultado sea el correcto en un tiempo óptimo.
- D) El estudiante tomará y desarrollará con eficiencia las asignaturas de nivel superior

Las *competencias específicas* que se pretende alcanzar en el estudio de cada unidad del curso de Base de Datos I son:

#### *COMPETENCIA GENÉRICA*

- A) Conceptualiza y compara los principios de base de datos y lenguajes de manejo de base de datos.

#### *Competencias específicas*

- a) Diferencia que son datos, información y conocimiento.
- b) Conceptualiza las bases de datos.

- c) Identifica que es un Sistema de Gestión de Base de Datos y sus componentes principales.
- d) Identifica las funciones del Sistema de Gestión de Base de Datos.
- e) Compara los Sistemas de Gestión de Base de Datos.
- f) Define los Modelos de Datos y su importancia.
- g) Compara los Modelos Lógicos Basados en Registros y los Modelos Basados en Objetos.
- h) Aplica y compara las Herramientas Case.

### *COMPETENCIA GENÉRICA*

B) Diseña e implementa bases de datos conforme a las características del sistema de información al cual va a soportar dicha base de datos, según los requerimientos establecidos, definidos o proporcionados, asegurándose de que exista integridad de los datos, utilizando el lenguaje SQL y herramientas CASE de apoyo, observando estándares y metodologías.

### *Competencias específicas*

- a) Identifica el Modelo Entidad Relación y sus componentes.
- b) Identifica y comprende el Modelo Relacional
- c) Identifica las metodologías como Diseño Conceptual, Diseño Lógico, Diseño Físico.
- d) Aplica los Lenguajes Relacionales (Algebra y Cálculo Relacional)
- e) Implementa una base de datos y aplica una Herramienta Case.
- f) Aplica las Formas de Normalización.
- g) Utiliza el Diccionario de Datos partiendo de la aplicación desarrollada con la Herramienta Case.
- h) Utiliza el lenguaje SQL - DDL para crear, modificar y eliminar tablas.
- i) Implementa una base de datos utilizando una herramienta case y aplicando integridad, normalización y SQL-DDL.

### *COMPETENCIA GENÉRICA*

C) Estructura y ejecuta consultas sobre los datos para responder las necesidades de información de los usuarios, utilizando el lenguaje SQL, asegurándose que el resultado sea el correcto en un tiempo óptimo.

#### *Competencias específicas*

- a) Identifica y aplica un conjunto de instrucciones procedimentales SQL, para manipular la información en cualquier modelo de datos.
- b) Realiza las consultas sobre tablas y tablas relacionadas.

### *COMPETENCIA GENÉRICA*

D) El estudiante tomará y desarrollará con eficiencia las asignaturas de nivel superior.

#### *Competencia específica*

- a) Aplicará los conocimientos adquiridos en el curso de Base de Datos I en asignaturas de nivel superior.

## **3.2. Objetos de Aprendizaje**

Los objetos de aprendizaje del curso de Base de Datos I están divididos por niveles: superior, medio e inferior.

### **OBJETO DE APRENDIZAJE de nivel superior**

- **Curso:** Base de Datos I

### **Objetos de aprendizaje de nivel medio:**

- **Unidad académica:** Generalidades de las Bases de Datos

### **Objetos de aprendizaje de nivel medio:**

- **Lección:** Bases de datos
- **Lección:** Sistemas de Gestión de Base de Datos
- **Lección:** Modelos de Bases de Datos
- **Lección:** Herramientas Case

- **Unidad académica:** Proceso de Desarrollo de la Base de Datos

**Objetos de aprendizaje de nivel medio:**

- **Lección:** Modelo Entidad – Relación
  - **Lección:** Modelo Relacional
  - **Lección:** Lenguajes Relacionales
  - **Lección:** Creación y Desarrollo de una Base de Datos
  - **Lección:** Diseño Conceptual, Lógico y Físico
  - **Lección:** Aplicación de una Herramienta Case
  - **Lección:** Normalización
  - **Lección:** Diccionario de Datos
  - **Lección:** Implementación de una Base de Datos utilizando SQL – DDL
- 
- **Unidad académica:** SQL Lenguaje de Consulta Relacional Estructurado
- Objetos de aprendizaje de nivel medio:**
- **Lección:** SQL-DML
  - **Lección:** Consultas sobre tablas y tablas relacionales
  - **Lección:** Funciones de Agregación.

A continuación se describen los objetos de aprendizaje de nivel medio (Unidades y Lecciones), objetos de aprendizaje de nivel inferior (Temas):

**OBJETO DE APRENDIZAJE de nivel medio**

- **Unidad académica:** Generalidades de las Bases de Datos

*COMPETENCIA GENÉRICA*

Conceptualiza y compara los principios de base de datos y lenguajes de manejo de base de datos.

## OBJETOS DE APRENDIZAJE **de nivel medio**

- **Lección:** Bases de datos

### OBJETOS DE APRENDIZAJE **de nivel inferior**

- Introducción
- Dato, Información y Conocimiento
  - Dato
  - Información
  - Conocimiento
  - Ejemplos de Dato, Información y Conocimiento
- Definición de la Base de Datos

## OBJETOS DE APRENDIZAJE **de nivel medio**

- **Lección:** Sistemas de Gestión de Base de Datos

### OBJETOS DE APRENDIZAJE **de nivel inferior**

- Introducción
- Componentes de un entorno SGBD
  - Hardware
  - Software
  - Personas
  - Procedimientos
  - Datos
- Historia de los SGBD
- Definición del SGBD
- Objetivos de los Sistemas de Gestión de Base de Datos
  - Componentes principales del SGBD
  - Funciones Básicas del SGBD
  - SGBD más usados

## OBJETOS DE APRENDIZAJE **de nivel medio**

- **Lección:** Modelos de Bases de Datos

### OBJETOS DE APRENDIZAJE **de nivel inferior**

- Introducción
- Modelo
- Modelar
- Modelo de Datos
- Importancia de los Modelos de Datos
- Tipos de Modelos de Datos
  - Modelo Lógicos Basados en Registros
    - Modelo Jerárquico
    - Modelo de Red
    - Modelo Relacional
  - Modelos Lógicos Basados en Objetos
    - Modelo Entidad – Relación
    - Modelo Orientado a Objetos

## OBJETOS DE APRENDIZAJE **de nivel medio**

- **Lección:** Herramientas Case

### OBJETOS DE APRENDIZAJE **de nivel inferior**

- Introducción
- Definición
- Evolución de las Herramientas Case
- Características de las Herramientas Case
- Clasificación de las Herramientas Case
- Componentes de una Herramienta Case
- Herramientas Case más usadas

## OBJETO DE APRENDIZAJE **de nivel medio**

- **Unidad académica:** Proceso de Desarrollo de la Base de Datos

### *COMPETENCIA GENÉRICA*

Diseña e implementa bases de datos conforme a las características del sistema de información al cual va a soportar dicha base de datos, según los requerimientos establecidos, definidos o proporcionados, asegurándose de que exista integridad de los datos, utilizando el lenguaje SQL y herramientas CASE de apoyo, observando estándares y metodologías.

## OBJETOS DE APRENDIZAJE **de nivel medio**

- **Lección:** Modelo Entidad – Relación

### OBJETOS DE APRENDIZAJE **de nivel inferior**

- Introducción
- Reseña Histórica
- Entidades
  - Entidades Regulares
  - Entidades Débiles
- Relación
  - Relación Reflexiva o Recursiva
  - Relación Binaria
  - Relación Ternaria
- Cardinalidad
  - Uno a Uno (1:1)
  - Uno a Varios (1:N)
  - Varios a Uno (N:1)
  - Varios a Varios (N:N)
- Atributos
  - Atributos Simples
  - Atributos Compuesto
  - Atributos Univaluados o Monovaluados
  - Atributos Multivaluados o Multivalorados
    - Atributos Derivados
    - Atributos Claves

- Dominios
- Restricciones
  - Restricciones Inherentes al modelo
  - Restricciones de Integridad
- Identificadores
- Cardinalidad de Atributos
- Semántica de las Relación
- Dependencia en existencia e identificación
  - Dependencia en Existencia
  - Dependencia en Identificación
- Control de redundancias
- Relación n-arias
- Generalización y herencia
- Agregación
- Dimensión Temporal

#### OBJETOS DE APRENDIZAJE **de nivel medio**

- **Lección:** Modelo Relacional

#### OBJETOS DE APRENDIZAJE **de nivel inferior**

- Introducción
- Reseña Histórica
- Elementos Básicos
  - Relación
  - Atributo
  - Dominio
  - Tupla
  - Dominios y Atributos
  - Relaciones

- Clases de Relaciones
  - Persistentes
  - Base
  - Vistas
  - Instantáneas
  - Temporales
- Claves
  - Candidatas
  - Ajenas
- Restricciones
  - Inherentes
  - Semánticas
- Esquemas Relacionales
- Sistemas de Gestión de Base de Datos Relacionales
  - Reglas de Codd
- Tratamiento de valores nulos

#### OBJETOS DE APRENDIZAJE **de nivel medio**

- **Lección:** Lenguajes Relacionales

#### OBJETOS DE APRENDIZAJE **de nivel inferior**

- Introducción
- Álgebra Relacional
  - Definición
  - Operadores Primitivos
    - Operaciones Unarias
    - Operaciones de Conjuntos
  - Operadores Derivados
- Operadores de Consulta Adicionales
- Operadores con Valores Nulos
- Cálculo Relacional
  - Cálculo Relacional de Tuplas
  - Cálculo Relacional de Dominios

## OBJETOS DE APRENDIZAJE **de nivel medio**

- **Lección:** Creación y Desarrollo de una Base de Datos

### OBJETOS DE APRENDIZAJE **de nivel inferior**

- Ciclo de Vida de una Base de Datos
- Estudio Previo y Plan de Trabajo
  - Decisión política y fijación de objetivos (estudio de viabilidad)
  - Evaluación previa de medios y costes
  - Aprobación de una estructura orgánica
  - Plan de trabajo detallado
- Concepción de la Base de Datos y selección de equipo
  - Concepción de la Base de Datos
  - Especificaciones de las necesidades de equipo físico y lógico
    - Diseño y Carga
  - Diseño Lógico y Físico
  - Carga y optimización de la base
- Metodología de Diseño
  - Concepto de la Metodología
- Enfoques de Diseño
- Etapas de la Metodología
  - Modelado Conceptual
  - Diseño Lógico
  - Diseño Físico
- Entradas y Salidas del proceso

## OBJETOS DE APRENDIZAJE **de nivel medio**

- **Lección:** Diseño Conceptual, Diseño Lógico y Diseño Físico

### OBJETOS DE APRENDIZAJE **de nivel inferior**

- Introducción
- Proceso de diseño de la base de datos
  - Análisis de requerimientos
  - Diseño Conceptual de la Base de Datos
  - Diseño Lógico de la Base de Datos
  - Diseño Físico de la Base de Datos

## OBJETOS DE APRENDIZAJE **de nivel medio**

- **Lección:** Aplicación de una Herramienta Case

### OBJETOS DE APRENDIZAJE **de nivel inferior**

- Herramienta Power Designer
- Power Designer versión 15

## OBJETOS DE APRENDIZAJE **de nivel medio**

- **Lección:** Normalización

### OBJETOS DE APRENDIZAJE **de nivel inferior**

- Introducción
- Dependencia Funcional
  - Dependencia Funcional Plena
  - Dependencia funcional trivial
  - Dependencia funcional elemental
- La necesidad de normalizar
  - Pasos para normalizar
- Formas Normales
  - Primera Forma Normal (1NF)
  - Segunda Forma Normal (2NF)
  - Tercera Forma Normal (3NF)
  - Forma Normal de Boyce – Codd (FNBC)

- Cuarta Forma Normal (4NF)
- Quinta Forma Normal (5NF)

#### OBJETOS DE APRENDIZAJE **de nivel medio**

- **Lección:** Diccionario de Datos

##### OBJETOS DE APRENDIZAJE **de nivel inferior**

- Introducción
- Definición y Características
- Contenido de un Registro del Diccionario
  - Elementos de datos
  - Estructura de datos
    - Data Flows
  - Razones para la utilización del diccionario de datos

#### OBJETOS DE APRENDIZAJE **de nivel medio**

- **Lección:** Implementación de la base de datos utilizando SQL-DDL

##### OBJETOS DE APRENDIZAJE **de nivel inferior**

- Introducción
- Crear Tablas
- Modificar Tablas
- Eliminar Tablas
- Crear Índices
- Eliminar Índices

## OBJETO DE APRENDIZAJE **de nivel medio**

- **Unidad académica:** SQL Lenguaje de Consulta Relacional Estructurado

### *COMPETENCIA GENÉRICA*

Estructura y ejecuta consultas sobre los datos para responder las necesidades de información de los usuarios, utilizando el lenguaje SQL, asegurándose que el resultado sea el correcto en un tiempo óptimo.

## OBJETOS DE APRENDIZAJE **de nivel medio**

- **Lección:** SQL-DML

### OBJETOS DE APRENDIZAJE **de nivel inferior**

- Introducción
- Insertar Datos
- Modificar Datos
- Eliminar Datos

## OBJETOS DE APRENDIZAJE **de nivel medio**

- **Lección:** Consultas sobre tablas y tablas relacionales

### OBJETOS DE APRENDIZAJE **de nivel inferior**

- Introducción
- Consulta sobre tablas
  - SELECT
  - FROM
  - WHERE
  - GROUP BY
  - HAVING
  - ORDER BY
- Consulta sobre tablas relacionadas JOIN
  - Combinación Interna (INNER JOIN)
  - Combinación externa (*OUTER JOIN*)
  - Combinación completa (*FULL OUTER JOIN*)

## OBJETOS DE APRENDIZAJE **de nivel medio**

- **Lección:** Funciones de agregación

### OBJETOS DE APRENDIZAJE **de nivel inferior**

- AVG
- COUNT
- MAX
- MIN
- SUM

### **3.3. Diseño Instruccional (Aplicación del Modelo ADDIE)**

“El Diseño Instruccional es un proceso sistémico y sistemático por medio del cual a partir del análisis de una necesidad de aprendizaje, se seleccionan y desarrollan las actividades y recursos para satisfacerla, así como los procedimientos para evaluar si dicho aprendizaje fue alcanzado.”<sup>12</sup>

El Diseño Instruccional realiza un análisis de los requerimientos educativos para luego diseñar e implementar un método que permita lograr las metas propuestas, lo que implica el proceso de desarrollo de elementos necesarios y acciones requeridas, posteriormente se realizarán las pruebas y evaluaciones. La planificación de los cursos a realizarse deberá contener el objetivo que se requiere lograr, las personas a las que está dirigido el curso, los elementos que se requieren y cómo se realizará la evaluación. Esto implica establecer la información y describir los objetos de aprendizaje. En el desarrollo del presente curso se va a utilizar el Modelo ADDIE del diseño instruccional por ser un modelo genérico, sencillo y que además se ajusta a los requerimientos de aprendizaje.

---

<sup>12</sup> <http://ateneo-empresarial.com/archives/category/disenio-instruccional>

### **3.3.1. Análisis Instruccional**

#### **3.3.1.1. Características de la audiencia**

Es necesario definir los tipos de usuarios antes de especificar la audiencia: Los principales usuarios son los aprendientes que requieran aprobar la materia de Base de Datos I para cumplir con la malla curricular de Ingeniería de Sistemas e Informática. Para esto se requiere que el estudiante haya aprobado anteriormente la materia de Estructura de Datos como prerrequisito para tomar esta materia.

Las personas que deseen aprender temas o la asignatura de Base de Datos I sin ser estudiantes de la carrera de Ingeniería de Sistemas e Informática, para este caso se requiere que las personas tengan conocimientos previos de: Programación Estructurada, Programación Orientada a Objetos, Matemáticas Discretas y Estructura de Datos. Y la autorización para tomar únicamente esta asignatura.

**Nota importante:** Los aprendientes deben cumplir lo establecido en los reglamentos de la institución.

#### **3.3.1.2. Identificación de las necesidades de conocimientos**

Para estructurar el presente curso de Base de Datos I se requiere identificar las necesidades de conocimientos de los aprendientes, conforme a las competencias constantes en la malla curricular que determina el Departamento de Ciencias de la Computación (DECC) de la ESPE. Estas necesidades de conocimiento están conformadas por 3 unidades:

#### **Unidad 1: Generalidades de las Bases de Datos**

Lección 1.1: Bases de Datos

Lección 1.2: Sistemas de Gestión de Bases de Datos

Lección 1.3: Modelo de Datos

Lección 1.4: Herramientas Case

## **Unidad 2: Proceso de Desarrollo de la Base de Datos**

Lección 2.1: Modelo Entidad – Relación

Lección 2.2: Modelo Relacional

Lección 2.3: Lenguajes Relacionales

Lección 2.4: Creación y Desarrollo de una Base de Datos

Lección 2.5: Diseño Conceptual, Lógico y Físico

Lección 2.6: Aplicación de una Herramienta Case

Lección 2.7: Normalización

Lección 2.8: Diccionario de Datos

Lección 2.9: Implementación de una Base de Datos utilizando SQL – DDL

## **Unidad 3: SQL Lenguaje de Consulta Estructurado**

Lección 3.1: SQL – DML

Lección 3.2: Consulta sobre tablas y tablas relacionadas

Lección 3.3: Funciones de Agregación

### **3.3.1.3. El presupuesto requerido y disponible**

Para el diseño y elaboración de un curso virtual se debe especificar detalladamente los recursos requeridos como también la disponibilidad de los mismos debiendo ajustarse cuando sea posible los requerimientos.

En el caso del presente curso la plataforma Moodle es software libre y está a la disposición del Departamento de Ciencias de la Computación que además auspicia el desarrollo de este curso y dio su apoyo en lo referente a hardware y software, pero no existe un presupuesto determinado, por lo que el costo total del proyecto lo asumió la estudiante que está a cargo del mismo. Para el desarrollo de cursos en la plataforma Moodle el DECC ha reducido costos y empleos únicamente se ha trabaja con profesores especialistas en la materia de Base de Datos I y Diseño de Cursos Virtuales, en conjunto con egresados que realizan las tesis de grado quienes poseen conocimientos de programación, multimedia y diseño gráfico. El costo estimado del desarrollo del curso es de (trabajo del(a) tesista es de seis mil dólares americanos.

#### **3.3.1.4. Los medios de difusión**

La plataforma Moodle del DECC en la que se han subido los cursos, es un recurso obligatorio. Los cursos servirán de apoyo para los cursos presenciales, y de abrirse la Carrera de Ingeniería de Sistemas e Informática en modalidad virtual constituirán el principal recurso de aprendizaje.

Inicialmente el profesor y los alumnos de la modalidad presencial serán quienes realizarán la evaluación y ajustes necesarios.

#### **3.3.1.5. Limitaciones**

El funcionamiento de la plataforma Moodle depende del servicio de Internet y la interrupción o saturación del mismo afecta al funcionamiento.

El corto tiempo que se dedique a la revisión del curso en las fases de diseño y desarrollo se convierte en un inconveniente porque involucra más tiempo en la etapa de ajustes y evaluación del curso.

#### **3.3.1.6. Actividades de aprendizaje que necesitan hacer los estudiantes para el logro de competencias**

El curso de Base de Datos I está estructurado a partir de los temas planteados para la asignatura, en el presente proyecto, se estructura el curso de la siguiente manera:

Unidades

Lecciones

Visión Global

Contenido

Tareas (Actividades de Aprendizaje)

Prácticas (Actividades de Aprendizaje)

Lecturas Recomendadas

Resumen

Autoevaluación

Actividades de Aprendizaje

### **3.3.2. Diseño Instruccional**

#### **3.3.2.1. Selección del mejor ambiente**

Para determinar los objetivos del aprendizaje del contenido del curso de Base de Datos I, se realiza un análisis previo de los tipos de destrezas cognitivas, con lo que se asegura una adecuada estructuración del curso para alcanzar un buen aprendizaje.

#### **3.3.2.2. Señalamiento de los objetivos instruccionales**

Los objetivos que se desean lograr en el curso de Base de Datos I son:

- Alcanzar el conocimiento y dominio de la asignatura de Base de Datos I.
- Emplear el aprendizaje de Base de Datos I a largo plazo.
- Agregar al aprendizaje de contenidos la práctica de Base de Datos.

#### **3.3.2.3. Selección de estrategias pedagógica**

“Las estrategias pedagógicas utilizadas en un proceso de enseñanza-aprendizaje, se definen como un conjunto de acciones que tienen como propósito lograr uno o más objetivos de aprendizaje, a través de la utilización de diferentes métodos y/o recursos.”<sup>13</sup>

En el desarrollo de este curso se va a utilizar las estrategias pedagógicas por los siguientes aspectos:

- Aporte activo de los aprendientes y no la memorización.
- Agregar la enseñanza individual y grupal para que el aprendizaje del curso sea adecuado.
- Desarrollar el aprendizaje utilizando mecanismos como el aprendizaje basado en proyectos.

#### **3.3.2.4. Bosquejo de unidades, lecciones y módulos**

Las unidades y lecciones del curso de Base de Datos I, se detalla en el contenido del capítulo 4 (Contenido del Curso de Base de Datos I), de esta tesis.

---

<sup>13</sup> <http://www.miro.cl/duocuc/estpeda.htm>

### **3.3.2.5. Diseño del contenido del curso, teniendo en cuenta los medios interactivos electrónicos**

La plataforma Moodle o LMS Moodle hace posible usar los wikis, blogs y los sistemas que manejan grupos, pero su utilización es criterio de los profesores de acuerdo a las necesidades que se presente.

### **3.3.3. Desarrollo Instruccional**

#### **3.3.3.1. Se selecciona, obtiene o crea el medio requerido**

En el desarrollo de este proyecto se utiliza un medio virtual en Internet, el servidor se encuentra ubicado en la sede de la ESPE en Sangolquí, los estudiantes accederán al curso de Base de Datos I, desde cyber cafés o desde sus hogares. De acuerdo a los requerimientos y disponibilidad de recursos se utilizarán laboratorios móviles, laboratorios en el centro de apoyo e inclusive en determinados casos en la sede principal de la ESPE.

#### **3.3.3.2. Se determinan las interacciones apropiadas**

El estudiante escoge las interacciones apropiadas para obtener una experiencia innovadora, no obstante el tutor formula actividades de aprendizaje que permite estudiar la respectiva materia. Las actividades de aprendizaje incluyen las autoevaluaciones y foros (lo que permitirá aclarar ciertos temas).

#### **3.3.3.3. Planificación de actividades**

Para que los estudiantes asimilen de manera adecuada el aprendizaje se desarrollará una planificación de actividades de acuerdo a las unidades que se deben cubrir en la asignatura de Base de Datos I y los tiempos que se deban cumplir en cada unidad. La estructura de las unidades está de acuerdo a las competencias y éstas se encuentran asociadas con los objetos de aprendizaje constituidos por unidades y lecciones.

### **3.3.4. Implantación e Implementación Instruccional**

#### **3.3.4.1. Duplican y distribuyen los materiales**

Moodle es una plataforma soportada en Internet por lo cual está disponible para en cualquier lugar y cualquier hora para un buen número de personas, únicamente se debe basar en las normas que la institución establezca para el efecto.

#### **3.3.4.2. Implanta e implementa el curso**

El curso se está levantando en una plataforma de desarrollo, para su implementación se lo debe realizar sobre una plataforma de producción, esto se aplicará tanto para la evaluación de los cursos con los profesores y estudiantes de la modalidad presencial y además en la implementación de los mismos en la modalidad virtual.

#### **3.3.4.3. Resuelve problemas técnicos y se discuten planes alternos**

El aprendiente se puede enfrentar a problemas con el Internet en el ancho de banda, o más aún en la no disponibilidad de Internet, por lo que recibirá un CD con la información necesaria de la materia y además en el CD constarán las herramientas de apoyo como software: Winzip, Acrobat Reader y software libre.

Además el aprendiente podrá encontrarse con problemas técnicos en lo referente a las prácticas, para esto el aprendiente deberá asistir a los centros de apoyo (por esta razón nuestro curso es B-Learning), con la ayuda del tutor de la materia y la utilización del laboratorio móvil el estudiante desarrollará las prácticas. Existe un grupo de trabajo que está dedicado al diseño de esta infraestructura tecnológica en los centros de apoyo y de los laboratorios móviles que se requieren para impartir la Carrera de Ingeniería de Sistemas e Informática en modalidad virtual.

### **3.3.5. Evaluación Instruccional**

El curso de Base de Datos I y otros cursos que se están desarrollando en el grupo de trabajo se aplicará como apoyo en cursos presenciales, esto llevará alrededor de 6 meses, luego de este tiempo los profesores y estudiantes evaluarán el curso, realizarán las respectivas revisiones y observaciones para actualizar el curso antes de que entre a producción. Se debe recalcar que debido al rápido desarrollo de las Ciencias de la Computación el curso necesitará actualizaciones periódicas para mantenerse en producción.

El formato en el cual se encuentran desarrollados los diferentes cursos virtuales permitirá a los mismos profesores de las asignaturas realizar el mantenimiento y actualización de los contenidos de los cursos y de sus actividades de aprendizaje.

### **3.4. Metodología OOHDM**

En este punto se van aplicar los conocimientos relacionados con la metodología OOHDM que se expusieron en el Marco Teórico en el Capítulo 2.

#### **3.4.1. Diseño Conceptual**

En esta etapa se va a realizar una representación conceptual que muestra los objetos del dominio, sus relaciones y la colaboración. “En las aplicaciones hipermedia estándares, cuyos componentes no son modificados durante la ejecución, se podría usar un modelo de datos semánticas estructurales, de este modo en los casos de uso en que la información pueda cambiar dinámicamente se intente ejecutar cálculos complejos, se necesitara mejorar el comportamiento del modelo de objeto.”<sup>14</sup>

#### **3.4.2. Mapas de Navegación**

“En esta etapa de la metodología se desarrolla una topología navegacional que permita al usuario ejecutar las tareas requeridas por el curso utilizando la aplicación Moodle.

---

<sup>14</sup> Carrera Alejandro y Del Castillo Gabriela. (2008). *Tesis de Grado Curso E-Learning de Ingeniería de Software I*. Quito: Escuela Politécnica del Ejército.

La idea principal es mostrar e integrar una serie de tareas para utilizar el diseño navegacional intrínseco al sistema. Para el Curso B-Learning de Base de Datos I el mapa navegacional consta del ingreso a la página, donde el aprendiente encuentra el curso, las unidades y el contenido de las lecciones.

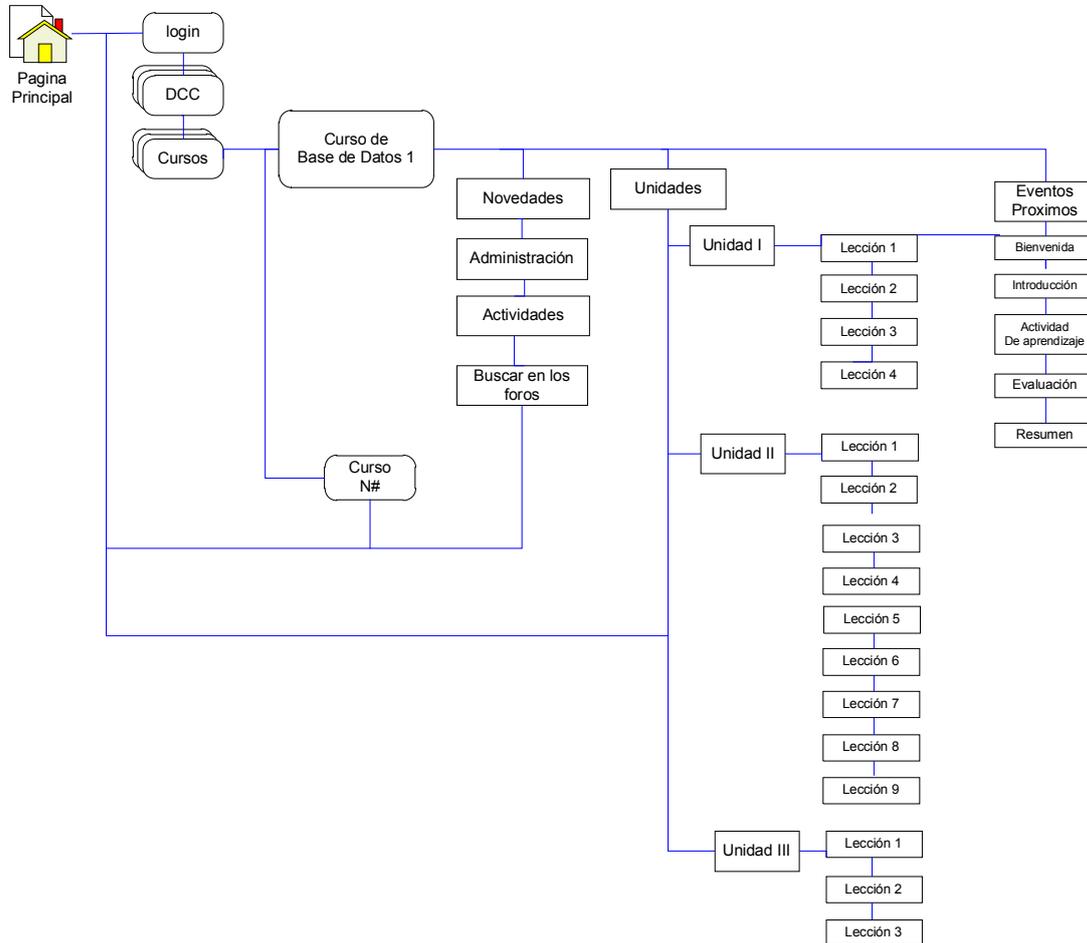


Figura: 3.1 Mapa Navegacional del Curso

### 3.4.3. Diseño de interfaces abstractas

Esta fase describe un modelo orientado a objetos que permite detallar la organización y la forma como se comporta la interface del sistema hipertexto con cada usuario. Para la implementación de nuestro curso no es necesario realizar esta fase, ya que la plataforma a usarse tiene estructurada esta interface. Este proyecto únicamente es el desarrollo de los contenidos y de las actividades de aprendizaje del curso más no del software de la plataforma.

### 3.4.4. Implementación

Esta es la etapa final o también conocida como fase de la metodología OOHDM en la que se realiza la ejecución del sistema hipermedia diseñado, de acuerdo al modelo de navegación y de interface de objetos con sus respectivos contenidos y facilidades para navegar.

### 3.4.5. Casos de Uso

Casos de uso de los módulos utilizados en el curso de Base de Datos 1.

#### Casos de uso del sistema – Ingreso a los cursos en la plataforma

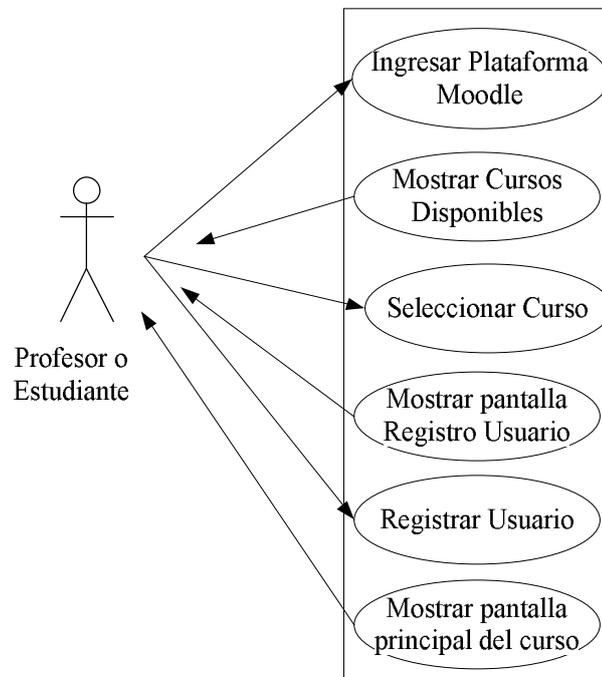


Figura 3.2: Caso de Uso - Ingresar a la plataforma

Tabla 3.1: Especificación del Caso de Uso – Ingresar a la Plataforma

<b>Caso Real:</b>	Ingresar a la Plataforma	
<b>Actor:</b>	Profesor / Alumno	
<b>Propósito:</b>	Permitir al profesor o al alumno acceder a la plataforma Moodle.	
<b>Flujo Principal:</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	Ingresar a la dirección web <b>http://midecc.espe.edu.ec/</b>	Mostrar la pantalla de cursos disponibles.
	Seleccionar el curso en el que le corresponde trabajar y hacer clic sobre la palabra “Entrar”.	Mostrar la pantalla de registro de usuario.
	Ingresar el nombre de usuario y contraseña y hacer clic sobre el botón “Entrar”.	Mostrar la pantalla principal del curso.

### Caso de uso foro

La participación en foros puede ser una parte integral de la experiencia de aprendizaje, ayuda a los alumnos a aclarar y desarrollar su comprensión del tema.

### Agregar Foro

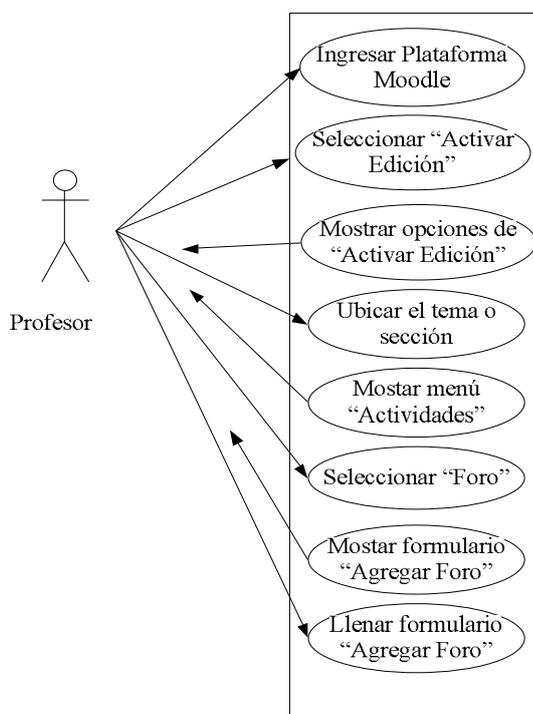


Figura 3.3: Caso de Uso – Agregar Foro

Tabla 3.2: Especificación del Caso de Uso – Agregar Foro

<b>Casos Real:</b>	Agregar Foro	
<b>Actor:</b>	Profesor	
<b>Propósito:</b>	Permitir al profesor crear el foro de la unidad o tema	
<b>Flujo Principal:</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	<p>Ingresar a la plataforma</p> <p>Seleccionar la función “Activar edición”.</p> <p>Ubicarse en qué tema o sección desea agregar la actividad foro. Seleccionar la opción “Foro”.</p> <p>El profesor ingresa la información correspondiente en los campos del formulario y hace clic en el botón “Guardar cambios”.</p>	<p>Mostrar las opciones de “Activar Edición” para cada uno de los recursos del curso.</p> <p>Mostrar el menú desplegable “Agregar Foro”</p> <p>Desplegar pantalla “Agregar Foro”.</p>

### Actualizar Foro

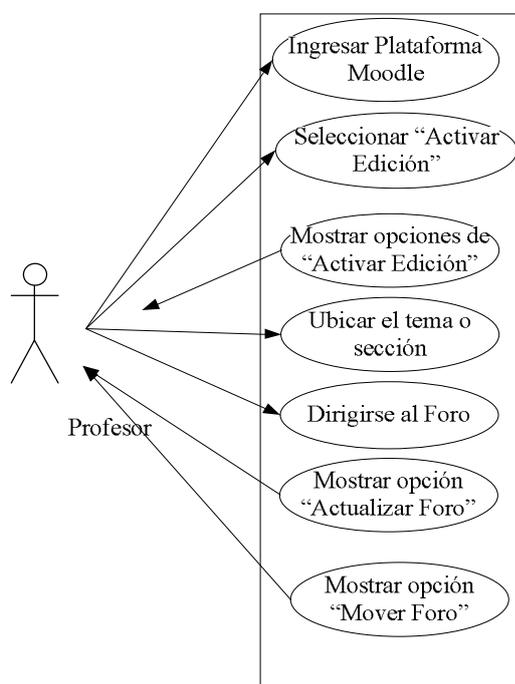


Figura 3.4: Caso de Uso – Actualizar Foro

Tabla 3.3: Especificación del Caso de Uso – Actualizar Foro

<b>Casos Real:</b>	Actualizar Foro	
<b>Actor:</b>	Profesor	
<b>Propósito:</b>	Permitir al profesor actualizar un foro de la unidad o tema	
<b>Flujo Principal:</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	<p>El profesor se ubica en la página principal del curso</p> <p>El profesor activa la función Activar edición.</p> <p>El profesor decide el foro de la Unidad o Tema que desea actualizar.</p> <p>El profesor hace clic en la función “Actualizar” del foro seleccionado.</p> <p>El profesor actualiza la información correspondiente en los campos del formulario</p> <p>El profesor hace clic en el botón “Guardar cambios”.</p> <p>El profesor confirma estar de acuerdo con las fechas haciendo clic sobre el botón “Continuar”.</p>	<p>El sistema despliega el formato predefinido del formulario foro.</p> <p>La plataforma muestra al profesor el tema del foro</p> <p>La plataforma ubica al profesor en la página principal del curso.</p>

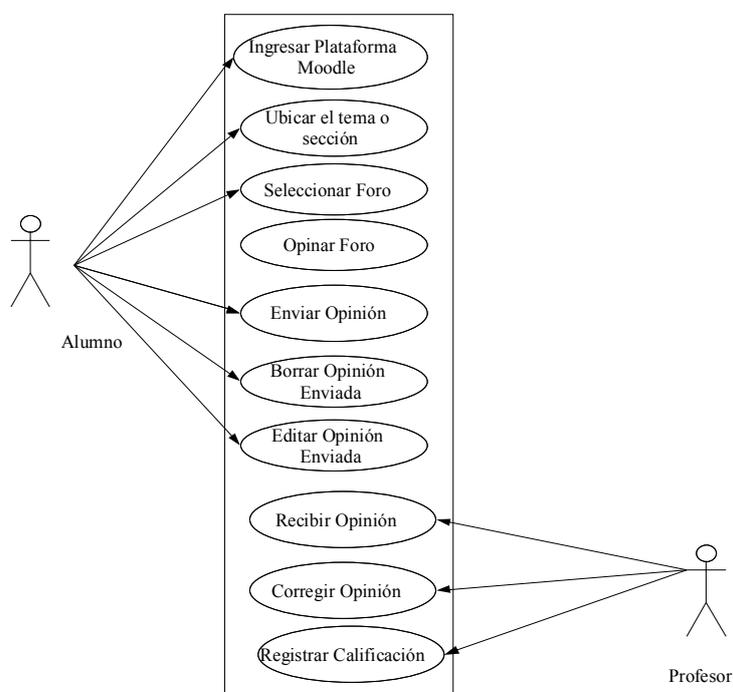


Figura 3.5: Caso de Uso – Desarrollar Foro

Tabla 3.4: Especificación del Caso de Uso – Desarrollar Foro

<b>Casos Real:</b>	Desarrollar Foro	
<b>Actor:</b>	Alumno	
<b>Propósito:</b>	Permitir al alumno enviar opinión del foro.	
<b>Flujo Principal:</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	<p>El estudiante se ubica en la página principal del curso</p> <p>El estudiante selecciona el foro del tema en cuestión.</p> <p>El estudiante hace clic sobre el foro que va a desarrollar.</p> <p>El estudiante llena el formulario para el Envío de foro.</p> <p>El estudiante hace clic sobre el botón “Enviar Opinión”.</p> <p>El estudiante hace clic sobre el botón “Continuar”.</p>	<p>La plataforma registra y confirma el envío de la Opinión.</p> <p>La plataforma muestra la información sobre las fechas y horas programadas para el foro, además de los envíos realizados.</p>

Tabla 3.5: Especificación del Caso de Uso - Editar Foro

<b>Casos Real:</b>	<b>Editar Foro</b>	
<b>Actor:</b>	Alumno	
<b>Propósito:</b>	Permitir al alumno editar el contenido de una opinión ya enviada.	
<b>Flujo Principal:</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	<p>El estudiante se ubica en la página principal del curso</p> <p>El estudiante selecciona el foro del tema en cuestión.</p> <p>El estudiante hace clic sobre el foro de la lección que desea editar.</p> <p>El estudiante hace clic sobre la acción “Editar” del correspondiente foro.</p> <p>El estudiante llena nuevamente el formulario con la información deseada “Editar Envío”.</p> <p>El estudiante hace clic sobre el botón “Guardar mi envío”</p> <p>El estudiante hace clic sobre el botón “Continuar”.</p>	<p>La plataforma muestra la información sobre las fechas y horas programadas para el foro, además de los envíos realizados.</p> <p>La plataforma registra y confirma el envío de la opinión.</p> <p>La plataforma muestra la información sobre las fechas y horas programadas para el foro, además de los envíos realizados.</p>

## Caso de Uso Tarea

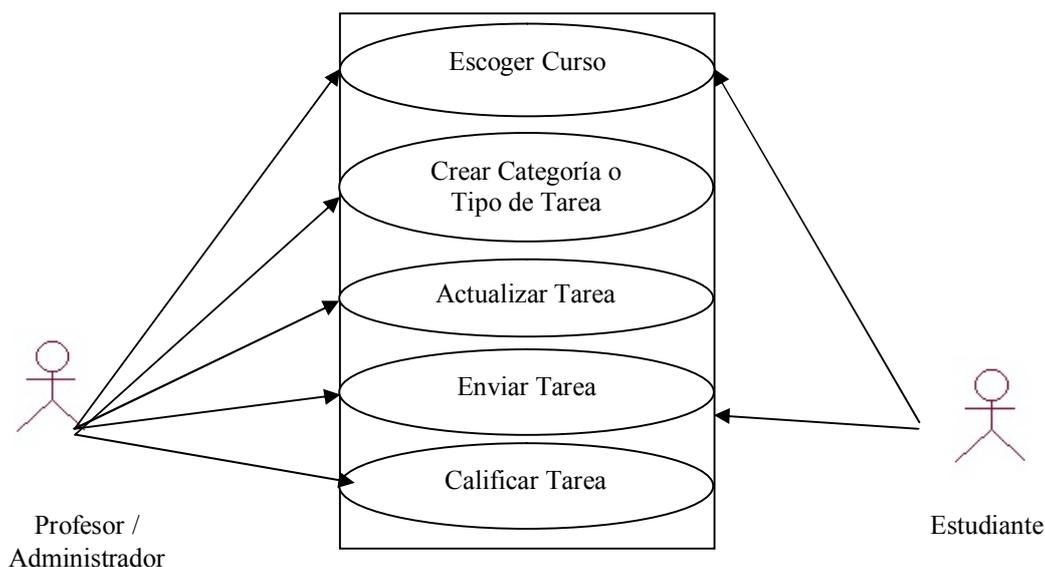


Figura 3.6: Caso de Uso – Tarea

El módulo de tarea, sirve para incluir tareas en un curso; pueden ser: subida avanzada, en línea, subir un solo archivo, actividad no en línea. Texto sin formato, archivos subidos, enlaces Web, Wiki o HTML o una referencia bibliográfica.

Tabla 3.6: Especificación del Caso de Uso - Escoger Curso

<b>Casos Real:</b>	Escoger Curso	
<b>Actor:</b>	Administrador, profesor, estudiante	
<b>Propósito:</b>	Permitir al profesor/estudiante ingresar al curso	
<b>Flujo Principal:</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	El estudiante se ubica en la página principal del curso El estudiante escribe su login y password y da clic en aceptar Elige la materia que se encuentra cursando actualmente Elige la opción tareas Elige la tarea correspondiente	Despliega la pantalla de autenticación Ingresa a la pantalla del DECC virtual Despliega la ventana del curso actual. Despliega la ventana que contiene las diferentes tareas.

Tabla 3.7: Especificación del Caso de Uso - Crear Clase o Tipo de Tarea

<b>Casos Real:</b>	Crear clase o tipo de tarea	
<b>Actor:</b>	Administrador, profesor	
<b>Propósito:</b>	Permitir al profesor o administrador crear un tipo o categoría de tarea de una lección	
<b>Flujo Principal:</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	El profesor pasa al modo de edición	El sistema muestra la interface de modo edición
	El profesor ingresa a la sección de agregar una actividad	El sistema muestra un listado de tareas como: Subida avanzada de archivos, Texto en línea
	El profesor selecciona la opción guardar cambios.	Subir un solo archivo Actividad no en línea

Tabla 3.8: Especificación del Caso de Uso - Actualizar Tarea

<b>Casos Real:</b>	Actualizar Tarea	
<b>Actor:</b>	Administrador, profesor	
<b>Propósito:</b>	Permitir al profesor o administrador modificar un tipo o categoría de tarea del curso	
<b>Flujo Principal:</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	El profesor pasa al modo de edición	El sistema muestra la interface de modo edición
	El profesor ingresa a la sección de modificar una actividad	
	El profesor selecciona la opción cambiar categoría o tipo de foro.	
	El profesor ingresa todos los campos necesario dentro del formulario de configuración cambiar o modificar	
El profesor selecciona la opción guardar cambios	El sistema despliega la ventana de envío de los cambios	

Tabla 3.9: Especificación del Caso de Uso - Calificar Tarea

<b>Casos Real:</b>	Calificar Tarea	
<b>Actor:</b>	Administrador, profesor	
<b>Propósito:</b>	Permitir al profesor o administrador calificar la tarea del curso	
<b>Flujo Principal:</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	El profesor pasa al modo de edición El profesor ingresa a la sección ver tarea El profesor selecciona la tarea a calificar El profesor revisa y califica la tarea El profesor selecciona la opción calificación y asienta la nota	El sistema muestra la interface de modo edición  El sistema abre el documento de la tarea enviada por el estudiante  El sistema despliega la ventana y visualiza la calificación.

### Caso de Uso Glosario

#### Agregar Glosario

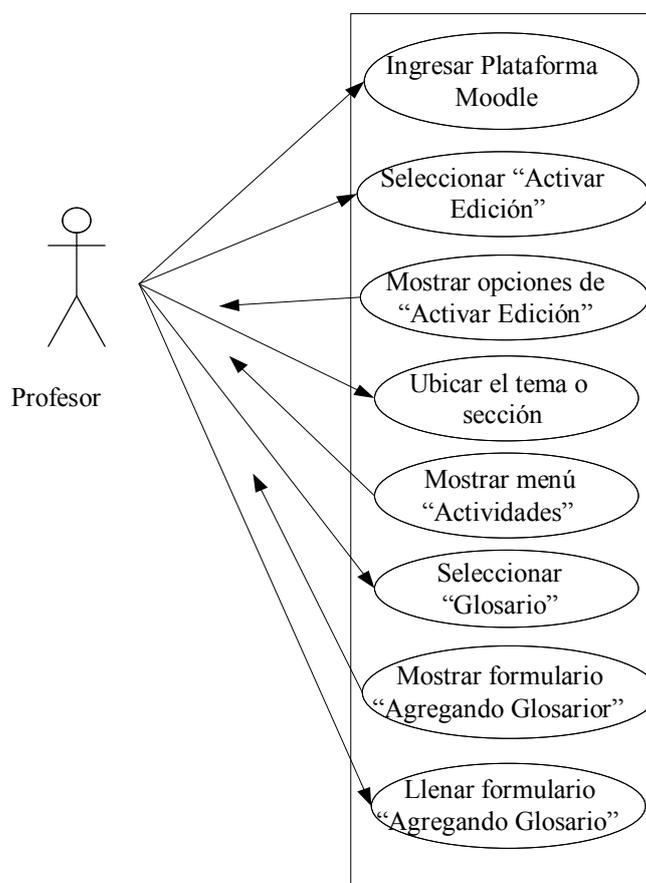


Figura 3.7: Caso de Uso – Agregar Glosario

Tabla 3.10: Especificación del Caso de Uso - Agregar Glosario

<b>Casos Real:</b>	Agregar Glosario	
<b>Actor:</b>	Profesor	
<b>Propósito:</b>	Permitir al profesor crear el glosario de la unidad o tema	
<b>Flujo Principal:</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	<p>Ingresar a la plataforma</p> <p>El profesor seleccionar la función "Activar edición"</p> <p>El profesor se ubica en el tema o sección desea agregar la actividad glosario</p> <p>El profesor selecciona la opción "Glosario"</p> <p>El profesor ingresa la información correspondiente en los campos del formulario</p> <p>El profesor ingresa a agregar entradas en el glosario creado, dentro de ajustes generales ingresa el concepto y la definición, puede escoger el tipo de letra, el tamaño, la alineación, formato. En este punto del formulario se puede agregar la palabra clave con el que se buscará el concepto que se ingresó anteriormente y si se requiere se puede ingresar también una imagen. Además se puede adjuntar un archivo hasta de 1MB que contenga información sobre las palabras del glosario</p> <p>El profesor hace clic en el botón "Guardar cambios".</p>	<p>Mostrar las opciones de "Activar Edición" para cada uno de los recursos del curso.</p> <p>Mostrar el menú desplegable "Agregar Actividad"</p> <p>Desplegar pantalla "Agregando Glosario".</p> <p>La plataforma ubica al profesor en la página principal del curso</p>

## Agregar Entrada Glosario

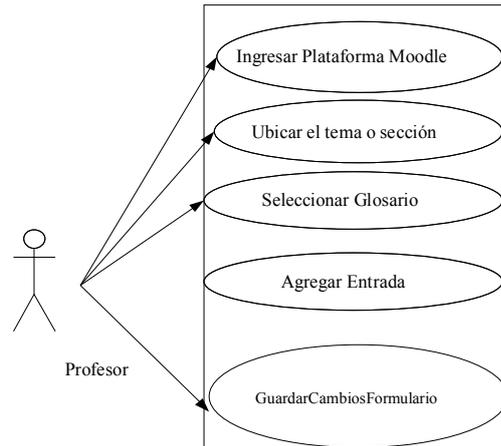


Figura 3.8: Caso de Uso – Agregar Entrada Glosario

Tabla 3.11: Especificación del Caso de Uso - Agregar Entrada Glosario

<b>Casos Real:</b>	Agregar Entrada Glosario	
<b>Actor:</b>	Profesor	
<b>Propósito:</b>	Permitir al profesor ingresar palabras a un glosario de trabajo de la unidad o tema	
<b>Flujo Principal:</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	El profesor se ubica en la página principal del curso	El sistema muestra las opciones de actividades
	El profesor selecciona el glosario del tema en cuestión	
	El profesor hace clic sobre el glosario que va a desarrollar	
	El profesor escoge la opción agregar entrada	La plataforma ubica al profesor en la página principal del curso
	El profesor ingresa datos al formulario y adjunta imágenes si es el caso	
El profesor guarda los cambios del formulario		



Tabla 3.13: Especificación del Caso de Uso - Actualizar Glosario

<b>Casos Real:</b>	Actualizar Glosario	
<b>Actor:</b>	Profesor	
<b>Propósito:</b>	Permitir al profesor actualizar un glosario de trabajo de la unidad o tema	
<b>Flujo Principal:</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	<p>El profesor se ubica en la página principal del curso</p> <p>El profesor activa la función Activar edición</p> <p>El profesor decide el glosario que desea actualizar</p> <p>El profesor hace clic en la función “Actualizar” del glosario elegido</p> <p>El profesor actualiza la información correspondiente en los campos del formulario.</p> <p>El profesor hace clic en el botón “Guardar cambios”</p> <p>El profesor confirma estar de acuerdo con las fechas haciendo clic sobre el botón “Continuar”.</p>	<p>El sistema despliega el formato predefinido del formulario glosario</p> <p>La plataforma muestra al profesor el glosario</p> <p>La plataforma ubica al profesor en la página principal del curso.</p>

Tabla 3.14: Especificación del Caso de Uso - Eliminar Glosario

<b>Casos Real:</b>	Eliminar Glosario	
<b>Actor:</b>	Profesor	
<b>Propósito:</b>	Permitir al profesor eliminar un glosario de trabajo de la unidad o tema	
<b>Flujo Principal:</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	<p>El profesor se ubica en la página principal del curso</p> <p>El profesor activa la función Activar edición</p> <p>El profesor decide el glosario que desea eliminar</p> <p>El profesor selecciona la función “Borrar”</p> <p>El profesor confirma o no la eliminación del glosario</p> <p>El profesor confirma estar de acuerdo con las fechas haciendo clic sobre el botón “Continuar”.</p>	<p>La plataforma muestra si el recurso fue eliminado o no</p> <p>La plataforma ubica al profesor en la página principal del curso</p>

Tabla 3.15: Especificación del Caso de Uso - Consultar Glosario

<b>Casos Real:</b>	Consultar Glosario	
<b>Actor:</b>	Profesor / Estudiante	
<b>Propósito:</b>	Permitir al profesor y/o estudiante consultar un glosario de trabajo de la unidad o tema	
<b>Flujo Principal:</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	El profesor y/o estudiante ingresa al sistema a actividades El profesor y/o estudiante ingresa a la actividad glosario El profesor y/o estudiante busca la palabra que requiere sea ingresando la palabra clave o navegando por el glosario usando el índice El profesor y/o estudiante consulta el concepto que requiere	El sistema muestra las opciones de actividades

**Casos de Uso Lección**

**Agregar Lección**

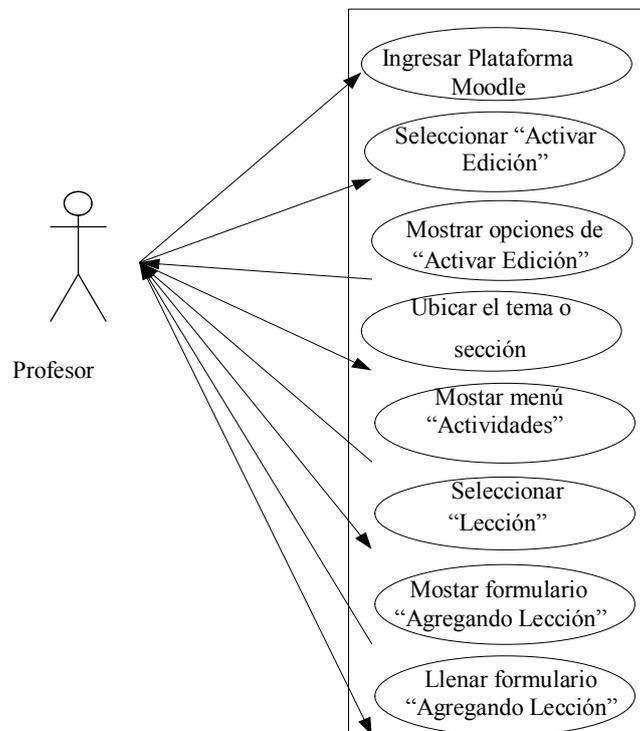


Figura 3.10: Caso de Uso – Agregar Lección

Tabla 3.16: Especificación del Caso de Uso - Agregar Lección

<b>Casos Real:</b>	Agregar Lección	
<b>Actor:</b>	Profesor	
<b>Propósito:</b>	Permitir al profesor crear el lección de la unidad o tema	
<b>Flujo Principal:</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	<p>Ingresar a la plataforma                  Seleccionar la función “Activar edición”.                  Ubicarse en qué tema o sección desea agregar la actividad lección.                  Seleccionar la opción “Lección”.                  El profesor continúa ingresando el formulario de configuración y en opciones de calificación debe especificar si es o no una lección de práctica, la calificación máxima, si le permite o no al estudiante retomar una lección, el manejo de nuevos intentos y la puntuación acumuladas que obtuvo el estudiante.                  El profesor hace clic en el botón “Guardar cambios”.</p>	<p>Mostrar las opciones de “Activar Edición” para cada uno de los recursos del curso.                  Mostrar el menú desplegable “Agregar Actividad”                  Desplegar pantalla “Agregando Lección”.</p>

### Modificar Lección

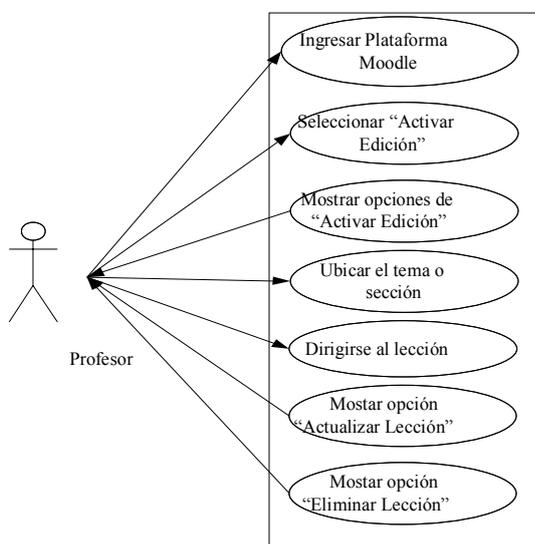


Figura 3.11: Caso de Uso – Modificar Lección

Tabla 3.17: Especificación del Caso de Uso - Modificar Lección

<b>Casos Real:</b>	Modificar Lección	
<b>Actor:</b>	Profesor	
<b>Propósito:</b>	Permitir al profesor modificar la lección de la unidad o tema	
<b>Flujo Principal:</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	<p>Ingresar a la plataforma                      El profesor pasa al modo de edición                      El profesor ingresa a la sección de modificar una actividad                      El profesor selecciona la opción modificar lección                      El profesor ingresa todos los campos necesario dentro del formulario de configuración cambiar o modificar                      El profesor hace clic en el botón "Guardar cambios".</p>	<p>El sistema muestra la interface de modo edición</p> <p>La plataforma ubica al profesor en la página principal del curso.</p>

Tabla 3.18: Especificación del Caso de Uso - Eliminar Lección

<b>Casos Real:</b>	Eliminar Lección	
<b>Actor:</b>	Profesor	
<b>Propósito:</b>	Permitir al profesor modificar la lección de la unidad o tema	
<b>Flujo Principal:</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	<p>Ingresar a la plataforma                      El profesor pasa al modo de edición                      El Profesor ingresa a la gestión de actividades                      El Profesor elige eliminar lección                      El Profesor acepta o cancela la eliminación de la lección</p> <p>El profesor hace clic en el botón "Guardar cambios".</p>	<p>El sistema muestra la interface de modo edición</p> <p>La plataforma ubica al profesor en la página principal del curso</p>

## Ingresar a lección

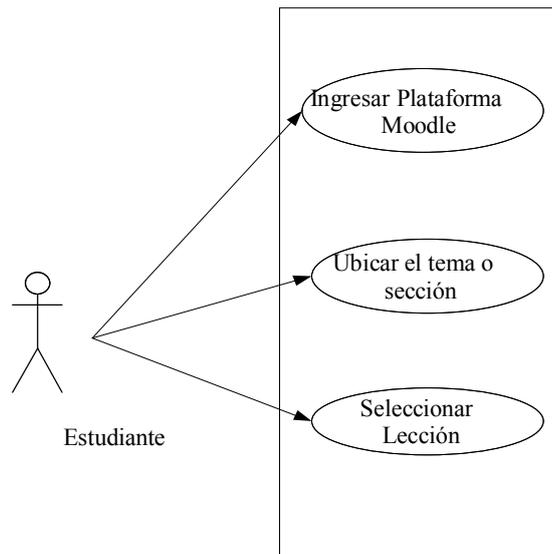


Figura 3.12: Caso de Uso – Ingresar a Lección

Tabla 3.19: Especificación del Caso de Uso - Ingresar a Lección

<b>Casos Real:</b>	Ingresar a Lección	
<b>Actor:</b>	Estudiante	
<b>Propósito:</b>	Permitir a un usuario ingresar a una lección	
<b>Flujo Principal:</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	<p>Ingresar a la plataforma</p> <p>El usuario ingresa al sistema a actividades</p> <p>El sistema muestra las opciones de actividades</p> <p>El usuario ingresa a la actividad o lección</p>	<p>El sistema muestra la interface de modo edición</p>

## Casos de Uso Taller

### Caso de Uso Específico – Agregar Taller

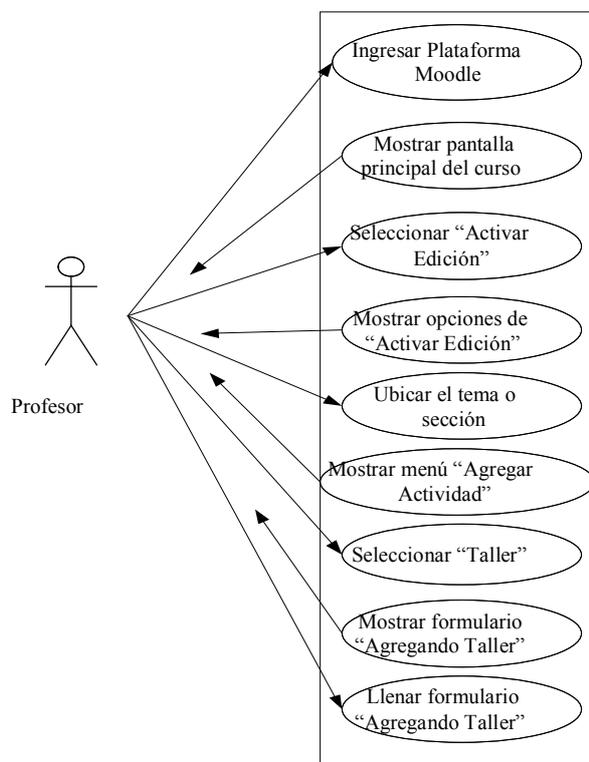


Figura 3.13: Caso de Uso – Agregar Taller

Tabla 3.20: Especificación del Caso de Uso - Agregar Taller

<b>Casos Real:</b>	Agregar Taller	
<b>Actor:</b>	Profesor	
<b>Propósito:</b>	Permitir al profesor crear el taller de la unidad o tema	
<b>Flujo Principal:</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	Ingresar a la plataforma Seleccionar la función "Activar edición". Se ubica en el tema o sección que desea agregar la actividad taller. Seleccionar la opción "Taller". Ingresar la información correspondiente en los campos del formulario y hace clic en el botón "Guardar cambios".	Mostrar la pantalla principal del curso.  Mostrar las opciones de "Activar Edición" para cada uno de los recursos del curso. Mostrar el menú desplegable "Agregar Actividad". Mostrar formulario "Agregarando Taller a Tema".

## Caso de Uso General - Actualizar Taller

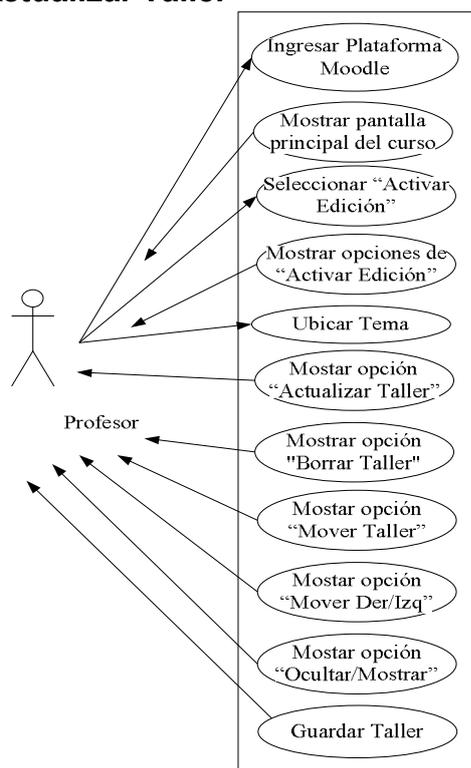


Figura 3.14: Caso de Uso – Actualizar Taller

Tabla 3.21: Especificación del Caso de Uso - Actualizar Taller

<b>Casos Real:</b>	Actualizar Taller	
<b>Actor:</b>	Profesor	
<b>Propósito:</b>	Permitir al profesor actualizar un taller de trabajo de la unidad o tema	
<b>Flujo Principal:</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	Ingresar a la plataforma Seleccionar la función “Activar edición”. Ubicar el taller de la unidad o tema que desea actualizar. Hacer clic en la función “Actualizar” del taller elegido. Actualizar la información correspondiente en los campos del formulario y hace clic en el botón “Guardar cambios”. Confirmar el estar de acuerdo con las fechas haciendo clic sobre el botón “Continuar”.	Mostrar la pantalla principal del curso. Mostrar las opciones de “Activar Edición” para cada uno de los recursos del curso. Mostrar el formato predefinido del formulario taller. Mostrar al profesor un resumen de las fechas predefinidas en el formulario Agregar Taller. Mostrar la página principal del curso.

Tabla 3.22: Especificación del Caso de Uso - Mover Derecha / Izquierda Taller

<b>Casos Real:</b>	Mover Derecha/Izquierda Taller	
<b>Actor:</b>	Profesor	
<b>Propósito:</b>	Permitir al profesor mover por niveles a la derecha o izquierda el taller de la unidad o tema.	
<b>Flujo Principal:</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	<p>Ingresar a la plataforma</p> <p>Seleccionar la función “Activar edición”.</p> <p>Decidir el taller de la unidad o tema que desea mover. Hace clic sobre la acción “Mover a la Derecha” o “Mover a la Izquierda” del taller elegido.</p>	<p>Mostrar la pantalla principal del curso.</p> <p>Mostrar las opciones de “Activar Edición” para cada uno de los recursos del curso.</p> <p>Desplegar la página principal del curso con el taller movido según la acción elegida a la izquierda o a la derecha.</p>

Tabla 3.23: Especificación del Caso de Uso - Mover Taller

<b>Casos Real:</b>	Mover Taller	
<b>Actor:</b>	Profesor	
<b>Propósito:</b>	Permitir al profesor mover entre las unidades o temas al taller.	
<b>Flujo Principal:</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	<p>Ingresar a la plataforma</p> <p>Seleccionar la función “Activar edición”.</p> <p>Decidir el taller de la unidad o tema que desea mover. Selecciona la función “Mover”.</p> <p>Seleccionar uno de los recuadros a donde mover el taller.</p>	<p>Mostrar la pantalla principal del curso.</p> <p>Mostrar las opciones de “Activar Edición” para cada uno de los recursos del curso.</p> <p>Mostrar los recuadros al lugar al cual podemos elegir moverlo.</p> <p>Mostrar la página principal del curso.</p>

Tabla 3.24: Especificación del Caso de Uso - Borrar Taller

<b>Casos Real:</b>	Borrar Taller	
<b>Actor:</b>	Profesor	
<b>Propósito:</b>	Permitir al profesor borrar un taller de la unidad o tema	
<b>Flujo Principal:</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	<p>Ingresar a la plataforma</p> <p>Seleccionar la función “Activar edición”.</p> <p>Decidir que taller de que Unidad o tema desea borrar. Selecciona la función “Borrar”.</p> <p>Seleccionar la opción Sí.</p>	<p>Mostrar la pantalla principal del curso.</p> <p>Mostrar las opciones de “Activar Edición” para cada uno de los recursos del curso.</p> <p>Mostrar mensaje “¿Está seguro de que desea borrar definitivamente el Taller?” Si o No.</p> <p>Mostrar la página principal del curso.</p>

Tabla 3.25: Especificación del Caso de Uso - Ocultar Taller

<b>Casos Real:</b>	Ocultar Taller	
<b>Actor:</b>	Profesor	
<b>Propósito:</b>	Permitir al profesor ocultar un taller de la unidad o tema	
<b>Flujo Principal:</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	<p>Ingresar a la plataforma</p> <p>Seleccionar la función “Activar edición”.</p> <p>Decidir que taller de que sección del curso desea ocultar. Seleccionar la función “Ocultar”.</p>	<p>Mostrar la pantalla principal del curso.</p> <p>Mostrar las opciones de “Activar Edición” para cada uno de los recursos del curso.</p> <p>Mostrar la página principal del curso, con la particularidad de que el nombre del taller oculto se muestra en color gris.</p>

Tabla 3.26: Especificación del Caso de Uso - Mostrar Taller

<b>Casos Real:</b>	Mostrar Taller	
<b>Actor:</b>	Profesor	
<b>Propósito:</b>	Permitir al profesor mostrar un taller de la unidad o tema	
<b>Flujo Principal</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	Ingresar a la plataforma  Seleccionar la función “Activar edición”.  Decidir que taller de que sección del curso desea mostrar. Seleccionar la función “Mostrar”.	Mostrar la pantalla principal del curso.  Mostrar las opciones de “Activar Edición” para cada uno de los recursos del curso.  Mostrar la página principal del curso, con la particularidad de que el nombre del taller ahora ya es de color negro.

### Caso de Uso Específico – Desarrollar Taller

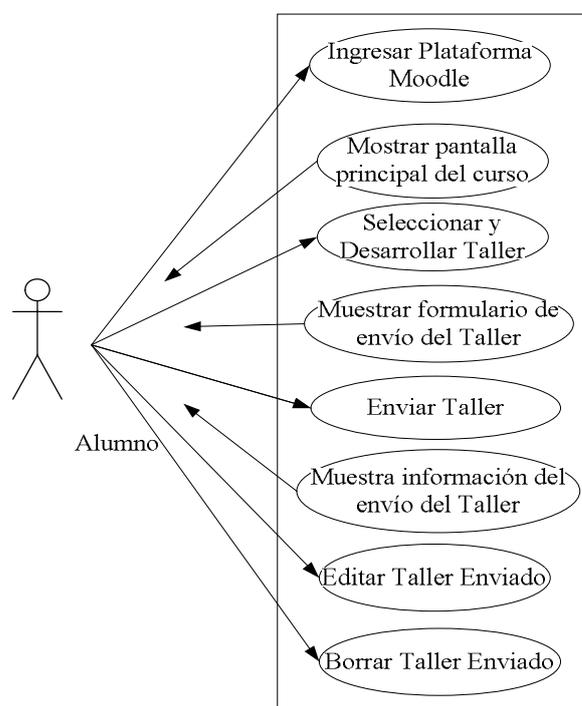


Figura 3.15: Caso de Uso – Desarrollar Taller

Tabla 3.27: Especificación del Caso de Uso - Desarrollar Taller

<b>Casos Real:</b>	Desarrollar Taller	
<b>Actor:</b>	Alumno	
<b>Propósito:</b>	Permitir al alumno desarrollar y enviar los planteamientos del taller.	
<b>Flujo Principal:</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	<p>Ingresar a la plataforma</p> <p>Seleccionar y desarrollar el taller del tema en cuestión.</p> <p>Llenar el formulario para el envío de taller y hacer clic sobre el botón “Enviar Tarea”.</p> <p>Hacer clic sobre el botón “Continuar”.</p>	<p>Mostrar la pantalla principal del curso.</p> <p>Mostrar formulario de envío del Taller.</p> <p>Registrar y confirmar el envío del taller.</p> <p>Mostrar información sobre las fechas y horas programadas para el taller, además de los envíos realizados.</p>

Tabla 3.28: Especificación del Caso de Uso - Editar Taller

<b>Casos Real:</b>	<b>Editar Taller</b>	
<b>Actor:</b>	Alumno	
<b>Propósito:</b>	Permitir al alumno editar el contenido de un taller ya enviado.	
<b>Flujo Principal:</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	<p>Ingresar a la plataforma</p> <p>Seleccionar el taller del tema en cuestión que desea editar.</p> <p>Hacer clic sobre la acción “Editar” del correspondiente taller y llena nuevamente el formulario con la información deseada “Editar Envío”.</p> <p>Hacer clic sobre el botón “Guardar mi envío”.</p>	<p>Mostrar la pantalla principal del curso.</p> <p>Mostrar la información sobre las fechas y horas programadas para el taller, además de los envíos realizados.</p> <p>Registrar y confirmar el envío de la tarea.</p> <p>Mostrar la información sobre las fechas y horas programadas para el taller, además de los envíos realizados.</p>

Tabla 3.29: Especificación del Caso de Uso - Borrar Taller

<b>Casos Real:</b>	<b>Borrar Taller</b>	
<b>Actor:</b>	Alumno	
<b>Propósito:</b>	Permitir al alumno borrar un taller ya enviado.	
<b>Flujo Principal:</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	<p>Ingresar a la plataforma</p> <p>Seleccionar el taller del tema en cuestión que desea borrar.</p> <p>Hace clic sobre la acción "Borrar" del correspondiente taller.</p> <p>El estudiante hace clic sobre el botón continuar.</p>	<p>Mostrar la pantalla principal del curso.</p> <p>Mostrar la información sobre las fechas y horas programadas para el taller, además de los envíos realizados.</p> <p>Pedir confirmación de la eliminación del taller.</p> <p>Borrar el taller.</p> <p>Mostrar la información sobre las fechas y horas programadas para el taller, además de los envíos realizados.</p>

### Caso de Uso Específico - Pregunta

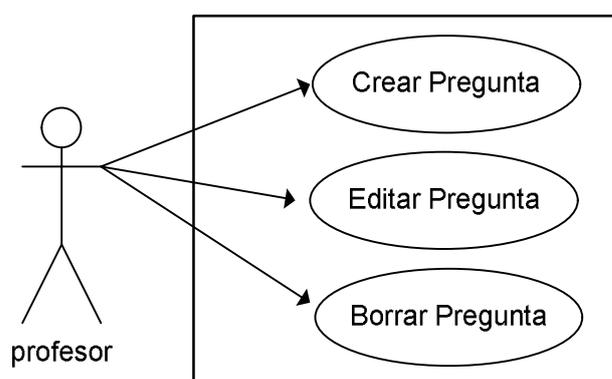


Figura 3.16: Caso de Uso – Añadir Preguntas al Cuestionario

Tabla 3.30: Especificación del Caso de Uso - Crear Pregunta

<b>Casos Real:</b>	Crear Pregunta	
<b>Actor:</b>	Profesor	
<b>Propósito:</b>	Permitir al profesor crear pregunta de una lección o unidad.	
<b>Flujo Principal:</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	Ingresar a la plataforma Seleccionar la actividad "Cuestionarios". Seleccionar la opción "Editar preguntas". Seleccionar mediante el menú desplegable la categoría de las preguntas a crear. Elegir el tipo de pregunta en el menú desplegable. Ingresar la información correspondiente en los campos del formulario y hace clic en "Guardar cambios".	Mostrar la pantalla principal del curso.  Mostrar listado de cuestionarios.  Desplegar menú "Editar preguntas".  Desplegar formulario en función del tipo de pregunta.  Desplegar menú Editar preguntas "Banco de Preguntas", donde lista las preguntas por categoría.

Tabla 3.31: Especificación del Caso de Uso - Editar Pregunta

<b>Casos Real:</b>	Editar Pregunta	
<b>Actor:</b>	Profesor	
<b>Propósito:</b>	Permitir al profesor editar una pregunta del cuestionario.	
<b>Flujo Principal:</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	Ingresar a la plataforma Seleccionar la actividad "Cuestionarios". Seleccionar la opción "Editar preguntas". Seleccionar mediante el menú desplegable la categoría de la pregunta a editar. Seleccionar la acción "Editar" de la respectiva pregunta. El profesor ingresa editar la información correspondiente en los campos del formulario y hace clic en "Guardar cambios".	Mostrar la pantalla principal del curso. Mostrar listado de cuestionarios. Desplegar la pestaña "Preguntas". Mostrar un listado de preguntas de la categoría seleccionada. Desplegar formulario de la pregunta a editar. Desplegar menú Editar preguntas "Banco de Preguntas", donde lista las preguntas por categoría.

Tabla 3.32: Especificación del Caso de Uso - Borrar Pregunta

<b>Casos Real:</b>	Borrar Pregunta	
<b>Actor:</b>	Profesor	
<b>Propósito:</b>	Permitir al profesor borrar una pregunta del cuestionario.	
<b>Flujo Principal:</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	Ingresar a la plataforma Seleccionar la actividad "Cuestionarios". Seleccionar la opción "Editar preguntas". Selecciona mediante el menú desplegable la categoría de la pregunta a borrar. Seleccionar la acción "Borrar" de la respectiva pregunta. Seleccionar "Si" en caso afirmativo de estar seguro de borrar la pregunta. Seleccionar "Continuar".	Mostrar la pantalla principal del curso. Mostrar listado de cuestionarios. Desplegar la pestaña "Preguntas". Mostrar un listado de preguntas de la categoría seleccionada. Preguntar si se está absolutamente seguro de que se desea borrar la pregunta. Desplegar la opción "Continuar". Desplegar menú Editar preguntas "Banco de Preguntas", donde lista las preguntas por categoría.

### Caso de Uso General - Cuestionario (Evaluación)

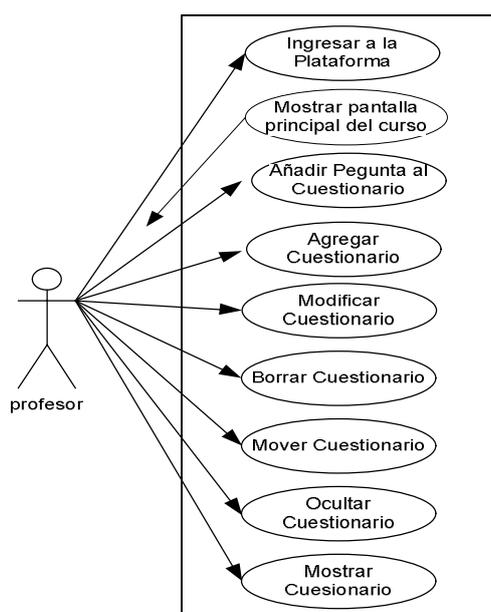


Figura 3.17: Caso de Uso – Cuestionario (Evaluación)

Tabla 3.33: Especificación del Caso de Uso - Añadir Pregunta al Cuestionario

<b>Casos Real:</b>	Añadir Pregunta al Cuestionario	
<b>Actor:</b>	Profesor	
<b>Propósito:</b>	Permitir al profesor agregar preguntas al cuestionario.	
<b>Flujo Principal:</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	Ingresar a la plataforma Seleccionar la actividad "Cuestionarios". Seleccionar la opción "Editar preguntas". Selecciona mediante el menú desplegable la categoría de la pregunta a borrar. Seleccionar la acción "Borrar de la respectiva pregunta. Seleccionar "Si" en caso afirmativo de estar seguro de borrar la pregunta. Seleccionar "Continuar".	Mostrar la pantalla principal del curso. Mostrar listado de cuestionarios. Desplegar la pestaña "Preguntas". Muestra un listado de preguntas de la categoría seleccionada. Preguntar si se está absolutamente seguro de que se desea borrar la pregunta. Desplegar la opción "Continuar". Desplegar menú Editar preguntas "Banco de Preguntas", donde lista las preguntas por categoría.

Tabla 3.34: Especificación del Caso de Uso - Agregar Cuestionario

<b>Casos Real:</b>	Agregar Cuestionario	
<b>Actor:</b>	Profesor	
<b>Propósito:</b>	Permitir al profesor crear el examen de la unidad o tema	
<b>Flujo Principal:</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	Ingresar a la plataforma  Decidir en qué tema o sección desea agregar la actividad. Seleccionamos en el menú desplegable Agregar Actividad el "Cuestionario".  El profesor ingresa a editar la información correspondiente en los campos del formulario y hace clic en "Guardar cambios".	Mostrar la pantalla principal del curso.  Mostrar formulario "Agregar Cuestionario".  Desplegar el caso de uso ya desarrollado "Añadir preguntas al cuestionario" (Tabla 3.4.1.5_34).

Tabla 3.35: Especificación del Caso de Uso - Modificar Cuestionario

<b>Casos Real:</b>	Modificar Cuestionario	
<b>Actor:</b>	Profesor	
<b>Propósito:</b>	Permitir al profesor modificar un cuestionario de una lección	
<b>Flujo Principal:</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	<p>Ingresar a la plataforma</p> <p>Decidir el cuestionario de que sección del curso desea modificar. Selecciona la función "Actualizar".</p> <p>El profesor ingresa a editar la información correspondiente en los campos del formulario y hace clic en "Guardar cambios".</p> <p>El profesor puede actualizar las preguntas del cuestionario haciendo clic sobre la pestaña "Editar."</p>	<p>Mostrar la pantalla principal del curso.</p> <p>Mostrar formulario "Actualizando Cuestionario".</p> <p>Mostrar una a una las preguntas que forman parte del cuestionario</p> <p>Desplegar el caso de uso ya desarrollado "Añadir preguntas al cuestionario" (Tabla 3.4.1.5_34).</p>

Tabla 3.36: Especificación del Caso de Uso - Borrar Cuestionario

<b>Casos Real:</b>	Borrar Cuestionario	
<b>Actor:</b>	Profesor	
<b>Propósito:</b>	Permitir al profesor borrar un cuestionario de una lección	
<b>Flujo Principal:</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	<p>Ingresar a la plataforma</p> <p>Seleccionar la función "Activar edición".</p> <p>Decidir el cuestionario de que sección del curso desea borrar. Selecciona la función "Borrar".</p> <p>Seleccionar la opción Si.</p>	<p>Mostrar la pantalla principal del curso.</p> <p>Mostrar las opciones de "Activar Edición" para cada uno de los recursos del curso.</p> <p>Mostrar mensaje "¿Está seguro de que desea borrar definitivamente Cuestionario?" Si o No.</p> <p>Mostrar la página principal del curso.</p>

Tabla 3.37: Especificación del Caso de Uso - Mover Cuestionario

<b>Casos Real:</b>	Modificar Cuestionario	
<b>Actor:</b>	Profesor	
<b>Propósito:</b>	Permitir al profesor mover un cuestionario de una lección	
<b>Flujo Principal</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	<p>Ingresar a la plataforma</p> <p>Seleccionar la función “Activar edición”.</p> <p>Decidir el cuestionario de que sección del curso desea mover. Selecciona la función “Mover”.</p> <p>Seleccionar uno de los recuadros a donde mover el cuestionario.</p>	<p>Mostrar la pantalla principal del curso.</p> <p>Mostrar las opciones de “Activar Edición” para cada uno de los recursos del curso.</p> <p>Mostrar los recuadros al lugar al cual podemos elegir moverlo.</p> <p>Mostrar la página principal del curso.</p>

Tabla 3.38: Especificación del Caso de Uso - Ocultar Cuestionario

<b>Casos Real:</b>	Ocultar Cuestionario	
<b>Actor:</b>	Profesor	
<b>Propósito:</b>	Permitir al profesor ocultar un cuestionario de una lección	
<b>Flujo Principal</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	<p>Ingresar a la plataforma</p> <p>Seleccionar la función “Activar edición”.</p> <p>Decidir el cuestionario de que sección del curso desea ocultar. Seleccionar la función “Ocultar”.</p>	<p>Mostrar la pantalla principal del curso.</p> <p>Mostrar las opciones de “Activar Edición” para cada uno de los recursos del curso.</p> <p>Mostrar la página principal del curso, con la particularidad de que el nombre del cuestionario oculto se muestra en color gris.</p>

Tabla 3.39: Especificación del Caso de Uso - Mover Cuestionario

<b>Casos Real:</b>	Mostrar Cuestionario	
<b>Actor:</b>	Profesor	
<b>Propósito:</b>	Permitir al profesor mostrar un cuestionario de una lección	
<b>Flujo Principal</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	Ingresar a la plataforma  Seleccionar la función “Activar edición”.  Decidir el cuestionario de que sección del curso desea mostrar.  Seleccionar la función “Mostrar”.	Mostrar la pantalla principal del curso.  Mostrar las opciones de “Activar Edición” para cada uno de los recursos del curso.  Mostrar la página principal del curso, con la particularidad de que el nombre del cuestionario ahora ya es de color negro.

### Caso de Uso Específico – Responder Cuestionario

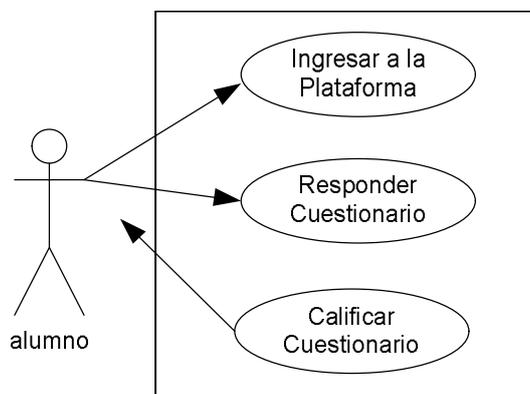


Figura 3.18: Caso de Uso – Responder Cuestionario

Tabla 3.40: Especificación del Caso de Uso - Responder Cuestionario

<b>Casos Real:</b>	Responder Cuestionario	
<b>Actor:</b>	Alumno	
<b>Propósito:</b>	Permitir al alumno responder las preguntas de un cuestionario.	
<b>Flujo Principal</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	<p>Ingresar a la plataforma</p> <p>El estudiante selecciona el cuestionario a desarrollar de la unidad correspondiente.</p> <p>Seleccionar la opción "Comenzar". Responder una a una las preguntas y puede guardar sin enviar, enviar la página o enviar todo y terminar.</p> <p>Confirmar el envío de las respuestas.</p>	<p>Mostrar la pantalla principal del curso.</p> <p>Mostrar información general del cuestionario y la opción de "Comenzar".</p> <p>Desplegar la primera pregunta del cuestionario.</p> <p>Mostrar información relacionada al cuestionario realizado tal como hora de inicio, fin, duración, calificación.</p>

Tabla 3.41: Especificación del Caso de Uso - Calificar Cuestionario

<b>Casos Real:</b>	Calificar Cuestionario	
<b>Actor:</b>	Profesor	
<b>Propósito:</b>	Calificar respuestas de un cuestionario.	
<b>Flujo Principal</b>	<b>Eventos del Actor</b>	<b>Eventos de la Plataforma</b>
	<p>Confirma el envío de las respuestas.</p> <p>Recibir notas.</p>	<p>Recibir cuestionario con las respuestas del estudiante.</p> <p>El profesor y/o la plataforma califican las respuestas de cuestionario.</p> <p>Registrar la nota del estudiante.</p> <p>Enviar la nota al estudiante.</p>

### 3.4.6. Diagramas de Secuencia

Mediante los diagramas de secuencia se puede conocer las clases que intervienen, las funciones que tendrán las clases y pantallas del sistema. A continuación se presente un diagrama general de la secuencia de lecciones de la Unidad I.

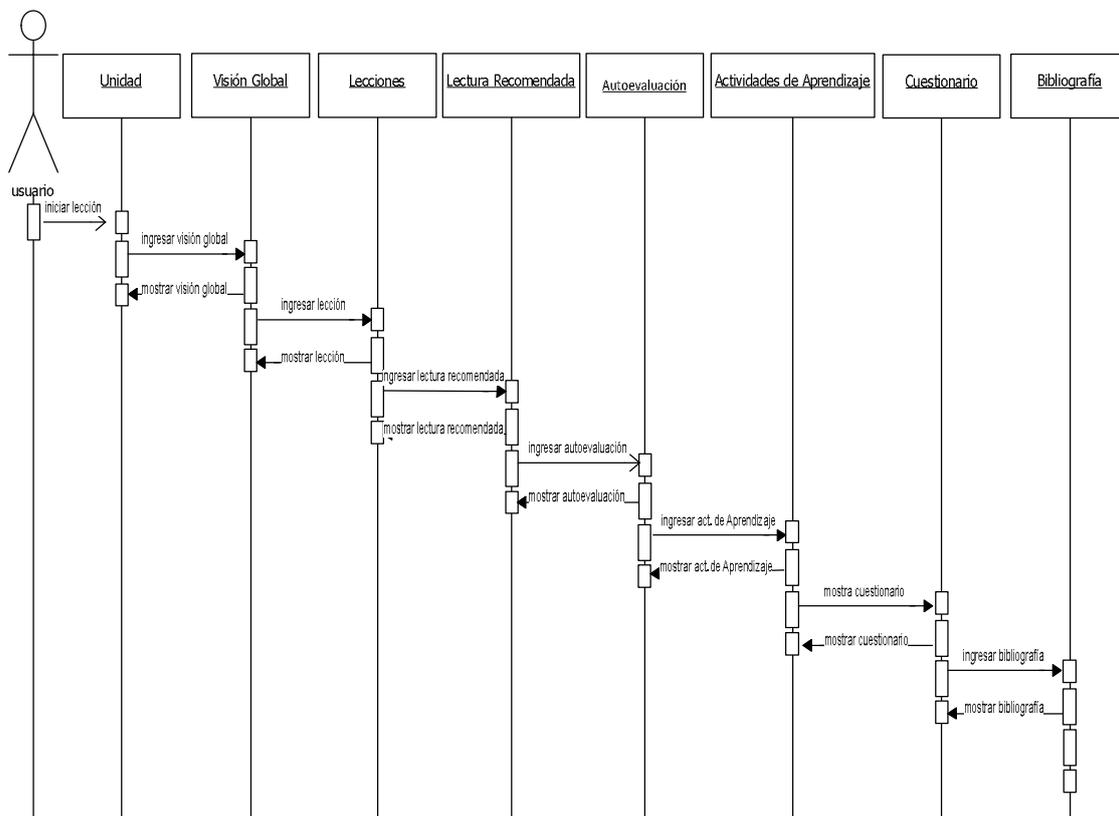


Figura 3.19: Ejemplo de Diagramas de Secuencia Unidad 1 <sup>15</sup>

### 3.4.7. Diagramas de Clases Navegacionales

Estos diagramas permiten identificar las llamadas principales de cada interfaz

<sup>15</sup> Hermosa Andrea y Chica Fernanda. (2008). *Tesis de Grado Curso E-Learning de Planificación Informática*. Quito: Escuela Politécnica del Ejército.

Este diagrama permite identificar las llamadas principales de cada interfaz.

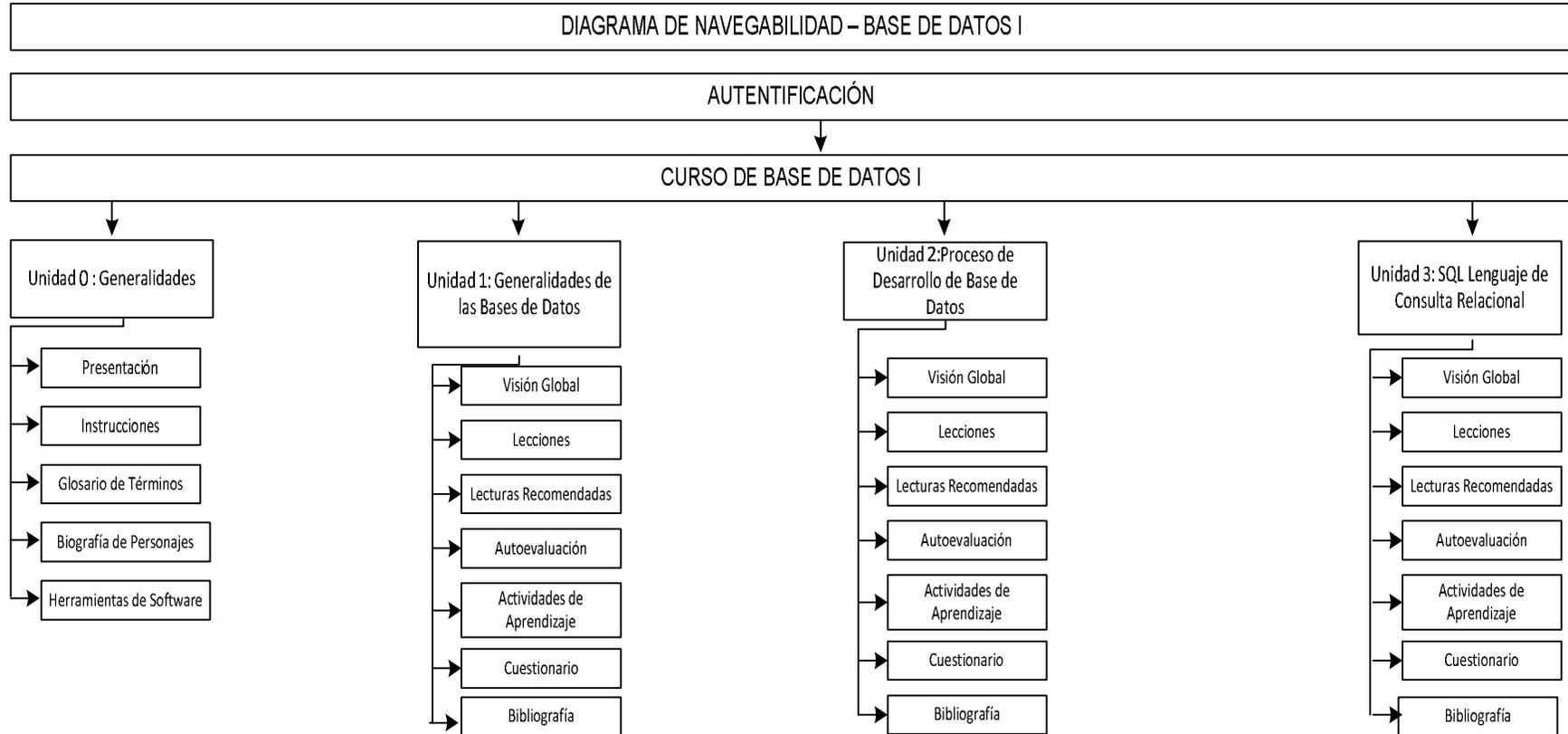


Figura 3.20: Diagrama de Clases Navegacionales <sup>16</sup>

<sup>16</sup> Hermosa Andrea y Chica Fernanda. (2008). *Tesis de Grado Curso E-Learning de Planificación Informática*. Quito: Escuela Politécnica del Ejército.

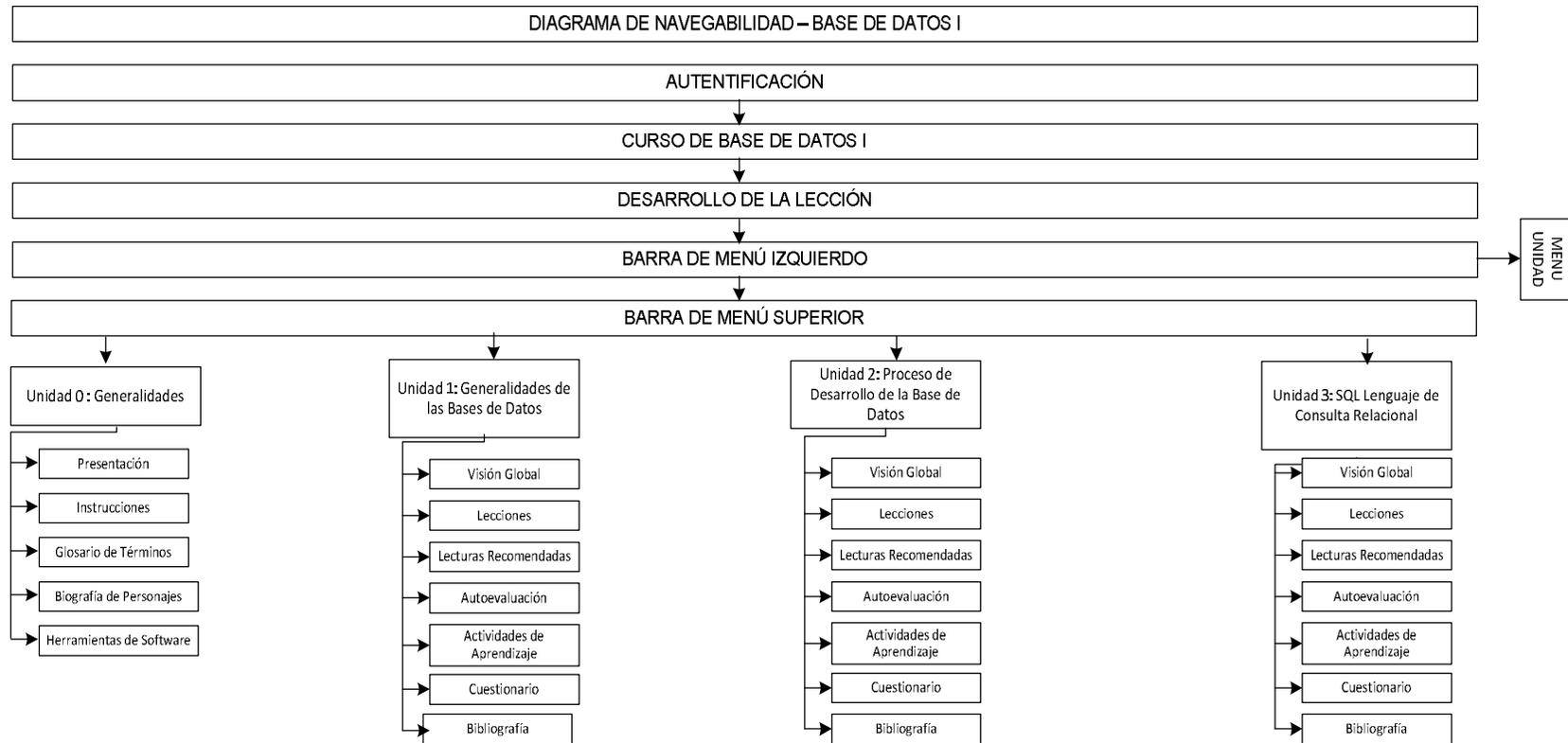


Figura 3.21: Diagrama Navegacional del Desarrollo de la Lección <sup>17</sup>

<sup>17</sup> Hermosa Andrea y Chica Fernanda. (2008). *Tesis de Grado Curso E-Learning de Planificación Informática*. Quito: Escuela Politécnica del Ejército.

### **3.4.8. Desarrollo del Curso B-Learning de Base de Datos I con el Método Ergoglífico**

Como se detalló en el capítulo anterior el Método Ergoglífico consta de las siguientes fases: Análisis, Diseño, Construcción y Evaluación; a continuación el detalle de los procesos que se aplicaron en este curso. En la fase de Análisis, se verifica las necesidades del curso, se establecen los objetivos que plantea la metodología para realizar el diagrama de análisis.

En la fase de Diseño, se escogerá la estrategia de trabajo, los tipos de curso, el diseño de los contenidos hasta proyectar las especificaciones y realizar el diseño del framework de trabajo.

En la Construcción se desarrolla el diseño de flujo de trabajo y se establece estándares que se usarán para la definición de páginas ya concretas.

La última etapa de Evaluación en la que se realiza pruebas de aplicación, correcciones, rendimiento del curso.

#### **3.4.8.1. Análisis Ergoglífico**

Fase importante en la que se deben establecer los objetivos que van a obtener, conclusiones exitosas para el curso. Cuando no se realiza esta fase el curso carece de una estructura adecuada, se debe realizar un análisis de la situación y no pasar a las fases de diseño y construcción directamente. Además se debe tener en cuenta el ahorro del tiempo y dinero lo que disminuye errores y ayuda a que las actividades que se ha planteado sean precisas. Se debe cumplir las siguientes sub fases:

- Conocimiento de los Usuarios
- Identificación de las Necesidades de Conocimiento
- Establecimiento de Objetivos
- Elaboración del Diagrama de Análisis

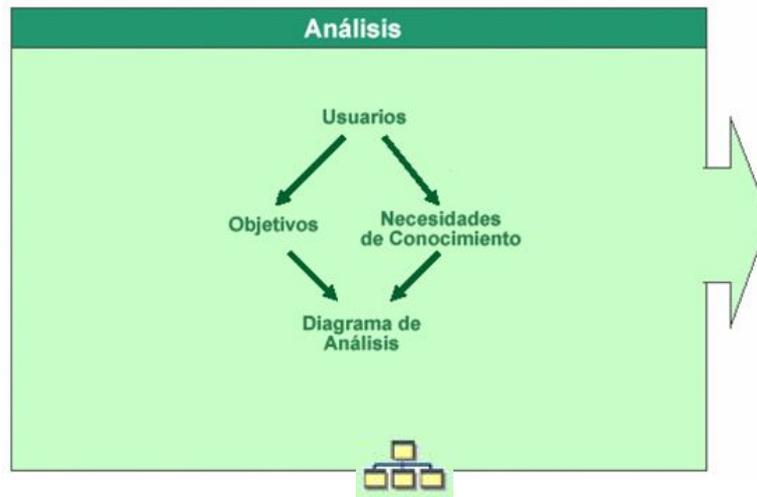


Figura 3.22: Diagrama de Fase de Análisis <sup>18</sup>

### 3.4.8.2. Conocimiento de los Usuarios

Se debe conocer a los usuarios que van acceder al curso, previo se debe definir a los mismos. Existen dos tipos de usuarios:

**El Docente o Profesor:** Es la persona que dirige, motiva, evalúa proceso de seguimiento de curso. El profesor debe cumplir los siguientes requisitos:

- Conocimiento y dominio de los temas expuestos en el sistema
- Conocimientos y manejo de computadoras, a un nivel superior
- Conocimiento y manejo Internet, paquetes informáticos (Microsoft Office)

**El Alumno:** Es la persona a quién va dirigido el curso. Esta persona debe cumplir los siguientes requisitos:

- Capacidad lógica y racionalidad
- Operar con objetos que tienen que estar presentes para ser percibidos y manipulados
- Razonamientos sencillos a partir de lo que observa, percibe y vive
- Realismo objetivo. Curiosidad
- Distingue lo reciente de lo que no lo es.
- Capacidad de colaboración
- Auto disciplinado

<sup>18</sup> Hermosa Andrea y Chica Fernanda. (2008). *Tesis de Grado Curso E-Learning de Planificación Informática*. Quito: Escuela Politécnica del Ejército.

Las anteriores se denominan características psicopedagógicas del estudiante, además de éstas se puede identificar importantes en el desarrollo del sistema:

- Conocimiento básico de los temas expuestos en el sistema
- Conocimiento y manejo de computadores a un nivel medio

### **3.4.8.3. Restricciones Generales**

Este curso es dirigido a estudiantes que se encuentren cursando la Carrera de Ingeniería de Sistemas e Informática. Contempla temas vigentes en la materia de Base de Datos I que se imparte en la carrera en mención.

#### ***Identificación de las necesidades del conocimiento***

Antes de iniciar con la Construcción se identificó lo siguiente:

El temario que se encuentra vigente de la materia de Base de Datos I. Se requiere reforzar conocimientos por parte de los alumnos, conceptos, habilidades que requieren y su prioridad. Se ha procedido con la recopilación de información de cada tema y se ha estructurado las lecciones de la siguiente forma:

- Visión Global de la Lección
- Contenido de la Lección
- Desarrollo de la Lección
- Resumen
- Actividades de Aprendizaje
  - Taller
- Actividades de Evaluación
  - Autoevaluación
  - Examen
  - Foro
- Lecturas Recomendadas
  - Lecturas Recomendadas Páginas Web
  - Lecturas Recomendadas PDF

### **Elaboración del diagrama de análisis**

Se debe realizar un diagrama de análisis del proyecto en la etapa final del análisis, que recopile los conceptos, acuerdos, compromisos y decisiones en un documento denominado “Formulario General del Proyecto”, que sirve también para la realización de etapas consecuentes a esta en el desarrollo del de la aplicación.

#### **3.4.8.4. Etapa de Diseño**

Esta fase se basa en la información obtenida en la anterior, para definir el contenido que va a tener la aplicación y el framework (área de trabajo) para la construcción del curso.

Los objetivos de aplicación en esta fase se convierten en detalles específicos y reales ya que se desarrollará los procesos de diseño en forma organizada y secuencial; con lo que se establece una estrategia de conocimiento y se trasmite hechos específicos a los usuarios del curso.

Cuando ya se desarrolló la estrategia de conocimiento se diseña los módulos del curso; iniciando con el módulo principal del que se despliegan sub módulos los que contienen objetos de conocimiento, esto se muestra en la siguiente figura:

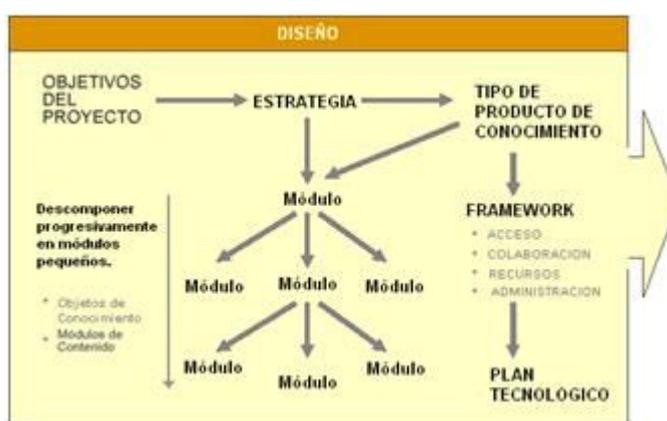


Figura 3.23: Diagrama de Diseño <sup>19</sup>

<sup>19</sup> Hermosa Andrea y Chica Fernanda. (2008). *Tesis de Grado Curso E-Learning de Planificación Informática*. Quito: Escuela Politécnica del Ejército.

En la aplicación el área de trabajo se forma mecanismos de acceso como menú, mapa del curso y mecanismos complementarios como glosario y enlaces que se recomienda para que el usuario revise.

Se debe agregar el plan tecnológico (detalle de los requerimientos de Hardware y Software) al área de trabajo y el contenido de los módulos; esto es importante para el desarrollo y uso de la aplicación. La etapa de Diseño tiene las siguientes sub fases:

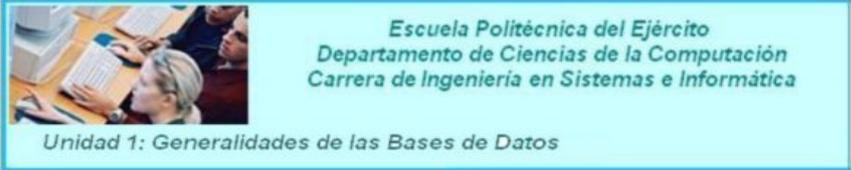
- ❖ Seleccionar la Estrategia
- ❖ Seleccionar el Tipo de Producto de Conocimiento
- ❖ Diseño Detallado del Contenido
- ❖ Especificación del Framework
- ❖ Listar requerimientos tecnológicos

Las sub fases se detallarán en la Construcción del Curso B-Learning de Base de Datos I, tomando en cuenta la aplicación en una lección específica del curso.

### **Especificación de Framework**

Se refiere a todo lo que no es contenido del curso, como cabecera, menús, botones que sirven para ingresar a las distintas páginas

Tabla 3.42: Especificación del Framework <sup>20</sup>

<b>Ámbito</b>	Archivos y Directorios																		
<b>Sección</b>	Contenido gráfico de archivos																		
<b>Formato</b>	<p>Los archivos DOC o PDF deberán tener una cabecera</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;">  </div> <p>El título general está precedido de un gráfico para cada capítulo posible de acuerdo a la tabla a continuación.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;"></td> <td>Visión Global</td> </tr> <tr> <td style="text-align: center;"></td> <td>Contenido de la lección</td> </tr> <tr> <td style="text-align: center;"></td> <td>Desarrollo de la lección</td> </tr> <tr> <td style="text-align: center;"></td> <td>Resumen de la lección</td> </tr> <tr> <td style="text-align: center;"></td> <td>Auto evaluación de la lección</td> </tr> <tr> <td style="text-align: center;"></td> <td>Examen de la lección (Alumno y maestro)</td> </tr> <tr> <td style="text-align: center;"></td> <td>Actividades de aprendizaje casos prácticos de la lección</td> </tr> <tr> <td style="text-align: center;"></td> <td>Actividades de aprendizaje Taller de la lección</td> </tr> <tr> <td style="text-align: center;"></td> <td>Lecturas recomendadas</td> </tr> </table> <p>El fondo de la aplicación debe ser Papel seda azul de Microsoft Office – Word Y los recuadros que se utilicen deben tener bordes color verde RGB (153, 204,0).</p>		Visión Global		Contenido de la lección		Desarrollo de la lección		Resumen de la lección		Auto evaluación de la lección		Examen de la lección (Alumno y maestro)		Actividades de aprendizaje casos prácticos de la lección		Actividades de aprendizaje Taller de la lección		Lecturas recomendadas
	Visión Global																		
	Contenido de la lección																		
	Desarrollo de la lección																		
	Resumen de la lección																		
	Auto evaluación de la lección																		
	Examen de la lección (Alumno y maestro)																		
	Actividades de aprendizaje casos prácticos de la lección																		
	Actividades de aprendizaje Taller de la lección																		
	Lecturas recomendadas																		

<sup>20</sup> Hermosa Andrea y Chica Fernanda. (2008). *Tesis de Grado Curso E-Learning de Planificación Informática*. Quito: Escuela Politécnica del Ejército.

## Lista de Requerimientos Tecnológicos

Se realiza un listado de los requerimientos tecnológicos que debe disponer el alumno.

Tabla 3.43: Requisitos de Hardware y Software<sup>21</sup>

Requisitos mínimos de Software	Requisitos mínimos de Hardware
Tener instalado un Sistema Operativo Windows 98 o superior	Procesador Intel Pentium IV superior, o en sus defectos compatibles.
Tener Instalado los Paquetes Microsoft Office 3	Monitor Super VGA de 800 x 600
Conexión a Internet	1GB de RAM
T A B	Espacio en disco duro de al menos 20 MB de memoria libre.  Kit Multimedia

## Participantes y procesos

### Diseño del Flujo de Trabajo

#### - Participantes y proceso

Al referirse a participantes se habla de las personas que se encuentran incluidas para realizar el curso, es decir los profesores, los aprendientes y otros usuarios. Se denomina authoring a la creación del curso que contiene a los programadores, diseñadores instruccionales y expertos en desarrollo de cursos. Se conoce a los usuarios del curso como aprendientes o estudiantes.

<sup>21</sup> Hermosa Andrea y Chica Fernanda. (2008). *Tesis de Grado Curso E-Learning de Planificación Informática*. Quito: Escuela Politécnica del Ejército.

- **Participantes y que necesitan ellos**

Las personas que participan en el curso (productor y consumidor) requieren todas las facilidades de la tecnología para lograr sus propósitos.

Tabla 3.44: Necesidades de los Participantes <sup>22</sup>

	<b>Productor</b>	<b>Consumidor</b>
<b>Hardware</b>	Computadora con los Paquetes de Software Necesario para desarrollar el curso	Computador Personal
<b>Software</b>	Software realizado por el autor o creador	Conexión a Internet con Internet Explorer

- **Herramientas para el curso**

A continuación se describe el software utilizado para el desarrollo del Curso B-Learning de Base de Datos I.

El browser o navegador que el usuario (alumno) va utilizar para entra en el sitio o Campus Virtual es Microsoft Internet Explorer.

El hosting o servidor será el que utiliza la Escuela Politécnica del Ejército.

Para desarrollar el contenido de las lecciones se utilizó *Microsoft Word 2007*.

Moodle (Modular Object Oriented Distance Learning Environment) para el levantamiento del curso E-Learning.

<sup>22</sup> Hermosa Andrea y Chica Fernanda. (2008). *Tesis de Grado Curso E-Learning de Planificación Informática*. Quito: Escuela Politécnica del Ejército.

## **Establecimiento de Estándares Internos de Interfaces**

- ***Selección de colores***

Se requiere establecer los colores que se van a usar en la aplicación, los mismos se encuentran en valores hexadecimales lo que se conoce como paleta de colores. El curso va a tener colores definidos lo que se describe en el Formato de Color.

- ***Selección del tipo de letra***

Es necesario establecer el tipo de letra que se usará en el desarrollo del curso, estas descripciones son parte de plantillas definidas. La tipografía se encuentra en el Formato de Fuentes.

***Nota importante:***

La tipografía que se empleará en el curso que se levantará en la plataforma Moodle y no en el documento de tesis.

Tabla 3.45: Paleta de Colores <sup>23</sup>

Paleta de Color									
Nombre del color o numero	Ejemplo	Valores RGB			Valores de C.L.S. de color			Valor Hexadec	Donde es usado el color
		Rojo	Verde	Azul	Color	Luz	Sat		
Verde Lima		153	204	0	75	100	80	99CC00	Cuadro de notas calve dentro de documento
Celeste		212	255	255	180	17	100	D4FFFF	Imagen de cabecera de documentos
Agua Marina		51	204	204	180	75	80	33CCCC	Cuadro de mensaje motivacional
Verde Azulado		0	128	128	180	100	50	008080	Titulo general de capítulos
Azul Marino		0	0	208	240	0	208	F000D0	Subtítulos de contenido de documentos
Añil		51	51	153	240	67	60	333399	Línea bajo el titulo
Azul Oscuro		0	0	128	240	100	50	000080	Texto motivacional
Rojo		215	38	3	10	99	84	D72603	Títulos de contenidos
Negro		0	0	0	10	0	0	000000	Color de texto

<sup>23</sup> Hermosa Andrea y Chica Fernanda. (2008). *Tesis de Grado Curso E-Learning de Planificación Informática*. Quito: Escuela Politécnica del Ejército.

Tabla 3.46: Tipografía<sup>24</sup>

Tipografía									
Componente	Fuente						Párrafo		
	Nombre	Tamaño	Estilo	Color	Fondo	Alineación	Ancho	Espacio	Sangría
	Titulo 1	<i>Arial Rounded MT Bold,</i>	20	Negrita	RGB 000128128	RGB 204236255	Izquierda	-	-
Titulo 2	<i>Arial Rounded MT Bold,</i>	20	Normal	RGB 215038003	RGB 204236255	Izquierda	-	-	-
Titulo	<i>Arial</i>	12	Negrita	RGB 000000128	RGB 204236255	Izquierda	-	-	-
Subtitulo	<i>Times New Roman</i>	12	Negrita cursiva	RGB 000000000	RGB 204236255	Izquierda	-	-	-
Texto	<i>Times New Roman</i>	10	Normal	RGB 000000000	RGB 204236255	Justificada	-	Sencillo	Ninguna
Mensaje 1	<i>Century Gothic</i>	11	Negrita	RGB 000000128	RGB 204236255	Justificada	-	Sencillo	Ninguna

<sup>24</sup> Hermosa Andrea y Chica Fernanda. (2008). *Tesis de Grado Curso E-Learning de Planificación Informática*. Quito: Escuela Politécnica del Ejército.



Figura 3.24: Diagrama de Diseño

### **Disposición de Ventanas y Frames**

Se debe describir la disposición de las ventanas, su tamaño, ubicación de botones, etc. Todo esto se detalla en el siguiente formato:

### **Utilización de Plantillas**

Se conoce como plantillas a los modelos de páginas para crear el contenido del curso.

### **Plantilla para las Páginas del Contenido del Curso**

Consiste en plantillas preestablecidas que se utilizará para colocar el contenido de curso.

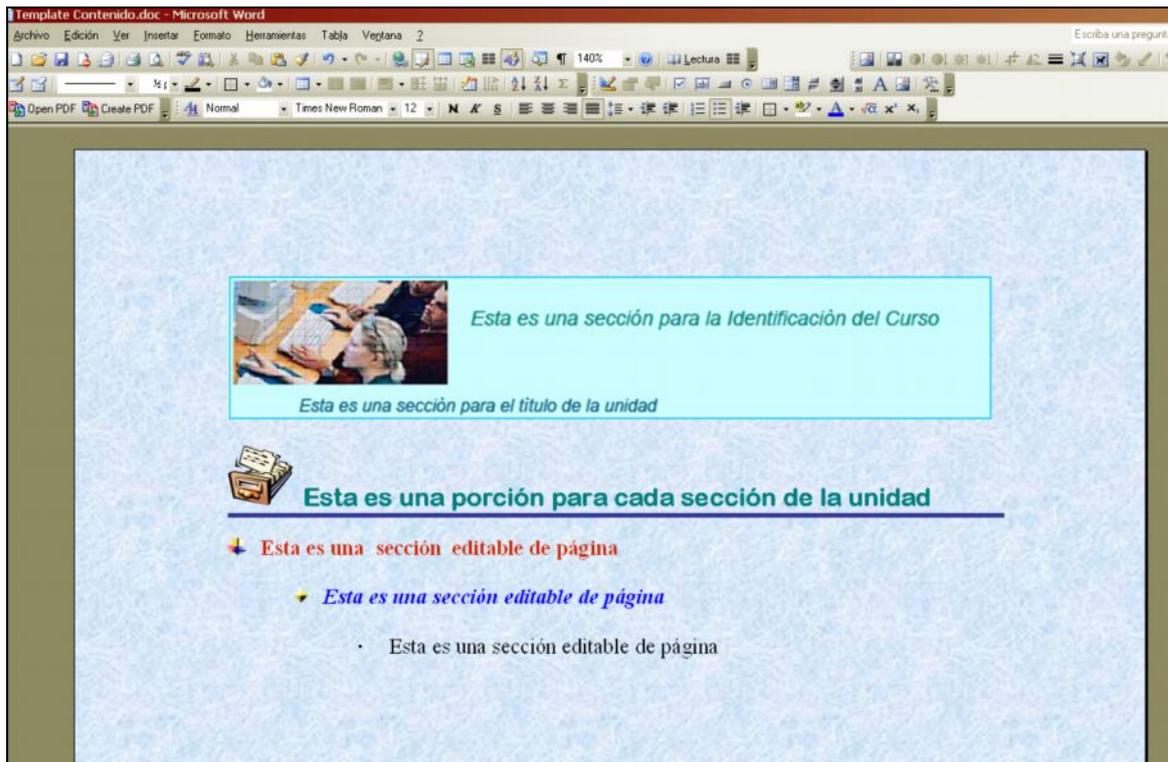


Figura 3.25: Plantilla de Contenidos del Curso <sup>25</sup>

### Plantillas para una página

La denominada plantilla para una página facilita el esfuerzo que se requiere para crear una página ya que tiene una disposición.

Además estandariza los elementos de las páginas del curso, los autores se orientan al contenido requerido para obtener el propósito de la página.

### Plantillas para Módulos de Contenido

Además de las plantillas para todas las páginas, se puede establecer plantillas particulares de módulos de contenido. Ejemplo: Formato complejo de tablas, una vez que se ha creado la plantilla se la puede usar en cualquier parte del curso.

<sup>25</sup> Hermosa Andrea y Chica Fernanda. (2008). *Tesis de Grado Curso E-Learning de Planificación Informática*. Quito: Escuela Politécnica del Ejército.

### **3.4.8.5. Pruebas de calidad y rendimiento del curso desarrollado**

#### **Procedimiento de Manejo**

Al inicio del curso el usuario ingresa con login y password, de inmediato iniciar el curso con sus respectivos menús, a interfaz es totalmente amigable para el usuario y la navegabilidad muy fácil.

En el curso hay imágenes que se han creado para adoptar aspectos de conceptualización de contenidos, mostrando al estudiante lo que se pretende enseñar en cada lección.

#### **Pruebas del Sistema**

Las pruebas establecerán la efectividad del uso del curso y además los posibles errores en la ejecución.

Para realizar las pruebas se debe cumplir los siguientes pasos:

- Recuperación
- Seguridad
- Resistencia
- Rendimiento

#### **Recuperación**

Se ha desarrollado pruebas que verifiquen las fortalezas del curso para los alumnos y profesores, estableciendo que no existen bloqueos. Se determina que la plataforma funciona bien, los problemas ocasionados son la falta de ancho de banda adecuado y la caída del sistema en las Tics; por esta razón para que el alumno acceda al curso cuando existan estos inconvenientes se desarrolló en CD de contenidos.

#### **Seguridad**

Se utilizan las características que ya dispone la plataforma Moodle como:  
*Ocultar Tareas:* La plataforma Moodle con esta opción permite ocultar y que las tareas sean visualizadas por los alumnos en fechas deseadas.

*Fechas límite:* Moodle permite agregar eventos a un calendario que será visualizado por usuarios individuales o grupos de estudiantes, adhiere fechas límite a actividades de aprendizaje (tareas, cuestionarios, foros, etc).

*Respaldos:* Moodle permite realizar copias de seguridad del sitio y del curso, para guardar los datos de forma confiable y que la recuperación sea en el menor tiempo posible.

### **Resistencia**

Comprueba la potencialidad del curso, debe ser utilizado al máximo para la respectiva evaluación tanto en hardware como software, debe usar los recursos mínimos para el correcto funcionamiento. Al momento se encuentran levantados varios cursos que están siendo utilizados como: Gestión de Tecnologías, Auditoría de Sistemas, los mismos cuentan aproximadamente con 40 alumnos cada uno. Alrededor de 200 alumnos ingresan a los cursos y no han encontrado problemas de caída de la plataforma, los problemas que han tenido son sólo de ancho de banda.

### **Rendimiento**

Comprobación de los sistemas operativos su funcionamiento, verificando los recursos necesarios para que luego no existan problemas o fallos. Con la ayuda del administrador de tareas se puede verificar el porcentaje de memoria y procesamiento cuando se ejecuta el curso.

### **Control de Calidad:**

Una vez terminado el curso y puesto en producción debe ser evaluado con alumnos y profesores de un curso presencial (en el período siguiente luego de terminada la tesis) para realizar los ajustes correspondientes como el tiempo de desarrollo de los cuestionarios, el número de alumnos asignados a cada tutor.

## CAPÍTULO 4

### CONTENIDO DEL CURSO DE BASE DE DATOS 1

Se detalla a continuación el contenido del Curso de Base de Datos 1, que se encuentra formado por 3 Unidades:

- **Unidad 1: Generalidades de la Base de Datos**

- Lección 1.1: Bases de Datos

- Lección 1.2: Sistemas de Gestión de Base de Datos

- Lección 1.3: Modelos de Datos

- Lección 1.4: Herramientas Case

- **Unidad 2: Procesos de Desarrollo de la Base de Datos**

- Lección 2.1: Modelo Entidad – Relación

- Lección 2.2: Modelo Relacional

- Lección 2.3: Lenguajes Relacionales

- Lección 2.4: Creación y Desarrollo de una Base de Datos

- Lección 2.5: Diseño Conceptual, Lógico y Físico

- Lección 2.6: Aplicación de una Herramienta Case

- Lección 2.7: Normalización

- Lección 2.8: Diccionario de Datos

- Lección 2.9: Implementación de una Base de Datos utilizando SQL – DDL

- **Unidad 3: SQL Lenguaje de Consulta Estructurado**

- Lección 3.1: SQL-DML

- Lección 3.2: Consultas sobre tablas y tablas relacionadas

- Lección 3.3: Funciones de Agregación

## Estructura del Curso

El curso B-Learning de Base de Datos I consta de la siguiente estructura:

1. Visión General
2. Plan de la Lección
3. Desarrollo de la Lección
4. Recursos de la Lección
5. Resumen
6. Actividades de Aprendizaje

En las diferentes lecciones se desarrollarán actividades de aprendizaje como:

- Autoevaluaciones
- Foros
- Tareas
- Cuestionarios

Las actividades de aprendizaje de este curso se aplicarán de acuerdo a las necesidades que se presenten en cada lección, existirán casos en los que se enviará varias tareas, también habrá lecciones en las que no exista foros, todo va a depender de la complejidad de cada lección. Las actividades de aprendizaje en algunos casos corresponderán a la **lección** (objeto de aprendizaje de nivel medio) y en otros a los **temas** (objetos de aprendizaje de nivel inferior).

# DESARROLLO DEL CURSO

## UNIDAD 1: GENERALIDADES DE LAS BASES DE DATOS

### COMPETENCIA A SER DESARROLLADA POR EL ESTUDIANTE:

Conceptualiza y compara los principios de base de datos y lenguajes de manejo de base de datos.

### LECCIÓN 1.1: Base de Datos



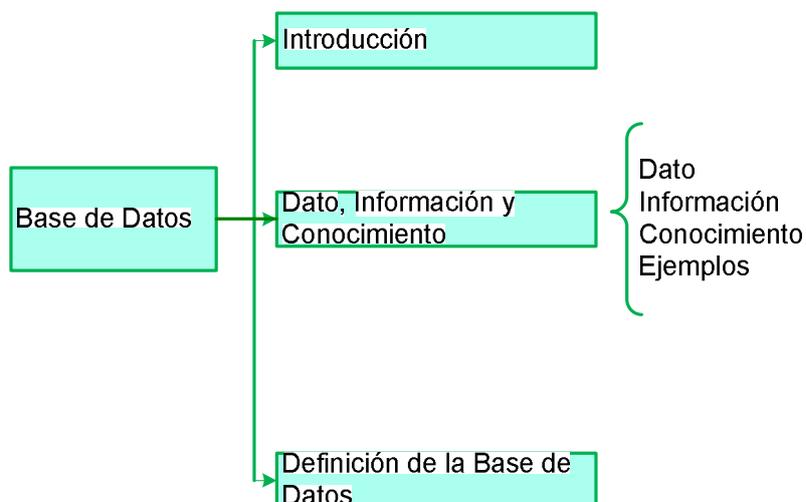
#### Visión General

En esta lección se tratan elementos generales de las bases de datos, tal que permitan al aprendiente contextualizar el ámbito de desarrollo de las bases de datos. Aquí:

[Volver Inicio](#)



#### Plan de la Lección





## Desarrollo de la Lección

### 1.1. Base de Datos

#### 1.1.1. Introducción

Una base de datos es un conjunto de datos consistente y usualmente persistente, organizado en un modo específico que permita acceder a la información de forma fácil y rápida. Las bases de datos actualmente son muy importantes para una organización por su función de almacenamiento de cantidades muy grandes de información

Las bases de datos son un contexto informático muy significativo por sus aplicaciones dentro de instituciones educativas en bibliotecas, nóminas, entidades bancarias, etc. Una biblioteca puede considerarse una base de datos compuesta en su por documentos y textos impresos en papel e indexados para su consulta. En la actualidad, y debido al desarrollo tecnológico de campos las bases de datos están en formato digital, que ofrece un amplio rango de soluciones al problema de almacenar datos.

Esta colección de datos cumple las siguientes propiedades:

- Están estructurados independientemente de las aplicaciones y del soporte de almacenamiento que los contiene.
- Presentan la menor redundancia posible.
- Son compartidos por varios usuarios y/o aplicaciones.

Existen programas denominados sistemas gestores de bases de datos (SGBD), que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de estos SGBD, así como su utilización y administración, se estudian en el contexto del presente curso.

### 1.1.2. Dato, Información y Conocimiento

Estos tres componentes son muy importantes dentro de una organización de éstos depende tomar decisiones, aunque sus definiciones en ocasiones no son totalmente claras, por lo que es adecuado comprender en qué consiste cada uno y mostrar sus diferencias.

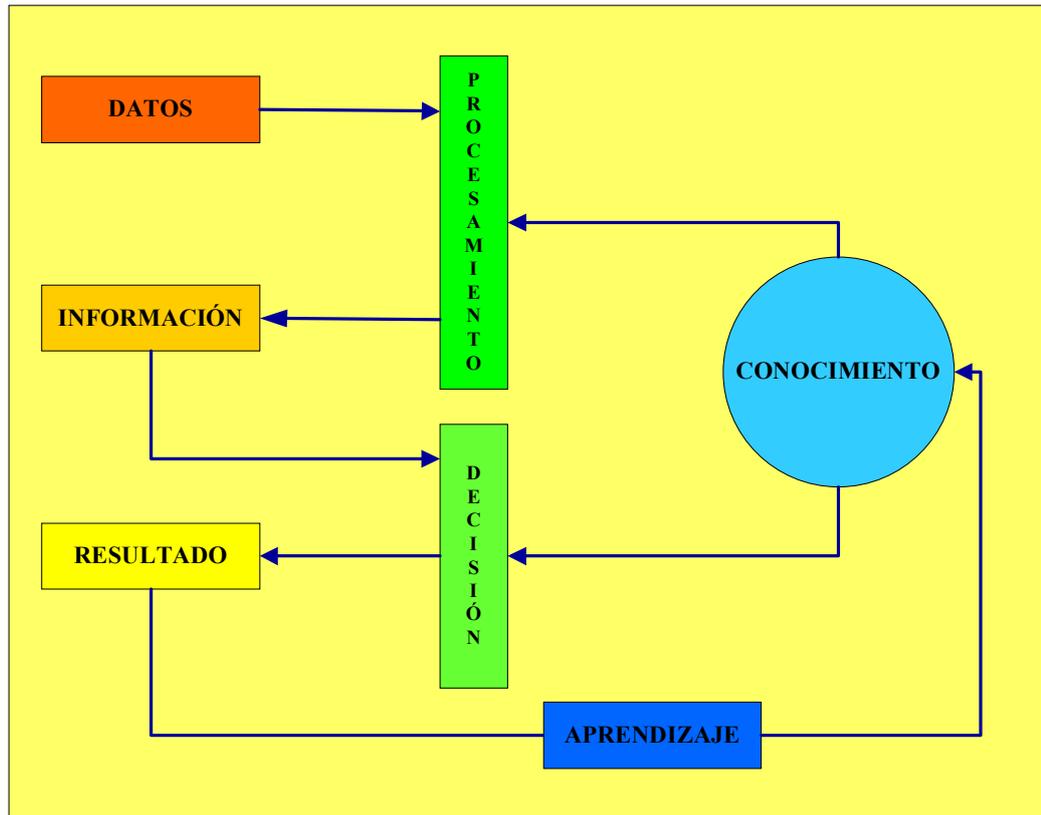


Figura 1.1.2\_1: Relación entre Datos, Información y Conocimiento

#### 1.1.2.1. Dato

Dato es la representación de hechos mediante símbolos (números, letras, etc.). Por si solo un dato no es capaz de tener un significado requiere de otros datos para que la colección de estos se convierta en información.

En ocasiones el dato es confundido con información por lo que es necesario comprender que si el dato no se transfiere o se lo manipula no se transforma en información o como se lo denomina mensaje.

En informática un dato es ingresado a un sistema, el conjunto organizado y procesado de éstos se convierte en información.

En la actualidad todas las instituciones requieren datos y dependen de ellos, las organizaciones diariamente manipulan millones de transacciones, los datos representan una parte de lo real, las organizaciones toman decisiones sobre los datos pero esto no significa que indique las medidas que la organización debe tomar sobre resultados de los datos.

**Por ejemplo:** un nombre, número telefónico, dirección así simplemente son datos que no tienen ningún argumento, finalidad o utilidad no sirven para la toma de decisiones.

#### **1.1.2.2. Información**

La información se denomina como la colección de datos, el concepto de información según investigaciones realizadas se detalla como un mensaje y como se conoce un mensaje consta de un emisor y un receptor. El receptor es quien verifica si el mensaje recibido realmente es una información es decir que contenga sentido y conste de un significado válido.

A modo general a la información se la conoce como la colección de datos que se encuentran organizados.

“Las necesidades de información de nuestra sociedad actual se dejan sentir de forma cada vez más imperiosa. El problema de la información está estrechamente relacionado con el desarrollo económico y social. La investigación, la planificación y la toma de decisiones exigen una información precisa, oportuna, completa, coherente y adaptada a las necesidades específicas de cada usuario y de cada circunstancia (características de la información).”<sup>26</sup>

---

<sup>26</sup> Piattini Mario G; Marcos Esperanza; Calero Coral y Vela Belén. (2006). *Tecnología y Diseño de Base de Datos*. México: Alfaomega

Como se mencionó anteriormente la información debe ser **precisa**, lo que significa la proporción de información acertada sobre la información global es decir sin errores, con lo que se obtendrá la credibilidad de la información frente al usuario.

Además la información debe ser **oportuna**, el tiempo que ha pasado desde que se ocasionó el dato hasta cuando la información es puesta a disposición de los usuarios.

La información debe ser **significativa**, es decir tener un contenido estructurado para que sea admitida como información verdadera y también la información debe ser **completa** para lograr los objetivos propuestos por una organización.

Es adecuado que la información tenga **seguridad** lo que comprende que debe ser confiable, debe estar disponible e íntegra.

**Por ejemplo:** describimos los datos del volcán Chimborazo como: altitud, latitud, situación geográfica, etc., entonces podríamos escribir un párrafo de un libro sobre Volcanes del Ecuador, esto constituye la información.

### **1.1.2.3. Conocimiento**

La diferencia entre información y conocimiento se encuentra en la magnitud de las experiencias que se requieren para llegar a conocer. La colección de experiencias establece el conocimiento que se vincula con las personas y se basa en datos e información.

Al conocimiento también se lo denomina como la colección de procesamientos y destrezas por medio de los cuales las personas están en la capacidad de resolver problemas.

Algunas características del conocimiento son:

- El conocimiento es propio de cada persona, es la asimilación del resultado de experiencias.
- Es una guía para las personas ayuda a tomar decisiones en el momento requerido, lo que permite optimizar los resultados.

#### 1.1.2.4 Ejemplo de datos, información y conocimiento

Una universidad de prestigio realizó una encuesta a alumnos de los sextos cursos de colegios de la ciudad de Quito, para conocer que motiva a escoger una carrera universitaria.

Para esto la universidad se valió de un grupo de estudiantes y profesores quienes realizaron las encuestas y procesaron los datos para obtener el estudio completo, para esto utilizaron una encuesta donde se recogió las apreciaciones de los estudiantes, relacionado a los siguientes factores: ubicación de la universidad y carreras que ofrece, calidad de educación de la universidad, años de estudio, costo de la carrera.

Los **datos** desarrollados y estructurados dieron la información significativa respecto a escoger una carrera universitaria siendo el factor principal el costo de la carrera (43%); seguido de la calidad de educación de las universidades (30%), luego la ubicación de la universidad y carreras que ofrece (18%) y por último los años de estudio de la carrera (9%). De acuerdo con la **información** del estudio y la experiencia de la universidad de realizó planes de captación de alumnos (**conocimiento**) como programas de becas académicas para los alumnos con mejores promedios, becas socioeconómicas, becas culturales, becas deportivas, o descuento por hermanos.

#### 1.1.3. Definición de Base de Datos

“Es una colección compartida de datos lógicamente relacionados, junto con una descripción de estos datos, que están diseñados para satisfacer las necesidades de información de una organización.”<sup>27</sup>

“Es una fuente central de datos destinados a compartirse entre muchos usuarios para una diversidad de aplicaciones.”<sup>28</sup>

---

<sup>27</sup> Connolly, Thomas M. y Begg, Carolyn E. (2005). *Sistemas de bases de datos - Un enfoque práctico para diseño, implementación y gestión*, Madrid: Pearson Addison Wesley.

<sup>28</sup> Kendall, Julie. (2005). *Análisis y Diseño de Sistemas*.

“Es un conjunto de datos persistentes que es utilizado por los sistemas de aplicación de alguna empresa dada.”<sup>29</sup>

“Colección o depósito de datos integrados, almacenados en soporte secundario y con redundancia controlada. Los datos, que han de ser compartidos por diferentes usuarios y aplicaciones, deben mantenerse independientes de estos y su definición (estructura de la base de datos) única y almacenada junto con los datos, se han de apoyar en un modelo de datos, el cual ha de permitir captar las interrelaciones y restricciones existentes en el mundo real. Los procedimientos de actualización y recuperación comunes y bien determinados, facilitará la seguridad del conjunto de datos.”<sup>30</sup>

“Una base de datos es una colección de datos estructurados según un modelo que refleje las relaciones y restricciones existentes en el mundo real. Los datos, son compartidos por diferentes usuarios y aplicaciones y deben mantenerse independientes de éstas. Asimismo, los tratamientos que sufran estos datos tendrán que conservar la integridad y seguridad”.<sup>31</sup>

[Volver Inicio](#)

---

<sup>29</sup> Date, C.J. (2006). *Introducción a los Sistemas de Bases de Datos*.

<sup>30</sup> Piattini, Mario; Marcos, Esperanza; Calero, Coral y Vela, Belén. (2006). *Tecnología y Diseño de Base de Datos*. México: Alfaomega

<sup>31</sup> Sabana, Maribel. (2006). *Modelamiento e Implementación de Base de Datos*: Lima: Megabyte Grupo Editorial.



## U1. Lección 1 tarea 1

### ACTIVIDAD DE APRENDIZAJE

---

Denominación de la tarea:

Definición de la base de datos

### ASESORÍA DIDÁCTICA

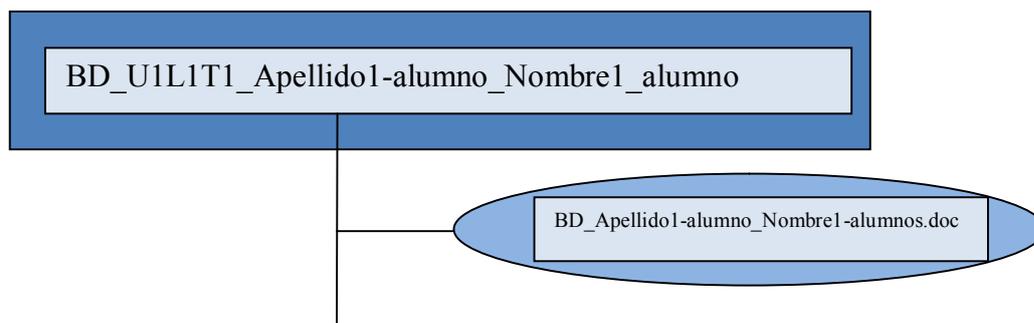
Para desarrollar la siguiente actividad de aprendizaje, debe leer y comprender el contenido de la lección 1.

### ACTIVIDAD DE APRENDIZAJE

Detalle un ejemplo real de base de datos (comercial, educación, etc), explique que son las bases de datos y para qué sirven.

Envíe a su tutor la tarea en una carpeta comprimida como .rar

Formato de la carpeta





## Resumen Lección 1

---

### Base de Datos

Esta lección ha presentado una visión global de base de datos, incluyendo:

- **Dato, Información y Conocimiento**
- **Definición de la Base de Datos**

#### **Destacándose que:**

El dato, la información y el conocimiento son términos totalmente diferentes. Dato es la representación de hechos mediante símbolos. La información se denomina como la colección de datos. El conocimiento es la colección de procesamientos y destrezas por medio de los cuales las personas están en la capacidad de resolver problemas.

Existen varias definiciones de base de datos como:

Es la colección de datos relacionados lógicamente que se diseñan para satisfacer las necesidades de información de una determinada organización.

Otra definición de base de datos es una colección de datos estructurados según un modelo que refleje las relaciones y restricciones del mundo real. Los datos, son compartidos por diferentes usuarios y aplicaciones, y deben mantenerse independientes de éstas. Asimismo, los tratamientos que sufran estos datos tendrán que conservar la integridad y seguridad.

# UNIDAD 1: GENERALIDADES DE LAS BASES DE DATOS

## LECCIÓN 1.2: Sistemas de Gestión de Base de Datos



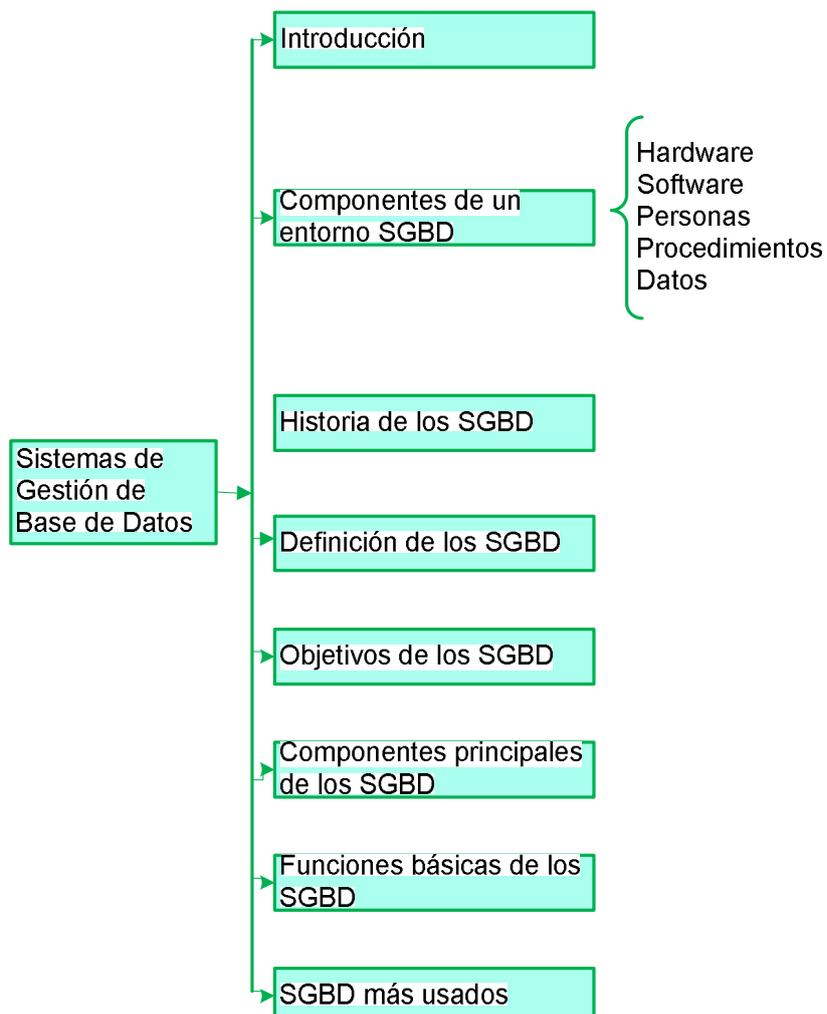
### Visión General

En esta lección se tratan los Sistemas de Gestión de Bases de Datos, sus componentes, funciones, objetivos.

[Volver Inicio](#)



### Plan de la Lección





## Desarrollo de la Lección

### 1.2. Sistemas de Gestión de Base de Datos

#### 1.2.1. Introducción

Los Sistemas de Gestión de Base de Datos permiten manejar de forma fácil y ordenada una colección de datos que luego se transformarán en importante información para obtener una adecuada administración de los datos. Actualmente los SGBD se han transformado en el software de procesamiento de datos más usado, existen SGBD libres como: MySQL, Postgresql, etc y comerciales como: Oracle, Microsoft SQL Server, DB/2, etc.

#### 1.2.2. Definición del SGBD

Un **Sistema Gestión de Bases de Datos –SGBD-** (Data Base Management System DBMS).

Es un sistema que facilita al usuario el acceso a un conjunto de datos interrelacionados, el SGBD interactúa con los programas de aplicación del usuario y con la base de datos.

#### 1.2.3. Componentes de un entorno SGBD

El Sistema de Gestión de Base de Datos según la administración general tiene cinco elementos importantes: Hardware, Software, Personas, Procedimientos y Datos.

##### 1.2.3.1. Hardware

Son los elementos físicos de un sistema donde se ejecutará el SGBD y sus aplicativos, puede ser desde un computador hasta un mainframe, además el hardware comprende todos los dispositivos de entrada y salida además los componentes electrónicos que sirven para armar una red de computadoras.

### **1.2.3.2. Software**

Este componente incluye, el sistema operativo como Unix, Windows, OS/2, etc. Además el software del SGBD que es el encargado de administrar las bases de datos, como: Oracle, Microsoft SQL Server, DB2, etc. También incluye los aplicativos y utilerías para ingresar a los datos del SGBD y manipular los mismos.

### **1.2.3.3. Personas**

Este elemento contiene a los usuarios del sistema de base de datos y son de cinco clases:

- *Administradores de sistemas*: controlan las ejecuciones del sistema en forma general.
- *Administrador de la base de datos*: son conocidos como DBA, controlan el funcionamiento del SGBD y su administración.
- *Creadores de base de datos*: son los que esquematizan la estructura de la base de datos, del esquema depende la eficiencia del sistema.
- *Programadores de sistema*: realizan procesos, reportes, para que el usuario final tenga acceso y maneje los datos del sistema.
- *Usuarios finales*: son quienes usan los programas y efectúan diariamente operaciones.

### **1.2.3.4. Procedimientos**

Es un elemento que contiene instrucciones para el esquema y utilización de la base de datos, cumple una función importante ya que permiten el cumplimiento de las normas por las que se rige una organización tanto interna como externamente en el caso de los clientes.

### **1.2.3.5. Datos**

Este componente es la representación de hechos mediante símbolos que se guardan en el sistema de base de datos, los creadores de la base de datos establecen qué datos son los que se implantan y su forma de organización.

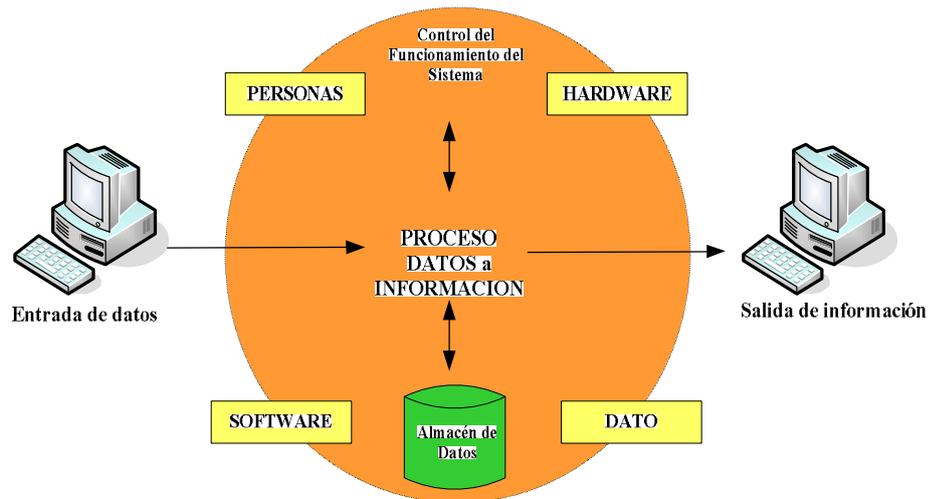


Figura 1.2.3.5\_1: Componentes de un entorno de SGBD

#### 1.2.4. Historia de las SGBD

Las bases de datos pretenden acumular cantidades inmensas de datos organizados cumpliendo con esquemas determinados que permitan el acceso a los datos, su recuperación y modificación.

A finales de los años sesenta nacen las bases de datos, el cuadro siguiente muestra una breve síntesis de la evolución de mismas:

Tabla 1.2.4\_1: Historia de los SGBD

Año	Descripción
1960 – 1969	<p>En esta década prevalece el uso de discos esto permitía acceder a los datos directamente y disminuir tiempo de trabajo, además con esto se inició las bases de datos de red y jerárquicas ya que era posible guardar estructuras de datos como listas y árboles.</p>
1970 – 1979	<p>En el año 1970 Edgar Frank Codd creó el modelo relacional. Aparecieron las bases de datos relacionales.</p> <p>En esta década aparecen los SGBD como: DB2, Oracle, Informix, Sybase.</p> <p>Codasyl en el año de 1978 planteó el Lenguaje de Definición de Datos (DDL) y el Lenguaje de Manipulación de Datos (DML).</p>
1980 – 1989	<p>En esta década prevalecen las bases de datos relacionales sobre las bases de datos jerárquicas y de red por el nivel muy sencillo y bajo de éstas últimas. Se inicia las bases de datos distribuidas.</p> <p>Estándares SQL (ANSI, ISO).</p> <p>Se da inicio a las bases de datos orientada a objetos.</p>
1990 - 1999	<p>Se crea el lenguaje de consulta estructurado SQL.</p> <p>Se profundiza las bases de datos orientada a objetos</p> <p>Arquitectura Cliente – Servidor (2 capas)</p> <p>Con la aparición en esta década de la Word Wide Web (www) se facilitó las consultas de bases de datos con la amplia capacidad de almacenamiento de información.</p>
2000 – 2009	<p>Modelo Objeto – Relacional</p> <p>Arquitectura Cliente – Servidor (3 capas)</p>

### 1.2.5. Estado de arte SGBD

Oracle liberó recientemente una actualización de seguridad de varios de sus productos a los que se ha denominado “Critical Patch Update” (Oracle Database, Oracle Application Server, Oracle Secure Backup, Oracle Open Office y StarOffice, Oracle E-Business Suite, Oracle Sun, Oracle WebLogic Server, Oracle PeopleSoft Enterprise, etc.), con lo cual se resuelve las vulnerabilidades encontradas. Oracle ha recomendado instalar inmediatamente ya que usuarios malignos están afectando la integridad de los sistemas.

Además Oracle presentó el módulo de inteligencia de negocios denominado **Oracle Financial Analytics para SAP**, es parte de la versión 7.9.7 de Oracle BI Applications, permite realizar análisis financieros.

Por otra parte Microsoft lanzará la versión SQL Server 2011 denominada Denali, a la fecha (Enero 2011) se encuentra en estado beta, se puede descargarla pero únicamente funciona sobre versiones de Windows desde Vista SP2.

Lo nuevo de esta versión es:

- **Sequences:** “Objeto cuyo valor se autoincrementa cada vez que es consultado. Es un excelente reemplazo de los identities, ya que es un objeto externo a la tabla. Dentro de poco veremos las flamewars en blogs argumentando si es mejor o peor usar sequences o identities como campo PK de una tabla. Sin embargo, la incorporación de este objeto facilitara enormemente la migración de Oracle y PostgreSQL a SQL Server.
- **Paginación:** Con SQL Server 2011 va a ser posible paginar nativamente una query declarando un OFFSET en la cláusula ORDER BY”<sup>32</sup>

---

<sup>32</sup> <http://grimpipev.wordpress.com/2011/01/18/lonuevoensqlserver2011/>

**Data warehouse**, es la información almacenada en la base de datos donde existe información transaccional y operacional. Es diseñada para análisis eficientes de datos (en lo referente a OLAP, que es el procesamiento en línea).

#### 1.2.6. Objetivos de los Sistemas de Gestión de Base de Datos

Los objetivos principales de un sistema de base de datos son los siguientes:

- **Controlar la información centralizadamente.** Acceso a la base de datos por medio de un único programa –administrador de bases de datos.
- **Disminuir la redundancia y evitar la inconsistencia.** Al disminuir la redundancia se trata de evitar que los datos agrupados en un mismo objeto se repitan, pero al no poder controlar esto y la información se duplica aparece la inconsistencia.
- **Disponer de un acceso seguro.** Asignar claves y considerar el perfil de acceso de cada usuario; por ejemplo: el perfil de usuario de un profesor en un sistema de registro de notas no es el mismo que de un alumno, ya que el profesor tiene acceso a ingresar notas, el alumno únicamente puede revisar sus notas.
- **Proporcionar independencia de datos.** Se pretende separar los programas y datos física y lógicamente de manera que cualquier cambio en las estructuras no afecte a las aplicaciones.
- **Mantener la integridad.** Llevar controles que comprueben que los datos ingresados sean correctos, realizando las respectivas validaciones.

### 1.2.7. Funciones Básicas de SGBD

Un SGBD realiza funciones importantes entre las cuales se encuentran las siguientes:

1. *Administración del diccionario de datos.* El SGBD requiere que los elementos de datos y sus relaciones se almacenen en un diccionario de datos y que todas las aplicaciones que pueden acceder a los datos de la base de datos funcionen a través del SGBD. Los cambios que se realicen en la organización de la base de datos quedan registrados en el diccionario de datos. El diccionario de datos es elemento muy importante dentro del SGBD y es un repositorio de información que detalla los datos de la base de datos, la cantidad de datos varía de un SGBD a otro. El tema de diccionario de datos se detallará en la Lección 6 de la Unidad 2.
2. *Administración del almacenamiento de datos.* El SGBD administra el almacenamiento de datos con lo que resulta más fácil para el usuario, no solo permite almacenar datos sino también tiene otras funciones como formularios de ingreso, definiciones de reportes, código de procedimientos, formatos de imagen y video, etc.
3. *Transformación y presentación de datos.* El SGBD libera al usuario de diferenciar el formato lógico del formato físico de los datos, al brindar programas de aplicación con independencia de software y abstracción de datos. El SGBD convierte las solicitudes lógicas en instrucciones que localizan físicamente y recuperan los datos.
4. *Administración de la seguridad.* El SGBD establece un sistema de seguridad tanto para el usuario y para los datos, las normas de seguridad fijan que usuario accede a la base de datos, a qué datos y a qué operaciones (leer, agregar, modificar, eliminar).

5. *Control de acceso de usuarios múltiples.* El SGBD establece estructuras que dan acceso a usuarios múltiples a la base de datos sin complicar la integridad de la base de datos y permitiendo tener consistencia y seguridad en los datos.
6. *Administración de tareas de respaldo y recuperación.* El SGBD brinda herramientas que permiten al DBA realizar respaldos y recuperación rutinariamente, administración de recuperación de la base de datos luego de fallos, garantizando la seguridad e integridad.
7. *Administración de la integridad de los datos.* El SGBD cumple las normas para eliminar los inconvenientes de integridad de datos, lo que proporciona la reducción de redundancia de los datos y el aumento de su consistencia.
8. *Lenguajes de acceso a base de datos e interfaces de programación de aplicaciones.* El SGBD permite el acceso a los datos por medio del lenguaje de consulta, este contiene:  
Lenguaje de definición de datos (DDL), concreta las estructuras donde se alojan los datos.  
Lenguaje de manipulación de datos (DML), extrae los datos de la base de datos.
9. *Interfaces de comunicación de base de datos.* El SGBD debe integrarse con software de comunicaciones. Gran parte de los usuarios se conectan a la base de datos desde estaciones de trabajo que pueden estar conectadas directamente, en otros casos se conectan remotamente y se comunican a través de una red al SGBD.

Además tiene funciones de comunicación para acceder a la base de datos a través de internet por medio de los diferentes exploradores.

### 1.2.8. SGBD más usados

Tabla 1.2.8\_1: SGBD más usados

CARACTERÍSTICAS	ORACLE	MICROSOFT SQL SERVER	MYSQL	DB2	POSTGRESQL
Multiplataforma (Cuando una base de datos funciona en diversas plataformas)	✓	Solo se instala sobre Windows	✓	✓	✓
Escalable (Capacidad de cambiar el tamaño o configuración para adaptarse a las circunstancias cambiantes)	✓	✓	✓	✓	✓
Estabilidad (Una base de datos es estable cuando su nivel de fallos disminuye por debajo de un determinado umbral)	✓	✓	✓	✓	✓
Soporta Data Warehouse (Data Warehouse es una colección de datos en donde se integra la información de una organización que sirve para toma de decisiones gerenciales)	✓	✓	✓	✓	✓
Versiones 2011	11g	2008	5.5	9.7	9.0
Tamaño máximo de base de datos que soporta	La última versión soporta 4 petabytes	✓	65536 terabytes		64 TB

Fabricante	<u>Oracle Corporation</u>	Microsoft	MySQL AB	IBM	PostgreSQL Global Development Group
Costos de licenciamiento	El costo de la licencia comercial es alto	El costo de la licencia comercial es medio	El costo de la licencia comercial es muy bajo	El costo de la licencia comercial es alto	No tiene costos de licenciamiento
Licencia Comercial	✓	✓	Existe una versión completa de MySQL que es pagada.	✓	No tiene productos bajo licenciamiento
Licencia Libre	La versión ORACLE Express Edition es gratuita	La versión SQL Server Express Edition es gratuita	Usa la licencia GPL (licencia general pública) para aplicaciones no comerciales.	DB2 Express-C es la versión gratuita soportada por la comunidad de DB2 que permite desarrollar, implementar y distribuir aplicaciones	Bajo licencia BSD (es una licencia de software libre permisiva. Esta licencia tiene menos restricciones en comparación con otras como la GPL



## U1. Lección 2 tarea 1

### ACTIVIDAD DE APRENDIZAJE

---

Denominación de la tarea: **Realice una consulta sobre las versiones actuales de cinco SGBD.**

### ASESORÍA DIDÁCTICA

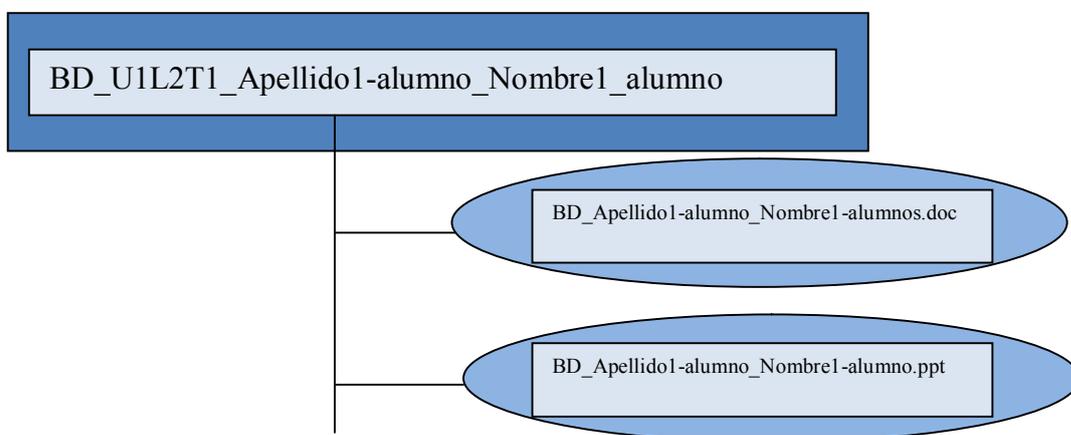
Para desarrollar la siguiente actividad de aprendizaje, debe leer y comprender el contenido de la lección 2.

### ACTIVIDAD DE APRENDIZAJE

Realice la consulta sobre las últimas versiones de los SGBD y su respectiva presentación.

Envíe a su tutor la tarea en una carpeta comprimida como .rar

Formato de la carpeta





## U1. Lección 2 tarea 2

### ACTIVIDAD DE APRENDIZAJE

---

Denominación de la tarea: **Realice una entrevista a tres usuarios de SGBD y verifique lo siguiente:**

#### ASESORÍA DIDÁCTICA

Para desarrollar la siguiente actividad de aprendizaje, debe leer y comprender el contenido de la lección 2.

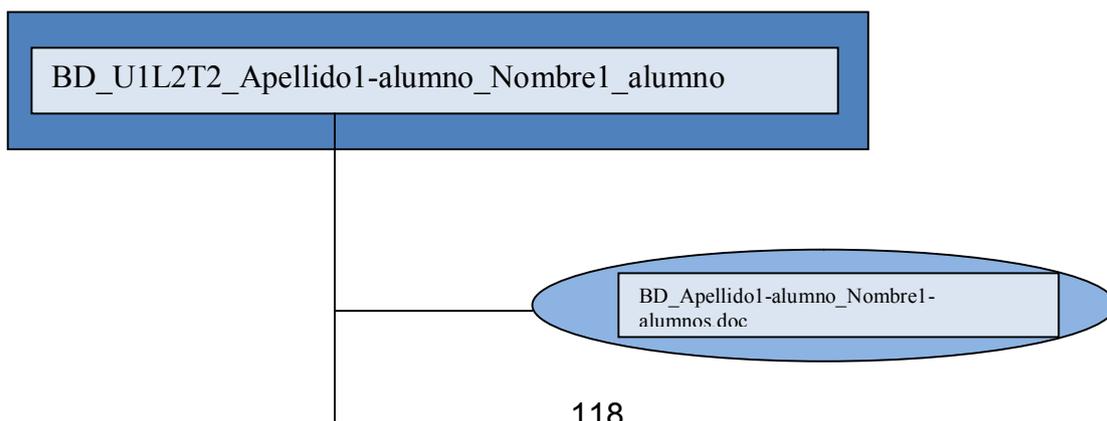
#### ACTIVIDAD DE APRENDIZAJE

Realice una entrevista a tres usuarios de SGBD y verifique las preguntas:

- **Para cada usuario qué características de los SGBD encuentran más útiles y por qué**
- **Estos usuarios que piensan sobre las ventajas y desventajas de los SGBD que usan**

Envíe a su tutor la tarea en una carpeta comprimida como .rar

Formato de la carpeta:





## U1. Lección 2 tarea 3

### ACTIVIDAD DE APRENDIZAJE

---

Denominación de la tarea: **Instalación de un SGBD**

#### ASESORÍA DIDÁCTICA

Para desarrollar la siguiente actividad de aprendizaje, debe leer y comprender el contenido de la Lección 2 Unidad 1.

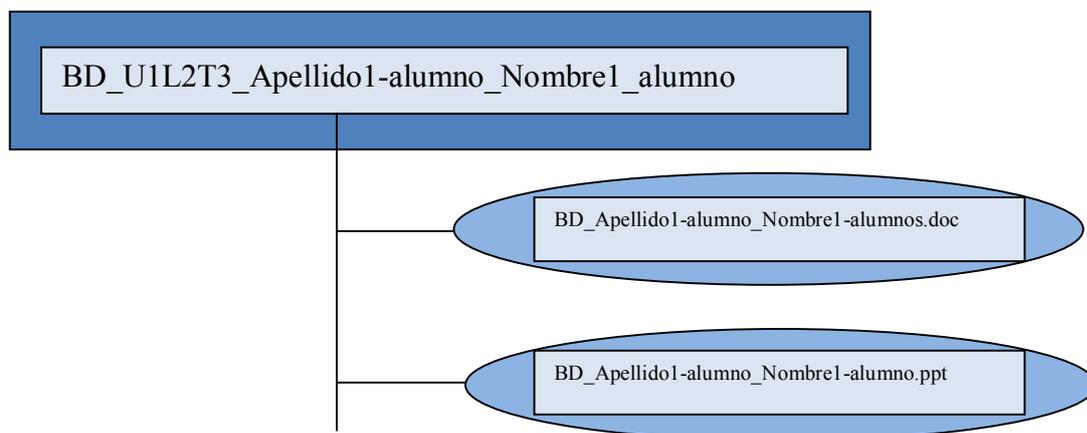
#### ACTIVIDAD DE APRENDIZAJE

Realice la instalación de un SGBD (Oracle, SQL Server, MySQL, etc).

Capture pantallas y explique cada una.

Envíe a su tutor la tarea en una carpeta comprimida como .rar

Formato de la carpeta





## Resumen Lección 2

---

### Sistemas de Gestión de Base de Datos

Esta lección ha presentado una visión global de base de datos, incluyendo:

- **Componentes de un entorno SGBD**
- **Historia de los SGBD**
- **Definición del SGBD**
- **Objetivos de los Sistemas de Gestión de Base de Datos**
- **Componentes principales del SGBD**
- **Funciones Básicas del SGBD**
- **SGBD más usados**

#### **Destacándose que:**

Los Sistemas de Gestión de Base de Datos es un sistema que facilita al usuario el acceso a un conjunto de datos interrelacionados, el SGBD interactúa con los programas de aplicación del usuario y con la base de datos.

Sus componentes de entorno son hardware, software, personas, datos y procedimientos. El inicio de las base de datos se da en el año 1960 y se ve cómo ha evolucionado hasta la actualidad.

Los objetivos de las bases de datos son:

- Disminuir la información centralizadamente
- Disminuir la redundancia y evitar la inconsistencia
- Disponer de un acceso seguro
- Proporcionar independencia de datos
- Mantener la integridad

Los componentes principales del SGDB son:

- Procesador de consultas
- Gestor de base de datos
- Gestor de archivos
- Procesador DML
- Compilador DDL
- Gestor de catálogo

Y finalmente las funciones principales del SGBD son:

- Administración del diccionario de datos
- Administración del almacenamiento de datos
- Transformación y presentación de datos
- Administración de la seguridad
- Control de acceso de usuarios múltiples
- Administración de tareas de respaldo y recuperación
- Administración de la integridad de los datos
- Lenguajes de acceso a base de datos e interfaces de programación de aplicaciones
- Interfaces de comunicación de base de datos

# UNIDAD 1: GENERALIDADES DE LAS BASES DE DATOS

## LECCIÓN 1.3: Modelos de Datos



### Visión General

En esta lección se tratan los modelos de datos su clasificación basada en registros y en objetos.

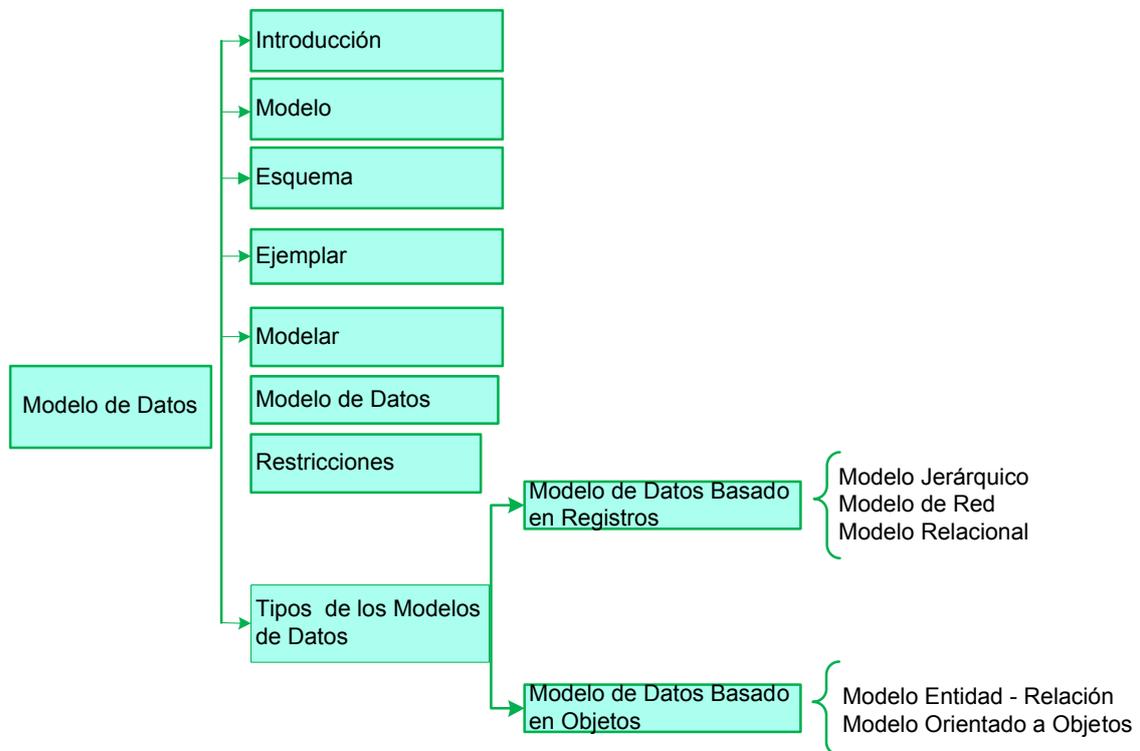
#### Competencias que adquiere el Aprendiziente:

- a) Conoce los modelos de datos y su importancia.
- b) Compara los Modelos Lógicos Basados en Registros y los Modelos Basados en Objetos.

[Volver Inicio](#)



### Plan de la Lección





## Desarrollo de la Lección

### 1.3.1. Introducción

Mediante estos modelos se representa a los datos, lo que permitirá realizar el diseño de la base de datos de forma adecuada. Un modelo de datos permite tener notaciones que sean entendibles tanto para los diseñadores como para los usuarios, lo que facilitará la comunicación precisa y la comprensión de los datos en la organización.

### 1.3.2. Modelo

Es una colección de abstracciones que forman un enfoque de una parte del mundo. En el mundo de las bases de datos se usa la simbología gráfica. Se convierte en la herramienta que se utiliza en parte del mundo real para lograr una organización de datos a la que se conoce como esquema.

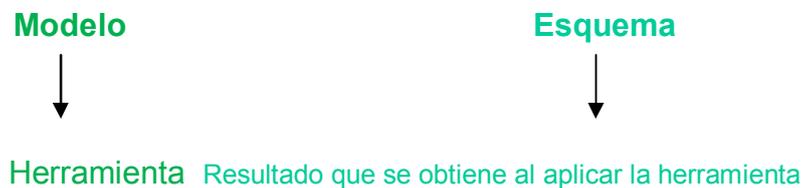


Figura: 1.3.2\_1: Modelo vs Esquema

### 1.3.3. Esquema

“Es la representación de un determinado mundo real en términos de un modelo de datos”<sup>33</sup>

### 1.3.4. Ejemplar

Es un componente de un esquema (datos que se almacenan en un esquema.)

---

<sup>33</sup> Piattini Miguel (1999). *Fundamentos y Modelos de Base de Datos*. Ra-Ma

### 1.3.5. Modelar

Se fundamenta en detallar un mundo abstracto que permita obtener conclusiones que correspondan al mundo real. Para el diseño de una base de datos se debe concretar primero el universo de discurso (herramienta que se aplica en el mundo real), estableciendo propósitos de acuerdo al mundo real.



Figura 1.3.5\_1: Qué es modelar

### 1.3.6. Modelo de Datos

“Se define a un modelo como la descripción analógica utilizada para visualizar algo que no puede observarse directamente”.<sup>34</sup>

“Un modelo de datos es la representación relativamente simple generalmente gráfica, de estructuras de datos complejas del mundo real, su principal función es ayudar en entender las complejidades del ambiente del mundo real”.<sup>35</sup>

“Es una colección de herramientas conceptuales que permiten describir los datos, las relaciones que existen entre ellos, la semántica asociada a los datos y restricciones de consistencia”.<sup>36</sup>

#### 1.3.6.1. Importancia de los Modelos de Datos

Un modelo de datos es importante por los siguientes aspectos:

- Facilita la interoperabilidad entre el diseñador, el desarrollador de aplicaciones y el cliente.
- Cuando el modelo de datos es generado correctamente se comprende a la organización para la cual se hizo el modelado de la base de datos.

<sup>34</sup> Rob, Peter y Coronel, Carlos. (2006). *Sistemas de bases de datos: diseño, implementación y administración*. España: Thomson.

<sup>35</sup> Rob, Peter y Coronel, Carlos. (2006). *Sistemas de bases de datos: diseño, implementación y administración*. España: Thomson.

<sup>36</sup> Sabana, Maribel. (2006). *Modelamiento e Implementación de Base de Datos*: Lima: Megabyte Grupo Editorial.

- Si se construye un buen esquema de la base de datos se obtendrá aplicaciones acertadas al hablar de aplicaciones nos referimos a que éstas operan los datos para convertirlos en información.
- Cada una de las personas que trabajan en una compañía tienen un enfoque diferente de los datos, los desarrolladores de aplicaciones tienen otra perspectiva de los datos un aspecto importante que toman en cuenta es la reportología, por lo que se debe llevar el control de un plano de ejecución de la base de datos.
- Si el plano de ejecución es desarrollado acertadamente no interesa la perspectiva que tengan los miembros de una compañía y la del desarrollador de aplicaciones pero si no habrá inconvenientes no se puede crear una base de datos sin antes tener un modelo apropiado.

### **1.3.7. Mecanismo de Abstracción**

De acuerdo al nivel de abstracción, que se muestra en la arquitectura ANSI, los modelos de datos se clasifican en:

- Externos: Este modelo verifica el enfoque de un usuario en particular.
- Globales: Modelo que se basa en el enfoque de varios usuarios
- Internos: Se toma en cuenta la implementación física como sistema operativo, archivos, etc.

Los modelos Externos y Globales se conocen como modelos lógicos ya que detallan aspectos lógicos de datos.

Los modelos Internos describen aspectos físicos de los datos.

La abstracción es un proceso que permite el modelamiento de datos centrándose en lo fundamental, sin tomar en cuenta aspectos que no se considera importante para representar al mundo real.

Los mecanismos de abstracción facilitan la representación de datos, el esquema es la consecuencia de la aplicación de un proceso de abstracción en el mundo real.

Los mecanismos más importantes son:

#### **1.3.7.1. Clasificación:**

Este mecanismo escoge características comunes a un conjunto de instancias (ocurrencias), creando categorías de las que son parte estas instancias. Por ejemplo:

#### **1.3.7.2. Agregación:**

Este mecanismo construye un nuevo componente del modelo formado por otros elementos.

Los tipos de agregación son:

- **Agregación de propiedades individuales para describir una clase**  
Esta es aceptada por todos los modelos de datos.
- **Agregación de propiedades para obtener una propiedad compuesta**  
Es aceptada por ciertos modelos de datos como Codasyl
- **Agregación de clases**  
Lo que permite adquirir una clase compuesta, esta agregación la incluyen los modelos semánticos: E-R, UML.

#### **1.3.7.3. Generalización:**

Este mecanismo admite la abstracción de características comunes de varias clases conocidas también como subclases las que sirven para construir una clase general que es la superclase.

#### **1.3.7.4. Asociación:**

Este mecanismo se usa para unir dos o más clases (esto incluye las instancias y ejemplares) de lo cual se obtiene un componente de tipo diferente.

## 1.3.8. Restricciones

### 1.3.8.1. Clasificación

Las restricciones son las reglas que imposibilitan que algunos objetos se consoliden en una base de datos. Son de dos tipos:

- **Restricciones Inherentes:** del propio modelo o estructura, si se toma el caso del modelo jerárquico la estructura usada es árbol cuya relación entre padres e hijos son 1:n y no se puede representar como m:n.
- **Restricciones de Integridad Semánticas (RIS) o explícitas,** por ejemplo: restricciones de rango y ecuaciones lógicas.

Son:

- **Propias del Modelo de Datos (semántica integrada):** La definición es del diseñador, el modelo de datos es responsable de la gestión.
- **Ajenas al modelo de datos (semántica dispersa):** La responsabilidad es completamente del diseñador, el modelo de datos no los reconoce ni muestra herramientas para manejarlos.
- **Restricciones Propias:** Se detallan cuando se describe un esquema por medio del lenguaje de definición de datos (DDL) se almacena en la metabase de datos, no puede ser violada por ninguna aplicación, las actualizaciones deben ser respetadas.

Las acciones son:

- **Acción General:** Se debe programar un determinado procedimiento que establezca lo que se debe realizar. Se divide en:
  - **Procedimientos almacenados:** Se determinan en forma procedimental.
  - **Disparadores (triggers):** Se expresa una condición de manera declarativa y al efectuar una proposición lógica, el cumplimiento dispara la acción.

- **Acción Específica:** La acción esta expresada en la misma restricción. Se divide en dos restricciones:
  - **Condición General:** Se especifica por una proposición lógica, por medio de una actualización, existen dos tipos de esta condición la *Verificación* (Ejemplo: la cláusula CHECK en un CREATE TABLE) y la *Aserción* (Ejemplo: CREATE ASSERTION).
  - **Condición Específica:** Las formas que hacen más sencillo a los modelos de datos cuando se detallan los componentes del esquema que son restricciones. Ejemplo: Las cláusulas PRIMARY KEY, NOT NULL.

### 1.3.8.2. Componentes de una restricción

Los siguientes componentes se muestran en las condiciones de integridad:

- *Operación* de inserción (actualización, modificación y borrado)
- *Condición* que es un proposición lógica definida sobre esquemas que puede ser verdadero o falso.
- *Acción* la cual se realiza dependiendo del resultado de la condición.

### 1.3.9. Tipos de Modelos de Datos

#### 1.3.9.1. Modelos Lógicos Basados en Registros

Este tipo de modelos se usan para detallar el entorno general de la base de datos y una representación de la implementación de la base en más alto nivel.

Actualmente la tendencia de los sistemas comerciales están basados en el modelo relacional y sistemas antiguos se han basado en registros.

- **Modelo Jerárquico**

La estructura de este modelo es en forma equivalente a un árbol, son modelos obsoletos un ejemplo de éstos son versiones anteriores: de Adabas (Adaptable Database System), Focus.

Son modelos no admitidos en la actualidad y su principal desventaja es la imposibilidad de interpretar la redundancia de datos. Se debe tener en cuenta que en este modelo un padre tiene muchos hijos pero un hijo tiene un único padre.

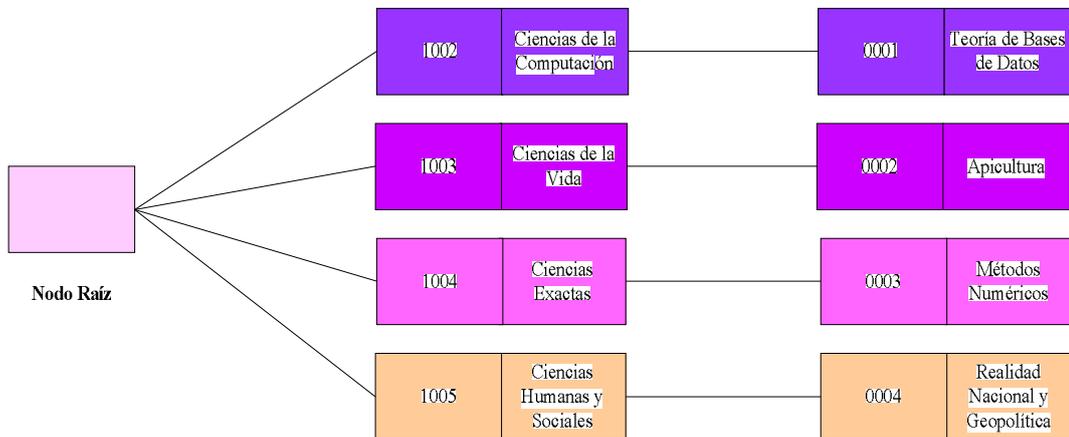


Figura 1.3.9.1.\_1: Modelo Jerárquico

- **Modelo de Red**

El creador de este modelo fue Charles Bachman, este modelo se caracteriza principalmente porque consiste en que un mismo nodo tenga varios padres, esto permitirá que los nodos hermanos tengan relación.

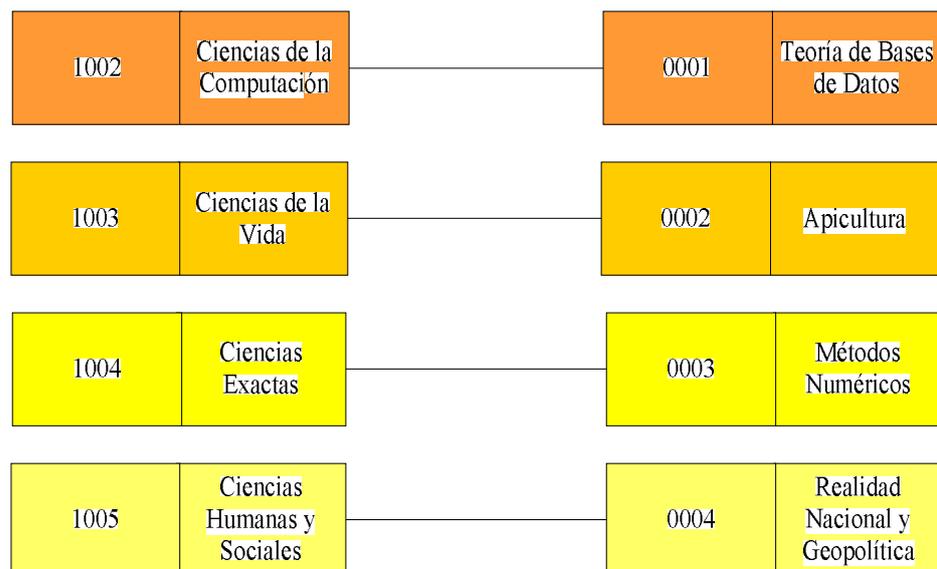


Figura 1.3.9.1\_2: Modelo de Red

## ▪ Modelo Relacional

El modelo relacional fue diseñado por E.E.Codd en 1970, éste modelo representa las relaciones entre entidades, los datos y las relaciones se simbolizan por tablas, además puede conectar datos de diversas tablas para que los usuarios tengan la información requerida. Las características más importantes del modelo relacional son:

- Un atributo únicamente toma un solo valor del dominio, no se permite múltiples valores.
- En la base de datos, la información es presentada como datos claros, no hay apuntadores entre las tablas.

El modelo relacional se forma de:

- Estructura de datos: Se forma de relaciones.
- Manipulación de datos: Es una colección de operadores que sirven para la recuperación y modificación de los datos.
- Integridad de datos: Es un conjunto de normas que precisan en la base de datos la consistencia.

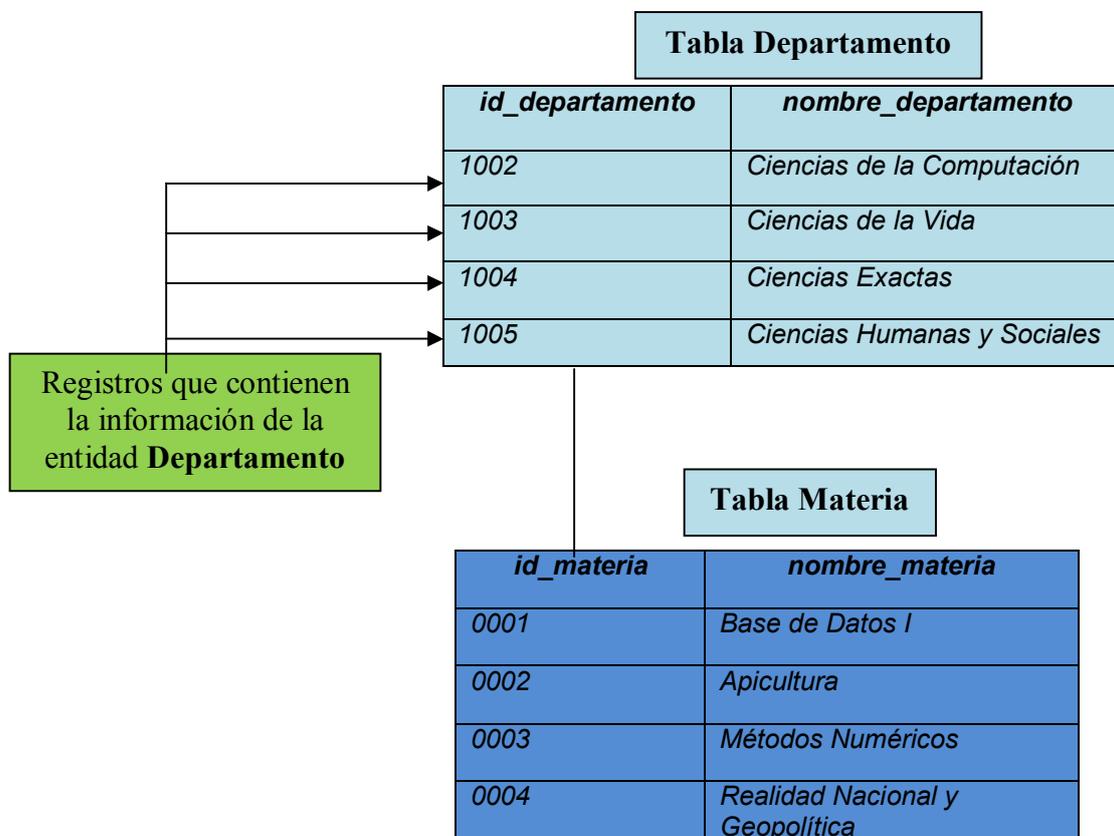


Figura 1.3.9.1\_3: Modelo Relacional

### **1.3.9.2. Modelos Lógicos Basados en Objetos**

Los modelos de datos basados en objetos utilizan conceptos tales como entidades, atributos y relaciones. Una entidad es un objeto singular (una persona, lugar, cosa, concepto, suceso) dentro de una organización hay que representar mediante una base de datos. Un atributo es una propiedad que describe algún aspecto del objeto suficientemente relevante como para registrarlo y una relación es una asociación entre entidades.

- **Modelo Entidad - Relación**

El Modelo Entidad – Relación conocido como Modelo E-R es una herramienta para modelar bases de datos, fue diseñado por Peter Chen en el año 1976. Es el esquema conceptual de la base de datos.

Este modelo se fundamenta en la simbología, normas y técnicas que admitirán representar de forma gráfica los datos del mundo real. También muestra un enfoque normal de los datos, apartando los objetos de sus agrupaciones, obteniendo un mejor nivel de independencia de los datos, además permite la dependencia de los componentes de un modelo.

Este modelo se ha convertido en una norma para diseñar bases de datos. Por medio de entidades describe la realidad y es un modelo con el cual se representan adecuadamente los datos.

Dentro del modelo E-R se encuentran los siguientes aspectos importantes:

Entidad. Se conoce a los objetos que existen el entorno de la realidad, por ejemplo: la entidad DEPARTAMENTO, la entidad MATERIA.

Atributo. Son las características propias de cada entidad. Por ejemplo: La entidad DEPARTAMENTO tiene atributos como: id\_departamento, nombre\_departamento.

La entidad MATERIA tiene atributos como: id\_materia, nombre\_materia.

En la Unidad 2.3 se verá a mayor detalle las Entidades y Atributos.

**Relación.** Es el enlace o asociación de una entidad con otras. Por ejemplo:

La entidad DEPARTAMENTO se relaciona con la entidad MATERIA.

Un DEPARTAMENTO *tiene* varias MATERIAS.

En la Unidad 2.4 se verá a mayor detalle las Relaciones.

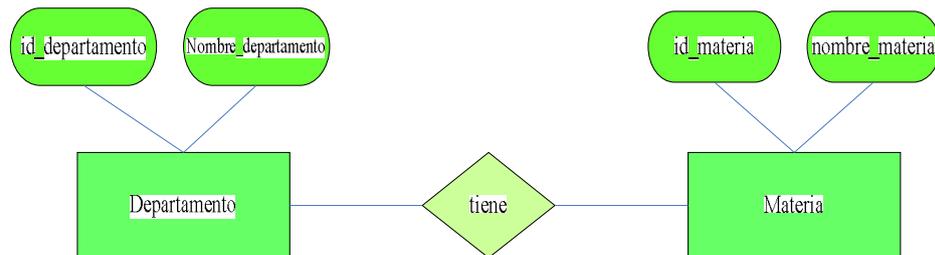


Figura 1.3.9.2\_1: Modelo Entidad - Relación

#### ▪ **Modelo Orientado a Objetos**

El Modelo Orientado a Objetos se fundamenta en una reunión de objetos (cada identidad del mundo real es *denominada objeto*), *un objeto tiene fracciones de código* denominados *métodos* y éstos trabajan en el objeto.

A la agrupación de los objetos que tienen características equivalentes métodos se denomina *clase*.

Otro término importante dentro de este modelo es *herencia* la que se fundamenta en que una clase hereda variables y métodos a las denominadas subclases.

Se conoce a la *encapsulación* a la función que admite ocultar la información a los demás objetos, los que son erróneos no accedan.

El *polimorfismo* es la función de una operación la que se aplica a diferentes tipos de objetos.

La *sobrecarga* es una operación que asocia diferentes métodos, depende del sistema el cual toma la decisión de realizar la operación.

[Volver Inicio](#)



## U1. Lección 3 tarea 1

### ACTIVIDAD DE APRENDIZAJE

---

Denominación de la tarea: **Establecer las diferencias entre Modelos de Datos Basados en Registros y Modelos de Datos Basados en Objetos y de cada una de sus clasificaciones**

#### ASESORÍA DIDÁCTICA

Para desarrollar la siguiente actividad de aprendizaje, debe leer y comprender el contenido de la lección 3.

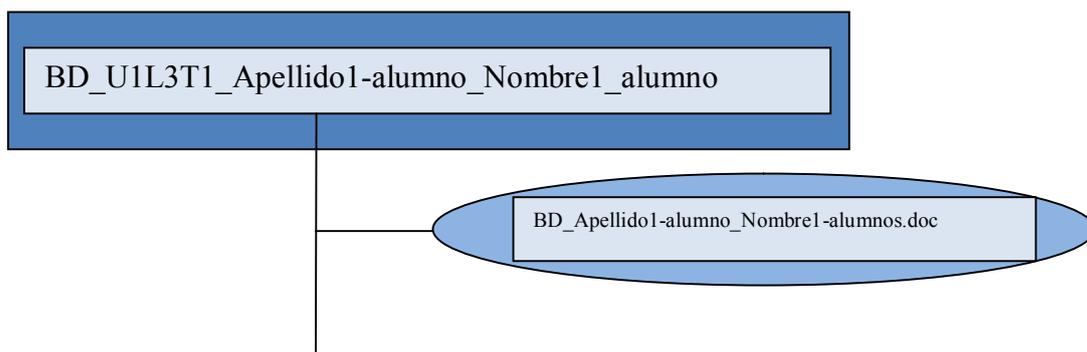
Para establecer las diferencias entre los dos enfoques, implica una comparación crítica. La realización de este análisis requiere determinar previamente el aspecto o criterio de comparación.

#### ACTIVIDAD DE APRENDIZAJE

1. Estructure un cuadro comparativo entre Modelos de Datos Basados en Registros y Modelos de Datos Basados en Objetos y su respectiva clasificación.

Envíe a su tutor la tarea en una carpeta comprimida como .rar

Formato de la carpeta:





## Resumen Lección 3

---

### Modelo de Datos

Esta lección ha presentado una visión global de base de datos, incluyendo:

- **Modelo**
- **Esquema**
- **Ejemplar**
- **Modelar**
- **Modelo de Datos**
- **Restricciones**
- **Tipos Modelos de Datos**

#### **Destacándose que:**

El modelo es una colección de abstracciones que forman un enfoque de una parte del mundo.

El modelar se fundamenta en detallar un mundo abstracto que permita obtener conclusiones que correspondan al mundo real.

El modelo de datos se define a un modelo como la descripción analógica utilizada para visualizar algo que no puede observarse directamente.

El uso del modelo de datos es importante porque permite la interoperabilidad entre el diseñador, el desarrollador de aplicaciones y el cliente. Todas las aplicaciones dependen de la generación correcta del modelo de datos.

Los modelos de datos son de dos tipos:

#### Modelos Lógico Basado en Registros

Este tipo de modelos se usan para detallar el entorno general de la base de datos y una representación de la implementación de la base en más alto nivel.

- Modelo Jerárquico
- Modelo de Red,
- Modelo Relacional

#### Modelos Lógico Basado en Objetos

Los modelos de datos basados en objetos utilizan conceptos tales como entidades, atributos y relaciones.

- Modelo entidad – relación
- Modelo orientado a objetos

## UNIDAD 1: GENERALIDADES DE LAS BASES DE DATOS

### LECCIÓN 1.4: Herramientas Case



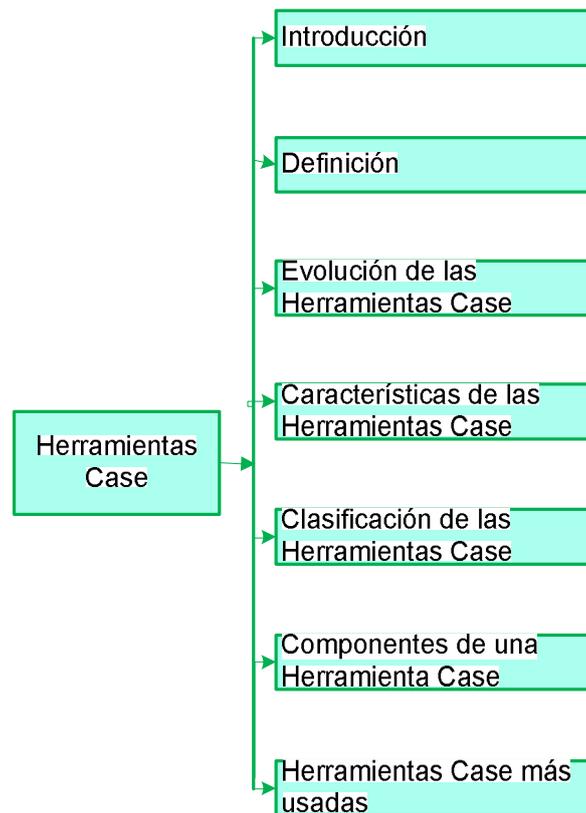
#### Visión General

En esta lección se tratan elementos generales de las bases de datos, tal que permitan al aprendiente contextualizar el ámbito de desarrollo de las bases de datos.

[Volver Inicio](#)



#### Plan de la Lección





## **Desarrollo de la Lección**

### **1.4.1. Introducción**

En la actualidad las Herramientas Case facilitan el modelamiento de la información de un determinado negocio, estas herramientas permiten interpretar objetos de datos del negocio, son un apoyo para nuevos diseños de estrategias en sistemas de información. El uso de las herramientas case es importante ya que las organizaciones están automatizando los pasos del ciclo de vida del desarrollo de un software.

### **1.4.2. Definición**

Las herramientas CASE son un agrupación de metodologías y normas que permiten la sistematización durante todo el Ciclo de Vida del Desarrollo de Software o en una de sus etapas. Las etapas del Ciclo de Vida del Desarrollo de Software son: Análisis, Diseño, Implementación, Instalación.

El uso de las herramientas CASE permiten que los desarrollos de software sean de calidad además tener un crecimiento de la efectividad dentro de la organización.

### **1.4.3. Características de las Herramientas Case**

- Permite la sistematización del desarrollo de software, mejorando la productividad y la calidad de los sistemas de información.
- El uso de las herramientas admiten la aplicación práctica y fácil de metodologías estructuradas y orientadas a objetos.
- Las herramientas case proporcionan facilidad en la ejecución de prototipos.
- Ayuda a generalizar la documentación.
- Permiten reutilizar los componentes de software.
- Facilitan el desarrollo visual de aplicaciones, por medio del uso de gráficos.

#### 1.4.4. Evolución de las Herramientas Case

Tabla 1.5.3\_1: Historia de las Herramientas Case

AÑO	OBSERVACIÓN
1970 - 1979	En esta década las herramientas CASE agregaron procedimientos gráficos ya que únicamente se veían como procesadores de texto además introdujeron los diagramas de flujo de datos, lo que permitió incluir a los diccionarios de datos.
1980 - 1989	En el año 1982 fue importante por la aparición del primer producto de este tipo de herramientas. En el año 1985 las herramientas CASE demostraron ser muy importantes dentro del desarrollo de software. A mediados de esta década” apareció el diseño automático de análisis y pruebas. Repositorios automáticos de información de sistemas. A finales de esta década se da la generación automática de código desde especificaciones de diseño.”
1990 - 1999	“A inicios de esta década inicia la metodología inteligente. Interface de Usuarios reusable como una metodología de desarrollo.”

#### 1.4.5. Clasificación de las Herramientas Case

Se puede clasificar a las herramientas Case de acuerdo a:

- Soporte de plataformas
- En qué etapa del ciclo de vida de desarrollo de sistemas se encuentran
- La función que cumple.

A continuación se detalla la clasificación de las herramientas Case a partir de la etapa del ciclo de vida de desarrollo de sistemas en la que se encuentran:

- **Herramientas integradas, I-CASE (Integrated CASE, CASE Integrado):**  
 Este tipo de herramientas son conocidas también como CASE Workbench y cubren todas las etapas del ciclo de vida de desarrollo de sistemas.
  
- **Herramientas de alto nivel, U-CASE (Upper CASE – CASE superior) o front-end,**  
 Estas herramientas están dentro de las etapas de Análisis y Diseño del ciclo de vida de desarrollo de sistemas.
  
- **Herramientas de bajo nivel, L-CASE (Lower CASE – CASE inferior) o back-end,**  
 Estas herramientas están dentro de las etapas de Construcción e Implantación del ciclo de vida del desarrollo de sistemas.
  
- **Juegos de herramientas o Tools – Case**  
 Este tipo de herramientas son el prototipo más sencillo y dentro de éstas se encuentran las herramientas conocidas como de reingeniería y son las que permiten las que permiten automatizar una etapa dentro del ciclo de vida del desarrollo de sistemas.

#### **1.4.6. Componentes de una Herramienta Case**

Los componentes según su forma estructural de las herramientas case son los siguientes:

- **Repositorio:** lugar en el que se agrupan los componentes establecidos por la herramienta case y para la cual se requiere el soporte del Sistema de Gestión de Base de Datos. Gran parte de estas herramientas tienen un repositorio de su propiedad y las otras funcionan con un repositorio proporcionado por otros creadores.

- **Metamodelo:** conforma el ámbito para determinar los métodos admitidos por la herramienta case. Al metamodelo casi siempre no se lo ve es decir se muestra invisible.
- **Carga o descarga de datos:** este componente comunica con otras herramientas, crean a partir de la misma herramienta programas, diseño de bases de datos, etc, y a su vez fomentan otros sistemas.
- **Generador de informes:** es el informe de la documentación que se relaciona con las metodologías a usar.
- **Interfaz de usuario:** estará conformado por el editor de texto e instrumentos de diseño, con el manejo de programas con interfaces gráficas detallan diagramas de las diferentes metodologías.

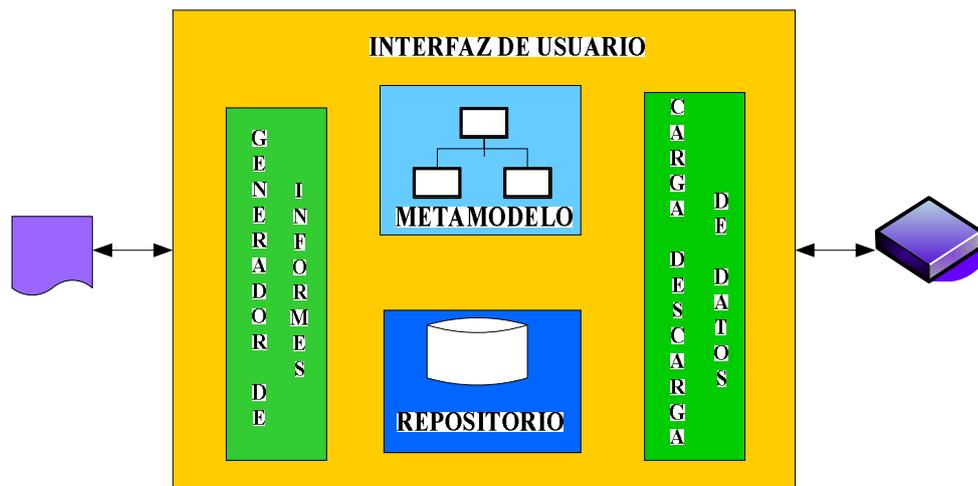


Figura 1.5.6\_1: Componentes de Herramientas Case<sup>37</sup>

<sup>37</sup> De Miguel, Adoración; Piattini, Mario; Marcos, Esperanza. (2000). *Diseño de Bases de Datos Relacionales*, Colombia: Alfaomega S.A.

### 1.4.7. Herramientas Case más usadas

Tabla 1.4.7\_1: Herramientas Case más usadas

CARACTERÍSTICAS	POWER DESIGNER	RATIONAL ROSE	ERWIN	ORACLE DESIGNER	DATABASE DESIGNER	XCASE
Multiplataforma	✓	✓	Sólo funciona bajo Windows	✓	Sólo funciona bajo Windows	✓
Soporta bases de Datos	Oracle, SQL Server, Sybase, DB2, e Informix, Access	DB2, Oracle, SQL Server	Oracle, SQL Server, Sybase, DB2, e Informix	Oracle, SQL Server, Sybase	PostgreSQL	Oracle, SQL Server, Sybase, DB2, Informix, MySQL, PostgreSQL
Licencia Comercial	✓	✓	✓	✓	✓	✓
Fabricante	Sybase	IBM	<u>Computer Associates</u>	Oracle	<u>MicroOlap</u>	Resolution Software
Versión 2011	15.1	7.0	7.1	11g	1.2.11	6.0

[Volver Inicio](#)



## U1. Lección 4 tarea 1

### ACTIVIDAD DE APRENDIZAJE

---

Denominación de la tarea: Versiones actuales sobre Herramientas Case

#### ASESORÍA DIDÁCTICA

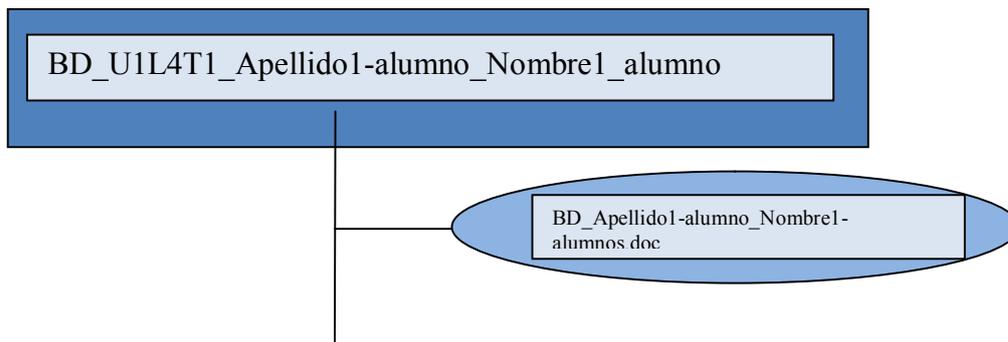
Para desarrollar la siguiente actividad de aprendizaje, debe leer y comprender el contenido de la lección 4.

#### ACTIVIDAD DE APRENDIZAJE

Consulte 3 Herramientas Case para Base de Datos que se usan actualmente y realice un cuadro comparativo de 4 características de cada una.

Envíe a su tutor la tarea en una carpeta comprimida como .rar

Formato de la carpeta





## Resumen Lección 4

---

### Herramientas Case

Esta lección ha presentado una visión global de herramientas case, incluyendo:

- **Evolución**
- **Características**
- **Clasificación**
- **Componentes**
- **Herramientas Case más usadas**

#### **Destacándose que:**

Las herramientas CASE son un agrupación de metodologías y normas que permiten la sistematización durante todo el Ciclo de Vida del Desarrollo de Software o en una de sus etapas.

Entre sus principales características:

- Permite la sistematización del desarrollo de software, mejorando la productividad y la calidad de los sistemas de información.
- El uso de las herramientas admiten la aplicación práctica y fácil de metodologías estructuradas y orientadas a objetos.

Las Herramientas Case de acuerdo al ciclo de vida del desarrollo de sistemas se clasifica en:

- Herramientas integradas, I-CASE (Integrated CASE, CASE Integrado):
- Herramientas de alto nivel, U-CASE (Upper CASE – CASE superior) o front-end,
- Herramientas de bajo nivel, L-CASE (Lower CASE – CASE inferior) o back-end,
- Juegos de herramientas o Tools – Case

Los componentes de una Herramienta Case son:

- Repositorio
- Metamodelo
- Carga o descarga de datos
- Generador de informes
- Interfaz de usuario

## UNIDAD 2: PROCESO DE DESARROLLO DE LA BASE DE DATOS

### COMPETENCIA A SER DESARROLLADA POR EL ESTUDIANTE:

Diseña e implementa bases de datos conforme a las características del sistema de información al cual va a soportar dicha base de datos, según los requerimientos establecidos, definidos o proporcionados, asegurándose de que exista integridad de los datos, utilizando el lenguaje SQL y herramientas CASE de apoyo, observando estándares y metodologías.

### LECCIÓN 2.1: Modelo Entidad - Relación



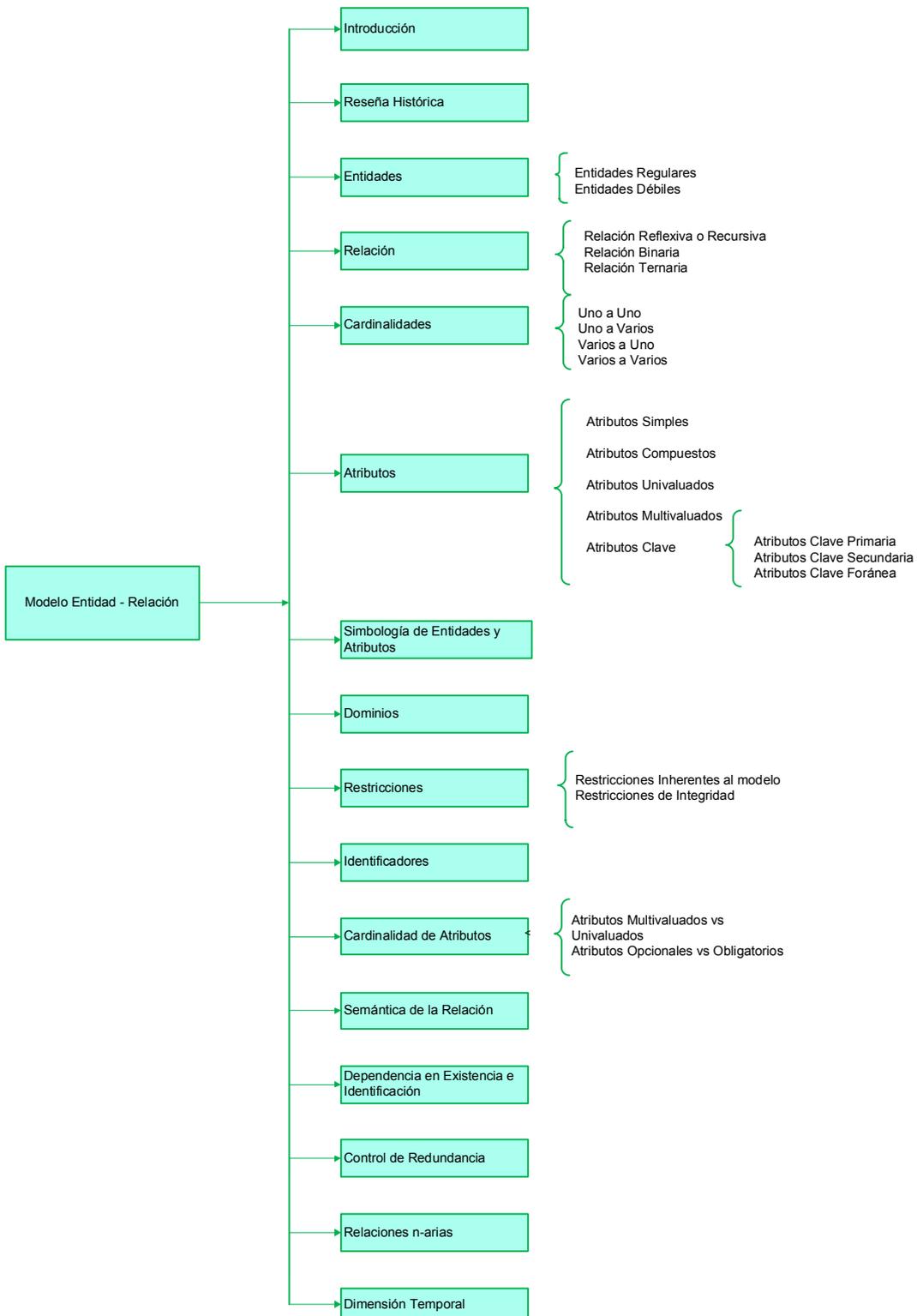
#### Visión General

En esta lección trata el modelo entidad – relación que sirve para modelar una base de datos. Sus principales elementos son las entidades, atributos y relaciones.

[Volver Inicio](#)



## Plan de la Lección





## Desarrollo de la Lección

### 2.1.1. Introducción

Las entidades y atributos se han convertido en aspectos fundamentales, el objeto básico que representa el modelo E-R es una entidad, cada entidad posee atributos. Las entidades y atributos tienen su respectiva clasificación y simbología que permite distinguirlos.

### 2.1.2. Reseña Histórica

El modelo Entidad – Relación definido en el año 1976 por Peter Chen es uno de los más usados en el diseño de base de datos.

Inicialmente este modelo sólo contenía definiciones sobre entidades, atributos y relación, después se añadieron conceptos como jerarquías, atributos compuestos.

### 2.1.3. Entidades

Una entidad es un elemento que representa a un objeto que puede ser físico como por ejemplo un libro, profesor o puede ser algo abstracto, como por ejemplo una materia.

**Una definición importante dentro de las entidades es:**

**Conjunto de Entidades.** Es el conjunto de entidades que tienen iguales atributos.

Por ejemplo: Todos los estudiantes de Décimo Año de Educación Básica de la Provincia de Pichincha, tienen iguales atributos: número de identificación, nombre, dirección, edad.

Los tipos de Entidades son:

#### 2.1.3.1. Entidades Regulares

Son las entidades que existen por si solas que no depende de otras.

Ejemplo:

La entidad ESTUDIANTE

### 2.1.3.2. Entidades Débiles

Son las entidades que requieren de la existencia de otras.

### 2.1.4. Relación

Se denomina relación a la asociación entre entidades.

Ejemplos:

Estudiante **posee** libros

Persona **tiene** número de identificación

**Características de las relaciones:** Una relación tiene las siguientes características:

- **Nombre:** Con el nombre se puede identificar una relación unívocamente.
- **Grado:** Corresponde al número de tipos de una entidad en las que se ejecuta la asociación. Son las relaciones binarias y ternarias.

Existen tres tipos de relación:

#### 2.1.4.1. Relación Reflexiva o Recursiva

Se denomina relación recursiva aquella entidad que participa más de una vez en una relación pero cumple diferentes funciones.

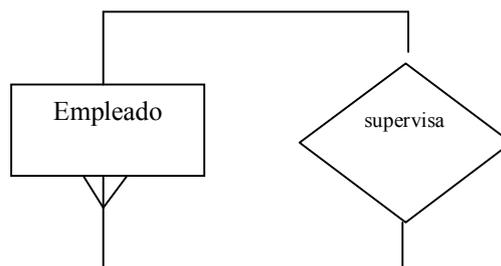


Figura 2.1.4.1\_1: Relación Recursiva

### 2.1.4.2. Relación Binaria

Es una relación en la que participan dos entidades.

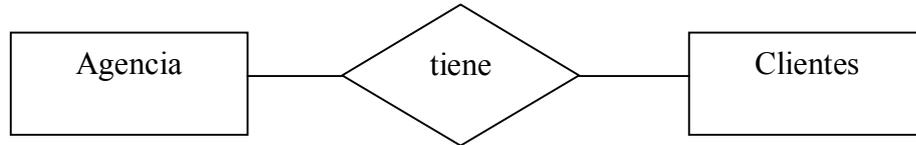


Figura 2.1.4.2\_1: Relación Binaria

### 2.1.4.3. Relación Ternaria

Es una relación en la que participan tres entidades.

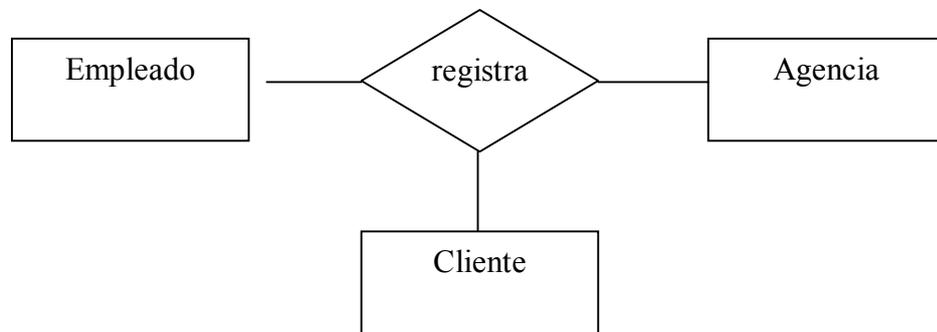


Figura 2.1.4.3\_1: Relación Ternaria

### 2.1.5. Cardinalidad

Se denomina cardinalidad a la representación de cómo un objeto de una entidad se relaciona con un objeto de la otra entidad.

La representación de la cardinalidad se lo realiza de la siguiente forma: la relación se simboliza con un rombo el cual tiene dos líneas cada una con su extremo ahí se coloca los números que mencionan la cantidad de entidades que participan en la interrelación. De esto parte la verificación del número de entidades que intervienen.

### 2.1.5.1. Uno a Uno (1:1)

Esta se da sólo un objeto de la entidad1 se relaciona solo con un objeto de la entidad2.

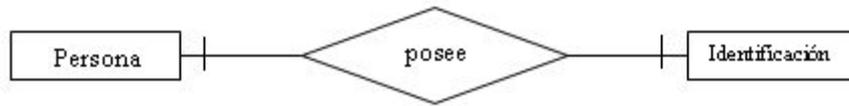


Figura 2.1.5.1\_1: Cardinalidad Uno a Uno

### 2.1.5.2. Uno a Varios (1:N)

Este tipo se da cuando un objeto de la entidad1 se puede relacionar con muchos objetos de la entidad2, pero cada objeto de la entidad2 se relaciona tan solo con un objeto de la entidad1.



Figura 2.1.5.2\_1: Cardinalidad Uno a Varios

### 2.1.5.3. Varios a Uno (N:1)

Este tipo se da cuando muchos objetos de la entidad1 se relacionan con un objeto de la entidad2, pero cada objeto de la entidad2 se relaciona solamente con un objeto de la entidad1.



Figura 2.1.5.3\_1: Cardinalidad Varios a Uno

#### 2.1.5.4. Varios a Varios (N:N)

Este tipo de relación existe cuando un objeto de la entidad1 se puede relacionar con muchos objetos de la entidad2 y un objeto de la entidad2 se puede relacionar con muchos objetos de la entidad1.



Figura 2.1.5.4\_1: Cardinalidad Varios a Varios

#### 2.1.6. Atributos

Un atributo es la descripción de las características de cada entidad, por ejemplo:

Para la entidad profesor, algunos atributos serían: nombre, cédula, profesión, edad, estos detalles va a permitir diferenciar un profesor de otro.

Para la entidad libro, algunos atributos serían: nombre, autor, editorial, al igual que el ejemplo anterior estas características propias de la entidad lo distinguen de otros.

Los atributos tienen el siguiente esquema:

- **Nombre:** Descripción física del objeto.
- **Tipo:** Corresponde al tipo de dato de cada atributo puede ser entero, cadena, fecha, etc.
- **Dominio:** Es la colección de valores admitidos para un atributo. Poseen normas para la comprobación de que cierto dato este en un dominio. Los atributos se clasifican en:

##### 2.1.6.1. Atributos Simples

Es un atributo que no se lo puede dividir porque no tiene sentido hacerlo. Por ejemplo:

Identificación de una persona → **número de cédula**

### 2.1.6.2. Atributos Compuesto

Este tipo de atributo es aquel que se puede dividir sin que cambie el sentido de la característica de la entidad.

Por ejemplo:

Dirección de una persona → ***calle, ciudad, país, número catastral***

Nombre de una persona → ***primer nombre, segundo nombre apellido paterno, apellido materno***

“Un atributo compuesto se divide sólo por razones de manejo a nivel del lenguaje de consulta o programación o por requerimientos del usuario, si no hay necesidad no se debe dividir ya que en algunas ocasiones se vuelve complejo el manejo de esta situación, es decir el atributo compuesto se trabaja como un atributo simple. Así se puede concluir que un atributo compuesto es la suma (concatenación) de los valores de los atributos simples que lo conforman.”<sup>38</sup>

### 2.1.6.3. Atributos Univaluados o Monovaluados

Este tipo de atributos son aquellos que únicamente toman un valor para la entidad.

Por ejemplo:

***Edad de una persona***

### 2.1.6.4. Atributos Multivaluados o Multivalorados

Los atributos denominados multivaluados son los que pueden asumir algunos valores para una respectiva entidad, por ejemplo, teléfono ya que una persona puede tener varios números telefónicos. Por ejemplo:

Número teléfono de una persona → ***número telefónico de la casa, número telefónico celular, número telefónico oficina***

---

<sup>38</sup> [http://www.unalmed.edu.co/~mstabare/disenio\\_conceptual.htm](http://www.unalmed.edu.co/~mstabare/disenio_conceptual.htm)

### 2.1.6.5. Atributos Derivados

Este tipo de atributos son parte del valor de otros atributos.

Por ejemplo:

Edad de una *persona* → *año en curso menos año de nacimiento de una persona*

### 2.1.6.6. Atributos Claves

Este tipo de atributos identifican una o varias entidades, los tipos de atributos clave son:

- **Atributos de Clave Primaria**

Esta clave no admite valores nulos y es el identificador de los demás atributos de una entidad. Más de un atributo puede considerarse como clave primaria, se debe escoger el que más importante. Ejemplos:

*código\_estudiante*

*número\_matrícula*

- **Atributos de Clave Secundaria**

Son aquellos que identifican a una o varias entidades en una determinada colección de entidades.

- **Atributos de Clave Foránea**

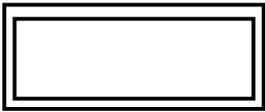
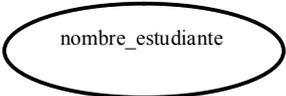
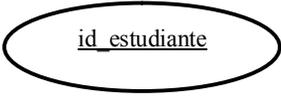
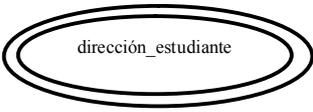
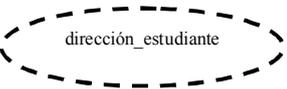
Este tipo de clave se da en el momento que el atributo tenga referencia a la clave de una entidad proporcionada en otra relación.

- **Atributos de Clave Candidata**

Este tipo de de clave identifican de manera alterna a una sola entidad, es equivalente a la clave primaria.

## 2.1.7. Simbología de Entidades y Atributos

Tabla 2.1.7\_1: Simbología de Entidades y Atributos

Simbología	Representación
	Entidad
	Entidad Débil
	Atributo
	Clave Primaria
	Atributo Multivalorado
	Atributo Derivado

## 2.1.8. Dominios

Un dominio es un conjunto de valores similares que tienen un nombre que lo identifica. Las características son las siguientes:

- Un objeto adquiere valores que pertenecen a un dominio.
- Un dominio está asociado a un predicado que realiza la comprobación de que un valor pertenece a un dominio.

$$D = \{ v : p(v) \}$$

D: Dominio

v: Valor

p: Predicado

## 2.1.9. Restricciones

### 2.1.9.1. Restricciones Inherentes al modelo:

Únicamente admite establecer relación entre entidades, no está permitido entre entidades e relación ni sólo entre relación.

### 2.1.9.2. Restricciones de Integridad:

Sólo se toma en cuenta las restricciones específicas, diferenciando entre:

- Restricciones *sobre valores*, que se forman por medio de la definición del dominio.
- Restricciones *estructurales*, que se refieren a lo siguiente:
  - Atributos:** Identificadores y Cardinalidades
  - Relación:** Cardinalidad mínima y máxima, Dependencia en existencia e identificación, Otras Restricciones.

Entre los atributos de un cierto tipo de entidad siempre existen uno o varios los que *van a identificar unívocamente* cada ejemplar de ese tipo de entidad.

## 2.1.10. Identificadores

La colección de cada atributo se conoce como *Identificador Candidato (IC)*. Los IC deben cumplir la condición de unívocos y mínimo: cuando un identificador candidato es compuesto el número de atributos que lo conforman debe ser mínimo en la eliminación de cada uno que con lo que se perdería la característica de identificador. Entre los identificadores candidato hay el denominado *Identificador Principal (IP)* los demás son *Identificadores Alternativos (IA)*.

## 2.1.11. Cardinalidad de Atributos

- **Atributos Multivaluados vs Atributos Univaluados**

Los atributos multivaluados toman más de un valor al mismo tiempo (por ejemplo una persona puede tener más de un teléfono).

- **Atributos Opcionales vs Obligatorios**

Los atributos opcionales obligan a un atributo de cierto tipo de entidad a tomar como mínimo un valor del dominio, es decir el valor de ese atributo es obligatorio y no puede ser nulo.

### 2.1.12. Semántica de las Relación

Se conoce como semántica al número máximo y mínimo de ejemplares de cierto tipo de entidad que se interrelacionan con ejemplares de otros tipos de entidad.

Notación: "I(E1(n1,m1)):E2(n2,m2)) siendo:

I un tipo de interrelación binaria,

E1 y E2 los tipos de entidad asociados por ella,

n1 y n2 las cardinalidades mínimas de E1 y E2 respectivamente, y

m1 y m2 las cardinalidades máximas de E1 y E2 respectivamente.

– Semántica:

- un ejemplar de E2 puede estar relacionado con un mínimo de n1 y un máximo de m1 ejemplares de E1.
- un ejemplar de E1 puede estar relacionado con un mínimo de n2 y un máximo de m2 ejemplares de E2."<sup>39</sup>

### 2.1.13. Dependencia en existencia e identificación

#### 2.1.13.1. Dependencia en Existencia

Existe una dependencia de existencia cuando una relación se encuentra enlazado a un tipo de entidad regular con una entidad débil, de modo que el tipo de entidad débil no exista sin el tipo de entidad regular. Solo si el tipo de entidad regular existe, coexisten las de tipo entidad débil. En el siguiente ejemplo, se determina un tipo de relación se\_forma que asocia el tipo de entidad regular Edificio con el tipo de entidad débil Oficina, ésta es una dependencia en existencia porque los datos acerca de las Oficinas de un Edificio sólo tendrán sentido si perdura en la base de datos.

---

<sup>39</sup> De Miguel, Adoración; Piattini, Mario. (1999). *Fundamentos y modelo de Bases de Datos*, México: Alfaomega S.A.

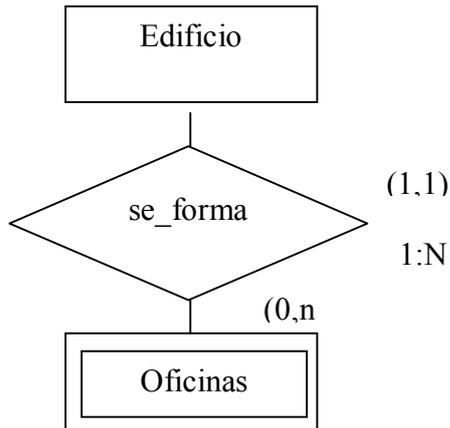


Figura 2.1.13.1\_1: Dependencia en Existencia

### 2.1.13.2. Dependencia en Identificación

Esta dependencia se da cuando el tipo de entidad débil no se puede identificar sólo con sus atributos, sino que necesita agregar la clave de la entidad regular de la que depende. En el siguiente ejemplo se verifica “la relación tiene asocia la entidad regular Libro con el tipo de entidad débil Ejemplar, es dependiente en identificación, ya que un ejemplar determinado, además de depender en existencia de un cierto libro, está identificado con la clave del libro (cod\_libro) del cual depende el ejemplar, más un código propio (Num\_Ejemplar).

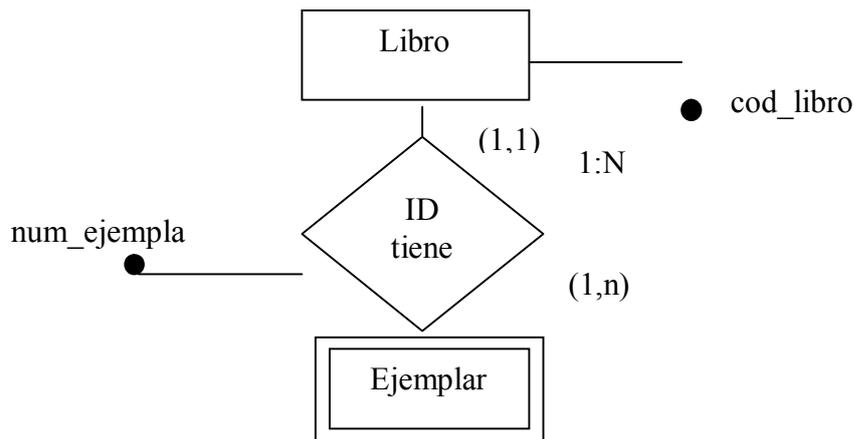


Figura 2.1.13.2\_1: Dependencia en Identificación

<sup>40</sup> De Miguel, Adoración; Piattini, Mario. (1999). *Fundamentos y modelo de Bases de Datos*, México: Alfaomega S.A.

#### **2.1.14. Control de redundancias**

En el modelo Entidad – Relación se debe evitar las redundancias para no tener inconsistencia de la representación.

El componente de un esquema es redundante si se lo puede eliminar sin perder la semántica. Hay dos formas de redundancia:

- En los atributos derivados no tienen inconsistencias a pesar de ser redundantes, pero siempre cuando el esquema indique la condición respectiva de derivados.
- Las relaciones conocidas como derivadas son redundantes si cuando se elimina no incluye la perder la semántica por existir la posibilidad de realizar una asociación de ejemplares por medio de otras relaciones. La condición que se necesita forma parte de un ciclo del modelo entidad – relación.
- Un relación puede ser eliminada cuando:
  - Exista un ciclo
  - Las relaciones que conforman el ciclo son iguales semánticamente.
  - Se pueden relacionar las entidades interrelacionadas aún cuando se haya eliminado la interrelación.
  - La relación no debe tener atributos que puedan trasladarse a otro esquema para no perder la semántica.

#### **2.1.15. Relación n-arias**

Las relaciones binarias son más sencillas de manejar que las relaciones de grado superior a 2.

Las relaciones de grado 4 no son permitidas.

En ocasiones se puede convertir en relación ternaria o varias binarias que tengan la misma semántica.

### **2.1.16. Generalización y herencia**

La Generalización es un determinado caso de la relación entre uno o varios tipos de entidad (conocidos como subtipos) y un tipo más general (conocido como supertipo).

Las características de la generalización son similares para todos los subtipos:

Entre los subtipos y supertipo se crea una relación "ES\_UN". Es decir una forma de subtipo ES\_UN (también) del supertipo.

Hay características mínimas y máximas que son: (1,1) en supertipo y (0,1) en subtipos.

Se conoce como especialización al componente de contrario de abstracción. Estas interrelaciones (relaciones) tienen una característica importante que se conoce como herencia.

Todos los atributos, relación del supertipo van a ser atributos de los subtipos.

### **2.1.17. Agregación**

La agregación es una clase especial de relación dentro del modelo entidad – relación extendida, existen dos clases de agregación:

Compuesto: Una agregación de este tipo se logra por la unión de de elementos que podrían ser tipos de entidades.

Miembro: La agregación se representa como una reunión de elementos de un mismo tipo de entidad.

### **2.1.18. Dimensión Temporal**

Es importante establecer un método semántico y gráfico que acumule en un esquema conceptual el paso del tiempo y cómo este influye en el cambio de datos.

Hay varios tipos de aproximaciones:

Atributos de tipo fecha agrupados con ciertas entidades e relación:

Un solo atributo de cierto tipo para eventos instantáneos es decir que no tienen duración.

Se requiere de fecha de inicio y fin en un evento que sucede en un intervalo de tiempo.

El atributo fecha es multivaluado cuando en una relación de dos ejemplares se repite en cierto tiempo:

Un atributo de estado se usará para la representación un cierto tipo de entidad a lo largo del tiempo, con lo cual se mostrará el estado de la entidad.

En cierto casos la fecha en la que se ha realiza un cambio de estado se asocia a un atributo.

### **2.1.19. Ejemplos:**

#### **“Distribuidora de Películas**

Una distribuidora de películas desea mantener cierta información acerca de las películas que va a lanzar próximamente, con el fin de promocionarlas adecuadamente.

Cada cierto tiempo se envían anuncios para inserta periódicos. Todos los anuncios incluyen el título de la película, el nombre del actor o actores principales y el nombre del director (considerando que hay un solo director por película). Además, para las películas que han recibido premios importantes, el anuncio suele incluir un resumen de los mismos, detallando el nombre del certamen, la ciudad donde se celebró y el tipo de premio recibido (mejor director, película, actor, etc). Y algunas veces en los anuncios se reseñan también las mejores críticas, indicando el nombre del periódico o revista donde se publicó la crítica, la fecha de publicación y el autor que firma la crítica.

Otra parte importante de la promoción consiste en hacer de intermediarios entre los medios de comunicación y los participantes en la película, principalmente el director, los productores, y especialmente los actores. Para concertar y planificar las citas es preciso mantener alguna información personal acerca de los intervinientes en la película, como su nombre completo, dirección

y teléfono de contacto. Además, con el fin de facilitar la elaboración del material de las entrevistas, se proporciona a los medios de comunicación otros datos relacionados con la película como la aportación económica de cada productor, un resumen del argumento, un listado con el pape concreto que realiza cada actor en la película (hay que considerar que un actor en una película hace un único papel) y también el tipo de papel (protagonista, secundario, etc), así como un resumen de los premios recibidos. Es frecuente también que determinados medios de comunicación nos soliciten información más personal sobre los actores como su fecha y lugar de nacimiento, su estado civil y sobre el sueldo que han cobrado por realizar la película.

Las semanas previas al estreno se informa además a los distintos medios de la fecha y el lugar de estreno de la película a nivel nacional. Anualmente la distribuidora realiza un estudio donde se incluye, entre otras cosas, un ranking de las películas más premiadas por nacionalidad.

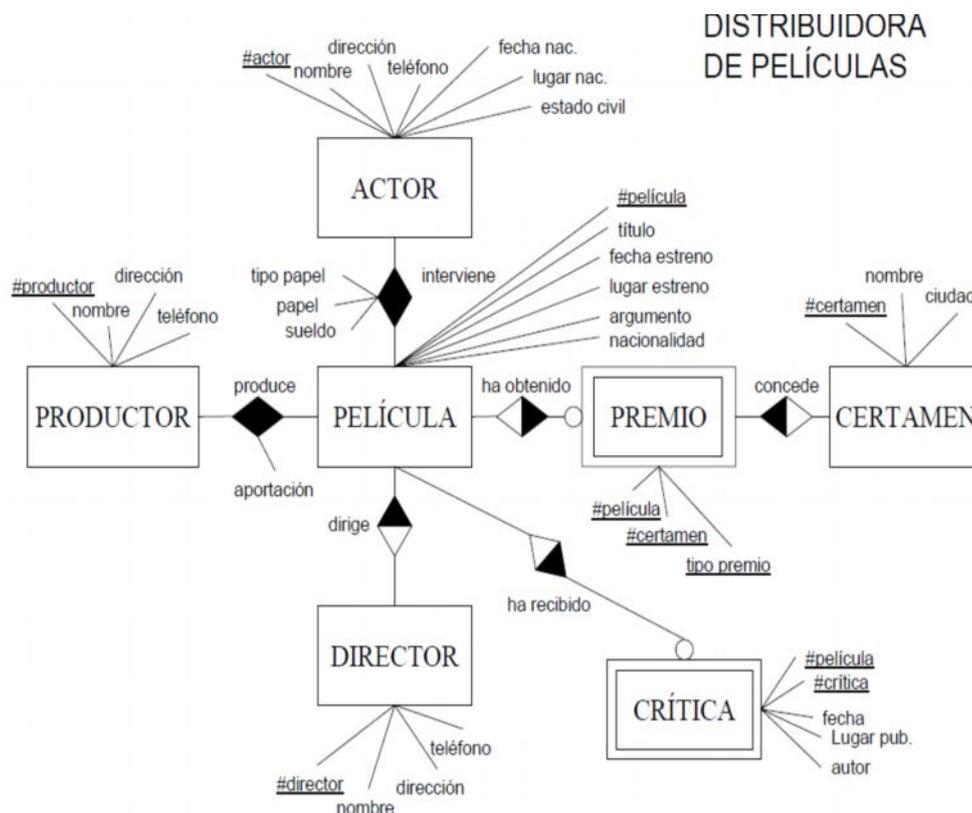


Figura 2.1.19\_1: Modelo E-R Distribuidora de Películas<sup>41</sup>

<sup>41</sup> [www.inf.ucv.cl/~jrubiado/docs/2009-02/.../ejemplos%20er.pdf](http://www.inf.ucv.cl/~jrubiado/docs/2009-02/.../ejemplos%20er.pdf)

- ESTRENO no es obligatorio representarlo como una entidad (aunque podría haberse hecho) porque se ha considerado que sólo de guarda el estreno a nivel nacional como se ha puesto en el enunciado. Si fuesen los estrenos en n países sería un atributo multivaluado que se transformaría en entidad dependiente.
- PREMIOS surge de la asociación N:M entre PELÍCULAS y CERTAMEN donde una película en un certamen puede recibir varios tipos de premios.

### **Campeonato de Fútbol**

La organización de un campeonato de fútbol de selecciones nacionales decide implantar un sistema que les permita mantener información actualizada sobre diversos aspectos.

Dos meses antes del comienzo deben conocerse los equipos que participarán, que se habrán clasificado mediante unas ligullas previas, de las cuáles nuestro sistema no guardará ninguna información. De cada selección se debe conocer inicialmente, al menos, el país al que representa y el nombre de sus datos personales y el puesto en que juegan habitualmente (portero, defensa, delantero, centro, lateral izquierdo...).

Un mes antes del comienzo se habrá concretado la planificación de los partidos estableciéndose la ciudad donde se jugará cada uno, la fecha y la hora de comienzo, los equipos que se enfrentarán, el nombre del colegiado principal que arbitrará el partido y el resto de colegiados que intervendrán en el mismo, así como al menos dos suplentes. Esta información se le remitirá oficialmente a cada selección, indicándole además qué equipos se enfrentarán en cada partido y cuál actuará de equipo local y cuál de visitante. Por otra parte, a cada colegiado se le informará por carta de los partidos en que intervendrá y su función en el partido (árbitro principal, juez de línea o asistente).

Una vez comenzado el mundial, diariamente se sacará un informe general de los partidos jugados y el resultado final. Además para cada partido se obtendrá un informe más detallado que incluirá los nombres y nacionalidades de los colegiados que han arbitrado, el nombre de los jugadores que han intervenido en el partido, el puesto en que cada uno ha jugado (que puede ser diferente de su puesto habitual) y los minutos que ha jugado. En este punto hay que señalar que el mismo jugador sólo puede intervenir una vez en cada partido, es decir, una vez que el jugador ha salido del campo no puede volver a entrar. Por otra parte, en partidos diferentes el mismo jugador puede jugar en distintos puestos. Otra cuestión a tener en cuenta con respecto a los partidos son los controles antidopaje que se realizan a los jugadores que intervienen en ellos. De cada partido se escogen una serie de jugadores al azar de cada uno de los equipos participantes para realizarle los análisis. Para cada muestra extraída, existe un miembro de la Federación Internacional de Fútbol que se encarga de custodiar la muestra, realizar la tramitación de la muestra y los resultados y un laboratorio encargado de realizar los análisis. La elección de ambos se realiza al azar teniendo en cuenta que un jugador nunca puede tener dos análisis ni con el mismo laboratorio ni con el mismo miembro de la Federación; sin embargo, varios jugadores (por ejemplo, todos los de un partido) podrían tener asignado un mismo laboratorio y miembro de la Federación.

De cada miembro de la Federación se almacena su nombre, su teléfono y su número de tarjeta identificativa y de los laboratorios, su nombre, dirección, teléfono. Al final del campeonato se entregan, además de los premios a las tres primeras selecciones clasificadas, un premio al jugador que más goles ha marcado y otro al jugador que ha marcado el gol más temprano (más cerca del comienzo de un encuentro), por lo que es preciso

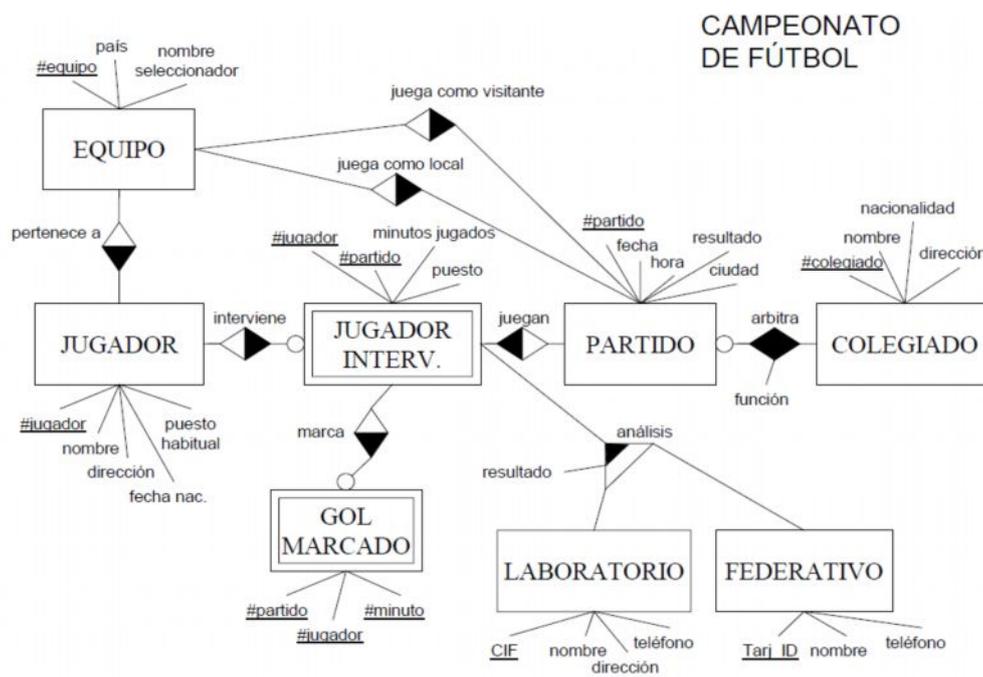


Figura 2.1.19.\_2: Modelo E-R Campeonato de Fútbol<sup>42</sup>

[Volver Inicio](#)

<sup>42</sup> [epcc.unex.es/modules.php?op=modload&name=Downloads](http://epcc.unex.es/modules.php?op=modload&name=Downloads)



## U2. Lección 1 tarea 1

### ACTIVIDAD DE APRENDIZAJE

---

Denominación de la tarea:

Modelo Entidad - Relación

### **ASESORÍA DIDÁCTICA**

Para desarrollar la siguiente actividad de aprendizaje, debe leer y comprender el contenido de la Lección 1 Unidad 2.

### **ACTIVIDAD DE APRENDIZAJE**

Construir el diagrama Entidad-Relación que modela el siguiente problema y realizar una breve explicación del mismo.

En cierto laboratorio de química existen varios proyectos de investigación por resolver, cada uno de ellos realizado por varios ingenieros. En cada proyecto al menos trabaja un ingeniero. Cada ingeniero trabaja únicamente en un proyecto en el laboratorio obligadamente. De cada proyecto se requiere conocer el nombre y la fecha de inicio.

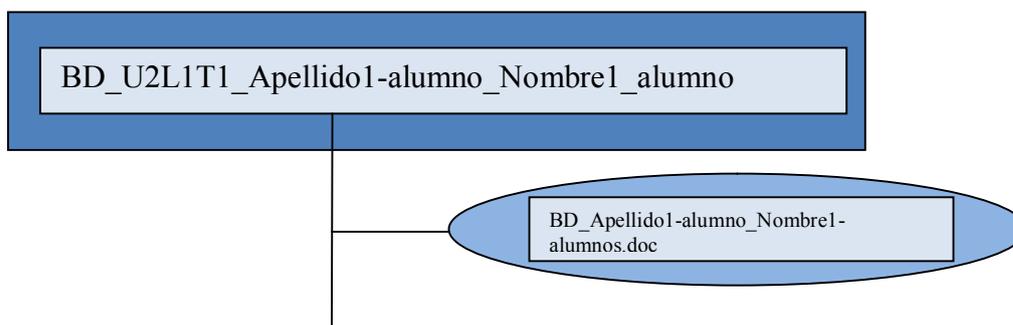
Los proyectos de de investigación son de dos tipos los nuevos y los que están en estudio. De los nuevos proyectos se requiere el presupuesto que se dispone para éste, de los proyectos en estudio se solicita guardar un texto donde se explique porque se lo puso en estudio.

Entre los ingenieros hay jefes, cada ingeniero que no es jefe es supervisado por un jefe, a los que son jefes nadie los supervisa, de cada ingeniero se requiere el nombre completo, identificación, dirección, dirección y teléfono.

Los ingeniero deberán realizar conferencias con otros laboratorios sobre las investigaciones, cada conferencia será realizada por uno o varios ingenieros, se requiere conocer de cada conferencia el nombre identificativo, fecha, hora de inicio, número de horas de la exposición y lugar.

Envíe a su tutor la tarea en una carpeta comprimida como .rar

Formato de la carpeta





## U2. Lección 1 tarea 2

### ACTIVIDAD DE APRENDIZAJE

---

Denominación de la tarea:

Modelo Entidad - Relación

### **ASESORÍA DIDÁCTICA**

Para desarrollar la siguiente actividad de aprendizaje, debe leer y comprender el contenido de la Lección 1 Unidad 2.

### **ACTIVIDAD DE APRENDIZAJE**

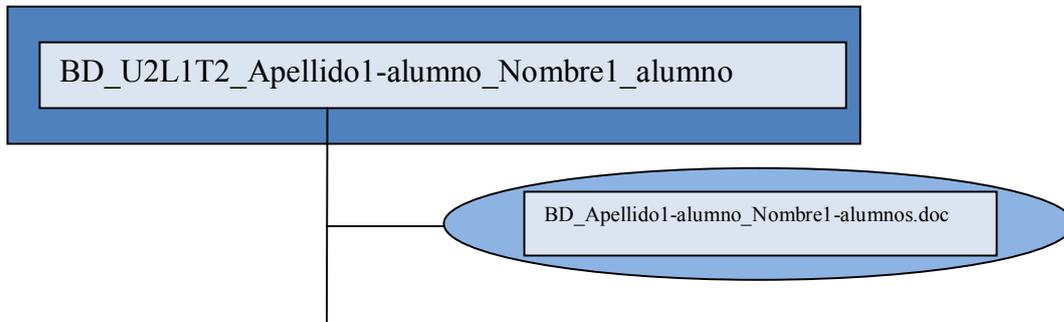
Construir el diagrama Entidad-Relación que modela el siguiente problema y realizar una breve explicación del mismo.

Se desea almacenar la información de una compañía aérea en una Base de Datos Relacional. La compañía aérea tiene tres recursos principales: Aviones, pilotos, tripulación. De cada pila se desea conocer su código nombre y horas de vuelo. De los miembros de la tripulación sólo se tendrá el código y el nombre. Pilotos y tripulación tienen una base a la que regresan después de cada jornada un vuelo va desde un origen a un destino a una hora concreta y tiene número de vuelo, de cada vuelo que se va a realizar durante los próximos 3 meses, así como de los vuelos que se han realizado se desea saber el avión en el que se va a hacer o en el que se ha hecho, el piloto y la tripulación.

Cada avión tiene un código, es de un tipo (boeing, airbus, entre otros). Y tiene una base donde es sometido a mantenimiento.

Envíe a su tutor la tarea en una carpeta comprimida como .rar

Formato de la carpeta





## Resumen Lección 1

---

### Modelo Entidad - Relación

Esta lección ha presentado una visión global del modelo Entidad – Relación, incluyendo:

- **Entidades**
- **Relación**
- **Cardinalidad**
- **Atributos**
- **Simbología de Entidades y Atributos**
- **Dominios**
- **Restricciones**
- **Identificadores**
- **Cardinalidad de Atributos**
- **Semántica de la Relación**
- **Dependencia en Existencia e Identificación**
- **Control de Redundancia**
- **Relaciones n-arias**
- **Dimensión Temporal**

#### **Destacándose que:**

Las entidades son elementos que representa a un objeto que puede ser físico. Las entidades son de dos tipos:

- Entidades Regulares
- Entidades Débiles

Se denomina relación a la asociación entre entidades. Se clasifican en:

- Relación Reflexiva o Recursiva
- Relación Binaria
- Relación Ternaria

La cardinalidad a la representación de cómo un objeto de una entidad se relaciona con un objeto de la otra entidad. Son:

- Uno a Uno (1:1)
- Uno a Varios (1:N)
- Varios a Uno (N:1)
- Varios a Varios (N:N)

Un atributo es la descripción de las características de cada entidad.

Un dominio es un conjunto de valores similares que tienen un nombre que lo identifica.

Las restricciones son de dos tipos:

- Restricciones Inherentes al modelo
- Restricciones de Integridad

La cardinalidad de atributos son:

- Atributos Multivaluados vs Atributos Univaluados
- Atributos Opcionales vs Obligatorios

La semántica de la relación es el número máximo y mínimo de ejemplares de cierto tipo de entidad que se interrelacionan con ejemplares de otros tipos de entidad.

La dependencia se clasifica en:

- Dependencia en Existencia
- Dependencia en Identificación

El control de redundancia sirven para evita las mismas y no tener inconsistencia en la representación del modelo.

La generalización es un caso de la relación entre uno o varios tipos de entidad (conocidos como subtipos) y un tipo más general (conocido como supertipo).

La agregación es una clase especial de relación dentro del modelo entidad – relación extendido

## UNIDAD 2: PROCESO DE DESARROLLO DE LA BASE DE DATOS

### LECCIÓN 2.2: Modelo Relacional



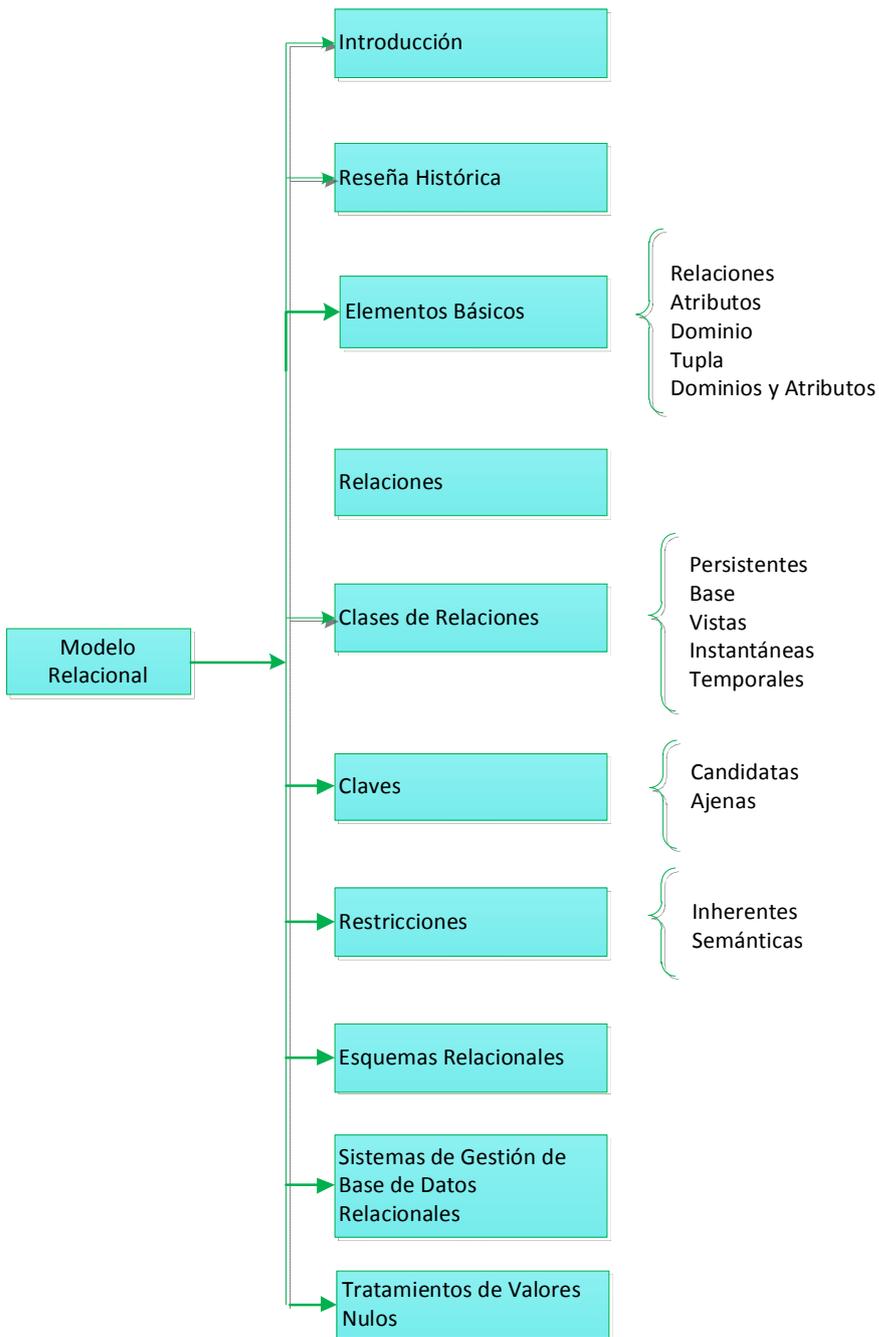
#### **Visión General**

En esta lección se trata el Modelo Relacional, reseña histórica, sus elementos básicos y componentes.

[Volver Inicio](#)



## Plan de la Lección





## Desarrollo de la Lección

### 2.2.1. Introducción

El modelo relacional representa el mundo real, agrega mecanismo de consulta independiente del SGBD, mantiene información sobre los metadatos.

### 2.2.2. Reseña Histórica

En el año 1970 Codd propuso un nuevo modelo de datos. Este modelo da mucha importancia en la independencia de la representación lógica de los datos respecto al almacenamiento in

“Codd concedió mucha importancia al tema de la **independencia** de la representación lógica de los **datos** respecto a su almacenamiento interno, que concretó en tres tipos de independencia:

de **ordenación**,

de **indización**, y

de los **caminos de acceso**”<sup>43</sup>

---

<sup>43</sup> De Miguel, Adoración; Piattini, Mario. (1999). *Diseño de Base de Datos Relacionales*, México: Alfaomega S.A.

## EVOLUCIÓN DEL MODELO RELACIONAL

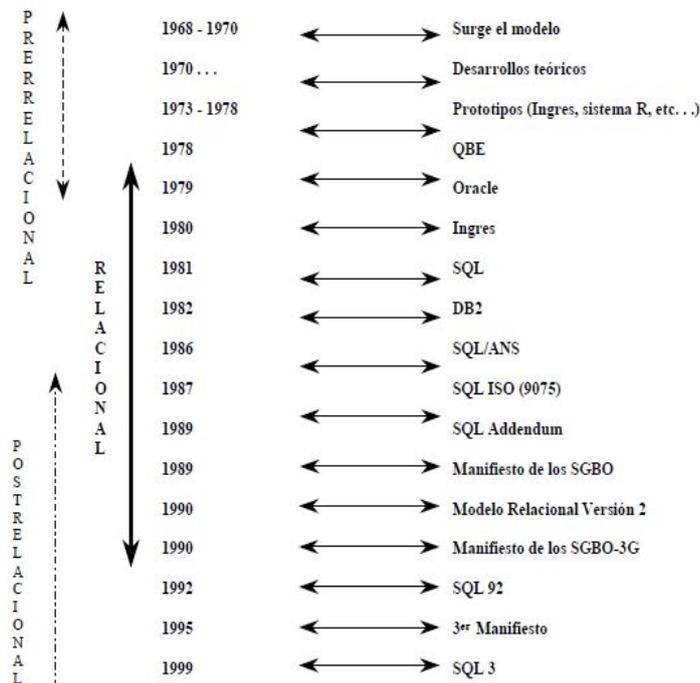


Figura 2.2.2.\_1: Evolución del Modelo Relacional<sup>44</sup>

### 2.2.3. Elementos Básicos

Los elementos básicos del Modelo Relacional son:

#### 2.2.3.1. Relación

Según Matemática cuando se define una relación sobre un conjunto de dominios este es un subconjunto del producto cartesiano de  $n$  dominios, donde los componentes de la relación (tupla) son valores ordenados:

$$R \subseteq D_1 \times D_2 \times \dots \times D_n$$

R: Relación

D: Dominio

$n$ : grado de la relación

<sup>44</sup> De Miguel, Adoración; Piattini, Mario. (1999). *Fundamentos y modelo de Bases de Datos*, México: Alfaomega S.A.

Este concepto no toma en cuenta los atributos por esto en base de datos se usa otro concepto que agrega los siguientes elementos: nombre, cabecera, cuerpo, esquema y estado.

- “**Nombre:** Las relaciones se identifican por un nombre ciertas relaciones que no necesitan identificarse (por ejemplo, resultados intermedios) pueden no tener nombre.
- **Cabecera de relación:** Conjunto de  $n$  pares atributo-dominio subyacente,  
 $\{ (A_i : D_i) \}_{i=1}^n$  donde  $n$  es el **grado**;  
Se corresponde con la primera fila cuando la relación se representa como tabla.  
El conjunto  $A$  de atributos sobre los que se define la relación se llama **contexto** de la misma.
- **Cuerpo de la relación:** Conjunto de  $m$  tuplas,  
 $\{ t_1, t_2, \dots, t_m \}$   
siendo cada tupla un conjunto de  $n$  pares atributo-valor:  
 $\{ (A_i : V_{ij}) \}$  siendo  $V_{ij}$  el valor  $j$  del dominio  $D_i$  asociado al atributo  $A_i$  el número de tuplas  $m$  es la **cardinalidad**.  
Mientras que la cabecera es invariante, el cuerpo varía en el transcurso del tiempo, al igual que la cardinalidad.
- **Esquema de una relación** está constituido por el nombre  $R$  -si existe- y la cabecera:  
 $R (\{ A_i : D_i \}_{i=1}^n)$  representa la parte definitoria y estática y también se denomina **intensión**; se corresponde con lo que hemos llamado *tipo* (de entidad) en el ME/R.
- **Estado  $r$  de una relación de esquema  $R$** , al que también se denominará simplemente **relación**, se representa como  **$r(R)$**  y está constituido por el esquema y el cuerpo de la relación:  
 $r(R) = \langle \text{esquema, cuerpo} \rangle$

Siendo el cuerpo el conjunto de tuplas que, en un instante dado, satisface el correspondiente esquema de relación.

También se llama **extensión**.”

Es la base del modelo relacional la cual se representa mediante una tabla.

### 2.2.3.2. Atributo

Es la propiedad de una relación, su representación es mediante una columna.

#### - **Atributos Compuestos**

Los atributos y dominios compuestos son denominados elementos únicos de información como valores atómicos.

### 2.2.3.3. Dominios

Son los valores válidos que toma un atributo.

Una base de datos relacional está formada por conjunto de dominios descritos como  $\{D_i\}$  y de relaciones  $\{R_i\}$  que se detallan sobre los dominios.

El dominio es una colección homogénea de valores atómicos y tiene las siguientes características:

#### **Dominio Simple**

Cada dominio tiene un nombre, un número finito de valores del mismo tipo y cada valor es elemental respecto al modelo relacional no podrá ser a su vez relación. Los dominios se detallan de dos formas:

#### - **Extensión:** Los valores posibles:

Por ejemplo: *meses del año = {enero, febrero, marzo, abril, mayo, junio, julio, agosto, septiembre, octubre, noviembre, diciembre}*

#### - **Intensión:** Por un tipo de dato

Edad = entero

## **Dominio Compuesto**

Se compone de dominios simples a los que se aplica restricciones de integridad. Por ejemplo el dominio fecha se compone de año, mes y día. Además de los dominios compuestos hay los atributos compuestos.

### **2.2.3.4. Tupla**

Se representa por una fila y se refiere a la ocurrencia de la relación. Los atributos y dominios compuestos son denominados elementos únicos de información como valores atómicos.

## **2.2.4. Clases de Relaciones**

### **2.2.4.1. Persistentes**

El esquema de esta relación se encuentra en la base de datos sólo se elimina con una operación explícita del usuario.

### **2.2.4.2. Base:**

Este tipo de relaciones “existen por sí mismas, no en función de otras relaciones.

- Se crean especificando explícitamente su esquema de relación (nombre y conjunto de pares atributo/dominio).
- Sus extensiones (ocurrencias de relación), al igual que su definición, también se encuentran almacenadas.”<sup>45</sup>

### **2.2.4.3. Vistas**

Las vistas son relaciones derivadas que se especifican cuando se da un nombre a cierta consulta en la base de datos. No poseen datos almacenados únicamente recopila la definición en términos de otras relaciones con nombre éstas pueden ser relaciones base, vistas o instantáneas.

---

<sup>45</sup> De Miguel, Adoración; Piattini, Mario. (1999). *Diseño de Base de Datos Relacionales*, México: Alfaomega S.A.

#### **2.2.4.4. Instantáneas**

Igual que las vistas son consideradas derivadas pero poseen propios datos almacenados que resultan de ejecutar una sentencia.

Estas relaciones no permiten actualizar cuando existen cambios en los datos de las relaciones en las que están definidas sólo cada cierto tiempo se refresca de acuerdo con lo señalado por el usuario cuando se creó.

#### **2.2.4.5. Temporales**

Una relación temporal desaparece sin necesidad de que el usuario lo elimine (esto lo diferencia de las relaciones persistentes), por ejemplo cuando se termina una determinada sesión.

### **2.2.5. Claves**

#### **2.2.5.1. Candidatas**

Una clave candidata o Candidate Key es una colección de atributos que permiten la identificación unívoca de cada tupla de una relación.

En una relación puede existir más de una clave de este tipo, cuando se da este caso se debe diferenciar entre:

- **Primary Key (Claves Primarias)**

Es una clave candidata que determina la columna como clave primaria de una tabla, lo que se requiere es que la columna no tenga valores nulos, no debe constar de valores duplicados.

- **Alternative Key (Claves Alternativas)**

Se denomina a las claves candidatas que no se ha escogido como primary key.

#### **2.2.5.2. Ajenas**

La clave ajena en una relación RA es una colección no vacía de atributos donde los valores coinciden con los valores de una clave candidata de la relación RB.

La clave ajena y candidata deben especificarse en un mismo dominio.

## 2.2.6. Restricciones

### 2.2.6.1. Inherentes

Estas restricciones son dadas por el modelo de datos, en el caso del relacional, una relación posee ciertas propiedades intrínsecas que no posee una tabla y provienen de una definición matemática.

Un conjunto en una relación:

- La prohíbe que dos tuplas sean iguales
- No interesa el orden de las tuplas al igual que se los atributos
- Un atributo puede tener un valor único de dominio.
- Permite grupos que se repitan como los atributos de una tupla.

Se debe tener en cuenta que las definiciones de tabla y relación son diferentes:

- Una tabla es la representación de una relación.
- En una tabla no existen restricciones inherentes de relación.
- En una tabla se puede encontrar dos filas iguales, éstas deben ser ordenadas por el valor de la clave primaria.
- Los atributos se ordenan como se haya especificado en la tabla.

### 2.2.6.2. Semánticas

El modelo relacional posee las restricciones semánticas en donde los usuarios puedan reflejar en el esquema la semántica del mundo real.

Las restricciones semánticas que se permiten en el modelo relacional son:

- Clave Primaria (PRIMARY KEY),
- Unicidad (UNIQUE),
- Obligatoriedad (NOT NULL),
- Integridad Referencial (FOREIGN KEY),
- Restricciones de Rechazo:
  - o Verificación (CHECK), y
  - o Aserción (ASSERTION).
- Disparador (*trigger*), incluido en SQL3 pero no en SQL92.

- **PRIMARY KEY**

Esta condición se usa para determinar la columna como clave primaria de una tabla, lo que se requiere es que la columna no tenga valores nulos, no debe constar de valores duplicados.

- **UNIQUE**

Esta condición permite describir un índice único (no admite valores duplicados), en una columna. Si se detalla esta restricción no puede existir dos filas que contengan igual valor, se usa esta sentencia para que el sistema examine que no hayan valores que ya existen.

- **NOT NULL**

Esta sentencia muestra que la columna no debe tener valores nulos, cuando se requiera indicar que una columna no tenga valor nulo se lo puede realizar sin usar la condición CONSTRAINT.

- **FOREIGN KEY**

“Sirve para definir una clave foránea sobre una columna o una combinación de columnas. Una clave foránea es una columna o conjunto de columnas que contiene un valor que hace referencia a una fila de otra tabla, en una restricción 1 se puede definir con la cláusula REFERENCES. Para definir una clave foránea en una restricción de tipo 2 debemos empezar por las palabras FOREIGN KEY después indicamos entre paréntesis la/s columna/s que es clave foránea, a continuación la palabra reservada REFERENCES seguida del nombre de la tabla a la que hace referencia, opcionalmente podemos indicar entre paréntesis el nombre de la/s columna/s donde tiene que buscar el valor de referencia, por defecto coge la clave principal de la tabla2, si el valor que tiene que buscar se encuentra en otra/s columna/s de tabla2, entonces debemos escribir el nombre de esta/s columna/s entre paréntesis.

Además sólo podemos utilizar una columna (o combinación de columnas) que esté definida con una restricción de UNIQUE, de lo contrario la sentencia CREATE TABLE nos dará un error.”<sup>46</sup>

Ejemplo:

```
CREATE TABLE tabla1(  
columna1 INTEGER,  
columna2 CHAR(20) NOT NULL,  
columna3 CHAR(15),  
columna4 INTEGER,  
CONSTRAINT pk PRIMARY KEY (columna1),  
CONSTRAINT uni1 UNIQUE (columna3)  
CONSTRAINT fk5 FOREIGN KEY (columna4) REFERENCES  
tablan);
```

- **Restricciones de Rechazo**

“El usuario formula una condición mediante un predicado definido sobre un conjunto de atributos, tuplas o dominios, que debe ser verificado en toda operación de actualización para que el nuevo estado constituya una ocurrencia válida del esquema.”<sup>47</sup>

Existen dos tipos de restricciones de rechazo:

- **Verificación (CHECK)**

Verifica en cada sentencia de actualización si el predicado es verdadero o falso. Si es falso rechaza la operación, esta restricción se especifica en un solo elemento, dentro de un CREATE TABLE.

- **Aserción (ASSERTION)**

Es similar a la Verificación pero afecta a varios elementos o tablas

---

<sup>46</sup> [http://www.aulaclac.es/sql/t\\_8\\_2.htm](http://www.aulaclac.es/sql/t_8_2.htm)

<sup>47</sup> De Miguel, Adoración; Piattini, Mario. (1999). *Diseño de Base de Datos Relacionales*, México: Alfaomega S.A.

- **Disparador**

El disparador o trigger es una restricción en donde un usuario puede definir una acción ante cierta condición.

Los triggers requieren que el usuario defina un procedimiento que se aplicará cuando una condición se cumpla.

### 2.2.7. Esquemas Relacionales

Un esquema de relación es:

"R<A:D, S>      Donde:

- R: el nombre de la relación,
- A: la lista de atributos,
- D: los dominios sobre los que están definidos los atributos, y
- S: las restricciones de integridad intraelementos (afectan a atributos y/o tuplas de una única relación).

Y el **esquema de una base de datos relacional** será:

$E < \{R_i\}, \{I_i\} >$

Donde:

- E el nombre del esquema relacional,
- $\{R_i\}$  el conjunto de esquemas de relación, y
- $\{I_i\}$  el conjunto de restricciones de integridad interelementos (afectan a más de una relación y/o dominio).

En términos de implementación en SQL92, un esquema E tendrá la siguiente forma:

$E < R, D, T, V >$

Donde:

- R el conjunto de esquemas de relación (CREATE TABLE),
- D el conjunto de definiciones de dominios (CREATE DOMAIN),
- T el conjunto de restricciones interrelación y sobre dominios (CREATE ASSERTION, CREATE TRIGGER, ...), y
- V el conjunto de vistas (CREATE VIEW).<sup>48</sup>

---

<sup>48</sup> De Miguel, Adoración; Piattini, Mario. (1999). *Diseño de Base de Datos Relacionales*, México: Alfaomega S.A.

## 2.2.8. Sistemas de Gestión de Base de Datos Relacionales

### 2.2.8.1. Reglas de Codd

Regla 0: Un Sistema de Gestión de Base de Datos relacional debe usarse para la gestión de una base de datos en especial sus opciones relacionales.

Esta es la regla genérica y de esta proceder las siguientes:

1. **Regla de información:** La información se representa en valores en columnas de filas de tablas en una base de datos.
2. **Regla de acceso garantizado:** El valor escalar individual puede direccionarse mostrando nombres de la tabla, columna, clave primaria.
3. **Tratamiento sistemático de valores nulos:** Un SGBS debe soportar la manipulación de información desconocida o que no sea aplicable.
4. **Catálogo en línea:** Denominado diccionario de datos que se basa en el modelo relacional.
5. **Sublenguaje de datos completo:** Todo SGBD soporta un lenguaje relacional.
6. **Actualización de vistas:** Las vistas deben ser actualizables teórica y prácticamente.
7. **Inserción, modificación y borrado de tuplas de alto nivel:** Las sentencias de manipulación de datos deben darse en los conjuntos de filas.

- 8. Independencia física de los datos:** Los cambios que se realicen en los procedimientos de acceso físico (almacenamiento) no debe afectar al acceso lógico.
- 9. Independencia lógica de los datos:** Las tablas con integridad no deben afectar a programas de aplicación.
- 10. Independencia de la integridad:** Los programas deben ser separados de las restricciones de integridad se las debe acumular en el diccionario de datos.
- 11. Independencia de la distribución:** Las aplicaciones no debe verse afectadas cuando se cambia su distribución que existe en una Base de Datos.
- 12. Regla de no subversión:** Las interfaces de bajo nivel no pueden alterar el sistema por evitar restricciones de seguridad o integridad.

### **2.2.9. Tratamiento de valores nulos**

Valor nulo: “Señal utilizada para representar información desconocida, inaplicable, inexistente, no válida, no proporcionada, indefinida, etc.”<sup>49</sup>

El valor nulo en base de datos se usa para lo siguiente:

- Crea tuplas con atributos cuyos valores no se conocen.
- Agrega un nuevo atributo (no tendrá valor para las tuplas) en una relación

En el tratamiento de valores nulos debe constar:

- Operaciones de comparación
- Operaciones aritméticas
- Operaciones algebraicas
- Funciones de agregación

---

<sup>49</sup> De Miguel, Adoración; Piattini, Mario. (1999). *Diseño de Base de Datos Relacionales*, México: Alfaomega S.A.

Se debe agregar operadores especiales, en operaciones de comparación de debe establecer la lógica trivaluada donde se incorpora al valor quizás.

### 2.2.10. Ejemplo:

“Transformar el siguiente diagrama E-R en un esquema relacional completo eficiente:

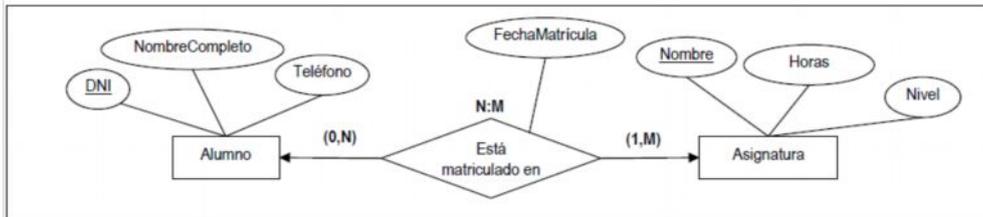


Figura 2.2.10.\_1: Diagrama E-R

NOTA: Hay que tener en cuenta que en esta base de datos se almacenará la información de matriculación de alumnos en un centro durante varios años (en las mismas tablas), por tanto, si un alumno repite podrá matricularse de la misma asignatura el siguiente año y quedará registrada en la misma base de datos esa información.

NOTA (2): “Horas” almacena el número de horas anuales del curso, que podrá tener un decimal, por ejemplo: 210 horas, 215,5 horas; y “Nivel” indica el nombre del curso, por ejemplo: “1ºASI”, “1ºBACH-CS”, “2ºESO”.

«50

**SOLUCIÓN:** Tablas, atributos y claves primarias:  
 ALUMNOS(DNI, NombreCompleto, Teléfono)  
 ASIGNATURAS(Nombre, Horas, Nivel)  
 MATRÍCULAS(DNIAlumno, NombreAsignatura, FechaMatricula)

Diagramas referenciales:

MATRÍCULAS  $\xrightarrow{\text{DNIAlumno}}$  ALUMNOS: DNI del alumno que realiza la matrícula.

MATRÍCULAS  $\xrightarrow{\text{NombreAsignatura}}$  ASIGNATURAS: Nombre identificativo de la asignatura en la que se matricula el alumno.

Dominios de los atributos:

DNI, DNIAlumno: Cadena de 9 caracteres.  
 NombreCompleto, Nombre, NombreAsignatura: Cadena de 40 caracteres.  
 Horas: Número de 3 cifras enteras y 1 decimal.  
 FechaMatricula: Fecha.  
 Nivel: Cadena de 10 caracteres.

Figura 2.2.2.\_2: Modelo Relacional

[Volver Inicio](#)



## U2. Lección 2 tarea 1

### ACTIVIDAD DE APRENDIZAJE

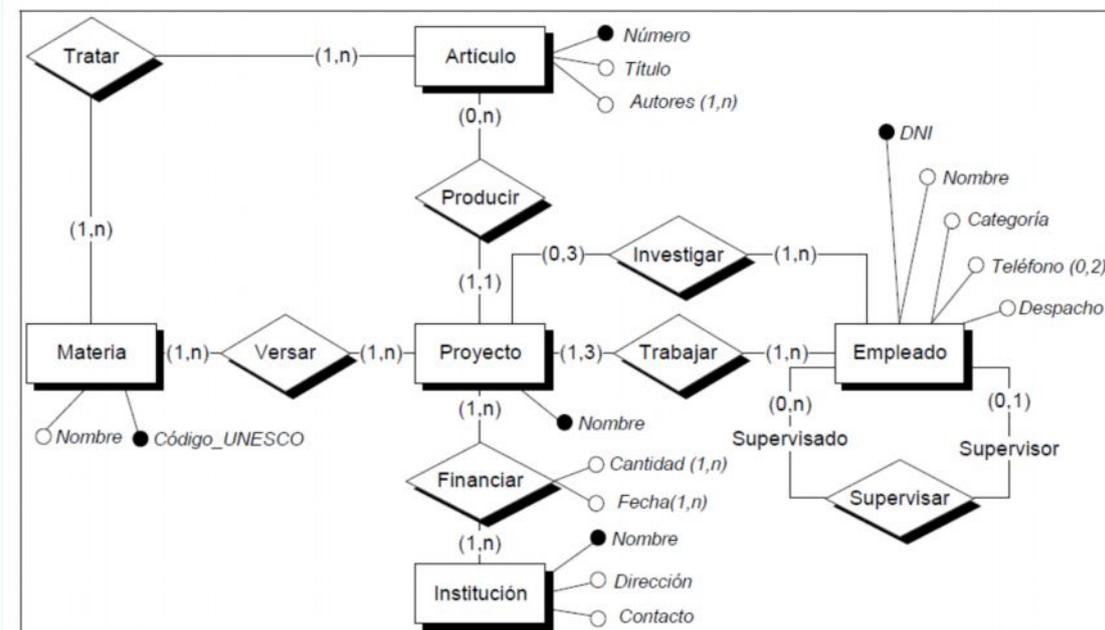
Denominación de la tarea:

Obtener el Modelo Relacional

### ASESORÍA DIDÁCTICA

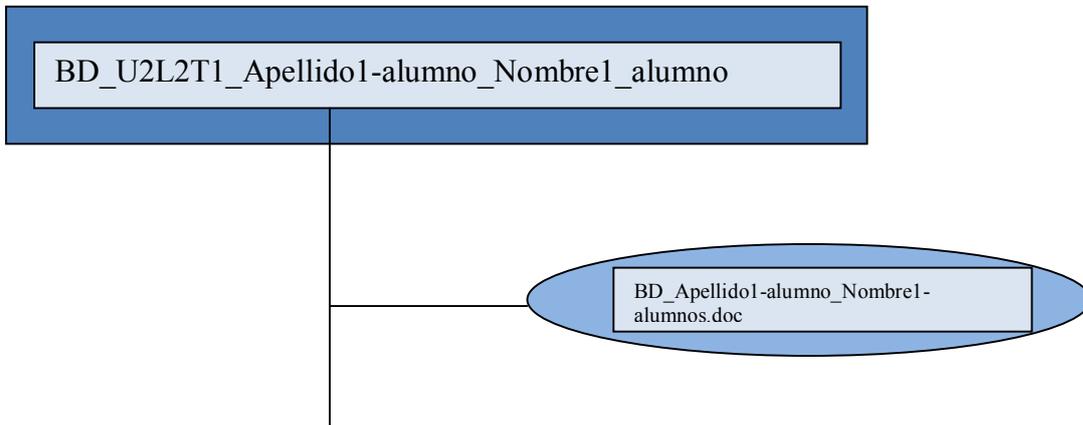
Para desarrollar la siguiente actividad de aprendizaje, debe leer y comprender el contenido de la Lección 2 Unidad 2.

### ACTIVIDAD DE APRENDIZAJE



Envíe a su tutor la tarea en una carpeta comprimida como .rar

Formato de la carpeta





## Resumen Lección 2

---

### Modelo Relacional

Esta lección ha presentado una visión global del modelo Relacional, incluyendo:

- **Elementos Básicos**
- **Clases de Relaciones**
- **Claves**
- **Restricciones**
- **Esquemas Relacionales**
- **Sistemas de Gestión de Base de Datos Relacionales**
- **Tratamiento de Valores Nulos**

#### **Destacándose que:**

Los elementos básicos del modelo relacional son: Relación, Atributo, Dominio, Tupla.

Las clases de relaciones son:

- Persistentes
- Base
- Vistas
- Instantáneas
- Temporales

Las claves se clasifican en:

- Candidatas
- Ajenas

Las restricciones son:

- Inherentes
- Semánticas

Los Sistemas de Gestión de Base de Datos Relacionales presentan las reglas de Codd:

- Regla de información
- Regla de acceso garantizado
- Tratamiento sistemático de valores nulos
- Catálogo en línea
- Sublenguaje de datos completo
- Actualización de vistas
- Inserción, modificación y borrado de tuplas de alto nivel
- Independencia física de los datos
- Independencia lógica de los datos
- Independencia de la integridad
- Independencia de la distribución
- Regla de no subversión

El valor nulo en base de datos se usa para lo siguiente:

- Crea tuplas con atributos cuyos valores no se conocen.
- Agrega un nuevo atributo (no tendrá valor para las tuplas) en una relación.

## UNIDAD 2: PROCESOS DE DESARROLLO DE LA BASE DE DATOS

### LECCIÓN 2.3: Lenguajes Relaciones



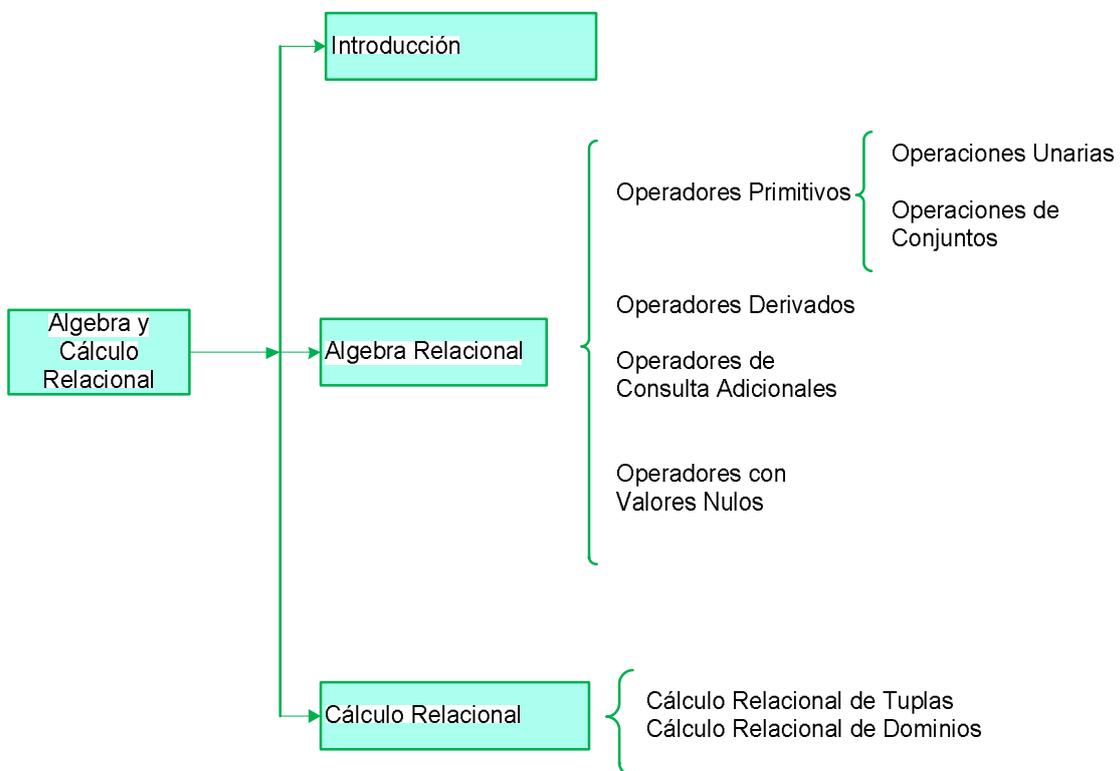
#### Visión General

En esta lección se trata el álgebra relacional y el cálculo relacional dentro de una base de datos. El álgebra relacional es una colección de operaciones que calculan las respuestas sobre relaciones y como se muestran en el modelo relacional. El cálculo relacional es como un lenguaje de consulta que detalla una respuesta sobre la base de datos.

[Volver Inicio](#)



#### Plan de la Lección





## Desarrollo de la Lección

### Álgebra y Cálculo Relacional

#### 2.3.1. Introducción

El álgebra relacional describe como un lenguaje con operaciones que se aplican a las relaciones, con el propósito de describir otra relación sin cambiar las originales. Los operandos y los resultados son relaciones. En las expresiones del álgebra relacional se debe definir de manera explícita el orden, mientras que el cálculo relacional sólo importa lo que se va a extraer. Al cálculo relacional también se lo conoce como cálculo de predicados, cuando se emplea en la base de datos se presenta de dos formas: cálculo relacional de tuplas y cálculo relacional de dominios.

#### 2.3.2. Álgebra Relacional

Se conoce como álgebra relacional a las expresiones con operaciones que se emplean en las relaciones con el objetivo de establecer otra relación sin alterar las relaciones dadas originalmente, de esto se obtiene que los operandos y los resultados se mantengan como relaciones. El álgebra relacional establece los operadores (semejante a los operadores aritméticos  $+$ ,  $-$ ) que trabajan en las tablas. Se debe tener claro lo que se desea conseguir para poder obtenerlo.

##### 2.3.2.1. Operadores Primitivos

Las operaciones fundamentales del álgebra relacional son:

- Seleccionar
- Proyectar
- Unión
- Diferencia
- Intersección
- Producto Cartesiano

De las cuales las operaciones Seleccionar y Proyectar se las denomina **Operaciones Unarias** y a las operaciones Unión, Diferencia, Intersección y Producto Cartesiano de las denomina **Operaciones de Conjuntos**.

- **Operaciones Unarias**

A estas operaciones se las conoce como unarias porque operan directamente sobre una tabla.

- **Selección ( $\sigma$ )**

Esta operación permite escoger de un subconjunto de tuplas en una determinada relación aquellas que pertenecen a una condición establecida. La operación Seleccionar tiene la siguiente estructura:

“ $\sigma$  (condición) (RELACION)

Esta operación es conmutativa, es decir:

$$\sigma (\text{condición1}) (\sigma (\text{condición2})(R)) = \sigma (\text{condición2}) (\sigma (\text{condición1}) (R))^{51}$$

- **Proyección ( $\pi$ )**

Esta operación retorna una relación con columnas excluidas con lo que se evita obtener igual información o que no corresponda a una determinada consulta, eliminando aquellos que estén duplicados. La operación proyectar tiene la siguiente estructura:

“ $\pi$ <lista de atributos> (RELACION)”<sup>52</sup>

---

<sup>51</sup> Sabana, Maribel. (2006). *Modelamiento e Implementación de Base de Datos*: Lima: Megabyte Grupo Editorial.

<sup>52</sup> Sabana, Maribel. (2006). *Modelamiento e Implementación de Base de Datos*: Lima: Megabyte Grupo Editorial.

- **Operaciones de Conjuntos**

Las operaciones de conjuntos son un subgrupo de las operaciones binarias que permiten realizar una combinación de información de varias relaciones a diferencia de las operaciones unarias como se vio anteriormente que operan sobre una tabla.

Las operaciones de conjuntos son las siguientes:

- **Unión**

Esta operación de conjuntos agrupa o suma las tuplas de una relación con las de otra relación, esto se realiza siempre y cuando las relaciones posean igual número de atributos y sus dominios sean similares.

- **Diferencia**

Esta operación de conjuntos indica que de la diferencia de dos relaciones se obtiene un resultado, éste debe contener las tuplas que se encuentran en la primera relación pero que no se encuentren en la segunda relación. Su estructura es la siguiente:

“Tabla\_A **MINUS** Tabla\_B;”

- **Producto Cartesiano**

Esta operación es la vinculación de cada una de las tuplas de una primera relación con todas las tuplas de una segunda relación; para entender de mejor forma, el producto cartesiano realiza una multiplicación de dos relaciones y se obtiene una relación resultado que se conforma por las posibles combinaciones de tuplas de las dos relaciones. La estructura de esta operación es la siguiente:

“Tabla\_A x Tabla\_B”

### 2.3.2.2. Operadores Derivados

- **Combinación**

La combinación theta describe una relación que tiene tuplas del producto cartesiano de A y B que determinan el predicado P.

- **Combinación Natural**

Es una equicombinación de todos los atributos de dos relaciones que tienen el mismo nombre, el grado de esta combinación es la sumatoria de los grados de las dos relaciones menos el número de atributos.

- **Intersección**

Esta operación de conjuntos determina que existe una relación que contenga las tuplas de la primera y segunda relación, esta operación se efectúa cuando se cumpla la norma establecida en la unión. La estructura de esta operación es la siguiente:

“Tabla\_A INTERSEC Tabla\_B”

- **División**

La división describe una relación sobre los atributos C que está conformada por una colección de tuplas A que se relacionan con una combinación de todas las tuplas de B.

### 2.3.2.3. Operadores de Consulta Adicionales

- **Agrupación (Group By)**

Agrupar tuplas en subconjuntos que tienen valores en común en algunos atributos. Group By permite en su sintaxis funciones de agregación (SUM, COUNT, AVG, etc).

- **Cierre Transitivo**

“Operación unaria definida sobre dos atributos compatibles (que comparten el mismo dominio), que se obtiene por sucesivas operaciones de combinación, proyección y unión, consistente en añadir a la relación de origen todas las tuplas que se deducen sucesivamente, por transitividad hasta la saturación: *si existen las tuplas (a,b) y (b,c) se añade (a,c)*”<sup>53</sup>

---

<sup>53</sup> Piattini, Mario; Marcos, Esperanza; Calero, Coral y Vela, Belén. (2006). *Tecnología y Diseño de Base de Datos*. México: Alfaomega

#### 2.3.2.4. Operadores con Valores Nulos

- **Operadores Maybe**

Estos operadores usan la lógica trivaluada (en el resultado agrega las tuplas que van a ser ciertas).

#### 2.3.3. Cálculo Relacional

El cálculo relacional se clasifica en:

- Cálculo relacional de dominios
- Cálculo relacional de tuplas.

##### 2.3.3.1. Cálculo Relacional de Tuplas

Este cálculo se deriva del cálculo de predicados el cual tiene las siguientes características:

- Todas las variables se unifican a las tuplas.
  - Operadores lógicos como el “NOT”, “AND” y “OR”, el operador universal y el operador existencial son admitidos.

El cálculo relacional de tuplas tiene la siguiente estructura:

“ [DEFINICION DE LAS VARIABLES DE TUPLA]  
OPERADOR OBJETIVO PREDICADO”

**Definición de las Variables de Tupla.** En esta sentencia se expresa una variable para desplazar sobre las tuplas de una relación. “Sobre una misma relación es posible definir más de una variable de tupla, y en una misma expresión pueden intervenir varias variables de tuplas referidas a la misma o a distintas relaciones. Cuando no exista ambigüedad en la consulta se puede prescindir de la declaración de las variables de tupla. La asignación de la variable a las tuplas de la relación es dinámica, de modo que se puede dar el mismo nombre a las distintas variables que intervienen en consultas distintas.”<sup>54</sup>

---

<sup>54</sup> Piattini, Mario; Marcos, Esperanza; Calero, Coral y Vela, Belén. (2006 ). *Tecnología y Diseño de Base de Datos*. México: Alfaomega

**Operador.** Establece la actividad que se debe efectuar con los datos escogidos (insertar, modificar, borrar, recuperar).

**Objetivo.** Determina los atributos que se va a recuperar. “Si en la lista de atributos (target list) aparecen relaciones distintas, se realiza el producto cartesiano de las correspondientes relaciones y se aplica la restricción contenida en el predicado y después la proyección sobre los atributos de la lista (el orden de las operaciones lo establecerá el sistema de acuerdo con los criterios de optimización incluidos en el mismo). Como pueden aparecer atributos de distintas relaciones con el mismo nombre, es necesario cualificar los atributos, es decir, dar el nombre de la relación (o de la variable de tupla) seguido de un punto y del nombre del atributo. Así un ejemplo de lista objetivo sería:

$$R1.A1_1, R1.A1_2, \dots, R1.A1_n, R2.A2_1,$$

Donde R1 y R2 son nombres de las relaciones o de variables de tupla, definidas, en este último caso, en sentencias tipo:

RECORRIDO de R1 es S1

RECORRIDO de R2 ES S2

Donde las variables de tupla R1 y R2 se han definido sobre las relaciones S1 y S2 de la base de datos  $A1_1, A1_2, \dots, A1_n, \dots$  son atributos de las respectivas relaciones, pudiendo ocurrir que algunos de los nombres de los  $A1_i$  coincida con alguno de los  $A2_j$  por lo que es necesario cualificar los nombres de los atributos añadiéndoles los nombres de las relaciones.”<sup>55</sup>

**Predicado.** Detalla el condicionamiento que se debe comprobar en las tuplas para su elección, esto se realiza usando los símbolos del cálculo de predicado, la formulación debe hacerse correctamente.

**Área.** Es un lugar de trabajo que los lenguajes de programación usan en donde se lee o escribe relaciones de la manera en el que el lenguaje las maneja.

---

<sup>55</sup> Piattini, Mario; Marcos, Esperanza; Calero, Coral y Vela, Belén. (2006). *Tecnología y Diseño de Base de Datos*. México: Alfaomega

Dentro de los lenguajes del cálculo relacional de tuplas están: ALPHA, QUEL y INGRES.

Algunas de las funciones del lenguaje ALPHA son:

PUT: Esta función sirve para insertar tuplas.

GET: Esta función sirve para recuperar una relación.

DELETE: Esta función permite borrar tuplas.

“La sintaxis de una operación sería: operador área (lista objetivo): predicado”<sup>56</sup>

### **2.3.3.2. Cálculo Relacional de Dominios**

El cálculo relacional de dominio se fundamenta en el cálculo de predicados, se diferencia de cálculo relacional de tuplas porque utiliza el dominio en vez de las tuplas, las variables de dominio se delimitan sobre un dominio.

En este dominio prevalecen las siguientes características:

- Las variables están ligados a los dominios de los atributos de una relación (atributoA: variableA, atributoB: variableB, atributoC: variableC, ....). Por ejemplo: Modelo de Portátil (marca: A, modelo B).
- Se forman los predicados igual que el cálculo relacional de tuplas.

Uno de los lenguajes principales del cálculo relacional de dominios es el QBE (Query-By-Example), este se basa en la utilización de un editor de pantallas que realiza consultas.

---

<sup>56</sup> Piattini, Mario; Marcos, Esperanza; Calero, Coral y Vela, Belén. (2006). *Tecnología y Diseño de Base de Datos*. México: Alfaomega

#### 2.3.4. Ejemplos:

Para la siguiente Base de Datos Relacional:

SINDICATO (Nombre\_Federación, Dirección, Teléfono)

MIEMBRO (Codigo, Nombre\_Miembro, Título)

COMPOSICIÓN (Nombre, Codigo, Cargo, Fecha\_Inicio)

Se solicita las siguientes consultas algebraicas:

- Obtener el nombre del presidente del sindicato:

*Solución:*

$$\Pi_{\text{Nombre\_Federación}}(\sigma_{\text{Cargo} = \text{'Presidente'}}(\text{COMPOSICION}) * \text{MIEMBRO})$$

- Obtener los sindicatos que tienen todos los cargos

*Solución:*

$$\Pi_{\text{Nombre\_Federación, Cargo}}(\text{COMPOSICION}) \div \Pi_{\text{Cargo}}(\text{COMPOSICIÓN})$$

Para la siguiente Base de Datos Relacional

SALA (Código\_Sala, Capacidad)

PELÍCULA (Código\_Película, Título\_Película, Tipo\_Película)

PROYECCIÓN (Codigo\_Sala, Código\_Película, Fecha\_Inicio, Fecha\_Fin)

Se solicita las siguientes consultas algebraicas:

- Mostrar las salas en las que se ha proyectado todas las películas

*Solución:*

$$\Pi_{\text{Código\_Sala, Código\_Película}}(\text{PROYECCION}) \div \Pi_{\text{Código\_Película}}(\text{PELÍCULA})$$

- Mostrar las salas que proyecten películas de tipo Terror

*Solución:*

$$\Pi_{\text{Código\_Sala}}(\text{PROYECCION}) - \Pi_{\text{Código\_Sala}}(\text{PROYECCIÓN} * \sigma_{\text{Tipo\_Película} \neq \text{'TERROR'}}(\text{PELÍCULA}))$$

[Volver Inicio](#)



## U2. Lección 3 tarea 1

### ACTIVIDAD DE APRENDIZAJE

---

Denominación de la tarea:

Consultas usando Algebra Relacional

Para la siguiente base relacional:

MAQUINARIA (Código\_Maquinaria, Tipo, PreciHora\_Alquiler

HACIENDA (Código\_Hacienda, Nombre, Extensión)

EMPLEADO (Código\_Empleado, Nombre, Dirección, Teléfono)

PIEZAS(Código\_Empleado, Código\_Maquinaria, Código\_Hacienda, Fecha, Tipo\_Faenamiento, Tiempo)

### **ASESORÍA DIDÁCTICA**

Para desarrollar la siguiente actividad de aprendizaje, debe leer y comprender el contenido de la lección 3 Unidad 2.

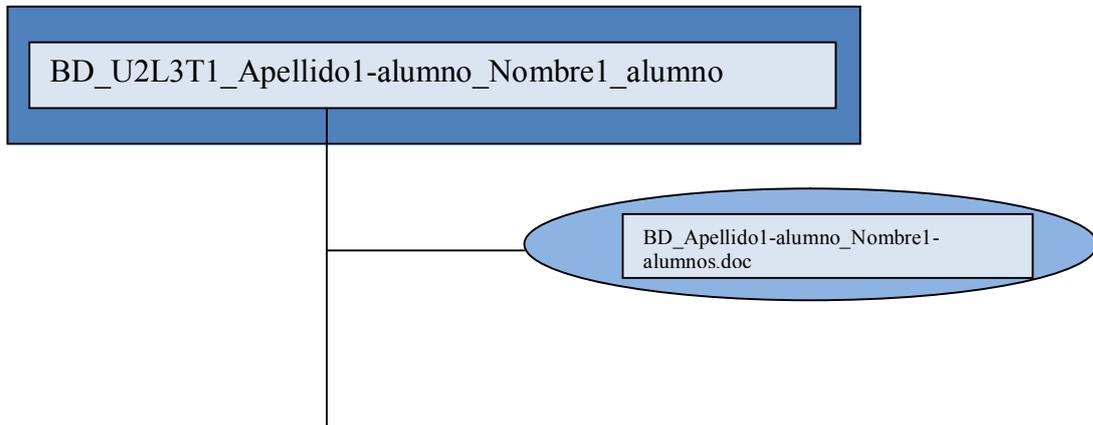
### **ACTIVIDAD DE APRENDIZAJE**

Solucionar por medio de algebra relacional las siguientes consultas:

1. Mostrar todos los Empleados que se usan las máquinas del tipo A.
2. Mostrar el valor de Maquinaria para las máquinas que nunca se ha usado en algún trabajo.
3. Mostrar todas las haciendas en las que se ha realizado trabajos con las máquinas M1 y M2.

Envíe a su tutor la tarea en una carpeta comprimida como .rar

Formato de la carpeta:





## U2. Lección 3 tarea 2

### ACTIVIDAD DE APRENDIZAJE

---

Denominación de la tarea:

Consultas usando Algebra Relacional

Para la siguiente base relacional:

ALMACEN (Código\_Almacén, Nombre\_Almacén, Dirección\_Almacén, Teléfono)

MUEBLES (Código\_Mueble, Descripción, Precio)

CLIENTE (Código\_Cliente, Nombre\_Cliente, Dirección\_Cliente)

VENTA (Código\_Tienda, Código\_Mueble, Código\_Cliente, Descuento)

### **ASESORÍA DIDÁCTICA**

Para desarrollar la siguiente actividad de aprendizaje, debe leer y comprender el contenido de la lección 3 Unidad 2.

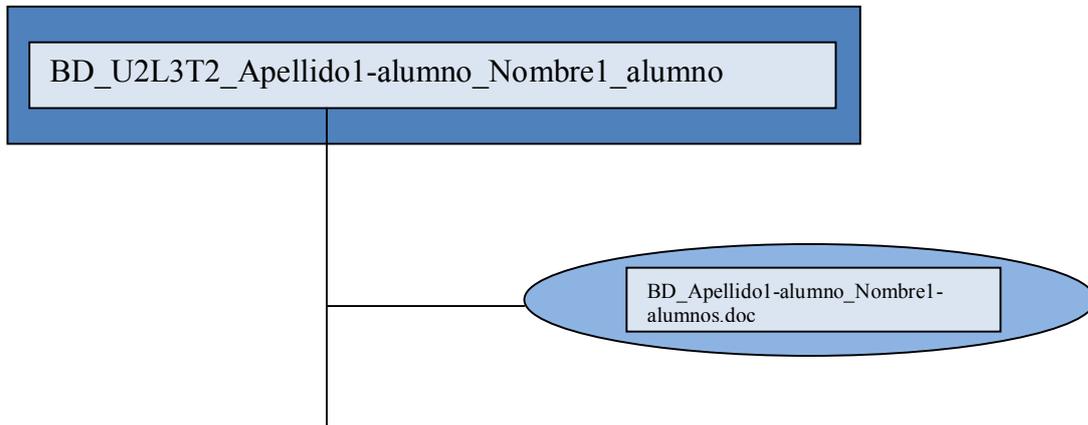
### **ACTIVIDAD DE APRENDIZAJE**

Solucionar por medio de algebra relacional las siguientes consultas:

4. Mostrar los nombre y direcciones de los cliente que hayan comprado un mueble con un descuento mayor al 25%
5. Mostrar los precios y descripciones de los muebles que se han vendido en todos los almacenes.
6. Mostrar los nombres de los almacenes que no hayan vendido el mueble A.
7. Mostrar los códigos y nombres de los almacenes que han hecho descuentos menores al 15%.

Envíe a su tutor la tarea en una carpeta comprimida como .rar

Formato de la carpeta:





## Resumen Lección 3

---

### + Lenguajes Relacionales

Esta lección ha presentado una visión global de los Lenguajes Relacionales, incluyendo:

- **Algebra Relacional**
  - **Operadores Primitivos**
  - **Operadores Derivados**
  - **Operadores de Consulta Adicional**
  - **Operadores con Valores Nulos**
- **Cálculo Relacional**
  - **Cálculo Relacional de Tuplas**
  - **Cálculo Relacional de Dominios**

#### **Destacándose que:**

El Algebra Relacional se conoce como álgebra relacional a las expresiones con operaciones que se emplean en las relaciones con el objetivo de establecer otra relación sin alterar las relaciones dadas originalmente, de esto se obtiene que los operandos y los resultados se mantengan como relaciones.

#### **Los Operadores Primitivos**

Las operaciones fundamentales del álgebra relacional son:

- Seleccionar
- Proyectar
- Unión
- Diferencia
- Intersección
- Producto Cartesiano

Las Operaciones Unarias son:

- Selección
- Proyección

Las Operaciones de Conjuntos son:

- Unión
- Diferencia
- Producto Cartesiano

Los Operadores Derivados son:

- Combinación
- Combinación Natural
- Intersección
- División

Los Operadores de Consulta Adicional son:

- Agrupación (GROUP BY)
- Cierre Transitivo

Los Operadores con Valores Nulos son:

- Operadores Maybe

El Cálculo Relacional se clasifica en:

- Cálculo Relacional de Tuplas

Este cálculo se deriva del cálculo de predicados el cual tiene las siguientes características:

Todas las variables se unifican a las tuplas.

Operadores lógicos como el "NOT", "AND" y "OR", el operador universal y el operador existencial son admitidos.

- Cálculo Relacional de Dominios

El cálculo relacional de dominio se fundamenta en el cálculo de predicados, se diferencia de cálculo relacional de tuplas porque utiliza el dominio en vez de las tuplas, las variables de dominio se delimitan sobre un dominio.

## UNIDAD 2: PROCESO DE DESARROLLO DE LA BASE DE DATOS

### LECCION 2.4: Creación y Desarrollo de Bases de Datos



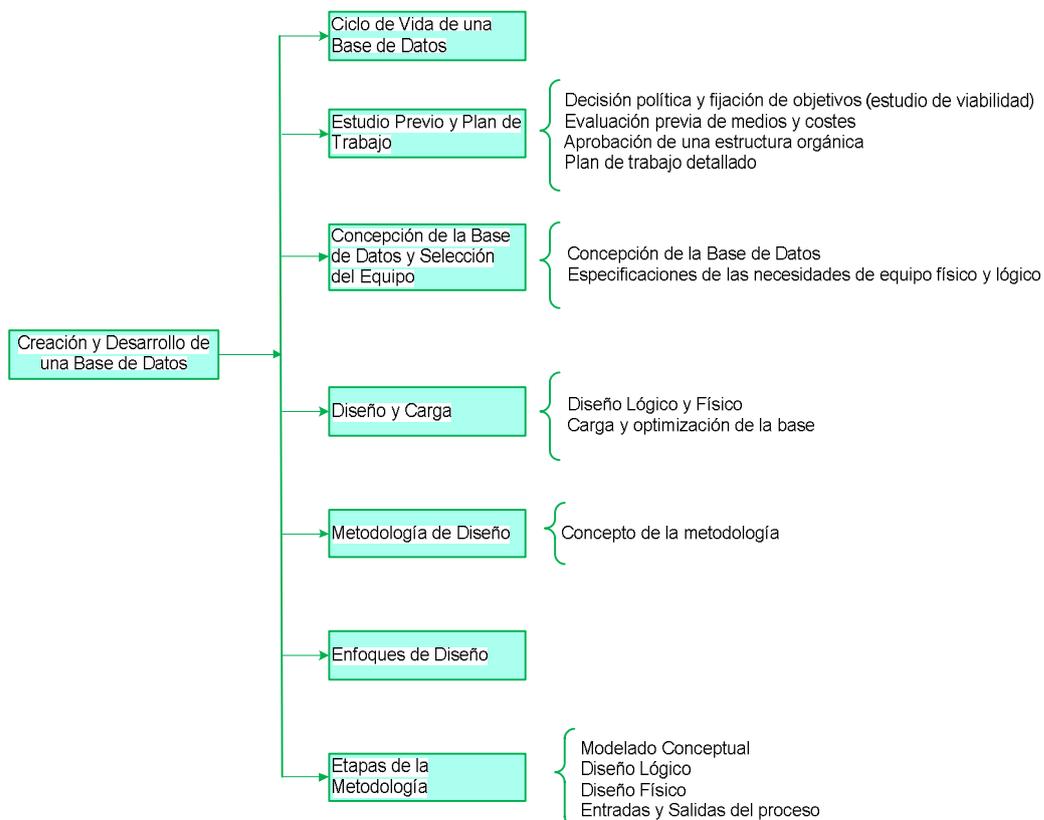
#### Visión General

En esta lección se trata la Creación y Desarrollo de la Base de Datos, el ciclo de vida de las mismas, su estudio previo y plan de trabajo, metodología y enfoques de diseño; además las etapas de las metodologías como los modelos conceptual, lógico y físico que permitirán modelar una base de datos.

[Volver Inicio](#)



#### Plan de la Lección





## **Desarrollo de la Lección**

### **2.4.1. Ciclo de Vida de una Base de Datos**

La creación de una base de datos no siempre resulta fácil, no solamente implica a la parte técnica (informáticos) sino a todos los niveles de una empresa (directivos, usuarios).

### **2.4.2. Estudio Previo y Plan de Trabajo**

#### **2.4.2.1. Decisión política y fijación de objetivos (estudio de viabilidad)**

Es la fase de previo análisis en la que se especifica la voluntad de directivos que colaborarán en el proyecto, detallando objetivos claros para alcanzar el desarrollo propuesto.

Los técnicos intervienen limitadamente en el estudio de viabilidad que es de corta duración. Este estudio es importante para lograr que el sistema de información a desarrollar se integre con toda la organización.

Además los técnicos implicados en este proceso deben tener el apoyo de los directivos quienes deben conocer que el proyecto y entender su alcance.

#### **2.4.2.2. Evaluación previa de medios y costes**

Cuando la dirección inicie las operaciones que conlleven al sistema de base de datos y se ha establecido los objetivos generales del proyecto se debe hacer una evaluación de los recursos (medios) y costos que se requerirá.

Se determinará el costo general del proyecto ya que sin un estudio profundo no se puede determinar un análisis detallado pero se debe los responsables deben presentar una cifra aproximada de los gastos y medios requeridos.

#### **2.4.2.3. Aprobación de una estructura orgánica**

Se debe definir la unidad administrativa al inicio del desarrollo del sistema que controlará la base de datos y define la estructura y elementos del equipo de desarrollo.

Desde el inicio del proyecto el administrador de la base de datos debe conocer sus funciones, teniendo en cuenta aspectos como contenido, actualización, estandarización de la base de datos.

No se ha determinado una norma general que establezca cual es la mejor organización o que verifique las funciones de un administrador de base de datos porque depende de cada organización.

La fase antes de establecer la concepción de la base de datos es la organización de la misma y del quienes intervienen en la creación. La dirección de la organización es quien debe tomar decisiones y aprobar la estructura del equipo a desarrollar el proyecto.

Los aspectos como especificar quién usará la base de datos, como se realizará la actualización de datos igualmente es función de la dirección, el administrador de la base de dato conjuntamente con los usuarios redactará las reglas de este proceso.

#### **2.4.2.4. Plan de trabajo detallado**

En este plan se describen las fases con los plazos que se requiere para cada una de ellas.

Se obtiene mejores resultados en el desarrollo de sistemas cuando no se integra al mismo tiempo todas las aplicaciones de una base de datos y se logra los siguientes objetivos:

- Con la experiencia de verifica los errores y solución de los mismos.
- Con la evolución de adaptará a los usuarios informáticos y no informáticos.
- Los resultados se obtendrán en corto plazo y esto beneficia a directivos y usuarios.

En la planificación se debe contar con los usuarios por lo que deben estar de acuerdo en la participación que tendrán.

La dirección aprueba el plan de trabajo detallado antes de pasar a otra etapa y si es rechazado implica nuevamente la elaboración del mismo.

### **2.4.3. Concepción de la Base de Datos y selección de equipo**

En esta etapa se realiza un análisis de la información que se va a integrar en una base de datos para lograr los objetivos deseados y se detalla en un modelo conceptual en el Sistema de Gestión de Base de Datos que se vaya a usar.

#### **2.4.3.1. Concepción de la Base de Datos**

Se analiza el sistema existente y las necesidades de los usuarios (funciones que debe integrar y modificaciones de la base de datos). Con las anteriores descripciones se logrará a un listado de informaciones que la organización requiere y requisitos del sistema.

Actividades de la organización que serán analizadas en términos del sistema y subsistema.

Esto permitirá verificar las características del sistema y su arquitectura además el contenido de la base de datos.

Con el contraste del esquema conceptual se termina la fase de concepción que muestra la realidad y es donde se realiza ciertos cambios hasta en un proceso iterativo lograr la mejor representación de esquemas externos de cada usuario.

#### **2.4.3.2. Especificaciones de las necesidades de equipo físico y lógico**

Luego de la fase de concepción es necesario verificar lo requerido en equipo, SGBD y características del computador (memoria, procesador, etc).

Para elegir a un SGBD se debe realizar un estudio de los que existen a la actualidad sus características para escoger al que se adapte de mejor forma a lo requerido por la compañía.

#### **2.4.4. Diseño y Carga**

En esta fase se da el diseño lógico y la codificación de la carga de datos además pruebas de programas.

##### **2.4.4.1. Diseño Lógico y Físico**

En la fase anterior se obtuvo el modelo conceptual el cual debe tener su estructura de acuerdo al SGBD que se vaya a usar. Luego de obtener la estructura lógica de obtendrá la física.

##### **2.4.4.2. Carga y optimización de la base**

Una vez que se define la estructura física de la base de datos de debe cargar datos. La mayor parte de datos resultan de automatizaciones anteriores en donde sólo se procederá a cargar sólo los archivos, teniendo en cuenta que los SGBD soportan migración.

Cuando se realice un plan de trabajo se debe contar con esta etapa que resulta costosa tanto económicamente y en tiempo, si los datos no se están en soporte de computador.

Conjuntamente con esta etapa se debe desarrollar la fase de diseño los programas y procesos para la implementación de reglas de gestión de la fase de concepción.

Cuando se encuentre cargados los archivos en la base de datos de debe realizar las respectivas pruebas para verificar el rendimiento y realizar ajustes en los diseños lógicos y físicos.

Cuando la base de datos es bien desarrollada en su diseño la independencia de las estructuras permite la que la base sea óptima.

### **2.4.5. Metodología de Diseño**

La metodología para el desarrollo de la base de datos contempla lo siguiente:

#### **2.4.5.1. Concepto de la Metodología**

El diseño de una base de datos se ha enfocado a la teoría de normalización cuando se debe incluir etapas que van desde la concepción hasta la implementación.

Según “Una metodología de diseño puede concebirse como un conjunto de herramientas y técnicas empleadas dentro de un marco organizacional que puede ser aplicado consistentemente a proyectos sucesivos de desarrollo de la estructura de una base de datos”

Otra definición importante es:

“Una metodología es un conjuntos de modelos, lenguajes y otras herramientas que nos facilitan la representación de los datos en cada fase del proceso de diseño de una base de datos, junto con las reglas que permiten el paso de una fase a la siguiente”.

A continuación se describirá cada uno de estos elementos de la metodología:

- **Herramienta:**

Se conoce como herramienta al recurso particular que usa la metodología para realizar ciertas operaciones.

- **Modelo de datos:**

Es la colección de conceptos, normas que detallan y manipulan datos de un mundo real.

- **Lenguaje de datos:**

Se basa en un lenguaje de datos se define de la sintaxis de éste con lo que se muestra un esquema.

- **Documentación:**

Muestra los resultados en forma normalizada.

#### **2.4.6. Enfoques de Diseño**

Uno de los inconvenientes en la metodología definida es la comunicación entre las personas que participan en el proceso ya que cada uno tiene pensamientos distintos pero al agruparse y trabajar en equipo debe desarrollar un proyecto útil.

Por las siguientes causas un diseño es equivocado:

- No existe conocimiento de la aplicación por parte del diseñador pero si lo conoce el usuario.
- No existe experiencia en el modelado en el caso del usuario pero que el diseñador si posee.

La falta de comunicación entre el diseñador y usuario se resuelve con un enfoque del modelo entidad – relación.

El MER muestra el diálogo entre el usuario y diseñador además representa los objetos que se van a modelar.

#### **2.4.7. Etapas de la Metodología**

Las etapas de la metodología de la base de datos son:

##### **2.4.7.1. Modelado Conceptual**

Permite obtener la representación de los recursos de información de cierta compañía, muestra la independencia de aplicaciones y usuarios.

##### **2.4.7.2. Diseño Lógico**

En este diseño se adapta un modelo conceptual en modelo de datos por medio de un SGBD.

##### **2.4.7.3. Diseño Físico**

Este diseño logra la implementación eficiente del diseño lógico.

Las etapas del diseño de la base de datos tienen relación con las del diseño clásico:

- Análisis funcional: conforma el modelo conceptual, a partir de los requisitos se obtiene el esquema.
- Diseño: compuesto por los diseños lógico y físico.

#### **2.4.8. Entradas y Salidas del proceso**

En el desarrollo de una base de datos existen las siguientes entradas y salidas:

##### **Entradas:**

- **Requisitos de información y objetivos:**

Los requisitos para el desarrollo de la base de datos se obtienen de entrevistas y análisis con los usuarios. Además se plantean los objetivos del proyecto.

- **Requisitos de los procesos:**

Se denomina requisitos a las características que deben tener las aplicaciones como el tiempo de respuesta de cierto proceso.

- **Especificaciones del SGBD:**

Determinar las herramientas a usar como las CASE que pueden ser proporcionadas por los SGBD.

- **Configuración del equipo físico y del sistema operativo:**

Interviene en el diseño y desarrollo de la base de datos.

##### **Salidas:**

- **Estructuras lógicas de datos:**

Del proceso de desarrollo se obtiene los diseños conceptual y lógico.

- **Estructura de almacenamiento:**

Corresponde al esquema interno donde se detalla los esquemas del diseño físico del sistema.

- **Normativa de explotación:**

Corresponde a la seguridad de la base de datos.

- **Especificaciones para los programas de aplicación:**

Ciertas condiciones que deben cumplir los programas de aplicación en especial sobre el mantenimiento de seguridad de la base de datos.

[Volver Inicio](#)



## Resumen Lección 4

---

### Creación y Desarrollo de Base de Datos

Esta lección ha presentado una visión global de la Creación y Desarrollo de Base de Datos, incluyendo:

- **Ciclo de Vida de una Base de Datos**
- **Estudio Previo y Plan de Trabajo**
- **Concepción de la Base de Datos y Selección del Equipo**
- **Diseño y Carga**
- **Metodología de Diseño**
- **Enfoques de Diseño**
- **Etapas de la Metodología**

#### **Destacándose que:**

La creación de una base de datos no siempre resulta fácil, no solamente implica a la parte técnica (informáticos) sino a todos los niveles de una empresa (directivos, usuarios).

#### **Decisión política y fijación de objetivos (estudio de viabilidad)**

Es la fase de previo análisis en la que se especifica la voluntad de directivos que colaborarán en el proyecto, detallando objetivos claros para alcanzar el desarrollo propuesto.

#### **Evaluación previa de medios y costes**

Cuando la dirección inicie las operaciones que conlleven al sistema de base de datos y se ha establecido los objetivos generales del proyecto se debe hacer una evaluación de los recursos (medios) y costos que se requerirá.

### **Aprobación de una estructura orgánica**

Se debe definir la unidad administrativa al inicio del desarrollo del sistema que controlará la base de datos y define la estructura y elementos del equipo de desarrollo.

### **Plan de trabajo detallado**

En este plan se describen las fases con los plazos que se requiere para cada una de ellas.

Se obtiene mejores resultados en el desarrollo de sistemas cuando no se integra al mismo tiempo todas las aplicaciones de una base de datos

### **Concepción de la Base de Datos y selección de equipo**

En esta etapa se realiza un análisis de la información que se va a integrar en una base de datos para lograr los objetivos deseados y se detalla en un modelo conceptual en el Sistema de Gestión de Base de Datos que se vaya a usar.

### **Diseño y Carga**

En esta fase se da el diseño lógico y la codificación de la carga de datos además pruebas de programas.

### **Metodología de Diseño**

La metodología para el desarrollo de la base de datos contempla lo siguiente:

- Concepto de la Metodología
- Enfoques de Diseño
- Etapas de la Metodología

Las etapas de la metodología de la base de datos son:

- **Modelado Conceptual**

Permite obtener la representación de los recursos de información de cierta compañía, muestra la independencia de aplicaciones y usuarios.

- **Diseño Lógico**

En este diseño se adapta un modelo conceptual en modelo de datos por medio de un SGBD.

- **Diseño Físico**

Este diseño logra la implementación eficiente del diseño lógico.

## UNIDAD 2: PROCESO DE DESARROLLO DE LA BASE DE DATOS

### LECCIÓN 2.5: Diseño Conceptual, Diseño Lógico y Diseño Físico



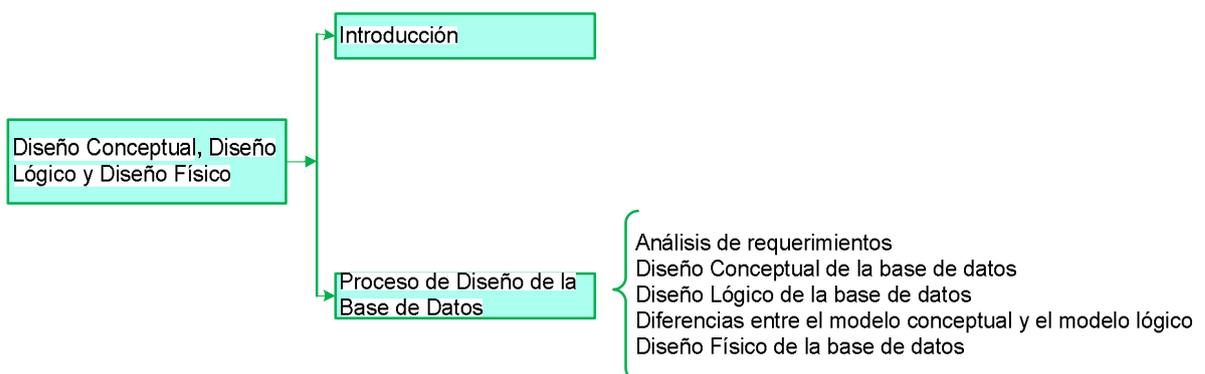
#### Visión General

En esta lección se tratan los modelos conceptual, lógico y físico que permitirán modelar una base de datos.

[Volver Inicio](#)



#### Plan de la Lección



#### Desarrollo de la Lección

##### 2.5.1. Introducción:

Dentro del proceso de desarrollo de una base de datos se encuentran tres etapas: el diseño conceptual, diseño lógico y diseño físico.

Los modelos de datos permiten que el significado de los datos sea entendible y se lo realiza para tener claros los siguientes puntos:

- El enfoque de los datos que posee cada usuario.
- El entorno de los datos no dependientes de la interpretación física.
- Los modelos de datos se usan para esquematizar el enfoque que tiene el diseñador sobre la base de datos.
- Las organizaciones actualmente especifican ciertas políticas en los procedimientos desarrollados diariamente dentro de una organización.
- Al hablar de reglas de negocio se debe pensar en una descripción concisa de una política o procedimiento de una organización.

## 2.5.2. Proceso de diseño de la base de datos

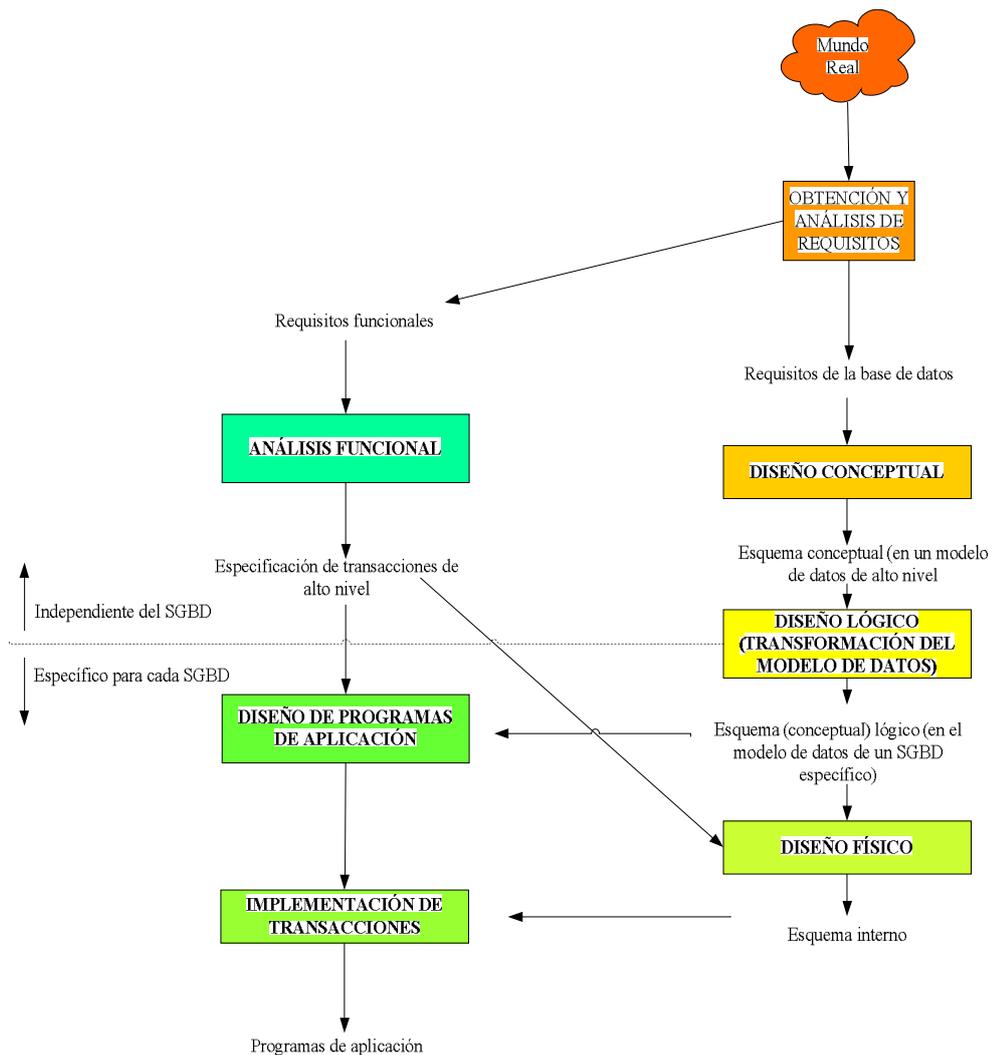


Figura 2.5.2\_1: Fases del Diseño de la Base de Datos <sup>57</sup>

La figura 2.1.1, indica el detalle del proceso de diseño de base de datos. Los pasos para este proceso son:

### 2.5.2.1. Análisis de requerimientos

El primer paso es obtener y analizar los requerimientos en el que los diseñadores dialogan con los usuarios para comprender y documentar detalladamente las necesidades.

Equivalentemente con la descripción de los requerimientos de datos se debe detallar los requerimientos funcionales de la aplicación, los cuales tienen

<sup>57</sup> Ramez, Elmasri. y Shamkant, Navathe. (2002). *Fundamentos de Sistemas de Bases de Datos*, Madrid: Pearson Addison Wesley.

transacciones que utilizarán en la base de datos y también contienen los datos y su actualización. El diseño de software usa diagramas de flujo de datos, de secuencia o escenarios para describir los requisitos funcionales.

- **Reglas de Negocio**

Las reglas del negocio son restricciones o limitaciones en las actividades del negocio y requieren manifestarse en la base de datos y además en sus aplicaciones. También se conoce a las reglas de negocio como una representación breve, oportuna de los procedimientos que ayudan a establecer y asignar acciones en una organización. Si las reglas de negocio se describen correctamente precisan entidades, atributos, relaciones, etc.

Las características principales de las reglas de negocio son:

Las reglas del negocio representan un lenguaje sencillo y entendible para ser eficientes, el enfoque de las mismas depende de la visualización de la organización.

Los principales orígenes de las reglas del negocio son los gerentes quienes realizan los procedimientos, manuales de operación, esquemas de la organización. Un principio de las reglas de negocio más directo son diálogos con los usuarios finales pero al mismo tiempo puede convertirse en menos confiable, a pesar que los usuarios finales apoyan al avance de las reglas de negocio es conveniente identificar su percepción. Cada persona tiene una percepción diferente del trabajo que realiza y de sus elementos, esto no ayuda al diseñador de la base de datos y en este caso deberá encargarse de encajar las diferencias y comprobar los resultados adecuados para asegurar que las reglas de negocio sean precisas y adecuadas.

Al tener las reglas de negocio claras le facilitará al diseñador conocer el funcionamiento del negocio y entender como los datos operan dentro de la organización. Es diseñador debe tener claro las reglas de negocio de la organización su efecto, desempeño y alcance.

“Permiten que el diseñador entienda los procesos del negocio.

Permiten que el diseñador desarrolle reglas de participación de relación apropiadas y restricciones para clave foránea.

El último punto es especialmente digno de hacerse notar, el que una relación sea obligatoria u opcional, en general, depende de la regla de negocio aplicable. Ejemplos:

“Un cliente puede hacer muchos pagos a cuenta

- Cada pago a cuenta es acreditado a sólo un cliente
- El operador de una máquina no puede trabajar más de diez horas en cualquier periodo de 24 horas.
- El destino de un viaje de negocios debe ser programado para menos de diez personas o para más de 30.
- La estructura del avión de una compañía debe ser inspeccionada cada 100 horas de vuelo.
- Un laboratorio no puede ser programado para propósitos de demostración más de una por día.
- Un cliente puede generar muchas facturas.
- Cada factura es generada sólo por un cliente.”<sup>58</sup>

#### **2.5.2.2. Diseño Conceptual de la Base de Datos**

El siguiente punto es la creación de un modelo conceptual de datos de alto nivel, todo esto conforma el **diseño conceptual**.

Este esquema es un detalle de los tipos de entidad, relaciones y restricciones. Esta perspectiva permite a los diseñadores a centralizarse en describir las propiedades de los datos y despreocuparse del almacenamiento. Se puede utilizar las operaciones básicas del modelo de datos cuando ya se diseñe el esquema conceptual, lo que sirve como comprobación de que el esquema conceptual cumpla con los requerimientos funcionales descritos.

---

<sup>58</sup> Rob, Peter y Coronel, Carlos. (2006). *Sistemas de bases de datos: diseño, implementación y administración*. España: Thomson.

Para realizar el desarrollo del modelo de datos el diseñador debe tener claro los datos de la organización aunque los éstos por si solos no causan el conocimiento que la empresa requiere lo que se logra cuando se concretan las reglas de negocio.

La creación de un diagrama conceptual implica lo siguiente:

- Identificar los conjuntos de entidades útiles para modelar el problema.
- Identificar los conjuntos de interrelaciones y determinar su grado y tipo (1:1, 1:N o M:N)
- Trazar un primer diagrama E-R
- Identificar atributos y dominios para los conjuntos de entidades y relaciones
- Seleccionar las claves principales para los conjuntos de entidades
- Verificar que el modelo resultante cumple el planteamiento del problema. Si no es así, se vuelve a repasar el proceso desde el principio.”<sup>59</sup>

En la fase del diseño conceptual se usa el modelo de datos para establecer distribuciones de la base de datos que simbolicen al mundo de la forma más real. El modelo conceptual debe ser entendible es decir representar lo que es la empresa, no depende del hardware y software.

El modelo conceptual muestra un enfoque general de los datos, este modelo es el fundamento para determinar y representar sin entrar en mayores detalles los datos, de este tipo de modelos el más conocido es el Modelo Entidad – Relación (E-R).

El objetivo de este modelo es detallar lo que contiene la información de la base de datos. Los siguientes son considerados modelos conceptuales:

- Modelo Entidad – Relación (ER)
- Modelo Semántico.

A continuación se presenta el modelo conceptual del caso práctico descrito anteriormente, en la Lección 2.4 se mostrará cómo usar la Herramienta Case Power Designer y como se obtiene cada uno de los modelos.

---

<sup>59</sup> Sabana, Maribel. (2006). *Modelamiento e Implementación de Base de Datos*: Lima: Megabyte Grupo Editorial.

### 2.5.2.3. Diseño Lógico de la Base de Datos

El paso siguiente es la implementación de la base de datos usando un SGBD el cual en la mayor parte usan el modelo de datos de implementación como el relacional o el orientado a objetos, con lo que el esquema conceptual se convierte del modelo de datos de alto nivel en el modelo de implementación; esto se conoce como **diseño lógico**.

La siguiente etapa del proceso de desarrollo de la base de datos es el diseño lógico el que tiene un modelo lógico de datos. Este diseño permite crear las relaciones.

El modelo lógico es un detalle de la organización de la base de datos, se considera a este modelo también como un lenguaje que define esquemas lógicos como el modelo relacional.

#### **Ejemplos de modelos lógicos:**

- Modelo Relacional
- Modelo Jerárquico

### 2.5.2.4. Diseño Físico de la Base de Datos

El último paso es el **diseño físico**, en el que se describe la organización de almacenamiento interno, es decir por donde acceder y la estructura de los índices de la base de datos. Equivalentemente se realiza el diseño y la ejecución de programas de aplicación como transacciones de la base de datos.

Es la etapa final del proceso de desarrollo de la base de datos, en ésta el creador o diseñador resuelve como se efectuará la implementación de la base de datos, aparece el modelo físico

El modelo físico depende tanto del hardware y del software, el diseñador debe comprender estos dos elementos para crear la base de datos. El diseño físico necesita del SGBD, en este diseño se usa el lenguaje de definición de datos.

### **2.5.3. EJERCICIO:**

#### **EMPRESA PACB CIA. LTDA (QUITO – ECUADOR)**

##### **INFORMACIÓN GENERAL DE LA EMPRESA**

PACB CIA LTDA es una empresa familiar dedicada a la compra y distribución de gaseosas y otros productos fabricados por DBCB S.A quienes proporcionan todos los productos de acuerdo a un estándar de pedidos.

Cada pedido va destinado a una determinada ruta de clientes que deben ser visitados diariamente. Para lo cual, PACB CIA LTDA ha adquirido cuatro vehículos pesados para satisfacer cada una de las rutas.

La empresa se encuentra conformada por cuatro socios, tres de los mismos son hermanos y el socio mayoritario es el padre.

SOCIO MAYORITARIO: Luis Pérez

DEMÁS SOCIOS:

- Juan Pérez
- Luis Pérez
- María Pérez

El monto de compra en producto en el año 2009 fue de 3'700 000 dólares.

De este monto la empresa recibe el 6% de comisión por parte de PACB CIA LTDA.

La empresa se ha mantenido desde hace 30 años, liderada por Luis Pérez, en el año 2000 decidió incorporar a sus tres hijos dándole a cada uno una labor en particular.

Desde entonces Juan Pérez quedó en la administración de la empresa.

Al principio solamente se distribuía el producto a zonas específicas con dos camiones propiedad de la empresa.

Luego se les dio la posibilidad de abrir tres bodegas en sitios estratégicos del Norte de Quito, dichas bodegas se mantienen hasta hoy.

##### **ADMINISTRACIÓN DE LA EMPRESA**

La administración está al mando de Juan Pérez, quien organiza la entrega de los productos adquiridos a DBCB S.A de acuerdo a los cuatro rutero elaborados por dicha empresa, en la que constan los pedidos de los

clientes (detallistas), que son a quienes se les entrega el producto, que a su vez son comercializados al consumidor final.

Además se contrata personal como: choferes profesionales, estibadores, los mismos que no están calificados y requieren de una previa capacitación para cada función que cumplen dentro de la empresa como son: el manejo de vehículos, manipulación y entrega del producto y la atención personalizada en las distintas bodegas.

## **FORMA DE TRABAJO**

La forma de trabajar de la empresa es la siguiente:

1. Toma de pedidos por teléfono o preventa (marketing directo) por parte de DBCB S.A, realizada el día anterior de la entrega.
2. Compra de la mercadería a DBCB S.A de acuerdo a los pedidos de los clientes y para las bodegas (que a su vez conforman otros clientes)
3. Carga en los distintos camiones de las compras realizadas (día anterior a la distribución)
4. Salida de los camiones a las distintas rutas para la distribución al cliente, que a su vez se convierte en el proveedor minoritario (tiendas, micro-mercados, bares, sitios de diversión, billares, restaurantes, panaderías, etc.)

## **ESTRATEGIAS, PLANES Y POLÍTICAS**

En realidad la empresa no tiene una estrategia sólida para llevar a cabo su labor, tal vez el marketing directo es la única herramienta que se la puede incluir como tal.

Además no existe plan de ventas, ni una estructura bien definida para no dejar de lado ningún detalle importante.

Los trabajadores no tienen la adecuada identidad con la empresa, se pudo ver que algunos de ellos solo trabajan por obligación “para ganar un sueldo”, esto se aduce al hecho de que la mayoría no es personal calificado.

No existe una política definida por parte del administrador actual, quien necesita de un asesoramiento de un equipo de trabajo emprendedor, con experiencia en administración de empresas y relaciones humanas – laborales, para ayudar a tomar acciones en cuanto a: control de ingresos y egresos de mercadería, entrada y salida del personal, planes y estrategias para enfrentar cambios, desarrollo de una cultura empresarial, etc.

## - Modelo Conceptual

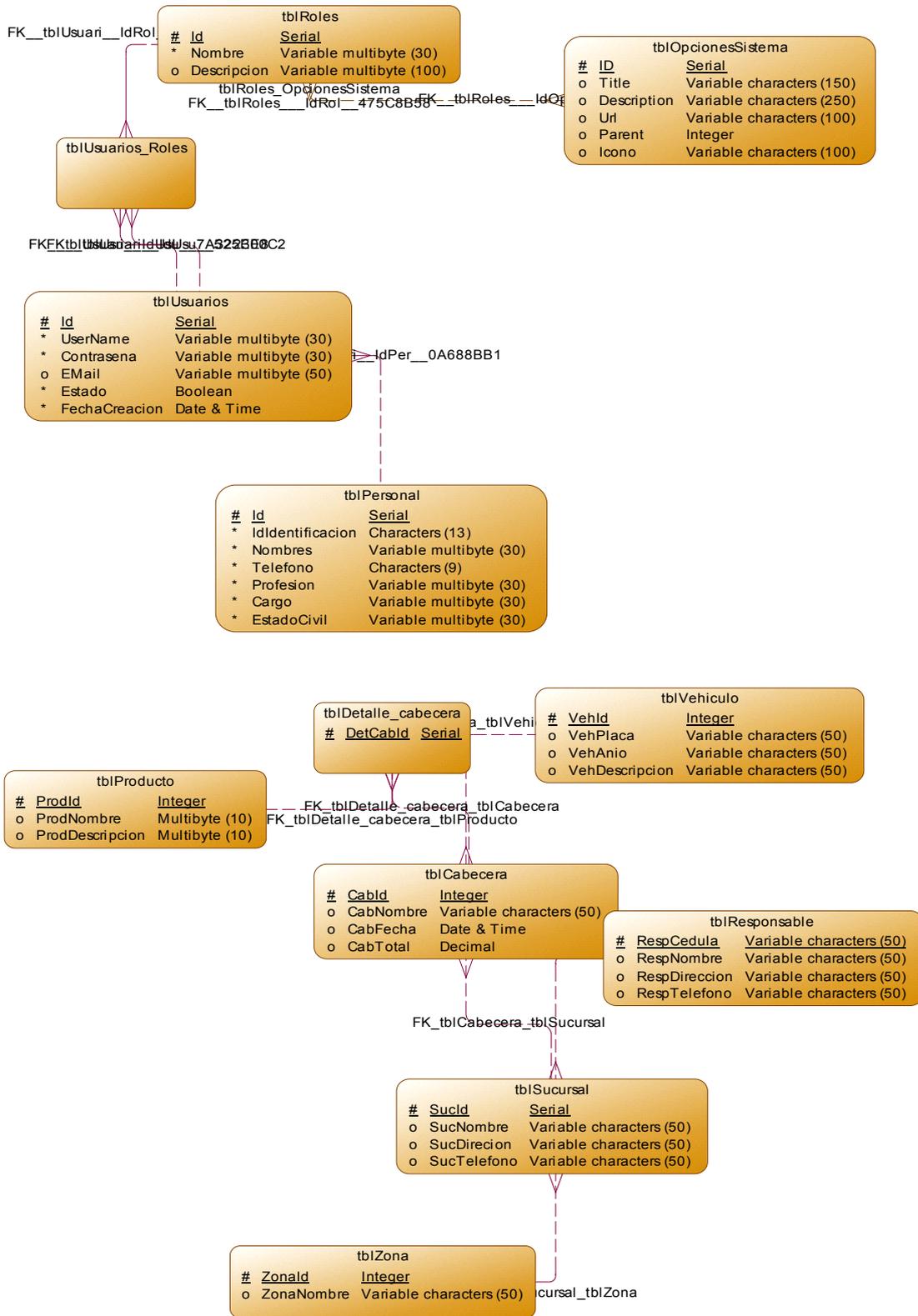


Figura 2.5.3\_1: Modelo Conceptual

- **Modelo Físico**

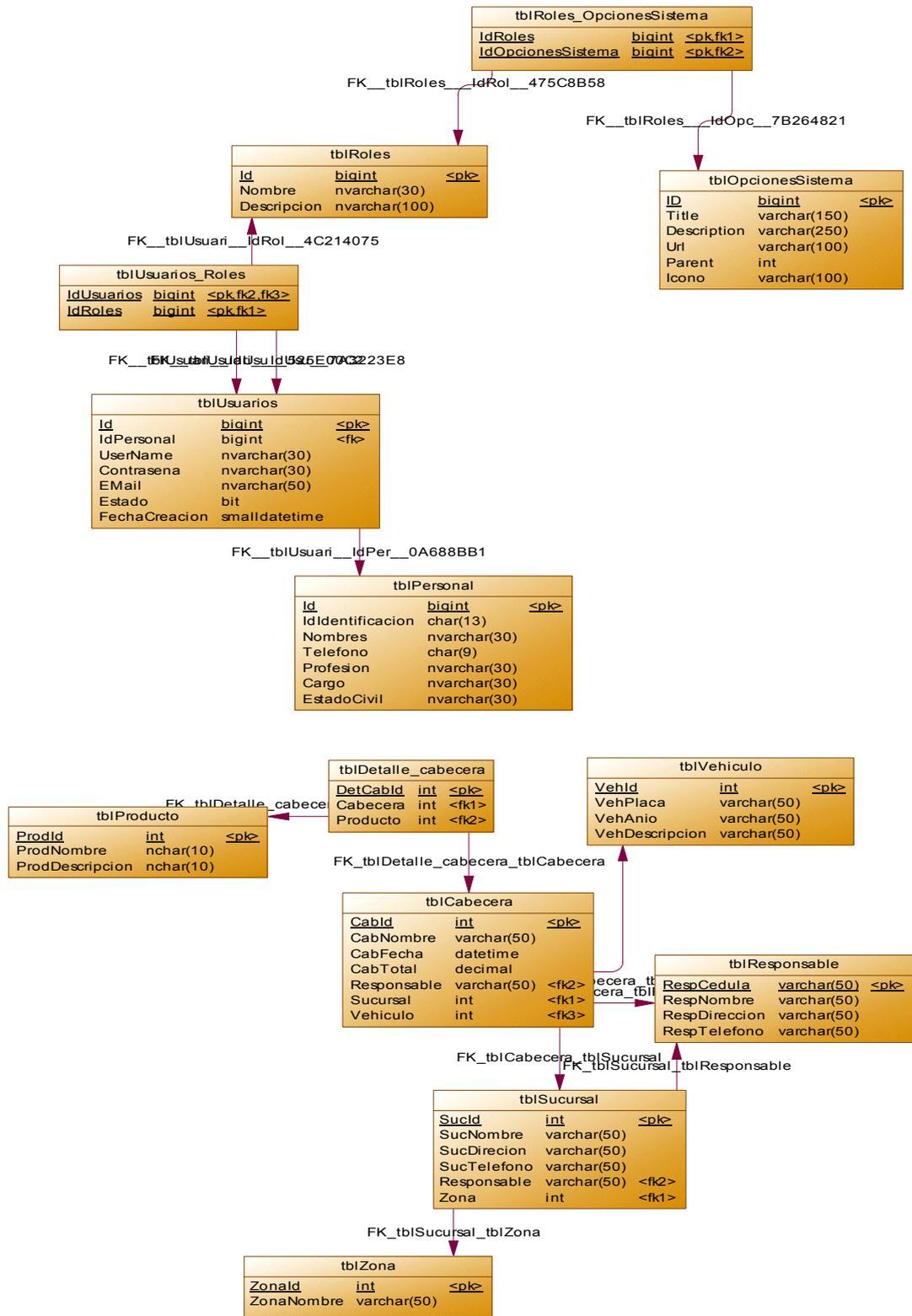


Figura 2.5.3\_2: Modelo Físico

[Volver Inicio](#)



## U2. Lección 5 tarea 1

### ACTIVIDAD DE APRENDIZAJE

---

Denominación de la tarea:

Obtener el diseño conceptual, lógico y físico del siguiente enunciado:

### **ASESORÍA DIDÁCTICA**

Para desarrollar la siguiente actividad de aprendizaje, debe leer y comprender el contenido de la Lección 5 Unidad 2.

### **ACTIVIDAD DE APRENDIZAJE**

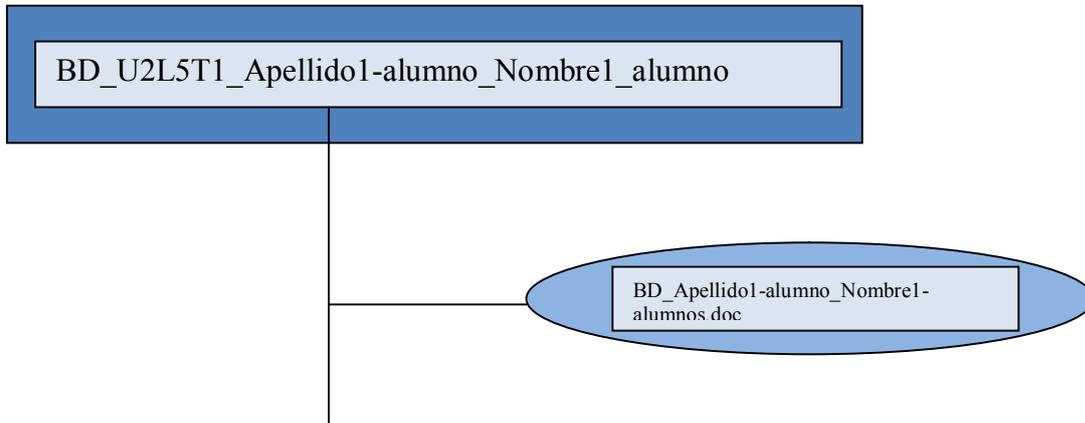
Se desea almacenar la información de una Compañía Aérea en una Base de Datos relacional. La Compañía Aérea tiene tres recursos principales: Aviones, pilotos, tripulación. De cada pila se desea conocer su código, nombre y horas de vuelo. De los miembros de la tripulación solo se tendrá el código y el nombre.

Los pilotos y tripulación tienen una base a la que regresan después de cada jornada un vuelo va desde un origen a un destino a una hora concreta y tiene número de vuelo, de cada vuelo que se va a realizar durante los próximos 2 meses, así como de los vuelos que se han realizado se desea saber el avión en el que se va a hacer o en el que se ha hecho, el piloto y la tripulación.

Cada avión tiene un código, es de un tipo (boing, airbus, entre otros). Y tiene una base donde es sometido a mantenimiento.

Envíe a su tutor la tarea en una carpeta comprimida como .rar

Formato de la carpeta:





## U2. Lección 5 tarea 2

### ACTIVIDAD DE APRENDIZAJE

---

Denominación de la tarea:

**Realice un cuadro comparativo sobre el diseño conceptual, lógico y físico de la base de datos.**

### ASESORÍA DIDÁCTICA

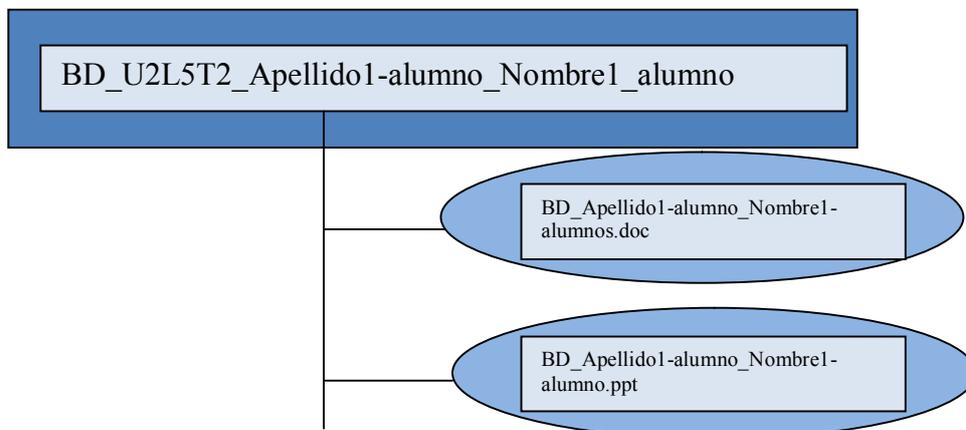
Para desarrollar la siguiente actividad de aprendizaje, debe leer y comprender el contenido de la Lección 5 Unidad 2.

### ACTIVIDAD DE APRENDIZAJE

Realice un cuadro comparativo sobre el diseño conceptual, lógico y físico de la base de datos.

Envíe a su tutor la tarea en una carpeta comprimida como .rar

Formato de la carpeta:





## Resumen Lección 5

---

### Diseño Conceptual, Diseño Lógico y Diseño Físico

Esta lección ha presentado una visión global de los Diseños Conceptual, Lógico y Físico incluyendo:

- **Proceso de diseño de la base de datos**
  - **Análisis de requerimientos**
  - **Diseño Conceptual de la Base de Datos**
  - **Diseño Lógico de la Base de Datos**
  - **Diseño Físico de la Base de Datos**

#### **Destacándose que:**

El primer paso es obtener y analizar los requerimientos en el que los diseñadores dialogan con los usuarios para comprender y documentar detalladamente las necesidades.

Las reglas del negocio son restricciones o limitaciones en las actividades del negocio y requieren manifestarse en la base de datos y además en sus aplicaciones.

El diseño conceptual es un detalle de los tipos de entidad, relaciones y restricciones.

La implementación de la base de datos usando un SGBD el cual en la mayor parte usan el modelo de datos de implementación como el relacional o el orientado a objetos, con lo que el esquema conceptual se convierte del modelo de datos de alto nivel en el modelo de implementación; esto se conoce como diseño lógico.

El diseño físico, en el que se describe la organización de almacenamiento interno, es decir por donde acceder y la estructura de los índices de la base de datos.

## UNIDAD 2: PROCESO DE DESARROLLO DE LA BASE DE DATOS

### LECCION 2.6: Aplicación de una Herramienta Case



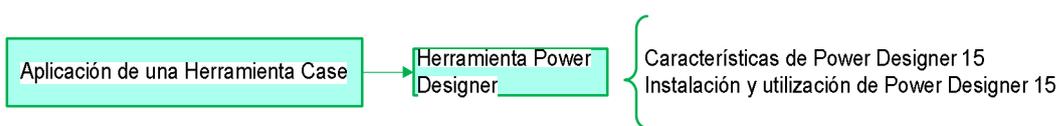
#### Visión General

En esta lección se explica la herramienta case Power Designer.

[Volver Inicio](#)



#### Plan de la Lección



#### Desarrollo de la Lección

##### 2.6.1. Herramienta Power Designer

Es una herramienta de modelamiento que posibilita representar, analizar y manejar metadatos, obteniendo una correcta arquitectura de información dentro de una organización. Además ofrece una visión que se basa en modelos, lo que permitirá organizar el negocio mediante la tecnología de información. Esta herramienta posee técnicas para analizar, diseñar y gestionar los metadatos.

Power Designer es compatible con herramientas de desarrollo como .NET, Power Builder, Java, Eclipse, lo que permite facilitar a las organizaciones en su análisis de negocios y diseño de la base de datos, esta herramienta también se usa con muchas bases de datos relacionales.

## **2.6.2. Power Designer Versión 15**

Es la versión más reciente de Power Designer propiedad de la Compañía Sybase Inc, permite realizar el diseño de modelos de datos y para las organizaciones que requieren construir procedimientos activos de negocio de manejar muy coherente y rápida.

### **2.6.2.1. Características**

Esta última versión de Power Designer desarrolla la capacidad de estudiar el impacto y presentar una mejor percepción a los usuarios.

Mejor organización de metadatos y relación de procedimientos del negocio.

Disminuye la redundancia de los sistemas.

Es una herramienta consistente para la administración de modelos de datos y metadatos que se usan para organizar los objetivos de una empresa.

A continuación se presenta la instalación de una Herramienta Case que se la debe realizar presencialmente como parte del curso B-Learning:

### **2.6.2.2. Instalación y utilización de Power Designer 15.0**

- **Instalación**

1. Pantalla inicial para la instalación de Power Designer

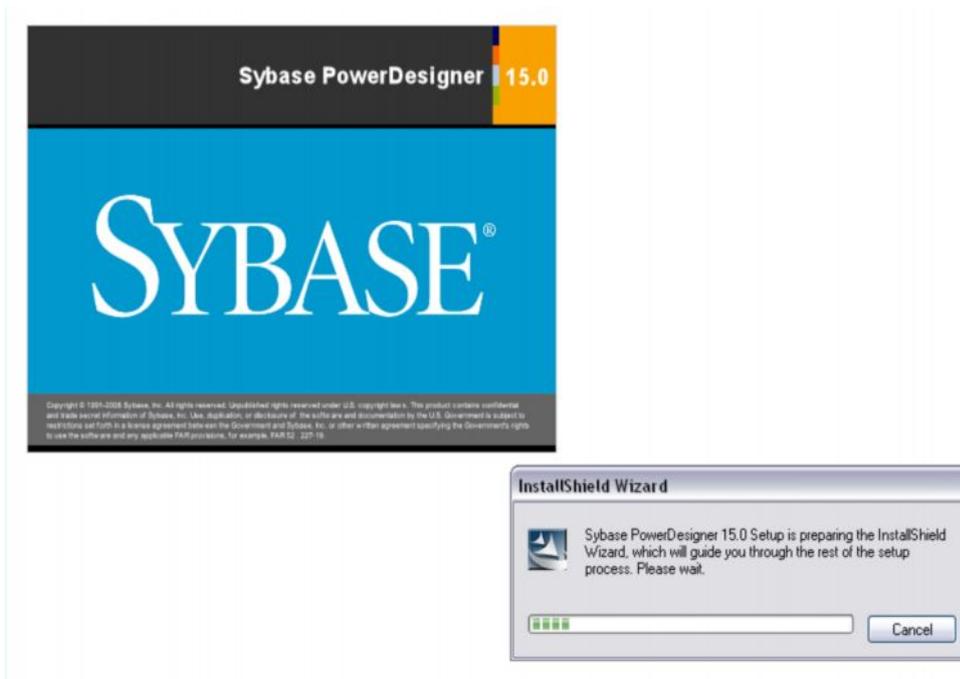


Figura 2.6.2.2\_1: Pantalla inicial de la instalación de Power Designer

## 2. Pantalla de bienvenida

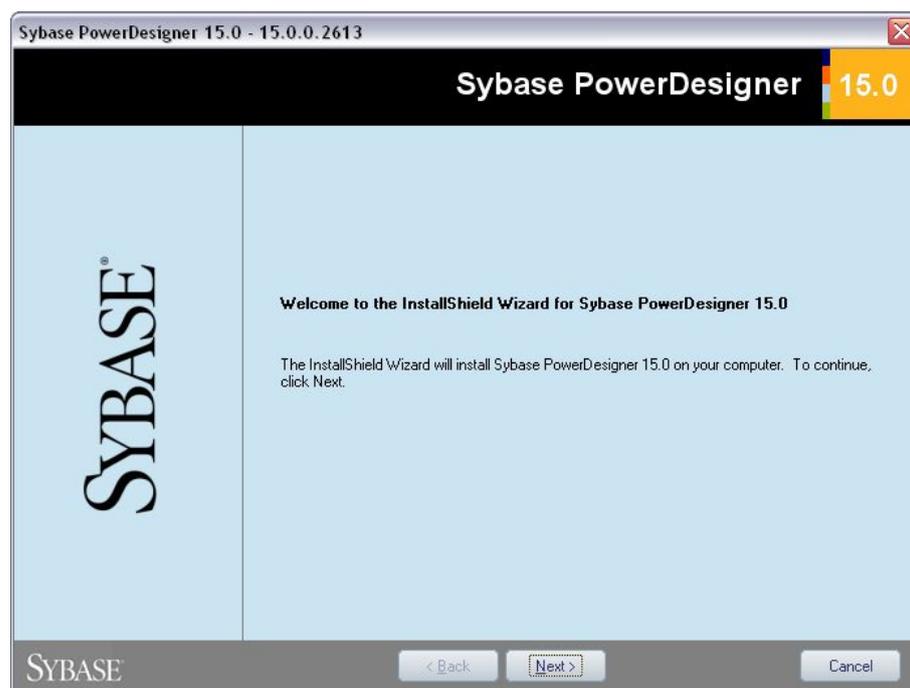


Figura 2.6.2.2\_2: Pantalla de Bienvenida a la instalación de Power Designer

3. En esta pantalla se muestra el contrato de licenciamiento para aceptarlo o no.

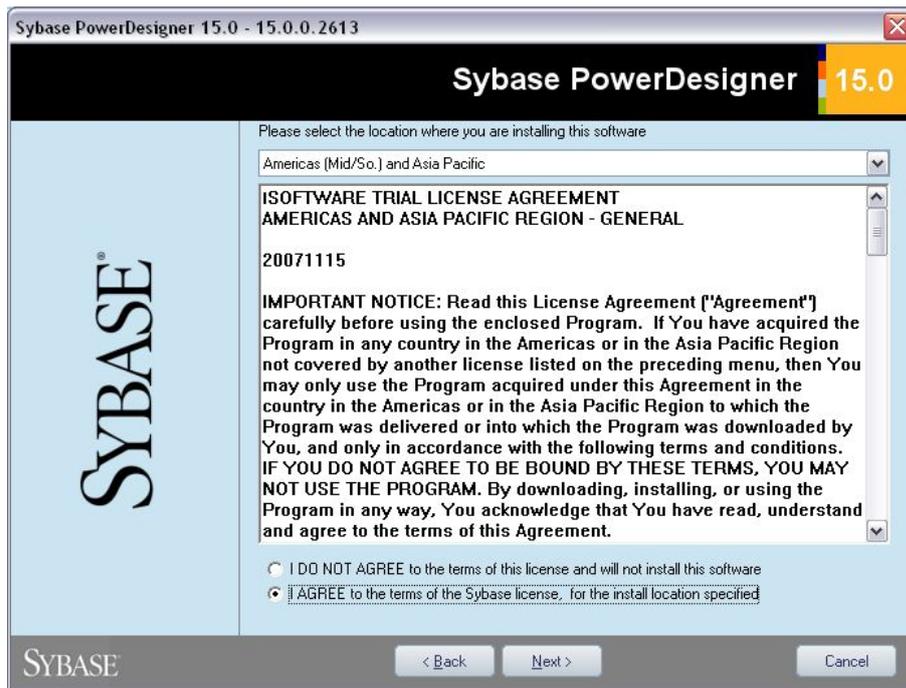


Figura 2.6.2.2\_2: Pantalla de contrato de licencia

La siguiente pantalla muestra el directorio donde se instalará la herramienta

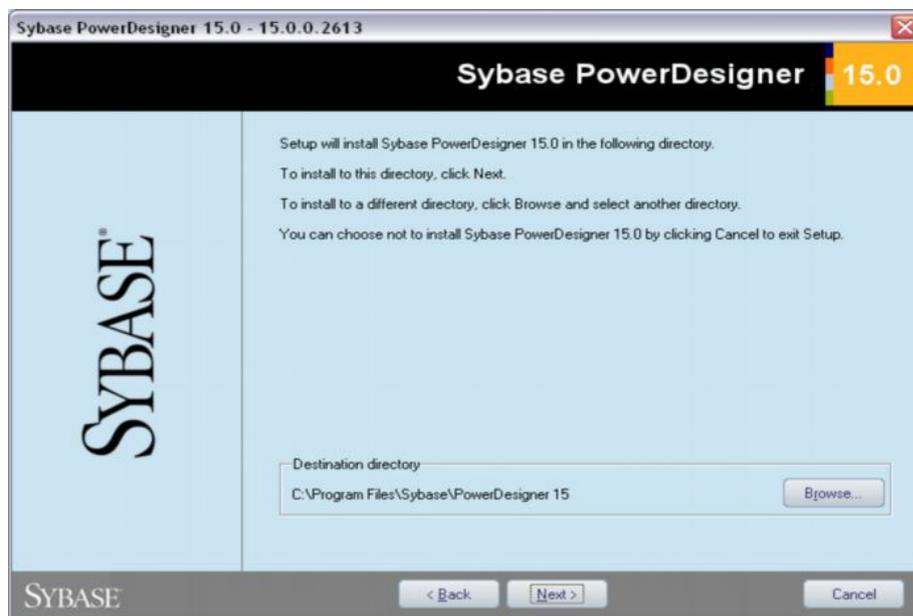


Figura 2.6.2.2\_3: Pantalla donde se indica el directorio donde se instalará Power Designer

4. En la pantalla que se muestra a continuación se escoge las opciones a instalar

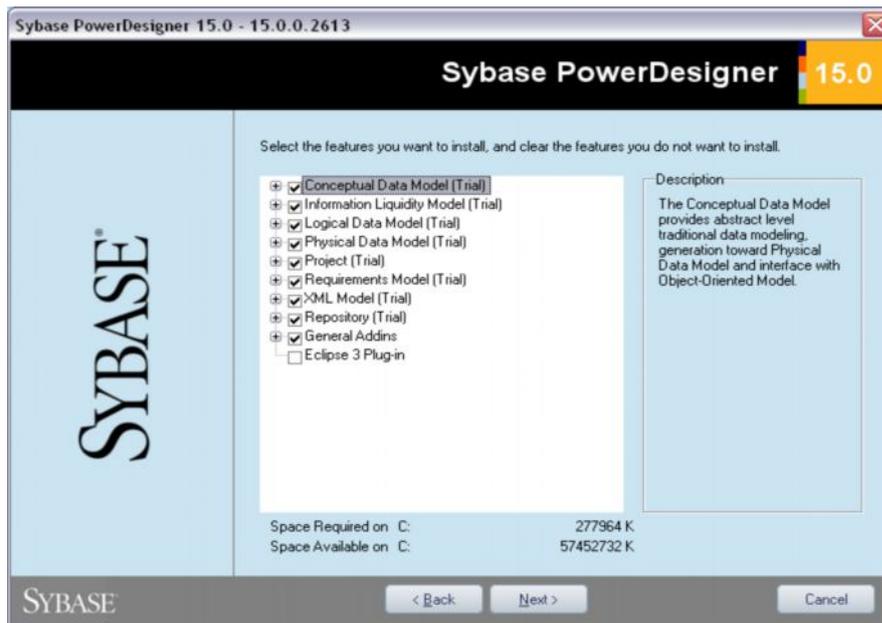


Figura 2.6.2.2\_4: Pantalla donde se escoge lo que se instalará

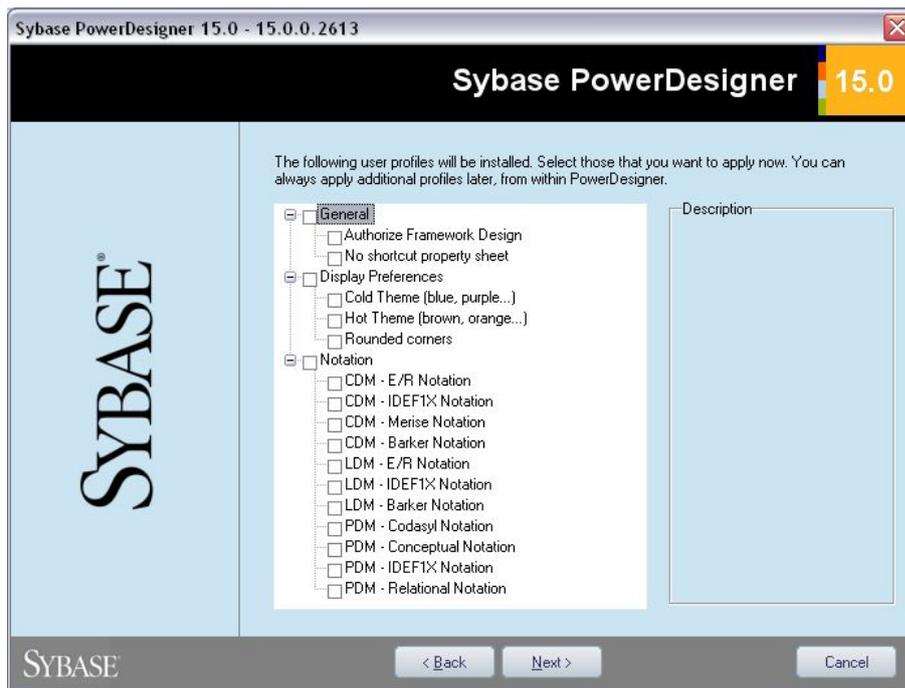


Figura 2.6.2.2\_5: Pantalla donde aparece lo que se instalará

5. La siguiente pantalla muestra la carpeta donde se instalará Power Designer

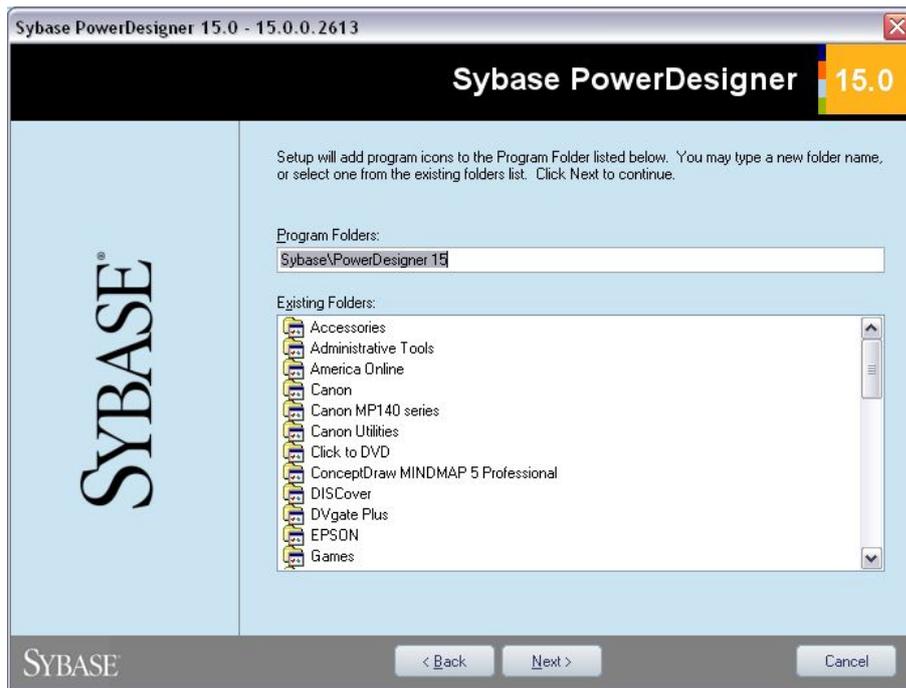


Figura 2.6.2.2\_6: Pantalla donde muestra el directorio donde se instalará la aplicación



Figura 2.6.2.2\_7: Pantalla donde se indica el estado de licencia

Esta pantalla muestra la instalación del Power Designer

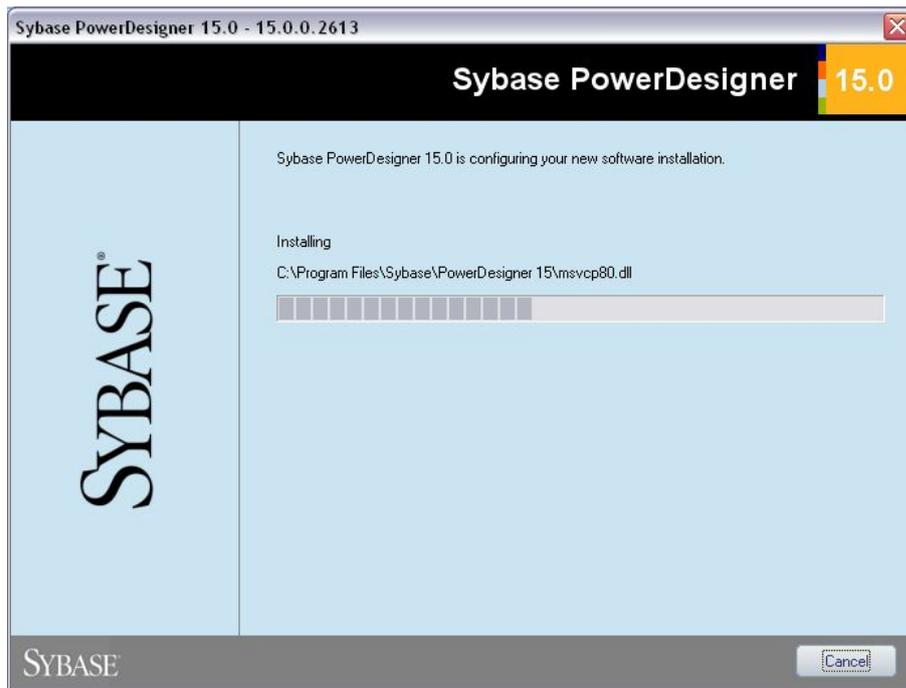


Figura 2.6.2.2\_8: Pantalla de Instalación de Power Designer

## 6. Finalización de la instalación

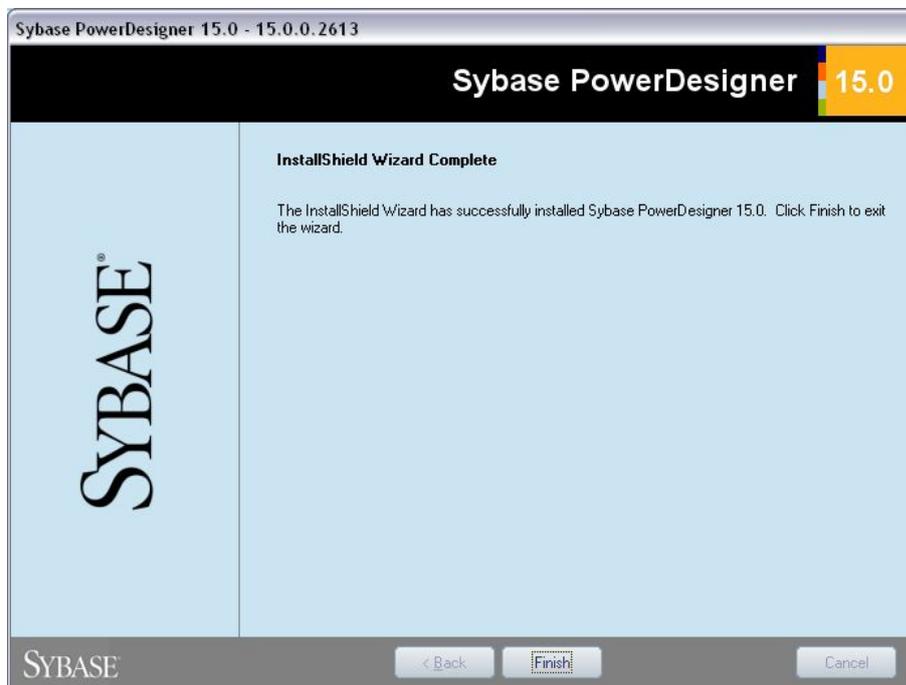


Figura 2.6.2.2\_9: Pantalla de Finalización de Instalación de Power Designer

- **Utilización**

1. Se ingresa a Power Designer : Aparece la siguiente pantalla para crear un modelo conceptual

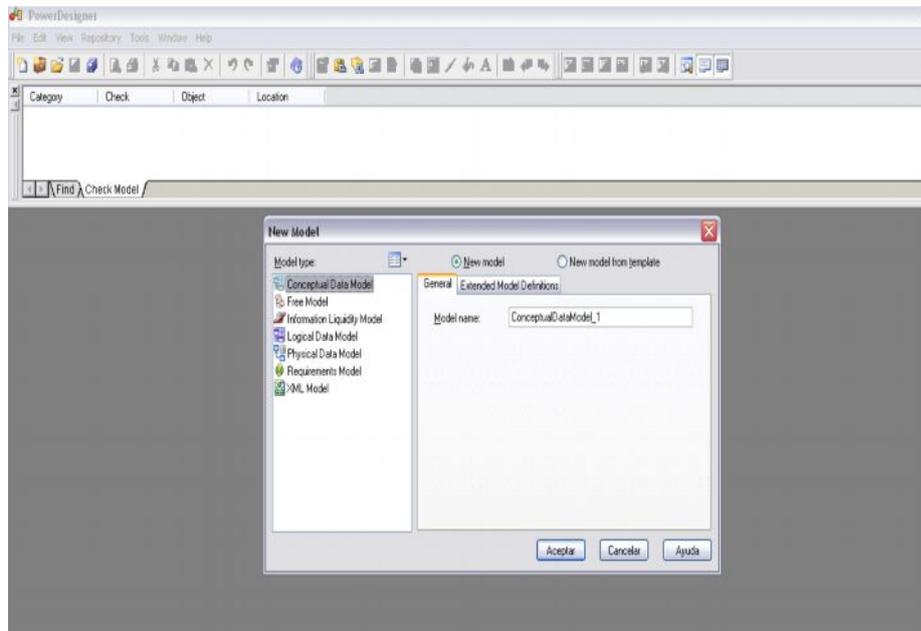


Figura 2.6.2.2\_10: Pantalla de Power Designer

2. Una vez creado el modelo conceptual aparece la siguiente pantalla:

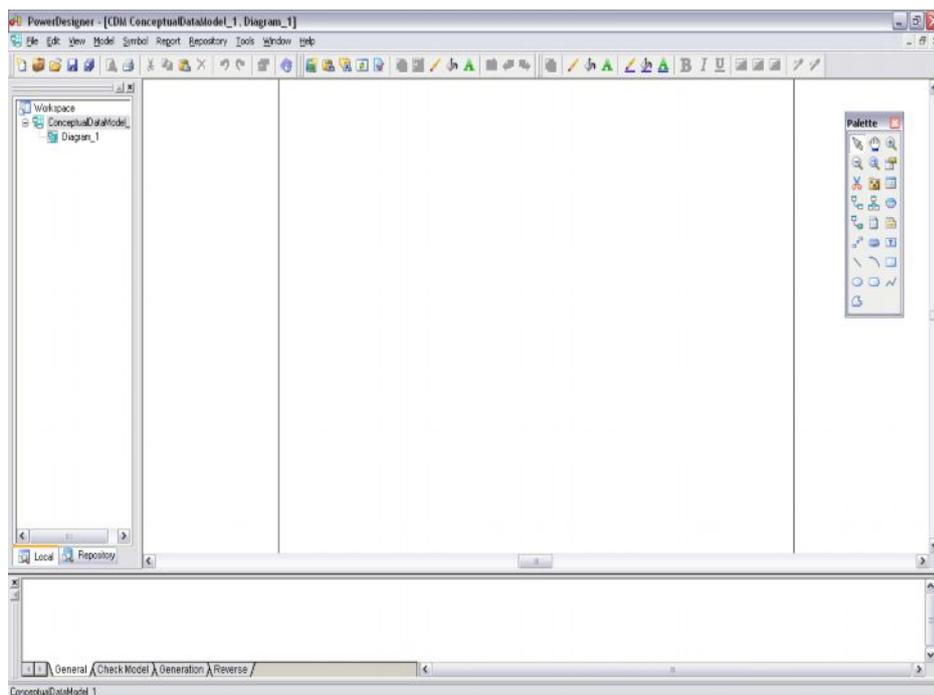


Figura 2.6.2.2\_11: Pantalla inicial de Power Designer

3. La paleta que aparece a continuación es la que nos permite diagramar las entidades, atributos, relaciones, etc.

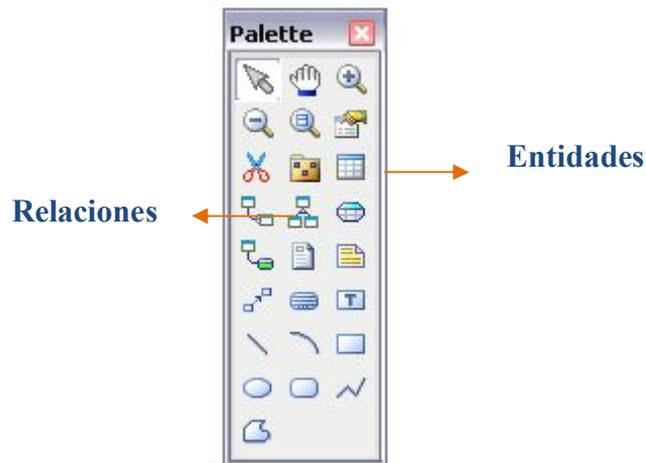


Figura 2.4.2.2\_12: Paleta

#### Crear entidades:

4. Seleccionar botón  de la paleta mostrada anteriormente
- a. Clic en hoja de trabajo para insertar la entidad:

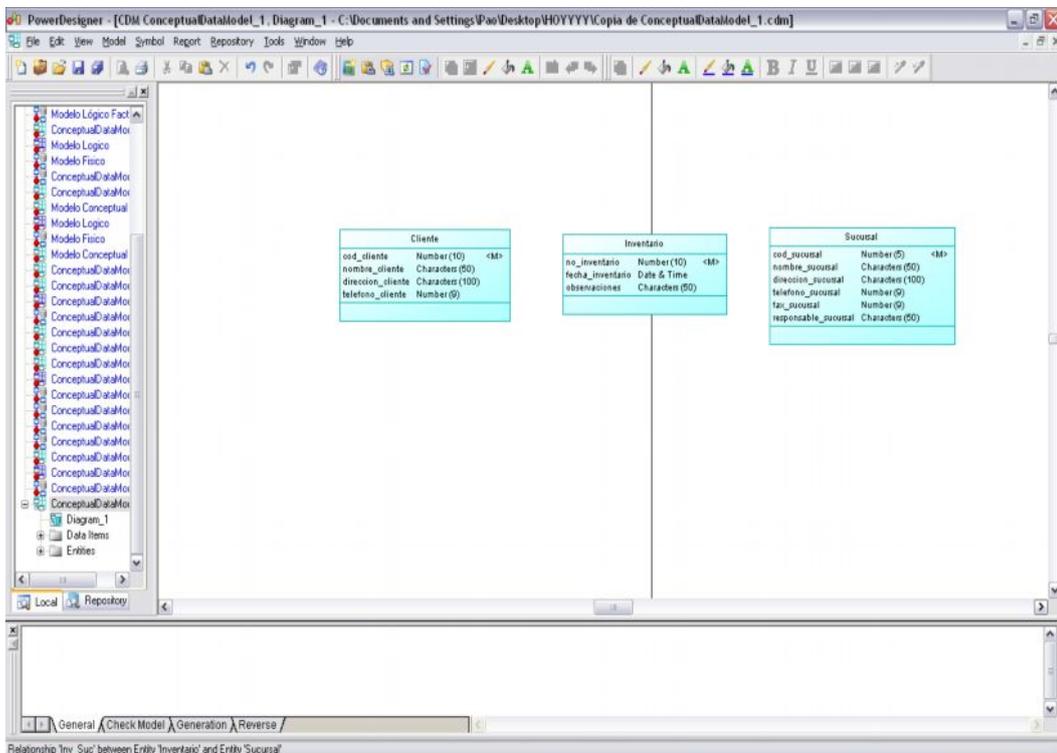


Figura 2.6.2.2\_13: Pantalla Creación de Entidades

5. Para general el modelo lógico, se debe seguir los siguientes pasos:

Herramientas

→ Generar Modelo Lógico

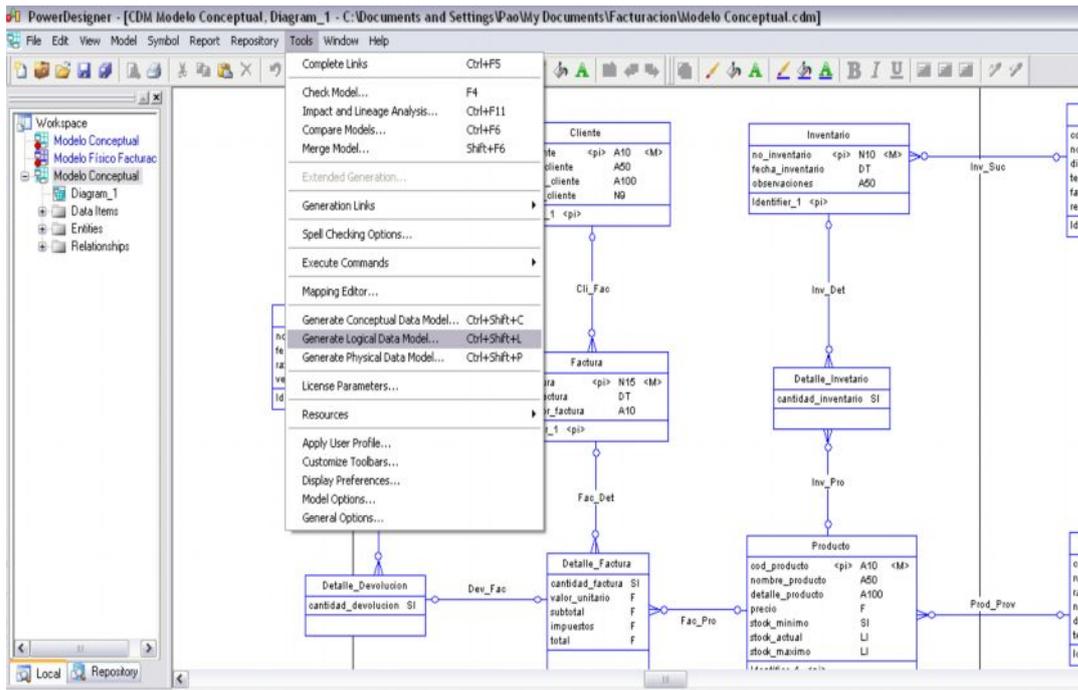


Figura 2.6.2.2\_14: Pantalla Generación de Modelo Lógico

6. Luego aparece la siguiente pantalla, en la pestaña General se coloca el nombre del modelo.

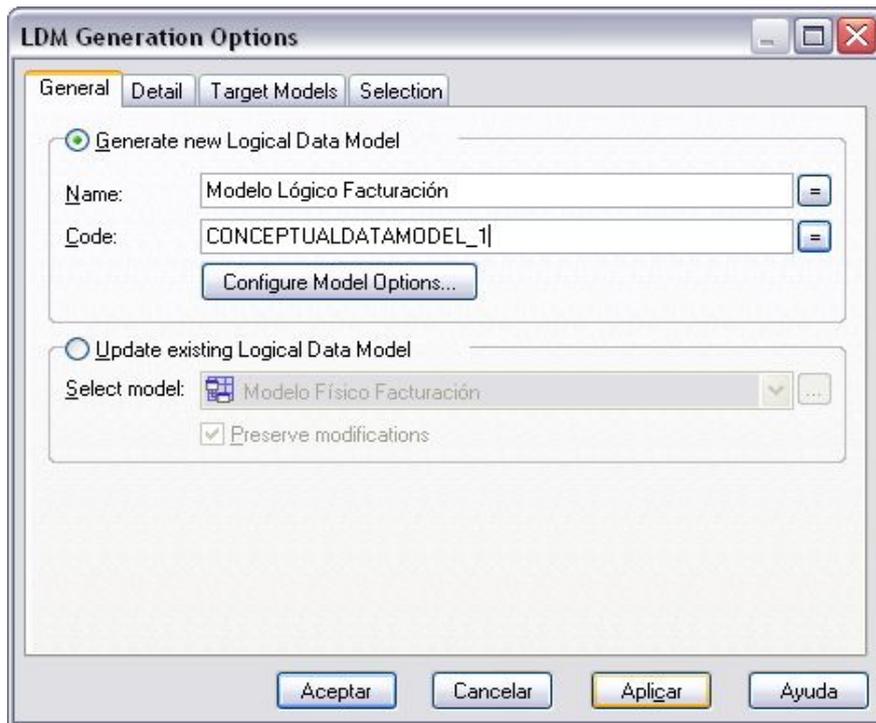


Figura 2.6.2.2\_15: Pantalla Generación de Modelo Lógico

7. Para generar el modelo físico:

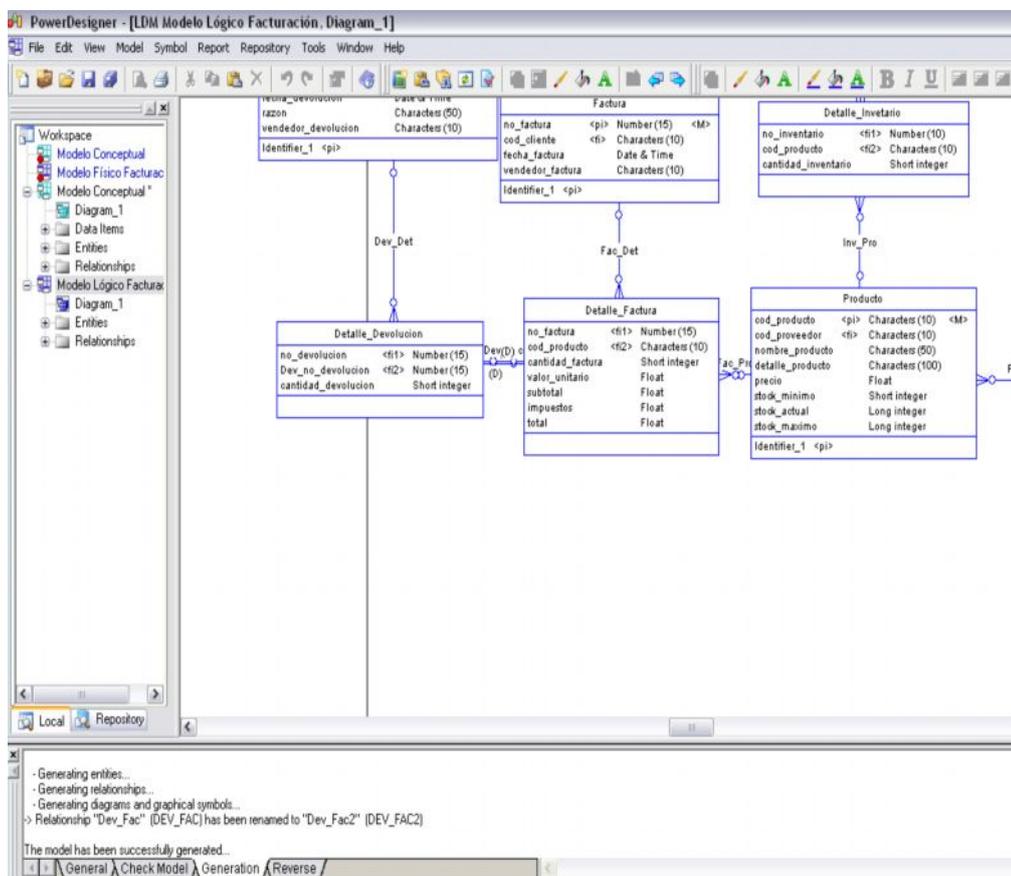


Figura 2.6.2.2\_16: Pantalla Modelo Lógico

8. Para generar el modelo físico se debe seguir los siguientes pasos:

Herramientas

➔ Generar Modelo Físico

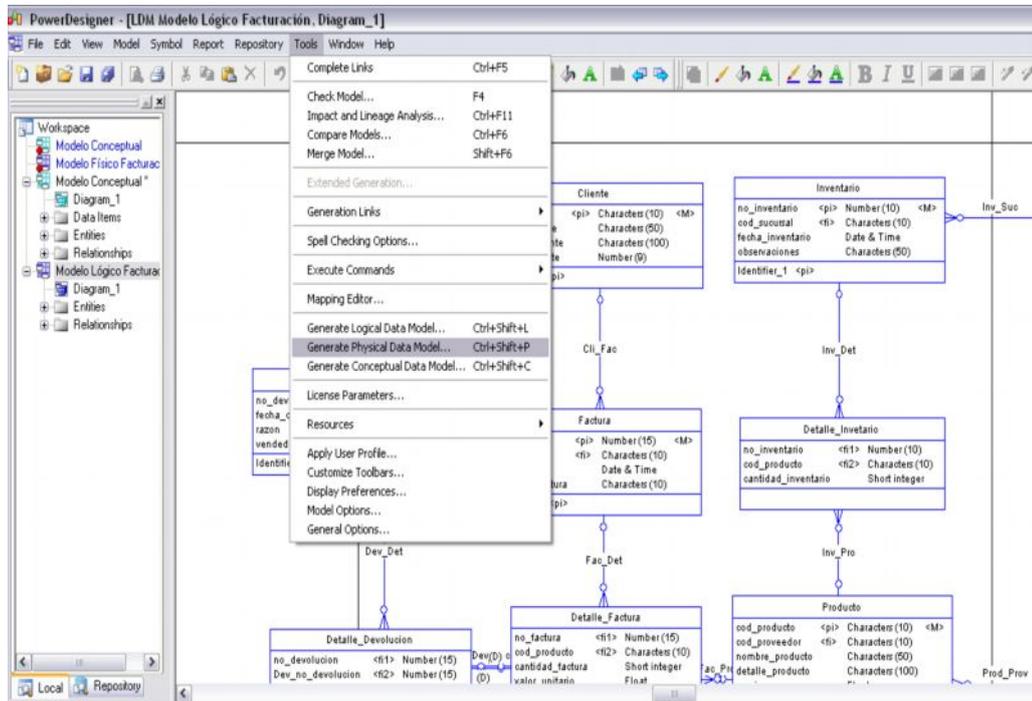


Figura 2.6.2.2\_17: Pantalla de Generación de Modelo Físico

9. Aparece la siguiente pantalla en la pestaña General escoge el SGBD que se va a usar para la aplicación en nuestro caso escogemos Microsoft SQL Server 2005 y colocamos el nombre del modelo.



11. Para generar el script se debe ir a las siguientes opciones:

Base de Datos

Generar Base de Datos

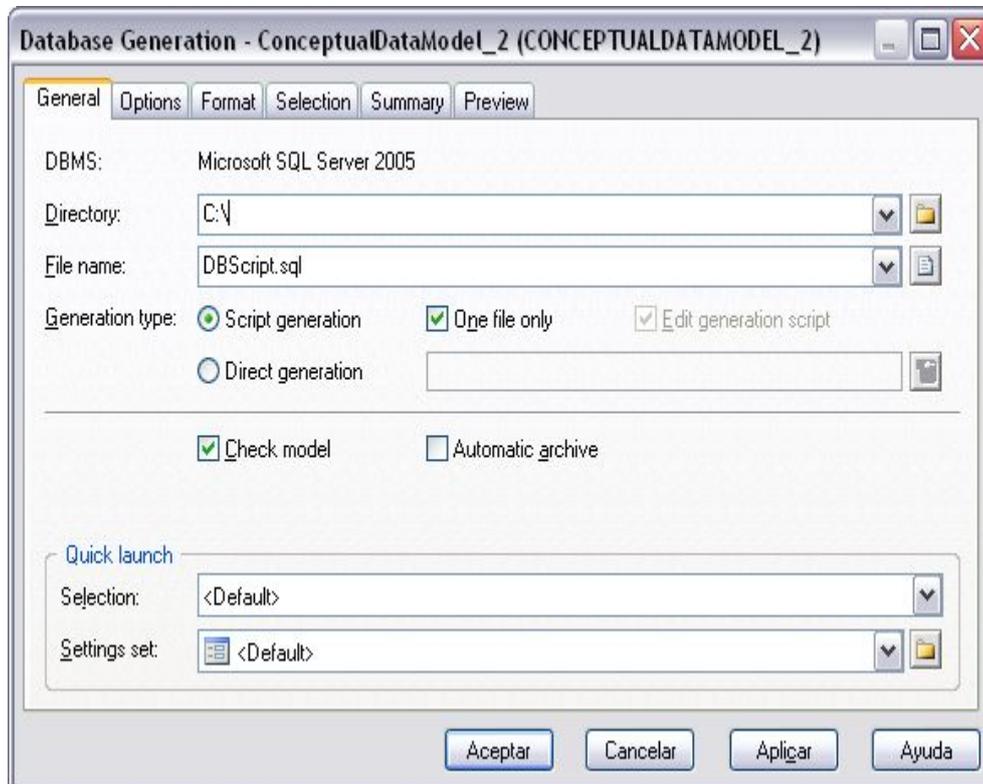


Figura 2.6.2.2\_20: Pantalla Generación de Script

[Volver Inicio](#)



## U2. Lección 6 tarea 1

### ACTIVIDAD DE APRENDIZAJE

---

Denominación de la tarea:

Instalación de una Herramienta Case

### ASESORÍA DIDÁCTICA

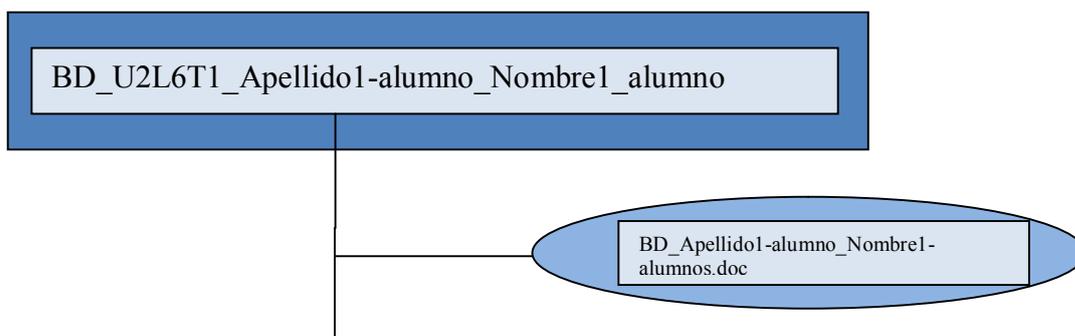
Para desarrollar la siguiente actividad de aprendizaje, debe leer y comprender el contenido de la Lección 6 Unidad 2.

### ACTIVIDAD DE APRENDIZAJE

Instalar una Herramienta Case, capturar pantallas y realizar una breve explicación

Envíe a su tutor la tarea en una carpeta comprimida como .rar

Formato de la carpeta:





## Resumen Lección 6

---

### Aplicación de una Herramienta Case

Esta lección ha presentado una visión global la aplicación de una Herramienta Case (Power Designer), incluyendo:

- **Introducción a la herramienta Power Designer**
- **Power Designer Version 15**
  - **Características**
  - **Instalación**

#### **Destacándose que:**

Es una herramienta de modelamiento que posibilita representar, analizar y manejar metadatos, obteniendo una correcta arquitectura de información dentro de una organización

Las principales características de esta herramienta son:

- Mejor organización de metadatos y relación de procedimientos del negocio.
- Disminuye la redundancia de los sistemas.
- Es una herramienta consistente para la administración de modelos de datos y metadatos que se usan para organizar los objetivos de una empresa.

## UNIDAD 2: PROCESO DE DESARROLLO DE LA BASE DE DATOS

### LECCIÓN 2.7: Normalización



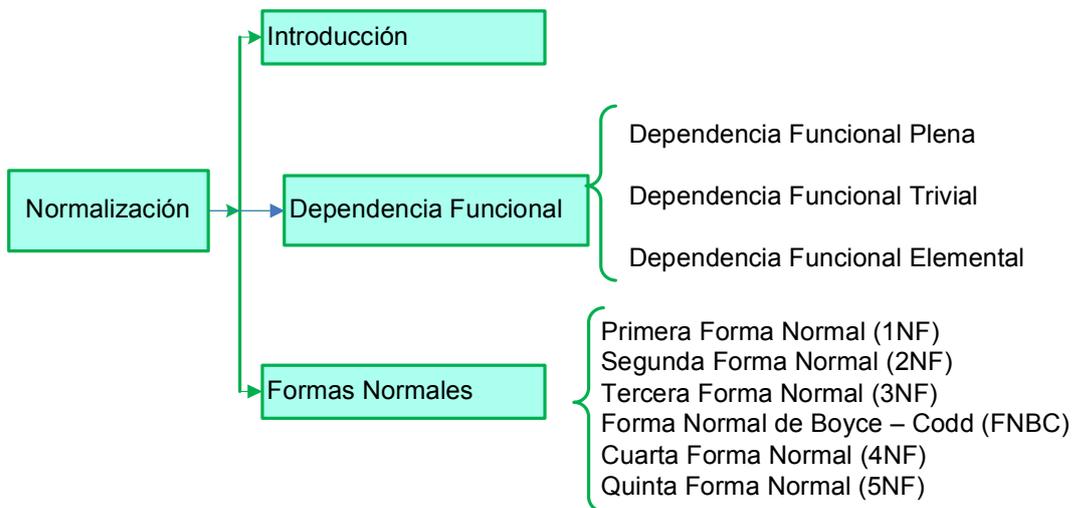
#### Visión General

En esta lección se trata la normalización para una base de datos donde se aplican reglas a las relaciones que se logran del paso del modelo entidad – relación al modelo relacional.

[Volver Inicio](#)



#### Plan de la Lección





## Desarrollo de la Lección

### 2.7.1. Introducción

La normalización comprueba que se cumplan las reglas que aseguran eliminar las inseguridades que existen en el diseño de la base de datos, esto se lo realiza mediante la ejecución de reglas denominadas Formas Normales que a veces son causantes de separar los datos en distintas relaciones. El resultado de la normalización debe ser mantener los atributos y tuplas y además las dependencias.

### 2.7.2. Dependencia Funcional

Es un enlace entre uno o más atributos.

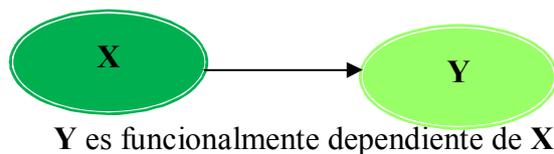


Figura 2.7.2\_1: Dependencia Funcional

Por ejemplo si se conoce el valor de *FechaDeCreaciónDeCarrera* se puede saber los *AñosDeExistencia*.

Como se observa en el gráfico la dependencia funcional del sistema se describe usando una flecha.

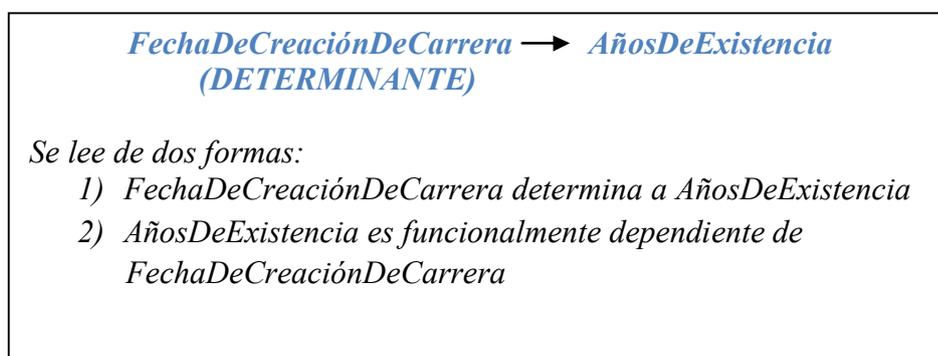


Figura 2.7.2\_2: Ejemplo de Dependencia Funcional

### 2.7.2.1. Dependencia Funcional Plena

“Se dice que Y tiene dependencia funcional completa o plena de X, si depende funcionalmente de X, pero no depende de ningún subconjunto de éste. Se representa por  $X \Rightarrow Y$

Por lo tanto,

$$X \Rightarrow Y \text{ sii } \neg \exists X' \subset X \mid X' \rightarrow Y$$

Un ejemplo de esta funcionalidad es:

MATRICULA(Cod\_Curso, Cod\_Alumno, Cod\_Materia, Nota)

Donde la Nota que obtiene un Alumno en cierta materia (un curso puede tener varias materias), si un estudiante puede matricularse en varias materias de un curso y además en un curso se matriculan varios alumnos se da la siguiente dependencia funcional:

$\text{Cod\_Curso, Cod\_Materia, Cod\_Alumno} \rightarrow \text{Nota}$

Pero:

$$\nexists X' (\text{Cod\_Curso, Cod\_Materia, Cod\_Alumno}) \mid X' \rightarrow \text{Nota}$$

Donde se puede ver que Nota depende funcionalmente de manera completa de Cod\_Curso, Cod\_Materia, Cod\_Alumno:

$\text{Cod\_Curso, Cod\_Materia, Cod\_Alumno} \Rightarrow \text{Nota}$ <sup>60</sup>

### 2.7.2.2. Dependencia funcional trivial

“Una dependencia funcional  $X \rightarrow Y$  se dice que es trivial si Y es un subconjunto de X ( $Y \subseteq X$ ). Por ejemplo son triviales las siguientes dependencias:

---

<sup>60</sup> De Miguel, Adoración; Piattini, Mario. (1999). *Diseño de Base de Datos Relacionales*, México: Alfaomega S.A.

Cód\_Alumno → Cod\_Alumno

Cod\_Curso, Cod\_Materia → Cod\_Curso”

### 2.7.2.3. Dependencia funcional elemental

“Decimos que una dependencia funcional  $X \rightarrow Y$  es elemental si  $Y$  es un atributo único no incluido en  $X$  y no existe  $X'$  incluido en  $X$  tal que  $X' \rightarrow Y$ , es decir, la dependencia funcional elemental es una dependencia funcional completa, no trivial y en la que el implicado es un atributo único:

$X \rightarrow Y$  es elemental sii  $(\neg \exists Y' \subset Y) \wedge (Y \subseteq X) \wedge (\neg \exists X' \subset X \square X' \rightarrow Y)$

No todas las dependencias funcionales son útiles para la teoría de la normalización, sino solamente las elementales, que acabamos de definir, y que son un subconjunto de las dependencias funcionales.”<sup>61</sup>

### 2.7.3. La necesidad de normalizar

Se conoce a la normalización como el procedimiento en el que se utilizan normas o reglas (cada una de estas se basa en un antecedente) sobre las relaciones que se derivan al pasar del modelo entidad – relación al modelo relacional.

Una base de datos relacional se normalizan con la finalidad de que no exista redundancia de datos ni inconvenientes al actualizar los datos de las tablas y además para proteger la integridad de los datos.

El procedimiento de la normalización se lo realiza mediante las formas normales las cuales se cumplen cuando una relación satisface una colección de restricciones.

---

<sup>61</sup> De Miguel, Adoración; Piattini, Mario. (1999). *Diseño de Base de Datos Relacionales*, México: Alfaomega S.A.

### 2.7.3.1. Pasos para normalizar

- “Descomponer todos los grupos de datos en registros bidimensionales.
- Eliminar todas las relaciones en la que los datos no dependan completamente de la clave primaria del registro.
- Eliminar todas las relaciones que contengan dependencias transitivas”.<sup>62</sup>

### 2.7.4. Formas Normales

Son procedimientos para evitar incoherencias en las tablas la cual puede estar en cualquier forma normal según la estructura que tenga.

Las formas normales fueron creadas por Codd en 1970 y continuó el autor Boyce.

Se tiene los siguientes datos que constan en una factura:

Número\_Factura

Fecha\_Factura

Código\_Cliente

Nombre\_Cliente

Dirección\_Cliente

Teléfono\_Cliente

Descripción\_Item

Cantidad\_Item

Precio\_Unitario\_Item

Valor\_Total\_Item

Subtotal\_Factura

Descuento\_Cliente (Por cliente frecuente)

Bono\_Descuento (Por temporada)

Monto\_Descuento (Por promoción y saldos)

Porcentaje\_Descuento (Sobre el valor total de la compra)

---

<sup>62</sup> Sabana, Maribel. (2006). *Modelamiento e Implementación de Base de Datos*: Lima: Megabyte Grupo Editorial.

IVA\_Factura  
Total\_Factura  
Observaciones\_Factura

#### 2.7.4.1. Primera Forma Normal (1FN)

Una estructura de datos este en 1FN si y solo si todos los dominios de todos sus atributos son indivisibles (ley de la atomicidad); entonces la estructura quedaría como sigue:

Tablas 2.7.4.1\_1-2: Primera Forma Normal

<b>FACTURA</b>	<b>ITEM</b>
<b>Número_Factura</b>	<b>Número_Factura</b>
Código_Cliente	<b>Número_Item</b>
Fecha_Factura	Cantidad_Item
Nombre_Cliente	Precio_Unitario_Item
Dirección_Cliente	Valor_Total_Item
Teléfono_Cliente	Monto_Descuento
Subtotal_Factura	
Descuento_Cliente	
Bono_Descuento	
Porcentaje_Descuento	
IVA_Factura	
Total_Factura	
Observaciones_Factura	

### 2.7.4.2. Segunda Forma Normal (2NF)

Una estructura de datos está en 2FN si y sólo si está en 1FN y además, todos los atributos que no son clave primaria y no forma parte de una clave primaria dependen completamente de una clave primaria; consecuentemente, la estructura queda así:

Tablas 2.7.4.2\_1-2-3: Segunda Forma Normal

<b>FACTURA</b>
<b>Número_Factura</b>
<b>Código_Cliente</b>
Fecha_Factura
Nombre_Cliente
Dirección_Cliente
Teléfono_Cliente
Subtotal_Factura
Descuento_Cliente
Bono_Descuento
Porcentaje_Descuento
IVA_Factura
Total_Factura
Observaciones_Factura

<b>ITEM</b>
<b>Número_Factura</b>
<b>Código_Producto</b>
<b>Número_Item</b>
Cantidad_Item
Valor_Total_Item
Monto_Descuento

<b>PRODUCTO</b>
<b>Código_Producto</b>
Descripción_Producto
Precio_Unitario_Producto

### 2.7.4.3. Tercera Forma Normal (3NF)

Una estructura de datos está en 3FN si y solo si, está en 2FN y además no existen dependencias transitivas entre los atributos que no son clave o forma parte de una clave primaria; por lo tanto la estructura queda así:

Tablas 2.7.4.3\_1-2-3-4: Tercera Forma Normal

<b>ITEM</b>
<b>Número_Factura</b>
<b>Código_Producto</b>
<b>Número_Item</b>
Cantidad_Item
Valor_Total_Item
Monto_Descuento

<b>PRODUCTO</b>
<b>Código_Producto</b>
Descripción_Producto
Precio_Unitario_Producto

<b>FACTURA</b>
<b>Número_Factura</b>
<b>Código_Cliente</b>
Fecha_Factura
Subtotal_Factura
Bono_Descuento
Porcentaje_Descuento
IVA_Factura
Total_Factura
Observaciones_Factura

<b>CLIENTE</b>
<b>Código_Cliente</b>
Nombre_Cliente
Dirección_Cliente
Teléfono_Cliente
Descuento_Cliente

#### 2.7.4.4. Forma Normal de Boyce – Codd (FNBC)

Una estructura de datos está en FNBC si y sólo si está en 3FN y además no existen varias claves candidatas compuestas que comparten algún atributo.

Tablas 2.7.4.4\_1-2-3-4: FNBC

ITEM	PRODUCTO
<b>Número_Factura</b>	<b>Código_Producto</b>
<b>Código_Producto</b>	Descripción_Producto
<b>Número_Item</b>	Precio_Unitario_Producto
Cantidad_Item	
Valor_Total_Item	
Monto_Descuento	

FACTURA	CLIENTE
<b>Número_Factura</b>	<b>Código_Cliente</b>
<b>Código_Cliente</b>	Número_Cliente
Fecha_Factura	Dirección_Cliente
Subtotal_Factura	Teléfono_Cliente
Bono_Descuento	Descuento_Cliente
Porcentaje_Descuento	
IVA_Factura	
Total_Factura	
Observaciones_Factura	

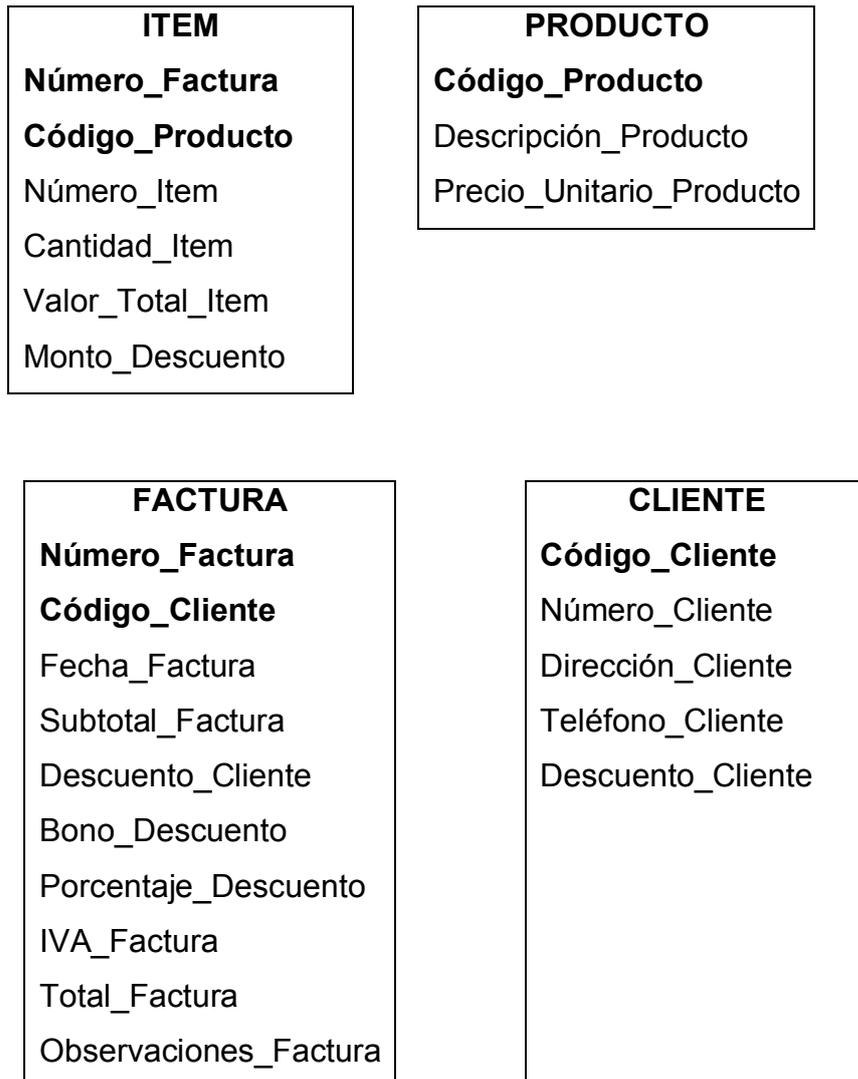
“Posibilidad de eliminar **Número\_Item** de la superclave (**Número\_Factura, Código\_Producto, Número\_Item**) en el caso que se decide hacer coincidir al **Número\_Item** con el **Código\_Producto**. O **Número\_Item** ya no es clave candidata y solo sería un atributo ordinal para el reporte”<sup>63</sup>

<sup>63</sup> Ing. Díaz Oswaldo (2012), Escuela Politécnica del Ejército

#### 2.7.4.5. Cuarta Forma Normal (4NF)

Una estructura de datos está en 4FN si y solo si, está en 3FN o en FNBC y además no existe redundancia en las relaciones múltiples (n-m); entonces la estructura quedaría como sigue:

Tablas 2.7.4.5\_1-2-3-4: Cuarta Forma Normal



“No se elimina Número\_Item para eliminar la redundancia (con el código del producto) y se lo conserva como un atributo ordinal para el reporte”<sup>64</sup>

<sup>64</sup> Ing. Díaz Oswaldo (2012), Escuela Politécnica del Ejército

### 2.7.4.6. Quinta Forma Normal (5NF)

La 5FN se da si una estructura de datos está en 4FN y además la dependencia de juntura (join) tiene implicación directa de las claves candidatas.

Tablas 2.7.4.6\_1-2-3-4-5-6: Quinta Forma Normal

<b>FACTURA</b>
<b>Número_Factura</b>
<b>Código_Cliente</b>
Fecha_Factura
Subtotal_Factura
IVA_Factura
Total_Factura
Observaciones_Factura

<b>CLIENTE</b>
<b>Código_Cliente</b>
Número_Cliente
Dirección_Cliente
Teléfono_Cliente
Descuento_Cliente

<b>ITEM</b>
<b>Número_Factura</b>
<b>Código_Producto</b>
Número_Item
Cantidad_Item
Valor_Total_Item
Monto_Descuento

<b>PRODUCTO</b>
<b>Código_Producto</b>
Descripción_Producto
Precio_Unitario_Producto

<b>DESCUENTOS_FACTURA</b>
<b>Número_Factura</b>
<b>Código_Descuento</b>
Valor_Descuento_Factura
Descripción_Descuento

<b>DESCUENTO</b>
<b>Código_Descuento</b>
Rubro_Descuento
Bono_Descuento
Porcentaje_Descuento

## Ejemplos:

1. Suponer que se tiene la siguiente tabla:

**Departamento:** ID\_Alumno, Nombre\_Alumno, Edad\_Alumno,  
Codigo\_Carrera, Nombre\_Carrera, Codigo\_Materia,  
Semestre\_Materia, Nota\_Materia, Codigo\_Profesor,  
Nombre\_Profesor, Codigo\_Curso, Nombre\_Curso

Normalizar hasta 3NF

1NF: ID\_Alumno, Nombre\_Alumno, Edad\_Alumno

2FN: Codigo\_Carrera, Nombre\_Carrera

3NF: Codigo\_Materia, Nota\_Materia, Semestre\_Materia

4FN: Codigo\_Profesor, Nombre\_Profesor

5FN: Codigo\_Curso, Nombre\_Curso

2. Se tiene el siguiente esquema: R(A,B,C,D,E)

Sea tiene el siguiente esquema: **R (a, b, c, d, e)**

Y el conjunto de dependencias funcionales **F = {a ->bc, c->d, b->e}**.

Normalizar en FNBC

R1 (a,b,c)

R2 (b,e)

R3 (c,d)

[Volver Inicio](#)



## U2. Lección 7 tarea 1

### ACTIVIDAD DE APRENDIZAJE

---

Denominación de la tarea:

Dependencias Funcionales y Formas Normales

### **ASESORÍA DIDÁCTICA**

Para desarrollar la siguiente actividad de aprendizaje, debe leer y comprender el contenido de la Lección 7 Unidad 2.

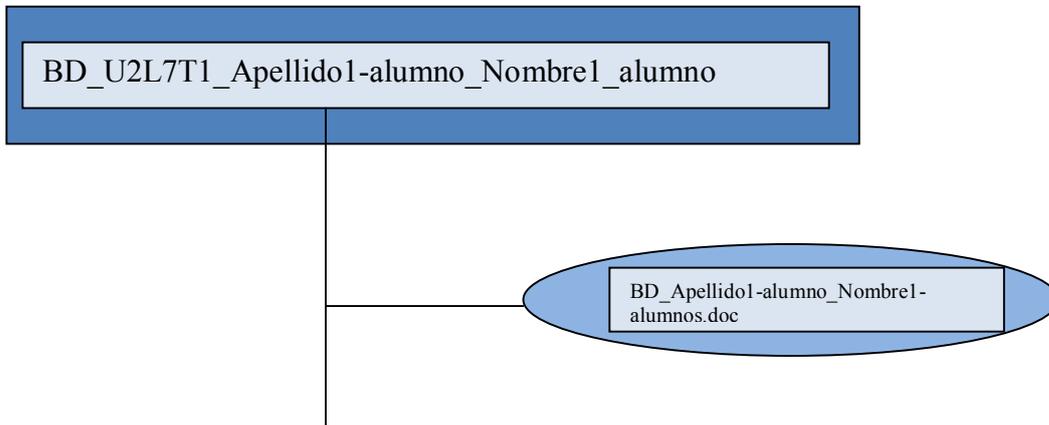
### **ACTIVIDAD DE APRENDIZAJE**

Se tiene la siguiente relación PRÉSTAMO\_LIBROS (Unidad\_Educativa, Profesor, Asignatura, Aula, Curso, Libro, Biblioteca, Fecha\_Préstamo) que contiene información relativa a los préstamos que realizan las bibliotecas a los profesores de un colegio para su evaluación en alguna de las asignaturas que imparten. Se solicita:

- a) Indicar las dependencias funcionales utilizando las siguientes abreviaturas:  
Unidad\_Educativa (UE), Profesor (P), Asignatura (A), Aula (AU), Curso (CU), Libro (L), Biblioteca (B) y Fecha\_Préstamo (F)
- b) En qué forma normal se encuentra la relación y por qué.

Envíe a su tutor la tarea en una carpeta comprimida como .rar

Formato de la carpeta:





## U2. Lección 7 tarea 2

### ACTIVIDAD DE APRENDIZAJE

---

Denominación de la tarea:

Formas Normales

#### ASESORÍA DIDÁCTICA

Para desarrollar la siguiente actividad de aprendizaje, debe leer y comprender el contenido de la Lección 7 Unidad 2.

#### ACTIVIDAD DE APRENDIZAJE

En cierta Universidad existe una residencia donde viven los alumnos. Además existen clubes y cada alumno puede pertenecer a uno o más de estos clubes.

Considere las siguientes tablas, que describen la situación:

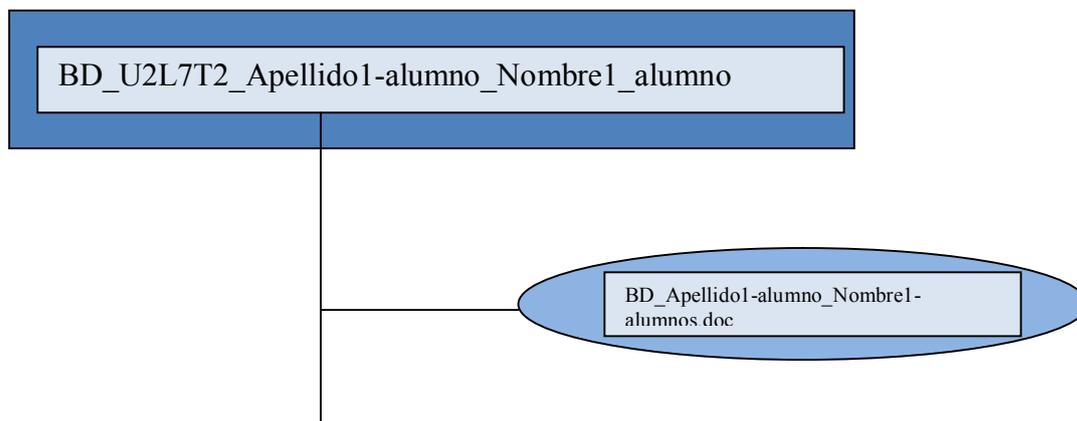
ALUMNO\_HABITACION(Id\_Alumno, Habitación,  
Precio\_Semestral\_Habitación)

ALUMNO\_CLUB(Id\_Alumno, Club, Precio\_Semestral\_Club)

- Para cada tabla indicar la dependencia funcional y forma normal, además la clave.
- Cada tabla transformar a Tercera Forma Normal

Envíe a su tutor la tarea en una carpeta comprimida como .rar

Formato de la carpeta:





## Resumen Lección 7

---

### Normalización

Esta lección ha presentado una visión global de base de datos, incluyendo:

- **Dependencia Funcional**
- **La necesidad de normalizar**
- **Formas Normales**

#### **Destacándose que:**

Se conoce a la normalización como el procedimiento en el que se utilizan normas o reglas (cada una de estas se basa en un antecedente) sobre las relaciones que se derivan al pasar del modelo entidad – relación al modelo relacional.

Las formas normales son:

#### **Primera Forma Normal (1NF)**

Esta regla define que se debe eliminar las columnas que se repiten y ponerlas separadamente en otras tablas. Esta forma normal soluciona el inconveniente del encabezamiento de las columnas múltiples es decir columnas que contengan datos idénticos.

#### **Segunda Forma Normal (2NF)**

Esta regla define que se debe eliminar las dependencias parciales (son los a los que se les puede identificar sin depender de la clave primaria) e independizarlas en otras tablas.

#### **Tercera Forma Normal (3NF)**

Esta regla define que se debe colocar en otra tabla los datos que no depende de la clave primaria, cuando se tiene la tercera forma normal se evitan errores lógicos al insertar o borrar datos.

### **Forma Normal de Boyce – Codd (FNBC)**

Es la reestructuración de la tercera forma normal , la cual indica que una relación se encuentra en esta forma cuando el determinante es clave candidata.

### **Cuarta Forma Normal (4NF)**

Esta regla de normalización indica que una relación está en 4FN cuando se encuentra en BCNF y es independiente de valores múltiples.

### **Quinta Forma Normal (5NF)**

Esta regla de normalización no se aplica en todos los casos ya que excluye dependencias de proyección que no se encuentran regularmente en las bases de datos.

## UNIDAD 2: PROCESO DE DESARROLLO DE LA BASE DE DATOS

### LECCIÓN 2.8: Diccionario de Datos



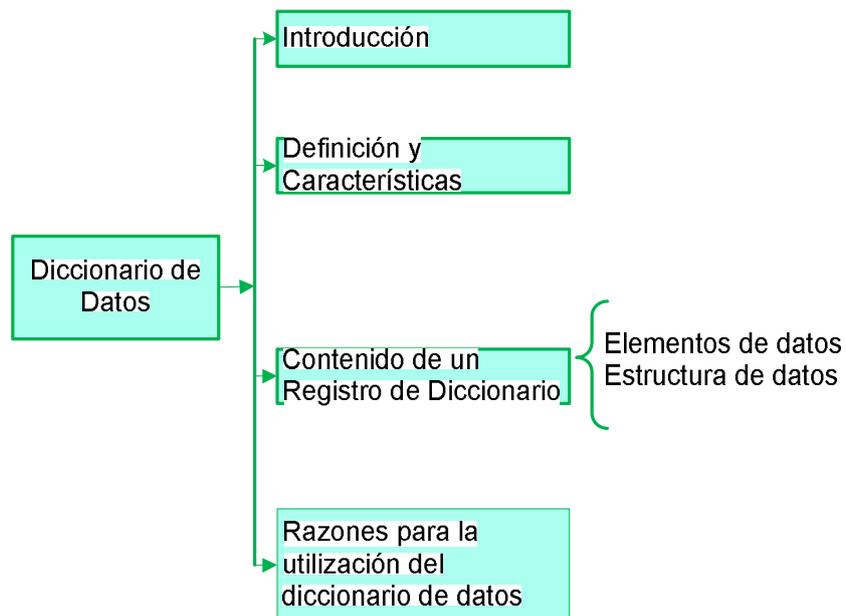
#### Visión General

En esta lección se trata el Diccionario de Datos que organiza los datos que pertenecen a un determinado sistema, sus características, contenido (elementos y estructura de datos).

[Volver Inicio](#)



#### Plan de la Lección





## Desarrollo de la Lección

### 2.8.1. Introducción

El SGBD facilita las funciones del diccionario de datos o catálogo del sistema que es el almacenamiento de información del detalle y la implantación de una base de datos. Se puede guardar en el diccionario esquemas, restricciones de integridad y seguridad. El diccionario de datos debe permitir realizar consultas igual que una base de datos.

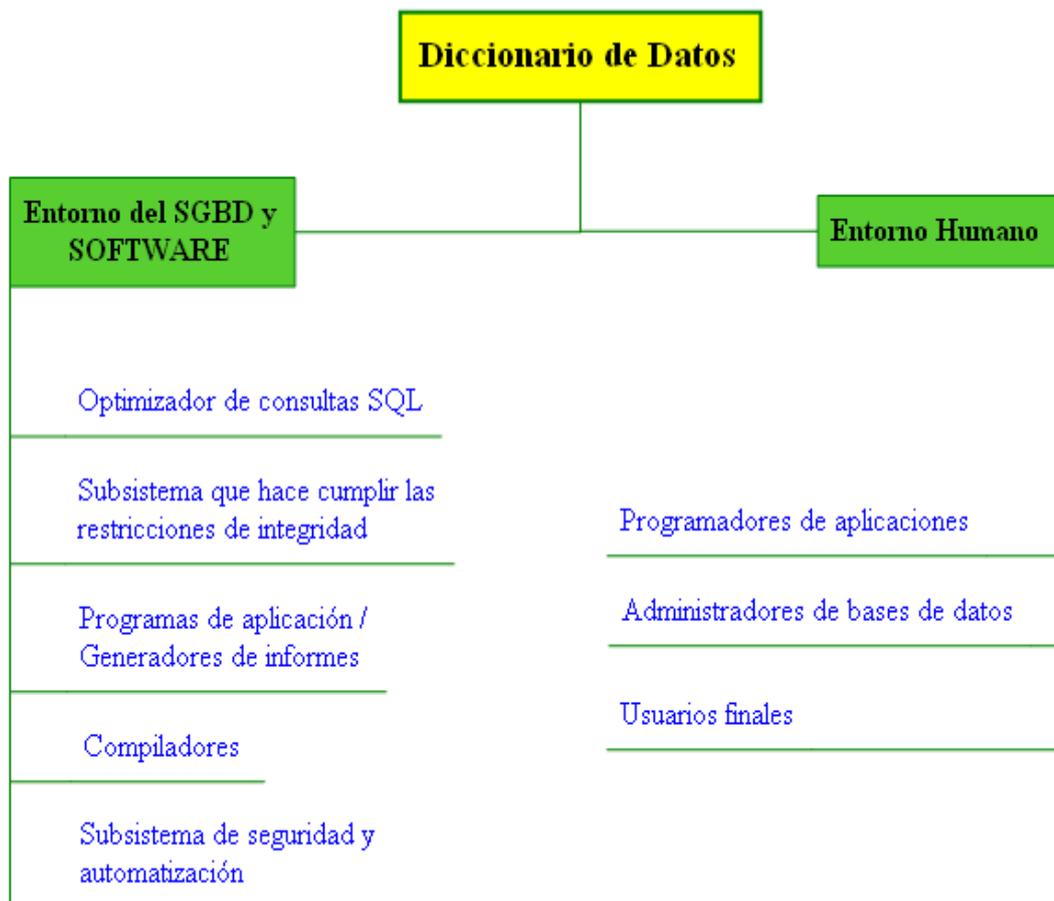


Figura 2.8.1\_1: Entorno del Diccionario de Datos

### **2.8.2. Definición y Características**

Un diccionario de datos es la agrupación de metadatos, el cual se usa para tener un detalle de las tablas que han sido generadas por el usuario, diseñador o los dos. A continuación las características más importantes del diccionario de datos:

- Almacena y establece los detalles del Diagrama de de Flujo de Datos (DFD), y todos sus elementos.
- Muestra los procesos en donde están los datos y el lugar donde se requiere acceder oportunamente a la información.

### **2.8.3. Contenido de un registro del diccionario**

El diccionario de datos se compone de:

#### **2.8.3.1. Elementos de datos**

Su colección establece la estructura de datos, son las unidades básicas y son importantes en el sistema de información. Ejemplo: cédula de identidad, nombre, dirección.

El diccionario de datos detalla los siguientes elementos de datos:

- Name: Se refiere al nombre del elemento de datos
- Alias: Se denomina alias a otro nombre que se use para identificar a los elementos de datos.
- Type y Size: En el tipo (type) se describe al elemento de datos que tiene valor numérico o carácter. El tamaño (size) detalla los dígitos o caracteres máximos que posee los elementos de datos.
- Output Format or Edit Mask: Permite observar como aparece un reporte sea en pantalla o mediante impresión.
- Default Value: Se refiere al valor del elemento de datos.

- Prompt, Column Header or Field Caption: Es un título que se muestra en pantalla o en el reporte.
- Source: Muestra el inicio del valor del elemento de datos.
- Security: Determina los usuarios que tienen permiso para cambiar el elemento de datos.
- Responsible User: Determina los usuarios encargados de ingresar y modificar los valores que constan en el elemento de datos.
- Acceptable Data and Data Validation: Se detallan los valores aceptados, cumpliendo las reglas de validación.
- Derivation Formula: Se refiere a las fórmulas que se usan para obtener un resultado.
- Description or Comments: Son las descripciones de información adicional.

### **2.8.3.2. Estructura de datos**

Se denomina a la relación de un conjunto de datos que detallan un componente del sistema.

- **Data Flows (Flujo de Datos)**

Las características del flujo de datos son:

- Name: Se refiere al nombre del flujo de datos como se encuentra en el DFD (Diagrama de Flujo de Datos).
- Alias: Se refiere a otro nombre con el que se identifique al flujo de datos

- Abreviation or ID: Es el número de identificación que permite acceder rápidamente al flujo de datos.
- Description: Es el detalle del flujo de datos y su objetivo.
- Origin: Muestra el origen del flujo de datos.
- Destination: Se refiere a la terminación del flujo de datos.
- Record: El flujo de datos simboliza un conjunto de elementos de datos relacionados, los records se describen separadamente de los flujos de datos.
- Volume and Frequency: Se detalla el número de ocurrencias que se espera en un tiempo determinado para el flujo de datos.

### **2.8.3.3. Razones para su utilización del diccionario de datos**

El diccionario de datos se usa por las siguientes razones:

- Para manejar los datos en sistemas ya que éstos sufren cambios permanentemente por lo que se vuelve complejo admitir todas las descripciones, por lo que es necesario un registro de la información es por esto que aparece el diccionario de datos que sirve para analizar y diseñar el software.
- Es importante conocer los documentos de los componentes, características y procesos del sistema, esto facilitará el entendimiento de la información.
- Realizar un análisis de cambios del sistema involucra conocer cómo se recupera la información lo que permitirá obtener determinadas respuestas, detallar las transacciones trascendentales para la organización, además la habilidad del sistema para efectuar el proceso y almacenar datos.



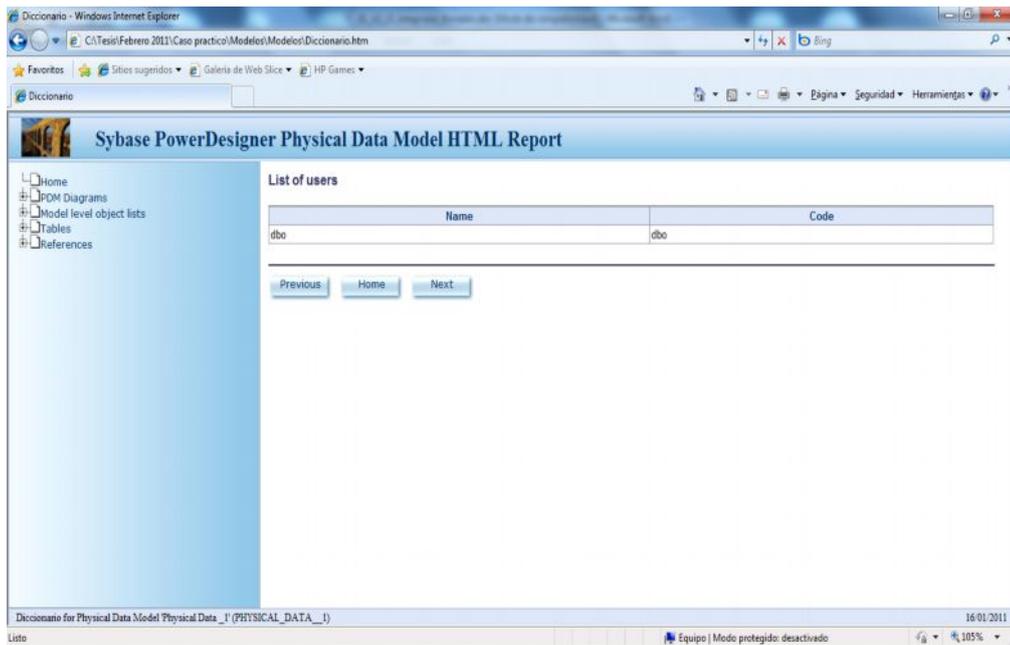


Figura 2.8.3.3\_3: Diccionario de Datos

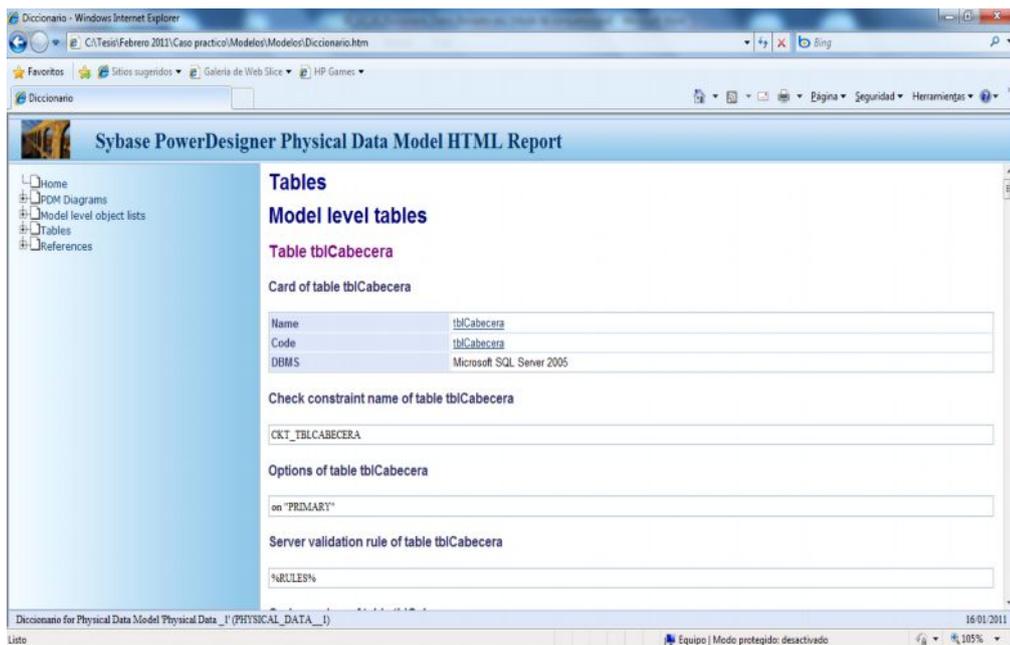


Figura 2.8.3.3\_4: Diccionario de Datos

[Volver Inicio](#)



## U2. Lección 8 tarea 1

### ACTIVIDAD DE APRENDIZAJE

---

Denominación de la tarea:

Diccionario de Datos

### **ASESORÍA DIDÁCTICA**

Para desarrollar la siguiente actividad de aprendizaje, debe leer y comprender el contenido de la Lección 8 Unidad 2.

### **ACTIVIDAD DE APRENDIZAJE**

Envíe a su tutor la tarea en una carpeta comprimida como .rar

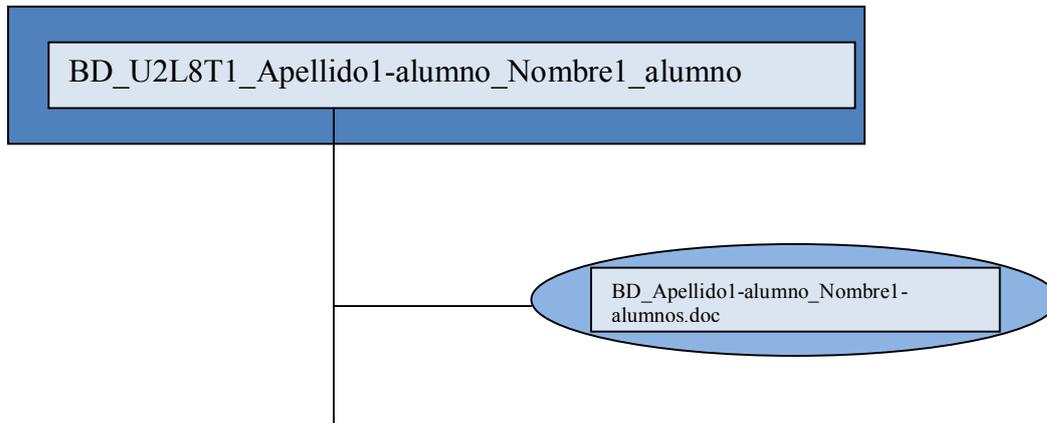
Obtener el diccionario de datos del enunciado de la Tarea 1 Unidad 2 Lección 5. Se desea almacenar la información de una Compañía Aérea en una Base de Datos relacional. La Compañía Aérea tiene tres recursos principales: Aviones, pilotos, tripulación. De cada pila se desea conocer su código, nombre y horas de vuelo. De los miembros de la tripulación solo se tendrá el código y el nombre.

Los pilotos y tripulación tienen una base a la que regresan después de cada jornada un vuelo va desde un origen a un destino a una hora concreta y tiene número de vuelo, de cada vuelo que se va a realizar durante los próximos 2 meses, así como de los vuelos que se han realizado se desea saber el avión en el que se va a hacer o en el que se ha hecho, el piloto y la tripulación.

Cada avión tiene un código, es de un tipo (boing, airbus, entre otros). Y tiene una base donde es sometido a mantenimiento.

Envíe a su tutor la tarea en una carpeta comprimida como .rar

Formato de la carpeta:





## Resumen Lección 8

---

### Diccionario de Datos

Esta lección ha presentado una visión global del diccionario de datos incluyendo:

- **Definición y Características**
- **Contenido de un registro en el diccionario**
  - o **Elementos de Datos**
  - o **Estructura de Datos**
  - o **Razones para la utilización del diccionario de datos**

#### **Destacándose que:**

Un diccionario de datos es la agrupación de metadatos, el cual se usa para tener un detalle de las tablas que han sido generadas por el usuario, diseñador o los dos.

El contenido de un registro del diccionario es:

#### **Elementos de datos:**

Su colección establece la estructura de datos, son las unidades básicas y son importantes en el sistema de información.

#### **Estructura de datos:**

Se denomina a la relación de un conjunto de datos que detallan un componente del sistema. El diccionario de datos se usa para:

Manejar los datos en sistemas ya que éstos sufren cambios permanentemente por lo que se vuelve complejo admitir todas las descripciones, por lo que es necesario un registro de la información es por esto que aparece el diccionario de datos que sirve para analizar y diseñar el software.

## UNIDAD 2: PROCESO DE DESARROLLO DE LA BASE DE DATOS

### LECCIÓN 2.9: Implementación de la base de datos utilizando SQL – DDL



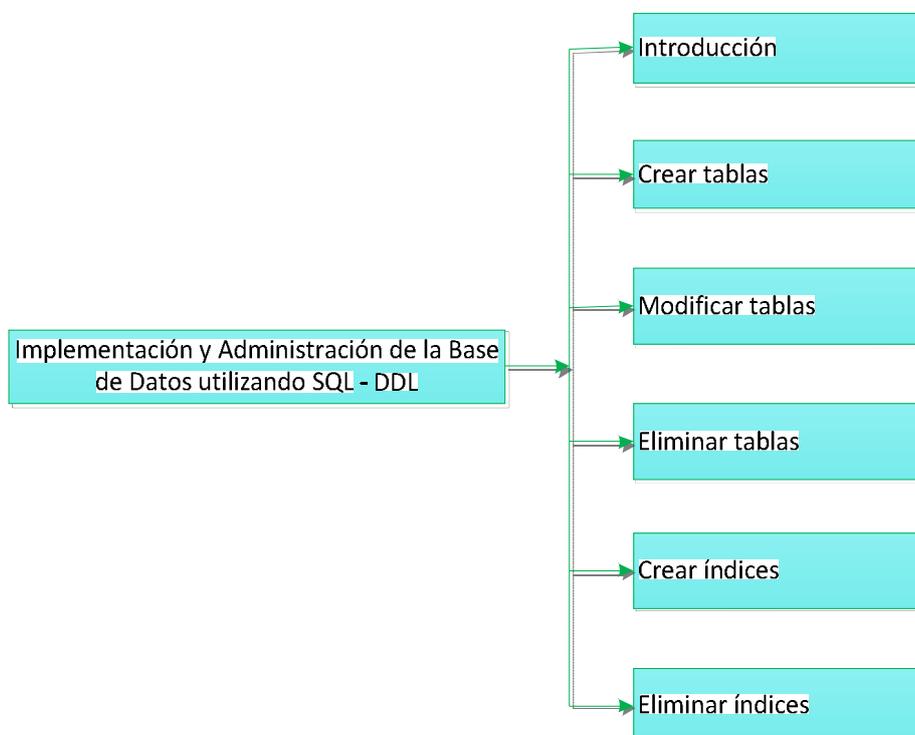
#### Visión General

En esta lección se trata la implementación de la base de datos aplicando el lenguaje de definición de datos (DDL). El DDL (Lenguaje de Definición de Datos) es la parte de SQL más variable de un sistema a otro ya que muestra la organización interna de los datos.

[Volver Inicio](#)



#### Plan de la Lección





## Desarrollo de la Lección

### 2.9.1. Introducción

El Lenguaje de Definición de Datos (DDL) es parte de SQL, permite especificar el esquema de la base de datos mediante una colección de definiciones.

Las sentencias de DDL son usadas por el administrador de una base de datos y contiene desde la creación de los objetos que forman una base de datos y el respectivo mantenimiento.

DDL admite sentencias como CREATE, DROP y ALTER, esto se aplica a tablas, vistas, procedimientos almacenados y triggers, en esta lección se detallará estas sentencias aplicadas a tablas.

### 2.9.2. Crear Tablas (CREATE TABLE)

Esta sentencia crea una nueva tabla en la base de datos y la prepara para admitir datos. La tabla debe tener un nombre diferente al de las otras tablas de la base de datos. La sentencia CREATE TABLE tiene la siguiente sintaxis:

```
CREATE TABLE nombre_tabla(nombre_columna tipo,  
restricción1, restricción2)
```

Figura 2.9.2\_1: Create Table

Tipo: Es el tipo de dato de la columna.

Restricción: Es una característica adicional de una columna como valores no nulos, clave principal, clave foránea.

Existen dos tipos de restricciones:

“**Restricción1**: una restricción de tipo 1 es una restricción que aparece dentro de la definición de la columna después del tipo de dato y afecta a una columna, la que se está definiendo.

**Restricción2**: una restricción de tipo 2 es una restricción que se define después de definir todas las columnas de la tabla y afecta a una columna o a una combinación de columnas.

Para escribir una sentencia **CREATE TABLE** se empieza por indicar el **nombre de la tabla** que queremos crear y a continuación **entre paréntesis** indicamos **separadas por comas las definiciones de cada columna** de la tabla, la definición de una columna **consta de su nombre, el tipo de dato** que tiene **y** podemos añadir **si queremos una serie de especificaciones** que deberán cumplir los datos almacenados en la columna, **después** de definir cada una de las columnas que compone la tabla **se pueden añadir** una serie de **restricciones**, esas restricciones son las mismas que se pueden indicar para cada columna pero ahora **pueden afectar a más de una columna** por eso tienen una sintaxis ligeramente diferente”<sup>65</sup>

Son restricciones de tipo 1:

- **NOT NULL**

Esta sentencia muestra que la columna no debe tener valores nulos, cuando se requiera indicar que una columna no tenga valor nulo se lo puede realizar sin usar la condición CONSTRAINT.

- **CONSTRAINT**

Esta sentencia se usa para determinar una restricción que se puede eliminar sin borrar la columna, se puede asignar un nombre a cada restricción que se utilizará para la identificación y eliminación cuando se requiera.

---

<sup>65</sup> [http://www.aulacltic.es/sql/t\\_8\\_1.htm](http://www.aulacltic.es/sql/t_8_1.htm)

- **PRIMARY KEY**

Esta condición se usa para determinar la columna como clave primaria de una tabla, lo que se requiere es que la columna no tenga valores nulos, no debe constar de valores duplicados.

- **UNIQUE**

Esta condición permite describir un índice único (no admite valores duplicados), en una columna. Si se detalla esta restricción no puede existir dos filas que contengan igual valor, se usa esta sentencia para que el sistema examine que no hayan valores que ya existen.

- **REFERENCES**

Esta sentencia define a una “clave foránea que es una columna o conjunto de columnas que contiene un valor que hace referencia a una fila de otra tabla, después de la palabra reservada se indica a qué tabla hace referencia, opcionalmente se puede indicar entre paréntesis el nombre de la columna donde tiene que buscar el valor de referencia, por defecto coge la clave principal de la tabla2, si el valor que tiene que buscar se encuentra en otra columna de tabla2, entonces se debe indicar el nombre de esta columna entre paréntesis, además sólo se puede utilizar una columna que esté definida con una restricción de **UNIQUE**, si la columna2 que indicamos no está definida sin duplicados, la sentencia **CREATE** nos dará un error.”<sup>66</sup>

Ejemplo:

```
CREATE TABLE tabla1(  
columna1 INTEGER CONSTRAINT pk PRIMARY KEY,  
columna2 CHAR(20) NOT NULL,  
columna3 CHAR(15) CONSTRAINT uni1 UNIQUE,  
columna4 INT CONSTRAINT fk4 REFERENCES tablan);
```

---

<sup>66</sup> [http://www.aulacltic.es/sql/t\\_8\\_1.htm](http://www.aulacltic.es/sql/t_8_1.htm)

Se ha creado la tabla 1 formada por una columna 1 tipo entero que será la clave primaria, la columna 2 tipo carácter no puede contener valores nulos, la columna 3 igual de tipo carácter no puede contener valores que se repitan, la columna 4 de tipo entero clave foránea hace referencia a la clave principal de la tabla. Son restricciones de tipo 2:

- **FOREIGN KEY**

“Sirve para definir una clave foránea sobre una columna o una combinación de columnas. Una clave foránea es una columna o conjunto de columnas que contiene un valor que hace referencia a una fila de otra tabla, en una restricción 1 se puede definir con la cláusula REFERENCES. Para definir una clave foránea en una restricción de tipo 2 debemos empezar por las palabras FOREIGN KEY después indicamos entre paréntesis la/s columna/s que es clave foránea, a continuación la palabra reservada REFERENCES seguida del nombre de la tabla a la que hace referencia, opcionalmente podemos indicar entre paréntesis el nombre de la/s columna/s donde tiene que buscar el valor de referencia, por defecto coge la clave principal de la tabla2, si el valor que tiene que buscar se encuentra en otra/s columna/s de tabla2, entonces debemos escribir el nombre de esta/s columna/s entre paréntesis, además sólo podemos utilizar una columna (o combinación de columnas) que esté definida con una restricción e UNIQUE, de lo contrario la sentencia CREATE TABLE nos dará un error.”<sup>67</sup>

Ejemplos:

Crear la tabla dbo.tblZona:

```
CREATE TABLE dbo.tblZona(Zonaid INTEGER CONSTRAINT pk
PRIMARY KEY NOT NULL, ZonaNombre VARCHAR(50) NULL)
```

---

<sup>67</sup> [http://www.aulacltic.es/sql/t\\_8\\_2.htm](http://www.aulacltic.es/sql/t_8_2.htm)

Crear la tabla dbo.tblSucursal:

```
CREATE TABLE dbo.tblSucursal(SucId INTEGER CONSTRAINT pk
PRIMARY KEY NOT NULL, SucNombre VARCHAR(50) NULL,
SucDireccion VARCHAR(50) NULL, SucTelefono VARCHAR(50) NULL)
```

### 2.9.3. Modificar Tablas (ALTER TABLE)

La sentencia ALTER TABLE permite agregar nueva columna a la tabla, determinar una clave primaria o foránea y eliminarlas también. Permite además la creación de nuevas restricciones o borrar las que existen. Se debe conocer el significado de los siguientes términos: ADD añadir, ALTER modificar, DROP eliminar, CONSTRAINT restricción.

Su sintaxis es la siguiente:

<pre><b>ALTER TABLE</b> nombre_tabla <b>ADD</b> nombre_columna tipo restriccion1 <b>ADD</b> restricción 2 <b>DROP</b> nombre_columna <b>DROP</b> constraint nombre_restriccion</pre>
--

Figura 2.9.3\_1: Sentencia Alter Table

Existe también la sentencia **ADD COLUMN** que agrega una nueva columna en una tabla.

Ejemplo:

```
ALTER TABLE dbo.tblZona ADD ZonaDireccion varchar(50) NOT NULL;
```

En este ejemplo se está agregando a la tabla dbo.tblZona una columna ZonaDireccion tipo varchar(50), no admite valores nulos.

Se usa la sentencia **DROP COLUMN** para borrar cierta columna.

Ejemplo:

```
ALTER TABLE dbo.tblZona DROP COLUMN ZonaDireccion
```

Existe otra sintaxis para esta sentencia:

```
ALTER TABLE dbo.tblZona DROP ZonaDireccion
```

Se usa la sentencia DROP CONSTRAINT para eliminar una restricción, únicamente de borra la definición de restricción y no los datos.

Ejemplo:

```
ALTER TABLE dbo.tblZona DROP CONSTRAINT c1
```

#### 2.9.4. Eliminar Tablas (DROP TABLE)

La sentencia DROP TABLE elimina una tabla que no requiera la base de datos. Se eliminan tablas propias del usuario y de otros siempre que se tengan los permisos suficientes. Se debe tener cuidado al usar esta sentencia, ya que tiene efectos graves se pierde todo el contenido de una respectiva tabla. Esta sentencia tiene la siguiente sintaxis:

**DROP TABLE** *nombre\_tabla*

Figura 2.9.4\_1: Sentencia Drop Table

Ejemplos:

```
DROP TABLE dbo.tblZona
```

Con esta sentencia se borra la tabla dbo.tblZona

#### 2.9.5. Crear Índices (CREATE INDEX)

Sentencia que se usa para crear un índice sobre columnas de cierta tabla.

La sintaxis es la siguiente:

**CREATE INDEX** *nombre\_indice* **ON**  
*nombre\_tabla(nombre\_columna)*

Figura 2.9.5\_1: Sentencia Create Index

Luego del nombre de la columna se puede usar las cláusulas ASC o DESC.

La sentencia **ASC** se asume por defecto e indica el orden ascendente para el índice.

La sentencia **DESC** ordena descendentemente a los índices.

Ejemplo:

```
CREATE UNIQUE INDEX índice1 ON dbo.tblSucursal (SucNombre ASC, SucTelefono DESC)
```

Sobre la tabla dbo.tblSucursal se crea un índice1 que está compuesto por las columnas SucNombre y SucTelefono

Cuando se agrega la cláusula UNIQUE el índice no va a permitir duplicados en este caso dos sucursales no podrán tener el mismo teléfono.

### 2.9.6. Eliminar Índices (DROP INDEX)

Esta sentencia elimina sólo un índice más no las columnas que lo forman.

La sintaxis es la siguiente:

<pre><b>DROP INDEX</b> nombre_indice ON nombre_tabla</pre>
--

Figura 2.9.6\_1: Sentencia Drop Table

Ejemplo:

```
DROP INDEX indice1 ON dbo.tblSucursal
```

[Volver Inicio](#)



## U2. Lección 9 tarea 1

### ACTIVIDAD DE APRENDIZAJE

---

Denominación de la tarea: **CREACIÓN DE TABLAS**

#### **ASESORÍA DIDÁCTICA**

Para desarrollar la siguiente actividad de aprendizaje, debe leer y comprender el contenido de la Lección 9 Unidad 2.

#### **ACTIVIDAD DE APRENDIZAJE**

Se tiene los siguientes datos:

EMPLEADO(codigo\_empleado, cedula, nombre, dirección, departamento, cargo)

SALARIOS (codigo\_empleado, fecha\_pago, sueldo, descuentos, cargo)

DEPARTAMENTO(codigo\_empleado, nombre\_departamento, dirección)

CARGO(codigo\_cargo, detalle)

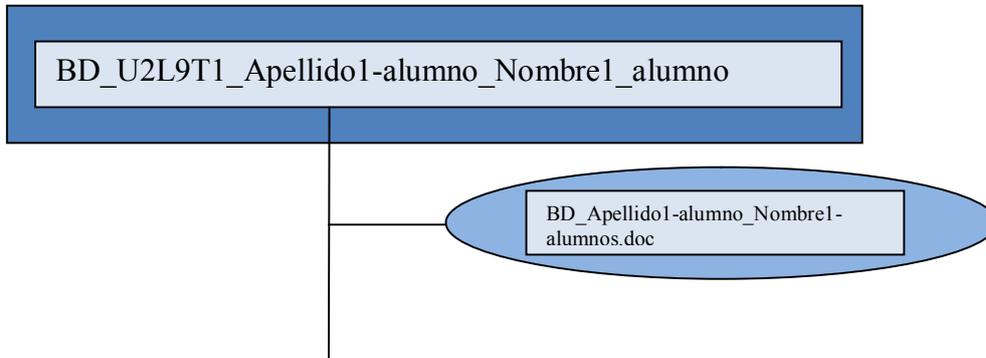
Se solicita:

Crear las tablas indicadas bajo las siguientes consideraciones:

- Confirmar que solo puedan existir empleados del departamento 10, 20, 30, 40
- Los departamentos solo pueden ser de Quito, Guayaquil, Cuenca

Envíe a su tutor la tarea en una carpeta comprimida como .rar

Formato de la carpeta:





## U2. Lección 9 tarea 2

### ACTIVIDAD DE APRENDIZAJE

---

Denominación de la tarea: **CREACIÓN DE TABLAS**

#### ASESORÍA DIDÁCTICA

Para desarrollar la siguiente actividad de aprendizaje, debe leer y comprender el contenido de la Lección 9 Unidad 2.

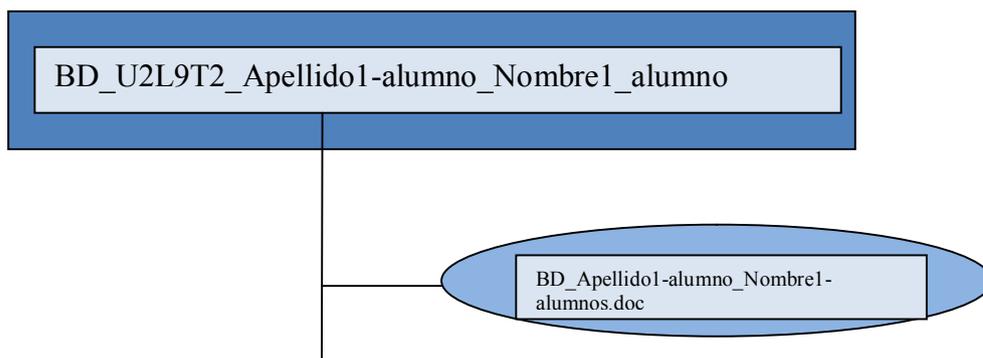
#### ACTIVIDAD DE APRENDIZAJE

Se solicita obtener las sentencias para el siguiente enunciado.

- Crear una tabla Oficina con las respectivas claves principal y foránea.
- Crear una tabla Producto con la respectiva clave principal.
- Crear la tabla Pedido sin clave principal, con la clave foránea que haga referencia a Producto.
- Borrar la tabla Oficina.

Envíe a su tutor la tarea en una carpeta comprimida como .rar

Formato de la carpeta:





## Resumen Lección 9

---

### Implementación y Administración de la Base de Datos utilizando SQL - DDL

Esta lección ha presentado una visión global de SQL -DDL incluyendo:

- **Creación de Tablas**
- **Modificación de Tablas**
- **Eliminación de Tablas**
- **Creación de Índices**
- **Eliminación de Índices**

**Destacándose que:**

**CREATE TABLE** se usa para crear una nueva tabla en la base de datos y la prepara para admitir datos. La tabla debe tener un nombre diferente al de las otras tablas de la base de datos.

**ALTER TABLE** permite agregar nueva columna a la tabla, determinar una clave primaria o foránea y eliminarlas también. Permite además la creación de nuevas restricciones o borrar las que existen.

**DROP TABLE** elimina una tabla que no requiera la base de datos. Se eliminan tablas propias del usuario y de otros siempre que se tengan los permisos suficientes.

#### **CREATE INDEX**

Sentencia que se una para crear un índice sobre columnas de cierta tabla.

#### **DROP INDEX**

Esta sentencia elimina sólo un índice más no las columnas que lo forman.

## UNIDAD 3: SQL LENGUAJE DE CONSULTA RELACIONAL

### LECCIÓN 3.1: SQL-DML



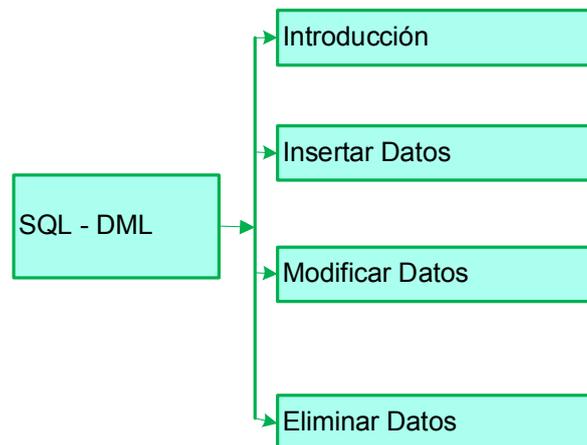
#### Visión General

En esta lección se explica el lenguaje de manipulación de datos que permite realizar consultas de datos organizados, las sentencias DML son Insert, Delete y Update.

[Volver Inicio](#)



#### Plan de la Lección





## Desarrollo de la Lección

### 3.1.1. Introducción

Una de las funciones del SGBD es proporcionar un lenguaje de manipulación de datos por medio del cual se puedan realizar instrucciones que realicen procesamientos requeridos. El Lenguaje de Manipulación de Datos (DML), permite a los usuarios insertar, actualizar, borrar y extraer datos de una base de datos.

En la Unidad 3 se usarán las siguientes tablas de la base de datos proyecto:

Tabla - dbo.tblCabecera							
	CabId	CabNombre	CabFecha	CabTotal	Responsable	Sucursal	Vehiculo
	1	Gaseosa A	05/08/2011 0:0...	100	1723491030	6	201
	2	Gaseosa B	11/08/2011 0:0...	100	1723491030	6	201
	3	Gaseosa A	15/08/2011 0:0...	50	1766001023	6	202
	4	Gaseosa F	17/08/2011 0:0...	200	1718983201	7	203
	5	Gaseosa D	20/08/2011 0:0...	300	1719203882	7	204
	6	Gaseosa F	25/08/2011 0:0...	200	1734987048	7	205

Tabla 3.1.1\_1: TLBCABECERA

Tabla - dbo.tblDetalle_cabecera			
	DetCabId	Cabecera	Producto
	1	1	1380
	2	2	1381
	3	3	1380
	4	4	1385
	5	5	1383
	6	6	1385

Tabla 3.1.1\_2: TLBDETALLE\_CABECERA

Tabla - dbo.tblPersonal							
	Id	IdIdentificacion	Nombres	Telefono	Profesion	Cargo	EstadoCivil
	1	1732109870	Lorena Escobar	022335890	Ingeniero	Jefe de Ventas	Casado
	3	1716623056	Luis Almeida	022678512	Contador	Vendedor	Soltero
	4	1790872380	María Romero	022645120	Ingeniero	Jefe Contabilidad	Casado

Tabla 3.1.1\_3: TLBPERSONAL

Tabla - dbo.tblProducto			
	ProdId	ProdNombre	ProdDescripcion
	1380	Gaseosa A	1 Litros
	1381	Gaseosa B	2 Litros
	1382	Gaseosa C	2 Litros
	1383	Gaseosa D	1 Litro
	1384	Gaseosa E	2 Litros
	1385	Gaseosa F	1 Litros

Tabla 3.1.1\_4: TLBPRODUCTO

Tabla - dbo.tblResponsable				
	RespCedula	RespNombre	RespDireccion	RespTelefono
	1718983201	Doris Escobar	Guayaquil	028974791
	1719203882	Luis Ayala	Esmeraldas	042894865
	1720678921	Juan López	Ambato	022334908
	1723491030	José Morales	Quito	022074567
	1728811040	Juan Pérez	Loja	074666454
	1734987048	Marcela Acosta	Ibarra	022678309
	1766001023	Carlos Ruiz	Cuenca	062288191

Tabla 3.1.1\_5: TLBRESPONSABLE

Tabla - dbo.tblSucursal						
	SucId	SucNombre	SucDireccion	SucTelefono	Responsable	Zona
	5	Ambato	Ambato	022334789	1720678921	2
	6	Quito	Quito	032897178	1723491030	3
	7	Ibarra	Ibarra	067299212	1734987048	4
	8	Guayaquil	Guayaquil	047829109	1718983201	5
	9	Esmeraldas	Esmeraldas	062992111	1719203882	2
	10	Cuenca	Cuenca	073671719	1766001023	6

Tabla 3.1.1\_6: TLBSUCURSAL

Tabla - dbo.tblVehiculo				
	VehId	VehPlaca	VehAnio	VehDescripcion
	201	PAC-908	2002	Vehículo 1
	202	PBC-2901	2010	Vehículo 2
	203	PBR-3089	2011	Vehículo 3
	204	PCB-2988	2011	Vehículo 4
	205	PAB-1909	2011	Vehículo 5
	206	PBS-7831	2011	Vehículo 6

Tabla 3.1.1\_7: TLBVEHICULO

Tabla - dbo.tblZona		
	ZonaId	ZonaNombre
	1	Quito
	2	Esmeraldas
	3	Ambato
	4	Ibarra
	5	Guayaquil
	6	Cuenca
	7	Loja

Tabla 3.1.1\_8: TLBZONA

### 3.1.2. Insertar Datos

El insertar datos significa usar la sentencia INSERT, que permite agregar una nueva fila a una tabla. Como se observará a continuación la sintaxis de esta sentencia consta de la condición INTO que define la tabla que recibe la fila nueva. Además la sintaxis consta de la condición VALUE que describe los valores de datos que tendrá la nueva fila.

La sintaxis de de la inserción de datos es la siguiente:

```
INSERT
INTO nombre de la tabla que se va a insertar
VALUES (conjunto de valores del registro);
```

Figura 3.1.2\_1: Sentencia Insert

Ejemplos:

```
INSERT INTO dbo.tblZona Values (7, 'Loja')
```

En este caso se inserta una nueva Zona en la table dbo.tblZona que tendrá como Zonald :7 y Zona Nombre: Loja.

```
INSERT INTO dbo.tblResponsable Values (1728811040, 'Juan Pérez', 'Loja','074666454')
```

En este caso se inserta un nuevo Responsable en la tabla dbo.tblResponsable que tendrá como RespCedula: 1728811040, RespNombre: Juan Pérez, RespDireccion: Loja, RespTelefono: 074666454

### 3.1.3. Modificar Datos

La modificación de datos implica a la condición UPDATE que actualiza los valores de una o varias columnas en una fila escogida de una tabla. Para poder modificar es necesario tener los suficientes permisos, la tabla que se actualizará se indicará en la sentencia. La condición WHERE permitirá seleccionar filas de la tabla que será actualizada. La condición SET describirá las columnas que serán modificadas y se encarga de calcular los valores nuevos. La sintaxis es la siguiente:

```
UPDATE nombre de la tabla  
SET nombre de la columna = expresión  
WHERE condición de búsqueda
```

Figura 3.1.3\_1: Sentencia Update

Ejemplos:

```
UPDATE dbo.tblZona SET ZonaNombre = 'Portoviejo' WHERE Zonald = '2';
```

En este caso se actualiza el nombre de la Zonald 2 que pertenecía a Esmeraldas por Portoviejo.

```
UPDATE dbo.tbIResponsable SET RespDireccion = 'Portoviejo' WHERE  
RespCedula= ' 1719203882';
```

Esta sentencia actualiza el RespDireccion (Dirección del responsable) perteneciente al responsable con cédula 1719203882 de Esmeraldas a Portoviejo.

### 3.1.4. Eliminar Datos

La eliminación de datos involucra a la condición DELETE la cual elimina filas que son seleccionadas en una tabla. Con la condición FROM se describe la tabla que abarca la fila, además la condición WHERE describe las filas de la tabla que se van a eliminar. La sintaxis es la siguiente:

```
DELETE FROM nombre de tabla de la que se va a eliminar la fila  
WHERE condición de búsqueda
```

Figura 3.1.4\_1 Sentencia Delete

Ejemplos:

```
DELETE FROM dbo.tbISucursal WHERE Suclid = '5'
```

Esta sentencia elimina de la tabla dbo.tbISucursal la sucursal correspondiente al Id 5

[Volver Inicio](#)



### U3. Lección 1 tarea 1

#### ACTIVIDAD DE APRENDIZAJE

---

Denominación de la tarea:

Se desea tener una base de datos con la siguiente información acerca de los alumnos de una Academia de CISCO donde se imparten varios cursos.

- Información acerca de los **alumnos**, que constará de su nombre y apellidos, un código único para cada alumno, su dirección, fecha de nacimiento y sexo('H' o 'M').
- Estos alumnos estarán matriculados en un solo **curso** cada curso tendrá un nombre, un código único que lo identifica, el número máximo de alumnos recomendado, un profesor, la fecha de inicio y de finalización del curso y el número de horas del curso.
- Los **profesores** tendrán una ficha en la academia donde se especifican sus datos personales y los datos del contrato. Es decir, nombre, DNI, dirección, titulación, cuota por hora.

#### **ASESORÍA DIDÁCTICA**

Para desarrollar la siguiente actividad de aprendizaje, debe leer y comprender el contenido de la lección 3.

#### **ACTIVIDAD DE APRENDIZAJE**

1. Generar las siguientes tablas para guardar esta información

**TABLA ALUMNOS****TABLA CURSO****TABLA CURSO**

NOMBRE

NOMBRE\_CURSO

NOMBRE

APELLIDO1

COD\_CURSO

APELLIDO1

APELLIDO2

DNI\_PROFESOR

APELLIDO2

DNI

MAXIMO\_ALUMNOS

DNI

DIRECCION

FECHA\_INICIO

DIRECCIÓN

SEXO

FECHA\_FIN

TITULO

FECHA\_NACIMIENTO

NUM\_HORAS

GANA

CURSO

En las definiciones establecer las siguientes restricciones:

- No es posible dar de alta un alumno si no se matricula en un curso.
- La información del número de horas del curso es imprescindible para almacenarlo.
- El campo GANA de la tabla PROFESORES no puede estar en ningún caso vacío.
- Dos cursos no pueden llamarse igual. Lo mismo le pasa a los profesores.
- Podemos identificar las tuplas de las tablas CURSOS mediante el atributo CODIGO y PROFESORES y ALUMNOS usando el DNI.
- Cumplir la relación normal entre fecha comienzo y fecha fin (orden cronológico).
- Los valores para el atributo sexo son sólo M y H (en mayúsculas).
- Se ha de mantener la regla de integridad de referencia.

2. Insertar las siguientes tuplas:

Tabla **PROFESORES**

<b>NOMBRE</b>	APELLIDO1	APELLIDO2	DNI	DIRECCION	TITULO	GANA
Juan	Montenegro	López	32432455	San Blas	Ing. Informática	1500
María	Romero	Rubio	43215643	San Rafael	Ing. Electrónico	2000

3. Insertar la siguiente tupla en ALUMNOS

NOMBRE	APELLIDO1	APELLIDO2	DNI	DIRECCION	SEXO	FECHA_NAC	CURSO
SERGIO	NAVAS	RETAL	123523	ALHAMAR 3	P		

4. Añadir el campo edad de tipo numérico a la tabla PROFESORES

5. Añadir las siguientes restricciones:

- La edad de los profesores está entre 18 y 65 años
- No se puede añadir un curso si su número de alumnos máximo es menor que 10.
- El número de horas de los cursos debe ser mayor que 100;

6. Eliminar la profesora María Romero

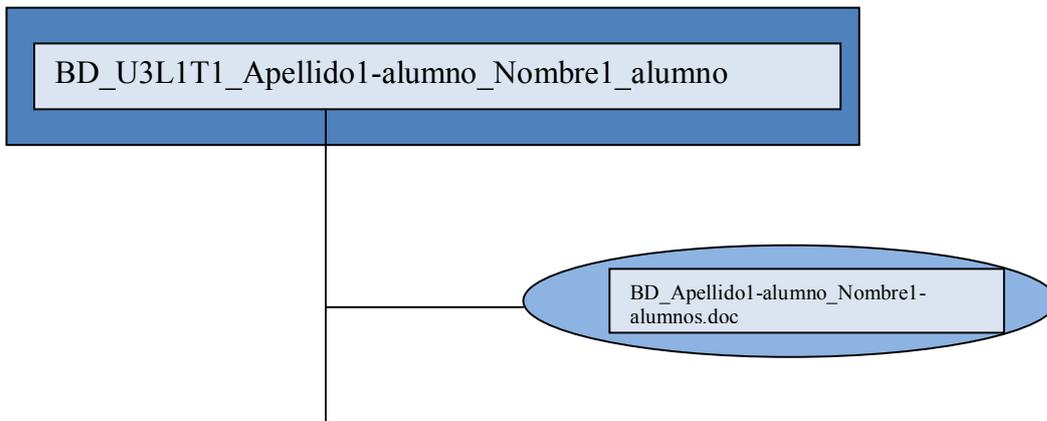
7. Borrar las tablas

PROFESORES

CURSOS

Envíe a su tutor la tarea en una carpeta comprimida como .rar

Formato de la carpeta:





## Resumen Lección 1

---

### SQL – DML

Esta lección ha presentado una visión global de base de datos, incluyendo:

- **Insertar Datos**
- **Modificar Datos**
- **Eliminar Datos**

#### **Destacándose que:**

El insertar datos significa usar la sentencia INSERT, que permite agregar una nueva fila a una tabla.

La modificación de datos implica a la condición UPDATE que actualiza los valores de una o varias columnas en una fila escogida de una tabla.

La eliminación de datos involucra a la condición DELETE la cual elimina filas que son seleccionadas en una tabla.

## UNIDAD 3: SQL LENGUAJE DE CONSULTA RELACIONAL

### LECCIÓN 3.2: Consultas sobre tablas y tablas relacionadas



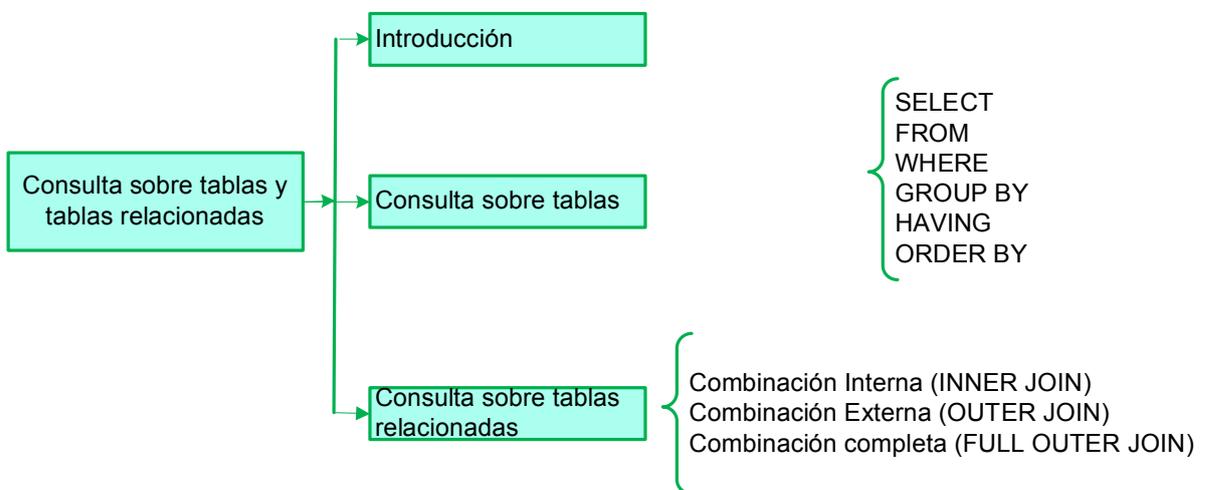
#### Visión General

En esta lección se tratan las consultas sobre tablas y tablas relacionadas dentro de una base de datos.

[Volver Inicio](#)



#### Plan de la Lección





## Desarrollo de la Lección

### 3.2.1. Introducción

La consulta de datos es un proceso importante en una base de datos, es una de las operaciones que se usa frecuentemente con su sentencia SELECT en donde existen condiciones usando WHERE u otras.

### 3.2.2. Consulta Simple

#### 3.2.2.1. SELECT

Esta condición inicia cada sentencia SELECT y describe los parámetros de datos a extraer en una consulta. Los parámetros se detallan por medio de un listado de selección, un listado de parámetros de selección que están espaciados por comas. Cada parámetro de la selección del listado forma una columna de resultado.

La condición SELECT y FROM de la sentencia son muy importantes. Las condiciones WHERE, HAVING, ORDER BY, GROUP BY se las usa opcionalmente. A éstas condiciones se les agrega al SELECT cuando se requiere usarlas.

A continuación se describe la sintaxis de la sentencia SELECT.

```
SELECT parámetro seleccionado  
FROM especificación de la tabla  
WHERE condición de búsqueda  
GROUP BY columna de agrupación  
HAVING condición de búsqueda  
ORDER BY especificación de ordenación
```

Figura 3.2.2.1\_1: Sentencias

### **3.2.2.2. FROM**

A la condición FROM le sigue una descripción de las tablas y éstas se encuentran separadas por comas. Describir una tabla es identificar la tabla que está conformada por datos que serán recuperados por medio de la consulta.

### **3.2.2.3. WHERE**

La condición WHERE se utiliza para definir las filas que vayan a recuperarse. Se selecciona sólo las filas que cumplan la condición esta condición. Ejemplos:

Realizar una sentencia para seleccionar el código del producto Gaseosa A.

```
SELECT ProdId FROM dbo.tbIProducto WHERE ProdNombre = 'Gaseosa A';
```

### **3.2.2.4. GROUP BY**

A la condición GROUP BY se le conoce como función agrupada ya que reúne los datos de las tablas origen y se obtiene una sola fila de cada grupo de filas.

### **3.2.2.5. HAVING**

La sintaxis de esta condición es similar a la de la sentencia WHERE, tiene una condición de búsqueda. Esta condición indica a SQL que se agregue a determinados grupos que se obtuvieron como resultado de la sentencia GROUP BY

#### **3.2.2.6. ORDER BY**

La condición ORDER BY ordena ascendente o descendentemente lo producido por una consulta en una base de datos, se basa en los valores de una o varias columnas.

Se usa opcionalmente las palabras reservadas ASC y DESC que significan ascendente (0-9 A-Z), y descendente (9-0 Z-A), respectivamente, las cuales indican el orden; si no se describe estas palabras el ORDER BY asume que el orden es ascendente.

#### **3.2.2.7. DISTINCT / ALL**

Cuando se incluye la sentencia DISTINCT en el SELECT, se elimina las repeticiones de filas de cierto resultado. En cambio la sentencia ALL se usa cuando se quiere aparezcan todas las filas incluyendo las duplicadas.

#### **3.2.2.8. TOP**

Esta sentencia permite obtener las n primeras filas de la una tabla denominada origen, no escoge entre valores iguales, se basa por la columna de ordenación, puede ser la que obtenga en la sentencia ORDER BY o la clave principal de una tabla.

#### **3.2.2.9. Condiciones de Selección**

Estas condiciones aparecen en la sentencia WHERE.

- **COMPARACIÓN**

Permite comparar un valor de una sentencia con otro. Estos son:

=	igual que
<>	distinto que
>	mayor que
<	menor que
>=	mayor o igual
<=	menor o igual

Figura 3.2.2.9\_1: Sentencias Select

- **BETWEEN / NOT BETWEEN**

BETWEEN muestra los puntos extremos de un determinado rango, mientras que NOT BETWEEN comprueba si un valor se encuentra fuera del rango.

- **IN**

Comprueba si un valor de una expresión es uno de los valores incluidos en una determinada lista

- **IS NULL**

“Cuando una columna que interviene en una condición de selección contiene el valor nulo, el resultado de la condición no es verdadero ni falso, sino nulo, sea cual sea el test que se haya utilizado. Por eso si queremos listar las filas que tienen valor en una determinada columna, no podemos utilizar el test de comparación, la condición oficina = null devuelve el valor nulo sea cual sea el valor contenido en oficina. Si queremos preguntar si una columna contiene el valor nulo debemos utilizar un test especial, el test de valor nulo.”<sup>68</sup>

---

<sup>68</sup> [http://www.aulaclie.es/sql/t\\_2\\_4.htm](http://www.aulaclie.es/sql/t_2_4.htm)

- **LIKE / NOT LIKE**

La sentencia LIKE se usa cuando se va a utilizar comodines (caracteres) para la comparación.

Son considerados comodines:

- ? cualquier carácter
- ceros o más caracteres
- # dígito cualquiera (0-9)

### **3.2.3. Consulta sobre tablas relacionadas**

#### **3.2.3.1. JOIN**

Esta sentencia permite la combinación de registros de tablas en una base de datos relacional. Existen tres tipos de JOIN: interno, externo y cruzado.

- **Combinación Interna (INNER JOIN)**

Con esta sentencia se obtiene el producto cruzado de los registros, la regla a seguir es que los registros de la tabla 1 se combinan con los registros de la tabla 2, pero únicamente persisten los que satisfagan el contexto requerido, este tipo es el más usado. La sintaxis es la siguiente:

```
SELECT  
FROM tabla1 INNER JOIN tabla2 ON tabla1.col1-comp-tabla2.col2
```

La tabla1 y tabla2 define las tablas en las que se van a combinar sus registros.

Se denomina a columna1 y columna2 como las columnas de emparejamiento, las cuales deben tener el mismo tipo de dato.

Dentro de la cláusula ON las columnas deben tener nombres

cualificados es decir tienen delante de la tabla el nombre y un punto. El comp se refiere a la comparación (=, <, >, <=, >=, o <>) y se usa para formar la condición de igualdad.

Las condiciones de emparejamiento se pueden unir por operadores como AND y OR, las condiciones entre paréntesis.

#### ▪ LEFT JOIN Y RIGHT JOIN

La composición de tablas es diferente, se las conoce como composiciones externas; a diferencia del INNER JOIN y Producto Cartesiano que son internas ya que los valores de las filas resultado se encuentran en las tablas que se van a combinar.

La sintaxis del LEFT JOIN es:

```
FROM tabla1 LEFT JOIN tabla2 ON table.col1 comp tabla2.col2
```

Como se puede observar la sintaxis es la misma del INNER JOIN.

“Esta operación consiste en **añadir al resultado** del **INNER JOIN** las **filas** de la **tabla** de la **izquierda** que **no tienen correspondencia** en la otra tabla, y **rellenar** en esas filas los **campos** de la **tabla** de la **derecha** con **valores nulos**.”<sup>69</sup>

La sintaxis de RIGHT JOIN es:

```
FROM tabla1 RIGHT JOIN tabla2 ON tabla1.col1 comp tabla2.col2.
```

“Esta operación consiste en **añadir al resultado** del **INNER JOIN** las **filas** de la **tabla** de la **derecha** que **no tienen correspondencia** en la otra tabla, y **rellenar** en esas filas los **campos** de la **tabla** de la **izquierda** con **valores nulos**.”<sup>70</sup>

---

<sup>69</sup> [www.aulaclie.es/sql/t\\_3\\_5.htm](http://www.aulaclie.es/sql/t_3_5.htm)  
<sup>70</sup> [www.aulaclie.es/sql/t\\_3\\_5.htm](http://www.aulaclie.es/sql/t_3_5.htm)

“Una operación **LEFT JOIN** o **RIGHT JOIN** se puede anidar dentro de una operación **INNER JOIN**, pero una operación **INNER JOIN** no se puede **anidar dentro** de **LEFT JOIN** o **RIGHT JOIN**. Los anidamientos de **JOIN** de distinta naturaleza no funcionan siempre, a veces depende del orden en que se colocan las tablas, en estos casos lo mejor es probar y si no permite el anudamiento, cambiar el orden de las tablas (y por tanto de los **JOINS**) dentro de la cláusula **FROM**.”<sup>71</sup>

### **Combinación externa (OUTER JOIN)**

Esta sentencia no necesita que los registros de las tablas en las que se va a trabajar posean registros semejantes en la otra tabla. Se conserva el registro en la tabla combinada mientras no haya la relación con otro registro

### **Combinación completa (FULL OUTER JOIN)**

Mediante esta sentencia se obtiene resultados de tabla izquierda y derecha, aunque no posean relación en la otra tabla. La tabla que se combine contiene los registros de las dos tablas y el resultado serán valores nulos NULL para registros.

### **Ejemplos:**

Para realizar las consultas se van a usar las siguientes tablas:

---

<sup>71</sup> [www.aulaclie.es/sql/t\\_3\\_5.htm](http://www.aulaclie.es/sql/t_3_5.htm)

Tabla - dbo.Producto				
	Id_Producto	Detalle_Producto	Precio_Producto	Existencia_Pro...
	701	Libro A	15,85	100
	702	Libro B	20,35	207
	703	Libro C	10,12	36
	704	Libro D	29,8	12
	705	Libro E	11,65	104
	706	Libro F	8,05	59
	707	Libro G	5	10
	708	Libro H	6,12	13
	709	Libro I	40,89	2

Tabla 3.2.3.1\_1: TLBPRODUCTO

Tabla - dbo.Pedido						
	Id_Pedido	Fecha_Pedido	Nombre_Cliente	Producto_Pedido	Cantidad_Prod...	Fabricante_Pro...
	201	01/09/2011 0:0...	Ana Rosales	Libro A	80	Fabricante X
	202	02/09/2011 0:0...	Luis Borja	Libro D	10	Fabricante X
	203	01/09/2011 0:0...	Maria Aguilar	Libro E	100	Fabricante Y
	204	15/09/2011 0:0...	Ana Rosales	Libro B	200	Fabricante Y
	205	20/09/2011 0:0...	Veronica Haro	Libro F	50	Fabricante X
	206	08/09/2011 0:0...	Luis Borja	Libro A	10	Fabricante X

Tabla 3.2.3.1\_2: TLBPEDIDO

Tabla - dbo.Oficina						
	Id_Oficina	Ciudad_Oficina	Regional_Oficina	Direccion_Oficina	Indicador_Ventas	Ventas_Oficina
	601	Quito	Pichincha	Norte	30000	25000
	602	Ibarra	Imbabura	Norte	15000	10000
	603	Ambato	Tungurahua	Sur	18000	15000
	604	Guayaquil	Guayaas	Norte	40000	38000

Tabla 3.2.3.1\_3: OFICINA

Tabla - dbo.Empleado							
	Id_Empelado	Nombre_Emple...	Edad_Empleado	Fecha_Contrato	Jefe_Empleado	Cuota_Empleado	Ventas_Empleado
	101	Luis Miranda	30	01/09/2011 0:0...	501	5000	4000
	102	Marco Almeida	40	05/09/2011 0:0...	501	1500	1900
	103	Ivan Mejia	35	10/09/2011 0:0...	502	1000	800
	104	Maria Torres	40	12/09/2011 0:0...	501	500	1000
	105	Marcia Ruiz	31	20/09/2011 0:0...	503	1000	4000

Tabla 3.2.3.1\_4: TLBEMPLEADO

Tabla - dbo.Cliente				
	Id_Cliente	Nombre_Cliente	Codigo_Jefe	Maximo_Credito
	301	Ana Rosales	501	2000
	302	Carmen Aguas	502	1000
	303	Luis Borja	501	3000
	304	Maria Aguilar	503	1500
	305	Gustavo Llerena	502	3800
	306	Veronica Haro	503	5000
	307	Alberto Morales	503	500
	308	Lorena Moya	503	2700
	309	Edison Soria	502	1000

Tabla 3.2.3.1\_5: CLIENTE

Obtener por una consulta el código, cantidad de producto y fabricante para un pedido.

```
SELECT Id_Pedido, Cantidad_Producto, Fabricante_Producto
FROM Pedido
```

Realizar una consulta que permita obtener los pedidos ordenados por fecha de pedido

```
SELECT *
FROM Pedido
ORDER BY Fecha_Pedido
```

Realizar una consulta para obtener los pedidos del mes de Septiembre

```
SELECT *
FROM Pedido
WHERE MONTH(Fecha_Pedido) = 9
```

Obtener el código de las oficinas que no tiene indicador de ventas

```
SELECT Id_Oficina
FROM Oficina
WHERE Indicador_Ventas IS NULL
```

Obtener por medio de una consulta los datos de las oficinas de las regionales Pichincha y Guayas, primero deben aparecer las regionales de Pichincha

```
SELECT *
FROM Oficina
WHERE Regional_Oficina IN ('Pichincha','Guayas')
ORDER BY Regional_Oficina DESC
```

Obtener los empleados con nombre María

```
SELECT *
FROM Empleado
WHERE Nombre_Empleado LIKE 'Maria'
```

Obtener los códigos de empleado que tienen asignada una oficina en una ciudad

```
SELECT Id_Oficina
FROM Oficina
WHERE Ciudad_Oficina IS NULL
```

[Volver Inicio](#)



### U3. Lección 2 tarea 1

#### ACTIVIDAD DE APRENDIZAJE

---

Denominación de la tarea:

En relación a los siguientes datos, conteste la información correspondiente:

#### VIDEOS

Cod_video	fecha_prestamo	cod_cliente	cod_sala
1	02/01/2011	100	10
3	03/01/2011	110	10
11	10/01/2011	105	11
14	11/01/2011	105	12
15	12/01/2011	104	12
2	15/01/2011	100	11
7	21/02/2011	102	12
6	25/02/2011	100	11
4	28/02/2011	102	11
5	31/02/2011	110	11

#### VIDEO\_EXISTENTE

Cod_video	Idioma	Calificación	stock	clase
1	Español	Excelente	1	ACCION
3	Inglés	Buena	5	TERROR
11	Italiano	Muy Buena	3	DRAMA
14	Español	Excelente	1	DRAMA
15	Inglés	Buena	2	INFANTIL
2	Italiano	Muy Buena	5	ACCION
7	Español	Excelente	2	TERROR
6	Inglés	Excelente	3	TERROR
4	Inglés	Muy Buena	5	ACCION

## CLIENTE

Cod_cliente	cedula	dirección	edad	teléfono	Nombre
100	1778845390	Quito	40	2445897	María Jiménez
101	978567892	Ambato	23	2453897	Ángel Fierro
102	789673452	Ambato	17	2321452	Carlos Haro
103	1467834562	Quito	40	2999491	Luis Moreno
104	1256783768	Guayaquil	41	2678356	Matilde Mora
105	1278979890	Guayaquil	56	6978234	Lorena Abad
106	1789090129	Ibarra	15	2876324	Luis Baca
107	1890673873	Quito	43	2875634	Norma Núñez
108	1778889890	Ibarra	34	2897653	Juan López
109	1799898987	Ambato	38	2675789	José Tobar
110	189809899	Ambato	23	2876590	Pedro Torres

## SALA

Cod_sala	Ciudad	Teléfono
10	Quito	2575689
11	Quito	2786256
12	Ambato	2897686
13	Ibarra	3765678
15	Guayaquil	2876576

## ASESORÍA DIDÁCTICA

Para desarrollar la siguiente actividad de aprendizaje, debe leer y comprender el contenido de la Lección 3 Unidad 3.

## ACTIVIDAD DE APRENDIZAJE

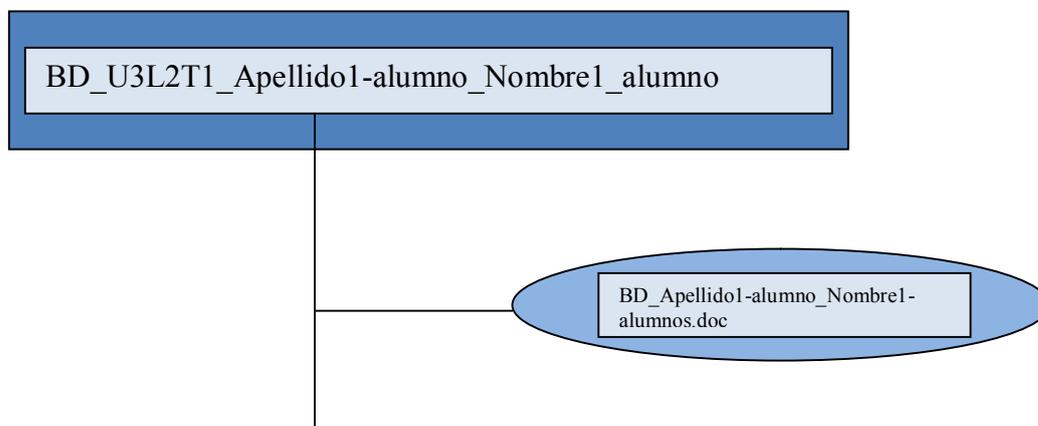
Realizar las siguientes consultas SQL:

1. Cuál es el cliente que más videos ha rentado?

2. Listar todos los videos de acción que se rentaron en febrero en la sala de Ambato por el cliente 100?
3. Cuántos clientes rentaron películas con calificación EXCELENTE en el último trimestre?
4. Que películas en idioma Inglés son de Terror?
5. Que sala de video renta más películas de Acción?

Envíe a su tutor la tarea en una carpeta comprimida como .rar

Formato de la carpeta:





## Resumen Lección 2

---

### Consultas sobre tablas y tablas relacionadas

Esta lección ha presentado una visión global de base de datos, incluyendo:

- Consulta sobre tablas
  - SELECT
  - FROM
  - WHERE
  - GROUP BY
  - HAVING
  - ORDER BY
- Consulta sobre tablas relacionadas
  - JOIN

#### **Destacándose que:**

La sentencia `SELECT` describe los parámetros de datos a extraer en una consulta.

A la condición `FROM` le sigue una descripción de las tablas y éstas se encuentran separadas por comas.

La condición `WHERE` se utiliza para definir las filas que vayan a recuperarse. Se selecciona sólo las filas que cumplan la condición esta condición.

A la condición `GROUP BY` se le conoce como función agrupada ya que reúne los datos de las tablas origen y se obtiene una sola fila de cada grupo de filas.

La condición HAVING indica a SQL que se agregue a determinados grupos que se obtuvieron como resultado de la sentencia GROUP BY.

La condición ORDER BY ordena ascendente o descendentemente lo producido por una consulta en una base de datos, se basa en los valores de una o varias columnas.

Esta sentencia permite la combinación de registros de tablas en una base de datos relacional. Existen tres tipos de JOIN: interno, externo y cruzado.

## UNIDAD 3: SQL LENGUAJE DE CONSULTA RELACIONAL

### LECCIÓN 3.3: Funciones de Agregación



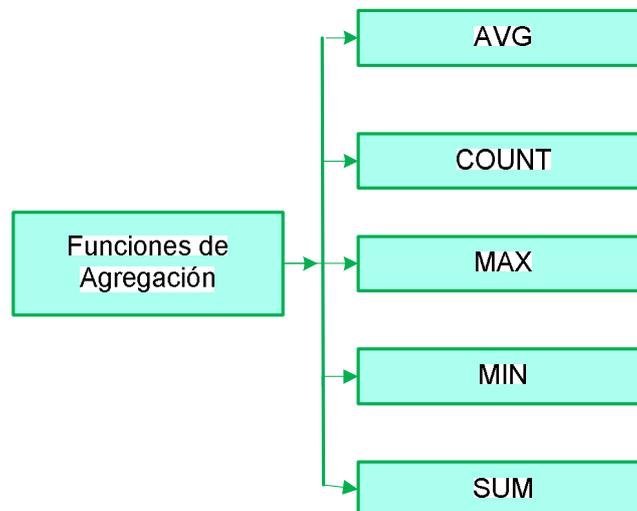
#### Visión General

En esta lección se tratan las funciones de agregación dentro del SQL.

[Volver Inicio](#)



#### Plan de la Lección



#### Desarrollo de la Lección

##### 3.3.1. Introducción

Las funciones de agregación trabajan sobre la colección de filas de una tabla, para obtener en un SELECT el resultado adecuado.

### **3.3.2. AVG**

Esta función permite obtener en una determinada columna el promedio de valores. La columna debe tener los mismos tipos de datos numéricos, el resultado de esta función puede ser un tipo de dato diferente al de la columna ya que lo que hace la función AVG es sumar los valores de una determinada columna y dividirlo por el número de valores.

### **3.3.3. COUNT**

Esta función de agregación cuenta el número de valores de datos que hay en una determinada columna. La columna puede tener datos de cualquier tipo, pero esta función siempre devuelve un entero no depende de los datos de la columna.

Ejemplo:

```
SELECT COUNT (numalun)
FROM alumno
```

También:

```
SELECT COUNT (*)
FROM alumno
```

### **3.3.4. MAX**

Esta función de agregación ubica en una determinada columna el valor máximo. En la columna puede haber datos tipo numérico, string, fecha/hora, pero el resultado debe conservar igual tipo de dato.

### **3.3.5. MIN**

Esta función ubica en una determinada columna el valor mínimo. En la columna puede haber datos tipo numérico, string, fecha/hora, pero el resultado debe conservar igual tipo de dato.

### 3.3.6. SUM

Esta función de agregación permite obtener el valor total de una determinada columna. En la columna debe haber datos del mismo tipo numérico (entero, decimal, etc), el resultado debe conservar igualmente el mismo tipo.

Ejemplo:

```
SELECT SUM(notas)
FROM alumno
```

#### Ejemplos:

Usando las mismas tablas de la Lección 2 Unidad 3 realizar las siguientes consultas:

Tabla - dbo.Producto				
	Id_Producto	Detalle_Producto	Precio_Producto	Existencia_Pro...
	701	Libro A	15,85	100
	702	Libro B	20,35	207
	703	Libro C	10,12	36
	704	Libro D	29,8	12
	705	Libro E	11,65	104
	706	Libro F	8,05	59
	707	Libro G	5	10
	708	Libro H	6,12	13
	709	Libro I	40,89	2

Tabla: 3.3.6\_1: TLBPRODUCTO

Tabla - dbo.Pedido						
	Id_Pedido	Fecha_Pedido	Nombre_Cliente	Producto_Pedido	Cantidad_Prod...	Fabricante_Pro...
	201	01/09/2011 0:0...	Ana Rosales	Libro A	80	Fabricante X
	202	02/09/2011 0:0...	Luis Borja	Libro D	10	Fabricante X
	203	01/09/2011 0:0...	Maria Aguilar	Libro E	100	Fabricante Y
	204	15/09/2011 0:0...	Ana Rosales	Libro B	200	Fabricante Y
	205	20/09/2011 0:0...	Veronica Haro	Libro F	50	Fabricante X
	206	08/09/2011 0:0...	Luis Borja	Libro A	10	Fabricante X

Tabla 3.3.6\_2: TLBPEDIDO

Tabla - dbo.Oficina						
	Id_Oficina	Ciudad_Oficina	Regional_Oficina	Direccion_Oficina	Indicador_Ventas	Ventas_Oficina
	601	Quito	Pichincha	Norte	30000	25000
	602	Ibarra	Imbabura	Norte	15000	10000
	603	Ambato	Tungurahua	Sur	18000	15000
	604	Guayaquil	Guayaas	Norte	40000	38000

Tabla 3.3.6\_3: TLBOFICINA

Tabla - dbo.Empleado							
	Id_Empelado	Nombre_Emple...	Edad_Empleado	Fecha_Contrato	Jefe_Empleado	Cuota_Empleado	Ventas_Empleado
	101	Luis Miranda	30	01/09/2011 0:0...	501	5000	4000
	102	Marco Almeida	40	05/09/2011 0:0...	501	1500	1900
	103	Ivan Mejia	35	10/09/2011 0:0...	502	1000	800
	104	Maria Torres	40	12/09/2011 0:0...	501	500	1000
	105	Marcia Ruiz	31	20/09/2011 0:0...	503	1000	4000

Tabla 3.3.6\_4: EMPLEADO

Tabla - dbo.Cliente				
	Id_Cliente	Nombre_Cliente	Codigo_Jefe	Maximo_Credito
	301	Ana Rosales	501	2000
	302	Carmen Aguas	502	1000
	303	Luis Borja	501	3000
	304	Maria Aguilar	503	1500
	305	Gustavo Llerena	502	3800
	306	Veronica Haro	503	5000
	307	Alberto Morales	503	500
	308	Lorena Moya	503	2700
	309	Edison Soria	502	1000

Tabla 3.3.6\_5: TLBCLIENTE

Obtener la media de la cuota y ventas de cada empleado

```
SELECT      AVG(Cuota_Empleado)      AS      Cuota_Media,
AVG(Ventas_Empleado) AS Ventas_Media
FROM Empleado
```

Obtener el la media de la cantidad de producto del pedido para el fabricante denominado X

```
SELECT AVG(Cantidad_Producto) AS p_medio_Fabricante
FROM Pedido
WHERE Fabricante_Producto = 'Fabricante X'
```

Determinar la fecha en la que se realizó el primer pedido

```
SELECT MIN(Fecha_Pedido) AS Primer_Pedido
FROM Pedido
```

Obtener cuántos pedido son mayores en su cantidad de producto

```
SELECT COUNT(*)
FROM Pedido
WHERE Cantidad_Producto > 50
```

Obtener las ventas de empleado de los pedidos realizados por el Empleado Marcas Ruiz

```
SELECT SUM(Ventas_Empleado)
FROM Empleado INNER JOIN Pedido ON Empleado.Cuota_Empleado =
Pedido.Id_Pedido
WHERE Nombre_Empleado = 'Marcia Ruiz'
```

[Volver Inicio](#)



### U3. Lección 3 tarea 1

#### **ACTIVIDAD DE APRENDIZAJE**

---

Denominación de la tarea:

Crear las siguientes tablas:

PRESTAMO(codigo\_agencia, codigo\_cliente, fecha\_prestamo, valor\_otorgado)

AGENCIA(codigo\_agencia, nombre\_agencia, dirección, telefono)

CLIENTE(codigo\_cliente, nombre, cedula, dirección, telefono, edad)

TIPO\_PRESTAMO(codigo\_tipo, descripcion)

#### **ASESORÍA DIDÁCTICA**

Para desarrollar la siguiente actividad de aprendizaje, debe leer y comprender el contenido de la Lección 3 Unidad 3.

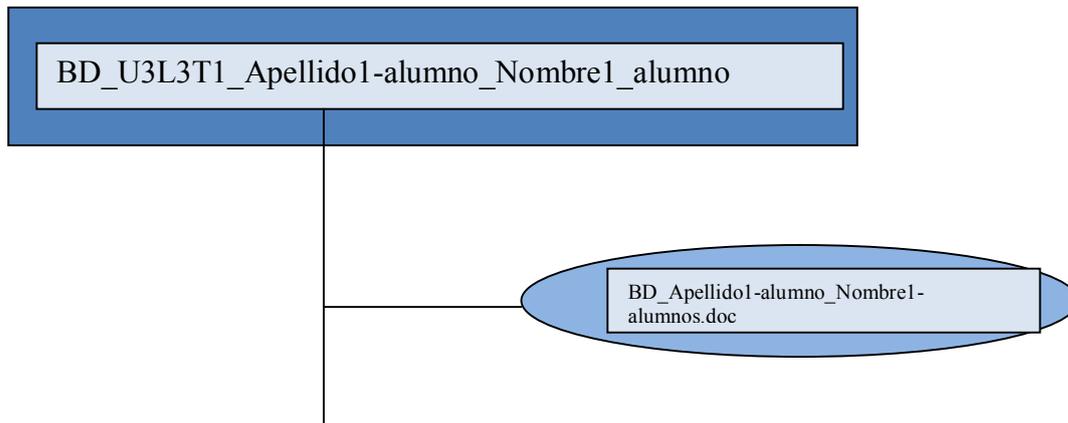
#### **ACTIVIDAD DE APRENDIZAJE**

Realizar las siguientes consultas:

1. Cuál es la agencia que otorgo el monto menor de préstamos.
2. En qué fecha y qué agencia otorgo mayor cantidad de préstamos.
3. Qué clientes tienen otorgado un préstamo en la misma agencia que se dio por primera vez un crédito al cliente cuyo código es 1122.
4. Qué clientes obtuvieron créditos en la misma fecha el cliente 1122
5. A qué clientes se des otorgó un crédito cuyo monos supera el que se le otorgó al cliente 112

Envíe a su tutor la tarea en una carpeta comprimida como .rar

Formato de la carpeta:





### U3. Lección 3 tarea 2

#### ACTIVIDAD DE APRENDIZAJE

---

Denominación de la tarea:

De las siguientes tablas creadas en la Tarea 1 de la Lección 9 Unidad 2 realizar las siguientes consultas:

EMPLEADO(codigo\_empleado, cedula, nombre, dirección, departamento, cargo)

SALARIOS (codigo\_empleado, fecha\_pago, sueldo, descuentos, cargo)

DEPARTAMENTO(codigo\_empleado, nombre\_departamento, dirección)

CARGO(codigo\_cargo, detalle)

#### **ASESORÍA DIDÁCTICA**

Para desarrollar la siguiente actividad de aprendizaje, debe leer y comprender el contenido de la Lección 1 Unidad 3.

#### **ACTIVIDAD DE APRENDIZAJE**

Realizar las siguientes consultas SQL:

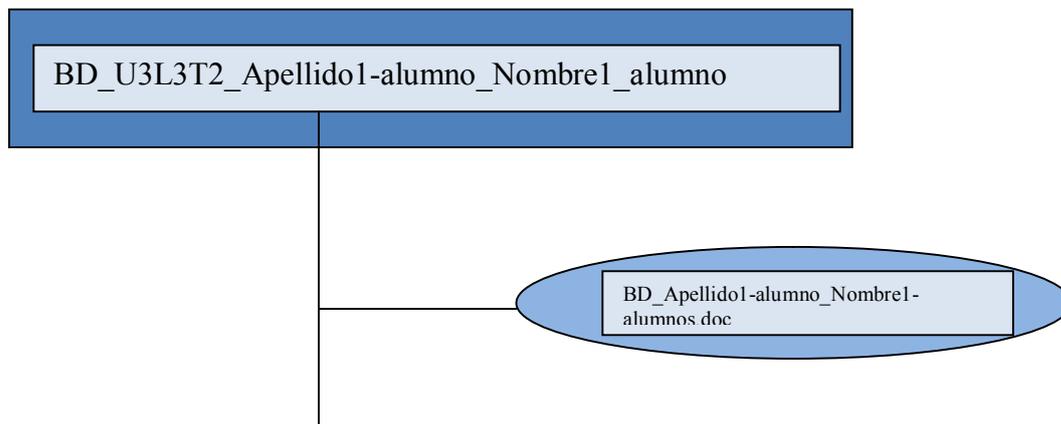
1. Listar que departamento pagó en su rol más que los otros departamentos.
2. Listar a que empleado se le canceló el mayor monto de salarios en el periodo 2010.

3. Listar a que empleado se le cancelo el menor monto de salarios en el periodo 2010.

4. Listar a que empleado se le descontó más en su rol en el período 2010

Envié a su tutor la tarea en una carpeta comprimida como .rar

Formato de la carpeta:





## Resumen Lección 3

---

### Funciones de Agregación

Esta lección ha presentado una visión global de base de datos, incluyendo:

- **AVG**
- **COUNT**
- **MAX**
- **MIN**
- **SUM**

#### **Destacándose que:**

La función AVG permite obtener en una determinada columna el promedio de valores.

La función COUNT cuenta el número de valores de datos que hay en una determinada columna.

La función SUM ubica en una determinada columna el valor máximo.

La función MIN ubica en una determinada columna el valor mínimo

La función MAX permite obtener el valor total de una determinada columna.

## **CAPÍTULO 5**

### **CONCLUSIONES Y RECOMENDACIONES**

#### **5.1. Conclusiones**

- 5.1.1.** Cada vez más se vuelve apremiante que los estudiantes de Ingeniería de Sistemas e Informática adquieran experiencia en el mundo real, por lo que se ven en la necesidad de trabajar desde los niveles inferiores de la Carrera, esto dificulta el asistir constantemente a la Universidad, los cursos virtuales se convierten en una ayuda para resolver este inconveniente.
- 5.1.2.** El curso se estructuró desarrollando objetos de aprendizaje para cumplir las competencias definidas para el curso, muchos de estos objetos de aprendizaje pueden ser requeridos y utilizados en otros cursos.
- 5.1.3.** El curso terminado y puesto en producción debe ser evaluado con alumnos y profesores de un curso presencial, esto permitirá realizar ciertos ajustes y se realizará una actualización constante de los contenidos.
- 5.1.4.** Durante el desarrollo del curso existieron cambios en la malla curricular lo que hizo que se prolongue el tiempo estimado, dado el dinamismo de las asignaturas de Sistemas e Informática esto se seguirá produciendo.

## **5.2. Recomendaciones**

- 5.2.1.** Continuar con el desarrollo de cursos virtuales para las asignaturas de los últimos niveles de Ingeniería y Postgrado.
- 5.2.2.** Para el desarrollo de nuevos cursos se debe aplicar otros estándares como SCORM y software como Exe Learning que permiten levantar en Moodle los cursos virtuales a través de los objetos de aprendizaje.
- 5.2.3.** El profesor titular de la asignatura debe actualizar constantemente los temas y actividades de aprendizaje.
- 5.2.4.** Se debe desarrollar un repositorio a nivel universitario para objetos de aprendizaje.
- 5.2.5.** Se debe resolver el problema de comunicación automática a los alumnos, cuando se levanten actividades de aprendizaje para que el proceso enseñanza – aprendizaje se vuelva más dinámico.

## BIBLIOGRAFÍA

Garcés Elena y Rivera Enríquez Cecilia. (2006). Evaluación de plataformas tecnológicas para la teleformación o e-learning para el ámbito universitario, tomando como caso de estudio e-educativa. Quito: Escuela Politécnica del Ejército.

[www.slideshare.net/wiesco/teorias-del-aprendizaje-312461](http://www.slideshare.net/wiesco/teorias-del-aprendizaje-312461)

[http://www.slideshare.net/plazaa\\_10\\_10/cognitivismo](http://www.slideshare.net/plazaa_10_10/cognitivismo)

TeoríasdeAprendizaje.ppt/ Dilayla Terraza

[http://ares.unimet.edu.ve/programacion/psfase3/modII/biblio/CONDUCTISMO\\_%20COGNITIVISMO\\_%20CONSTRUCTIVISMO.pdf](http://ares.unimet.edu.ve/programacion/psfase3/modII/biblio/CONDUCTISMO_%20COGNITIVISMO_%20CONSTRUCTIVISMO.pdf)

[http://ares.unimet.edu.ve/programacion/psfase3/modII/biblio/CONDUCTISMO\\_%20COGNITIVISMO\\_%20CONSTRUCTIVISMO.pdf](http://ares.unimet.edu.ve/programacion/psfase3/modII/biblio/CONDUCTISMO_%20COGNITIVISMO_%20CONSTRUCTIVISMO.pdf)

[http://ares.unimet.edu.ve/programacion/psfase3/modII/biblio/CONDUCTISMO\\_%20COGNITIVISMO\\_%20CONSTRUCTIVISMO.pdf](http://ares.unimet.edu.ve/programacion/psfase3/modII/biblio/CONDUCTISMO_%20COGNITIVISMO_%20CONSTRUCTIVISMO.pdf)

[www.slideshare.net/edgar\\_rg/diseo-instruccional-1737789](http://www.slideshare.net/edgar_rg/diseo-instruccional-1737789)

[http://nuria.worldhostsoft.com/demos/MII-L1\\_.html](http://nuria.worldhostsoft.com/demos/MII-L1_.html)

Hermosa Edgar (2009). Introducción a la Metodología para Diseñar Cursos Virtuales. Quito: Escuela Politécnica del Ejército.

<http://ateneo-empresarial.com/archives/category/diseo-instruccional>

<http://www.miro.cl/duocuc/estpeda.htm>

Carrera Alejandro y Del Castillo Gabriela. (2008). Tesis de Grado Curso E-Learning de Ingeniería de Software I. Quito: Escuela Politécnica del Ejército.

Hermosa Andrea y Chica Fernanda. (2008). Tesis de Grado Curso E-Learning de Planificación Informática. Quito: Escuela Politécnica del Ejército.

Connolly, Thomas M. y Begg, Carolyn E. (2005). Sistemas de bases de datos - Un enfoque práctico para diseño, implementación y gestión, Madrid: Pearson Addison Wesley.

Kendall, Julie. (2005). Análisis y Diseño de Sistemas.

Date, C.J. (2006). Introducción a los Sistemas de Bases de Datos.

<http://grimvidev.wordpress.com/2011/01/18/lonuevoensqlserver2011/>

Piattini Miguel (1999). Fundamentos y Modelos de Base de Datos. Ra-Ma

Rob, Peter y Coronel, Carlos. (2006). Sistemas de bases de datos: diseño, implementación y administración. España: Thomson.

Sabana, Maribel. (2006). Modelamiento e Implementación de Base de Datos: Lima: Megabyte Grupo Editorial.

De Miguel, Adoración; Piattini, Mario; Marcos, Esperanza. (2000). Diseño de Bases de Datos Relacionales, Colombia: Alfaomega S.A.

[http://www.unalmed.edu.co/~mstabare/disenio\\_conceptual.htm](http://www.unalmed.edu.co/~mstabare/disenio_conceptual.htm)

De Miguel, Adoración; Piattini, Mario. (1999). Fundamentos y modelo de Bases de Datos, México: Alfaomega S.A.

[epcc.unex.es/modules.php?op=modload&name=Downloads](http://epcc.unex.es/modules.php?op=modload&name=Downloads)

[http://www.aulaclie.es/sql/t\\_8\\_2.htm](http://www.aulaclie.es/sql/t_8_2.htm)

[www.juntadeandalucia.es/averroes/.../sqbd\\_EjerciciosResueltos\\_Rep1.pdf](http://www.juntadeandalucia.es/averroes/.../sqbd_EjerciciosResueltos_Rep1.pdf)

Piattini, Mario; Marcos, Esperanza; Calero, Coral y Vela, Belén. (2006). Tecnología y Diseño de Base de Datos. México: Alfaomega

Ramez, Elmasri. y Shamkant, Navathe. (2002). Fundamentos de Sistemas de Bases de Datos, Madrid: Pearson Addison Wesley.

Ing. Díaz Oswaldo (2012), Escuela Politécnica del Ejército

[http://www.aulaclie.es/sql/t\\_8\\_1.htm](http://www.aulaclie.es/sql/t_8_1.htm)

[http://www.aulaclie.es/sql/t\\_8\\_2.htm](http://www.aulaclie.es/sql/t_8_2.htm)