

**MÓDULO PROFESIONAL DE
SISTEMAS GESTORES / XESTORES DE
BASES DE DATOS**

Familia Profesional: Informática

Ciclo Superior de Administración de Sistemas Informáticos (CS ASI)

4 y 5 Trimestres

Profesor: Cesáreo García Rodicio

Fecha: Septiembre 2005

Índice de contenido

1 INTRODUCCIÓN.....	5
2 CONTEXTUALIZACIÓN.....	6
3 CAPACIDADES TERMINALES.....	8
4 REQUISITOS PREVIOS.....	11
5 CRITERIOS DE SECUENCIACIÓN DE LOS CONTENIDOS.....	12
6 ELEMENTOS CURRICULARES DE CADA UNIDAD DE TRABAJO.....	15
<i>Unidad Didáctica 1 Uso de Internet en el Desarrollo: documentación y búsqueda</i>	<i>15</i>
1.1 Contenidos Conceptuales.....	15
1.2 Contenidos Procedimentales.....	17
1.3 Criterios de Evaluación.....	18
<i>Unidad Didáctica 2 Introducción a las Bases de Datos.....</i>	<i>18</i>
2.1 Contenidos Conceptuales.....	18
2.2 Contenidos Procedimentales.....	21
2.3 Criterios de Evaluación.....	21
<i>Unidad Didáctica 3 Modelo Relacional y el Modelo E/R.....</i>	<i>22</i>
3.1 Contenidos Conceptuales.....	22
3.2 Contenidos Procedimentales.....	24
3.3 Criterios de Evaluación.....	26
<i>Unidad Didáctica 4 Crear Modelos de Datos (SQL I).....</i>	<i>26</i>
4.1 Contenidos Conceptuales.....	27
4.2 Contenidos Procedimentales.....	28

<i>Unidad Didáctica 5 Utilizar el Modelo de Datos (SQL II)</i>	30
5.1 Contenidos Conceptuales.....	30
5.2 Contenidos Procedimentales.....	32
5.3 Criterios de Evaluación.....	33
<i>Unidad Didáctica 6 El Servidor SGBDR (PostgreSQL)</i>	33
6.1 Contenidos Conceptuales.....	33
6.2 Contenidos Procedimentales.....	35
6.3 Criterios de Evaluación.....	35
<i>Unidad Didáctica 7</i>	35
7.1 Contenidos Conceptuales.....	36
7.2 Contenidos Procedimentales.....	38
7.3 Criterios de Evaluación.....	40
<i>Unidad Didáctica 8</i>	40
8.1 Contenidos Conceptuales.....	40
8.2 Contenidos Procedimentales.....	41
8.3 Criterios de Evaluación.....	42
<i>Unidad Didáctica 9 Análisis, Diseño e Implantación de un proyecto Real (Aplicación Web)</i>	42
9.1 Contenidos Conceptuales.....	43
9.2 Contenidos Procedimentales.....	44
9.3 Criterios de Evaluación.....	44
7 CONTENIDOS ACTITUDINALES	44
8 METODOLOGÍA	45
9 MEDIDAS DE ATENCIÓN A LA DIVERSIDAD	47
10 EVALUACIÓN	49

10.1 PROCEDIMIENTO DE EVALUACIÓN ORDINARIO	49
10.2 CRITERIOS Y PROCEDIMIENTOS PARA ESTABLECER LAS ACTIVIDADES DE RECUPERACIÓN DE LOS MÓDULOS PENDIENTES.....	52
10.3 PROCEDIMIENTOS DE EVALUACIÓN Y CUALIFICACIÓN EXTRAORDINARIA DEL ALUMNADO CON MÓDULOS PENDIENTES.....	53
10.4 CONTENIDOS MÍNIMOS EXIGIBLES.....	53
11 BIBLIOGRAFÍA.....	54

1 Introducción

Esta programación está referida al módulo profesional de Sistemas Gestores de Bases de Datos (SXBD¹) que se cursa en el 2º año del ciclo formativo de grado superior de Administración de Sistemas Informáticos, concretamente en el cuarto y quinto trimestres. Este ciclo de grado superior tiene una duración total de 2000 horas (equivalente a 5 trimestres de formación en centro educativo como máximo, más la formación en centro de trabajo correspondiente).

La programación didáctica se desarrolla siguiendo las prescripciones del currículo publicado en el diario oficial de Galicia (DOGA) que reúne los siguientes aspectos:

- Contextualización
- Objetivos.
- Contenidos y secuenciación.
- Metodología.
- Criterios y procedimientos de evaluación.
- Atención a los alumnos con necesidades educativas específicas.
- Bibliografía de aula y de departamento.

Esta programación didáctica para un módulo forma parte de la programación general o proyecto curricular de ciclo, que los profesores del Departamento de Informática elaboran

¹ En gallego, idioma cooficial de mi comunidad autónoma, la traducción es Sistemas Xestores de Bases de Datos, por eso empleo las siglas SXBD.

con el fin de dar unidad y coherencia a las enseñanzas de los módulos del ciclo formativo, teniendo en cuenta las directrices establecidas en el Proyecto Curricular del Centro.

El Proyecto Curricular del ciclo formativo constituye el instrumento pedagógico-didáctico que articula a largo plazo el conjunto de actuaciones educativas del centro y tiene como objetivo alcanzar las finalidades educativas del mismo. Según la propuesta en el DCB este módulo consta de 185 horas.

A partir del análisis del currículo, se han elaborado una serie de unidades de trabajo que han sido organizadas y secuenciadas en torno a supuestos prácticos, en base a los cuales se justifica el estudio de los contenidos básicos permitiendo relacionarlos con su aplicación práctica. Estos contenidos determinan las etapas más significativas de los procedimientos que se van a desarrollar (presentación, análisis y aplicación) en función de las variables diferenciadoras elegidas.

Por último, se han definido los **elementos curriculares** de cada unidad de trabajo, es decir, los objetivos que deben alcanzarse, los contenidos que deben desarrollarse, las actividades del proceso de enseñanza-aprendizaje que deben realizarse, la metodología a emplear y los criterios de evaluación que permitan medir el nivel de aprendizaje obtenido.

2 Contextualización

El diseño curricular del módulo pertenece al último nivel de concreción del Curriculum. Partiendo del Diseño Base del Título de Formación Profesional, que recordemos es Técnico Superior en Administración de Sistemas Informáticos, se define el ciclo dentro del

Plan Educativo del Centro y el Proyecto Curricular de Centro.

Vamos a contextualizar la programación en un centro educativo en zona urbana, bien comunicada y de un tamaño medio. Tiene los dos ciclos superiores de Informática, un ciclo superior de Administración y Finanzas y dos ciclos medios de Comercio. Además tiene programas de Garantía Social. La implantación de los ciclos superiores de informática son de reciente creación (dos promociones de técnicos superiores).

El centro pretende imprimir el carácter innovador, de educación “a medida” y adecuada al contexto laboral que se pretende imbuir en la “nueva formación profesional”. El centro pertenece a una ciudad urbana e industrial (Vigo) con presencia de empresas dedicadas al sector servicios informáticos de desarrollo de software y soporte técnico a infraestructuras informáticas. Las empresas demandan especialmente alumnos con formación en la arquitectura de Internet y administración de sistemas informáticos Windows y Linux.

En el tercer nivel de concreción curricular el ciclo formativo se coordinará con el otro ciclo de informática. El departamento de Informática ha decidido que el ciclo de DAI será por la mañana y el de ASI por la tarde, en grupos de máximo 30 alumnos y con al menos un ordenador por cada dos alumnos. Los locales asignados al ciclo cumplen en cuanto a instalaciones y dotación, los requisitos establecidos en el D.C.B., en el epígrafe 4.2. Requisitos mínimos de espacios e instalaciones para impartir estas enseñanzas.

Este módulo por lo tanto, se dará en horario de tarde lo que significa que los alumnos, en general, serán adultos y también que probablemente ya trabajen y algunos tengan problemas para asistir a clase.

Dentro de la contextualización en el centro y el ciclo formativo se ha visto que el uso de Internet como plataforma integral en la administración de sistemas informáticos es necesario en el alumno. Por lo tanto, una particularización del curriculum será centrarse en la plataforma Internet-Web como soporte para el desarrollo de nuevas funcionalidades, soporte a los usuarios y documentación-colaboración del desarrollo.

Además el centro quiere buscar en el uso de las nuevas tecnologías de la información, ,especialmente con la colaboración a través de Internet, una formación personalizada y adaptada a cada alumno.

3 Capacidades Terminales

Este módulo está asociado a varias capacidades terminales al ser un módulo transversal.

Tiene como capacidades terminales elementales:

- Analizar el tipo de información para registrar en función de las consultas o búsquedas que se vayan a realizar y las posibles aplicaciones que tendrán acceso a la base de datos.
- Obtener diseños de la estructura de las bases de datos, definiendo los conjuntos de entidades así como los conjuntos de relaciones.
- Representar gráficamente la estructura lógica global de las bases de datos mediante diagramas entidad-relación
- Transformar los diagramas entidad-relación en esquemas conceptuales definiendo

el conjunto de esquemas de relación.

- Aplicar procedimientos de normalización que minimicen las redundancias
- Crear el conjunto de tablas de información así como la relación de índices que permitan accesos óptimos en la recuperación de la información de las bases de datos.
- Obtener diseños útiles de formatos personalizados que faciliten la introducción y modificación de los datos y que permitan su validación.
- Diseñar y almacenar las vistas para facilitar las consultas individualizadas así como la seguridad de las bases de datos
- Consultar y actualizar las bases de datos mediante las utilidades que proporciona el SGBD Relacional.
- Manejar las herramientas disponibles en los SGBD para facilitar la recuperación y presentación de la información. Generadores de informes personalizados y Menús
- Realizar transferencias de datos entre la base de datos y programas software independientes, utilizando adecuadamente los mandatos del sistema gestor de la base de datos.
- Aplicar correctamente los mandatos y procedimientos del lenguaje SQL
- Analizar la lista de dependencias funcionales para los distintos esquemas de relación que permitan asegurar la bondad del diseño de la base de datos, así como probar que su posterior actualización no creará relaciones ilegales.
- Establecer las restricciones de dominio y de integridad referencial en la definición de tablas, asegurando que los cambios autorizados no lleven a una pérdida de

consistencia de la base de datos.

- Definir y establecer restricciones de integridad más complejas mediante el uso de afirmaciones.
- Implementar procedimientos disparadores que permitan al SGBD realizar acciones complejas ante determinadas actualizaciones autorizadas de la base de datos.
- Proteger las bases de datos otorgando distintas autorizaciones o privilegios sobre las vistas y otros objetos (tablas, formularios, programas, aplicaciones) de ellas.
- Describir procedimientos que permitan bloqueos en el caso de accesos concurrentes a las bases de datos.
- Planificar procedimientos que garanticen la seguridad física de la base de datos.
- Aplicar procedimientos para optimizar el rendimiento de la base de datos.

Resumiendo y sintetizando todas las capacidades terminales que iremos evaluando, tendríamos como objetivos del módulo:

- **Entender a un cliente** (usuario final) sobre una necesidad concreta
- **Documentar** su problema
- Diseñar el **Modelo de Datos** (SQL)
- Definir las **consultas y/o procedimientos** sobre los datos
- **Explotar una BD** (Instalarla, Configurarla y Mantenerla)

El contenido de este módulo es sobre todo de tipo procedimental y el alumno deberá adquirir una serie de destrezas y habilidades en el manejo y administración del sistema. Pretendemos que el módulo sea una propuesta interesante y actual para el alumno.

Los contenidos básicos de tipo actitudinal irán orientados a fomentar la autonomía, la resolución autónoma de problemas de diseño de datos y la necesidad de una formación continua como síntesis de los descritos en el Diseño Curricular Base.

4 Requisitos Previos

El alumno tendrá la capacitación prevista en los módulos de los primeros tres trimestres del ciclo. Estas competencias están descritas en los módulos correspondientes y resumiéndolas podríamos definir los siguientes:

- Administración y Gestión de la conectividad de Red, especialmente una TCP/IP
- Administración y Gestión Básica/Media de un sistema operativo multiusuario y multitarea
- Conocimientos básicos de lógica y programación de estructuras de datos.
- Conocimiento medio de un editor de texto avanzado

Además son de interés todos los contenidos conceptuales, procedimentales y actitudinales de todas las asignaturas de primer curso. Se prestará atención a los alumnos que tienen módulos pendientes o con notas iguales o inferiores al 6.

Se hará una evaluación inicial del perfil de la clase, para definir correctamente el contexto del grupo. Al ser el segundo curso del ciclo es más fácil que el grupo sea más homogéneo en conocimientos, sin embargo no debemos olvidar que en áreas como la informática existe un alto nivel de alumnos con formación autodidacta y que incluso en cursos de

verano han podido alcanzar algunos objetivos del módulo.

5 CRITERIOS DE SECUENCIACIÓN DE LOS CONTENIDOS

Los criterios generales de secuenciación adoptados son:

- Identificar los principales componentes de los contenidos seleccionados en el primer nivel de concreción (DCB –Diseño Curricular Base-)
- Analizar las relaciones que existen entre los componentes identificados y establecer las estructuras de contenido correspondientes
- Proponer una secuenciación de los componentes atendiendo a las relaciones y estructuras establecidas y a las leyes del aprendizaje significativo

Los contenidos se organizan en varios bloques distribuidos secuencialmente. Las unidades didácticas se pueden englobar en cuatro grandes bloques y una unidad didáctica (UD) de transición:

- Introducción a la Base de Datos y diseño (UDs 1,2 y 3)
- Diseño de modelos de datos y su uso con SQL (UDs 4 y 5)
- Administración de un servidor de BD relacional (UD 6)
- Desarrollo de un mini proyecto (UD 7)

Esta distribución en bloques permite una comprensión general del módulo, pero nos centraremos en una programación por unidad didáctica pues permite un mejor control y

evaluación.

Aunque hemos hecho la distribución por horas, en la práctica una clase es un conjunto de sesiones de 50 minutos, generalmente dos o tres sesiones. En realidad es más útil el concepto de clase para el docente, pues para el alumno englobaría el concepto de “lo que voy a ver en la clase de hoy”. Obviamente sólo la experiencia permite determinar exactamente cuales son los contenidos para cada clase, sin embargo es imprescindible tener una programación secuencial según el número de horas teniendo en cuenta las clases a impartir.

En realidad un trimestre consta aproximadamente de unas 12 semanas. Por lo tanto el módulo tiene una duración aproximada de 24 a 28 semanas. Según el diseño curricular base del ciclo el módulo consta de 185 horas en total. Esto significa 8 sesiones de 50 minutos, y según considere el centro se necesitarán más o menos semanas para dar el módulo. Las unidades didácticas serán impartidas siguiendo la secuencia siguiente:

UD	TÍTULO	Semanas	Clases	50 min
1	Uso de Internet en el Desarrollo: documentación y búsqueda	2	4	16
2	Introducción a las Bases de Datos	2	4	16
3	Modelo Relacional y el Modelo E/R	3	6	24
4	Crear Modelos de Datos (SQL I)	4	8	32
5	Utilizar el Modelo de Datos (SQL II)	4	8	32
6	El Servidor SGBDR (PostgreSQL)	4	8	32
7	Análisis, Diseño e Implantación de un proyecto	4	8	70

UD	TÍTULO	Semanas	Clases	50 min
	Real (Aplicación Web)			
		185	23	46 222

El módulo está dividido en dos evaluaciones en los dos primeros trimestres del segundo curso del Ciclo Superior, es decir el cuarto y quinto trimestre del ciclo. La secuenciación de las unidades didácticas debería permitir comenzar la segunda evaluación con el trabajo de desarrollo propiamente dicho, esto significaría que en el cuarto cuatrimestre se comenzaría con la UD 5.

Esta distribución es meramente orientativa, dado el carácter abierto y flexible del currículo, y en función de las necesidades del grupo podrán producirse pequeños ajustes.

6 Elementos Curriculares de cada Unidad de Trabajo

Unidad Didáctica 1 Uso de Internet en el Desarrollo: documentación y búsqueda

En esta unidad didáctica es la primera del módulo, esto significa que los alumnos vienen de su descanso vacacional y que se debe empezar con contenidos más prácticos para

introducirlos en el ritmo del trimestre. Esta unidad continuará con el trabajo de los módulos anteriores para el uso de Internet en clase, haciendo hincapié en el uso de Internet para el desarrollo de software, la documentación y la ayuda en línea.

En todo el módulo el uso de Internet estará presente como ayuda y documentación, como publicación y colaboración con el equipo de trabajo y finalmente como arquitectura de desarrollo.

1.1 Contenidos Conceptuales

1 Internet

1.1 Introducción

1.1.1 Nociones: Sockets, Puertos, Repaso de Redes TCP/IP

1.1.2 Colaboración por Internet para el módulo

1.1.3 Plataforma de Apoyo a las clases vía Web o grupos Yahoo

1.2 Descripción de recursos en Internet (URL Avanzado)

2 Correo Electrónico Avanzado

2.1 Opciones de Configuración: servidores, redireccionamiento, autenticación, etc

2.2 Configurar y Gestionar Carpetas

2.3 Importar y Exportar Mensajes de Correo Electrónico

2.4 Configuración de Varias Cuentas de Correo

2.5 Consulta de múltiples cuentas desde diferentes lugares

3 Buscar Información

3.1 Uso Básico de Google

3.2 Búsqueda Avanzada en Google

3.3 Formatos de Archivo: txt, html, pdf, sxw y doc

3.4 Búsqueda en listas de noticias

4 NetEtiquette

4.1 Normas de Educación en el envío de Mensajes

4.2 Relación Virtual: disponer de una identidad en Internet

4.3 Generación de confianza: escoger correctamente el nombre de usuario

5 Plataforma de Colaboración (grupos Yahoo):

5.1 Qué son

5.2 Cómo y para qué lo utilizaremos en clase

5.3 Servicios que nos ofrece

1.2 Contenidos Procedimentales

1. Gestión de una cuenta de Correo Electrónico

a) Crear una cuenta de correo electrónico gratuita. Conexión WebMail y POP

b) Configuración en un ordenador del centro

c) Manual de configuración de la cuenta de correo electrónico del colegio

2. Buscar información:

- a) Buscar información sobre un lenguaje de programación
 - b) Crear un archivo de texto con links (bookmarks) interesantes y su descripción
 - c) Añadir enlaces de Internet a Bookmarks o Mis Favoritos
 - d) Crear y Gestionar una cuenta de noticias (ejemplo lenguaje)
-

3. Gestión del Grupo Yahoo y/o Comunidad Colaboración

- a) Solicitar el ingreso en el grupo Yahoo y participar en él (mensaje de bienvenida)
- b) Enviar mensaje desde el portal Web y desde el correo electrónico

1.3 Criterios de Evaluación

- Ser capaz de manejar el correo electrónico de una manera eficiente
- Ser capaz de realizar búsquedas avanzadas con un buscador (p.ej. Google)
- Ser capaz de colaborar en la plataforma de colaboración escogida

Unidad Didáctica 2 Introducción a las Bases de Datos

En los sistemas de información actuales, las Bases de Datos son uno de los pilares fundamentales porque permiten dar consistencia y coherencia a los datos, para ser procesados desde diferentes aplicaciones.

Los contenidos son básicamente conceptuales y procedimentales aunque están inmersos los contenidos actitudinales en el trabajo en equipo. Las habilidades que intervienen son

de interpretación, análisis y síntesis de la información necesaria para buscar y comprender cada uno de los elementos necesarios para crear un programa independiente del lenguaje.

2.1 Contenidos Conceptuales

1 ¿Qué es un sistema Informático?

1.1 Hardware - Plataformas

1.2 Software - Sistema Operativo y Aplicaciones

1.3 Personas - Organización y Tipos de Usuarios

2 Modelo de Desarrollo de Tres capas: la importancia de los datos

2.1 ¿Qué es? Arquitectura Software dentro de una arquitectura C/S

2.2 Capas

2.2.1 La capa de Presentación: La estética

2.2.2 La capa de Programación (Business Logic): El código

2.2.3 La capa de Datos: La información

2.3 Posibles Ventajas: performance, flexibility, maintainability, reusability, and scalability

2.4 ¿Como construyo mi programa? Diseñar modelo de datos, diseñar lo que puedo hacer, permitir que el usuario lo utilice

3 Conceptos

3.1 ¿Qué es una BD? Datos consistentes, coherentes y fáciles de manejar

3.2 ¿Qué es un SGBD, Sistema Gestor de Bases de Datos?

3.3 ¿Qué es el apellido R, de Relacional?

3.4 ¿Es un archivo una Base de Datos?

3.5 Comparación de un SGBD con una estructura de datos dinámica/estática

4 Tipos y Arquitectura

4.1 Pequeña Historia de las BD

4.2 Componentes de una BD: HW, SW y Usuarios

4.3 Arquitectura ANSI/Sparc: Niveles Interno, Conceptual y Externo

4.4 Independencia de los Datos: Lógica y Física

4.5 Tipos de Modelos Lógicos: BD Jerárquicas, en Red, orientadas a Objetos y Relacionales

4.6 Diferencia entre Esquema de la Base de Datos y Ocurrencia/Estado de la BD

4.7 El modelo de Datos: describir la estructura de la BD. Entidades, Atributos e Interrelaciones

4.8 Ejemplos de SW de BD: Postgres, Oracle, DB2, Sybase, Informix, MS SQL Server, mysql, msql, solid, dBase, etc

4.9 Las hojas de cálculo: pseudo-bases de datos muy utilizadas

4.9.1 Hoja de Cálculo, Tablas, Filas, Columnas, Celdas

4.9.2 Relación entre Tablas y vínculos

4.9.3 Ventajas (simple y rápido) e Inconvenientes

4.10 MS Access: una pseudo-base de datos muy utilizada

5 Conceptos más avanzados

5.1 Los índices. Mejorar la localización de información

5.2 La concurrencia. Mejorar el servicio a varios clientes a la vez

5.3 Las transacciones. Mejorar el servicio de peticiones complejas (involucran más de una acción)

5.4 SGBD de verdad: Cumplen el Test ACID (Atomicity, Consistency, Isolation, Durability)

5.5 En la práctica: ¿Cómo hago para encontrar el Modelo de Datos?

5.5.1 Análisis

5.5.2 Diseño

5.5.3 Implantación

2.2 Contenidos Procedimentales

1 Analizar software según el modelo de Tres capas

1.1 Un programa en C (año pasado)

1.2 Cliente de Correo Electrónico

1.3 Kazaa (Descarga de Música)

2 Hacer una Base de Datos, en un archivo de texto (propongo para gestionar el préstamo de CDs)

3 Hacer una Base de Datos, utilizando Hojas de Cálculo

4 Terminar la BD con Hojas de Cálculo

2.3 Criterios de Evaluación

→ Entender los esquemas de datos en una aplicación

→ Diseñar una base de datos

→ Resolver problemas de datos

Unidad Didáctica 3 Modelo Relacional y el Modelo E/R.

El modelo Relacional permite una mayor consistencia y coherencia en el mantenimiento del modelo de datos. Aunque existen otros modelos (jerárquico, red, orientado a objetos) en la actualidad el adjetivo *relacional* es el más utilizado en los SGBD. Por otra parte, **el modelo E/R es la técnica de modelado más habitual** para realizar modelos de datos.

3.1 Contenidos Conceptuales

1 El Modelo Relacional

1.1 Situado en el Modelo ANSI/SPARC: Modelo Lógico)

1.2 Historia: E. F. Codd (1970), Segunda Generación

1.3 Relación: Filas y Columnas

1.4 Define: Estructura, Integridad y Manejo de los Datos

1.5 Conceptos Básicos:

1.5.1 Relación/Tabla

1.5.2 Columna/Atributo

1.5.3 Fila/Tupla/Registro

1.5.4 Dominio: conjunto de Valores Legales

1.5.5 Grado: número de atributos

1.5.6 Cardinalidad: número de tuplas

1.5.7 Definición Formal de Relación: Cabecera (Atributo/dominio) y Cuerpo (Atributo Valor)

1.6 Propiedades de las Relaciones:

1.6.1 Nombre único

1.6.2 Valores Atómicos (Normalizada)

1.6.3 Atributos Únicos

1.6.4 No importa el Orden de los Atributos

1.6.5 Tuplas Únicas

1.6.6 No importa el orden

1.7 Claves/Llaves/Keys

1.7.1 Para qué sirven

1.7.2 SuperClave: atributo o conjunto que identifican en modo único una tupla

1.7.3 Clave Candidata: única e irreductible

1.7.4 Claves simples y compuestas (número de atributos)

1.7.5 Claves Primarias y Claves Ajenas

1.7.6 Claves Ajenas: relación con una relación externa

1.8 Esquema de una Base de datos

1.8.1 Acercamiento al modelo de datos final

1.8.2 Nombre de tuplas y atributos, subrayando claves primarias

2 El Modelo Relacional (y 2)

2.1 Reglas de Integridad

2.1.1 Definir Reglas que los datos tienen que cumplir

2.1.2 Tipos: de dominio, de entidades, de integridad referencial, del usuario (lógica de negocio)

2.1.3 Concepto de Nulo: NULL (Operaciones Lógicas)

2.1.4 Restricción de Dominio: valores posibles

2.1.5 Restricción de Entidades: clave primaria no nula

2.1.6 Restricción de Integridad Referencial: clave ajena existente

2.1.6.1 Admitir Nulos

2.1.6.2 Borrado (Propagar, Restringir, Anular)

2.1.6.3 Modificación (Propagar, Restringir, Anular)

2.1.6.4 Restricción por reglas de negocio

2.2 Operaciones del Álgebra Relacional

2.2.1 Básicas

2.2.1.1 Selección

2.2.1.2 Proyección

2.2.1.3 Unión

2.2.1.4 Diferencia

2.2.1.5 Producto Cartesiano

2.2.2 Derivadas

2.2.2.1 Intersección

2.2.2.2 Concatenación-Join (Natural Join y Outer Join)

2.2.2.3 Extras: Asociación/Agregación (Theta-Join) y División

3 Modelo E/R

3.1 Qué es y para qué sirve

3.2 Breve Historia y Uso

3.3 Elementos

3.3.1 Entidades

3.3.2 Atributos

3.3.3 Relaciones

3.4 Relaciones: Fuertes y Débiles

3.4.1 Nombre

3.4.2 Grado

3.4.3 Cardinalidad: Cardinalidad (0,0)--(n,n) (mínimo, máximo) de ocurrencias

3.5 Clave Primaria y Candidata

4 Como **realizar el diseño**

4.1 Definir las Entidades

4.2 Seleccionar los Atributos y las Claves Primarias

4.3 Definir las relaciones entre entidades

4.4 Definir los atributos de las relaciones, los roles y la cardinalidad

4.5 Poner a prueba el modelo diseñado

5 Práctica Guiada y Explicada: [Gestión de Residuos](#)

6 Del Modelo E/R al esquema conceptual

6.1 El objetivo final es obtener un modelo de datos en SQL (desde el esquema conceptual)

6.2 Las entidades se convierten en tablas

6.3 Se define el tipo de relación

6.3.1 Relaciones 1:1: Atributos

6.3.2 Relaciones 1:N: Tabla y/o Atributo

6.3.3 Relaciones N:M: Tabla

6.4 Finalmente se realiza el esquema conceptual

6.5 Normalización

6.5.1 Objetivo Final: obtener un modelo de datos robusto, sin redundancias y que soporte las operaciones necesarias del sistema

6.5.2 Redundancia: datos en diferentes sitios

6.5.3 El problema de la redundancia y ventajas de la normalización

6.5.4 Normalizar: obtener nuevas tablas (relaciones) que no tengan redundancia de datos

6.5.5 Las formas normales:

6.5.5.1 1a: Valores atómicos, es decir, únicos e indivisibles

6.5.5.2 2a: Un atributo no clave, depende totalmente de la clave primaria (compuesta). y no solo de uno de los atributos que lo forman (Dependencia

funcional completa de la clave primaria)

6.5.5.3 3a: Un atributo no depende de otro atributo, es decir, sólo depende de la clave primaria. (Dependencia Funcional completa y no transitiva)

6.5.5.4 Boyce-Codd:

3.2 Contenidos Procedimentales

1 Introducción al Modelo Relacional

1.1 Análisis del sistema de préstamos de CDs (según modelo relacional)

1.2 Obtener el Esquema de la BD de préstamos de CDs

1.3 Obtener el Esquema de la BD de un sistema que ayude a gestionar la biblioteca del colegio

2 Reglas de Integridad y Operaciones

2.1 Definir las reglas de integridad

2.2 Definir consultas sobre el sistema de préstamos de CDs y Bibliotecas: Posibles operaciones

2.3 Operaciones de Agregación

3 Modelo Entidad / Relación

3.1 Diagrama E/R del sistema de préstamos de CDs

3.2 Diagrama E/R de la gestión de incidencias

4 Ejercicio Inventario Hardware Colegio

4.1 Ejercicio Previo

4.2 Ejercicio Guiado

5 Ejercicio Residuos Tóxicos

5.1 Ejercicio Previo

5.2 Ejercicio Guiado

6 Normalización

6.1 Ejemplos de Normalización

6.2 Normalizar la BD de los préstamos de los CDs

3.3 Criterios de Evaluación

- Entender los **conceptos Básicos** del Modelo Relacional
- Entender los **conceptos Básicos** del Modelo E/R
- Entender los **problemas de normalización de datos** (redundancia)

Unidad Didáctica 4 Crear Modelos de Datos (SQL I)

El lenguaje SQL se utiliza para la **descripción del modelo de datos**, y para la modelización de consultas y acciones sobre la base de datos. En esta unidad didáctica el objetivo es obtener un modelo de datos robusto y bien definido, para ello se utiliza el lenguaje DDL (Data Description Language).

4.1 Contenidos Conceptuales

1 Conectarse a un SGBDR: consola (psql) y Modo SQL Emacs

1.1 Qué es una base de datos y qué es una tabla

1.2 Sintaxis comando: (psql --help) Ejemplo: psql -h HOST -d BASEDATOS -U USUARIO; Listado de BD: psql -l

1.3 Pidiendo ayuda: \? (comandos de consola), \h (ayuda sobre SQL), \g (ejecutar consultas, vale ";"), \q (salir de consola)

1.4 Algunos Comandos útiles: \l, \d, \dt, \d tabla ...

1.5 Trabajo con la consola de Emacs

1.5.1 Configurar Emacs: M-x customize-options (sql-postgres-options, sql-user, sql-postgres-program)

1.5.2 SQL Buffer (M-p, M-n, M-r, M-s)

1.5.3 Modo SQL (C-c C-r, C-c C-b) - Ligado a un buffer interactivo

2 Introducción a SQL

2.1 Lenguaje SQL

2.1.1 Pequeña Historia (Investigación: 70s (SEQUEL/2), Producción: 80s (SQL))

2.1.2 Standards: ANSI X3.135 (1986), ISO (1987), SQL1 (1989), SQL2 (1992), SQL3 (1999) pero NINGUNO los cumple totalmente (el SGBDR más cercano es Postgres)

2.1.3 Características: declarativo (importa el qué no el como), ejemplo ("select nombre from alumnos where id=120")

2.1.4 Palabras Reservadas (tokens: palabras reservadas e Identificadores)

2.2 Estructura de una sentencia SQL (tres ejemplos): comando, expresiones, cláusula, operador y funciones de agregado

2.2.1 Comando: describe la acción a realizar (por ejemplo SELECT)

2.2.2 Palabras reservadas: comandos y símbolos especiales: (*, ', ")

2.2.3 Constantes: cadena ('cadena de texto'), numeros (15 o 14.34 o 14e+8), fechas ('2004-01-02'), boolean (true/false, t/f) etc

2.2.4 Expresiones: identificadores y operadores (por ejemplo 15.4*2 ó 15.4 > 4)

2.2.5 Operadores: relaciona un identificador con un valor (aritméticos y lógicos)

2.2.6 Cláusula: describe completamente el comando (FROM alumnos)

2.3 Comando Básico (por ejemplo SELECT)

2.3.1 Ayuda Interactiva: \help SELECT

2.3.2 Envíar el comando al SGBDR: \g ó más útil ; (punto y coma)

2.3.3 Uso de la palabra reservada AS (nombre de salida)

2.3.4 Sintaxis (no está completa):

```
SELECT * | expression [ AS output_name ] [, ...] [ FROM from_item [, ...]]
```

```
2.4 Ejemplos:      select      id,      num      from      paquetes;
update      alumnos      set      nombre      =      'Manuel';
create      table      prueba      (id      integer,      nombre      varchar(80));
INSERT INTO inventario VALUES ('0385121679', true);
```

3 Tipos de Datos

3.1 SQL: lenguaje fuertemente tipado (control de tipos de datos)

3.2 Tipos básicos (estándares): boolean, caracteres (char, varchar), numeros (integer, real, numeric(p,s)) y Tiempo (date, time, timestamp)

3.3 Ejemplos de Constantes: String, Integer y Boolean

3.4 Forzar el tipo: operador ::. Ejemplo: select 15::real;

3.5 En realidad hay muchos tipos de datos (URLs, geometría, etc): Ver con \dT en consola

4 Creación de Datos (DDL)

4.1 Creando Tablas

4.1.1 create table (tabla, atributos, dominios)

4.1.2 drop (destruir tablas, destruir columna)

4.1.3 alter table (modificar la estructura de una tabla)

4.1.4 Entender la sintaxis (\help comando SQL)

4.2 Restricciones (Constraints)

4.2.1 Sobre los campos: NOT NULL, DEFAULT, UNIQUE, PRIMARY KEY, check.

Por ejemplo:

```
create table prueba (ejemplo_campo integer default 9, otro_ejemplo varchar(80)
not null)
```

4.2.2 Con otras tablas: FOREIGN KEY, REFERENCES. Por ejemplo:

```
create table prueba (ejemplo_clave_ajena integer references usuarios (dni) )
```

4.2.3 Añadir restricciones (table_constraint) a la tabla ya creada

5 Manipulación Básica de Datos (DML)

5.1 ¿Qué es?

5.2 Consultas (SELECT) (Fijarse: AS output_name, FROM from_item, WHERE condition)

5.3 Operadores Comparación: aritméticos (>, <, =, OR, AND ...), caracteres (LIKE, expresiones regulares ~)

6 Realizar un modelo de datos (mini-banco)

7 Uso de la **Sintaxis de SQL**:

7.1 Caracteres Especiales

7.1.1 En mayúsculas: palabras reservadas

7.1.2 En minúsculas: expresiones

7.1.3 Entre Corchetes []: opcional

7.1.4 Entre Llaves { }: obligatorio

7.1.5 Tres puntos: repetir anterior

7.1.6 Con Operador | : Opciones Excluyentes

7.2 Sintaxis [CREATE TABLE](#)

7.2.1 Explicación Column Constraint

7.2.2 Explicación Table Constraint

7.3 Sintaxis [ALTER TABLE](#)

8 Repaso y crear nuevo modelo de datos (II)

9 Más sobre modelo de datos

9.1 Control de datos de entrada: check (expresion)

9.2 Creación de Secuencias

9.2.1 CREATE SEQUENCE (sintaxis)

9.2.2 Tipo de Datos: serial

4.2 Contenidos Procedimentales

1 Conectarse a un SGBDR: consola (psql) y Modo SQL Emacs

1.1 Qué es una base de datos y qué es una tabla

1.2 Sintaxis comando: (psql --help) Ejemplo: psql -h HOST -d BASEDATOS -U USUARIO; Listado de BD: psql -l

1.3 Pidiendo ayuda: \? (comandos de consola), \h (ayuda sobre SQL), \g (ejecutar consultas, vale ";"), \q (salir de consola)

1.4 Algunos Comandos útiles: \l, \d, \dt, \d tabla ...

1.5 Trabajo con la consola de Emacs

1.5.1 Configurar Emacs: M-x customize-options (sql-postgres-options, sql-user, sql-postgres-program)

1.5.2 SQL Buffer (M-p, M-n, M-r, M-s)

1.5.3 Modo SQL (C-c C-r, C-c C-b) - Ligado a un buffer interactivo

2 Introducción a SQL

2.1 Lenguaje SQL

2.1.1 Pequeña Historia (Investigación: 70s (SEQUEL/2), Producción: 80s (SQL))

2.1.2 Standards: ANSI X3.135 (1986), ISO (1987), SQL1 (1989), SQL2 (1992), SQL3 (1999) pero NINGUNO los cumple totalmente (el SGBDR más cercano es Postgres)

2.1.3 Características: declarativo (importa el qué no el como), ejemplo ("select nombre from alumnos where id=120")

2.1.4 Palabras Reservadas (tokens: palabras reservadas e Identificadores)

2.2 Estructura de una sentencia SQL (tres ejemplos): comando, expresiones, cláusula, operador y funciones de agregado

2.2.1 Comando: describe la acción a realizar (por ejemplo SELECT)

2.2.2 Palabras reservadas: comandos y símbolos especiales: (*, ', ")

2.2.3 Constantes: cadena ('cadena de texto'), numeros (15 o 14.34 o 14e+8), fechas ('2004-01-02'), boolean (true/false, t/f) etc

2.2.4 Expresiones: identificadores y operadores (por ejemplo 15.4*2 ó 15.4 > 4)

2.2.5 Operadores: relaciona un identificador con un valor (aritméticos y lógicos)

2.2.6 Cláusula: describe completamente el comando (FROM alumnos)

2.3 Comando Básico (por ejemplo SELECT)

2.3.1 Ayuda Interactiva: \help SELECT

2.3.2 Enviar el comando al SGBDR: \g ó más útil ; (punto y coma)

2.3.3 Uso de la palabra reservada AS (nombre de salida)

2.3.4 Sintaxis (no está completa):

```
SELECT * | expression [ AS output_name ] [, ...] [ FROM from_item [,  
...]]
```

2.4 Ejemplos:

```
select      id,      num      from      paquetes;  
update     alumnos  set      nombre  =      'Manuel';  
create     table    prueba  (id      integer,  nombre  varchar(80));  
INSERT INTO inventario VALUES ('0385121679', true);
```

3 Tipos de Datos

3.1 SQL: lenguaje fuertemente tipado (control de tipos de datos)

3.2 Tipos básicos (estándares): boolean, caracteres (char, varchar), numeros (integer, real, numeric(p,s)) y Tiempo (date, time, timestamp)

3.3 Ejemplos de Constantes: String, Integer y Boolean

3.4 Forzar el tipo: operador ::. Ejemplo: select 15::real;

3.5 En realidad hay muchos tipos de datos (URLs, geometría, etc): Ver con \dT en consola

4 Creación de Datos (DDL)

4.1 Creando Tablas

4.1.1 create table (tabla, atributos, dominios)

4.1.2 drop (destruir tablas, destruir columna)

4.1.3 alter table (modificar la estructura de una tabla)

4.1.4 Entender la sintaxis (\help comando SQL)

4.2 Restricciones (Constraints)

4.2.1 Sobre los campos: NOT NULL, DEFAULT, UNIQUE, PRIMARY KEY, check.

Por ejemplo:

```
create table prueba (ejemplo_campo integer default 9, otro_ejemplo varchar(80)
not null)
```

4.2.2 Con otras tablas: FOREIGN KEY, REFERENCES. Por ejemplo:

```
create table prueba (ejemplo_clave_ajena integer references usuarios (dni) )
```

4.2.3 Añadir restricciones (table_constraint) a la tabla ya creada

5 Manipulación Básica de Datos (DML)

5.1 ¿Qué es?

5.2 Consultas (SELECT) (Fijarse: AS output_name, FROM from_item, WHERE condition)

5.3 Operadores Comparación: aritméticos (>, <, =, OR, AND ...), caracteres (LIKE, expresiones regulares ~)

6 Realizar un modelo de datos (mini-banco)

7 Uso de la **Sintaxis de SQL**:

7.1 Caracteres Especiales

7.1.1 En mayúsculas: palabras reservadas

7.1.2 En minúsculas: expresiones

7.1.3 Entre Corchetes []: opcional

7.1.4 Entre Llaves { }: obligatorio

7.1.5 Tres puntos: repetir anterior

7.1.6 Con Operador | : Opciones Excluyentes

7.2 Sintaxis [CREATE TABLE](#)

7.2.1 Explicación Column Constraint

7.2.2 Explicación Table Constraint

7.3 Sintaxis [ALTER TABLE](#)

8 Repaso y crear nuevo modelo de datos (II)

9 Más sobre modelo de datos

9.1 Control de datos de entrada: check (expresion)

9.2 Creación de Secuencias

9.2.1 CREATE SEQUENCE (sintaxis)

9.2.2 Tipo de Datos: serial

4.3 Criterios de Evaluación

- Saber cómo **conectarse a una base de datos** Postgres
- Descripción completa de un **modelo de Datos** (DDL)

Unidad Didáctica 5 Utilizar el Modelo de Datos (SQL II)

Lo que se pretende en esta unidad didáctica es saber cómo utilizar la base de datos para **introducir, modificar y consultar datos**. Haremos especial hincapié en la creación de consultas, y también en entender consultas complejas escritas por otros.

5.1 Contenidos Conceptuales

1 Inserción de Datos: Sentencia [INSERT INTO](#)

2 Modificando y consultando datos:

2.1 Modificación de Datos (con cuidado): Sentencia [UPDATE](#)

2.2 Consultas Básicas: Sentencia `SELECT * | expression [AS output_name] [, ...]`
`[FROM from_item [, ...] [WHERE condition]]`. Por ejemplo `select nombre as "El nombre", apellidos from alumnos where id_alumno = 25 ó select 25*8 AS "Operacion"`

2.3 Condiciones (condition): expresiones que devuelvan valor true. Por ejemplo:
`nombre = 'Pepe' o nombre = 'Paco' AND edad < 25`

2.4 Borrado de Datos (con cuidado): Sentencia [DELETE](#)

3 Consultas con **varias tablas**

3.1 Uso de varias Tablas: `FROM tabla1, tabla2, tabla3`: producto cartesiano de todas las tablas

3.2 Se utilizan las claves ajenas (foreign key) para generar condiciones (WHERE) sobre ese producto cartesiano.

3.3 El uso de Alias en Campos y Tablas, permite claridad y facilidad en la generación de consultas complejas

4 Mejorando las consultas

4.1 Cláusulas `ORDER BY`, `LIMIT`, `OFFSET`

4.2 Uso de Funciones

4.2.1 Qué son (expresiones)

4.2.2 Tipos y Ejemplos

4.2.2.1 [Matemáticas](#): `log(x)`, `random()`

4.2.2.2 [Manejo de Caracteres](#): char_length('cadena'), substring ('cadena' from 'REGEXP')

4.2.2.3 [Tiempo y Fecha](#): current_time, current_date, now()

4.2.2.4 [De agregación](#): count, max, min, avg

4.3 Operaciones sobre cadenas: operador || (concatenar) y ~ (REGEXP)

5 Sintaxis completa de [SELECT](#)

5.1 Uso de JOIN en las consultas (uso en cláusula FROM)

5.1.1 CROSS Join: equivalente a separar por comas

5.1.2 Requieren Condiciones (qualifications): INNER Join y OUTER Join (right y left)

5.1.3 Las condiciones se añaden con la palabra ON y un expresión ON (join_condition)

5.1.4 Ejemplos

5.2 [Agregación de Registros](#):

5.2.1 Cláusulas GROUP BY: agrupar según campo y utilizar funciones de agregación

5.2.2 Cláusula HAVING: condiciones sobre las filas agrupadas

6 [Ejercicio Completo](#) de Modelo de Datos y Entender consultas complejas de otros

5.2 Contenidos Procedimentales

1 Lenguaje DML

1.1 Ejecutar Consultas Básicas (SELECT) (Ver el contenido de la tabla, ver datos, AS , FROM)

1.2 Introducir datos, todos los campos o por separado (INSERT INTO)

1.3 Probar cómo protege la base de datos su consistencia (forzar errores)

2 Modificar Valores, que he introducido

2.1 Modificar Filas / Datos (UPDATE)

2.2 Borrar Filas (DELETE)

3 Consultas Básicas y con Condiciones

4 Practicar con consultas con origen en diferentes tablas

5 Consultas con Funciones y trabajar con el modelo de datos

Modelo de Datos completo

Se busca construir un **sistema de información para gestionar una base de datos de películas**. Básicamente se requiere gestionar la información de las películas, las estrellas (actores / actrices) que participan en ella y los estudios de producción a las que pertenecen.

Además se requiere que las **consultas que tendrá que soportar** el modelo de datos sean:

1. Encontrar la dirección de un estudio
2. Encontrar todas las estrellas que participaron en películas realizadas en el año TAL o en alguna película que contenga la palabra TAL en el título.

3. ¿Quién fue la estrella de la película TAL?
4. ¿Qué películas tienen mayor duración que la película TAL?
5. Encontrar el título y la duración de todas las películas producidas por los estudios TAL en el año TAL, ordenando la salida por su duración
6. Encontrar todas las estrellas que son hombres o viven en TAL ciudad
7. ¿Qué estrellas distintas aparecen en las películas producidas por el estudio TAL, en TAL año? Si las películas salen repetidas, mostrarlas una sólo vez
8. ¿Cuál es la media, en años, de publicación de las películas?
9. Encontrar las películas que empiecen por L, que tengan más de una palabra y que contengan en el título las letras a,p y m
10. Encontrar la suma de la duración de todas las películas de cada estudio
11. Además, cada alumno deberá, añadir tres ejemplos de consultas más, para vender su producto.

El trabajo se hará utilizando la base de datos del colegio y las tablas de la base de datos serán precedidas por el username del alumno. Recordad que todos tenéis el mismo usuario, por lo tanto cuidado con las tablas que modificáis y pensad la manera de tener copias de seguridad del modelo de datos. Se dispondrá de tres clases para el trabajo y en la última clase iré revisando el trabajo.

5.3 Criterios de Evaluación

- Descripción de las **acciones sobre la base de datos** (DML)

- Descripciones de consultas, uso avanzado de SELECT

Unidad Didáctica 6 El Servidor SGBDR (PostgreSQL)

Postgres es una **base de datos avanzada** que cumple con el Test ACID. Es gratuita y con licencia BSD. Es una base de datos con toda la funcionalidad necesaria para cualquier tipo de aplicación desde la más sencilla a la más compleja por eso la hemos escogido como herramienta de uso y administración como motor de base de datos.

6.1 Contenidos Conceptuales

- 1 **Introducción:** Porqué, [Historia](#), Características, [Ventajas](#)/Inconvenientes
- 2 [Instalación desde Código Fuente](#) (procedimiento y cálculo de horas)
- 3 ¿Qué es?
- 4 Plan de Trabajo: qué necesito, software necesario (versión), módulos y documentación
- 5 [Instalación del Programa](#)
 - 5.1 Instalación: configuración (./configure), compilación (make), instalación (make install)
 - 5.2 Creación de usuario y directorio de la base de datos. (data y configurar permisos)
 - 5.3 Jerarquía de directorios: bin, doc, include, man, share, lib y DATOS
 - 5.4 Inicializar el directorio de Datos (initdb -D el_directorio_que_escoja)
- 6 Arrancar el SGBDR (postmaster y/o script pg_ctl). Poner SIEMPRE el directorio de la

BD

6.1 Probar que está funcionando (createdb y psql)

6.2 Trabajo Post-Instalación: configurar PATH, Librerías

6.3 Conexión a la base de datos (createdb, createuser, dropdb, dropuser)

7 Configuración Servidor

7.1 Archivo de Configuración: postgresql.conf (en \$PGDATA). Si es necesario modificar los parámetros [listen_addreses](#) y el [Puerto](#)

7.2 Archivos de Log: redireccionando la salida del comando o con opciones en el script

8 Control de Acceso (Autenticación): pg_hba.conf, pg_ident.conf

8.1 [Authentication](#)

8.1.1 Control Basado en Host (Host-Based Access pg_hba.conf)

8.1.2 Quién, desde dónde y a qué

8.1.3 Tipos de conexión localhost y vía host (TCP/IP)

8.2 Estructura pg_hba.conf:

8.2.1 Tipo de Conexión: local | host | hostssl

8.2.2 database (all | sameuser | name)

8.2.3 user

8.2.4 ip_addr netmask

8.2.5 authentication_method (password, crypt, ident, trust, md5, reject)

8.3 [Ejemplos](#) de Archivos de Configuración

9 Tareas de **Administración y Mantenimiento**

10 Tipos de usuarios: DBA y usuarios (normal, desarrollador, programador, etc)

11 Gestión de [Usuarios](#) (y Grupos)

11.1 Tablas del Sistema sobre usuarios:

11.1.1 pg_user: Tabla con información de usuarios

11.1.2 pg_shadow: Tabla con información sobre la clave de usuarios

11.2 Gestionar Usuarios

11.2.1 Lenguaje DCL: Data Control Language. Sentencias SQL: CREATE USER, DROP USER Y ALTER USER

11.2.2 Utilizando comandos: createuser, dropuser

12 Gestión de [Bases de Datos](#)

12.1 Utilizar Lenguaje DCL: Data Control Language. Sentencias SQL: CREATE DATABASE, DROP DATABASE y ALTER DATABASE

12.2 Utilizando comandos: createdb, dropdb

12.3 Tabla del Sistema: pg_database

13 Gestión de [Permisos sobre Objetos](#)

13.1 Tipos de permisos (hay más): SELECT, INSERT, UPDATE, DELETE,

REFERENCES, CREATE y ALL

13.2 La palabra PUBLIC especifica cualquier usuario

13.3 Se pueden otorgar ([GRANT](#)) o quitar permisos ([REVOKE](#))

13.4 Para ver los permisos de un objeto: \z nombre_objeto

13.5 Tablas del Sistema: pg_class. Ejemplo: select relname, relacl from pg_class;

14 Gestión de [Copias de Seguridad](#)

14.1 Utilizando comandos (El que hay que utilizar): pg_dump (para cada bd) y pg_dumpall (todas). Revisar las opciones (modelo de datos, contenido, comprimir, etc)

14.2 Copiar el Sistema de Archivos (PGDATA)

15 Otras funciones del mantenimiento:

15.1 Sobre la actividad de la BD (tablas del Sistema)

15.2 Sobre el Rendimiento del Sistema: Procesador, Memoria y Sistemas de Archivos)

15.3 Sobre la Documentación de los Modelos de Datos y Atención a los desarrolladores

6.2 Contenidos Procedimentales

1 Instalación de Postgres 8.0.1 (dos días):

1.1 Instalar, arrancar y parar (comando y scripts)

1.2 Conocer los diferentes archivos que hay en el directorio de instalación

1.3 Configuración del Servidor y Logs:

1.3.1 Arrancar/Parar Servidor con logs (creados por mí, y a través del script "control_postgres")

1.3.2 Añadir configuración en postgres.conf para activar conexión TCP y detalle del archivo de log

1.4 Configurar Acceso vía Host (modificando base de datos, usuarios y mecanismo de envío de la clave)

1.4.1 Acceso desde cualquier parte

1.4.2 Acceso desde una sola máquina

2 Gestión de Autenticación

2.1 Gestión de Usuarios: Añadir, Borrar y Modificar parámetros de usuarios

2.2 Gestión del Acceso vía host y usuario a las bases de datos

2.3 Revisar Tablas del sistema: pg_user, pg_shadow

3 Gestión de Permisos

3.1 Crear un usuario "invitado" que sólo pueda leer la tabla

3.2 Crear un usuario "web" que pueda añadir, leer, actualizar y borrar datos

3.3 Comprobar los permisos a través de la tabla pg_class o el comando \z

4 Administracion y Mantenimiento

4.1 Repaso a la situación interna: VACUUM y vacuumdb

4.2 Hacer copia de seguridad: pg_dump y pg_dumpall (¿cuando?, ¿cómo?, hot backup)

4.3 Recuperar la base de datos: pg_restore. Crear la BD , los usuarios, y los características de la BD

4.4 Copiar el sistema de archivos directamente

6.3 Criterios de Evaluación

- Conocer su **Arquitectura y Funcionamiento**
- **Instalación, Configuración y Mantenimiento** de Postgres
- Funciones y Trabajo de **DBA**

Unidad Didáctica 7 Análisis, Diseño e Implantación de un proyecto Real (Aplicación Web)

Esta unidad didáctica se coordinará con la asignatura de DFSI. Tendrá un contenido puramente procedimental desarrollando un proyecto pequeño de desarrollo de una utilidad software a través de Internet. Se denominará mini-proyecto en comparación con el módulo Proyecto Integrado.

Se ofrecerán varias alternativas en forma de requerimientos de un cliente, y los alumnos tendrán que poner en práctica la implementación del sistema de forma autónoma simulando un entorno profesional. Esta unidad didáctica se centrará en la supervisión y repaso de los contenidos conceptuales y procedimentales relacionados con el análisis y diseño de nuevas funcionalidades.

7.1 Contenidos Conceptuales

1 Planificación del Proyecto

1.1 Cómo leer los requisitos

1.2 Análisis de los Requisitos

1.3 Planteamientos de Diseño

1.3.1 Presentación

1.3.2 Datos

1.3.3 Lógica/Programación

2 Diseño y esbozo de la arquitectura a implementar

3 Puesta en marcha de una aplicación Software

3.1 Instalación

3.2 Mantenimiento y control de versiones

3.3 Documentación en línea

7.2 Contenidos Procedimentales

Desarrollo de un mini proyecto a escoger entre cuatro alternativas, o escoger otra alternativa (desde el proyecto integrado o una planteada por el alumno). En grupos de máximo dos personas.

7.3 Criterios de Evaluación

- Ser capaz de montar la arquitectura del sistema
- Ser capaz de realizar el modelo de datos del sistema
- Ser capaz de realizar las páginas dinámicas del sistema

7 CONTENIDOS ACTITUDINALES

Los contenidos relacionados a continuación son comunes a todas las unidades didácticas.

- Comportamiento autónomo para la utilización de bibliografía y otras fuentes de información
- Interés por seleccionar el entorno y las herramientas de programación más adecuados para conseguir mayor productividad
- Interés por utilizar las herramientas de programación más adecuadas para conseguir mayor productividad y portabilidad
- Constancia de la prueba funcional de los programas desarrollados y medida de los recursos del sistema
- Constancia en la documentación del sistema
- Interés por la producción de guías de usuario claras y sencillas
- Valoración de la redacción cuidados de los informes y la utilización adecuada de la

terminología informática.

→ Hábito de trabajo en grupo y respeto por las normas establecidas en el colectivo.

8 METODOLOGÍA

La metodología que se utilizará en clase potenciará, en lo posible los siguientes aspectos:

- Colaboración y Participación (Internet y clase)
- Interés (Contenidos Interesantes)
- Educación personalizada (Evaluación Continua)

Dentro del proceso de enseñanza-aprendizaje tradicional se pensaba que el conocimiento fluía del profesor al alumno. Para aumentar la participación e interés del alumno se hace necesario hacerlo intervenir en el proceso. Palabras como “Aprendizaje Significativo” y “Constructivismo” resuenan dentro de las teorías pedagógicas que señalan la participación del alumno en su propio aprendizaje.

El uso de una plataforma de colaboración por Internet permite potenciar la colaboración porque el alumno puede aportar, por ejemplo, una consulta, o un enlace de Internet que ha encontrado o incluso preguntas a otros alumnos. Además el profesor enviará antes de la clase un correo electrónico a todos los alumnos con un pequeño esquema con el mapa de contenidos y las actividades a realizar en clase. De esta forma los servicios por Internet (correo electrónico y navegación Web por el grupo de colaboración) permiten una comunicación fluida entre los alumnos, y con el profesor. Además, y es este uno de las

aspectos señalados en el Proyecto Educativo del Centro, el uso de Internet permite una mayor integración de los alumnos que trabajan y tienen dificultades para asistir de forma continua a clase.

Sin embargo este sistema en ningún caso podrá sustituir a la educación tradicional, y en ese sentido, las clases consistirán generalmente en:

- Explicación Oral del mapa de contenidos de la clase, prestando atención a la participación de los alumnos, y aprovechando sus preguntas/intervenciones para aportar los contenidos de la unidad didáctica
- Propuestas de actividades de enseñanza-aprendizaje, de tipo práctico para que el alumno (sólo o en grupo) las desarrolle en clase. Sería ideal también que los alumnos terminen proponiendo alguna actividad.

Además el diseño de la programación y las unidades didácticas presentan contenidos interesantes y actuales, dentro del planteamiento del Diseño Curricular Base del Ciclo Formativo para este módulo. El uso de enlaces y materiales de Internet permite que incluso los alumnos puedan proponer materiales para la unidad didáctica (siempre revisados por el profesor), se generan por lo tanto contenidos más dinámicos que si se utilizara un libro de texto. Hay que tener en cuenta que existirá un trabajo importante de supervisión de los materiales aportados por los alumnos y/o de su programación antes de poner en práctica sus propuestas.

El componente de educación personalizada, es quizá más complejo de delimitar. Para ello, además de la relación con el alumno de tipo personal (en clase y vía correo electrónico) se tratará de, en cada unidad didáctica, evaluar de forma personalizada e

incluso asignar tareas de formas personalizada. Para ello se utilizarán los criterios de clasificación de muy bien, bien, regular y mal en la evaluación práctica continua que permitirán conocer el grado de soltura en el manejo de contenidos de la unidad didáctica, que tiene el alumno. Estos aspectos se dejarán claros al comienzo del módulo tratando de aclarar que a cada alumno se le evaluará de forma personalizada según su nivel pero con pruebas generales (test) donde la evaluación será relativa a la clase.

Esta metodología pretende tener al alumno con una dinámica de estudio y trabajo continuo, sobre todo en clase.

9 MEDIDAS DE ATENCIÓN A LA DIVERSIDAD.

Las medidas de refuerzo educativo tendrán como objetivo potenciar los conocimientos y procedimientos englobados en los contenidos mínimos exigibles, que no fueron superados por los alumnos en las diferentes pruebas de evaluación. Para estos alumnos se realizarán las siguientes medidas:

- Ejercicios específicos
- Atención exclusiva con herramientas síncronas (Chat y Telefonía IP) y asíncronas (correo electrónico y Grupo Yahoo)
- Evaluación diaria con mayor supervisión

El profesor mantendrá un servicio Web de la asignatura

(<http://www.cesareox.com/docencia/sxbd/>) ,con la información del curso y que será actualizado diariamente. Además el profesor enviará un correo electrónico con el esquema de la clase y permitirá a estos alumnos comunicarse vía correo electrónico o chat según disponibilidad. Además el centro mantiene para cada ciclo un grupo yahoo de colaboración.

También existirán alumnos que obtendrán rápidamente los conocimientos y procedimientos de la unidad didáctica, ya sea por su capacidad o por los conocimientos autodidactas tan comunes en las áreas de informática. Para estos alumnos se tendrán en cuenta las siguientes medidas:

- Profundizarán en cada unidad didáctica y se hará supervisión
- Generarán documentación Web para el resto de alumnos

En el ámbito de la Formación Profesional, cada vez más coordinada con la realidad socio laboral, existen alumnos (especialmente en los turnos de tarde y noche) que trabajan y se les hace complicado asistir a todas las sesiones. Esta situación no es problema gracias al uso de herramientas Internet.

Además el uso de Internet y una plataforma de colaboración es de hecho la medida más importante de atención a la diversidad, pues permite que se potencia la relación entre alumnos que o no pueden asistir a clase o tienen dificultades en el seguimiento.

10 Evaluación

10.1 Procedimiento de Evaluación Ordinario

El sistema de evaluación empleado tiene como objetivo evaluar los contenidos, procedimientos y actitudes que pretendemos que el alumno tenga, pero además queremos transmitir la idea de la disciplina y la autoformación.

Por otra parte, basaremos la evaluación como una forma de supervisión en el mundo laboral y se permitirá en los exámenes el acceso a apuntes, notas, Internet o cualquier medio de documentación disponible también en el mundo laboral.

La evaluación continua es la esencia de este sistema de evaluación, ayudando a tener un ritmo de trabajo todo el curso.

Las medidas de evaluación serán, para cada evaluación las mismas. Recordemos que este módulo tiene dos evaluaciones. El sistema se basará en:

- Al menos tres exámenes tipo test de 40 preguntas, con cuatro alternativas y una respuesta única en cada pregunta. Los exámenes serán sorpresa e incluirán cualquier unidad didáctica (o parte de ella) dada en clase hasta el día anterior. La nota de cada alumno se comparará con el rendimiento de la clase en el examen. Será el 30% de la nota. La nota será entre 1 y 10, con un decimal.
- Se hará una evaluación diaria del trabajo práctico y procedimental de forma continua obteniendo entre 6 y 8 notas a lo largo de una evaluación. El profesor

podrá evaluar en cualquier momento. Será el 30% de la nota y será uno de los siguientes valores: Muy Bien, Bien, Regular y Mal. Se podrán pedir trabajos prácticos o tareas específicas al alumno, o grupo de alumnos fruto de este tipo de evaluación.

- Al final de la evaluación existirá un examen oral y personal para cada alumno. Tendrá una duración de 15 minutos y se podrá preguntar a cada alumno cualquier contenido teórico y práctico del módulo. Tendrá un valor del 30% de la nota y será una nota entre 1 y 10, con un sólo decimal. En la última evaluación este examen será un proyecto final.
- Valoración del trabajo en clase, la asistencia y la colaboración con otros alumnos. Esta nota supone el 10% del total y el profesor evaluará la participación en clase, la colaboración en la plataforma de Internet y la asistencia a las sesiones. Será una nota entre 1 y 10 con un decimal

No existirá el concepto de “recuperación” en el sentido de que existan exámenes específicos cuando no se superen unidades anteriores. Se hará una evaluación continua y personalizada, que permite evaluar al alumno todos los contenidos. No se dividirá en partes el módulo y no existirá el concepto de “unidades pendientes” porque se pretende transmitir al alumno un concepto integral del módulo.

En las evaluaciones se utilizarán dos tipos de notas, entre 1 y 10 con un decimal y uno entre cuatro valores (Mal, Regular, Bien y Muy Bien). En el área de la administración de sistemas informáticos la industria suele categorizar a los profesionales en Novato, Junior, Senior y Experto, por ellos hemos utilizado las notas como una forma de evaluar al alumno suponiendo una contratación del alumno por parte de la empresa “profesor”. De

esta manera la correspondencia de notas sería:

- 10 y 9. Estos alumnos pueden trabajar sin casi supervisión, son capaces incluso de tomar decisiones sobre nuevas tareas y manejar el concepto evaluado con soltura.
- 8. Estos alumnos tienen bastante soltura pero precisan de supervisión, no mucha pero necesitarían formación extra de corta duración para adquirir una soltura total.
- 7 Estos alumnos conocen las tareas y las pueden desarrollar pero necesitan de revisión muy detallada
- 5 y 6 Estos alumnos tienen los conocimientos mínimos pero deberían tener más práctica en las tareas y competencias del módulo. Para adquirir soltura necesitarían formación de larga duración.

En el caso de pérdida de evaluación continua, por ejemplo si las faltas de asistencia son mayores al 10% de las sesiones del módulo, se hará un examen final que constará de:

- Examen tipo test de 120 preguntas. Será eliminatorio y el alumno para aprobar debe sacar un mínimo de 70 respuestas correctas. Tendrá una duración de 100 minutos.
- Examen práctico de 4 horas de duración para evaluar contenidos procedimentales, delante del ordenador
- Examen oral y personal de 30 minutos para la evaluación de todo el módulo

10.2 Criterios y Procedimientos para establecer las actividades de recuperación de los módulos pendientes

La recuperación de este módulo, cuando esté pendiente se realizará en el último trimestre del segundo curso. El profesor encargado de las tareas de recuperación y evaluación será un profesor de segundo curso.

El alumno que lo desee como medida de refuerzo podrá asistir a clase siempre y cuando no interrumpa ni altere las clases del módulo de segundo curso y exista espacio físico en el aula. Si el alumno se incorpora al módulo, al repetir curso, se utilizarán los criterios de evaluación ordinarios

El sistema de recuperación se hará basándose en el trabajo personal del alumno y la consulta en horas de profesor en el último trimestre del curso. Las horas disponibles para recuperación serán aproximadamente la mitad de las del curso ordinario.

El sistema de recuperación se basará en el desarrollo práctico de un miniproyecto, equivalente a las dos últimas unidades didácticas y la evaluación continua de contenidos de las unidades didácticas a medida que avanza el trimestre.

10.3 Procedimientos de Evaluación y cualificación extraordinaria del alumnado con módulos pendientes

Para aprobar el módulo deberán ser capaces de realizar la configuración básica y la instalación del Servidor Web con una página Web. La nota vendrá dada por el desarrollo

del mini proyecto que se presenta en las dos últimas unidades didácticas del módulo.

Se evaluará este trabajo y se harán evaluaciones periódicas de tipo práctico para evaluar los contenidos procedimentales y conceptuales de las unidades didácticas.

10.4 Contenidos Mínimos Exigibles

El procedimiento de evaluación tratará de medir el alcance de los objetivos del módulo. Para cada unidad didáctica se seguirá el procedimiento de evaluación descrito en los apartados anteriores, teniendo en cuenta además el criterio de notas que nos hemos marcado.

Deste esta postura, un alumno, para aprobar, es decir para obtener un 5, ha de tener los conocimientos mínimos de las capacidades terminales del módulo. Para ello se evaluarán los contenidos de cada unidad didáctica (procedimentales, conceptuales y actitudinales) como al resto de los alumnos.

El contenido mínimo significa que el alumno tiene un conocimiento básico del contenido a evaluar, aunque no tenga la soltura suficiente para tomar decisiones sobre nuevas tareas. El criterio de las notas de evaluación sobre los contenidos se ha explicado en el apartado sobre criterios de evaluación.

11 Bibliografía

La bibliografía se apoyará fundamentalmente en recursos Web que se irán actualizando a

lo largo del módulo por los alumnos y el profesor y disponibles en la página Web de la asignatura. La creación de recursos dinámicos en un área que evoluciona tan rápido como la informática es relativamente sencillo con el uso de Internet.

El uso de Internet como fuente de recursos para el módulo, se incluye dentro del planteamiento general del centro descrito en el PEC y PCC. Esta situación va evolucionando a medida que avanza el curso, aunque algunas de las propuestas iniciales son revisar en sites² como los que mostramos a continuación:

- <http://www.tldp.org> Linux Documentation Project. Documentación de Manuales, formas de documentación, procesos y sistemas operativos (es.tldp.org es el proyecto de traducción)
- <http://www.w3c.org> World Wide Web Consortium. Estándares y Arquitectura Internet
- <http://www.postgresql.org> Servidor de BDR PostgreSQL
- <http://www.microsoft.com/> Arquitectura Windows y DOS, Ayuda de Sistemas operativos y librerías

Se utilizarán también los manuales de usuario, de administrador y las ayudas instaladas localmente para cada Sistema Operativo y las aplicaciones software que utilicemos. Especialmente para cada lenguaje de programación se utilizará la ayuda que ofrezca la comunidad de soporte.

También se utilizarán las especificaciones del lenguaje utilizando la documentación pública, por ejemplo RFCs y especificaciones de W3C e IETF. Sin embargo no se recomendará la compra de ningún libro.

² Ponemos el nombre del site no las páginas en concreto.

Algunos libros interesantes de consulta y que permiten profundizar más en los conceptos de Arquitectura Web son:

- Jakob Nielsen, Designing Web Usability. ISBN 1-56205-810-X. Diciembre 1999 Ed New Riders
- Philip Greenspun. Philip and Alex's Guide to Web Publishing ISBN 1-55860-534-7. Año 1999. Ed Morgan Kaufmann