

Estructuras de Datos

Curso 2017 / 18

Profesores:

Teoría: Francisco de Borja Rodríguez, Pablo Castells, Roberto Marabini,

Prácticas: Julia Díaz, Álvaro del Val, Rodrigo Castro, Ángel Mora,

Simone Santini

Escuela Politécnica Superior

Universidad Autónoma de Madrid

Datos generales de la asignatura

- ♦ Profesor del grupo 121 – Francisco de Borja Rodríguez Ortiz
 - Despacho B-328
 - Tutorías: por cita a petición del estudiante
- ♦ Profesores de la asignatura
 - Teoría: Francisco de Borja Rodríguez, Pablo Castells, **Robereto Marabini**
 - Prácticas: Julia Díaz, Álvaro del Val, Rodrigo Castro, Ángel Mora, Simone Santini
 - Coordinador asignatura: **Robereto Marabini**
- ♦ Horario Teoría (Grupo 121)
 - Martes 09h a 10h
 - Miércoles 09h a 11h
- ♦ Horario Prácticas (Grupos EDAT-1211, EDAT-1212 y EDAT-1213)
 - Viernes 09h a 11h y viernes de 11h a 13h
- ♦ Prueba final
 - Lunes 08 de enero 2018 (Mañana)

Datos generales de la asignatura

- ♦ Leer la guía de la asignatura

<http://www.uam.es/ss/Satellite/EscuelaPolitecnica/es/estudios/Page/sinContenido/repositorio-de-guias-docentes.htm>

¿Dé qué trata la asignatura?

Un primer curso de introducción a **bases de datos**, tanto a un nivel lógico como a un nivel de implementación

¿Qué es una base de datos?

¿Qué es una base de datos?

- ♦ Es un sistema informático que permite, organiza y administra el acceso a datos de una forma eficaz.
- ♦ Una fuente de información estructurada almacenada en memoria secundaria
- ♦ Habitualmente masiva en volumen de datos, variedad y complejidad de las estructuras
- ♦ Gestionada mediante tecnologías con un alto nivel de generalidad, desarrolladas y estandarizadas al efecto
- ♦ En la asignatura estudiaremos:
 - Tecnologías de gestión de bases de datos –SQL y nociones prácticas
 - Una introducción a las metodologías de diseño –modelo E/R
 - La base formal sobre la que se asientan las tecnologías de bases de datos –modelo relacional, formas normales, cálculo y álgebra relacional
 - La implementación de tecnologías de gestión de bases de datos –registros, índices, árboles B. etc..

Tipos de modelo de Bases de Datos

- ♦ BDs Jerarquicas (estructura de árbol): Adabas, GT.M, IMS, Focus
- ♦ BDs de Red (un nodo puede tener varios padres)
- ♦ BDs Transaccionales (envío y recepción de datos a grandes velocidades)
- ♦ **BDs Relacionales (uso de relaciones).**
- ♦ BDs Multidimensionales
- ♦ BDs Orientadas a objetos
- ♦ BDs Documentales
- ♦ BDs Deductivas

TODAS se manejan mediante un sistema de gestión de Bases de Datos: usaremos principalmente PostgreSQL que es un SGBD relacional orientado a objetos y libre.

Un ejemplo

- ♦ Supongamos que vamos a desarrollar una aplicación para escuchar música, con redes sociales
- ♦ Queremos manejar información sobre: canciones, artistas, álbumes, usuarios, grupos, eventos, registro de accesos a canciones...
- ♦ Esta información:
 - Tiene **estructura**: p.e. un usuario tiene propiedades (nombre, nick, email, etc.) y relaciones (amigos, artistas favoritos, escuchas a canciones, etc.)
 - Tiene que almacenarse de forma **persistente**
 - Almacenamiento en disco
 - Se tiene que poder inferir información de la almacenada y **crear nueva información**
 - Puede ser **masiva**: millones de usuarios, millones de canciones, miles de millones de registros de escuchas
 - Inviabile carga completa en RAM
 - Acceso continuo a disco en tiempo de ejecución

Un ejemplo

- ♦ Para la aplicación necesitamos:
 - Acceso (consulta) eficiente a los datos: mostrar a un usuario su lista de amigos, la información de una canción, etc.
 - Actualización eficiente de datos: añadir amigos, guardar un log de escuchas, etc.
 - Acceso concurrente, robustez, seguridad...
- ♦ Solución: guardar todos los datos en fichero(s) en disco y programar la funcionalidad de acceso
 - No perder la estructura de los datos
 - El acceso a memoria secundaria es muy costoso
 - El problema requiere soluciones elaboradas no triviales (las estudiaremos!)
- ♦ En rigor esto ya se podría considerar una base de datos!

- ♦ La **dificultad y complejidad de un desarrollo desde cero** son considerables, y por otro lado...
- ♦ Una buena parte del problema a resolver se **repite** en muchos dominios: gestión de personal, gestión de inventarios, gestión universitaria, bibliotecas, reserva de viajes, banca, finanzas, contabilidad, competiciones deportivas, correo electrónico, buscadores web...
- ♦ Por tanto hay ciertos aspectos comunes:
 - Diseño de las estructuras: tabular
 - Almacenamiento físico
 - Consulta y actualización
 - Integridad, robustez, concurrencia, seguridad...

} Lo estudiaremos (implementación)
- ♦ Tecnología de bases de datos
 - Herramientas {
 - Lenguaje estándar de creación, consultas y actualización: SQL
 - Motores de ejecución eficiente de las sentencias SQL
 - Interfaz de usuario
 - Interfaz de programación (ODBC, JDBC, PHP)

} SGBD
 - Método y teoría {
 - Metodologías de diseño: modelo Entidad / Relación
 - Paradigmas: modelo relacional, formas normales, cálculo y álgebra

Interfaz de usuario



Software
aplicación

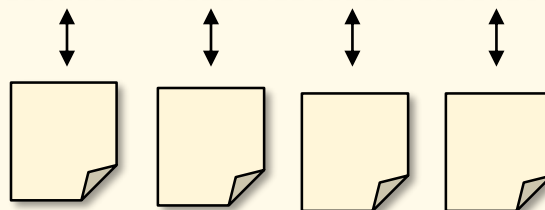


Programador
aplicación

Lógica de la aplicación

Acceso y gestión de datos

Archivos de datos
Memoria externa



Arquitectura ad hoc (se ha creado especialmente para esta situación concreta, por tanto, no generalizable ni utilizable para otros propósitos)

Usuario final

Interfaz de usuario



Programador aplicación

Lógica de la aplicación

Sentencias SQL

ODBC, JDBC, PHP...

API BD

Almacenamiento
Consultas
Actualización

SGBD

Bases de datos

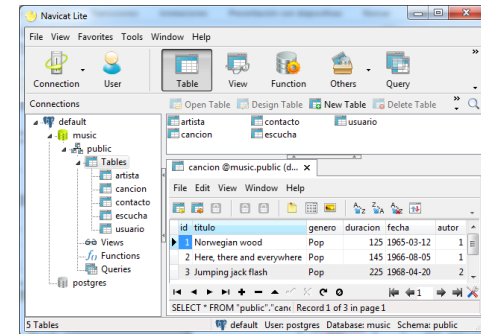


Arquitectura BD

Software aplicación

Administrador

Entorno / herramientas de administración



Navicat
SQLYog
etc.

- ♦ La dificultad y complejidad de un desarrollo desde cero son considerables, y por otro lado...
 - ♦ Una buena parte del problema a resolver se repite en muchos dominios: gestión de personal, gestión de inventarios, gestión universitaria, bibliotecas, reserva de viajes, banca, finanzas, contabilidad, competiciones deportivas, correo electrónico, buscadores web...
 - Diseño de las estructuras: tabular
 - Almacenamiento físico
 - Consulta y actualización
 - Integridad, robustez, concurrencia, seguridad...
- } Lo estudiaremos (implementación)
-
- ♦ Tecnología de bases de datos
- | | | | | |
|--------------------|---|--|---|------|
| Herramientas | { | <ul style="list-style-type: none"> – Lenguaje estándar de creación, consultas y actualización: SQL – Motores de ejecución eficiente de las sentencias SQL – Interfaz de usuario – Interfaz de programación (ODBC, JDBC, PHP) | } | SGBD |
| Método
y teoría | { | <ul style="list-style-type: none"> – Metodologías de diseño: modelo Entidad / Relación – Paradigmas: modelo relacional, formas normales, cálculo y álgebra | } | |

- ♦ La dificultad y complejidad de un desarrollo desde cero son considerables, y por otro lado...
- ♦ Una buena parte del problema a resolver se repite en muchos dominios: gestión de personal, gestión de inventarios, gestión universitaria, bibliotecas, reserva de viajes, banca, finanzas, contabilidad, competiciones deportivas, correo electrónico, buscadores web...
 - Diseño de las estructuras: tabular
 - Almacenamiento físico
 - Consulta y actualización
 - Integridad, robustez, concurrencia, seguridad...

} Lo estudiaremos (**implementación**)
- ♦ Tecnología de bases de datos

Herramientas

{

- Lenguaje estándar de creación, **consultas** y actualización: **SQL**
 - Motores de ejecución eficiente de las sentencias SQL
 - Interfaz de usuario
 - Interfaz de programación (ODBC, JDBC, PHP)

}

SGBD

Método
y teoría

{

- Metodologías de diseño: **modelo Entidad / Relación**
 - Paradigmas: **modelo relacional**, **formas normales**, **cálculo** y **álgebra**

Temario

- ♦ Introducción y fundamentos
- ♦ Introducción a SQL
- ♦ Modelo Entidad / Relación
- ♦ Modelo relacional
- ♦ Diseño relacional: formas normales
- ♦ Consultas: cálculo y álgebra relacional
- ♦ Implementación de bases de datos
 - Estructura física: campos y registros
 - Indexación: índices simples, árboles B, hashing
 - Compresión

Temario

- ♦ Introducción y fundamentos
- ♦ Introducción a SQL
- ♦ Modelo Entidad / Relación
- ♦ Modelo relacional
- ♦ Diseño relacional: formas normales
- ♦ Consultas: cálculo y álgebra relacional
- ♦ Implementación de bases de datos
 - Estructura física: campos y registros
 - Indexación: índices simples, árboles B, hashing
 - Compresión

Objetivos

Nivel práctico

- ♦ Uso / administración básica de BDs
- ♦ Análisis, diseño y creación de BDs
- ♦ Manipulación de BDs, consultas
- ♦ Manejo de SQL

Nivel conceptual

- ♦ Comprensión de los principios y formalismos sobre los que se asientan las tecnologías de BDs

Nivel técnico

- ♦ Conocimiento de las técnicas de implementación interna de un SGBD

Bibliografía

- ☞ Fundamentos de sistemas de bases de datos. Ramez Elmasri, Shamkant Navathe. Pearson Addison Wesley, 2007. INF/681.31.65/ELM.
- ◆ Database Management Systems. Raghu Ramakrishnan, Johannes Gehrke. McGraw-Hill, 2003. INF/C6160/RAM.
- ◆ Database Systems: The Complete Book. Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom. Prentice Hall, 2008.
- ◆ Fundamentos de diseño de bases de datos. Abraham Silberschatz. McGraw-Hill, 2007. INF/681.31.65/SIL.
- ☞ Estructuras de archivos: un conjunto de herramientas conceptuales. Michael J. Folk, Bill Zoellick. Addison-Wesley, 1992. INF/681.3.01/FOL.

Relación con otras asignaturas

- ♦ Programación I y II, Análisis de algoritmos
 - Programación y desarrollo de software → técnicas específicas para almacenamiento y acceso a datos estructurados masivos en disco
 - Algoritmia en RAM → revisión para datos en disco
- ♦ Análisis y Diseño de Software
 - Modelado de datos: UML
- ♦ Sistemas Informáticos I
 - Optimización de consultas, interfaces de programación
 - Bases de datos distribuidas
 - Transacciones
- ♦ Ingeniería del Software
 - Análisis y diseño de aplicaciones

Evaluación

- Ambas partes, teoría y prácticas se puntúan sobre 10 puntos.
- La nota final de la asignatura se obtiene de las notas de teoría y prácticas por medio de la ecuación:
Calificación Total: $0.4 * \text{Prácticas} + 0.6 * \text{Teoría}$
- Para aprobar la asignatura es obligatorio obtener una nota mayor o igual a 5 puntos, tanto en la parte de teoría como en las prácticas. En caso contrario, la nota final en actas será
Calificación: $0,4 * \min(5, \text{Prácticas}) + 0,6 * \min(5, \text{Teoría})$
- La nota correspondiente a la parte de Teoría es la que resulta de:
 - ✓ La calificación de la prueba final (60%).
 - ✓ La calificación de las pruebas/actividades/ejercicios intermedios (al menos dos) (40%).
- La media con la evaluación continua sólo se aplica si mejora la nota del examen final (en otro caso el peso del examen final sube al 100%)
- Convalidación de prácticas: >7 en Prácticas y >3 Teoría.

60% Teoría ≥ 5	40% Evaluación continua	25% Pruebas ejercicios (entrega de ejercicios al finalizar los diferentes partes, al menos 12+12)
		75% Examen intermedio (22 de noviembre). Examen intermedio eliminatorio con nota > 6 . En este caso la nota obtenida en la prueba intermedia se mapea en la parte correspondiente del examen final (en este caso el examen final cuenta el 90 % de la nota de teoría incluida la nota del parcial mapeada).
	60% Examen final (lunes 08 de enero 2018 (Mañana))	
40%	Prácticas ≥ 5 (cada práctica ≥ 3)	