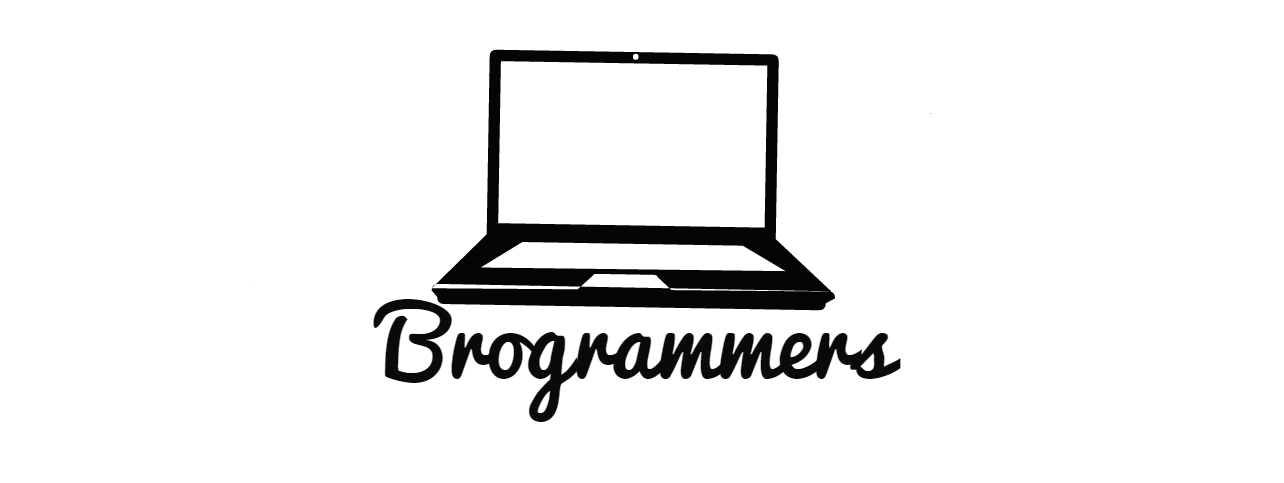
Trabajo Voluntario: Sistemas de Bases de Datos.



Héctor Gonzalo Andrés.

Konstantin Danielov Kostandev.

**Tabla de contenido**

[1 Introducción 3](#_Toc419120758)

[1.1 Contexto del trabajo y objetivos 3](#_Toc419120759)

[1.2 Herramientas utilizadas 3](#_Toc419120760)

[2 Modelo Entidad-Relación 4](#_Toc419120761)

[3 Descripción de las tablas generadas 4](#_Toc419120762)

[4 Estructura de la aplicación 6](#_Toc419120763)

[5 Capturas de pantalla de las ejecuciones 7](#_Toc419120764)

[6 Código fuente C 13](#_Toc419120765)

[7 Scripts SQL 30](#_Toc419120766)

# Introducción

Hemos optado por utilizar una mezcla entre la solución propuesta en Studium y nuestra propia respuesta del trabajo de Diseño de bases de datos para solucionar los errores existentes en nuestra primera versión, además de poder reutilizar nuestro script de relleno de la base.

## Contexto del trabajo y objetivos

El proyecto asignado consiste en el desarrollo de una base de datos para la gestión de preguntas tipo test que varios profesores de asignaturas similares han ido ideando a lo largo de su actividad docente. Estará estructurado teniendo en cuenta las asignaturas que imparten, incluyendo el temario de las mismas, y las preguntas tipo test que se refieren a cada tema.

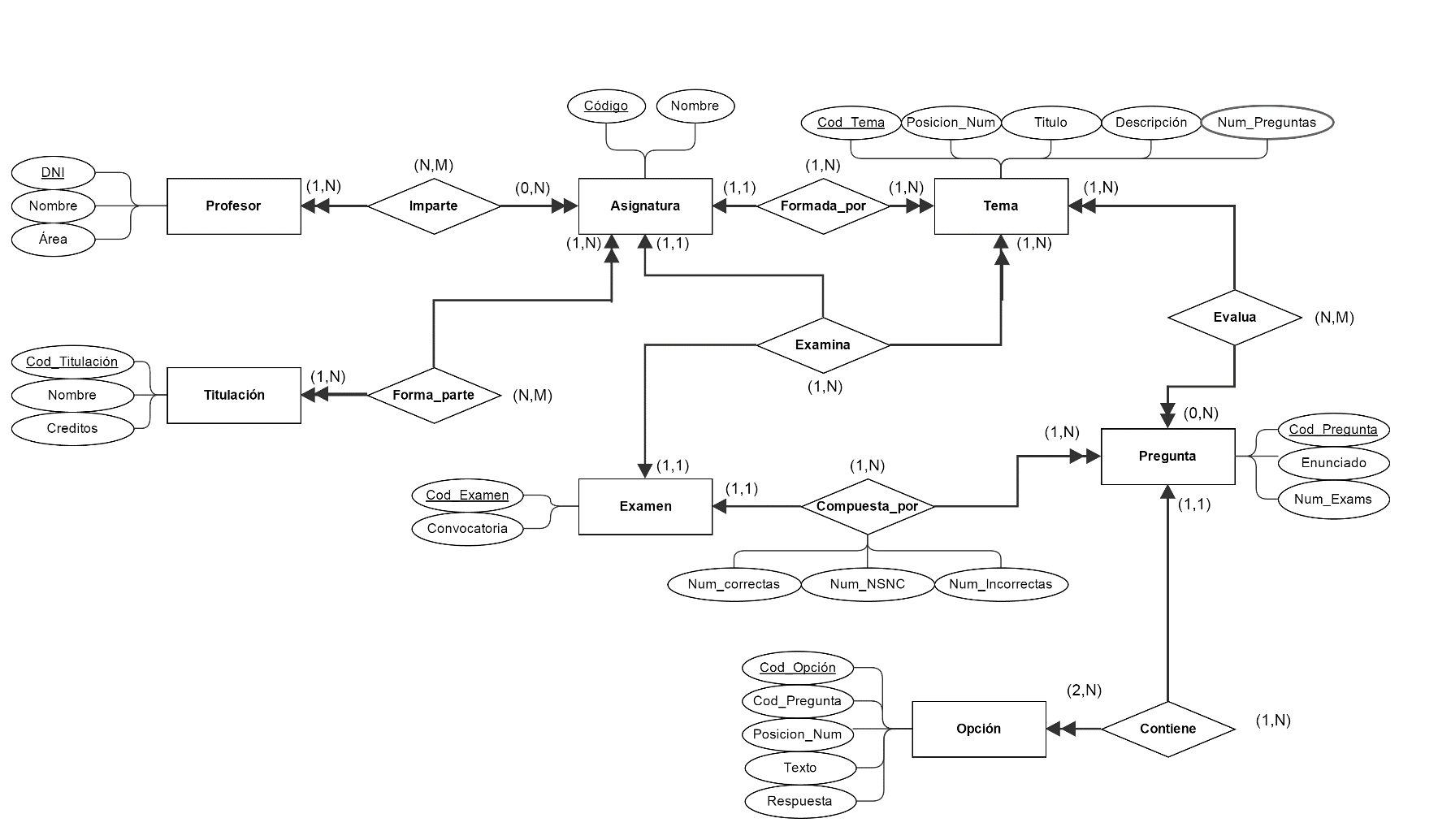
Posteriormente mostraremos de manera más detallada todas las tablas implicadas en la base de datos teniendo en cuenta las restricciones de integridad, explicando adecuadamente todos sus campos y los valores que pueden tomar. Los principales objetivos a realizar de nuestra aplicación serán:

* La creación del script SQL: Será necesario crear un script SQL para crear la base de datos con un pequeño juego de datos.
* Desarrollar un programa modular en C con SQL inmerso: Dicho programa incorpora funciones adicionales a la base de datos.

## Herramientas utilizadas

* Ubuntu Server 14.04: Se utilizó un servidor corriendo Ubuntu 14.04 x64 para poder permitir a ambos integrantes del equipo hacer pruebas de los script SQL fuera de la red Eduroam.
* Oracle 11G: Versión de Oracle instalada en el servidor usada tanto para prueba de scripts SQL como para consultas desde SQL inmerso en C.
* Oracle SQL Developer: Se utilizó como IDE de desarrollo de los scripts SQL y como herramienta de administración de la base de datos.
* Sublime Text: Usado para escribir el programa en C.
* GitHub: Se utilizó un repositorio privado para poder administrar el código y trabajar a distancia de forma sencilla y segura.
* Microsoft Word: Herramienta utilizada para organizar toda la información y hacer la memoria.

# Modelo Entidad-Relación



# Descripción de las tablas generadas

Hemos creado las siguientes tablas en nuestra base de datos:

PROFESOR (DNI, Nombre, Área)

Clave Primaria: DNI

ASIGNATURA (Código, Nombre, Titulación)

Clave Primaria: Código

Clave Candidata: Nombre, Titulación (porque no puede haber dos asignaturas con el mismo nombre en la misma titulación)

IMPARTE (DNI, CODASIGNAT)

Clave Primaria: DNI, CODASIGNAT.

Clave Externa: DNI sobre PROFESOR.

Clave Externa: CODASIGNAT sobre ASIGNATURA.

TEMA (Cod\_Tema, Posición\_Num, Título, Descr, Num\_preguntas)

Clave Primaria: Cod\_Tema.

Clave Candidata: Posición\_Num, Título (porque no puede haber dos temas con el mismo título en la misma posición numerada)

FORMADO\_POR (Cod\_Asig, Cod\_Tema)

Clave Primaria: Cod\_Asig, Cod\_Tema.

Clave Externa: Cod\_Asig sobre ASIGNATURA.

Clave Externa: Cod\_Tema sobre TEMA.

TITULACION (Cod\_Titulación, Nombre, Créditos)

Clave Primaria: Cod\_Titulación.

(Nombre podría haber sido clave pero por hacer más sencillo el trabajo decidimos introducir el campo Cod\_Titulación)

FORMA\_PARTE (Cod\_Titulación, Cod\_Asign)

Clave Primaria: Cod\_Titulación, Cod\_Asign.

Clave Externa: Cod\_Titulación sobre TITULACIÓN.

Clave Externa: Cod\_Asign sobre ASIGNATURA.

EXAMEN (Cod\_Examen, Convocatoria)

Clave Primaria: Cod\_Examen.

EXAMINA (Cod\_Examen, Cod\_Asign, Cod\_Tema)

Clave Primaria: Cod\_Examen, Cod\_Asign, Cod\_Tema.

Clave Externa: Cod\_Examen sobre EXAMEN.

Clave Externa: Cod\_Asign sobre ASIGNATURA.

Clave Externa: Cod\_Tema sobre TEMA.

PREGUNTA (Cod\_Pregunta, Enunciado)

Clave Primaria: Cod\_Pregunta.

EVALUA (Cod\_Pregunta, Cod\_Tema)

Clave Primaria: Cod\_Pregunta, Cod\_Tema.

Clave Externa: Cod\_Pregunta sobre PREGUNTA.

Clave Externa: Cod\_Tema sobre TEMA.

COMPUESTA\_POR (Cod\_Examen, Cod\_Pregunta, Num\_Correctas, Num\_NSNC, Num\_Incorrectas)

Clave Primaria: Cod\_Examen, Cod\_Pregunta.

Clave Externa: Cod\_Pregunta sobre PREGUNTA.

Clave Externa: Cod\_Examen sobre EXAMEN.

OPCIÓN (Cod\_Opción, Cod\_Pregunta, Posición\_Num, Texto, Respuesta)

Clave Primaria: Cod\_Opción.

Clave Externa: Cod\_Pregunta sobre PREGUNTA.

Además hemos utilizado los siguientes datos para rellenar cada una de las tablas con datos que sean coherentes:

En Titulación se han utilizado 5 titulaciones cargadas con 10 asignaturas repartidas entre ellas, recogiendo dicha relación en Forma\_parte.

También hemos cargado los datos de 5 profesores que se relacionan con las 10 asignaturas previamente cargadas, recogiendo la relación en Imparte.

Para terminar se han introducido 25 temas relacionados con Pregunta mediante Evalúa y además con Asignatura y Examen mediante Examina.

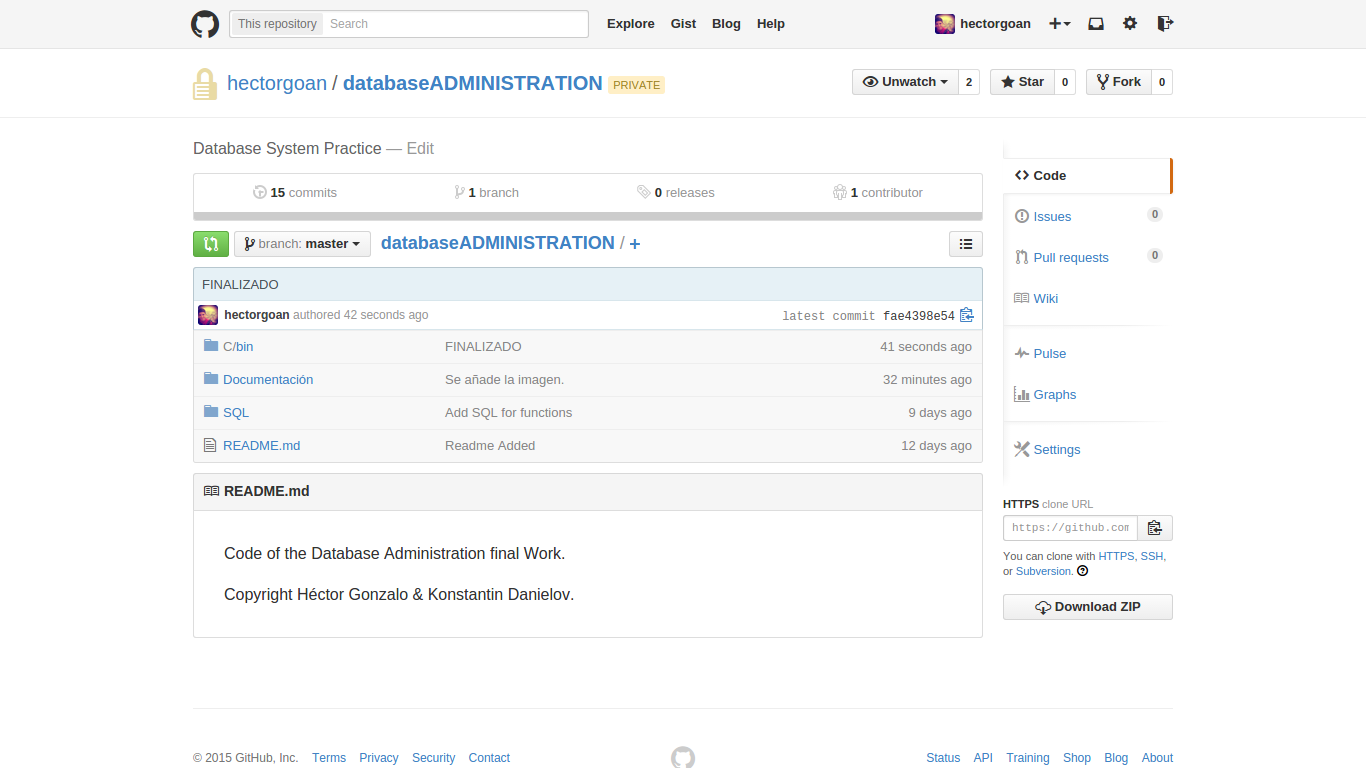
# Estructura de la aplicación

Nuestra aplicación se estructura en tres archivos, dos .sql y un .pc. El primer .sql es para la creación de las tablas y el segundo es para rellenar dichas tablas.

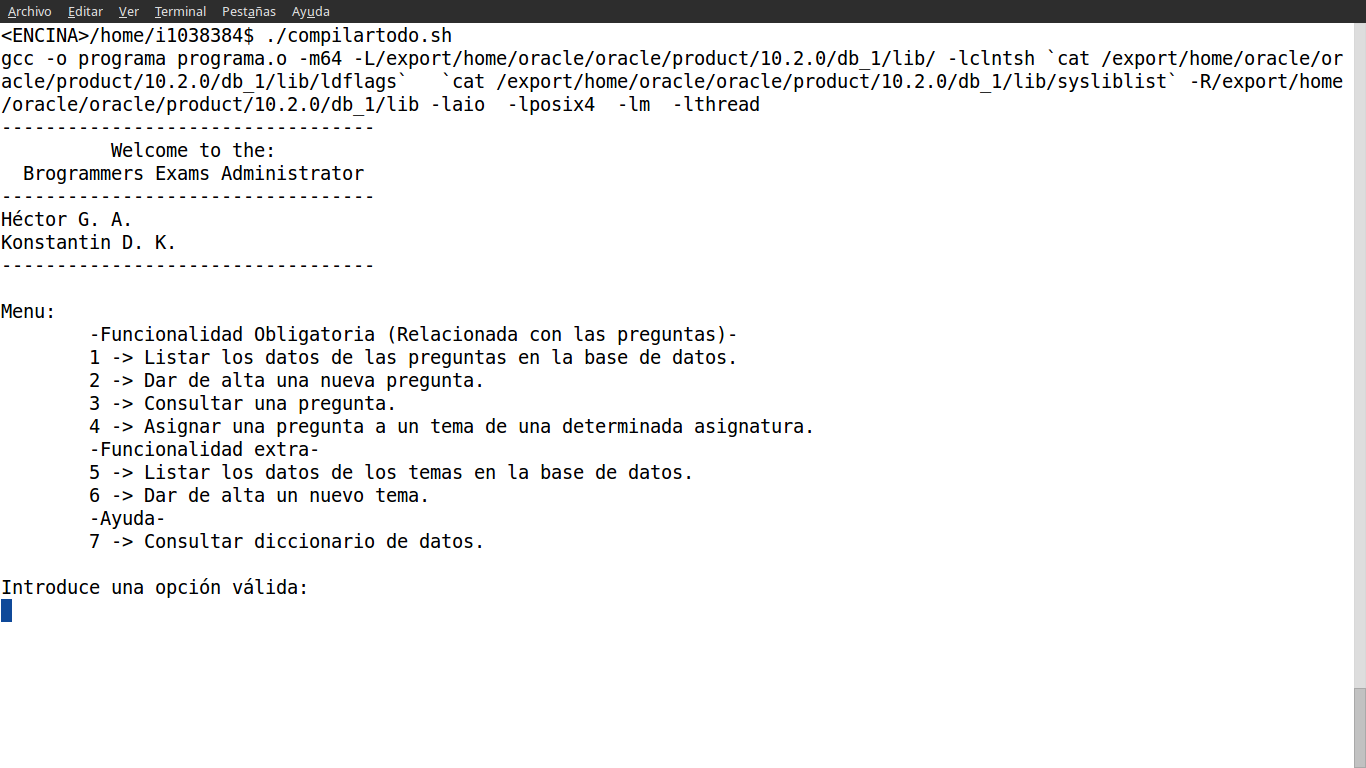
El .pc es el archivo de c con SQL inmerso, en el que recogemos las funcionalidades requeridas por el programa para trabajar con la base de datos creada anteriormente con los .sql.

# Capturas de pantalla de las ejecuciones

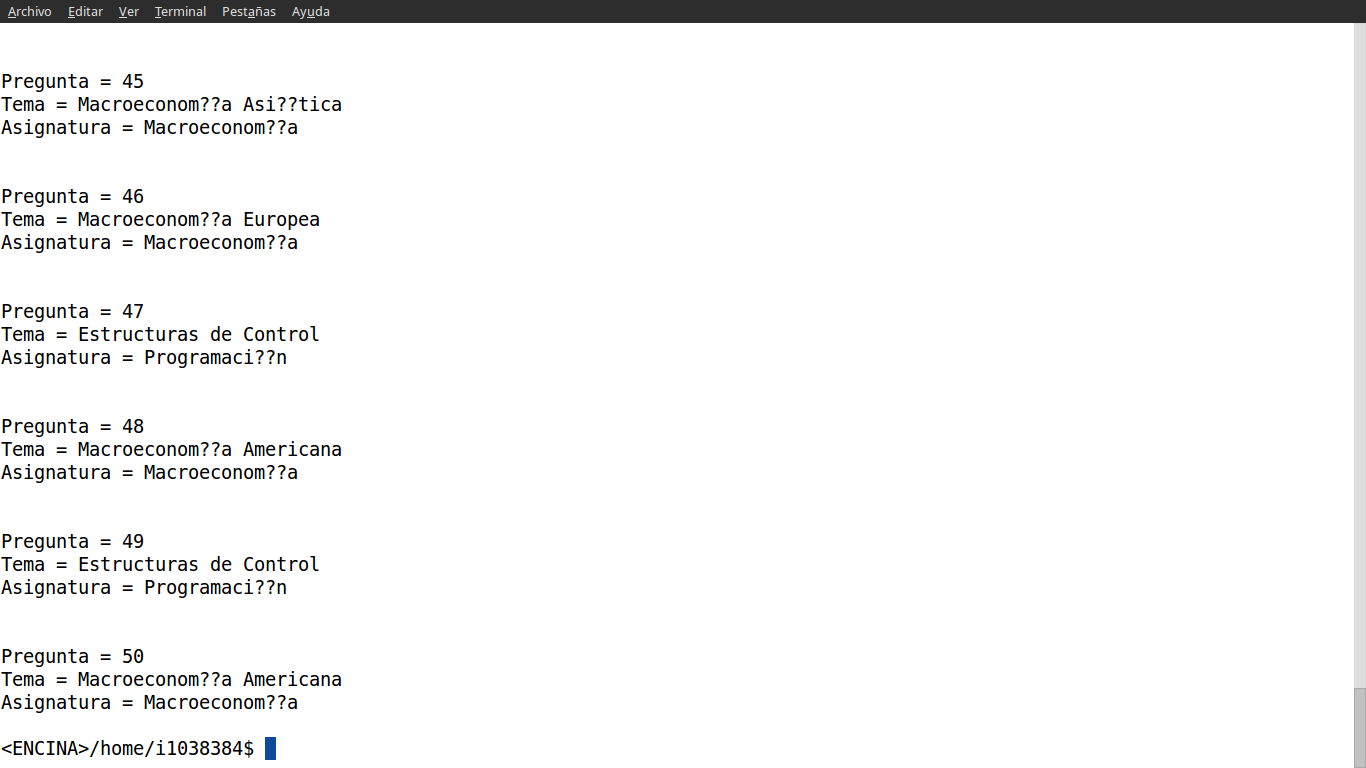
Para el control de versiones utilizamos un repositorio privado de GitHub.



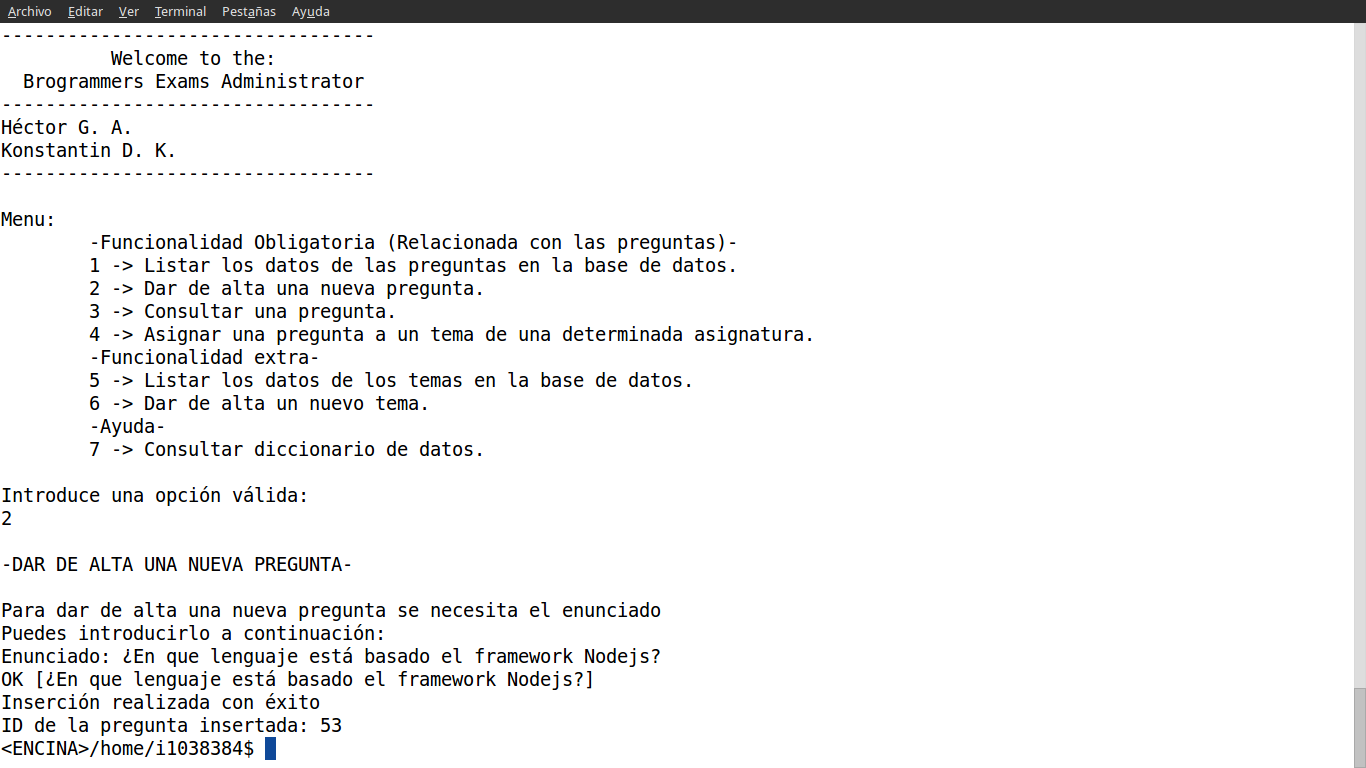
Al ejecutar el programa lo primero que aparece es un menú con las funcionalidades de la aplicación que hemos creado. A parte de las cuatro propuestas hemos decidido añadir otras tres.



Al ejecutar la primera opción obtenemos un listado de las preguntas que hay en la base de datos junto con su información.



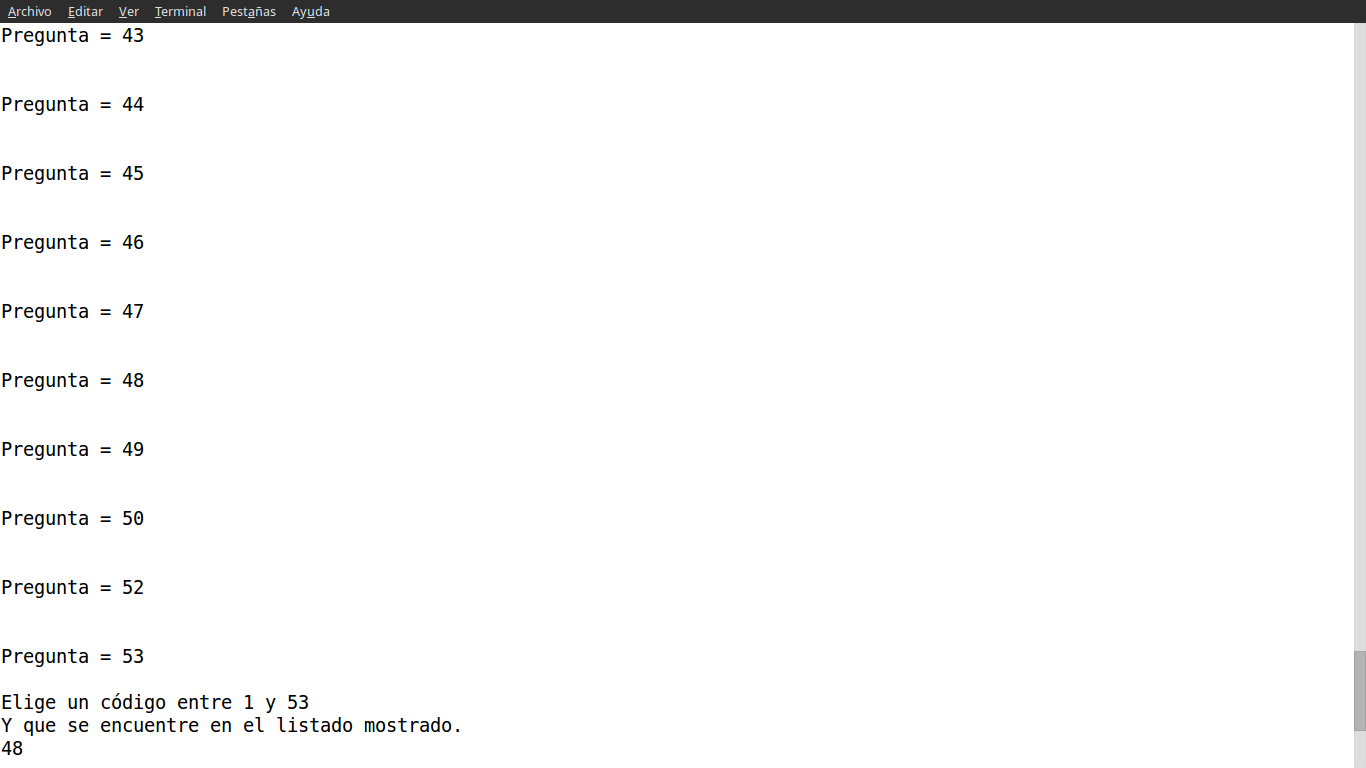
Al ejecutar la segunda opción podemos dar de alta una nueva pregunta en la base de datos. Nuestra aplicación halla el código de la pregunta a insertar sumándole uno al último que exista en la tabla pregunta. De esta manera solo se nos pedirá el enunciado de la pregunta. Se adjunta captura de SQL con la pregunta insertada.

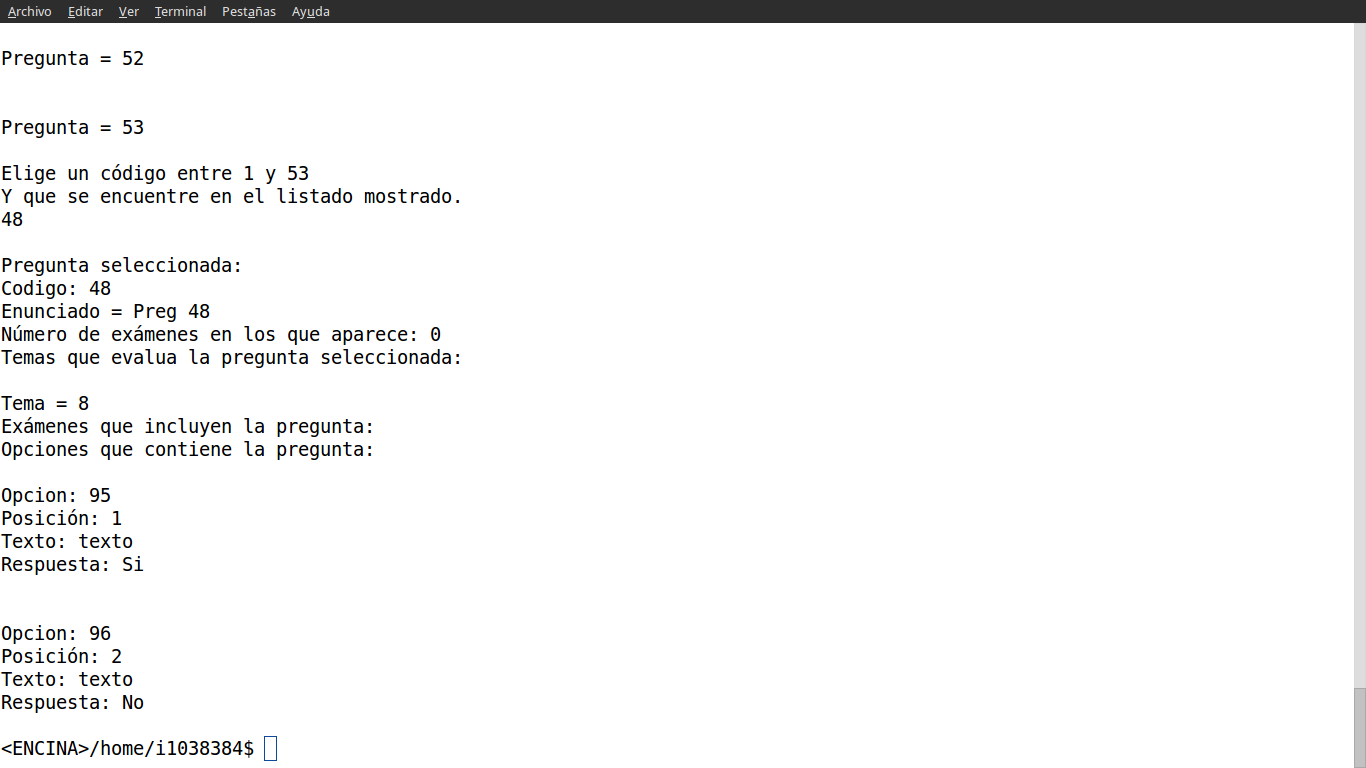




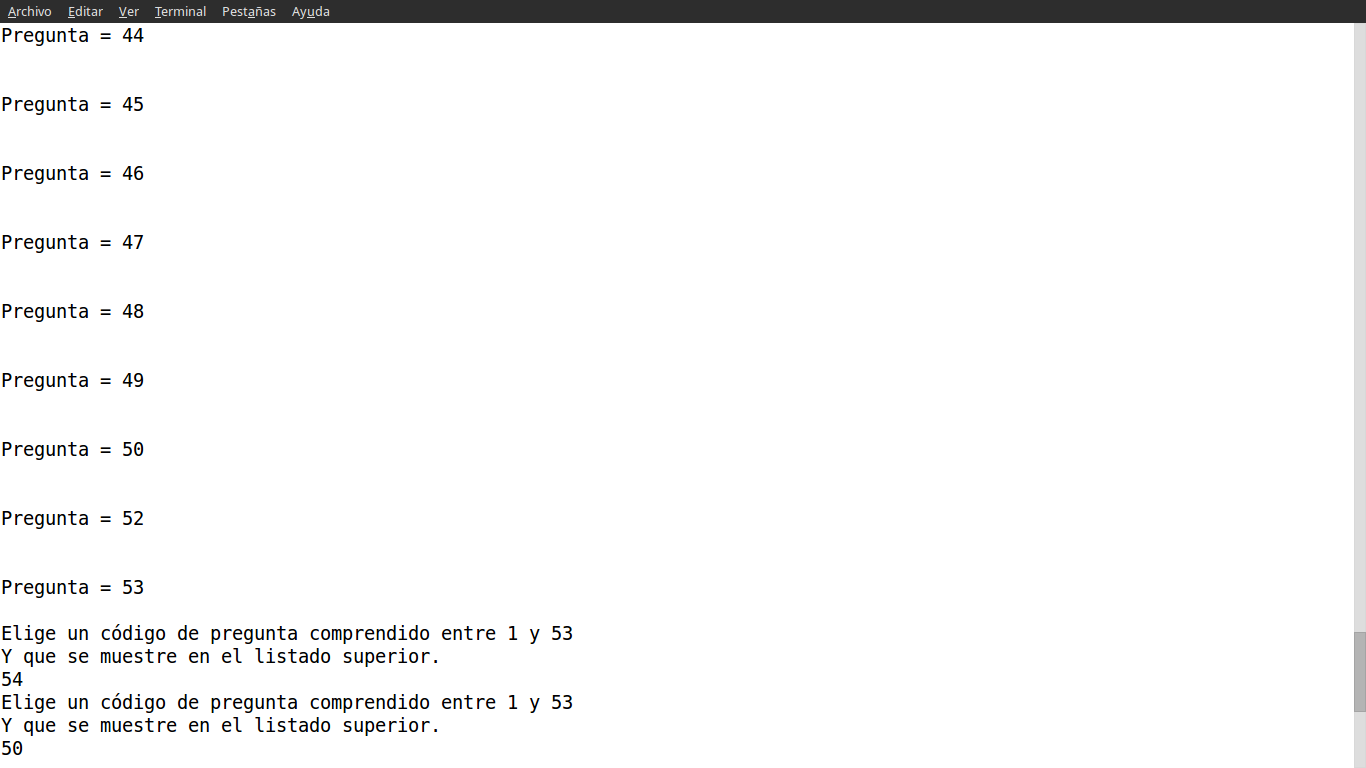
Al ejecutar la tercera opción podemos consultar los datos de una determinada pregunta.

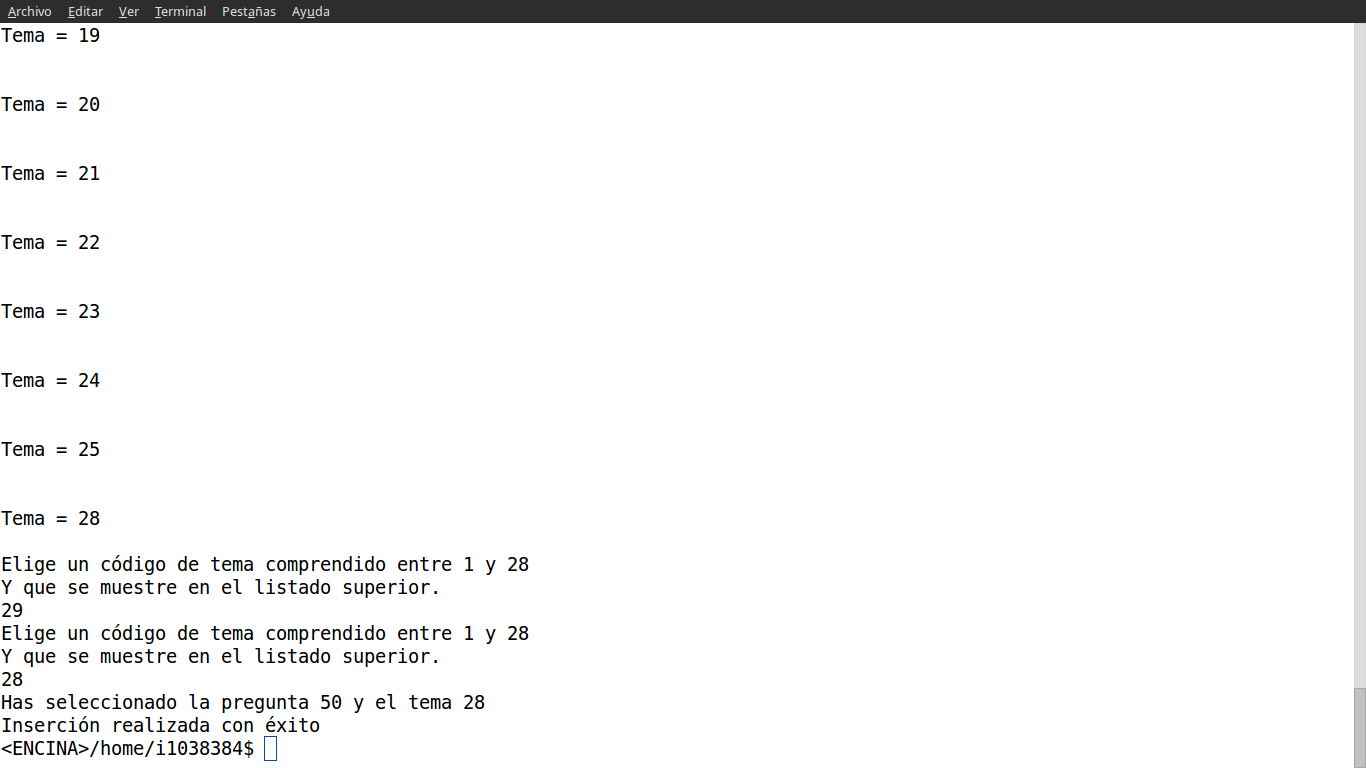
Para ello se nos pide el código.

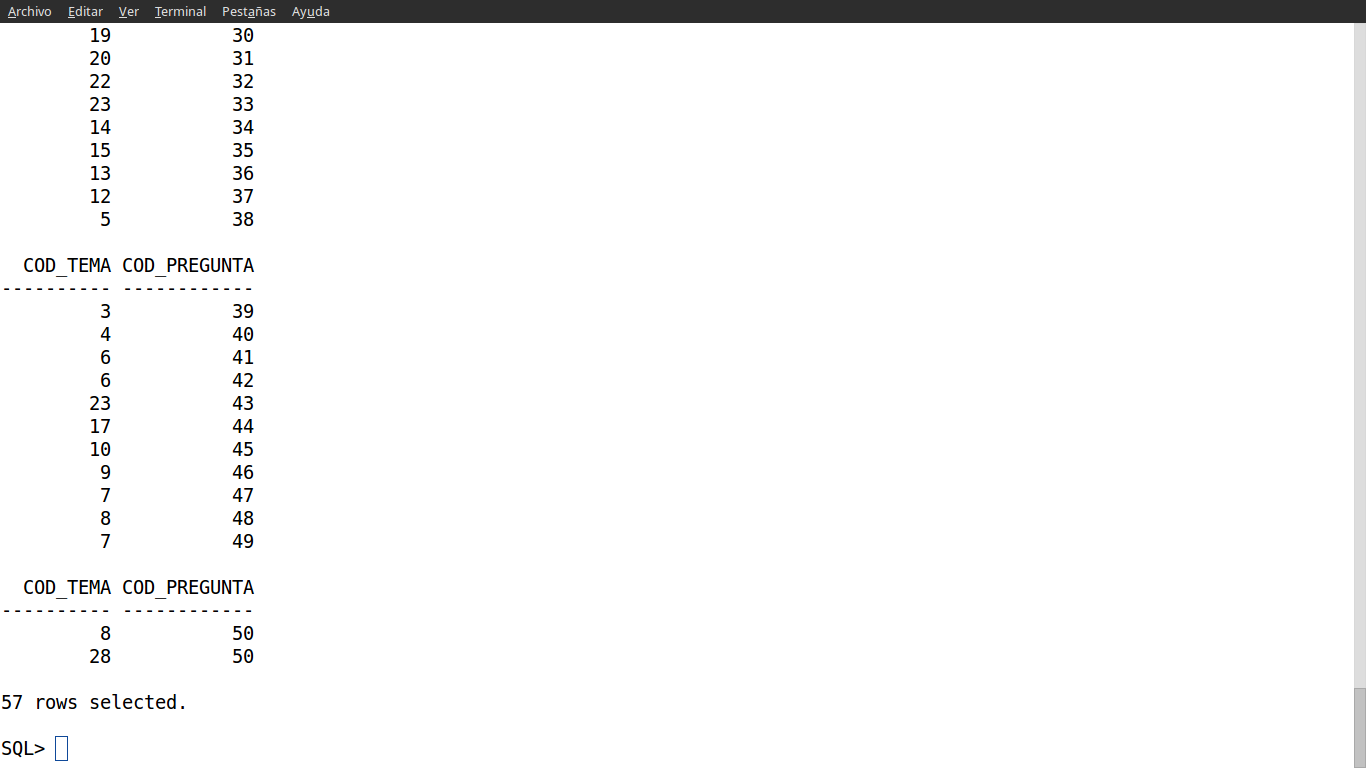




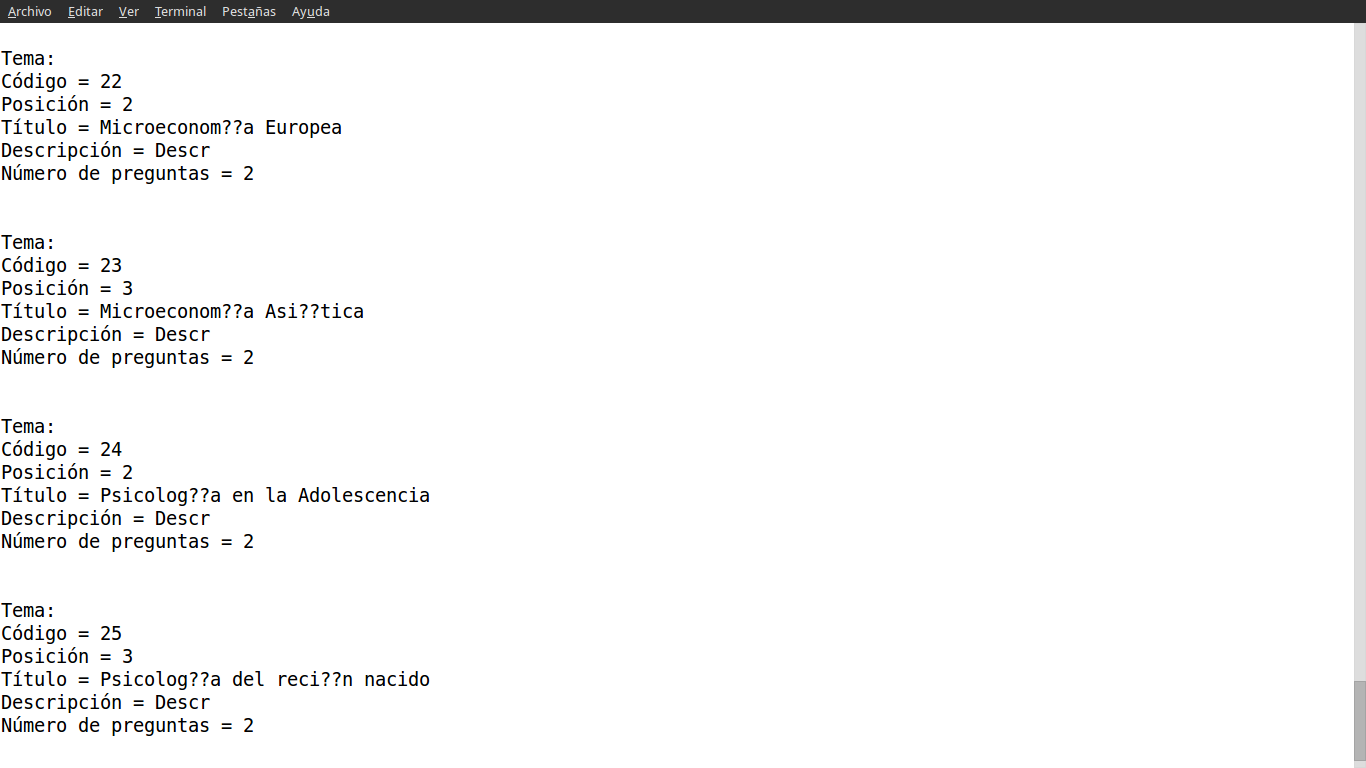
Al ejecutar la cuarta opción nos permite asignar una pregunta a un tema de una determinada asignatura.



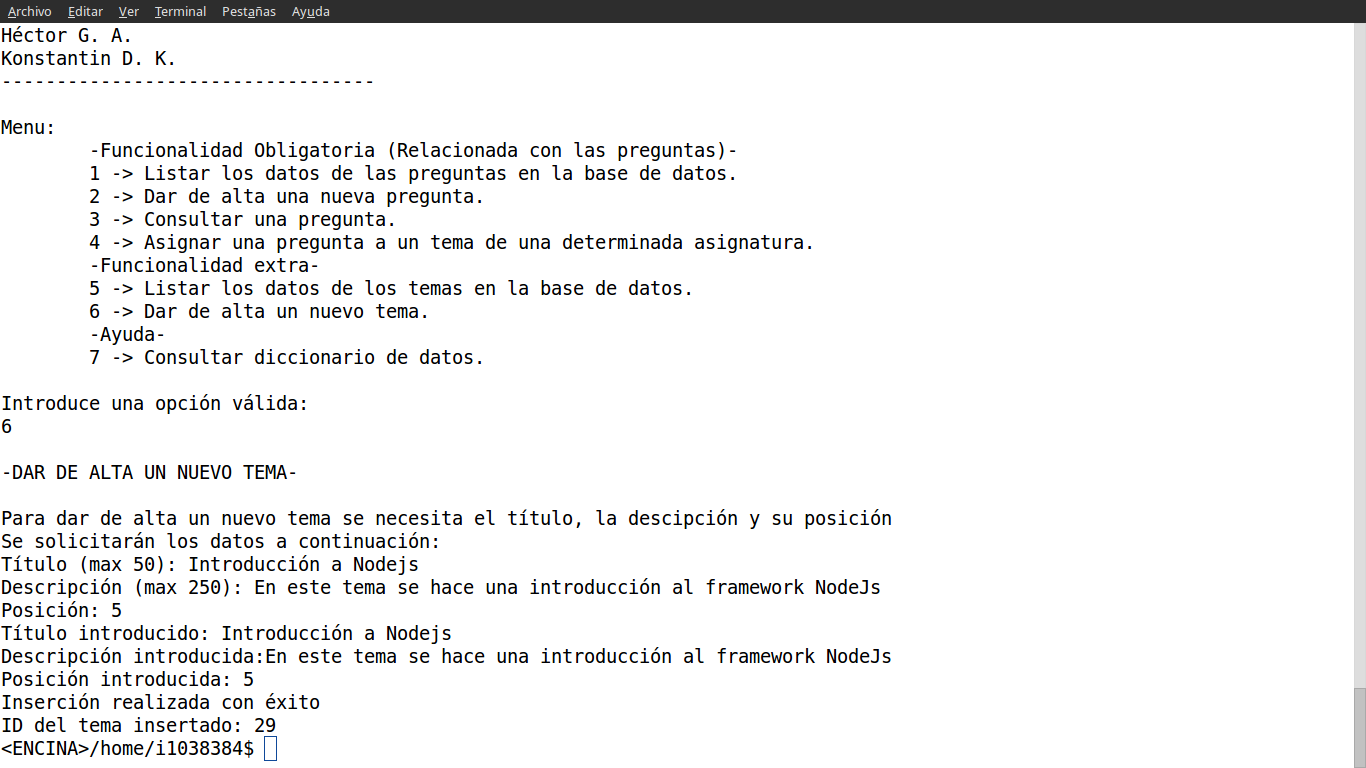




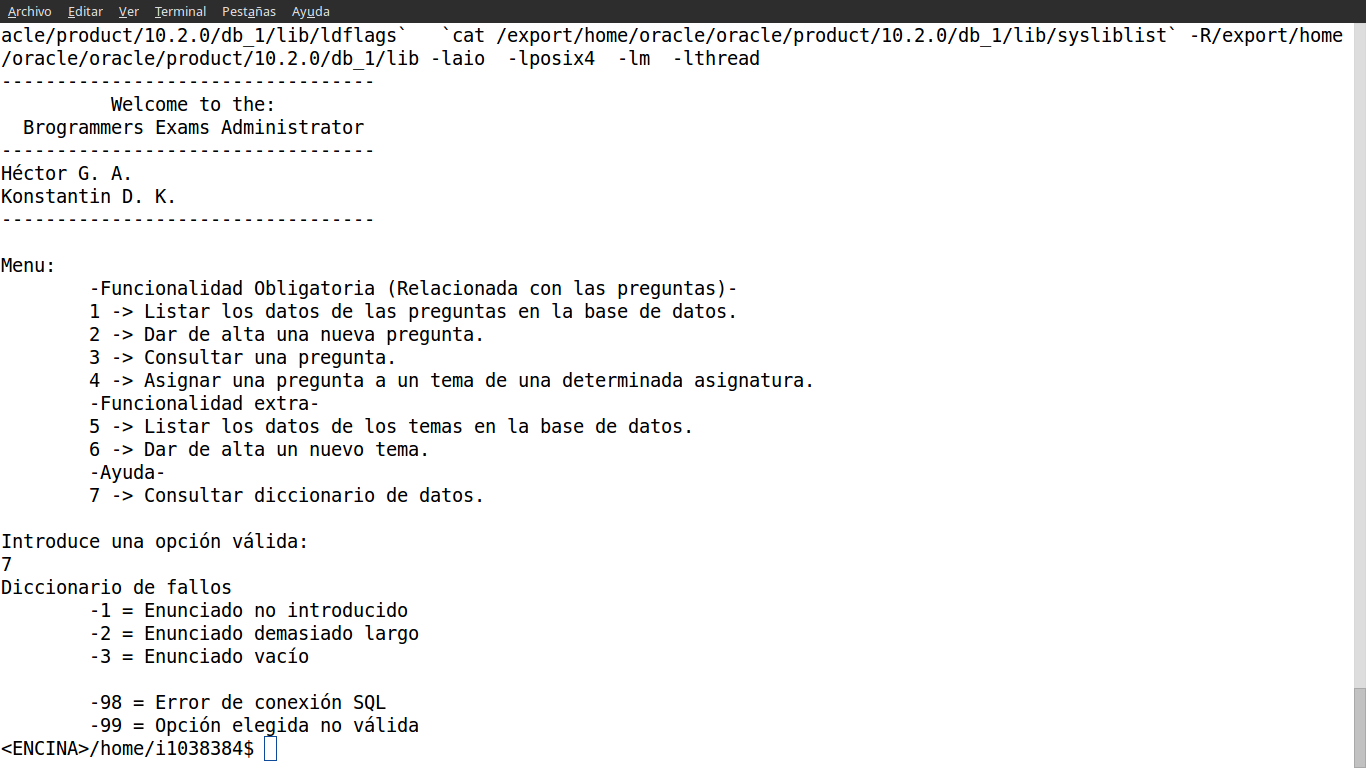
La quinta opción nos lista los datos de los temas existentes en la base de datos. Esta opción es una de las que hemos añadido nosotros ya que creemos que puede ser interesante a la hora de gestionar exámenes.



La sexta opción nos permite dar de alta un nuevo tema. En caso de sobrepasar el tamaño indicado para los campos solicitados se notifica al usuario y se cierra la ejecución del programa sin realizar la inserción. Hemos preferido esto en vez de cortar la cadena por el número de caracteres para que así no se pierda la integridad de lo que el usuario quiere introducir.



La séptima y última opción nos permite consultar el diccionario de los posibles errores que puede dar la aplicación.



# Código fuente C

#include <stdio.h>

#include <string.h>

EXEC SQL INCLUDE SQLCA;

#define OK 0

#define NO\_INPUT 1

#define TOO\_LONG 2

static int getLine (char \*prmpt, char \*buff, size\_t sz);

int main(void)

{

/\*

Brogrammers Database Administrator

\*/

//Zona de declaraciÃ³n de variables

//Variables para C

int option;

char enunciado[200];

int maximo1;

int rc;

char buff[10];

//Variables para SQL

EXEC SQL BEGIN DECLARE SECTION;

//Primero los structs de las tablas

struct profesor\_

{

VARCHAR dni[10];

VARCHAR nombre[50];

VARCHAR area[50];

}profesor;

struct asignatura\_

{

int codigo;

VARCHAR nombre[50];

}asignatura;

struct imparte\_

{

VARCHAR dni[10];

int codasignat;

}imparte;

struct tema\_

{

int cod\_tema;

int posicion\_num;

VARCHAR titulo[50];

VARCHAR descr[250];

int num\_preguntas;

}tema;

struct formada\_por\_

{

int cod\_asign;

int cod\_tema;

}formada\_por;

struct titulacion\_

{

int cod\_titulacion;

VARCHAR nombre[50];

int creditos;

}titulacion;

struct forma\_parte\_

{

int cod\_titulacion;

int cod\_asign;

}forma\_parte;

struct examen\_

{

int cod\_examen;

int convocatoria;

}examen, examen2;

struct examina\_

{

int cod\_examen;

int cod\_asignatura;

int cod\_tema;

}examina;

struct pregunta\_

{

int cod\_pregunta;

VARCHAR enunciado[200]; //En verdad son 201 lo que hay que poner, pero queda muy largo en el print

int num\_exams;

} pregunta, pregunta2;

struct evalua\_

{

int cod\_tema;

int cod\_pregunta;

}evalua;

struct compuesta\_por\_

{

int cod\_examen;

int cod\_pregunta;

int num\_correctas;

int num\_incorrectas;

int num\_nsnc;

}compuesta\_por, compuesta\_por2;

struct opcion\_

{

int cod\_pregunta;

int cod\_opcion;

int posicion\_num;

VARCHAR texto[50];

VARCHAR respuesta[2];

}opcion, opcion2;

//Otras variables para SQL

char enunciadoSQL[201];

char tituloSQL[51];

char descrSQL[251];

int MAXIMUM;

int MAXIMUM2;

int MAXIMUM3;

int CODIGOCONSULTA;

int CODIGOCONSULTA2;

int POSICION;

int PREGUNTITA;

int TEMITA;

char oracleid[]="/";

EXEC SQL END DECLARE SECTION;

//Cursores para SQL

EXEC SQL DECLARE micursor CURSOR FOR select p.Cod\_Pregunta, t.Titulo, a.Nombre from Pregunta p, Tema t, Asignatura a, Formada\_Por fp, Evalua e where p.Cod\_Pregunta = e.Cod\_Pregunta and t.Cod\_Tema = e.Cod\_Tema and t.Cod\_Tema = fp.Cod\_Tema and a.Codigo = fp.Cod\_Asign;;

EXEC SQL DECLARE jejecursor CURSOR FOR select \* from Tema;

EXEC SQL DECLARE cursorcillo CURSOR FOR select Cod\_Pregunta from Pregunta;

EXEC SQL DECLARE cursorcillo2 CURSOR FOR select Cod\_Tema from Tema;

//Cuerpo de programa

printf("----------------------------------\n");

printf(" Welcome to the: \n");

printf(" Brogrammers Exams Administrator \n");

printf("----------------------------------\n");

printf("HÃ©ctor G. A. \n");

printf("Konstantin D. K. \n");

printf("----------------------------------\n");

printf("\n");

printf("Menu:\n");

printf(" -Funcionalidad Obligatoria (Relacionada con las preguntas)-\n");

printf(" 1 -> Listar los datos de las preguntas en la base de datos.\n");

printf(" 2 -> Dar de alta una nueva pregunta.\n");

printf(" 3 -> Consultar una pregunta.\n");

printf(" 4 -> Asignar una pregunta a un tema de una determinada asignatura.\n");

printf(" -Funcionalidad extra-\n");

printf(" 5 -> Listar los datos de los temas en la base de datos.\n");

printf(" 6 -> Dar de alta un nuevo tema.\n");

printf(" -Ayuda-\n");

printf(" 7 -> Consultar diccionario de datos.\n");

printf("\n");

while(option!=1 && option!=2 && option!=3 && option!=4 && option!=5 && option!=6 && option!=7)

{

fseek(stdin,0,SEEK\_END); //Limpieza del buffer

printf("Introduce una opciÃ³n vÃ¡lida: \n");

scanf("%d", &option);

}

switch (option)

{

case 1: //FINISHED & CLEANED

printf("\n");

printf("-LISTAR LAS PREGUNTAS DE LA BASE DE DATOS-\n");

printf("\n");

EXEC SQL CONNECT :oracleid;

if (sqlca.sqlcode != 0) //ConexiÃ³n fallida

{

printf("Error, conexiÃ³n fallida\n");

printf("CÃ³digos de error de SQL: \n");

printf("%d\n", sqlca.sqlcode);

printf("%s\n", sqlca.sqlerrm.sqlerrmc);

printf("CÃ³digo de error -98\n");

printf("Saliendo del programa.\n");

return -98;

}

EXEC SQL OPEN micursor;

EXEC SQL FETCH micursor

INTO :pregunta.cod\_pregunta, :tema.titulo, :asignatura.nombre;

while (sqlca.sqlcode == 0)

{

printf("\n");

printf("Pregunta = %d \n", pregunta.cod\_pregunta);

printf("Tema = %.\*s \n", tema.titulo.len, tema.titulo.arr);

printf("Asignatura = %.\*s \n", asignatura.nombre.len, asignatura.nombre.arr);

printf("\n");

EXEC SQL FETCH micursor

INTO :pregunta.cod\_pregunta, :tema.titulo, :asignatura.nombre;

}

EXEC SQL CLOSE micursor;

EXEC SQL COMMIT WORK RELEASE;

break;

case 2: //FINISHED & CLEANED

fseek(stdin,0,SEEK\_END); //Limpieza del buffer

printf("\n");

printf("-DAR DE ALTA UNA NUEVA PREGUNTA- \n");

printf("\n");

printf("Para dar de alta una nueva pregunta se necesita el enunciado\n");

printf("Puedes introducirlo a continuaciÃ³n:\n");

rc = getLine ("Enunciado: ", enunciadoSQL, sizeof(enunciadoSQL));

if (rc == NO\_INPUT)

{

printf("No se ha introducido ningÃºn enunciado.\n");

printf("Saliendo del programa.\n");

printf("CÃ³digo del fallo: -1\n");

return -1;

}

if (rc == TOO\_LONG)

{

printf("El enunciado es demasiado largo,\n");

printf ("Solo se mantendrÃ­a: [%s]\n", enunciadoSQL);

printf("Saliendo del programa.\n");

printf("CÃ³digo del fallo -2\n");

return -2;

}

if (strlen(enunciadoSQL)<1)

{

printf("No hay enunciado\n");

printf("Saliendo del programa.\n");

printf("CÃ³digo de fallo: -3\n");

return -3;

}

printf ("OK [%s]\n", enunciadoSQL);

//Consulta para hallar el max

fseek(stdin,0,SEEK\_END); //Limpieza del buffer

EXEC SQL CONNECT :oracleid;

if (sqlca.sqlcode != 0) //ConexiÃ³n fallida

{

printf("Error, conexiÃ³n fallida\n");

printf("CÃ³digos de error de SQL: \n");

printf("%d\n", sqlca.sqlcode);

printf("%s\n", sqlca.sqlerrm.sqlerrmc);

printf("CÃ³digo de error -98\n");

printf("Saliendo del programa.\n");

return -98;

}

EXEC SQL select max(Cod\_Pregunta) into :MAXIMUM2 from pregunta;

EXEC SQL insert into pregunta values (:MAXIMUM2+1, :enunciadoSQL, 0);

printf("InserciÃ³n realizada con Ã©xito\n");

printf("ID de la pregunta insertada: %d\n", MAXIMUM2+1);

EXEC SQL COMMIT WORK RELEASE;

break;

case 3://FINISHED & CLEAN

fseek(stdin,0,SEEK\_END); //Limpieza del buffer

printf("\n");

printf("-MOSTRAR TODOS LOS DATOS DE UNA PREGUNTA A PARTIR DE SU CÃ“DIGO- \n");

printf("\n");

CODIGOCONSULTA = -1;

//Consulta para hallar el max

EXEC SQL CONNECT :oracleid;

if (sqlca.sqlcode != 0) //ConexiÃ³n fallida

{

printf("Error, conexiÃ³n fallida\n");

printf("CÃ³digos de error de SQL: \n");

printf("%d\n", sqlca.sqlcode);

printf("%s\n", sqlca.sqlerrm.sqlerrmc);

printf("CÃ³digo de error -98\n");

printf("Saliendo del programa.\n");

return -98;

}

EXEC SQL select max(Cod\_Pregunta) into :MAXIMUM from pregunta;

EXEC SQL COMMIT WORK RELEASE;

printf("A continuaciÃ³n se muestra un listado de los cÃ³digos de las preguntas disponibles.\n");

EXEC SQL CONNECT :oracleid;

if (sqlca.sqlcode != 0) //ConexiÃ³n fallida

{

printf("Error, conexiÃ³n fallida\n");

printf("CÃ³digos de error de SQL: \n");

printf("%d\n", sqlca.sqlcode);

printf("%s\n", sqlca.sqlerrm.sqlerrmc);

printf("CÃ³digo de error -98\n");

printf("Saliendo del programa.\n");

return -98;

}

EXEC SQL OPEN cursorcillo;

EXEC SQL FETCH cursorcillo

INTO :pregunta.cod\_pregunta;

while (sqlca.sqlcode == 0)

{

printf("\n");

printf("Pregunta = %d \n", pregunta.cod\_pregunta);

printf("\n");

EXEC SQL FETCH cursorcillo

INTO :pregunta.cod\_pregunta;

}

EXEC SQL CLOSE cursorcillo;

EXEC SQL COMMIT WORK RELEASE;

do

{

printf("Elige un cÃ³digo entre 1 y ");

printf("%d\n", MAXIMUM);

printf("Y que se encuentre en el listado mostrado.\n");

scanf("%d", &CODIGOCONSULTA);

}while(CODIGOCONSULTA > MAXIMUM);

//Obtener todo lo de pregunta

EXEC SQL CONNECT :oracleid;

if (sqlca.sqlcode != 0) //ConexiÃ³n fallida

{

printf("Error, conexiÃ³n fallida\n");

printf("CÃ³digos de error de SQL: \n");

printf("%d\n", sqlca.sqlcode);

printf("%s\n", sqlca.sqlerrm.sqlerrmc);

printf("CÃ³digo de error -98\n");

printf("Saliendo del programa.\n");

return -98;

}

EXEC SQL select \* into :pregunta from pregunta where cod\_pregunta = :CODIGOCONSULTA;

EXEC SQL COMMIT WORK RELEASE;

printf("\n");

printf("Pregunta seleccionada:\n");

printf("Codigo: %d \n", pregunta.cod\_pregunta);

printf("Enunciado = %.\*s \n", pregunta.enunciado.len, pregunta.enunciado.arr);

printf("NÃºmero de exÃ¡menes en los que aparece: %d \n", pregunta.num\_exams);

//Select sobre evalua

EXEC SQL DECLARE cursorevalua CURSOR FOR select \* from Evalua e where e.Cod\_Pregunta = :pregunta.cod\_pregunta;

printf("Temas que evalua la pregunta seleccionada:\n");

EXEC SQL CONNECT :oracleid;

if (sqlca.sqlcode != 0) //ConexiÃ³n fallida

{

printf("Error, conexiÃ³n fallida\n");

printf("CÃ³digos de error de SQL: \n");

printf("%d\n", sqlca.sqlcode);

printf("%s\n", sqlca.sqlerrm.sqlerrmc);

printf("CÃ³digo de error -98\n");

printf("Saliendo del programa.\n");

return -98;

}

EXEC SQL OPEN cursorevalua;

EXEC SQL FETCH cursorevalua

INTO :evalua;

while (sqlca.sqlcode == 0)

{

printf("\n");

printf("Tema = %d\n", evalua.cod\_tema);

EXEC SQL FETCH cursorevalua

INTO :evalua;

}

EXEC SQL CLOSE cursorevalua;

EXEC SQL COMMIT WORK RELEASE;

//Select sobre Compuesta\_Por

EXEC SQL DECLARE cursorcompuestapor CURSOR FOR select \* from Compuesta\_Por cp where cp.Cod\_Pregunta = :pregunta.cod\_pregunta;

printf("ExÃ¡menes que incluyen la pregunta:\n");

EXEC SQL CONNECT :oracleid;

if (sqlca.sqlcode != 0) //ConexiÃ³n fallida

{

printf("Error, conexiÃ³n fallida\n");

printf("CÃ³digos de error de SQL: \n");

printf("%d\n", sqlca.sqlcode);

printf("%s\n", sqlca.sqlerrm.sqlerrmc);

printf("CÃ³digo de error -98\n");

printf("Saliendo del programa.\n");

return -98;

}

EXEC SQL OPEN cursorcompuestapor;

EXEC SQL FETCH cursorcompuestapor

INTO :compuesta\_por;

while (sqlca.sqlcode == 0)

{

printf("\n");

printf("ExÃ¡men: %d\n", compuesta\_por.cod\_examen);

printf("NÃºmero de respuestas correctas: %d \n", compuesta\_por.num\_correctas);

printf("NÃºmero de respuestas incorrectas: %d\n", compuesta\_por.num\_incorrectas);

printf("NÃºmero de respuestas no sabe no contesta: %d\n", compuesta\_por.num\_nsnc);

EXEC SQL FETCH cursorcompuestapor

INTO :evalua;

}

EXEC SQL CLOSE cursorcompuestapor;

EXEC SQL COMMIT WORK RELEASE;

//Select sobre Opcion

EXEC SQL DECLARE cursoropcion CURSOR FOR select \* from Opcion o where o.Cod\_Pregunta = :pregunta.cod\_pregunta;

printf("Opciones que contiene la pregunta:\n");

EXEC SQL CONNECT :oracleid;

if (sqlca.sqlcode != 0) //ConexiÃ³n fallida

{

printf("Error, conexiÃ³n fallida\n");

printf("CÃ³digos de error de SQL: \n");

printf("%d\n", sqlca.sqlcode);

printf("%s\n", sqlca.sqlerrm.sqlerrmc);

printf("CÃ³digo de error -98\n");

printf("Saliendo del programa.\n");

return -98;

}

EXEC SQL OPEN cursoropcion;

EXEC SQL FETCH cursoropcion

INTO :opcion;

while (sqlca.sqlcode == 0)

{

printf("\n");

printf("Opcion: %d\n", opcion.cod\_opcion);

printf("PosiciÃ³n: %d \n", opcion.posicion\_num);

printf("Texto: %.\*s \n", opcion.texto.len, opcion.texto.arr);

printf("Respuesta: %.\*s \n", opcion.respuesta.len, opcion.respuesta.arr);

printf("\n");

EXEC SQL FETCH cursoropcion

INTO :opcion;

}

EXEC SQL CLOSE cursoropcion;

EXEC SQL COMMIT WORK RELEASE;

break;

case 4://FINISHED & CLEAN

printf("\n");

printf("-ASIGNAR UNA PREGUNTA A UN TEMA DE UNA DETERMINADA ASIGNATURA-\n");

printf("\n");

//Consulta para hallar el max

EXEC SQL CONNECT :oracleid;

if (sqlca.sqlcode != 0) //ConexiÃ³n fallida

{

printf("Error, conexiÃ³n fallida\n");

printf("CÃ³digos de error de SQL: \n");

printf("%d\n", sqlca.sqlcode);

printf("%s\n", sqlca.sqlerrm.sqlerrmc);

printf("CÃ³digo de error -98\n");

printf("Saliendo del programa.\n");

return -98;

}

EXEC SQL select max(Cod\_Pregunta) into :MAXIMUM from pregunta;

EXEC SQL COMMIT WORK RELEASE;

CODIGOCONSULTA2 = -1;

//Consulta para hallar el max

EXEC SQL CONNECT :oracleid;

if (sqlca.sqlcode != 0) //ConexiÃ³n fallida

{

printf("Error, conexiÃ³n fallida\n");

printf("CÃ³digos de error de SQL: \n");

printf("%d\n", sqlca.sqlcode);

printf("%s\n", sqlca.sqlerrm.sqlerrmc);

printf("CÃ³digo de error -98\n");

printf("Saliendo del programa.\n");

return -98;

}

EXEC SQL select max(Cod\_Tema) into :MAXIMUM2 from tema;

EXEC SQL COMMIT WORK RELEASE;

printf("A continuación se muestra un listado de los códigos de las preguntas disponibles.\n");

EXEC SQL CONNECT :oracleid;

if (sqlca.sqlcode != 0) //Conexión fallida

{

printf("Error, conexiÃ³n fallida\n");

printf("CÃ³digos de error de SQL: \n");

printf("%d\n", sqlca.sqlcode);

printf("%s\n", sqlca.sqlerrm.sqlerrmc);

printf("Código de error -98\n");

printf("Saliendo del programa.\n");

return -98;

}

EXEC SQL OPEN cursorcillo;

EXEC SQL FETCH cursorcillo

INTO :pregunta.cod\_pregunta;

while (sqlca.sqlcode == 0)

{

printf("\n");

printf("Pregunta = %d \n", pregunta.cod\_pregunta);

printf("\n");

EXEC SQL FETCH cursorcillo

INTO :pregunta.cod\_pregunta;

}

EXEC SQL CLOSE cursorcillo;

EXEC SQL COMMIT WORK RELEASE;

do

{

printf("Elige un código de pregunta comprendido entre 1 y ");

printf("%d\n", MAXIMUM);

printf("Y que se muestre en el listado superior.\n");

fseek(stdin,0,SEEK\_END); //Limpieza del buffer

scanf("%d", &PREGUNTITA);

}while (PREGUNTITA > MAXIMUM);

printf("A continuaciÃ³n se muestra la lista de temas\n");

EXEC SQL CONNECT :oracleid;

if (sqlca.sqlcode != 0) //ConexiÃ³n fallida

{

printf("Error, conexiÃ³n fallida\n");

printf("CÃ³digos de error de SQL: \n");

printf("%d\n", sqlca.sqlcode);

printf("%s\n", sqlca.sqlerrm.sqlerrmc);

printf("CÃ³digo de error -98\n");

printf("Saliendo del programa.\n");

return -98;

}

EXEC SQL OPEN cursorcillo2;

EXEC SQL FETCH cursorcillo2

INTO :tema.cod\_tema;

while (sqlca.sqlcode == 0)

{

printf("\n");

printf("Tema = %d \n", tema.cod\_tema);

printf("\n");

EXEC SQL FETCH cursorcillo2

INTO :tema.cod\_tema;

}

EXEC SQL CLOSE cursorcillo2;

EXEC SQL COMMIT WORK RELEASE;

do

{

printf("Elige un cÃ³digo de tema comprendido entre 1 y ");

printf("%d\n", MAXIMUM2);

printf("Y que se muestre en el listado superior.\n");

fseek(stdin,0,SEEK\_END); //Limpieza del buffer

scanf("%d", &TEMITA);

}while (TEMITA > MAXIMUM2);

printf("Has seleccionado la pregunta %d y el tema %d\n", PREGUNTITA, TEMITA);

fseek(stdin,0,SEEK\_END); //Limpieza del buffer

EXEC SQL CONNECT :oracleid;

if (sqlca.sqlcode != 0) //ConexiÃ³n fallida

{

printf("Error, conexiÃ³n fallida\n");

printf("CÃ³digos de error de SQL: \n");

printf("%d\n", sqlca.sqlcode);

printf("%s\n", sqlca.sqlerrm.sqlerrmc);

printf("CÃ³digo de error -98\n");

printf("Saliendo del programa.\n");

return -98;

}

EXEC SQL insert into evalua values (:TEMITA, :PREGUNTITA);

printf("InserciÃ³n realizada con Ã©xito\n");

EXEC SQL COMMIT WORK RELEASE;

break;

case 5://FINISHED & CLEAN

printf("\n");

printf("-LISTADO DE LOS TEMAS EN LA BASE DE DATOS- \n");

printf("\n");

EXEC SQL CONNECT :oracleid;

if (sqlca.sqlcode != 0) //ConexiÃ³n fallida

{

printf("Error, conexiÃ³n fallida\n");

printf("CÃ³digos de error de SQL: \n");

printf("%d\n", sqlca.sqlcode);

printf("%s\n", sqlca.sqlerrm.sqlerrmc);

printf("CÃ³digo de error -98\n");

printf("Saliendo del programa.\n");

return -98;

}

EXEC SQL OPEN jejecursor;

EXEC SQL FETCH jejecursor

INTO :tema;

while (sqlca.sqlcode == 0)

{

printf("\n");

printf("Tema:\n");

printf("CÃ³digo = %d \n", tema.cod\_tema);

printf("PosiciÃ³n = %d \n", tema.posicion\_num);

printf("TÃ­tulo = %.\*s \n", tema.titulo.len, tema.titulo.arr);

printf("DescripciÃ³n = %.\*s \n", tema.descr.len, tema.descr.arr);

printf("NÃºmero de preguntas = %d \n", tema.num\_preguntas);

printf("\n");

EXEC SQL FETCH jejecursor

INTO :tema;

}

EXEC SQL CLOSE jejecursor;

EXEC SQL COMMIT WORK RELEASE;

break;

case 6://FINISHED & CLEANED

fseek(stdin,0,SEEK\_END); //Limpieza del buffer

printf("\n");

printf("-DAR DE ALTA UN NUEVO TEMA- \n");

printf("\n");

printf("Para dar de alta un nuevo tema se necesita el tÃ­tulo, la descipciÃ³n y su posiciÃ³n\n");

printf("Se solicitarÃ¡n los datos a continuaciÃ³n:\n");

rc = getLine ("TÃ­tulo (max 50): ", tituloSQL, sizeof(tituloSQL));

if (rc == NO\_INPUT)

{

printf("No se ha introducido ningÃºn tÃ­tulo.\n");

printf("Saliendo del programa.\n");

printf("CÃ³digo del fallo: -1\n");

return -1;

}

if (rc == TOO\_LONG)

{

printf("El tÃ­tulo es demasiado largo,\n");

printf ("Solo se mantendrÃ­a: [%s]\n", tituloSQL);

printf("Saliendo del programa.\n");

printf("CÃ³digo del fallo -2\n");

return -2;

}

if (strlen(tituloSQL)<1)

{

printf("No hay tÃ­tulo\n");

printf("Saliendo del programa.\n");

printf("CÃ³digo de fallo: -3\n");

return -3;

}

rc = getLine ("DescripciÃ³n (max 250): ", descrSQL, sizeof(descrSQL));

if (rc == NO\_INPUT)

{

printf("No se ha introducido ningÃºn tÃ­tulo.\n");

printf("Saliendo del programa.\n");

printf("CÃ³digo del fallo: -1\n");

return -1;

}

if (rc == TOO\_LONG)

{

printf("El tÃ­tulo es demasiado largo,\n");

printf ("Solo se mantendrÃ­a: [%s]\n", descrSQL);

printf("Saliendo del programa.\n");

printf("CÃ³digo del fallo -2\n");

return -2;

}

if (strlen(descrSQL)<1)

{

printf("No hay tÃ­tulo\n");

printf("Saliendo del programa.\n");

printf("CÃ³digo de fallo: -3\n");

return -3;

}

printf("PosiciÃ³n: ");

scanf("%d", &POSICION);

printf("TÃ­tulo introducido: %s\n", tituloSQL);

printf("DescripciÃ³n introducida:%s\n", descrSQL);

printf("PosiciÃ³n introducida: %d\n", POSICION);

//Consulta para hallar el max e insertar

fseek(stdin,0,SEEK\_END); //Limpieza del buffer

EXEC SQL CONNECT :oracleid;

if (sqlca.sqlcode != 0) //ConexiÃ³n fallida

{

printf("Error, conexiÃ³n fallida\n");

printf("CÃ³digos de error de SQL: \n");

printf("%d\n", sqlca.sqlcode);

printf("%s\n", sqlca.sqlerrm.sqlerrmc);

printf("CÃ³digo de error -98\n");

printf("Saliendo del programa.\n");

return -98;

}

EXEC SQL select max(Cod\_Tema) into :MAXIMUM3 from tema;

EXEC SQL insert into tema values (:MAXIMUM3+1, :POSICION, :enunciadoSQL, :descrSQL, 0);

printf("InserciÃ³n realizada con Ã©xito\n");

printf("ID del tema insertado: %d\n", MAXIMUM3+1);

EXEC SQL COMMIT WORK RELEASE;

break;

case 7:

printf("Diccionario de fallos\n");

printf(" -1 = Enunciado no introducido\n");

printf(" -2 = Enunciado demasiado largo\n");

printf(" -3 = Enunciado vacÃ­o\n");

printf("\n");

printf(" -98 = Error de conexiÃ³n SQL\n");

printf(" -99 = OpciÃ³n elegida no vÃ¡lida\n");

break;

default:

printf("CÃ³digo de fallo: -99\n");

printf("La opciÃ³n introducida no es vÃ¡lida\n");

return(-99);

}

return 0;

}

static int getLine (char \*prmpt, char \*buff, size\_t sz) //FUNCION UTILIZADA PARA ESCANEAR CARACTÃ‰RES

{

int ch, extra;

// Get line with buffer overrun protection.

if (prmpt != NULL) {

printf ("%s", prmpt);

fflush (stdout);

}

if (fgets (buff, sz, stdin) == NULL)

return NO\_INPUT;

// If it was too long, there'll be no newline. In that case, we flush

// to end of line so that excess doesn't affect the next call.

if (buff[strlen(buff)-1] != '\n') {

extra = 0;

while (((ch = getchar()) != '\n') && (ch != EOF))

extra = 1;

return (extra == 1) ? TOO\_LONG : OK;

}

// Otherwise remove newline and give string back to caller.

buff[strlen(buff)-1] = '\0';

return OK;

}

# Scripts SQL

Scripts para la creación de las tablas de las bases de datos: creabd.sql.

/\*Lo primero que vamos a hacer es crear un usuario nuevo para la base de datos y así­ no trabajar directamente desde SYSTEM (esto es opcional)\*/

create user brogrammers identified by brogrammerspasswd default tablespace users temporary tablespace temp;

grant connect, resource to brogrammers;

/\*Nota:

Al crear la vista nos daba un error de privilegios, los solucionamos con la siguiente sentencia:\*/

grant all privileges to brogrammers identified by brogrammerspasswd;

/\*Y podremos ver si los privilegios han sido aplicados correctamente con la siguiente sentencia\*/

select \* from system\_privilege\_map;

/\*Si utilizamos una base de datos ya existente, podemos consultar nuestras tablas para comprobar que no entran en conflicto con ninguna de las que ya podamos tener\*/

select owner, table\_name from all\_tables where owner='brogrammers';

/\*O bien podemos asegurarnos de que no exista ninguna tabla con el nombre\*/

drop table Imparte;

drop table Profesor;

drop table Formada\_por;

drop table Forma\_Parte;

drop table Examina;

drop table Compuesta\_por;

drop table Evalua;

drop table Asignatura;

drop table Tema;

drop table Titulacion;

drop table Examen;

drop table Pregunta;

drop table Opcion;

/\*Nota

Hemos optado por utilizar una mezcla entre la solución propuesta en Studium y la nuestra propia, para así­ corregir errores, y poder aprovechar el script de relleno de la base de datos.\*/

/\*Código de creación de la Base de Datos\*/

/\* set autocommit on; Para guardar cambios automáticamente\*/

create table Profesor

(

DNI VARCHAR(10) NOT NULL PRIMARY KEY,

Nombre VARCHAR (50) NOT NULL,

Area VARCHAR (50) NOT NULL

);

create table Asignatura

(

CODIGO INTEGER NOT NULL PRIMARY KEY,

Nombre VARCHAR (50) NOT NULL

);

create table Imparte

(

DNI VARCHAR (10) NOT NULL REFERENCES PROFESOR,

CODASIGNAT INTEGER NOT NULL REFERENCES ASIGNATURA,

PRIMARY KEY (DNI, CODASIGNAT)

);

create table Tema

(

Cod\_Tema integer not null primary key,

Posicion\_Num integer not null,

Titulo varchar(50) not null,

Descr varchar(250) not null,

Num\_Preguntas INTEGER NOT NULL /\*Este es atributo derivado\*/

);

create table Formada\_por

(

Cod\_Asign integer not null references ASIGNATURA,

Cod\_Tema integer not null references TEMA,

primary key (Cod\_Asign, Cod\_Tema)

);

create table Titulacion

(

Cod\_Titulacion integer not null primary key,

Nombre varchar(50) not null,

Creditos integer not null

);

create table Forma\_Parte

(

Cod\_Titulacion integer not null references TITULACION,

Cod\_Asign integer not null references ASIGNATURA,

primary key (Cod\_Asign, Cod\_Titulacion)

);

create table Examen

(

Cod\_Examen integer not null primary key,

Convocatoria integer not null

);

create table Examina

(

Cod\_Examen integer not null references Examen,

Cod\_Asignatura integer not null references Asignatura,

Cod\_Tema integer not null references Tema,

primary key (Cod\_Examen, Cod\_Asignatura, Cod\_Tema)

);

create table Pregunta

(

Cod\_Pregunta INTEGER NOT NULL PRIMARY KEY,

Enunciado VARCHAR (200) NOT NULL,

Num\_Exams INTEGER NOT NULL /\*Este es atributo derivado\*/

);

create table Evalua

(

Cod\_Tema INTEGER NOT NULL REFERENCES Tema,

Cod\_Pregunta INTEGER NOT NULL REFERENCES Pregunta,

PRIMARY KEY (Cod\_Pregunta, Cod\_Tema)

);

create table Compuesta\_Por

(

Cod\_Examen integer not null references EXAMEN,

Cod\_Pregunta integer not null references PREGUNTA,

Num\_Correctas integer not null,

Num\_Incorrectas integer not null,

Num\_NSNC integer not null,

primary key (Cod\_Examen, Cod\_Pregunta)

);

create table Opcion

(

Cod\_Pregunta integer not null references PREGUNTA,

Cod\_Opcion integer not null primary key,

Posicion\_Num integer not null,

Texto varchar(50) not null,

Respuesta varchar(2)

);

/\*Triggers\*/

/\*Triger para Exámenes\*/

CREATE TRIGGER ADD\_EXAM

AFTER INSERT ON Compuesta\_por

FOR EACH ROW

WHEN (new.Cod\_Pregunta is not null)

BEGIN

UPDATE Pregunta SET Num\_Exams=Num\_Exams+1 WHERE Cod\_Pregunta=:new.Cod\_Pregunta;

END;

/

CREATE TRIGGER RM\_EXAM

AFTER DELETE ON Compuesta\_por

FOR EACH ROW

WHEN (old.Cod\_Pregunta is not null)

BEGIN

UPDATE Pregunta SET Num\_Exams=Num\_Exams-1 WHERE Cod\_Pregunta=:old.Cod\_Pregunta;

END;

/

CREATE TRIGGER UPD\_EXAM

AFTER UPDATE OF Cod\_Pregunta ON Compuesta\_por

FOR EACH ROW

WHEN(new.Cod\_Pregunta is not null or old.Cod\_Pregunta is not null)

BEGIN

UPDATE Pregunta SET Num\_Exams=Num\_Exams+1 WHERE Cod\_Pregunta=:new.Cod\_Pregunta;

UPDATE Pregunta SET Num\_Exams=Num\_Exams-1 WHERE Cod\_Pregunta=:old.Cod\_Pregunta;

END;

/

/\*Triger para Tema\*/

CREATE TRIGGER ADD\_PREGUNTA

AFTER INSERT ON Evalua

FOR EACH ROW

WHEN (new.Cod\_Tema is not null)

BEGIN

UPDATE Tema SET Num\_Preguntas=Num\_Preguntas+1 WHERE Cod\_Tema=:new.Cod\_Tema;

END;

/

CREATE TRIGGER RM\_PREGUNTA

AFTER DELETE ON Evalua

FOR EACH ROW

WHEN (old.Cod\_Tema is not null)

BEGIN

UPDATE Tema SET Num\_Preguntas=Num\_Preguntas-1 WHERE Cod\_Tema=:old.Cod\_Tema;

END;

/

CREATE TRIGGER UPD\_PREGUNTA

AFTER UPDATE OF Cod\_Pregunta ON Evalua

FOR EACH ROW

WHEN(new.Cod\_Tema is not null or old.Cod\_Tema is not null)

BEGIN

UPDATE Tema SET Num\_Preguntas=Num\_Preguntas+1 WHERE Cod\_Tema=:new.Cod\_Tema;

UPDATE Tema SET Num\_Preguntas=Num\_Preguntas-1 WHERE Cod\_Tema=:old.Cod\_Tema;

END;

/

/\*Declaración de privilegios\*/

grant all on Profesor to public;

grant all on Imparte to public;

grant all on Tema to public;

grant all on Compuesta\_por to public;

grant all on Titulacion to public;

grant all on Forma\_Parte to public;

grant all on Examen to public;

grant all on Pregunta to public;

grant all on Evalua to public;

grant all on Compuesta\_por to public;

grant all on Opcion to public;

commit;

Script para rellenar la base de datos: rellenabd.sql.

/\*Nos aseguramos de que las tablas están vací­as\*/

delete from Profesor;

delete from Titulacion;

delete from Asignatura;

delete from Forma\_Parte;

delete from Imparte;

delete from Tema;

delete from Pregunta;

delete from Opcion;

delete from Formada\_por;

delete from Evalua;

delete from Examen;

delete from Compuesta\_Por;

delete from Examina;

/\*DNI, Nombre, Area\*/

insert into Profesor values ('73048485', 'Pepe SÃ¡nchez GÃ³mez', 'InformÃ¡tica');

insert into Profesor values ('11396243', 'Ana GÃ³mez PÃ©rez', 'FÃ­sica');

insert into Profesor values ('11496939', 'Sandra SÃ¡nchez MartÃ­n', 'C.Sociales');

insert into Profesor values ('11123361', 'JosÃ© Rubin GÃ³mez', 'MatemÃ¡ticas');

insert into Profesor values ('73348485', 'Juan MartÃ­n MartÃ­n', 'InformÃ¡tica');

/\*Codigo, Nombre\*/

insert into Asignatura values (001, 'Bases de Datos');

insert into Asignatura values (002, 'PsicologÃ­a Infantil');

insert into Asignatura values (003, 'ProgramaciÃ³n');

insert into Asignatura values (004, 'Media, Mediana y Moda');

insert into Asignatura values (005, 'MacroeconomÃ­a');

insert into Asignatura values (006, 'FÃ­sica CuÃ¡ntica');

insert into Asignatura values (007, 'MicroeconomÃ­a');

insert into Asignatura values (008, 'Computadores');

insert into Asignatura values (009, 'ElectrÃ³nica');

insert into Asignatura values (010, 'PsicologÃ­a EconÃ³mica');

/\*DNI, CODASIGNAT\*/

insert into Imparte values ('73048485', 001);

insert into Imparte values ('73048485', 008);

insert into Imparte values ('11396243', 006);

insert into Imparte values ('11396243', 009);

insert into Imparte values ('11496939', 002);

insert into Imparte values ('11496939', 010);

insert into Imparte values ('11123361', 001);

insert into Imparte values ('73348485', 003);

insert into Imparte values ('73348485', 009);

/\*Cod\_titulacion, Nombre, Creditos\*/

insert into Titulacion values (001,'IngenierÃ­a InformÃ¡tica', 200);

insert into Titulacion values (002,'IngenierÃ­a FÃ­sica', 200);

insert into Titulacion values (003,'EstadÃ­stica', 200);

insert into Titulacion values (004,'PsicologÃ­a', 200);

insert into Titulacion values (005,'EconomÃ­a', 200);

/\*Cod\_Titulacion, Cod\_Asign\*/

insert into Forma\_Parte values (001, 001);

insert into Forma\_Parte values (001, 003);

insert into Forma\_Parte values (001, 008);

insert into Forma\_Parte values (001, 009);

insert into Forma\_Parte values (002, 006);

insert into Forma\_Parte values (002, 009);

insert into Forma\_Parte values (003, 004);

insert into Forma\_Parte values (004, 002);

insert into Forma\_Parte values (004, 010);

insert into Forma\_Parte values (005, 007);

insert into Forma\_Parte values (005, 005);

insert into Forma\_Parte values (005, 010);

/\*Cod\_Tema, Posicion\_Num, Titulo, Descr, Num\_Preguntas\*/

insert into Tema values (001, 01, 'El Modelo Relacional', 'Descr', 0);

insert into Tema values (002, 10, 'IntroducciÃ³n a SQL', 'Descr', 0);

insert into Tema values (003, 01, 'Media', 'Descr', 0);

insert into Tema values (004, 02, 'Mediana', 'Descr', 0);

insert into Tema values (005, 03, 'Moda', 'Descr', 0);

insert into Tema values (006, 01, 'IntroducciÃ³n a C', 'Descr', 0);

insert into Tema values (007, 03, 'Estructuras de Control', 'Descr', 0);

insert into Tema values (008, 01, 'MacroeconomÃ­a Americana', 'Descr', 0);

insert into Tema values (009, 02, 'MacroeconomÃ­a Europea', 'Descr', 0);

insert into Tema values (010, 03, 'MacroeconomÃ­a AsiÃ¡tica', 'Descr', 0);

insert into Tema values (011, 01, 'InstrucciÃ³n AND', 'Descr', 0);

insert into Tema values (012, 02, 'InstrucciÃ³n OR', 'Descr', 0);

insert into Tema values (013, 03, 'InstrucciÃ³n NOT', 'Descr', 0);

insert into Tema values (014, 03, 'Resistencias', 'Descr', 0);

insert into Tema values (015, 01, 'Transistores', 'Descr', 0);

insert into Tema values (016, 10, 'LEDS', 'Descr', 0);

insert into Tema values (017, 01, 'Protones', 'Descr', 0);

insert into Tema values (018, 02, 'Neutrones', 'Descr', 0);

insert into Tema values (019, 01, 'La EconomÃ­a y la Sociedad', 'Descr', 0);

insert into Tema values (020, 02, 'PsicologÃ­a en la EconomÃ­a', 'Descr', 0);

insert into Tema values (021, 01, 'MicroeconomÃ­a Americana', 'Descr', 0);

insert into Tema values (022, 02, 'MicroeconomÃ­a Europea', 'Descr', 0);

insert into Tema values (023, 03, 'MicroeconomÃ­a AsiÃ¡tica', 'Descr', 0);

insert into Tema values (024, 02, 'PsicologÃ­a en la Adolescencia', 'Descr', 0);

insert into Tema values (025, 03, 'PsicologÃ­a del reciÃ©n nacido', 'Descr', 0);

/\*Cod\_Asign, Cod\_Tema\*/

insert into Formada\_por values (001, 001);

insert into Formada\_por values (001, 002);

insert into Formada\_por values (002, 024);

insert into Formada\_por values (002, 025);

insert into Formada\_por values (003, 006);

insert into Formada\_por values (003, 007);

insert into Formada\_por values (004, 003);

insert into Formada\_por values (004, 004);

insert into Formada\_por values (004, 005);

insert into Formada\_por values (005, 008);

insert into Formada\_por values (005, 009);

insert into Formada\_por values (005, 010);

insert into Formada\_por values (006, 017);

insert into Formada\_por values (006, 018);

insert into Formada\_por values (007, 021);

insert into Formada\_por values (007, 022);

insert into Formada\_por values (007, 023);

insert into Formada\_por values (008, 011);

insert into Formada\_por values (008, 012);

insert into Formada\_por values (008, 013);

insert into Formada\_por values (009, 014);

insert into Formada\_por values (009, 015);

insert into Formada\_por values (009, 016);

/\*Cod\_Examen, Convocatoria\*/

/\*En un principio todos son de primera convocatoria\*/

/\*Tendremos 10 exámenes Iniciales\*/

insert into Examen values (001, 1);

insert into Examen values (002, 1);

insert into Examen values (003, 1);

insert into Examen values (004, 1);

insert into Examen values (005, 1);

insert into Examen values (006, 1);

insert into Examen values (007, 1);

insert into Examen values (008, 1);

insert into Examen values (009, 1);

insert into Examen values (010, 1);

/\*Cod\_Examen, Cod\_Asign. Cod\_Tema\*/

insert into Examina values (001, 006, 017);

insert into Examina values (001, 006, 018);

insert into Examina values (002, 001, 001);

insert into Examina values (002, 001, 002);

insert into Examina values (003, 007, 021);

insert into Examina values (003, 007, 022);

insert into Examina values (003, 007, 023);

insert into Examina values (004, 003, 006);

insert into Examina values (004, 003, 007);

insert into Examina values (005, 002, 024);

insert into Examina values (005, 002, 025);

insert into Examina values (006, 008, 011);

insert into Examina values (006, 008, 012);

insert into Examina values (006, 008, 013);

insert into Examina values (007, 004, 003);

insert into Examina values (007, 004, 004);

insert into Examina values (007, 004, 005);

insert into Examina values (008, 005, 008);

insert into Examina values (008, 005, 009);

insert into Examina values (008, 005, 010);

insert into Examina values (009, 010, 019);

insert into Examina values (009, 010, 020);

insert into Examina values (010, 009, 014);

insert into Examina values (010, 009, 015);

insert into Examina values (010, 009, 016);

/\*Cod\_Pregunta, Enunciado, Num\_Exams\*/

insert into Pregunta values (001, 'Preg 1', 0);

insert into Pregunta values (002, 'Preg 2', 0);

insert into Pregunta values (003, 'Preg 3', 0);

insert into Pregunta values (004, 'Preg 4', 0);

insert into Pregunta values (005, 'Preg 5', 0);

insert into Pregunta values (006, 'Preg 6', 0);

insert into Pregunta values (007, 'Preg 7', 0);

insert into Pregunta values (008, 'Preg 8', 0);

insert into Pregunta values (009, 'Preg 9', 0);

insert into Pregunta values (010, 'Preg 10', 0);

insert into Pregunta values (011, 'Preg 11', 0);

insert into Pregunta values (012, 'Preg 12', 0);

insert into Pregunta values (013, 'Preg 13', 0);

insert into Pregunta values (014, 'Preg 14', 0);

insert into Pregunta values (015, 'Preg 15', 0);

insert into Pregunta values (016, 'Preg 16', 0);

insert into Pregunta values (017, 'Preg 17', 0);

insert into Pregunta values (018, 'Preg 18', 0);

insert into Pregunta values (019, 'Preg 19', 0);

insert into Pregunta values (020, 'Preg 20', 0);

insert into Pregunta values (021, 'Preg 21', 0);

insert into Pregunta values (022, 'Preg 22', 0);

insert into Pregunta values (023, 'Preg 23', 0);

insert into Pregunta values (024, 'Preg 24', 0);

insert into Pregunta values (025, 'Preg 25', 0);

insert into Pregunta values (026, 'Preg 26', 0);

insert into Pregunta values (027, 'Preg 27', 0);

insert into Pregunta values (028, 'Preg 28', 0);

insert into Pregunta values (029, 'Preg 29', 0);

insert into Pregunta values (030, 'Preg 30', 0);

insert into Pregunta values (031, 'Preg 31', 0);

insert into Pregunta values (032, 'Preg 32', 0);

insert into Pregunta values (033, 'Preg 33', 0);

insert into Pregunta values (034, 'Preg 34', 0);

insert into Pregunta values (035, 'Preg 35', 0);

insert into Pregunta values (036, 'Preg 36', 0);

insert into Pregunta values (037, 'Preg 37', 0);

insert into Pregunta values (038, 'Preg 38', 0);

insert into Pregunta values (039, 'Preg 39', 0);

insert into Pregunta values (040, 'Preg 40', 0);

insert into Pregunta values (041, 'Preg 41', 0);

insert into Pregunta values (042, 'Preg 42', 0);

insert into Pregunta values (043, 'Preg 43', 0);

insert into Pregunta values (044, 'Preg 44', 0);

insert into Pregunta values (045, 'Preg 45', 0);

insert into Pregunta values (046, 'Preg 46', 0);

insert into Pregunta values (047, 'Preg 47', 0);

insert into Pregunta values (048, 'Preg 48', 0);

insert into Pregunta values (049, 'Preg 49', 0);

insert into Pregunta values (050, 'Preg 50', 0);

/\*Cod\_Examen, Cod\_Pregunta, Num\_Correctas, Num\_Incorrectas, Num\_NSNC\*/

/\*En principio cada examen cuenta con 3 preguntas\*/

/\*Aún nos quedarán preguntas para añadir en exámenes\*/

insert into Compuesta\_Por values (001, 044, 3, 2, 0);

insert into Compuesta\_Por values (001, 019, 2, 6, 1);

insert into Compuesta\_Por values (001, 021, 2, 6, 1);

insert into Compuesta\_Por values (002, 003, 4, 7, 8);

insert into Compuesta\_Por values (002, 005, 6, 7, 5);

insert into Compuesta\_Por values (002, 004, 7, 4, 3);

insert into Compuesta\_Por values (003, 017, 4, 6, 2);

insert into Compuesta\_Por values (003, 028, 2, 3, 5);

insert into Compuesta\_Por values (003, 033, 5, 2, 0);

insert into Compuesta\_Por values (004, 041, 4, 3, 4);

insert into Compuesta\_Por values (004, 047, 3, 2, 6);

insert into Compuesta\_Por values (004, 042, 6, 7, 6);

insert into Compuesta\_Por values (005, 025, 6, 4, 3);

insert into Compuesta\_Por values (005, 027, 5, 6, 0);

insert into Compuesta\_Por values (005, 026, 3, 7, 0);

insert into Compuesta\_Por values (006, 014, 7, 6, 0);

insert into Compuesta\_Por values (006, 009, 2, 5, 5);

insert into Compuesta\_Por values (006, 010, 2, 3, 5);

insert into Compuesta\_Por values (007, 001, 4, 3, 3);

insert into Compuesta\_Por values (007, 006, 5, 2, 6);

insert into Compuesta\_Por values (007, 038, 6, 5, 5);

insert into Compuesta\_Por values (008, 050, 6, 3, 4);

insert into Compuesta\_Por values (008, 008, 4, 7, 0);

insert into Compuesta\_Por values (008, 015, 3, 5, 0);

insert into Compuesta\_Por values (009, 030, 8, 7, 7);

insert into Compuesta\_Por values (009, 031, 5, 3, 4);

insert into Compuesta\_Por values (009, 018, 3, 2, 3);

insert into Compuesta\_Por values (010, 034, 6, 6, 7);

insert into Compuesta\_Por values (010, 011, 3, 4, 4);

insert into Compuesta\_Por values (010, 022, 3, 2, 3);

/\*Cod\_Tema, Cod\_Pregunta\*/

insert into Evalua values (001, 002);

insert into Evalua values (001, 003);

insert into Evalua values (002, 004);

insert into Evalua values (002, 005);

insert into Evalua values (003, 001);

insert into Evalua values (003, 039);

insert into Evalua values (004, 040);

insert into Evalua values (004, 006);

insert into Evalua values (005, 007);

insert into Evalua values (005, 038);

insert into Evalua values (006, 041);

insert into Evalua values (006, 042);

insert into Evalua values (007, 049);

insert into Evalua values (007, 047);

insert into Evalua values (008, 050);

insert into Evalua values (008, 048);

insert into Evalua values (009, 008);

insert into Evalua values (009, 046);

insert into Evalua values (010, 045);

insert into Evalua values (010, 015);

insert into Evalua values (011, 014);

insert into Evalua values (011, 013);

insert into Evalua values (012, 037);

insert into Evalua values (012, 009);

insert into Evalua values (013, 010);

insert into Evalua values (013, 036);

insert into Evalua values (014, 034);

insert into Evalua values (014, 023);

insert into Evalua values (015, 035);

insert into Evalua values (015, 011);

insert into Evalua values (016, 012);

insert into Evalua values (016, 022);

insert into Evalua values (017, 044);

insert into Evalua values (017, 021);

insert into Evalua values (018, 020);

insert into Evalua values (018, 019);

insert into Evalua values (019, 030);

insert into Evalua values (019, 029);

insert into Evalua values (020, 031);

insert into Evalua values (020, 018);

insert into Evalua values (021, 016);

insert into Evalua values (021, 017);

insert into Evalua values (022, 032);

insert into Evalua values (022, 028);

insert into Evalua values (023, 033);

insert into Evalua values (023, 043);

insert into Evalua values (024, 025);

insert into Evalua values (024, 024);

insert into Evalua values (025, 026);

insert into Evalua values (025, 027);

/\*Cod\_Pregunta, Cod\_Opcion, Posicion\_Num, Texto, Respuesta\*/

insert into Opcion values (001, 001, 1, 'texto', 'Si');

insert into Opcion values (001, 002, 2, 'texto', 'No');

insert into Opcion values (002, 003, 1, 'texto', 'Si');

insert into Opcion values (002, 004, 2, 'texto', 'No');

insert into Opcion values (003, 005, 1, 'texto', 'Si');

insert into Opcion values (003, 006, 2, 'texto', 'No');

insert into Opcion values (004, 007, 1, 'texto', 'Si');

insert into Opcion values (004, 008, 2, 'texto', 'No');

insert into Opcion values (005, 009, 1, 'texto', 'Si');

insert into Opcion values (005, 010, 2, 'texto', 'No');

insert into Opcion values (006, 011, 1, 'texto', 'Si');

insert into Opcion values (006, 012, 2, 'texto', 'No');

insert into Opcion values (007, 013, 1, 'texto', 'Si');

insert into Opcion values (007, 014, 2, 'texto', 'No');

insert into Opcion values (008, 015, 1, 'texto', 'Si');

insert into Opcion values (008, 016, 2, 'texto', 'No');

insert into Opcion values (009, 017, 1, 'texto', 'Si');

insert into Opcion values (009, 018, 2, 'texto', 'No');

insert into Opcion values (010, 019, 1, 'texto', 'Si');

insert into Opcion values (010, 020, 2, 'texto', 'No');

insert into Opcion values (011, 021, 1, 'texto', 'Si');

insert into Opcion values (011, 022, 2, 'texto', 'No');

insert into Opcion values (012, 023, 1, 'texto', 'Si');

insert into Opcion values (012, 024, 2, 'texto', 'No');

insert into Opcion values (013, 025, 1, 'texto', 'Si');

insert into Opcion values (013, 026, 2, 'texto', 'No');

insert into Opcion values (014, 027, 1, 'texto', 'Si');

insert into Opcion values (014, 028, 2, 'texto', 'No');

insert into Opcion values (015, 029, 1, 'texto', 'Si');

insert into Opcion values (015, 030, 2, 'texto', 'No');

insert into Opcion values (016, 031, 1, 'texto', 'Si');

insert into Opcion values (016, 032, 2, 'texto', 'No');

insert into Opcion values (017, 033, 1, 'texto', 'Si');

insert into Opcion values (017, 034, 2, 'texto', 'No');

insert into Opcion values (018, 035, 1, 'texto', 'Si');

insert into Opcion values (018, 036, 2, 'texto', 'No');

insert into Opcion values (019, 037, 1, 'texto', 'Si');

insert into Opcion values (019, 038, 2, 'texto', 'No');

insert into Opcion values (020, 039, 1, 'texto', 'Si');

insert into Opcion values (020, 040, 2, 'texto', 'No');

insert into Opcion values (021, 041, 1, 'texto', 'Si');

insert into Opcion values (021, 042, 2, 'texto', 'No');

insert into Opcion values (022, 043, 1, 'texto', 'Si');

insert into Opcion values (022, 044, 2, 'texto', 'No');

insert into Opcion values (023, 045, 1, 'texto', 'Si');

insert into Opcion values (023, 046, 2, 'texto', 'No');

insert into Opcion values (024, 047, 1, 'texto', 'Si');

insert into Opcion values (024, 048, 2, 'texto', 'No');

insert into Opcion values (025, 049, 1, 'texto', 'Si');

insert into Opcion values (025, 050, 2, 'texto', 'No');

insert into Opcion values (026, 051, 1, 'texto', 'Si');

insert into Opcion values (026, 052, 2, 'texto', 'No');

insert into Opcion values (027, 053, 1, 'texto', 'Si');

insert into Opcion values (027, 054, 2, 'texto', 'No');

insert into Opcion values (028, 055, 1, 'texto', 'Si');

insert into Opcion values (028, 056, 2, 'texto', 'No');

insert into Opcion values (029, 057, 1, 'texto', 'Si');

insert into Opcion values (029, 058, 2, 'texto', 'No');

insert into Opcion values (030, 059, 1, 'texto', 'Si');

insert into Opcion values (030, 060, 2, 'texto', 'No');

insert into Opcion values (031, 061, 1, 'texto', 'Si');

insert into Opcion values (031, 062, 2, 'texto', 'No');

insert into Opcion values (032, 063, 1, 'texto', 'Si');

insert into Opcion values (032, 064, 2, 'texto', 'No');

insert into Opcion values (033, 065, 1, 'texto', 'Si');

insert into Opcion values (033, 066, 2, 'texto', 'No');

insert into Opcion values (034, 067, 1, 'texto', 'Si');

insert into Opcion values (034, 068, 2, 'texto', 'No');

insert into Opcion values (035, 069, 1, 'texto', 'Si');

insert into Opcion values (035, 070, 2, 'texto', 'No');

insert into Opcion values (036, 071, 1, 'texto', 'Si');

insert into Opcion values (036, 072, 2, 'texto', 'No');

insert into Opcion values (037, 073, 1, 'texto', 'Si');

insert into Opcion values (037, 074, 2, 'texto', 'No');

insert into Opcion values (038, 075, 1, 'texto', 'Si');

insert into Opcion values (038, 076, 2, 'texto', 'No');

insert into Opcion values (039, 077, 1, 'texto', 'Si');

insert into Opcion values (039, 078, 2, 'texto', 'No');

insert into Opcion values (040, 079, 1, 'texto', 'Si');

insert into Opcion values (040, 080, 2, 'texto', 'No');

insert into Opcion values (041, 081, 1, 'texto', 'Si');

insert into Opcion values (041, 082, 2, 'texto', 'No');

insert into Opcion values (042, 083, 1, 'texto', 'Si');

insert into Opcion values (042, 084, 2, 'texto', 'No');

insert into Opcion values (043, 085, 1, 'texto', 'Si');

insert into Opcion values (043, 086, 2, 'texto', 'No');

insert into Opcion values (044, 087, 1, 'texto', 'Si');

insert into Opcion values (044, 088, 2, 'texto', 'No');

insert into Opcion values (045, 089, 1, 'texto', 'Si');

insert into Opcion values (045, 090, 2, 'texto', 'No');

insert into Opcion values (046, 091, 1, 'texto', 'Si');

insert into Opcion values (046, 092, 2, 'texto', 'No');

insert into Opcion values (047, 093, 1, 'texto', 'Si');

insert into Opcion values (047, 094, 2, 'texto', 'No');

insert into Opcion values (048, 095, 1, 'texto', 'Si');

insert into Opcion values (048, 096, 2, 'texto', 'No');

insert into Opcion values (049, 097, 1, 'texto', 'Si');

insert into Opcion values (049, 098, 2, 'texto', 'No');

insert into Opcion values (050, 099, 1, 'texto', 'Si');

insert into Opcion values (050, 100, 2, 'texto', 'No');

/\*

DEBUG:

define login = '11496939'

\*/

drop view profesorconsulta;

set verify off;

create view profesorconsulta as select count(e.Cod\_Pregunta) as preguntas from Imparte i, Formada\_por f, Evalua e where i.CODASIGNAT = f.cod\_asign and f.cod\_tema = e.cod\_tema and i.DNI = &login;