



Universitat de les
Illes Balears



Treball Final de Grau

GRAU D'ENGINYERIA ELECTRÒNICA INDUSTRIAL I
AUTOMÀTICA

Desarrollo de una APP para agilizar el
proceso de revisión de exámenes

RAMÓN PASCUAL GONZÁLEZ

Tutors

Javier Antich Tobaruela

Alberto Ortiz Rodríguez

Escola Politècnica Superior
Universitat de les Illes Balears
Palma, 6 de septiembre de 2017

ÍNDICE GENERAL

Índice general	i
Acrónimos	iii
Resumen	v
1 Introducción	1
1.1 ¿Qué problema pretendemos resolver?	1
1.2 ¿Qué solución queremos dar al problema?	1
1.3 Estructura del documento	2
2 Herramientas de desarrollo	3
2.1 MaríaDB	3
2.2 Grocery CRUD	4
2.3 Android Studio	5
2.4 Genymotion	5
2.5 NetBeans	6
2.6 XAMPP	6
3 Diseño de la base de datos	7
3.1 Modelo Entidad-Relación (MER)	7
3.1.1 Entidades y atributos	8
3.1.2 Interrelaciones	10
3.2 Modelo Relacional (MR)	14
3.3 Modificaciones en el MER y MR a consecuencia del Grocery CRUD	16
3.4 Tipos de datos en MaríaDB	20
3.5 Justificación de los tipos de datos utilizados	21
3.6 Views de la base de datos	24
3.7 Código para crear la base de datos en MaríaDB	24
4 Diseño de la aplicación web	25
4.1 Controlador o Controller	25
4.1.1 Acción de seleccionar	27
4.1.2 Creación de la contraseña de los profesores	27
4.1.3 Encriptar contraseña	28
4.1.4 Visualización del contenido de cada pregunta	28
4.2 Vista o View	28

4.3	Pantallas de la aplicación Web	30
4.3.1	Pantalla de selección de asignatura	30
4.3.2	Pantalla de selección del año académico	30
4.3.3	Pantalla de selección de exámenes	31
4.3.4	Pantalla de edición de modelos	31
4.3.5	Pantalla de creación de preguntas	32
4.3.6	Pantalla de soluciones de cada alumno	32
4.3.7	Pantalla para dar de alta a los alumnos	33
4.3.8	Pantalla para dar de alta a los profesores	33
5	Diseño de la aplicación móvil y servidor	35
5.1	Diseño de la aplicación Cliente	35
5.1.1	Pantallas creadas	37
5.2	Diseño de la aplicación Servidor	42
6	Pruebas de verificación	47
7	Conclusiones	51
8	Anexos	53
8.1	Base de datos	53
8.2	Grocery CRUD	67
8.2.1	Controlador	67
8.2.2	Vistas	78
8.3	Aplicación cliente	82
8.3.1	Pantalla de login	82
8.3.2	Pantalla de listado de asignaturas	87
8.3.3	Pantalla de selección del examen del año actual	91
8.3.4	Pantalla de selección de preguntas	97
8.3.5	Pantalla de soluciones de una pregunta	105
8.3.6	Pantalla de selección de exámenes de años anteriores	117
8.3.7	Pantalla de selección del tipo de examen	121
8.3.8	Pantalla de selección de preguntas de exámenes de años anteriores	124
8.3.9	Pantalla de solución de un pregunta de años anteriores	131
8.4	Aplicación Servidor	137
	Bibliografía	145

ACRÓNIMOS

PK Primary Key

MR Modelo Relacional

MER Modelo Entidad-Relación

TFG Trabajo Final de Grado

UIB Universitat de les Illes Balears

IDE Integrated Development Environment

UML Unified Modeling Language

APP 'Application'

UK 'Unique'

FK 'Foreign Key'

UML Unified Modeling Language

N 'NULL'

NN 'NOT NULL'

MD5 Message-Digest Algorithm 5

RESUMEN

Con el objetivo de reducir el tiempo destinado a la revisión de los exámenes, tanto por parte de los alumnos como de los profesores, se ha decidido implementar una aplicación con interfaz web y móvil que acceda a una base de datos centralizada. También se implementa una aplicación servidor, esta aplicación se comunicará con la aplicación móvil mediante el uso de sockets.

La interfaz web está destinada al uso por parte del profesor puesto que será desde aquí donde se subirán los exámenes y las correcciones individualizadas, es decir, será el lugar desde donde se alimentará a la base de datos. Por otra parte, los alumnos accederán a partir de una aplicación móvil, donde visualizarán, para cada asignatura que estén cursando, tanto sus exámenes actuales como anteriores junto con sus respectivas correcciones.

El objetivo a largo plazo de esta aplicación es que se acabe uniendo con la 'Application' (APP) ya existente de la Universitat de les Illes Balears (UIB), AgendaUIB, puesto que de esta manera se le añadiría una funcionalidad hasta ahora no cubierta.

El desarrollo de este Trabajo Final de Grado (TFG) ha requerido de conocimientos sobre el diseño de Bases de Datos y los lenguajes de programación: HTML, JAVA, SQL y PHP.

INTRODUCCIÓN

1.1 ¿Qué problema pretendemos resolver?

Actualmente se pierde mucho tiempo durante el proceso de revisión de exámenes ya que el alumno, cuando llega al despacho del profesor, desconoce cómo se ha corregido y dónde ha fallado; incluso algunos alumnos no recuerdan lo que escribieron en el examen. De esta forma, el propósito principal de este TFG es hacer más eficiente el proceso de revisiones de exámenes para los futuros alumnos de la UIB.

1.2 ¿Qué solución queremos dar al problema?

Para solucionar este problema, se ha decidido desarrollar tres aplicaciones distintas: web, móvil y servidor. El funcionamiento consiste en que el profesor inserte datos en la base de datos a través de la aplicación web, para poder ser consultada posteriormente por los alumnos en la aplicación móvil. Para poder insertar valores en la base de datos la aplicación web se conectará con la base de datos a través de un servidor web. Para mostrar el contenido de la base de datos en la aplicación móvil, se deben realizar consultas a la base de datos a través de la aplicación servidor. Estas consultas siguen un esquema Cliente-Servidor. De esta forma, la aplicación servidor realizará las consultas a la base de datos, en función de las peticiones que le haga la aplicación móvil.

El funcionamiento descrito se ilustra en la figura 1.1.

De esta forma, la aplicación móvil permite dar a conocer a los alumnos, toda la información que necesiten para la decisión de asistir o no a la revisión; y en el caso de decidir asistir, ir más preparados y conseguir agilizar el proceso.

1. INTRODUCCIÓN

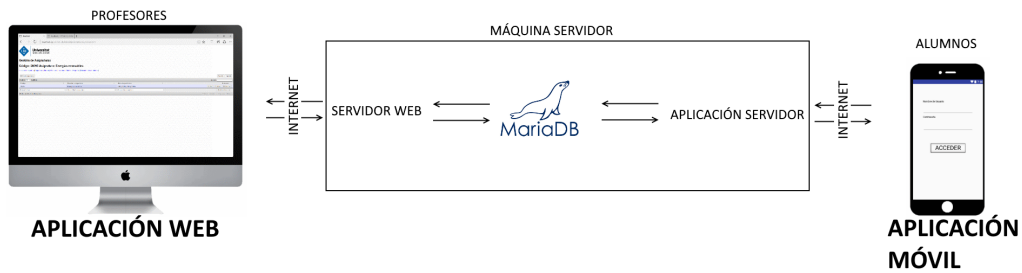


Figura 1.1: Esquema de funcionamiento

Por seguridad, cada alumno dispondrá de unas credenciales que le permitirán acceder a las asignaturas que estén matriculados en el curso actual, por esto se implementa un sistema de login.

1.3 Estructura del documento

El presente documento está estructurado de la siguiente manera:

- El segundo capítulo, mostrará las herramientas que se han usado para desarrollar el TFG. Estas herramientas, se explicarán brevemente y se justificará su selección.
- En el tercer capítulo, se describirá el proceso de creación de la base de datos, junto con todos los problemas encontrados y como consecuencia los cambios realizados.
- En el cuarto capítulo, aparece la explicación de los fragmentos de código más destacados para crear los distintos formularios de la aplicación web, junto con las operaciones de cada uno de ellos.
- En el quinto capítulo, aparece la misma información que en el capítulo cuatro, pero esta vez referenciado a la aplicación móvil y a la aplicación servidor.
- En el sexto capítulo, será donde se muestren las pruebas de verificación para demostrar el correcto funcionamiento de las aplicaciones.
- En el último capítulo se muestra la conclusión del TFG

HERRAMIENTAS DE DESARROLLO

Este trabajo ha requerido del uso de las siguientes herramientas de desarrollo: MaríaDB, Grocery CRUD, Android Studio, Genymotion, NetBeans y XAMPP. Estas herramientas se explican y detallan brevemente a continuación.

2.1 MaríaDB

MaríaDB es un gestor de base de datos de software libre muy popular y es usado por compañías pioneras en el ámbito de la tecnología como Google, Youtube y Facebook.

Se ha elegido este gestor de base de datos frente al resto porque destaca en varios factores:

- Es simple.
- Dispone de gran cantidad de documentación.
- Es gratuito.
- Proporciona gran conectividad, velocidad y seguridad para accesos a las base de datos desde Internet.
- Su ejecución requiere de pocos recursos en términos de consumo de memoria y procesador..

En la figura 2.1 se puede apreciar la consola de MaríaDB tras ejecutar ciertas instrucciones sobre la base de datos del **TFG**.

2. HERRAMIENTAS DE DESARROLLO

```
mysql -h localhost -u root
MariaDB [ app_database ]> show tables;
+-----+
| Tables_in_app_database |
+-----+
| alumnos                |
| anos_academicos        |
| asignaturas            |
| asignaturas_anos_academicos |
| cursa                  |
| examenes               |
| imparte                |
| modelos                |
| modelos_preguntas     |
| modelos_preguntas_alumnos |
| preguntas              |
| profesores             |
| vw_asignaturas_anos_academicos |
| vw_cursa               |
| vw_examenes           |
| vw_imparte             |
| vw_modelos             |
| vw_modelos_preguntas  |
| vw_modelos_preguntas_alumnos |
+-----+
19 rows in set (0.00 sec)

MariaDB [ app_database ]> desc examenes;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| gid_examenes | bigint(20) unsigned | NO | PRI | NULL | auto_increment |
| fecha_examen | datetime | NO | | NULL | |
| nom_examen | varchar(200) | NO | | NULL | |
| gid_asignatura | bigint(20) unsigned | NO | MUL | NULL | |
| fecha_revision | datetime | YES | | NULL | |
| coment_generales | text | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.01 sec)

MariaDB [ app_database ]>
```

Figura 2.1: Consola de la base de datos MaríaDB

2.2 Grocery CRUD

Grocery CRUD es un programa libre y de código abierto que automatiza la creación de formularios web mediante código PHP. Su nombre es un acrónimo de las cuatro operaciones que el programa incluye por defecto en cada formulario web (Create, Read, Update y Delete). Una vez se ha creado la base de datos, se deberá especificar sobre qué tabla o tablas queremos generar cada formulario web, y qué datos de esas tablas se quiere que se muestre. Se ha elegido este generador de formularios web por varias razones:

- Es compatible con un gran número de navegadores.
- El código es simple.
- Ofrece una gran capacidad de personalización de los formularios.
- Es compatible con MaríaDB.
- Es gratuito.

También es cierto que este programa tiene ciertas deficiencias en la documentación y en la generación de formularios sobre interrelaciones M:N, pero barajando todas las alternativas se ha considerado esta herramienta la más idónea para el caso que nos atañe.

2.3 Android Studio

Android Studio es un entorno de desarrollo integrado o Integrated Development Environment (IDE) para el sistema operativo Android que usa JAVA como lenguaje de programación. Es la alternativa al entorno ECLIPSE y nos da una mayor flexibilidad, claridad y robustez a la hora de programar. Se ha usado este programa para la creación de la aplicación móvil. (Ver figura 2.2).

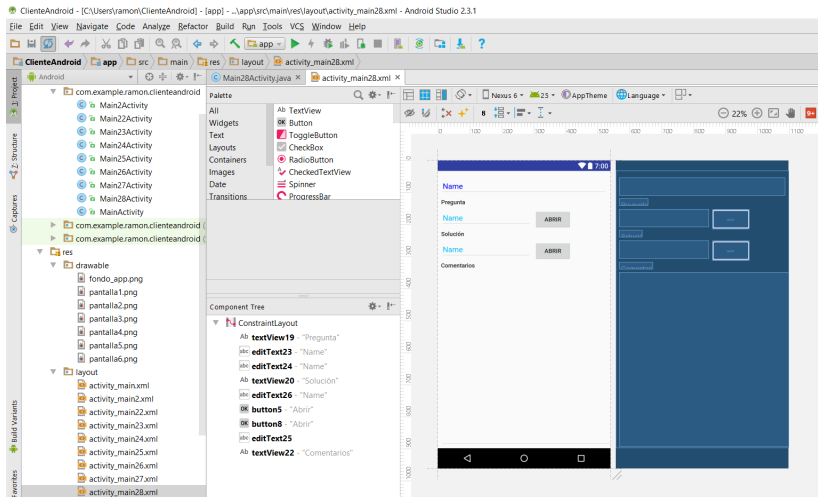


Figura 2.2: Pantalla de la aplicación Android Studio

2.4 Genymotion

Genymotion es un emulador móvil gratuito que destaca por su rendimiento y velocidad. Se ha usado para poder simular, desde el ordenador, la ejecución de la aplicación móvil en distintos dispositivos. Esto nos ha facilitado la depuración de dicha APP. En la figura 2.3 se muestra la pantalla de selección de dispositivo.

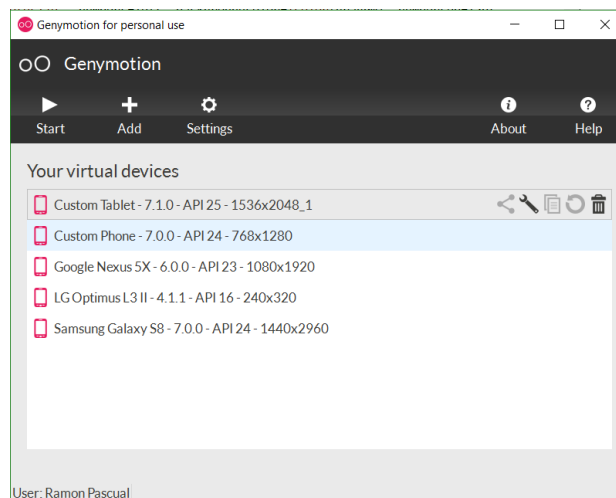


Figura 2.3: Pantalla de selección del móvil o tablet a emular

2.5 NetBeans

De la misma forma que Android Studio, NetBeans es un **IDE** que se programa en **JAVA**. En este **TFG** se ha usado para crear la aplicación servidor que conectará con la base de datos y la aplicación móvil. En la figura 2.4 se puede apreciar la pantalla principal de NetBeans.

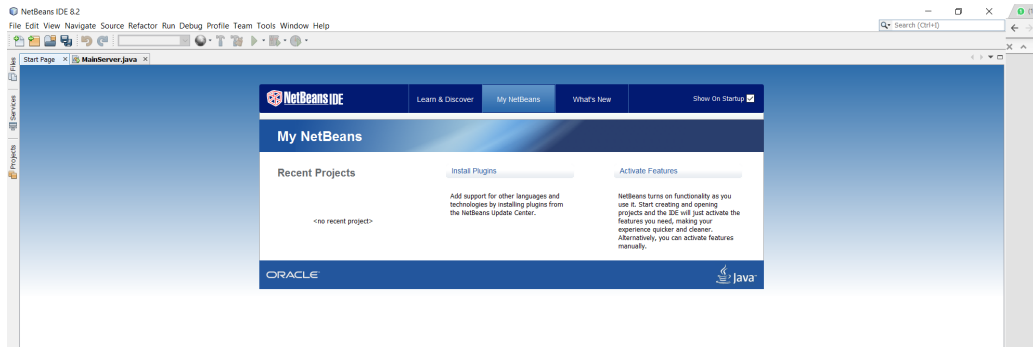


Figura 2.4: Pantalla principal de NetBeans

2.6 XAMPP

XAMPP es un paquete de software libre que incorpora un servidor web con soporte para los lenguajes de programación PHP y Pearl, y una base de datos MaríaDB. Su nombre es un acrónimo de X, Apache, MariaDB, PHP y Pearl.

- **X**, hace referencia a que es un programa multiplataforma, es decir, funciona en todos los sistemas operativos.
- **Apache**, es el servidor web.
- **MariaDB**, es la base de datos.
- **PHP**, es un lenguaje de programación interpretado.
- **Pearl**, es un lenguaje de programación que toma características del lenguaje C.

Entre las ventajas de XAMPP destaca su fácil instalación y la necesidad de realizar configuraciones mínimas.

DISEÑO DE LA BASE DE DATOS

El primer paso para diseñar una base de datos es crear el Modelo Entidad-Relación (**MER**) y el Modelo Relacional (**MR**). Como segundo y último paso, estos modelos se convertirán a tablas en MaríaDB.

3.1 Modelo Entidad-Relación (MER)

El **MER** es una herramienta que nos permite representar de forma visual la base de datos. En este caso se ha seguido el lenguaje de modelado Unified Modeling Language (**UML**), que permite visualizar y construir un sistema. Para entender los siguientes apartados es necesario explicar los conceptos de: entidad, atributo e interrelación.

Entidad

Es la representación de un conjunto de elementos de la base de datos. En el **MER** están representados como rectángulos.

Atributo

Son las características que se desean guardar de las distintas entidades. Se escriben dentro de la entidad a la que pertenecen. Los atributos pueden ser 'Unique' (**UK**), 'Foreign Key' (**FK**) o Primary Key (**PK**). La diferencia entre estas tres restricciones es la siguiente: un atributo **UK** es aquel que su valor es único pero puede ser 'NULL' (**N**), un atributo **PK** vuelve a ser único pero esta vez su valor es 'NOT NULL' (**NN**), finalmente un atributo **FK** es una **PK** que se ha heredado al realizar una interrelación con otra tabla, en este caso su valor puede ser **N**.

Interrelación

Determina cómo se relacionan las diferentes entidades. Pueden ser binarias si relacionan dos entidades, ternarias si relacionan tres entidades, ..., y n-arias si relacionan

n entidades con $n > 3$. Una interrelación tendrá siempre una conectividad, que en el caso de interrelaciones binarias podrá ser 1:1, 1:N y M:N, teniendo M y N el significado de muchos. La conectividad de una interrelación expresa el número de elementos de una entidad con los que un elemento de la otra entidad puede estar asociado según la interrelación. Como se usa el lenguaje UML la nomenclatura del tipo de conectividad se muestra en el cuadro 3.1.

Conectividad	
0..1	Cero o uno
1	Uno
1..*	Uno o muchos
*	Muchos

Cuadro 3.1: Tabla de conectividad UML

3.1.1 Entidades y atributos

Para determinar las entidades y atributos de han tenido en cuenta los siguientes factores:

- Cada profesor podrá insertar modelos a cada uno de sus exámenes.
- Los exámenes estarán constituidos por preguntas que el profesor previamente ha definido.
- Un mismo alumno puede estar matriculado en una misma asignatura pero en años académicos distintos.
- Se desea saber que profesor ha corregido que pregunta y a que alumno.
- Es importante conocer la fecha en la que se ha realizado el examen.

Las entidades que se han incluido en el MER son las siguientes:

- Alumnos
- Años académicos
- Asignaturas
- Asignaturas-Años académicos
- Exámenes
- Modelos
- Modelos-Preguntas
- Modelos-Preguntas-Alumnos
- Preguntas

- Profesores

A su vez, cada entidad está compuesta por los siguientes atributos:

- **Alumnos**

- Nombre de usuario
- Nombre
- Primer apellido
- Segundo apellido
- Foto
- Contraseña

- **Años académicos**

- Año académico

- **Asignaturas**

- Código
- Nombre

- **Asignaturas-Años académicos**

Carece de atributos.

- **Exámenes**

- Nombre del examen
- Fecha del examen
- Fecha de la revisión
- Comentarios generales

- **Modelos**

- Número de modelo

- **Modelos-Preguntas**

- Número de la pregunta

- **Modelos-Preguntas-Alumnos**

- Solución del alumno
- Corrección del profesor
- Comentarios individuales

- **Preguntas**

- Código de la pregunta

- Enunciado
- Solución del profesor
- Comentarios generales

- **Profesores**

- Nombre de usuario
- Nombre
- Primer apellido
- Segundo apellido
- Foto
- Contraseña
- Correo electrónico
- Código de despacho

3.1.2 Interrelaciones

Por necesidad de poder definir futuras interrelaciones, se ha decidido transformar las interrelaciones M:N, que posteriormente se relacionarían con una tercera tabla, en dos 1:N. El proceso de realización es sencillo, únicamente es necesario crear una nueva entidad intermedia que ejercerá de conexión con ambas tablas. De esta forma, cada vez que ambas tablas se tengan que relacionar con una tercera, únicamente será necesario relacionarla con la tabla intermedia.

Las interrelaciones que se han incluido en el **MER** son las siguientes:

Entre la entidad *Asignaturas* y la entidad *Años académicos* hay una interrelación M:N; es decir, una asignatura puede impartirse en N años académicos y en un año académico pueden impartirse M asignaturas. Como se ha explicado anteriormente, se ha decidido transformar esta interrelación M:N en dos interrelaciones 1:N, creando así una entidad intermedia llamada *Asignaturas-Años académicos*, como se aprecia en la figura 3.1.

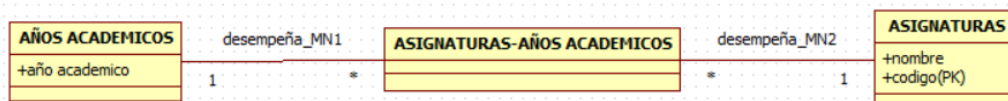


Figura 3.1: Interrelación intermedia entre asignaturas y años académicos

Esta entidad intermedia se relaciona con los *Alumnos* y los *Profesores*. En ambos casos con conectividad M:N; es decir, dada una asignatura en un determinado año académico pueden cursarla/impartirla N alumnos/profesores. A su vez, dado un alumno/profesor puede cursar/impartir distintas asignaturas en un año académico. Todo esto queda plasmado en la figura 3.2.

3.1. Modelo Entidad-Relación (MER)

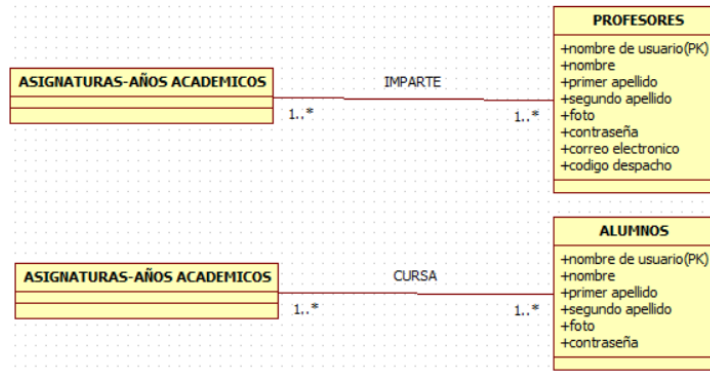


Figura 3.2: Interrelación M:N entre asignaturas y años académicos con los profesores y los alumnos

Al mismo tiempo, la entidad *Asignaturas* se relaciona con la entidad *Preguntas* porque, de esta manera, conseguimos filtrar las preguntas de los exámenes en función de las asignaturas y evitamos sobrecargar al usuario con información irrelevante de otras materias (ver figura 3.3).

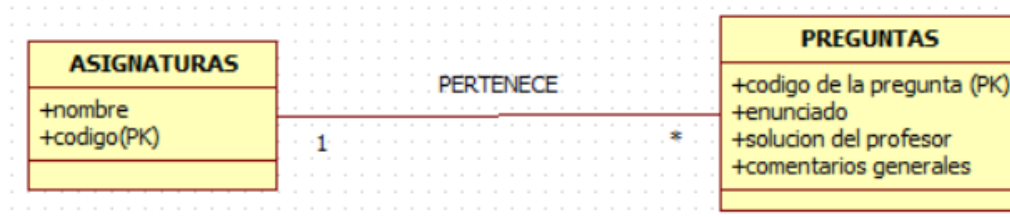


Figura 3.3: Interrelación 1:N entre asignaturas y preguntas

La entidad *Exámenes* se relaciona con la entidad *Modelos*. Esta interrelación es de tipo 1:N, es decir, un examen puede tener N modelos, pero un modelo sólo puede pertenecer a un único examen (ver figura 3.4).

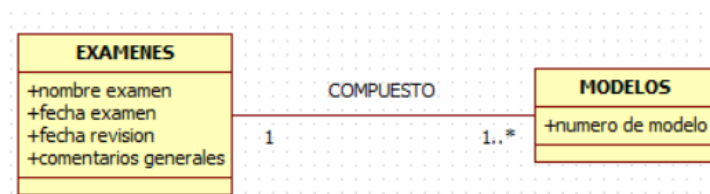


Figura 3.4: Interrelación 1:N entre exámenes y modelos

La interrelación entre las entidades *Modelos* y *Preguntas* es de tipo M:N, puesto que una misma pregunta puede aparecer en varios modelos de un examen y en un mismo modelo pueden aparecer distintas preguntas. Por las mismas razones explicadas anteriormente, se separan en dos interrelaciones 1:N. En este caso, la entidad intermedia se ha denominado *Modelos-Preguntas* (ver figura 3.5).

3. DISEÑO DE LA BASE DE DATOS

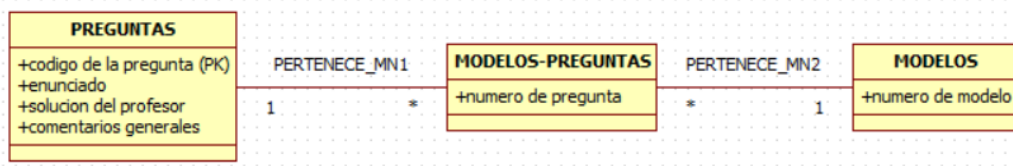


Figura 3.5: Interrelación intermedia entre modelos y preguntas

Esta entidad intermedia *Modelos-Preguntas* se relaciona con *Alumnos* de forma M:N. Otra vez se decide separarla en dos interrelaciones 1:N, creando la entidad intermedia *Modelos-Preguntas-Alumnos*. Esto permitirá determinar para cada alumno, qué preguntas ha contestado y a qué modelo de examen pertenecían estas preguntas (ver figura 3.6).

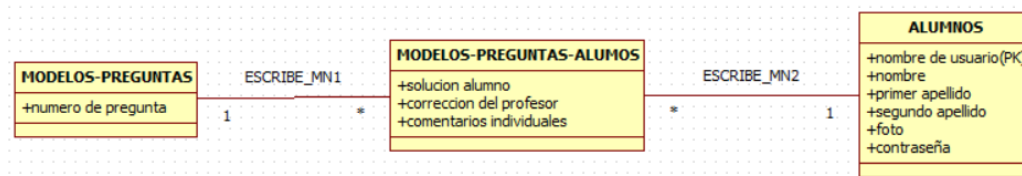


Figura 3.6: Interrelación intermedia entre modelos-preguntas y alumnos

Finalmente, la entidad intermedia *Modelos-Preguntas-Alumnos* se relaciona de forma 1:N con *Profesores*, puesto que un profesor puede corregir N preguntas de un determinado alumno, pero una pregunta de un alumno sólo puede ser corregida por un profesor (ver figura 3.7).

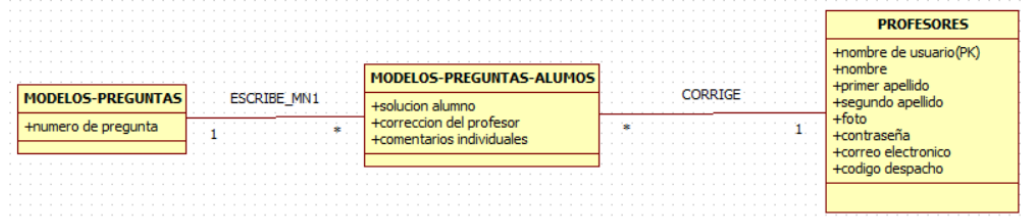


Figura 3.7: Interrelación 1:N entre modelos-preguntas y profesores

El MER completo aparece en la figura 3.8.

3.1. Modelo Entidad-Relación (MER)

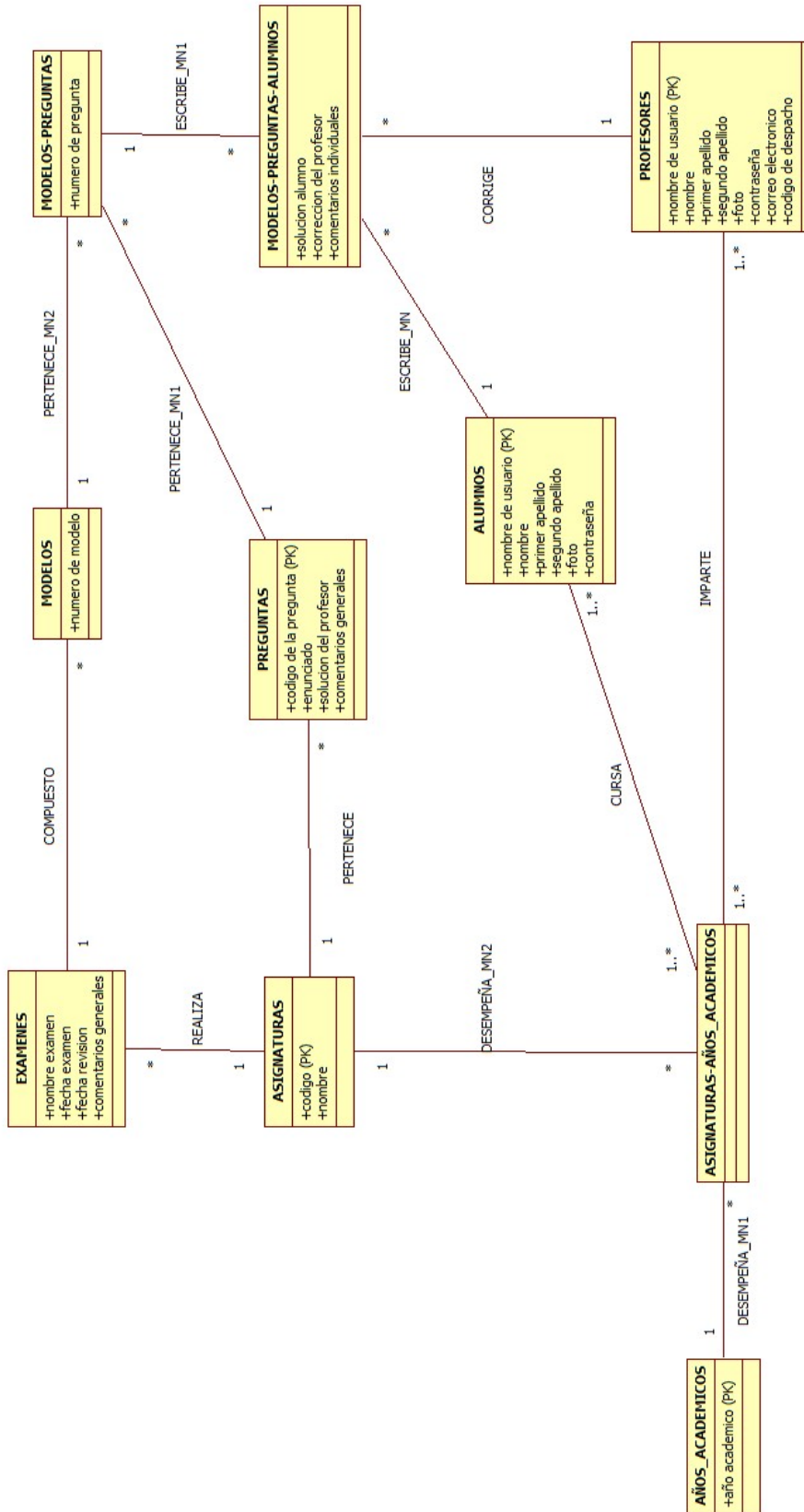


Figura 3.8: MER de la base de datos

3.2 Modelo Relacional (MR)

El **MR** se basa en el **MER** y tiene como objetivo modelar las entidades y sus interrelaciones en forma de predicado. Se caracteriza por ser más simple e intuitivo que el **MER** pues se indican las tablas que se modelarán en la base de datos junto con sus atributos incluyendo los heredados. En las interrelaciones 1:N se hereda la **PK** de la tabla cuya conectividad es 1 hacia la tabla cuya conectividad es N, en el caso de las interrelaciones M:N se crea una entidad nueva con el nombre de la interrelación y ambas **PK**.

El **MR** de la base de datos queda de la siguiente forma:

ALUMNOS (nombre de usuario, nombre, primer apellido, segundo apellido,
 PK
 NN NN NN N
 foto, contraseña)
 N NN

AÑOS ACADÉMICOS (año académico)
 PK
 NN

ASIGNATURAS (código asignatura, nombre)
 PK
 NN NN

ASIGNATURAS-AÑOS ACADÉMICOS (código asignatura, año académico)
 PK,FK PK,FK
 NN NN

CURSA (código asignatura, año académico, nombre de usuario del alumno)
 PK,FK PK,FK PK,FK
 NN NN NN

EXÁMENES (fecha examen, nombre examen, código asignatura,
 PK PK PK,FK
 NN NN NN
 fecha revisión, comentarios generales)
 N N

PROFESORES (nombre de usuario, nombre, primer apellido,

PK

NN

NN

NN

segundo apellido, foto, contraseña, correo electrónico, despacho)

N

N

NN

N

N

IMPARTE (código asignatura, año académico, nombre de usuario del profesor)

PK,FK

PK,FK

PK,FK

NN

NN

NN

MODELOS (número modelo, fecha examen, nombre examen,

PK

PK,FK

PK,FK

NN

NN

NN

código asignatura)

PK,FK

NN

PREGUNTAS (código pregunta, código asignatura, enunciado,

PK

FK

NN

NN

NN

solución profesor, comentarios generales)

N

N

MODELOS-PREGUNTAS (número modelo, fecha examen, nombre examen,

PK,FK

PK,FK

PK,FK

NN

NN

NN

código asignatura, código pregunta, número pregunta)

PK,FK

PK,FK

NN

NN

NN

MODELOS-PREGUNTAS-ALUMNOS (número modelo, fecha examen,

PK,FK

PK,FK

NN

NN

nombre examen, código asignatura, código pregunta, nombre de usuario del alumno,

PK,FK

PK,FK

PK,FK

PK,FK

NN

NN

NN

NN

nombre de usuario del profesor corrector, solución alumno, corrección profesor,

FK

N

NN

N

comentarios individuales)

N

3.3 Modificaciones en el MER y MR a consecuencia del Grocery CRUD

El Grocery CRUD tiene ciertas limitaciones y peculiaridades que han provocado que se tenga que modificar el diseño inicial de la base de datos. Una de estas peculiaridades es que la **PK** de las tablas debe estar formada por un único atributo. Un ejemplo que muestra este cambio, es la tabla EXÁMENES, esta tabla tenía una **PK** formada por 3 atributos; ahora pasa a tener un único atributo con nombre *gid_examen*. Una segunda peculiaridad es el hecho de que cuando se efectúa una interrelación de tipo M:N, la **PK** de las entidades interrelacionadas deben ser de tipo *auto incremental* (en bases de datos, un atributo de tipo auto incremental es un atributo de tipo entero que se inicia en 1 y va aumentando de 1 en 1 automáticamente cada vez que se inserta un nuevo dato en la tabla). Un ejemplo que muestra esta segunda peculiaridad es el que sucede en la interrelación M:N CURSA entre ALUMNOS y ASIGNATURAS-AÑOS ACADÉMICOS, donde nos vemos obligados a añadir el atributo *gid_alumnos* en la entidad ALUMNOS, puesto que su **PK** aún teniendo un solo atributo, este no era de tipo *auto incremental*. Por tanto, el **MER** y el **MR** anteriores sufrirán ciertas modificaciones.

El **MER** final aparece en la figura 3.9.

3.3. Modificaciones en el MER y MR a consecuencia del Grocery CRUD

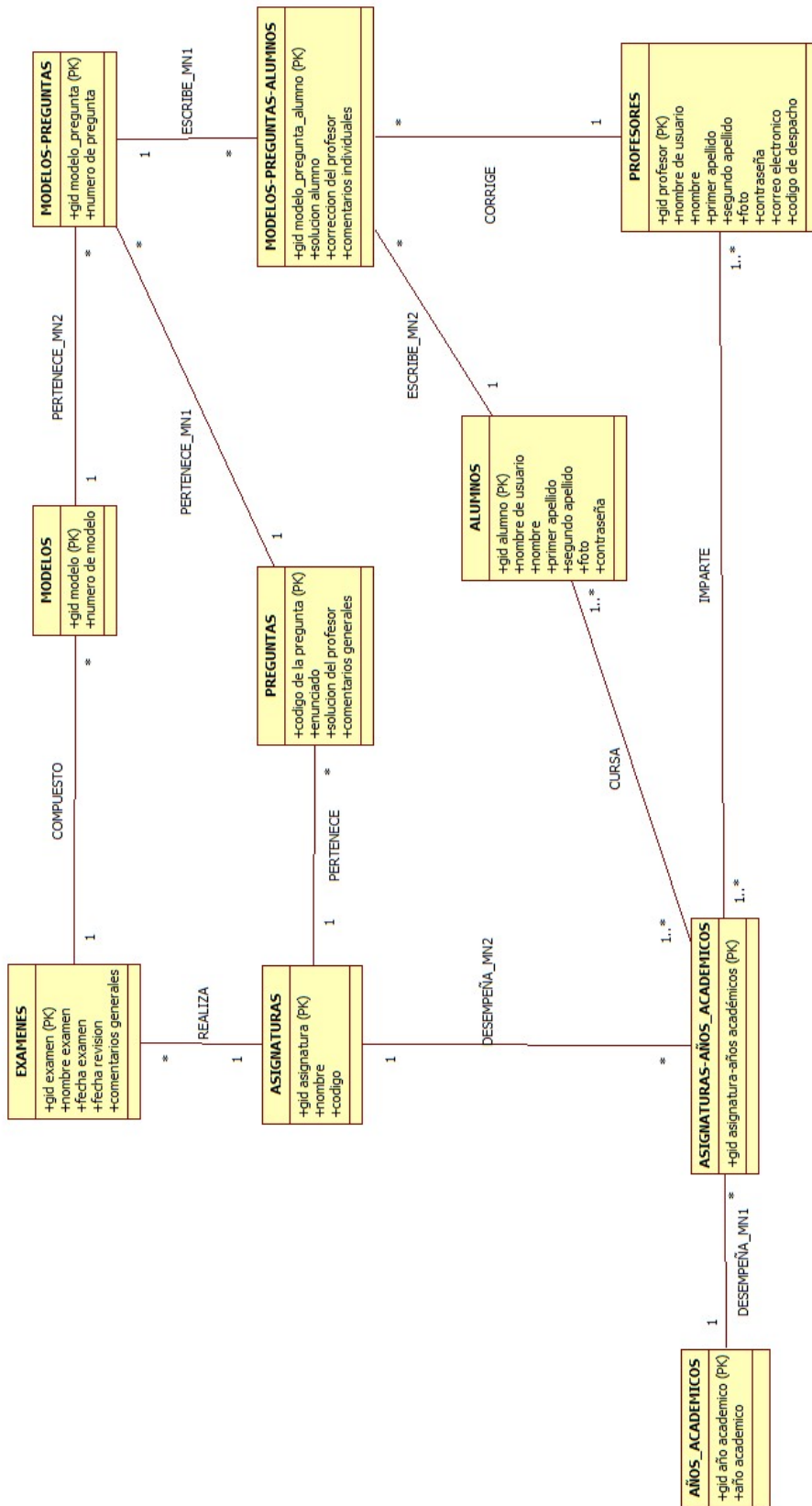


Figura 3.9: MER final de la base de datos

3. DISEÑO DE LA BASE DE DATOS

Estos cambios, han provocado que los atributos que formaban la **PK** en las relaciones del **MR** inicial, se definan como **UK**. El **MR** resultante es el siguiente:

ALUMNOS (gid alumno, nombre de usuario, nombre, primer apellido,
segundo apellido, foto, contraseña)

PK UK
NN NN NN NN
N N NN

AÑOS ACADÉMICOS (gid año académico, año académico)

PK UK
NN NN

ASIGNATURAS (gid asignatura, código asignatura, nombre)

PK UK
NN NN NN

ASIGNATURAS-AÑOS ACADÉMICOS (gid asignatura año, gid asignatura,
gid año académico)

PK FK,UK
NN NN
FK,UK
NN

CURSA (gid curso, gid alumno, gid año académico)

PK FK,UK FK,UK
NN NN NN

EXÁMENES (gid examen, gid asignatura, fecha examen, nombre examen,
fecha revisión, comentarios generales)

PK UK,FK UK UK
NN NN NN N
N N

PROFESORES (gid profesor, nombre de usuario, nombre, primer apellido,
segundo apellido, foto, contraseña, correo electrónico, despacho)

PK UK
NN NN NN NN

3.4 Tipos de datos en MaríaDB

Los tipos de datos contemplados en MaríaDB son los siguientes:

- Datos numéricos
 - *TINYINT*: Se trata de un número entero entre -128 y +127 en caso de ser con signo, y de 0 a 255 sin signo.
 - *BOOLEAN*: Este tipo de datos sólo puede valer 0 ó 1.
 - *SMALLINT*: Número entero entre -32768 y +32767 en caso de ser con signo, y de 0 a 65535 sin signo.
 - *MEDIUMINT*: Número entero entre -8388608 y +8388607 en caso de ser con signo, y de 0 a 16777215 sin signo.
 - *INT*: Número entero entre -2147483648 y +2147483647 en caso de ser con signo, y de 0 a 4294967295 sin signo.
 - *BIGINT*: Número entero entre -9223372036854775808 y +9223372036854775807 en caso de ser con signo, y de 0 a 18446744073709551615 sin signo.
 - *DECIMAL*: Es un tipo de datos que permite la inserción de números decimales. El máximo número de decimales admitidos es 30. El número de decimales deseados se puede especificar cuando se crea el atributo; por defecto es cero. El número máximo de dígitos antes del punto decimal es de 65.
 - *FLOAT*: Número decimal de precisión simple. Los valores válidos van desde $-3.402823466 \cdot 10^{38}$ a $-1.175494351 \cdot 10^{-38}$, el 0, y desde $1.175494351 \cdot 10^{-38}$ a $3.402823466 \cdot 10^{38}$.
 - *DOUBLE*: Número decimal de precisión doble. Los valores válidos van desde $-1.7976931348623157 \cdot 10^{308}$ a $-2.2250738585072014 \cdot 10^{-308}$, el 0, y desde $2.2250738585072014 \cdot 10^{-308}$ a $1.7976931348623157 \cdot 10^{308}$.
- Cadenas de datos
 - *CHAR*: Cadena de longitud fija. Longitud entre 0 y 255 caracteres. Los caracteres que no se hayan escrito, se rellenan con espacios.
 - *VARCHAR*: Cadena de longitud variable. La cadena puede contener entre 0 y 65535 caracteres.
 - *BINARY*: Muy similar al tipo de datos CHAR, pero, en este caso, se guarda una cadena de bytes binarios en lugar de una cadena de caracteres no binarios. Su longitud máxima es de 255 bytes.
 - *VARBINARY*: Es la versión con longitud variable del tipo de datos BINARY.
 - *BLOB*: Este tipo permite guardar datos binarios, tales como imágenes u otros tipos de ficheros (por ejemplo, PDFs). Su tamaño máximo es de 65535 bytes.
 - *TINYBLOB*: Permite guardar el mismo tipo de datos que un BLOB, pero esta vez con un tamaño máximo de 255.

- *MEDIUMBLOB*: Permite guardar el mismo tipo de datos que un BLOB, pero esta vez con un tamaño máximo de 16777215 bytes.
 - *LOB*: Permite guardar el mismo tipo de datos que un BLOB, pero esta vez con un tamaño máximo de 4294967295 bytes.
 - *TEXT*: Columna de texto con una longitud máxima de 65535 caracteres.
 - *TINYTEXT*: Columna de texto con una longitud máxima de 255 caracteres.
 - *MEDIUMTEXT*: Columna de texto con una longitud máxima de 16777215 caracteres.
 - *LONGTEXT*: Columna de texto con una longitud máxima de 4294967295 caracteres.
 - *ENUM*: Este tipo de datos restringe los valores que se le pueden dar a un atributo a una lista. Esta lista puede contener hasta 65535 valores distintos.
- datos de fecha y hora
 - *DATE*: La fecha se muestra con el siguiente formato: 'AAAA-MM-DD'. EL rango de valores es de '1000-01-01' a '9999-12-31'.
 - *TIME*: Este tipo de datos determina la hora con precisión de microsegundos: 'HH:MM:SS:sssss'.
 - *DATETIME*: Se trata de una combinación de los tipos de datos DATE y TIME. EL rango de valores se encuentra entre '1000-01-01 00:00:00.000000' y '9999-12-31 23:59:59.999999'.
 - *TIMESTAMP*: El formato es 'AAAA-MM-DD HH:MM:SS'.
 - *YEAR*: Permite guardar exclusivamente el año. Los años deben estar comprendidos entre 1901 y 2155.

3.5 Justificación de los tipos de datos utilizados

Como parte del trabajo de diseño de una base de datos es necesario elegir correctamente el tipo de datos de cada atributo. A continuación se justifica la elección para cada atributo, agrupados según el tipo de datos:

DATETIME:

Fecha de examen

Se elige este tipo de datos porque nos interesa conocer la fecha y la hora en la que se realizará el examen.

Fecha de revisión

Las razones son las mismas que en el caso del atributo 'fecha de examen'.

VARCHAR:

Nombre del examen

Se ha considerado que el nombre de un examen (por ejemplo, 'Primer Parcial', 'Segundo Parcial',...) no será nunca mayor a 200 caracteres.

Número de modelo

Algunos profesores distinguen sus modelos de examen con letras (A,B,...) y otros profesores lo hacen con números (1,2,...) para poder contemplar ambas situaciones se ha definido este atributo como un VARCHAR de 1 caracter de longitud.

Nombre asignatura

Se ha considerado que el nombre de una asignatura no será mayor a 50 caracteres.

Nombre de usuario

Sigue el formato utilizado en la aplicación UIBdigital: tres letras y tres números. Por tanto, es suficiente con seis caracteres. Un ejemplo de nombre de usuario es 'RPG899'.

Nombre

Se ha considerado que el nombre de una persona no será más largo de 30 caracteres.

Primer apellido y Segundo apellido

Es el mismo caso que el atributo Nombre.

Contraseña

Como se encripta según el algoritmo Message-Digest Algorithm 5 (MD5)¹, se usa este tipo de datos con 32 caracteres.

Correo electrónico

Se ha supuesto que un correo electrónico está formado por distintos caracteres, pero que no tiene una extensión mayor a 75.

Código de despacho

El código de los despachos está compuesto por dos letras que identifican el edificio y tres números que identifican el número de despacho. Por tanto, únicamente se deben guardar cinco caracteres. Por ejemplo, el código 'AT252' hace referencia al edificio Anselm Turmeda y al despacho 252.

TINYINT:

Numero de pregunta

De esta manera, cualquier modelo de un examen podrá tener hasta 256 preguntas.

AUTO INCREMENT:

¹Es un tipo de encriptación unidireccional de 128 bits, es decir, 32 símbolos hexadecimales. Es ampliamente usado para comprobar que un archivo no ha sufrido daños.

Código pregunta

Como este atributo tiene como único objetivo identificar la pregunta, se ha elegido este tipo de dato porque se va incrementando automáticamente.

BLOB:

Enunciado

En este atributo se guardará el enunciado de una pregunta en formato .PDF.

Solución profesor

Este atributo tiene el mismo objetivo que en el caso anterior pero con la solución del profesor.

Foto

El objetivo de este atributo es guardar un archivo en formato .PNG o .JPG.

Solución alumno

Este atributo vuelve a tener como objetivo guardar un archivo en formato .PDF.

Corrección profesor

Este atributo vuelve a tener como objetivo guardar un archivo en formato .PDF.

TEXT:

Comentarios generales

En este atributo se guardará un comentario general del profesor. Por tanto, hemos elegido este tipo de datos porque permite guardar largas cadenas de caracteres.

Comentarios individuales

Sucede lo mismo que en el caso de los comentarios generales.

MEDIUMINT:

Código asignatura

El código de una asignatura está compuesto por 5 dígitos. Por tanto, nos basta con usar este tipo de datos. Por ejemplo, la asignatura 'Informática industrial' del Grado en Ingeniería Electrónica Industrial y Automática tiene el código 22423.

YEAR:

Año académico

Un curso académico viene definido por dos años. Por ejemplo, 2017-2018. En nuestro caso, se ha decidido guardar el primer año y calcular el siguiente; es decir, para el ejemplo anterior la variable Año académico tendrá el valor 2017.

3.6 Views de la base de datos

Como se ha expuesto anteriormente, las limitaciones del Grocery CRUD han implicado que se hayan heredado los atributos 'gid', en lugar de los atributos que se tenían que heredar realmente. Esto ha generado un problema, puesto que cuando se quería mostrar los datos heredados en el formulario, la tabla no disponía de ellos. Para solucionar esto se han usado vistas. Una vista en MaríaDB es un tabla virtual cuya estructura se define a partir de los atributos y las relaciones entre las distintas tablas. En pocas palabras, nos permite guardar en un estilo de tabla todos los atributos que deseen. Un ejemplo de diseño de vista es el siguiente:

```

1 CREATE VIEW vw_modelos AS
2 SELECT t1.gid_asignatura, t2.gid_examenes, t3.gid_modelos, t3.modelo, t2.
   fecha_examen, t2.nom_examen, t1.cod_asignatura,
3 CONCAT (t1.nom_asignatura, ' ', t2.nom_examen, ' modelo: ', t3.modelo, ' fecha: ', t2
   .fecha_examen) as exam_asign
4 FROM asignaturas as t1
5 inner join examenes as t2 on
6 t1.gid_asignatura=t2.gid_asignatura
7 inner join modelos as t3 on
8 t3.gid_examenes=t2.gid_examenes;
9

```

Cuando se crea una vista se especifican los atributos que se desean que formen parte de ella. Para hacer esto, se debe determinar cual es su tabla origen. La relación entre las tablas se determinará comparando las primary key en una 'inner join', es decir, se cogerán aquellos valores que sean comunes en ambas tablas, repitiéndolo de forma escalonada con el resto de tablas. Esto se muestra en la figura 3.10.

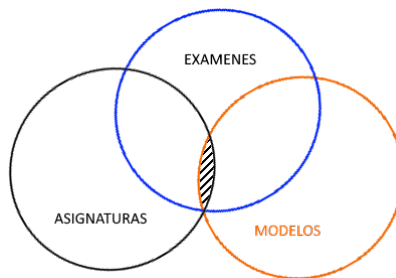


Figura 3.10: Iner join entre Asignaturas, Exámenes y Modelos

3.7 Código para crear la base de datos en MaríaDB

El código que genera la base de datos se puede encontrar en el anexo: '8.1 Base de datos'.

DISEÑO DE LA APLICACIÓN WEB

La interfaz web ha sido diseñada de la forma más intuitiva posible. El movimiento entre pantallas es simple y muy visual. Además, los formularios tienen nombres comprensibles que dotan de toda la información al usuario. Como se ha descrito anteriormente, se ha usado el programa Grocery CRUD para elaborar estos formularios. Cada formulario está definido por dos ficheros un controlador o 'controller' y una vista o 'view'.

4.1 Controlador o Controller

El controlador, es el fichero que define las relaciones entre las tablas de la base de datos, los atributos que se desean enseñar u ocultar, añade acciones adicionales como son las de selección de asignatura o examen y también permite añadir, borrar y modificar la información que aparece en la base de datos. Este fichero se programa con lenguaje PHP. En el controlador se crean distintas funciones y estas harán referencia a los distintos formularios. De esta forma podemos trabajar con un único controlador. A continuación, se muestra y explica un ejemplo de función en el controlador:

```
1 public function alumn_management()  
2 {  
3     $crud = new grocery_CRUD();  
4     $crud->set_theme('datatables');  
5     $crud->set_table('alumnos');  
6     $crud->set_subject('alumnos');  
7     $crud->set_language('spanish');  
8     $crud->set_primary_key('gid_asign_ano', 'vw_asignaturas_anos_academicos');  
9  
10    $crud->set_relation_n_n('asignaturas_y_años_academicos', 'cursa', '  
11        vw_asignaturas_anos_academicos', 'gid_alumno', 'gid_asign_ano', 'asign_ano');  
12  
13    $crud->display_as('nom_usuario_alumn', 'Nombre de usuario');  
14    $crud->display_as('apellido1', 'Primer apellido');  
15    $crud->display_as('apellido2', 'Segundo apellido');
```

4. DISEÑO DE LA APLICACIÓN WEB

```
16
17
18
19 $crud->field_type('password', 'password');
20 $crud->unset_columns('password');
21
22 $crud->required_fields('nom_usuario_alumn', 'nombre', 'password');
23
24 $crud->set_field_upload('foto', 'assets/uploads/files');
25 $crud->callback_before_insert(array($this, 'encrypt_password_callback'));
26
27
28 $output = $crud->render();
29
30 $data = array();
31 $data['cod_asignatura'] = $this->session->userdata('cod_asignatura');
32 $data['nom_asignatura'] = $this->session->userdata('nom_asignatura');
33 $data['nom_formulario'] = 'Alumnos';
34 $output->data = $data;
35 $this->load->view('accesos', $output);
36 }
```

La función anterior es del formulario de los alumnos. Se inicia la función, creando un objeto Grocery CRUD. Se selecciona el estilo de los formularios con la instrucción 'set_theme'. Para posteriormente especificar la tabla sobre la que trabajaremos con la instrucción 'set_table'. Las dos instrucciones siguientes especifican el nombre con el que nos referiremos a la tabla y el idioma con el que se muestren las cosas. Como en este formulario se trabaja con vistas de la base de datos, se tiene que especificar su primary key con la instrucción 'set_primary_key'. Una vez especificado el idioma, la tabla,... se empiezan a usar las instrucciones para relacionar, editar,... la tabla. La instrucción para relacionar una tabla con otra con conectividad M:N es la que aparece en la línea 8 del código anterior. En esta instrucción se deberá especificar el nombre que le queramos dar a la interrelación, la tabla de relación y la tabla de selección, sus respectivas primary key y el atributo que queramos sacar por pantalla. Cómo los nombres de los atributos están guardados de forma abreviada en la base de datos, se usa la instrucción 'display_as' para modificar el nombre y permitir al usuario visualizarlo de forma más clara. Las dos instrucciones siguientes son necesarios porque se esta trabajando sobre una contraseña. En la primera instrucción se especifica que el atributo 'password' es de tipo 'password', es decir, se le dice al Grocery CRUD que este atributo es una contraseña. Esto se hace porque de esta manera tiene un trato diferente. La segunda instrucción, elimina el campo 'password' de la visualización del formulario, puesto que no le aporta información al usuario. Seguidamente, se especifica que campos son obligatorios a la hora de insertar nuevos objetos en la tabla. Estrictamente, se debe especificar que el atributo 'foto' es un campo donde se insertará una imagen, para conseguir esto se usa la instrucción 'set_field_upload'. Una instrucción necesaria y con mucha importancia es la de 'callback_before_insert', esta instrucción llama a la función 'encrypt_password_callback' donde se encriptará en MD5 la contraseña. Finalmente, se ejecuta el render y se guarda en un array el nombre de la asignatura, su código y el nombre del formulario. El render y el array se pasan como parámetro a la vista para poder ejecutarla y tener una interfaz más detallada y personalizada.

4.1.1 Acción de seleccionar

Una operación importante para el mantenimiento de la base de datos es la selección de variables, tales como, año académico, asignatura, examen... Esta selección nos permitirá en un futuro filtrar las peticiones. El siguiente código, hace referencia a la creación del botón 'seleccionar':

```
1 $crud->add_action( 'Seleccionar', base_url( 'img/check_verde.jpg' ), 'examples/
  select_exam' );
```

Una vez se ha constituido el botón, es necesario crear la función que permita guardar la selección hecha. Un ejemplo de función es el siguiente:

```
1 public function select_asig($gid_asignatura)
2 {
3     $this->db->select( 'cod_asignatura , nom_asignatura ')->from( 'asignaturas ')->
4     where( 'gid_asignatura', $gid_asignatura );
5     $query = $this->db->get();
6
7     $newdata=array(
8         'cod_asignatura' => $query->row()->cod_asignatura ,
9         'nom_asignatura' => $query->row()->nom_asignatura ,
10        'gid_asignatura' => $gid_asignatura ,
11    );
12
13    $this->session->set_userdata( $newdata );
14    redirect( 'examples/selec_ano_management' );
15 }
```

En este ejemplo se selecciona el código de la asignatura junto con su nombre y se guarda en un array. Finalmente se redirecciona a la siguiente función.

4.1.2 Creación de la contraseña de los profesores

Cuando se da de alta a los profesores, no se pide que se inserte la contraseña. Esto es debido a que la contraseña se crea de forma automática. El siguiente código hace referencia a esta acción.

```
1 function generate_password_prof($post_array) {
2     $pwd = substr(md5(microtime()), 1, 8);
3     $this->db->query("insert into profesores(nom_usuario_prof, nombre, apellido1 ,
4     apellido2, foto, password, email, despacho) values('".$post_array['
5     nom_usuario_prof']."' , '".$post_array['nombre']."' , '".$post_array['apellido1 '
6     ]."' , '".$post_array['apellido2']."' , '".$
7     $post_array['foto']."' , '".$pwd."' , '".$post_array['email']."' , '".$post_array[
8     'despacho']."' )");
9 }
```

La primera instrucción calcula la contraseña de 8 caracteres en formato MD5. La segunda instrucción inserta dentro de los respectivos campos de la tabla profesores el nombre de usuario, nombre, primer y segundo apellido, foto, la contraseña calculada, email y el despacho.

4.1.3 Encriptar contraseña

Como el único que debe tener acceso a la contraseña es el propio usuario, cuando se inserta en la base de datos se encripta en MD5. Para poder encriptarla, se debe crear una función que se llamará cada vez que se desee cifrar.

```
1 function encrypt_password_callback($post_array) {
2     $post_array['password'] = md5($post_array['password']);
3     return $post_array;
4 }
```

En el código se aprecia como se pasa la contraseña como parámetro, se inserta en la función 'md5' y se devuelve la contraseña cifrada.

4.1.4 Visualización del contenido de cada pregunta

En la pantalla para editar los modelos de un examen, se permite abrir el documento que conforma cada pregunta. Para realizar esto se ha creado la siguiente función:

```
1 public function _callback_webpage_url($value, $row)
2 {
3     $exit = false;
4     $offset = 0;
5     $str = "";
6
7     while ($exit == false) {
8         $pos = strpos($value, ',', $offset);
9         if ($pos !== false) {
10            $len = $pos - $offset;
11            $sep = ", ";
12        }
13        else {
14            $len = strlen($value) - $offset;
15            $sep = "";
16            $exit = true;
17        }
18
19        $substrvalue = substr($value, $offset, $len);
20        $str = $str . "<a href='".base_url('assets/uploads/files/'. $substrvalue) .
21            "'>$substrvalue</a>" . $sep;
22
23        $offset = $pos + 1;
24    }
25    return $str;
26 }
```

4.2 Vista o View

Se trata de un archivo programado en HTML y PHP, que permite visualizar el controlador diseñado. Es aquí donde se podrán crear vínculos a otros formularios, insertar un fondo, un título, incluso textos informativos. A continuación, se muestra el código de una vista:

```
1 <!DOCTYPE html>
```

```

2 <html>
3 <head>
4     <meta charset="utf-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <?php
7 foreach($css_files as $file): ?>
8     <link type="text/css" rel="stylesheet" href="<?php echo $file; ?>" />
9 <?php endforeach; ?>
10 <?php foreach($js_files as $file): ?>
11     <script src="<?php echo $file; ?>"></script>
12 <?php endforeach; ?>
13 <style type='text/css '>
14 body
15 {
16     background-image: url(<?php echo base_url('img/background.jpg'); ?>);
17     background-repeat: repeat;
18     font-family: Arial;
19     font-size: 14px;
20 }
21 a {
22     color: blue;
23     text-decoration: none;
24     font-size: 14px;
25 }
26 </style>
27 </head>
28 <body>
29     <div>
30         
31         <h1 >C&ocute;digo: <font color="blue"><?php echo $data['cod_asignatura']?></font> Asignatura: <font color="blue"><?php echo $data['nom_asignatura']?></font>
32         <h1>Estas editando el examen <font color="blue"><?php echo $data['nom_examen']?></font> con fecha <font color="blue"><?php echo $data['fecha_examen']?></font></h1>
33
34
35         <a href='<?php echo site_url('examples/exam_management')?>'>ATRAS</a> |
36         <a href='<?php echo site_url('examples/model_management')?>'>Modelos</a>
37         |
38         <a href='<?php echo site_url('examples/preg_management')?>'>Preguntas</a>
39         |
40         <a href='<?php echo site_url('examples/exam_preg_alumn_management')?>'>
41         Soluciones de cada alumno</a>
42     </div>
43     <div style='height:20px;'></div>
44     <div>
45         <?php echo $output; ?>
46     </div>
47 </body>
</head>
</html>

```

Ahora se procede a explicar las partes más importantes del programa anterior. En la instrucción 'charset', se especifica la codificación de los caracteres en el documento. Dentro del 'body', se especifica la imagen que hará de fondo, que se repita tanto hori-

zontal como verticalmente para ocupar toda la pantalla y la fuente y el tamaño de la letra que aparecerá en la vista. Seguidamente, viene la etiqueta 'a', aquí se especificará el tamaño y el color de la fuente que contendrán los enlaces a los nuevos formularios. Las siguientes instrucciones dignas de mención son las que aparecen dentro del cuerpo. Estas instrucciones son: la inserción del logo de la Universidad, el título que aparecerá debajo del logo para informar al usuario con que formulario esta trabajando y los distintos enlaces a otros formularios. Para realizar enlaces a otros formularios, es necesario especificar el nombre de sus respectivas funciones.

4.3 Pantallas de la aplicación Web

En el presente apartado se mostrarán todas las pantallas que componen la aplicación web y, a su vez, se describirá la razón por la que cada una de ellas se ha diseñado.

4.3.1 Pantalla de selección de asignatura

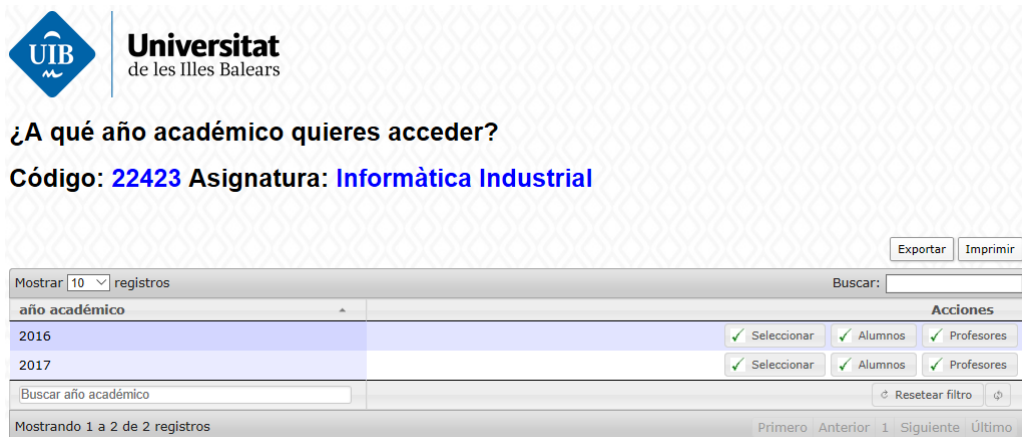
En esta pantalla aparecerán en forma de listado todas las asignaturas que estén dadas de alta en la base de datos, junto con el código de cada una de ellas. El objetivo de esta pantalla, es permitir al profesor que seleccione la asignatura con la que desea trabajar (ver figura 4.1).

Código asignatura	Nombre asignatura	Acciones
22427	Ampliació d'automatització industrial	✓ Seleccionar Ver Editar Eliminar
22439	Ampliació de robòtica	✓ Seleccionar Ver Editar Eliminar
22416	Automatització industrial	✓ Seleccionar Ver Editar Eliminar
22407	Components i sistemes electrònics	✓ Seleccionar Ver Editar Eliminar
22419	Control per computador	✓ Seleccionar Ver Editar Eliminar
20301	Càlcul	✓ Seleccionar Ver Editar Eliminar
22431	Diseny i Simulació Electrònica	✓ Seleccionar Ver Editar Eliminar

Figura 4.1

4.3.2 Pantalla de selección del año académico

Esta pantalla tiene como objetivo permitir al profesor seleccionar el año académico con el que desea trabajar. Los años académicos están compuestos por dos años, en el formulario aparece el primero de estos años; por ejemplo, cuando muestra 2017, hace referencia al año académico 2017-2018 (ver figura 4.2).



Universitat de les Illes Balears

¿A qué año académico quieres acceder?

Código: **22423** Asignatura: **Informàtica Industrial**

Exportar Imprimir

Mostrar 10 registros Buscar:

año académico	Acciones
2016	✓ Seleccionar ✓ Alumnos ✓ Profesores
2017	✓ Seleccionar ✓ Alumnos ✓ Profesores

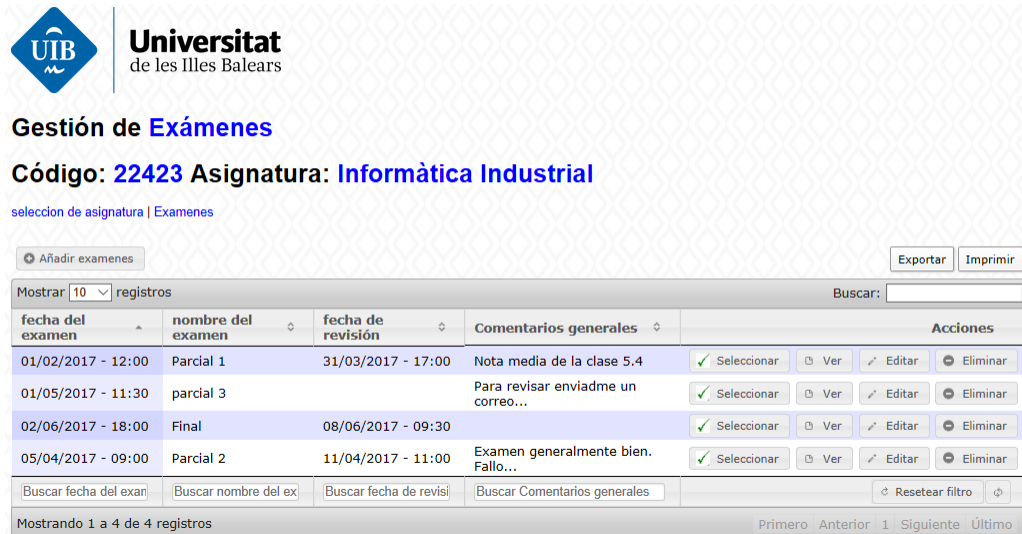
Buscar año académico Resetea filtro

Mostrando 1 a 2 de 2 registros Primero Anterior 1 Siguiente Último

Figura 4.2

4.3.3 Pantalla de selección de exámenes

En esta pantalla el profesor podrá dar de alta los exámenes para la asignatura seleccionada. Estos exámenes, se podrán ordenar por nombre, fecha de examen... y se podrán añadir comentarios generales comunes a todos los alumnos (ver figura 4.3).



Universitat de les Illes Balears

Gestión de **Exámenes**

Código: **22423** Asignatura: **Informàtica Industrial**

seleccion de asignatura | Exámenes

Añadir exámenes Exportar Imprimir

Mostrar 10 registros Buscar:

fecha del examen	nombre del examen	fecha de revisión	Comentarios generales	Acciones
01/02/2017 - 12:00	Parcial 1	31/03/2017 - 17:00	Nota media de la clase 5.4	✓ Seleccionar Ver Editar Eliminar
01/05/2017 - 11:30	parcial 3		Para revisar enviadme un correo...	✓ Seleccionar Ver Editar Eliminar
02/06/2017 - 18:00	Final	08/06/2017 - 09:30		✓ Seleccionar Ver Editar Eliminar
05/04/2017 - 09:00	Parcial 2	11/04/2017 - 11:00	Examen generalmente bien. Fallo...	✓ Seleccionar Ver Editar Eliminar

Buscar fecha del exan Buscar nombre del ex Buscar fecha de revisi Buscar Comentarios generales Resetea filtro

Mostrando 1 a 4 de 4 registros Primero Anterior 1 Siguiente Último

Figura 4.3

4.3.4 Pantalla de edición de modelos

En la presente pantalla, el profesor tendrá la capacidad de crear distintos modelos para un mismo examen. Los modelos se podrán nombrar tanto con letras: A,B,... como con números: 1,2... Los distintos modelos podrán contener preguntas distintas entre ellos e incluso se podrá hacer variar el orden de ellas (ver figura 4.4).

Universitat
de les Illes Balears

Código: 22423 Asignatura: Informàtica Industrial

Estas editando el examen Parcial 1 con fecha 2017-02-01 12:00:00

[Cambiar de examen](#) | [Modelos](#) | [Preguntas](#) | [Soluciones de cada alumno](#)

Mostrar 10 registros

Modelo	Preguntas de cada modelo	Acciones
A	b4a1d-sentencia-select-ii.pdf, 8b669-mer-red-social.pdf, 73523-mr-red-social.pdf	Ver Editar Eliminar
B	46055-sentencia-select.pdf, 8d1c4-sentencia-select-iii.pdf	Ver Editar Eliminar

Mostrando 1 a 2 de 2 registros

Figura 4.4

4.3.5 Pantalla de creación de preguntas

El objetivo de esta pantalla, es mostrar el listado de preguntas que estén insertadas en la base de datos y que pertenezcan a la asignatura previamente seleccionada y a su vez, permitir dar de alta nuevas preguntas. En la pantalla se mostrará el enunciado, la solución del profesor, los comentarios generales de la pregunta y en qué exámenes se ha insertado (ver figura 4.5). Una acción adicional para el profesor es la de poder visualizar el documento de cada pregunta unicamente pulsando sobre cada una de ellas.

Universitat
de les Illes Balears

Código: 22423 Asignatura: Informàtica Industrial

Estas editando el examen Parcial 1 con fecha 2017-02-01 12:00:00

[Cambiar de examen](#) | [Modelos](#) | [Preguntas](#) | [Soluciones de cada alumno](#)

Mostrar 50 registros

Enunciado	Solución del profesor	Comentarios generales	Modelos de cada pregunta	Acciones
8b669-mer-red-social.pdf	01d91-sol-mer-red-social.pdf	Hay más de una solución...	Informàtica Industrial Parcial...	Ver Editar Eliminar
46055-sentencia-select.pdf	07252-sol-sentencia-select.pdf		Informàtica Industrial Parcial...	Ver Editar Eliminar
55fc8-rr-.pdf	1412e-sol-rr-.pdf		Informàtica Industrial Parcial...	Ver Editar Eliminar
4c180-1st-.pdf	423ab-sol-1st-.pdf		Informàtica Industrial Final...	Ver Editar Eliminar

Figura 4.5

4.3.6 Pantalla de soluciones de cada alumno

Finalmente cuando se selecciona el examen sobre el que se quiere trabajar, se puede empezar a insertar las correcciones de alumnos en la base de datos. En el formulario actual se puede visualizar que alumno ha realizado una determinada pregunta, que

4.3. Pantallas de la aplicación Web

profesor la ha corregido y los comentarios individualizados. De esta forma se puede volver a visualizar el documento de cada pregunta pulsando sobre ella.

The screenshot shows the 'Examen Parcial 1' interface. At the top, it displays the 'Universitat de les Illes Balears' logo and the course code '22423 Asignatura: Informàtica Industrial'. Below this, it states 'Estas editando el examen Parcial 1 con fecha 2017-02-01 12:00:00'. There are navigation links for 'Cambiar de examen', 'Modelos', 'Preguntas', and 'Soluciones de cada alumno'. A table lists the questions and corrections:

Alumno	Modelo y pregunta	Profesor corrector	Solución del alumno	Corrección del profesor	Comentarios individuales	Acciones
Pascual González, Ramón	Modelo: A, enunciado:73523-mr-red-social.pdf, pregunta 1	Antich Tobaruela, Javier	3e054-ramon-pascual-pregunta-2.pdf	adb4b-sol-ramon-pascual-pregunta-2.pdf	Has cometido errores de concepto...	Ver, Editar, Eliminar
Pascual González, Ramón	Modelo: A, enunciado:8b669-mer-red-social.pdf, pregunta 0	Antich Tobaruela, Javier	a9a56-ramon-pascual-pregunta-1.pdf	79af9-sol-ramon-pascual-pregunta-1.pdf	Debes especificar el origen de...	Ver, Editar, Eliminar
Pascual González, Ramón	Modelo: A, enunciado:b4a1d-sentencia-select-ii.pdf, pregunta 2	Antich Tobaruela, Javier	f1289-ramon-pascual-pregunta-3.pdf	a950a-sol-ramon-pascual-pregunta-3.pdf	Quitado de algunos aspectos, perfecto...	Ver, Editar, Eliminar
Socias Uabres, Tomeu	Modelo: B, enunciado:46055-sentencia-select.pdf, pregunta 0	Antich Tobaruela, Javier	8fc9f-tomeu-socias-pregunta-1.pdf	610c0-sol-tomeu-socias-pregunta-1.pdf		Ver, Editar, Eliminar
Socias Uabres, Tomeu	Modelo: B, enunciado:8d1c4-sentencia-select-iii.pdf, pregunta 1	Antich Tobaruela, Javier	bf353-tomeu-socias-pregunta-2.pdf	81e41-sol-tomeu-socias-pregunta-2.pdf	Bien hecho	Ver, Editar, Eliminar

Figura 4.6

4.3.7 Pantalla para dar de alta a los alumnos

En la presente pantalla se muestra el listado de alumnos que están insertados en la base de datos, junto con su nombre de usuario, foto y las asignaturas en las que se ha ido matriculando (ver figura 4.7).

The screenshot shows the 'Dar de alta a los alumnos' interface. At the top, it displays the 'Universitat de les Illes Balears' logo and the heading 'SELECCIONA UNA DE LAS TRES OPCIONES'. Below this, there are navigation links for 'selección de asignatura', 'Dar de alta a alumnos', and 'Dar de alta a profesores'. A table lists the students:

Nombre de usuario	Nombre	Primer apellido	Segundo apellido	Foto	Asignaturas y años académicos	Acciones
ABC123	Andreu	Bauçà	Company		2010 22427 Ampliació d'automatització...	Ver, Editar, Eliminar
DAJ852	David	Arévalo	Jiménez		2010 22422 Xarxes de Comunicació...	Ver, Editar, Eliminar
DBM172	Daniel	Bujosa	Mateu		2010 22400 Matemàtiques per a...	Ver, Editar, Eliminar
DDS752	David	De Luis	Sanz		2011 20395 English for engineering...	Ver, Editar, Eliminar
DGO789	David	García	Ortiz		2010 22400 Matemàtiques per a...	Ver, Editar, Eliminar
DMG623	Daniel	Mañogil	González		2012 22427 Ampliació d'automatització...	Ver, Editar, Eliminar

Figura 4.7

4.3.8 Pantalla para dar de alta a los profesores

En esta pantalla sucede lo mismo que en la sección 'Pantalla para dar de alta a los alumnos' pero esta vez, referenciado a los profesores (ver figura 4.8).

4. DISEÑO DE LA APLICACIÓN WEB

The screenshot shows the header of the Universitat de les Illes Balears website. Below the header, there is a navigation menu with links for 'selección de asignatura', 'Dar de alta a alumnos', and 'Dar de alta a profesores'. A button labeled 'Añadir profesores' is visible. On the right side, there are 'Exportar' and 'Imprimir' buttons. Below these is a search bar labeled 'Buscar:'. The main content is a table with the following columns: 'Nombre de usuario', 'Nombre', 'Primer apellido', 'Segundo apellido', 'Foto', 'Email', 'Despacho', 'Asignaturas y años académicos', and 'Acciones'. The table contains four rows of data, each with a set of 'Ver', 'Editar', and 'Eliminar' buttons in the 'Acciones' column.

Nombre de usuario	Nombre	Primer apellido	Segundo apellido	Foto	Email	Despacho	Asignaturas y años académicos	Acciones
JAT951	Javier	Antich	Tobaruela		javier.antich@uib.es	AT001	2012 22410 Introducció a la Informàtica...	<input type="button" value="Ver"/> <input type="button" value="Editar"/> <input type="button" value="Eliminar"/>
ABV159	Alberto	Ballesteros	Varela				2014 22440 Laboratori d'automatització,...	<input type="button" value="Ver"/> <input type="button" value="Editar"/> <input type="button" value="Eliminar"/>
SBA369	Salvador	Barceló	Adrover				2010 22426 Sistemes Electrònics...	<input type="button" value="Ver"/> <input type="button" value="Editar"/> <input type="button" value="Eliminar"/>
MBG	Manuel Alejandro	Barranco	González			AT210	2010 22422 Xarxes de Comunicació...	<input type="button" value="Ver"/> <input type="button" value="Editar"/> <input type="button" value="Eliminar"/>

Figura 4.8

DISEÑO DE LA APLICACIÓN MÓVIL Y SERVIDOR

Enfocado al usuario, la aplicación móvil se ha realizado con una interfaz lo más sencilla e intuitiva posible.

Para poder comunicar la aplicación móvil con la base de datos se ha empleado un esquema de comunicación Cliente-Servidor. La aplicación Cliente (o móvil) envía peticiones a una aplicación Servidor. Esta última aplicación resuelve dichas peticiones ejecutando diferentes consultas sobre la base de datos.

5.1 Diseño de la aplicación Cliente

El diseño de la aplicación cliente se ha llevado a cabo a través del programa 'Android Studio'. Se han creado una sucesión de pantallas para poder verificar las credenciales, seleccionar la asignatura, los exámenes y las preguntas para, posteriormente, poder visualizarlos. El movimiento entre pantallas se ha realizado a través de botones que pueden hacer referencia tanto a las asignaturas, los exámenes o las preguntas.

Elementos de la pantalla

Se han usado objetos de varias clases para crear las pantallas de la aplicación móvil:

- 'Text View': Se trata de una cadena de texto corta que aporta información al usuario.
- 'Edit Text': Permite que el usuario escriba información o imprime por pantalla el resultado de una consulta a la base de datos.
- 'List View': Imprime por pantalla en forma de listado el resultado de una consulta a la base de datos.

- 'Button': Permite llevar a cabo la transición entre pantallas y la selección de diversas opciones.
- 'Multiline Text': Posibilita imprimir por pantalla largas cadenas de texto.

La aplicación cliente está constituida por un total de nueve pantallas. Para diseñar una pantalla en 'Android Studio' se deben implementar dos archivos, el 'Layout' y el 'Main Activity'.

Layout

El primer archivo que se debe implementar es el 'Layout'. Este archivo es el encargado de la interfaz del usuario, es decir, es lo que verá el usuario. En él podemos definir los objetos que queremos que compongan la pantalla, dónde se localizan y su tamaño, además de otros aspectos como son el fondo y el formato. Para crearlos existen dos modos de edición: el modo 'Design' o el modo 'Text'. En el modo 'Design' se arrastran los objetos a la pantalla y se cambia su tamaño estirándolos o acortándolos manualmente. En el modo 'Text' se insertan y editan los objetos mediante texto. Ambos mods son equivalentes en el siguiente sentido: si se crea el layout con el modo 'Design' existe un código equivalente en el modo 'Text' y viceversa. Un ejemplo de 'Layout' es el que se muestra en la figura 5.1.



Figura 5.1: pantalla de selección de una pregunta dado un examen

Main Activity

Una vez se ha implementado el 'Layout', se puede proceder a programar el archivo 'Main Activity' o actividad. Este archivo contiene la parte lógica de la pantalla, es decir, se especifican las acciones de cada objeto, las conexiones con servidores, el movimiento

entre pantallas... También es el lugar donde se crean las clases y se llaman a los distintos métodos. El código que permite declarar una nueva actividad es el que se muestra a continuación:

```

1 package com.example.ramon.myapplication;
2
3 import android.support.v7.app.AppCompatActivity;
4 import android.os.Bundle;
5
6 public class MainActivity extends AppCompatActivity {
7
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_main);
12     }
13 }

```

5.1.1 Pantallas creadas

Todas las pantallas tienen sus particularidades, así que, a continuación se explicará el porqué de cada pantalla y el código más importante vinculado con cada una de ellas.

Pantalla de acceso o login

Esta pantalla permite verificar el usuario y la contraseña del alumno. De esta forma, únicamente se podrá acceder a las siguientes pantallas si el alumno se encuentra registrado en la base de datos. El aspecto de esta pantalla se muestra en la figura 5.2 (a).

Cuando se inserta la contraseña, se debe encriptar según el algoritmo MD5 antes de enviarla al servidor. Esto se debe a que la contraseña se guarda cifrada en la base de datos. Para cifrar la contraseña, se ha usado el siguiente código:

```

1 contrasena=editText4.getText().toString();
2 contrasena=Hash.md5(contrasena);

```

Como se puede observar, como primer paso, se guarda la contraseña introducida por el usuario en una variable. Posteriormente, esta contraseña se encripta llamando al método 'md5' de la clase 'Hash'. El código vinculado con este método se expone a continuación:

```

1 public static class Hash {
2     private static final char[] CONSTS_HEX = { '0', '1', '2', '3', '4', '5', '6', '7', '8',
3         '9', 'a', 'b', 'c', 'd', 'e', 'f' };
4     public static String getHash(String txt) {
5         try {
6             java.security.MessageDigest md = java.security.MessageDigest.
7                 getInstance("MD5");
8             byte[] array = md.digest(txt.getBytes());
9             StringBuffer sb = new StringBuffer();
10            for (int i = 0; i < array.length; ++i) {
11                int bajo = (int)(array[i] & 0x0f);
12                int alto = (int)((array[i] & 0xf0) >> 4);
13                sb.append(CONSTS_HEX[alto]);
14                sb.append(CONSTS_HEX[bajo]);
15            }
16        } catch (Exception e) {
17            e.printStackTrace();
18        }
19    }
20 }

```

```
13         }
14         return sb.toString();
15     } catch (java.security.NoSuchAlgorithmException e) {
16         System.out.println(e.getMessage());
17     }
18     return null;
19 }
20 public static String md5(String txt) {
21     return Hash.getHash(txt);
22 }
23 }
```

Pantalla de selección de asignatura

Esta pantalla es la encargada de mostrar en forma de listado las asignaturas en las que el alumno se encuentra matriculado en el año académico actual. Para seleccionar una asignatura únicamente es necesario pulsar sobre ella. El servidor envía la respuesta como una cadena de texto separando los objetos por comas. De esta forma, para conseguir que la información que recibe del servidor esté en forma de listado se ejecuta el siguiente código:

```
1 ArrayList<String> ejemplo = new ArrayList<String>();
2 int Index, fromIndex = 0;
3 while (true) {
4     Index = value.indexOf(',', fromIndex);
5     if (Index == -1) {
6         ejemplo.add(value.substring(fromIndex));
7         break;
8     }
9     ejemplo.add(value.substring(fromIndex, Index));
10    fromIndex = Index + 1;
11 }
```

En este código se desglosan los valores de la cadena 'value' que estén separados por el caracter ','. Se puede apreciar cómo se ha tenido que contemplar el hecho de que el último valor de la cadena no termina con el caracter ','.

El aspecto de esta pantalla se muestra en la figura 5.2 (b).

Pantalla de selección del año del examen

En la actual pantalla, el usuario puede decantarse entre seleccionar un examen de un año académico anterior o del año académico actual. En lo referente al código para implementar la pantalla, no hay ningún fragmento de código digno de mención. El aspecto de esta pantalla se muestra la figura 5.2(c).

Pantalla de selección de un examen actual

En esta pantalla se muestran los exámenes de la asignatura anteriormente seleccionada y que pertenecen al curso actual junto con el profesor/es que imparten las clases. En esta pantalla no vuelve a haber ningún un código digno de mención, puesto que, el código referente a la separación de la respuesta del servidor en forma de lista ya se ha

explicado en la sección: 'Pantalla de selección de asignatura'. El aspecto de esta pantalla se muestra en la figura 5.2 (d).

Pantalla de selección de un examen de años anteriores

El objetivo de esta pantalla es el mismo que el en caso anterior, a excepción de que ahora aparecen los exámenes realizados durante los cursos anteriores. El aspecto de esta pantalla se muestra en la figura 5.2 (e).

Pantalla de selección de una pregunta dado un examen actual

La intención de esta pantalla es mostrar el listado de preguntas que componen el examen seleccionado junto con los comentarios del profesor y la fecha de realización y revisión. De esta forma, se dota de la máxima información al alumno. En esta pantalla vuelve a aparecer un fragmento de código que debe ser explicado. Este código permite quitar las etiquetas HTML de los comentarios. Las etiquetas HTML se insertan en el texto en el momento que la aplicación web guarda el comentario en la base de datos. El código es el siguiente:

```
1 private String REM_HTMLTags(String Data)
2 {
3     Data = Data.replaceAll("<p>", "");
4     Data = Data.replaceAll("</p>", "");
5     return Data;
6 }
```

EL código anterior solo elimina las etiquetas referentes a un párrafo. En caso de querer eliminar otras etiquetas, únicamente sería necesario añadirlas en el método REM_HTMLTags. El aspecto de esta pantalla se muestra en la figura 5.2 (f).

Pantalla de selección de una pregunta dado un examen pasado

Vuelve a suceder lo mismo que en el caso previo, pero con la excepción de que esta vez no se muestra la fecha de revisión, puesto que al no tratarse de un examen del año académico actual, la fecha de revisión no es un dato de interés para el alumno. El aspecto de esta pantalla se muestra en la figura 5.2 (g).

Pantalla de descarga de una pregunta dado un examen actual

El propósito de esta pantalla es conseguir descargar el examen resuelto por parte del alumno, la solución dada por el profesor, el enunciado y la corrección sobre la solución del alumno. Además, en esta pantalla se informa del profesor que ha corregido cada pregunta, y sus comentarios individuales y generales. Para abrir el documento en el navegador para, posteriormente, descargarlo es necesario ejecutar el siguiente código:

```
1 boton1.setOnClickListener(
2     new View.OnClickListener() {
3         public void onClick(View view) {
4             //La dirección IP de la máquina donde se encuentra la aplicación
5             //servidoe, la aplicación web y la base de datos es 192.168.112.2
6             Intent browserIntent = new Intent(Intent.ACTION_VIEW, Uri.parse("http
7             ://192.168.112.2/app_examen_uib/assets/uploads/files/"+NOMBRE_DEL_DOCUMENTO));
```

```
6         startActivity (browserIntent) ;  
7     }  
8 }  
9 );
```

En el código se aprecia cómo se crea la intención con la dirección en la que se encuentra el archivo para, posteriormente, ejecutar una nueva 'activity' con esta intención. Todo esto se encuentra dentro del método de pulsación del botón, es decir, sólo se ejecuta cuando se pulsa el botón 'ABRIR'.

En el momento en el que en la aplicación se ejecuta y se pulsa el botón abrir, se abre el navegador y el archivo se descarga. Posteriormente, se debe abrir el documento con un lector de PDF.

El aspecto de esta pantalla se muestra en figura 5.2 (h).

Pantalla de descarga de una pregunta dado un examen pasado

El proceso y el objetivo es el mismo que en su pantalla homóloga de la figura 5.2(h), a excepción de que en esta pantalla sólo aparece el enunciado y la solución del profesor. Esto es porque muchos alumnos prefieren prepararse los exámenes realizándolos por ellos mismos sin soluciones, y una vez consideran que han finalizado los corrigen. El aspecto de esta pantalla se muestra en la figura 5.2 (i).

Explicación de otros aspectos relevantes

Una vez se ha explicado la razón de cada pantalla, es necesario explicar cómo se consigue pasar de una pantalla a otra y cómo se comparten las variables. Compartir variables es un aspecto muy importante de la aplicación, puesto que nos permite conocer las selecciones hechas hasta el momento por el usuario.

La implementación del cambio de pantalla es simple, ya que únicamente es necesario ejecutar el siguiente código:

```
1 Intent intent = new Intent(PANTALLA ORIGEN, PANTALLA DESTINO);  
2 startActivity(intent)
```

Como se aprecia en el código, se debe crear una intención donde se especifiquen las pantallas origen y destino. Posteriormente, sólo se tiene que ejecutar esta intención con el método 'startActivity'. En el aspecto referente a la compartición de variables entre pantallas, únicamente es necesario introducir los siguientes códigos en la pantalla origen y destino, respectivamente:

```
1 Intent intent = new Intent(Pantalla origen, Pantalla destino);  
2 intent.putExtra("ETIQUETA", STRING);
```

```
1 final String VARIABLE = getIntent().getExtras().getString("ETIQUETA");
```

Para compartir la variable, la etiqueta debe ser común en ambos casos. Esto es así porque entre pantallas se pueden llegar a compartir varias variables.

5.1. Diseño de la aplicación Cliente

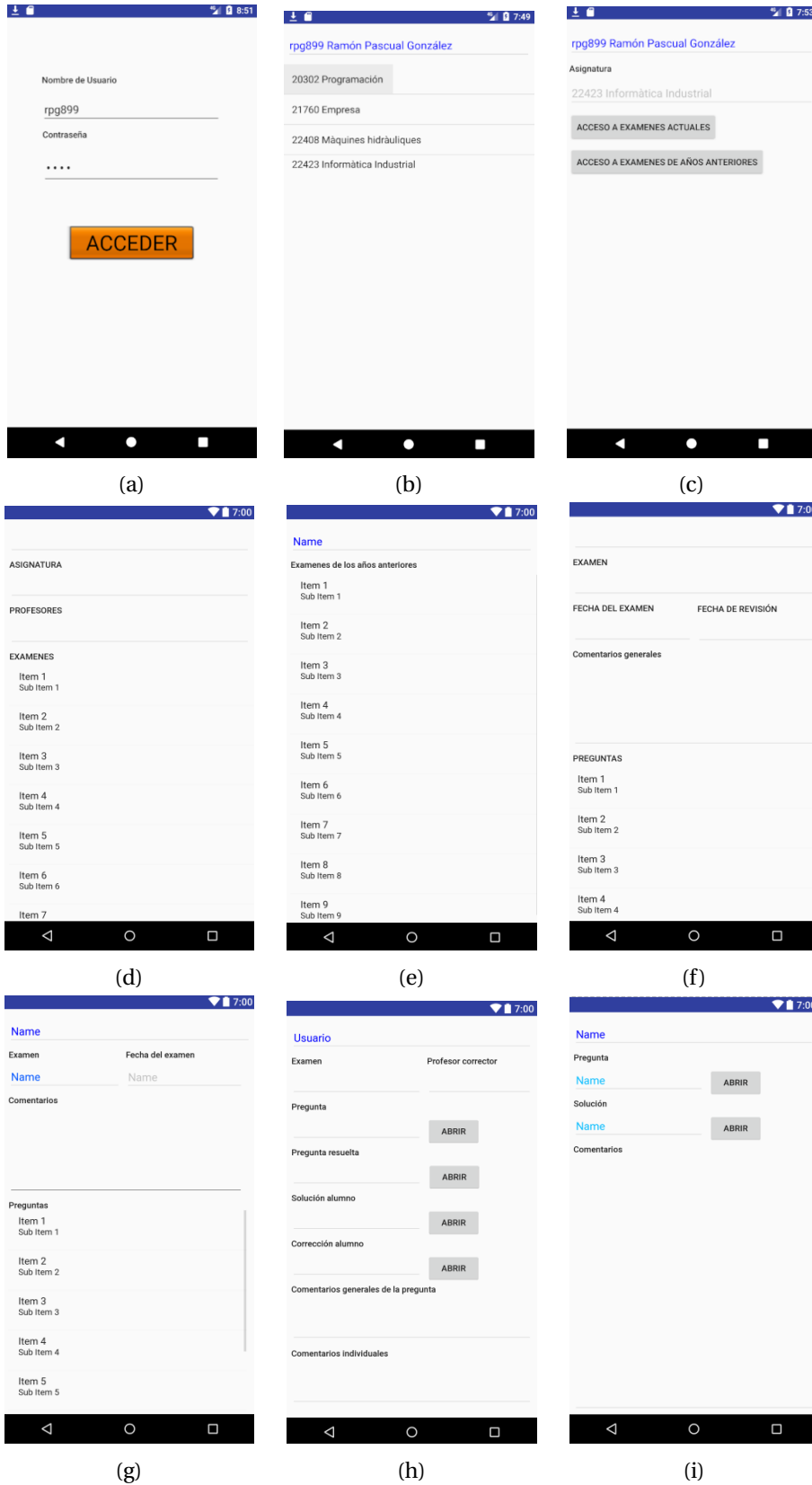


Figura 5.2: Pantallas de la aplicación móvil

5.2 Diseño de la aplicación Servidor

El diseño de la aplicación servidor se ha llevado a cabo a través del programa "Net-Beans". No tiene más objetivo que hacer de puente entre el usuario de la aplicación móvil y la base de datos. El cliente, es decir, la aplicación móvil enviará peticiones a la aplicación servidor en función de la pantalla en la que se encuentre el usuario. Cuando el servidor reciba esta información, la procesará y enviará una consulta a la base de datos en función de la petición realizada por el cliente. Una vez haya enviado la consulta, se mantendrá a la espera de la respuesta de la base de datos para una vez llegue remitirla al cliente.

El cliente puede enviar hasta nueve peticiones distintas al servidor. Como se ha descrito anteriormente, una petición distinta para cada pantalla. Para distinguir las peticiones, se inserta una etiqueta al inicio de la petición. Los nombres de las nueve etiquetas son los siguientes: MAC10, MAC20, MAC22, MAC23, MAC24, MAC25, MAC27 y MAC28.

Como se aprecia, el nombre de cada etiqueta está constituido por MAC ('Main Activity') más el número de la actividad que la envía. Después de la etiqueta, se añade la información necesaria para poder realizar la consulta con las condiciones adecuadas. Por ejemplo, en el caso de la petición con etiqueta MAC22, se envía el nombre del usuario y el código de la asignatura seleccionada, para poder así recibir los nombres de los profesores, los exámenes y las fechas de realización. El envío de una petición se lleva a cabo de acuerdo con el siguiente fragmento de código:

```
1 MyATaskCliente myATaskYW = new MyATaskCliente ();  
2 myATaskYW.execute("String de datos");
```

A la instrucción 'execute' se le pasa como parámetro un String de datos. Esta cadena de datos es la petición que se ha descrito anteriormente. Una vez se conoce cómo envía la petición el cliente, es necesario conocer cómo trata esta información el servidor. Primero se muestra el código donde el servidor lee la petición.

```
1 clientSocket = serverSocket.accept();  
2 BufferedReader input = new BufferedReader(new InputStreamReader(clientSocket.  
   getInputStream()));  
3 PrintStream output = new PrintStream(clientSocket.getOutputStream());  
4 String request = input.readLine();
```

La primera instrucción mantiene el Servidor a la espera de recibir una conexión. Cuando la recibe, la acepta y sigue ejecutando. Las siguientes dos instrucciones son la creación de dos objetos que, posteriormente, nos permitirán recibir la petición del cliente mediante el objeto 'input' y enviar al cliente la respuesta a través del objeto 'output'. Finalmente, la última instrucción lee y guarda la petición del cliente en la variable 'request' con el uso del objeto 'input' explicado anteriormente.

El próximo paso es conocer cómo se procesa esta información. Primero se pasa el contenido de la variable 'request' como parámetro al método constructor de la clase 'conexionSQL':

```
1 ArrayList str_BD = conexionSQL(request);
```

En esta clase se realiza el envío de la consulta a la base de datos. Esta consulta será diferente en función del contenido de la variable 'request'. De esta forma, el servidor puede responder a las distintas peticiones del cliente. Para entender correctamente el funcionamiento de la clase 'ConexionSQL' se explicará, seguidamente, de forma detallada y estructurada su contenido.

clase conexionSQL

El primer aspecto digno de mención es el SubString de cinco caracteres de la variable 'request'. Se extraen estos cinco caracteres porque serán estos los que nos indiquen que petición ha realizado el cliente.

```
1 String orden = request.substring(0,5);
```

Posteriormente, el servidor debe indicar a qué base de datos quiere acceder. Para ello, ejecuta las siguientes dos líneas de código:

```
1 Class.forName("com.mysql.jdbc.Driver");
2 Connection co = DriverManager.getConnection("jdbc:mysql://localhost/app_database"
    ,"ramon_pg", "password521");
```

En este código se aprecia como el servidor ejecuta instrucciones para acceder a una base de datos de tipo MaríaDB con nombre 'app_database'. Se accede con el usuario 'ramon_pg' y con la contraseña 'password521'.

Una vez se ha seleccionado la base de datos, se procede a mirar qué petición ha hecho el cliente. Para ello, se ha usado la instrucción 'switch'. Cada caso del 'switch' realizará una petición distinta a la base de datos. A continuación, se muestra en pseudocódigo la estructura general de este switch:

```
1 switch(orden) {
2     caso MAC10:
3         Separar contenido de la petición en un string de strings
4         Realizar consulta
5         Guardar el resultado de la consulta en una variable
6     caso MAC20:
7         Separar contenido de la petición en un string de strings
8         Realizar consulta
9         Guardar el resultado de la consulta en una variable
10    caso MAC22:
11        Separar contenido de la petición en un string de strings
12        Realizar consulta
13        Guardar el resultado de la consulta en una variable
14        Realizar dos consultas más y guardarlas en la misma variable separando
15        los resultados de las consultas con el caracter '#'
16    caso MAC23:
17        Separar contenido de la petición en un string de strings
18        Realizar consulta
19        Guardar el resultado de la consulta en una variable
20        Realizar dos consultas más y guardarlas en la misma variable separando
21        los resultados de las consultas con el caracter '#'
22    caso MAC24:
23        Separar contenido de la petición en un string de strings
24        Realizar consulta
25        Guardar el resultado de la consulta en una variable
```

5. DISEÑO DE LA APLICACIÓN MÓVIL Y SERVIDOR

```
24     Realizar cinco consultas más y guardarlas en la misma variable separando
    los resultados de las consultas con el caracter '#'
25     caso MAC25:
26         Separar contenido de la petición en un string de strings
27         Realizar consulta
28         Guardar el resultado de la consulta en una variable
29         Realizar una consulta más y guardarla en la misma variable separando los
    resultados de las consultas con el caracter '#'
30     caso MAC27:
31         Separar contenido de la petición en un string de strings
32         Realizar consulta
33         Guardar el resultado de la consulta en una variable
34         Realizar dos consultas más y guardarlas en la misma variable separando
    los resultados de las consultas con el caracter '#'
35     caso MAC28:
36         Separar contenido de la petición en un string de strings
37         Realizar consulta
38         Guardar el resultado de la consulta en una variable
39         Realizar una consulta más y guardarla en la misma variable separando los
    resultados de las consultas con el caracter '#'
40 }
```

La primera acción que se realiza dentro de cada caso es la separación del contenido de la petición en un string de strings. Esto se hace porque, cómo se ha descrito anteriormente, el cliente envía diferentes datos para que la consulta del servidor tenga los filtros necesarios. Una vez se tienen separados los diferentes datos, es momento de realizar la consulta. En esta consulta se especifica la tabla de la cual se extraerá el dato deseado junto con sus filtros. La respuesta de la base de datos se guarda en una variable que es un string de strings. De esta forma, cada dato devuelto por la consulta se guardará en una posición distinta. En los casos en los que se necesiten más de un dato o datos de otras tablas, se realizarán más consultas. Los resultados de las diferentes consultas se separan utilizando el caracter '#'.

Como último paso, el Servidor envía el resultado al Cliente. Para hacer esto, ejecuta el siguiente fragmento de código:

```
1     output.flush(); //vacía contenido
2     output.println(str_BD);
3     clientSocket.close();
4
```

En el momento que ha finalizado el envío Servidor-Cliente, se procede a limpiar el contenido y a cerrar la conexión.

Ya se ha explicado como el Cliente envía la petición, el servidor la procesa y envía la respuesta. Ahora iniciamos la explicación de como el cliente trata esta información.

```
1     protected void onPostExecute(String value) {
2         progressDialog.dismiss();
3     }
4
```

En este código, la variable 'value' del método `onPostExecute()` hace referencia a la respuesta del Servidor. Por tanto, si queremos tratar esta respuesta, debemos introducir código dentro del mencionado método.

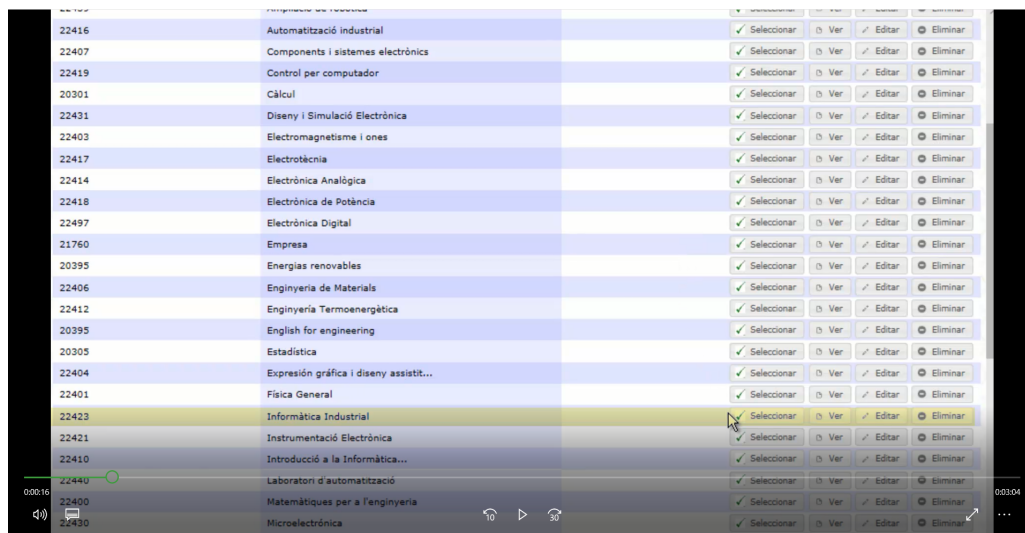
El código completo del Cliente y del Servidor se encuentra en los anexos 8.3: Cliente y 8.4: Servidor.

PRUEBAS DE VERIFICACIÓN

Para demostrar el correcto funcionamiento de la aplicación se adjunta un 'link' donde se prueba la aplicación web y móvil.

<https://www.youtube.com/watch?v=cohp260mDew>

El vídeo, está dividido en dos partes. En la primera parte del vídeo se muestra la aplicación web. Lo primero que aparece es el listado de las asignaturas. Se ha seleccionado la asignatura 'Informática industrial' como se muestra en la figura 6.1, junto con el año académico 2016-2017.



22416	Automatització industrial	✓ Seleccionar	Ver	✓ Editar	Eliminar
22407	Components i sistemes electrònics	✓ Seleccionar	Ver	✓ Editar	Eliminar
22419	Control per computador	✓ Seleccionar	Ver	✓ Editar	Eliminar
20301	Càlcul	✓ Seleccionar	Ver	✓ Editar	Eliminar
22431	Diseny i Simulació Electrònica	✓ Seleccionar	Ver	✓ Editar	Eliminar
22403	Electromagnetisme i ones	✓ Seleccionar	Ver	✓ Editar	Eliminar
22417	Electrotècnia	✓ Seleccionar	Ver	✓ Editar	Eliminar
22414	Electrònica Analògica	✓ Seleccionar	Ver	✓ Editar	Eliminar
22418	Electrònica de Potència	✓ Seleccionar	Ver	✓ Editar	Eliminar
22497	Electrònica Digital	✓ Seleccionar	Ver	✓ Editar	Eliminar
21760	Empresa	✓ Seleccionar	Ver	✓ Editar	Eliminar
20395	Energias renovables	✓ Seleccionar	Ver	✓ Editar	Eliminar
22406	Enginyeria de Materials	✓ Seleccionar	Ver	✓ Editar	Eliminar
22412	Enginyeria Termoenergètica	✓ Seleccionar	Ver	✓ Editar	Eliminar
20395	English for engineering	✓ Seleccionar	Ver	✓ Editar	Eliminar
20305	Estadística	✓ Seleccionar	Ver	✓ Editar	Eliminar
22404	Expresión gràfica i diseny assistit...	✓ Seleccionar	Ver	✓ Editar	Eliminar
22401	Física General	✓ Seleccionar	Ver	✓ Editar	Eliminar
22423	Informàtica Industrial	✓ Seleccionar	Ver	✓ Editar	Eliminar
22421	Instrumentació Electrònica	✓ Seleccionar	Ver	✓ Editar	Eliminar
22410	Introducció a la Informàtica...	✓ Seleccionar	Ver	✓ Editar	Eliminar
22440	Laboratori d'automatització	✓ Seleccionar	Ver	✓ Editar	Eliminar
22400	Matemàtiques per a l'enginyeria	✓ Seleccionar	Ver	✓ Editar	Eliminar
22430	Microelectrònica	✓ Seleccionar	Ver	✓ Editar	Eliminar

Figura 6.1: Imagen extraída del vídeo demostrativo

Una vez seleccionada la asignatura y el año académico, aparece el listado de exámenes. En esta parte del vídeo se va resaltando de forma temporal cada atributo en

6. PRUEBAS DE VERIFICACIÓN

un color distinto. De esta forma, se consigue que el observador del vídeo se fije en los aspectos más importantes de este formulario. Entre todos los exámenes, se selecciona el 'Parcial 1'. Cuando se cambia de formulario, se oscurece una parte del formulario para que el observador el vídeo, se percate que en la parte superior del formulario, aparecen los detalles del examen que se ha seleccionado (ver figura 6.2).

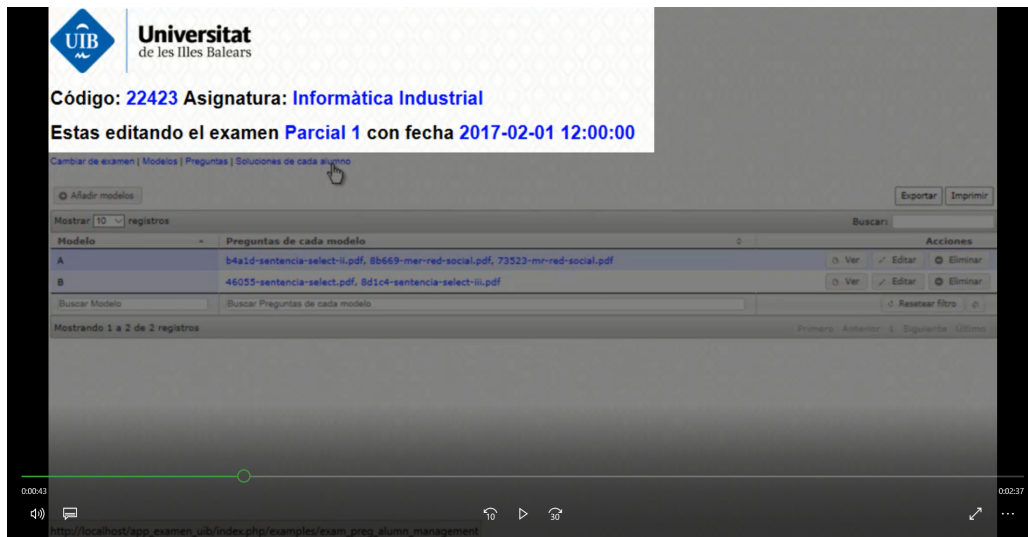


Figura 6.2: Imagen de la visualización de la cabecera

Posteriormente, se selecciona el formulario de la solución de cada alumno. En este formulario se muestran dos alumnos que han realizado dos modelos distintos de un mismo examen (ver figura 6.3). Las siguientes acciones que aparecen, son la de abrir los distintos archivos que conforman la solución del alumno y la corrección del profesor.



Figura 6.3: Imagen de las soluciones de los alumnos

En la segunda parte del vídeo se visualiza la aplicación móvil. Se accede a la aplica-

ción con el usuario 'RPG899'. En la pantalla siguiente se aprecia como este alumno se encuentra actualmente matriculado en cuatro asignaturas (ver figura 6.4). Otra vez, se volverá a seleccionar la asignatura 'Informática Industrial' y en la siguiente pantalla se decantará por los exámenes del año académico actual.

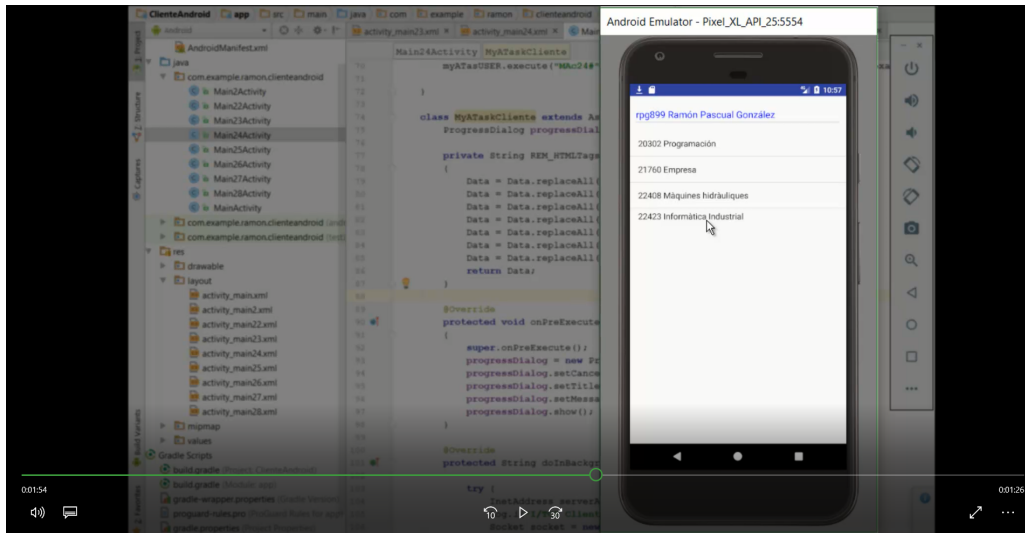


Figura 6.4: Imagen que muestra las asignaturas matriculadas

Se puede apreciar como los exámenes que ha realizado el alumno, coinciden con los que aparecen en la aplicación web. Nos hemos decantado por seleccionar el examen 'Parcial 1', puesto que éste ha sido el que se ha seleccionado desde la aplicación web y de esta forma se pueden comparar los datos (ver figura 6.5).

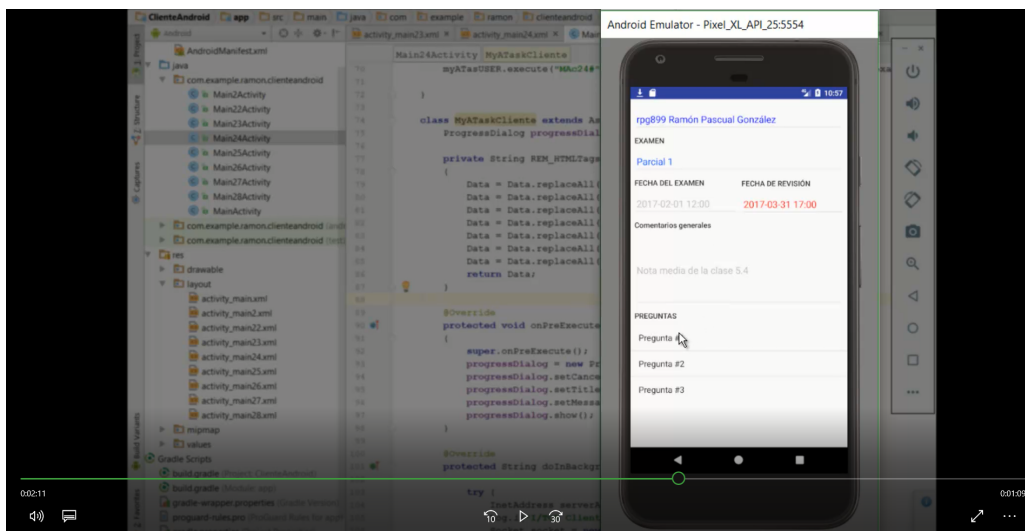


Figura 6.5: Imagen que muestra las características de un examen

En las siguientes acciones, se abrirán los documentos y se verificará que sean iguales a los que aparecían desde la aplicación web. Un ejemplo de ello es el que se

6. PRUEBAS DE VERIFICACIÓN

muestra en la figura 6.6, donde se comparan los resultados de la pregunta 1. Se puede observar como el documento se ha abierto de forma correcta, pues en ambos casos se observa el mismo archivo.

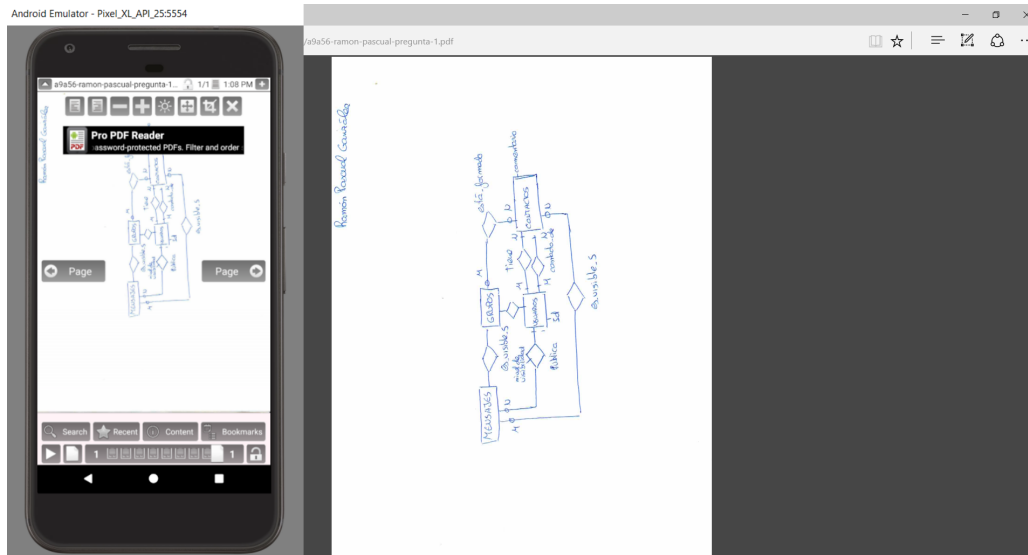


Figura 6.6: Solución a una pregunta

CONCLUSIONES

Habiendo ya finalizado **TFG**, se puede atestiguar que se han cumplido todos los objetivos propuestos por parte del tutor. A continuación, se procede a explicar los aspectos más relevantes tales como: un resumen del trabajo realizado, la dificultad y problemas encontrados por parte del autor, los conocimientos obtenidos y las posibilidades futuras de mejora del trabajo.

Finalmente se han realizado tres aplicaciones independientes con tres programas distintos que se interrelacionan entre ellos para obtener el funcionamiento deseado. A todo esto se le añade el hecho que se ha implementado y creado una base de datos.

La dificultad más relevante para el desarrollo de este **TFG** ha sido poder realizar la conexión Cliente-Servidor a través de Sockets. Esto se atribuye al hecho de que no se había realizado algo similar con anterioridad. Una vez resuelta la conexión, el resto del programa ha resultado ser de una complejidad razonable, a excepción de ciertos problemas puntuales, como por ejemplo es el problema de tratar la información que enviaba la aplicación servidor a la aplicación móvil, puesto que se tenía que transformar en un formato de lista. Otro problema que ha requerido mucho tiempo ha sido el de encontrar las limitaciones del Grocery CRUD y como consecuencia rediseñar el **MER** y **MR**.

En lo referente a los conocimientos adquiridos por parte del autor, se ha conseguido un manejo más fluido en los lenguajes JAVA y SQL, iniciarse y profundizar en los lenguajes PHP y HTML, una mayor independencia y solvencia a la hora de buscar y extraer información, junto con la capacidad de saber usar los conocimientos propios.

Como se ha escrito en el apartado 'Resumen', el objetivo final para la aplicación móvil es que pase a formar parte de la ya existente 'Agenda UIB'. A nuestra aplicación se le pueden añadir funcionalidades adicionales como es un chat entre profesor y alumno, añadir notificaciones emergentes con información sobre las últimas novedades, incluir información más detallada del profesorado como es el horario de tutorías, etc.

ANEXOS

8.1 Base de datos

```
1  -- phpMyAdmin SQL Dump
2  -- version 4.5.1
3  -- http://www.phpmyadmin.net
4  --
5  -- Servidor: 127.0.0.1
6  -- Tiempo de generación: 13-08-2017 a las 17:14:52
7  -- Versión del servidor: 10.1.8-MariaDB
8  -- Versión de PHP: 5.6.14
9
10 SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
11 SET time_zone = "+00:00";
12
13
14 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
15 /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
16 /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
17 /*!40101 SET NAMES utf8mb4 */;
18
19 --
20 -- Base de datos: 'test_database'
21 --
22 CREATE DATABASE IF NOT EXISTS 'test_database' DEFAULT CHARACTER SET latin1
23     COLLATE latin1_swedish_ci;
24 USE 'test_database';
25
26 -----
27 --
28 -- Estructura de tabla para la tabla 'alumnos'
29 --
30
31 DROP TABLE IF EXISTS 'alumnos';
```

8. ANEXOS

```
32 CREATE TABLE 'alumnos' (  
33 'gid_alumno' bigint(20) UNSIGNED NOT NULL,  
34 'nom_usuario_alumn' varchar(6) NOT NULL,  
35 'nombre' varchar(30) NOT NULL,  
36 'apellido1' varchar(30) NOT NULL,  
37 'apellido2' varchar(30) DEFAULT NULL,  
38 'foto' blob,  
39 'password' varchar(32) NOT NULL  
40 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
41  
42 ---  
43 --- RELACIONES PARA LA TABLA 'alumnos':  
44 ---  
45  
46 -----  
47  
48 ---  
49 --- Estructura de tabla para la tabla 'anos_academicos'  
50 ---  
51  
52 DROP TABLE IF EXISTS 'anos_academicos';  
53 CREATE TABLE 'anos_academicos' (  
54 'gid_ano_academico' bigint(20) UNSIGNED NOT NULL,  
55 'ano_academico' year(4) NOT NULL  
56 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
57  
58 ---  
59 --- RELACIONES PARA LA TABLA 'anos_academicos':  
60 ---  
61  
62 -----  
63  
64 ---  
65 --- Estructura de tabla para la tabla 'asignaturas'  
66 ---  
67  
68 DROP TABLE IF EXISTS 'asignaturas';  
69 CREATE TABLE 'asignaturas' (  
70 'gid_asignatura' bigint(20) UNSIGNED NOT NULL,  
71 'cod_asignatura' mediumint(5) UNSIGNED NOT NULL,  
72 'nom_asignatura' varchar(50) NOT NULL  
73 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
74  
75 ---  
76 --- RELACIONES PARA LA TABLA 'asignaturas':  
77 ---  
78  
79 -----  
80  
81 ---  
82 --- Estructura de tabla para la tabla 'asignaturas_anos_academicos'  
83 ---  
84  
85 DROP TABLE IF EXISTS 'asignaturas_anos_academicos';  
86 CREATE TABLE 'asignaturas_anos_academicos' (  
87 'gid_asign_ano' bigint(20) UNSIGNED NOT NULL,  
88 'gid_asignatura' bigint(20) UNSIGNED NOT NULL,
```

```

89 'gid_ano_academico' bigint(20) UNSIGNED NOT NULL
90 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
91
92 ---
93 --- RELACIONES PARA LA TABLA 'asignaturas_anos_academicos':
94 --- 'gid_ano_academico'
95 --- 'anos_academicos' -> 'gid_ano_academico'
96 --- 'gid_asignatura'
97 --- 'asignaturas' -> 'gid_asignatura'
98 ---
99
100 -----
101
102 ---
103 --- Estructura de tabla para la tabla 'cursa'
104 ---
105
106 DROP TABLE IF EXISTS 'cursa';
107 CREATE TABLE 'cursa' (
108 'gid_cursa' bigint(20) UNSIGNED NOT NULL,
109 'gid_alumno' bigint(20) UNSIGNED NOT NULL,
110 'gid_asign_ano' bigint(20) UNSIGNED NOT NULL
111 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
112
113 ---
114 --- RELACIONES PARA LA TABLA 'cursa':
115 ---
116
117 -----
118
119 ---
120 --- Estructura de tabla para la tabla 'examenes'
121 ---
122 ---
123
124 DROP TABLE IF EXISTS 'examenes';
125 CREATE TABLE 'examenes' (
126 'gid_examenes' bigint(20) UNSIGNED NOT NULL,
127 'fecha_examen' datetime NOT NULL,
128 'nom_examen' varchar(200) NOT NULL,
129 'gid_asignatura' bigint(20) UNSIGNED NOT NULL,
130 'fecha_revision' datetime DEFAULT NULL,
131 'coment_generales' text
132 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
133
134 ---
135 --- RELACIONES PARA LA TABLA 'examenes':
136 --- 'gid_asignatura'
137 --- 'asignaturas' -> 'gid_asignatura'
138 ---
139
140 -----
141
142 ---
143 --- Estructura de tabla para la tabla 'imparte'
144 ---
145

```

8. ANEXOS

```
146 DROP TABLE IF EXISTS 'imparte';
147 CREATE TABLE 'imparte' (
148   'gid_imparte' bigint(20) UNSIGNED NOT NULL,
149   'gid_profesor' bigint(20) UNSIGNED NOT NULL,
150   'gid_asign_ano' bigint(20) UNSIGNED NOT NULL
151 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
152
153 ---
154 --- RELACIONES PARA LA TABLA 'imparte':
155 ---   'gid_asign_ano'
156 ---     'asignaturas_anos_academicos' -> 'gid_asign_ano'
157 ---   'gid_profesor'
158 ---     'profesores' -> 'gid_profesor'
159 ---
160
161 --- -----
162
163 ---
164 --- Estructura de tabla para la tabla 'modelos'
165 ---
166
167 DROP TABLE IF EXISTS 'modelos';
168 CREATE TABLE 'modelos' (
169   'gid_modelos' bigint(20) UNSIGNED NOT NULL,
170   'modelo' varchar(1) NOT NULL,
171   'gid_exámenes' bigint(20) UNSIGNED NOT NULL
172 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
173
174 ---
175 --- RELACIONES PARA LA TABLA 'modelos':
176 ---   'gid_exámenes'
177 ---     'exámenes' -> 'gid_exámenes'
178 ---
179
180
181 --- -----
182
183 ---
184 --- Estructura de tabla para la tabla 'modelos_preguntas'
185 ---
186
187 DROP TABLE IF EXISTS 'modelos_preguntas';
188 CREATE TABLE 'modelos_preguntas' (
189   'gid_md1_prg' bigint(20) UNSIGNED NOT NULL,
190   'gid_modelos' bigint(20) UNSIGNED NOT NULL,
191   'cod_pregunta' bigint(20) UNSIGNED NOT NULL,
192   'num_pregunta' tinyint(3) UNSIGNED DEFAULT NULL
193 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
194
195 ---
196 --- RELACIONES PARA LA TABLA 'modelos_preguntas':
197 ---   'gid_modelos'
198 ---     'modelos' -> 'gid_modelos'
199 ---   'cod_pregunta'
200 ---     'preguntas' -> 'cod_pregunta'
201 ---
202
```



```

203
204
205
206 ---
207 --- Estructura de tabla para la tabla 'modelos_preguntas_alumnos'
208 ---
209
210 DROP TABLE IF EXISTS 'modelos_preguntas_alumnos';
211 CREATE TABLE 'modelos_preguntas_alumnos' (
212 'gid_mdl_prg_almn' bigint(20) UNSIGNED NOT NULL,
213 'gid_mdl_prg' bigint(20) UNSIGNED NOT NULL,
214 'gid_alumno' bigint(20) UNSIGNED NOT NULL,
215 'gid_profesor' bigint(20) UNSIGNED NOT NULL,
216 'sol_alumno' blob,
217 'correc_profesor' blob,
218 'coment_individuales' text
219 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
220
221 ---
222 --- RELACIONES PARA LA TABLA 'modelos_preguntas_alumnos':
223 --- 'gid_alumno'
224 --- 'alumnos' -> 'gid_alumno'
225 --- 'gid_mdl_prg'
226 --- 'modelos_preguntas' -> 'gid_mdl_prg'
227 --- 'gid_profesor'
228 --- 'profesores' -> 'gid_profesor'
229 ---
230
231 ---
232
233 ---
234 --- Estructura de tabla para la tabla 'preguntas'
235 ---
236
237 DROP TABLE IF EXISTS 'preguntas';
238 CREATE TABLE 'preguntas' (
239 'cod_pregunta' bigint(20) UNSIGNED NOT NULL,
240 'gid_asignatura' bigint(20) UNSIGNED NOT NULL,
241 'enunciado' blob NOT NULL,
242 'sol_profesor' blob,
243 'coment_generales' text
244 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
245
246 ---
247 --- RELACIONES PARA LA TABLA 'preguntas':
248 --- 'gid_asignatura'
249 --- 'exámenes' -> 'gid_asignatura'
250 ---
251
252 ---
253
254 ---
255 --- Estructura de tabla para la tabla 'profesores'
256 ---
257
258 DROP TABLE IF EXISTS 'profesores';
259 CREATE TABLE 'profesores' (

```

8. ANEXOS

```
260 'gid_profesor' bigint(20) UNSIGNED NOT NULL,
261 'nom_usuario_prof' varchar(6) NOT NULL,
262 'nombre' varchar(30) NOT NULL,
263 'apellido1' varchar(30) NOT NULL,
264 'apellido2' varchar(30) DEFAULT NULL,
265 'foto' blob,
266 'password' varchar(32) NOT NULL,
267 'email' varchar(75) DEFAULT NULL,
268 'despacho' varchar(5) DEFAULT NULL
269 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
270
271 --
272 -- RELACIONES PARA LA TABLA 'profesores':
273 --
274 --
275 --
276 -- Volcado de datos para la tabla 'profesores'
277 --
278
279 INSERT INTO 'profesores' ('gid_profesor', 'nom_usuario_prof', 'nombre', '
280 apellido1', 'apellido2', 'foto', 'password', 'email', 'despacho') VALUES
(1, 'JGS956', 'Jose', 'Guerrero', 'Sastre', NULL, '0LgdzQ2x5mmoxyzgkNJaXN5a/
281 udHmGLBM', NULL, NULL),
(6, 'JAT951', 'Javier', 'Antich', 'Tobaruela', NULL, '
282 AM9Y1sHcBGagztrVlyi8XFWmAikpEt6Y', 'javier.antich@uib.es', 'AT001'),
(8, 'MRA846', 'Miquel', 'Roca', 'Adrover', NULL, 'XKe2eNK392wkbiH+6
283 EwXawINMtuIf6Gu', NULL, NULL),
(9, 'MAT123', 'María Jesus', 'Álvarez', 'Torres', NULL, '9c2e1bd3', NULL, NULL),
284 (10, 'ABV159', 'Alberto', 'Ballesteros', 'Varela', NULL, '08076943', NULL, NULL),
285 (11, 'SBA369', 'Salvador', 'Barceló', 'Adrover', NULL, '1858fe4c', NULL, NULL),
286 (12, 'MBG', 'Manuel Alejandro', 'Barranco', 'González', NULL, '1b9d5aae', NULL, '
AT210'),
287 (13, 'SBF942', 'Sebastià', 'Bota', 'Ferragut', NULL, 'a05d229a', NULL, NULL),
288 (14, 'JMB521', 'José', 'María', 'Buades', NULL, '3a5b106a', NULL, NULL),
289 (15, 'ABB203', 'Antoni', 'Burguera', 'Burguera', NULL, '41404310', NULL, NULL),
290 (16, 'YGC643', 'Yolanda', 'González', 'Cid', NULL, '680cd385', NULL, NULL),
291 (17, 'EIR222', 'Eugeni Miguel', 'Isern', 'Riutort', NULL, '850ef6c5', NULL, NULL)
,
292 (18, 'VMM454', 'Victor', 'Martínez', 'Moll', NULL, '7009204c', NULL, NULL),
293 (19, 'GTC128', 'Gabriel', 'Torrens', 'Caldentey', NULL, 'e2b28420', NULL, NULL),
294 (20, 'FBF515', 'Francisco', 'Bonin', 'Font', NULL, '266e6e3c', NULL, NULL);
295
296
297
298 --
299 -- Estructura Stand-in para la vista 'vw_asignaturas_anos_academicos'
300 --
301 DROP VIEW IF EXISTS 'vw_asignaturas_anos_academicos';
302 CREATE TABLE 'vw_asignaturas_anos_academicos' (
303 'gid_ano_academico' bigint(20) unsigned
304 , 'gid_asignatura' bigint(20) unsigned
305 , 'gid_asign_ano' bigint(20) unsigned
306 , 'ano_academico' year(4)
307 , 'cod_asignatura' mediumint(5) unsigned
308 , 'nom_asignatura' varchar(50)
309 , 'asign_ano' varchar(64)
310 );
```

```
311
312 -----
313
314 ---
315 --- Estructura Stand-in para la vista 'vw_cursa'
316 ---
317 DROP VIEW IF EXISTS 'vw_cursa';
318 CREATE TABLE 'vw_cursa' (
319   'gid_cursa' bigint(20) unsigned
320   , 'gid_asignatura' bigint(20) unsigned
321   , 'gid_alumno' bigint(20) unsigned
322   , 'cod_asignatura' mediumint(5) unsigned
323   , 'nom_asignatura' varchar(50)
324   , 'nom_usuario_alumn' varchar(6)
325   , 'nombre' varchar(30)
326   , 'apellido1' varchar(30)
327   , 'apellido2' varchar(30)
328   , 'ano_academico' year(4)
329 );
330
331 -----
332
333 ---
334 --- Estructura Stand-in para la vista 'vw_examenes'
335 ---
336 DROP VIEW IF EXISTS 'vw_examenes';
337 CREATE TABLE 'vw_examenes' (
338   'coment_generales' text
339   , 'gid_examenes' bigint(20) unsigned
340   , 'fecha_examen' datetime
341   , 'fecha_revision' datetime
342   , 'nom_examen' varchar(200)
343   , 'cod_asignatura' mediumint(5) unsigned
344   , 'examenes_asignaturas' varchar(286)
345 );
346
347 -----
348
349 ---
350 --- Estructura Stand-in para la vista 'vw_imparte'
351 ---
352 DROP VIEW IF EXISTS 'vw_imparte';
353 CREATE TABLE 'vw_imparte' (
354   'gid_imparte' bigint(20) unsigned
355   , 'gid_profesor' bigint(20) unsigned
356   , 'nombre' varchar(30)
357   , 'apellido1' varchar(30)
358   , 'apellido2' varchar(30)
359   , 'cod_asignatura' mediumint(5) unsigned
360   , 'gid_asignatura' bigint(20) unsigned
361   , 'nom_asignatura' varchar(50)
362   , 'ano_academico' year(4)
363 );
364
365 -----
366
367 ---
```

8. ANEXOS

```
368 -- Estructura Stand-in para la vista 'vw_modelos'
369 --
370 DROP VIEW IF EXISTS 'vw_modelos';
371 CREATE TABLE 'vw_modelos' (
372 'gid_asignatura' bigint(20) unsigned
373 , 'gid_examenes' bigint(20) unsigned
374 , 'gid_modelos' bigint(20) unsigned
375 , 'modelo' varchar(1)
376 , 'fecha_examen' datetime
377 , 'nom_examen' varchar(200)
378 , 'cod_asignatura' mediumint(5) unsigned
379 , 'exam_asign' varchar(288)
380 );
381
382 -----
383
384 --
385 -- Estructura Stand-in para la vista 'vw_modelos_preguntas'
386 --
387 DROP VIEW IF EXISTS 'vw_modelos_preguntas';
388 CREATE TABLE 'vw_modelos_preguntas' (
389 'gid_examenes' bigint(20) unsigned
390 , 'gid_asignatura' bigint(20) unsigned
391 , 'gid_md1_prg' bigint(20) unsigned
392 , 'enunciado' blob
393 , 'sol_profesor' blob
394 , 'cod_pregunta' bigint(20) unsigned
395 , 'coment_generales' text
396 , 'modelo' varchar(1)
397 , 'fecha_examen' datetime
398 , 'fecha_revision' datetime
399 , 'nom_examen' varchar(200)
400 , 'cod_asignatura' mediumint(5) unsigned
401 , 'nom_asignatura' varchar(50)
402 , 'num_pregunta' tinyint(3) unsigned
403 , 'mdl_prg_vw' varchar(280)
404 );
405
406 -----
407
408 --
409 -- Estructura Stand-in para la vista 'vw_modelos_preguntas_alumnos'
410 --
411 DROP VIEW IF EXISTS 'vw_modelos_preguntas_alumnos';
412 CREATE TABLE 'vw_modelos_preguntas_alumnos' (
413 'nom_usuario_alumn' varchar(6)
414 , 'cod_asignatura' mediumint(5) unsigned
415 , 'nom_examen' varchar(200)
416 , 'fecha_examen' datetime
417 , 'num_pregunta' tinyint(3) unsigned
418 , 'nombre' varchar(30)
419 , 'apellido1' varchar(30)
420 , 'apellido2' varchar(30)
421 , 'coment_individuales' text
422 , 'enunciado' blob
423 , 'sol_alumno' blob
424 , 'correc_profesor' blob
```

```

425 );
426
427 -----
428
429 ---
430 --- Estructura para la vista de 'vw_asignaturas_anos_academicos' exportada como
431 una tabla
432 ---
433 DROP TABLE IF EXISTS 'vw_asignaturas_anos_academicos';
434 CREATE TABLE 'vw_asignaturas_anos_academicos' (
435 'gid_ano_academico' bigint(20) unsigned NOT NULL DEFAULT '0',
436 'gid_asignatura' bigint(20) unsigned NOT NULL DEFAULT '0',
437 'gid_asign_ano' bigint(20) unsigned NOT NULL DEFAULT '0',
438 'ano_academico' year(4) NOT NULL,
439 'cod_asignatura' mediumint(5) unsigned NOT NULL,
440 'nom_asignatura' varchar(50) COLLATE latin1_swedish_ci NOT NULL,
441 'asign_ano' varchar(64) COLLATE latin1_swedish_ci NOT NULL DEFAULT ''
442 );
443
444 -----
445 ---
446 --- Estructura para la vista de 'vw_cursa' exportada como una tabla
447 ---
448 DROP TABLE IF EXISTS 'vw_cursa';
449 CREATE TABLE 'vw_cursa' (
450 'gid_cursa' bigint(20) unsigned NOT NULL DEFAULT '0',
451 'gid_asignatura' bigint(20) unsigned NOT NULL DEFAULT '0',
452 'gid_alumno' bigint(20) unsigned NOT NULL DEFAULT '0',
453 'cod_asignatura' mediumint(5) unsigned NOT NULL,
454 'nom_asignatura' varchar(50) COLLATE latin1_swedish_ci NOT NULL,
455 'nom_usuario_alumn' varchar(6) COLLATE latin1_swedish_ci NOT NULL,
456 'nombre' varchar(30) COLLATE latin1_swedish_ci NOT NULL,
457 'apellido1' varchar(30) COLLATE latin1_swedish_ci NOT NULL,
458 'apellido2' varchar(30) COLLATE latin1_swedish_ci DEFAULT NULL,
459 'ano_academico' year(4) NOT NULL
460 );
461
462 -----
463 ---
464 --- Estructura para la vista de 'vw_examenes' exportada como una tabla
465 ---
466 DROP TABLE IF EXISTS 'vw_examenes';
467 CREATE TABLE 'vw_examenes' (
468 'coment_generales' text COLLATE latin1_swedish_ci DEFAULT NULL,
469 'gid_examenes' bigint(20) unsigned NOT NULL DEFAULT '0',
470 'fecha_examen' datetime NOT NULL,
471 'fecha_revision' datetime DEFAULT NULL,
472 'nom_examen' varchar(200) COLLATE latin1_swedish_ci NOT NULL,
473 'cod_asignatura' mediumint(5) unsigned NOT NULL,
474 'examenes_asignaturas' varchar(286) COLLATE latin1_swedish_ci NOT NULL DEFAULT ''
475 );
476
477 -----
478 ---
479 ---
480 ---

```

8. ANEXOS

```
481 -- Estructura para la vista de 'vw_imparte' exportada como una tabla
482 --
483 DROP TABLE IF EXISTS 'vw_imparte';
484 CREATE TABLE 'vw_imparte' (
485 'gid_imparte' bigint(20) unsigned NOT NULL DEFAULT '0',
486 'gid_profesor' bigint(20) unsigned NOT NULL DEFAULT '0',
487 'nombre' varchar(30) COLLATE latin1_swedish_ci NOT NULL,
488 'apellido1' varchar(30) COLLATE latin1_swedish_ci NOT NULL,
489 'apellido2' varchar(30) COLLATE latin1_swedish_ci DEFAULT NULL,
490 'cod_asignatura' mediumint(5) unsigned NOT NULL,
491 'gid_asignatura' bigint(20) unsigned NOT NULL DEFAULT '0',
492 'nom_asignatura' varchar(50) COLLATE latin1_swedish_ci NOT NULL,
493 'ano_academico' year(4) NOT NULL
494 );
495
496 -----
497
498 --
499 -- Estructura para la vista de 'vw_modelos' exportada como una tabla
500 --
501 DROP TABLE IF EXISTS 'vw_modelos';
502 CREATE TABLE 'vw_modelos' (
503 'gid_asignatura' bigint(20) unsigned NOT NULL DEFAULT '0',
504 'gid_examenes' bigint(20) unsigned NOT NULL DEFAULT '0',
505 'gid_modelos' bigint(20) unsigned NOT NULL DEFAULT '0',
506 'modelo' varchar(1) COLLATE latin1_swedish_ci NOT NULL,
507 'fecha_examen' datetime NOT NULL,
508 'nom_examen' varchar(200) COLLATE latin1_swedish_ci NOT NULL,
509 'cod_asignatura' mediumint(5) unsigned NOT NULL,
510 'exam_asign' varchar(288) COLLATE latin1_swedish_ci NOT NULL DEFAULT ''
511 );
512
513 -----
514
515 --
516 -- Estructura para la vista de 'vw_modelos_preguntas' exportada como una tabla
517 --
518 DROP TABLE IF EXISTS 'vw_modelos_preguntas';
519 CREATE TABLE 'vw_modelos_preguntas' (
520 'gid_examenes' bigint(20) unsigned NOT NULL DEFAULT '0',
521 'gid_asignatura' bigint(20) unsigned NOT NULL DEFAULT '0',
522 'gid_md1_prg' bigint(20) unsigned NOT NULL DEFAULT '0',
523 'enunciado' blob NOT NULL,
524 'sol_profesor' blob DEFAULT NULL,
525 'cod_pregunta' bigint(20) unsigned NOT NULL DEFAULT '0',
526 'coment_generales' text COLLATE latin1_swedish_ci DEFAULT NULL,
527 'modelo' varchar(1) COLLATE latin1_swedish_ci NOT NULL,
528 'fecha_examen' datetime NOT NULL,
529 'fecha_revision' datetime DEFAULT NULL,
530 'nom_examen' varchar(200) COLLATE latin1_swedish_ci NOT NULL,
531 'cod_asignatura' mediumint(5) unsigned NOT NULL,
532 'nom_asignatura' varchar(50) COLLATE latin1_swedish_ci NOT NULL,
533 'num_pregunta' tinyint(3) unsigned DEFAULT NULL,
534 'mdl_prg_vw' varchar(280) COLLATE latin1_swedish_ci NOT NULL DEFAULT ''
535 );
536
537 -----
```

```

538
539 ---
540 --- Estructura para la vista de 'vw_modelos_preguntas_alumnos' exportada como una
      tabla
541 ---
542 DROP TABLE IF EXISTS 'vw_modelos_preguntas_alumnos';
543 CREATE TABLE 'vw_modelos_preguntas_alumnos' (
544 'nom_usuario_alumn' varchar(6) COLLATE latin1_swedish_ci NOT NULL,
545 'cod_asignatura' mediumint(5) unsigned NOT NULL,
546 'nom_examen' varchar(200) COLLATE latin1_swedish_ci NOT NULL,
547 'fecha_examen' datetime NOT NULL,
548 'num_pregunta' tinyint(3) unsigned DEFAULT NULL,
549 'nombre' varchar(30) COLLATE latin1_swedish_ci NOT NULL,
550 'apellido1' varchar(30) COLLATE latin1_swedish_ci NOT NULL,
551 'apellido2' varchar(30) COLLATE latin1_swedish_ci DEFAULT NULL,
552 'coment_individuales' text COLLATE latin1_swedish_ci DEFAULT NULL,
553 'enunciado' blob NOT NULL,
554 'sol_alumno' blob DEFAULT NULL,
555 'correc_profesor' blob DEFAULT NULL
556 );
557
558 ---
559 --- Índices para tablas volcadas
560 ---
561 ---
562 ---
563 --- Indices de la tabla 'alumnos'
564 ---
565 ALTER TABLE 'alumnos'
566 ADD PRIMARY KEY ('gid_alumno'),
567 ADD UNIQUE KEY 'gid_alumno' ('gid_alumno'),
568 ADD UNIQUE KEY 'PROF_UQ' ('nom_usuario_alumn');
569
570 ---
571 --- Indices de la tabla 'anos_academicos'
572 ---
573 ALTER TABLE 'anos_academicos'
574 ADD PRIMARY KEY ('gid_ano_academico'),
575 ADD UNIQUE KEY 'gid_ano_academico' ('gid_ano_academico');
576
577 ---
578 --- Indices de la tabla 'asignaturas'
579 ---
580 ALTER TABLE 'asignaturas'
581 ADD PRIMARY KEY ('gid_asignatura'),
582 ADD UNIQUE KEY 'gid_asignatura' ('gid_asignatura');
583
584 ---
585 --- Indices de la tabla 'asignaturas_anos_academicos'
586 ---
587 ALTER TABLE 'asignaturas_anos_academicos'
588 ADD PRIMARY KEY ('gid_asign_ano'),
589 ADD UNIQUE KEY 'gid_asign_ano' ('gid_asign_ano'),
590 ADD KEY 'AAA_asign_FK' ('gid_asignatura'),
591 ADD KEY 'AAA_aacad_FK' ('gid_ano_academico');
592
593 ---

```

8. ANEXOS

```
594  -- Indices de la tabla 'curso'
595  --
596  ALTER TABLE 'curso'
597  ADD PRIMARY KEY ('gid_curso'),
598  ADD UNIQUE KEY 'gid_curso' ('gid_curso');
599
600  --
601  -- Indices de la tabla 'examenes'
602  --
603  ALTER TABLE 'examenes'
604  ADD PRIMARY KEY ('gid_examenes'),
605  ADD UNIQUE KEY 'gid_examenes' ('gid_examenes'),
606  ADD UNIQUE KEY 'EXA_ASSIGN_UQ' ('gid_asignatura', 'fecha_examen', 'nom_examen');
607
608  --
609  -- Indices de la tabla 'imparte'
610  --
611  ALTER TABLE 'imparte'
612  ADD PRIMARY KEY ('gid_imparte'),
613  ADD UNIQUE KEY 'gid_imparte' ('gid_imparte'),
614  ADD KEY 'imp_prof_FK' ('gid_profesor'),
615  ADD KEY 'imp_AAA_FK' ('gid_asign_ano');
616
617  --
618  -- Indices de la tabla 'modelos'
619  --
620  ALTER TABLE 'modelos'
621  ADD PRIMARY KEY ('gid_modelos'),
622  ADD UNIQUE KEY 'gid_modelos' ('gid_modelos'),
623  ADD UNIQUE KEY 'MDL_UQ' ('modelo', 'gid_examenes'),
624  ADD KEY 'MDL_EXA_FK' ('gid_examenes');
625
626  --
627  -- Indices de la tabla 'modelos_preguntas'
628  --
629  ALTER TABLE 'modelos_preguntas'
630  ADD PRIMARY KEY ('gid_mdl_prg'),
631  ADD UNIQUE KEY 'gid_mdl_prg' ('gid_mdl_prg'),
632  ADD UNIQUE KEY 'MDL_PRG_UQ' ('gid_modelos', 'cod_pregunta'),
633  ADD KEY 'MP_P_FK' ('cod_pregunta');
634
635  --
636  -- Indices de la tabla 'modelos_preguntas_alumnos'
637  --
638  ALTER TABLE 'modelos_preguntas_alumnos'
639  ADD PRIMARY KEY ('gid_mdl_prg_almn'),
640  ADD UNIQUE KEY 'gid_mdl_prg_almn' ('gid_mdl_prg_almn'),
641  ADD UNIQUE KEY 'MDL_PRG_ALMN_UQ' ('gid_mdl_prg', 'gid_alumno'),
642  ADD KEY 'MPA_P_FK' ('gid_profesor'),
643  ADD KEY 'MPA_A_FK' ('gid_alumno');
644
645  --
646  -- Indices de la tabla 'preguntas'
647  --
648  ALTER TABLE 'preguntas'
649  ADD PRIMARY KEY ('cod_pregunta'),
650  ADD UNIQUE KEY 'cod_pregunta' ('cod_pregunta'),
```



```
651 ADD KEY 'PGA_ASG_FK' ('gid_asignatura');
652
653 ---
654 --- Indices de la tabla 'profesores'
655 ---
656 ALTER TABLE 'profesores'
657 ADD PRIMARY KEY ('gid_profesor'),
658 ADD UNIQUE KEY 'gid_profesor' ('gid_profesor'),
659 ADD UNIQUE KEY 'PROF_UQ' ('nom_usuario_prof');
660
661 ---
662 --- AUTO_INCREMENT de las tablas volcadas
663 ---
664 ---
665 ---
666 --- AUTO_INCREMENT de la tabla 'alumnos'
667 ---
668 ALTER TABLE 'alumnos'
669 MODIFY 'gid_alumno' bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT, AUTO_INCREMENT
        =43;
670 ---
671 --- AUTO_INCREMENT de la tabla 'anos_academicos'
672 ---
673 ALTER TABLE 'anos_academicos'
674 MODIFY 'gid_ano_academico' bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT,
        AUTO_INCREMENT=12;
675 ---
676 --- AUTO_INCREMENT de la tabla 'asignaturas'
677 ---
678 ALTER TABLE 'asignaturas'
679 MODIFY 'gid_asignatura' bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT,
        AUTO_INCREMENT=52;
680 ---
681 --- AUTO_INCREMENT de la tabla 'asignaturas_anos_academicos'
682 ---
683 ALTER TABLE 'asignaturas_anos_academicos'
684 MODIFY 'gid_asign_ano' bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT,
        AUTO_INCREMENT=241;
685 ---
686 --- AUTO_INCREMENT de la tabla 'cursa'
687 ---
688 ALTER TABLE 'cursa'
689 MODIFY 'gid_cursa' bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT, AUTO_INCREMENT
        =375;
690 ---
691 --- AUTO_INCREMENT de la tabla 'examenes'
692 ---
693 ALTER TABLE 'examenes'
694 MODIFY 'gid_examenes' bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT, AUTO_INCREMENT
        =31;
695 ---
696 --- AUTO_INCREMENT de la tabla 'imparte'
697 ---
698 ALTER TABLE 'imparte'
699 MODIFY 'gid_imparte' bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT, AUTO_INCREMENT
        =154;
700 ---
```

8. ANEXOS

```
701  -- AUTO_INCREMENT de la tabla 'modelos'
702  --
703  ALTER TABLE 'modelos'
704  MODIFY 'gid_modelos' bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT, AUTO_INCREMENT
      =60;
705  --
706  -- AUTO_INCREMENT de la tabla 'modelos_preguntas'
707  --
708  ALTER TABLE 'modelos_preguntas'
709  MODIFY 'gid_md1_prg' bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT, AUTO_INCREMENT
      =93;
710  --
711  -- AUTO_INCREMENT de la tabla 'modelos_preguntas_alumnos'
712  --
713  ALTER TABLE 'modelos_preguntas_alumnos'
714  MODIFY 'gid_md1_prg_almn' bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT,
      AUTO_INCREMENT=17;
715  --
716  -- AUTO_INCREMENT de la tabla 'preguntas'
717  --
718  ALTER TABLE 'preguntas'
719  MODIFY 'cod_pregunta' bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT, AUTO_INCREMENT
      =35;
720  --
721  -- AUTO_INCREMENT de la tabla 'profesores'
722  --
723  ALTER TABLE 'profesores'
724  MODIFY 'gid_profesor' bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT, AUTO_INCREMENT
      =21;
725  --
726  -- Restricciones para tablas volcadas
727  --
728  --
729  --
730  -- Filtros para la tabla 'asignaturas_anos_academicos'
731  --
732  ALTER TABLE 'asignaturas_anos_academicos'
733  ADD CONSTRAINT 'AAA_aacad_FK' FOREIGN KEY ('gid_ano_academico') REFERENCES '
      anos_academicos' ('gid_ano_academico'),
734  ADD CONSTRAINT 'AAA_asign_FK' FOREIGN KEY ('gid_asignatura') REFERENCES '
      asignaturas' ('gid_asignatura');
735  --
736  --
737  -- Filtros para la tabla 'exámenes'
738  --
739  ALTER TABLE 'exámenes'
740  ADD CONSTRAINT 'EX_ASSIGN_FK' FOREIGN KEY ('gid_asignatura') REFERENCES '
      asignaturas' ('gid_asignatura');
741  --
742  --
743  -- Filtros para la tabla 'imparte'
744  --
745  ALTER TABLE 'imparte'
746  ADD CONSTRAINT 'imp_AAA_FK' FOREIGN KEY ('gid_asign_ano') REFERENCES '
      asignaturas_anos_academicos' ('gid_asign_ano'),
747  ADD CONSTRAINT 'imp_prof_FK' FOREIGN KEY ('gid_profesor') REFERENCES 'profesores'
      ('gid_profesor');
```

```

748
749 ---
750 --- Filtros para la tabla 'modelos'
751 ---
752 ALTER TABLE 'modelos'
753 ADD CONSTRAINT 'MDL_EXA_FK' FOREIGN KEY ('gid_examenes') REFERENCES 'examenes' ('
    gid_examenes');
754
755 ---
756 --- Filtros para la tabla 'modelos_preguntas'
757 ---
758 ALTER TABLE 'modelos_preguntas'
759 ADD CONSTRAINT 'MP_M_FK' FOREIGN KEY ('gid_modelos') REFERENCES 'modelos' ('
    gid_modelos'),
760 ADD CONSTRAINT 'MP_P_FK' FOREIGN KEY ('cod_pregunta') REFERENCES 'preguntas' ('
    cod_pregunta');
761
762 ---
763 --- Filtros para la tabla 'modelos_preguntas_alumnos'
764 ---
765 ALTER TABLE 'modelos_preguntas_alumnos'
766 ADD CONSTRAINT 'MPA_A_FK' FOREIGN KEY ('gid_alumno') REFERENCES 'alumnos' ('
    gid_alumno'),
767 ADD CONSTRAINT 'MPA_MP_FK' FOREIGN KEY ('gid mdl_prg') REFERENCES '
    modelos_preguntas' ('gid mdl_prg'),
768 ADD CONSTRAINT 'MPA_P_FK' FOREIGN KEY ('gid_profesor') REFERENCES 'profesores' ('
    gid_profesor');
769
770 ---
771 --- Filtros para la tabla 'preguntas'
772 ---
773 ALTER TABLE 'preguntas'
774 ADD CONSTRAINT 'PGA_ASG_FK' FOREIGN KEY ('gid_asignatura') REFERENCES 'examenes'
    ('gid_asignatura');
775
776 /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
777 /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
778 /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

```

8.2 Grocery CRUD

8.2.1 Controlador

```

1 <?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
2
3 class Examples extends CI_Controller {
4
5     public function __construct()
6     {
7         parent::__construct();
8
9         $this->load->database();
10        $this->load->helper('url');
11        $this->load->helper('security');
12        $this->load->library('session');

```

```
13 $this->load->library('grocery_CRUD');
14 }
15
16 public function _example_output($output = null)
17 {
18     $this->load->view('example.php', (array)$output);
19 }
20
21
22 public function index()
23 {
24     $this->_example_output((object)array('output' => '', 'js_files' => array(),
25     'css_files' => array()));
26 }
27
28 function encrypt_password_callback($post_array) {
29
30     $post_array['password'] = md5($post_array['password']);
31
32     return $post_array;
33 }
34
35 function dealwithnull($field_value) {
36     // if (strlen($field_value) == 0) {
37     //     return "NULL";
38     // }
39     // else {
40     //     return ($field_value);
41     // }
42 }
43
44 public function select_asig($gid_asignatura)
45 {
46     $this->db->select('cod_asignatura, nom_asignatura')->from('asignaturas')->
47     where('gid_asignatura', $gid_asignatura);
48     $query = $this->db->get();
49
50     $newdata=array(
51     'cod_asignatura'=> $query->row()->cod_asignatura,
52     'nom_asignatura'=> $query->row()->nom_asignatura,
53     'gid_asignatura'=> $gid_asignatura,
54     );
55
56     $this->session->set_userdata($newdata);
57     redirect('examples/selec_ano_management');
58 }
59
60 public function select_ano($gid_asign_ano)
61 {
62     $this->db->select('ano_academico, gid_ano_academico')->from('
63     vw_asignaturas_anos_academicos')->where('gid_asign_ano', $gid_asign_ano);
64     $query = $this->db->get();
65
66     $newdata=array(
67     'ano_academico'=> $query->row()->ano_academico,
```

```

67     'gid_ano_academico' => $query->row()->gid_ano_academico,
68     'gid_asign_ano' => $gid_asign_ano,
69     );
70
71     $this->session->set_userdata($newdata);
72     redirect('examples/exam_management');
73 }
74
75 public function select_ano_alumn ($gid_asign_ano)
76 {
77     $this->db->select('ano_academico, gid_ano_academico')->from('
78     vw_asignaturas_anos_academicos')->where('gid_asign_ano', $gid_asign_ano);
79     $query = $this->db->get();
80
81     $newdata=array(
82     'ano_academico' => $query->row()->ano_academico,
83     'gid_ano_academico' => $query->row()->gid_ano_academico,
84     'gid_asign_ano' => $gid_asign_ano,
85     );
86
87     $this->session->set_userdata($newdata);
88     redirect('examples/alumn_matri_management');
89 }
90
91 public function select_ano_prof ($gid_asign_ano)
92 {
93     $this->db->select('ano_academico, gid_ano_academico')->from('
94     vw_asignaturas_anos_academicos')->where('gid_asign_ano', $gid_asign_ano);
95     $query = $this->db->get();
96
97     $newdata=array(
98     'ano_academico' => $query->row()->ano_academico,
99     'gid_ano_academico' => $query->row()->gid_ano_academico,
100     'gid_asign_ano' => $gid_asign_ano,
101     );
102
103     $this->session->set_userdata($newdata);
104     redirect('examples/prof_matri_management');
105 }
106
107 public function select_exam($gid_examenes)
108 {
109     $this->db->select('nom_examen, fecha_examen')->from('examenes')->where('
110     gid_examenes', $gid_examenes);
111     $query = $this->db->get();
112
113     $newdata=array(
114     'nom_examen' => $query->row()->nom_examen,
115     'fecha_examen' => $query->row()->fecha_examen,
116     'gid_examenes' => $gid_examenes,
117     );
118
119     $this->session->set_userdata($newdata);
120     redirect('examples/model_management');
121 }

```

```
121
122 function generate_password_prof($post_array) {
123     $pwd = substr(md5(microtime()), 1, 8);
124     $this->db->query("insert into profesores(nom_usuario_prof, nombre, apellido1,
125         apellido2, foto, password, email, despacho)
126         values ('".$post_array['nom_usuario_prof']."' , '".$post_array['nombre']."' , '".$
127         $post_array['apellido1']."' , '".$post_array['apellido2']."' , '".$
128         $post_array['foto']."' , '".$pwd."' , '".$post_array['email']."' , '".$post_array[
129         'despacho']."' )");
130 }
131
132 public function selec_asig_management()
133 {
134     $crud = new grocery_CRUD();
135     $crud->set_theme('datatables');
136     $crud->set_table('asignaturas');
137     $crud->set_subject('asignaturas');
138     $crud->set_language('spanish');
139     $crud->columns(
140         'cod_asignatura',
141         'nom_asignatura'
142     );
143     $crud->set_relation_n_n('años_imparticion', 'asignaturas_años_academicos', '
144     años_academicos', 'gid_asignatura', 'gid_año_academico', 'año_academico');
145     $crud->display_as('cod_asignatura', 'Codigo asignatura');
146     $crud->display_as('nom_asignatura', 'Nombre asignatura');
147     $crud->add_action('Seleccionar', base_url('img/check_verde.jpg'), 'examples/
148     select_asig');
149     $crud->required_fields('cod_asignatura', 'nom_asignatura');
150
151     $output = $crud->render();
152     $data = array();
153     $output->data = $data;
154
155     $this->load->view('accesos', $output);
156 }
157
158 public function selec_año_management()
159 {
160     $crud = new grocery_CRUD();
161     $crud->set_theme('datatables');
162     $crud->set_table('vw_asignaturas_años_academicos');
163     $crud->set_subject('vw_asignaturas_años_academicos');
164     $crud->set_primary_key('gid_asign_año');
165     $crud->set_language('spanish');
166     $crud->columns(
167         'año_academico'
168     );
169     $crud->unset_columns();
170     $crud->unset_add();
171     $crud->unset_edit();
172     $crud->unset_read();
173     $crud->unset_delete();
```

```

173 $crud->add_action('Seleccionar', base_url('img/check_verde.jpg'), 'examples/
    select_ano');
174 $crud->add_action('Alumnos', base_url('img/check_verde.jpg'), 'examples/
    select_ano_alumn');
175 $crud->add_action('Profesores', base_url('img/check_verde.jpg'), 'examples/
    select_ano_prof');
176
177
178 $crud->field_type('gid_asignatura', 'hidden', $this->session->userdata('
    gid_asignatura'));
179 $crud->where('gid_asignatura', $this->session->userdata('gid_asignatura'));
180
181 $crud->display_as('ano_academico', 'año acad&eacute;mico');
182
183 $output = $crud->render();
184 $data = array();
185 $data['cod_asignatura'] = $this->session->userdata('cod_asignatura');
186 $data['nom_asignatura'] = $this->session->userdata('nom_asignatura');
187 $output->data = $data;
188
189 $this->load->view('anos_acad', $output);
190
191 }
192
193
194 public function asign_management()
195 {
196     $crud = new grocery_CRUD();
197     $crud->set_theme('datatables');
198     $crud->set_table('asignaturas');
199     $crud->set_subject('asignaturas');
200     $crud->set_language('spanish');
201     $crud->set_relation_n_n('años_imparticion', 'asignaturas_anos_academicos', '
        anos_academicos', 'gid_asignatura', 'gid_ano_academico', 'ano_academico');
202     $crud->unique_fields(array('cod_asignatura', 'url'));
203     $crud->display_as('cod_asignatura', 'C&oacute;digo');
204     $crud->display_as('nom_asignatura', 'Nombre asignatura');
205     $crud->field_type('gid_asignatura', 'hidden', $this->session->userdata('
        gid_asignatura'));
206     $crud->where('gid_asignatura', $this->session->userdata('gid_asignatura'));
207     $output = $crud->render();
208
209
210
211     $data = array();
212     $data['cod_asignatura'] = $this->session->userdata('cod_asignatura');
213     $data['nom_asignatura'] = $this->session->userdata('nom_asignatura');
214     $data['nom_formulario'] = 'Asignaturas';
215     $output->data = $data;
216
217     $this->_example_output($output);
218 }
219
220
221 public function ano_aca_management()
222 {
223

```

8. ANEXOS

```
224 $crud = new grocery_CRUD();
225 $crud->set_theme('datatables');
226 $crud->set_table('anos_academicos');
227 $crud->set_subject('años acad&eacute;micos');
228 $crud->set_language('spanish');
229 $crud->set_relation_n_n('asignaturas', 'asignaturas_anos_academicos', '
    asignaturas', 'gid_ano_academico', 'gid_asignatura', 'nom_asignatura');
230 $crud->display_as('ano_academico', 'Año acad&eacute;mico');
231 $crud->display_as('nom_asignatura', 'Nombre asignatura');
232 $crud ->required_fields('ano_academico');
233
234 $output = $crud->render();
235
236
237 $data = array();
238 $data['cod_asignatura'] = $this->session->userdata('cod_asignatura');
239 $data['nom_asignatura'] = $this->session->userdata('nom_asignatura');
240 $data['nom_formulario'] = 'años academicos';
241 $output->data = $data;
242 $this->_example_output($output);
243 }
244
245 public function alumn_matri_management()
246 {
247     $crud = new grocery_CRUD();
248     $crud->set_theme('datatables');
249     $crud->set_table('vw_asignaturas_anos_academicos');
250     $crud->set_subject('vw_asignaturas_anos_academicos');
251     $crud->set_language('spanish');
252     $crud->set_primary_key('gid_asign_ano', 'vw_asignaturas_anos_academicos');
253
254     $crud->set_relation_n_n('alumno', 'curso', 'alumnos', 'gid_asign_ano', '
        gid_alumno', '{nombre} {apellido1} {apellido2}');
255     $crud ->unset_columns('gid_asignatura', 'gid_asign_ano', 'gid_ano_academico', '
        asign_ano');
256     $crud->unset_add();
257     $crud->unset_edit();
258     $crud->unset_delete();
259
260     $crud->field_type('gid_asignatura', 'hidden', $this->session->userdata('
        gid_asignatura'));
261     $crud->where('gid_asignatura', $this->session->userdata('gid_asignatura'));
262
263     $crud->field_type('ano_academico', 'hidden', $this->session->userdata('
        ano_academico'));
264     $crud->where('ano_academico', $this->session->userdata('ano_academico'));
265
266     $output = $crud->render();
267     $data = array();
268     $data['cod_asignatura'] = $this->session->userdata('cod_asignatura');
269     $data['nom_asignatura'] = $this->session->userdata('nom_asignatura');
270     $data['nom_formulario'] = 'Alumnos matriculados';
271     $output->data = $data;
272     $this->load->view('alumn_matric', $output);
273 }
274
275 public function alumn_management()
```



```

276     {
277     $crud = new grocery_CRUD();
278     $crud->set_theme('datatables');
279     $crud->set_table('alumnos');
280     $crud->set_subject('alumnos');
281     $crud->set_language('spanish');
282     $crud->set_primary_key('gid_asign_ano', 'vw_asignaturas_anos_academicos');
283
284     $crud->set_relation_n_n('asignaturas_y_años_academicos', 'cursa', '
vw_asignaturas_anos_academicos', 'gid_alumno', 'gid_asign_ano', 'asign_ano');
285
286
287     $crud->display_as('nom_usuario_alumn', 'Nombre de usuario');
288     $crud->display_as('apellido1', 'Primer apellido');
289     $crud->display_as('apellido2', 'Segundo apellido');
290
291     $crud ->required_fields('nom_usuario_alumn', 'nombre', 'password');
292
293
294     $crud->field_type('password', 'password');
295     $crud ->unset_columns('password');
296
297     $crud->set_field_upload('foto', 'assets/uploads/files');
298     $crud->callback_before_insert(array($this, 'encrypt_password_callback'));
299
300
301     $output = $crud->render();
302
303     $data = array();
304     $data['cod_asignatura'] = $this->session->userdata('cod_asignatura');
305     $data['nom_asignatura'] = $this->session->userdata('nom_asignatura');
306     $data['nom_formulario'] = 'Alumnos';
307     $output->data = $data;
308     $this->load->view('accesos', $output);
309 }
310
311 public function prof_matri_management()
312     {
313     $crud = new grocery_CRUD();
314
315     $crud->set_theme('datatables');
316     $crud->set_table('vw_asignaturas_anos_academicos');
317     $crud->set_subject('vw_asignaturas_anos_academicos');
318     $crud->set_language('spanish');
319     $crud->set_primary_key('gid_asign_ano', 'vw_asignaturas_anos_academicos');
320
321     $crud->set_relation_n_n('Profesor', 'imparte', 'Profesores', 'gid_asign_ano', '
gid_profesor', '{nombre} {apellido1} {apellido2}');
322     $crud ->unset_columns('gid_asignatura', 'gid_asign_ano', 'gid_ano_academico', '
asign_ano');
323     $crud->unset_add();
324     $crud->unset_edit();
325     $crud->unset_delete();
326
327     $crud->field_type('gid_asignatura', 'hidden', $this->session->userdata('
gid_asignatura'));
328     $crud->where('gid_asignatura', $this->session->userdata('gid_asignatura'));

```

```

329
330     $crud->field_type('ano_academico', 'hidden', $this->session->userdata('ano_academico'));
331     $crud->where('ano_academico', $this->session->userdata('ano_academico'));
332
333     $output = $crud->render();
334     $data = array();
335     $data['cod_asignatura'] = $this->session->userdata('cod_asignatura');
336     $data['nom_asignatura'] = $this->session->userdata('nom_asignatura');
337     $data['nom_formulario'] = 'Profesores matriculados';
338     $output->data = $data;
339     $this->load->view('prof_matric', $output);
340
341 }
342
343 public function prof_management()
344 {
345     $crud = new grocery_CRUD();
346     $crud->set_theme('datatables');
347     $crud->set_table('profesores');
348     $crud->set_subject('profesores');
349     $crud->set_language('spanish');
350     $crud->set_primary_key('gid_asign_ano', 'vw_asignaturas_anos_academicos');
351     $crud->set_relation_n_n('asignaturas_y_años_academicos', 'imparte', 'vw_asignaturas_anos_academicos', 'gid_profesor', 'gid_asign_ano', 'asign_ano');
352     $crud->display_as('ano_academico', 'Año acad&eacute;mico');
353
354     $crud->display_as('nom_usuario_prof', 'Nombre de usuario');
355     $crud->display_as('apellido1', 'Primer apellido');
356     $crud->display_as('apellido2', 'Segundo apellido');
357
358     $crud->required_fields('nom_usuario_prof', 'nombre', 'password');
359
360     $crud->callback_insert(array($this, 'generate_password_prof'));
361     $crud->unset_fields('password');
362     $crud->unset_columns('password');
363     $crud->set_field_upload('foto', 'assets/uploads/files');
364
365     $output = $crud->render();
366
367     $data = array();
368     $data['cod_asignatura'] = $this->session->userdata('cod_asignatura');
369     $data['nom_asignatura'] = $this->session->userdata('nom_asignatura');
370     $data['nom_formulario'] = 'Profesores';
371     $output->data = $data;
372     $this->load->view('accesos', $output);
373
374 }
375 public function exam_management()
376 {
377     $crud = new grocery_CRUD();
378     $crud->set_theme('datatables');
379     $crud->set_table('examenes');
380     $crud->set_subject('examenes');
381     $crud->set_language('spanish');
382     $crud->display_as('fecha_examen', 'fecha del examen');
383     $crud->display_as('nom_examen', 'nombre del examen');

```

```

384 $crud->display_as('fecha_revision', 'fecha de revisi&oacute;n');
385 $crud->display_as('coment_generales', 'Comentarios generales');
386 $crud->required_fields('fecha_examen', 'nom_examen');
387
388 $crud->add_action('Seleccionar', base_url('img/check_verde.jpg'), 'examples/
select_exam');
389
390 $crud->field_type('gid_asignatura', 'hidden', $this->session->userdata('
gid_asignatura'));
391
392 $crud->where('gid_asignatura', $this->session->userdata('gid_asignatura'));
393
394 $crud->where("fecha_examen >=", $this->session->userdata('ano_academico').'
-09-01');
395 $crud->where("fecha_examen <=", (intval($this->session->userdata('
ano_academico'))+1).'-07-31');
396
397 $crud->unset_columns('gid_asignatura');
398 $output = $crud->render();
399
400 $data = array();
401 $data['cod_asignatura'] = $this->session->userdata('cod_asignatura');
402 $data['nom_asignatura'] = $this->session->userdata('nom_asignatura');
403 $data['nom_formulario'] = 'Exámenes';
404
405 $output->data = $data;
406 $this->_example_output($output);
407 }
408
409 public function model_management()
410 {
411     $crud = new grocery_CRUD();
412     $crud->set_theme('datatables');
413     $crud->set_table('modelos');
414     $crud->set_subject('modelos');
415     $crud->set_language('spanish');
416     $crud->set_primary_key('gid_examenes', 'vw_examenes');
417
418     $crud->set_relation_n_n('preguntas_de_cada_modelo', 'modelos_preguntas', '
preguntas', 'gid_modelos', 'cod_pregunta', 'enunciado', 'num_pregunta', array('
gid_asignatura'=>$this->session->userdata('gid_asignatura')));
419
420
421
422     $crud->callback_column('preguntas_de_cada_modelo', array($this, '
_callback_webpage_url'));
423
424
425
426     $crud->field_type('gid_examenes', 'hidden', $this->session->userdata('
gid_examenes'));
427     $crud->where('gid_examenes', $this->session->userdata('gid_examenes'));
428     $crud->unset_columns('gid_examenes');
429
430     $crud->required_fields('modelo');
431     $output = $crud->render();
432

```

```

433
434     $data = array();
435     $data['cod_asignatura'] = $this->session->userdata('cod_asignatura');
436     $data['nom_asignatura'] = $this->session->userdata('nom_asignatura');
437     $data['nom_examen'] = $this->session->userdata('nom_examen');
438     $data['fecha_examen'] = $this->session->userdata('fecha_examen');
439     $data['nom_formulario'] = 'Modelos';
440     $output->data = $data;
441     $this->load->view('model', $output);
442 }
443
444 public function _callback_webpage_url($value, $row)
445 {
446     $exit = false;
447     $offset = 0;
448     $str = "";
449
450     while ($exit == false) {
451         $pos = strpos($value, ',', $offset);
452         if ($pos !== false) {
453             $len = $pos - $offset;
454             $sep = ", ";
455         }
456         else {
457             $len = strlen($value) - $offset;
458             $sep = "";
459             $exit = true;
460         }
461
462         $substrvalue = substr($value, $offset, $len);
463         $str = $str . "<a href='".base_url('assets/uploads/files/'. $substrvalue).
464             "'>$substrvalue</a>" . $sep;
465
466         $offset = $pos + 1;
467     }
468     return $str;
469 }
470
471 public function preg_management()
472 {
473     $crud = new grocery_CRUD();
474     $crud->set_theme('datatables');
475     $crud->set_table('preguntas');
476     $crud->set_subject('preguntas');
477     $crud->set_language('spanish');
478     $crud->set_primary_key('gid_modelos', 'vw_modelos');
479     $crud->set_relation_n_n('modelos_de_cada_pregunta', 'modelos_preguntas', '
vw_modelos', 'cod_pregunta', 'gid_modelos', 'exam_asign', NULL, array('
gid_asignatura'=>$this->session->userdata('gid_asignatura')));
480     $crud->set_field_upload('enunciado', 'assets/uploads/files');
481     $crud->set_field_upload('sol_profesor', 'assets/uploads/files');
482
483     $crud->display_as('sol_profesor', 'Soluci&oacute;n del profesor');
484     $crud->display_as('coment_generales', 'Comentarios generales');
485
486

```

```

487 $crud->field_type('gid_asignatura', 'hidden', $this->session->userdata('
    gid_asignatura'));
488 $crud->where('gid_asignatura', $this->session->userdata('gid_asignatura'));
489
490 $crud->unset_columns('gid_asignatura');
491 $crud->required_fields('enunciado');
492 $output = $crud->render();
493
494 $data = array();
495 $data['cod_asignatura'] = $this->session->userdata('cod_asignatura');
496 $data['nom_asignatura'] = $this->session->userdata('nom_asignatura');
497 $data['nom_examen'] = $this->session->userdata('nom_examen');
498 $data['fecha_examen'] = $this->session->userdata('fecha_examen');
499 $data['nom_formulario'] = 'Preguntas';
500 $output->data = $data;
501 $this->load->view('model', $output);
502
503 }
504
505 public function exam_preg_alumn_management()
506 {
507     $crud = new grocery_CRUD();
508     $crud->set_theme('datatables');
509     $crud->set_table('modelos_preguntas_alumnos');
510     $crud->set_subject('examen de un alumno');
511     $crud->set_language('spanish');
512
513     $crud->set_primary_key('gid_mdl_prg', 'vw_modelos_preguntas');
514     $crud->set_primary_key('gid_profesor', 'vw_imparte');
515     $crud->set_primary_key('gid_alumno', 'vw_cursa');
516     $crud->set_relation('gid_profesor', 'vw_imparte', '{apellido1} {apellido2}, {
    nombre}', array('gid_asignatura'=>$this->session->userdata('gid_asignatura'), {
    'ano_academico'=>$this->session->userdata('ano_academico')));
517     $crud->set_relation('gid_alumno', 'vw_cursa', '{apellido1} {apellido2}, {nombre
    }', array('gid_asignatura'=>$this->session->userdata('gid_asignatura'), '
    ano_academico'=>$this->session->userdata('ano_academico')));
518     $crud->set_relation('gid_mdl_prg', 'vw_modelos_preguntas', 'Modelo: {modelo},
    enunciado:{enunciado}, pregunta {num_pregunta}', array('gid_asignatura'=>$this
    ->session->userdata('gid_asignatura'), 'fecha_examen'=>$this->session->
    userdata('fecha_examen')));
519
520     $crud->fields(
521         'gid_profesor',
522         'gid_alumno',
523         'gid_mdl_prg',
524         'sol_alumno',
525         'correc_profesor',
526         'coment_individuales'
527     );
528     $crud->columns(
529         'gid_alumno',
530         'gid_mdl_prg',
531         'gid_profesor',
532         'sol_alumno',
533         'correc_profesor',
534         'coment_individuales'
535     );

```

8. ANEXOS

```
536 $crud->set_field_upload('sol_alumno', 'assets/uploads/files');
537 $crud->set_field_upload('correc_profesor', 'assets/uploads/files');
538
539 $crud->display_as('gid_profesor', 'Profesor corrector');
540 $crud->display_as('gid_alumno', 'Alumno');
541 $crud->display_as('gid_md1_prg', 'Modelo y pregunta');
542 $crud->display_as('sol_alumno', 'Soluci&oacute;n del alumno');
543 $crud->display_as('correc_profesor', 'Correcci&oacute;n del profesor');
544 $crud->display_as('coment_individuales', 'Comentarios individuales');
545
546 $crud->where('gid_examenes', $this->session->userdata('gid_examenes'));
547 $crud->where('jbc36fd08.gid_asignatura', $this->session->userdata('
gid_asignatura'));
548 $crud->where('jbc36fd08.ano_academico', $this->session->userdata('
ano_academico'));
549 $crud->where('j88d52f71.gid_asignatura', $this->session->userdata('
gid_asignatura'));
550 $crud->where('j88d52f71.ano_academico', $this->session->userdata('
ano_academico'));
551
552 $crud->required_fields('gid_alumno', 'gid_md1_prg', 'sol_alumno');
553
554 $output = $crud->render();
555
556 $data = array();
557 $data['cod_asignatura'] = $this->session->userdata('cod_asignatura');
558 $data['nom_asignatura'] = $this->session->userdata('nom_asignatura');
559 $data['nom_examen'] = $this->session->userdata('nom_examen');
560 $data['fecha_examen'] = $this->session->userdata('fecha_examen');
561 $data['nom_formulario'] = 'Preguntas de cada alumno en funci&oacute;n del examen';
562 $output->data = $data;
563 $this->load->view('model', $output);
564 }
565 }
```

8.2.2 Vistas

vista de accesos

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8" />
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <?php
7 foreach($css_files as $file): ?>
8 <link type="text/css" rel="stylesheet" href="<?php echo $file; ?>" />
9 <?php endforeach; ?>
10 <?php foreach($js_files as $file): ?>
11 <script src="<?php echo $file; ?>"></script>
12 <?php endforeach; ?>
13 <style type='text/css'>
14 body
15 {
16     background-image: url(<?php echo base_url('img/background.jpg'); ?>);
17     background-repeat: repeat;
```

```

18     font-family: Arial;
19     font-size: 14px;
20 }
21 a {
22     color: blue;
23     text-decoration: none;
24     font-size: 14px;
25 }
26 </style>
27 </head>
28     <body>
29         <div>
30             
31             <h1>SELECCIONA UNA DE LAS TRES OPCIONES</font></h1>
32             <a href='<?php echo site_url('examples/selec_asig_management') ?>'>
33                 selecci&oacute;n de asignatura</a> |
34             <a href='<?php echo site_url('examples/alumn_management') ?>'>Dar de
35                 alta a alumnos</a> |
36             <a href='<?php echo site_url('examples/prof_management') ?>'>Dar de
37                 alta a profesores</a> |
38         </div>
39         <div style='height:20px;'></div>
40         <div>
41             <?php echo $output; ?>
42         </div>
43     </body>
44 </head>
45 </html>

```

vista de selecci3n de a1os acad3micos

```

1
2 <!DOCTYPE html>
3 <html>
4 <head>
5 <meta charset="utf-8" />
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 <?php
8 foreach($css_files as $file): ?>
9 <link type="text/css" rel="stylesheet" href="<?php echo $file; ?>" />
10 <?php endforeach; ?>
11 <?php foreach($js_files as $file): ?>
12 <script src="<?php echo $file; ?>"></script>
13 <?php endforeach; ?>
14 <style type='text/css'>
15 body
16 {
17     background-image: url(<?php echo base_url('img/background.jpg'); ?>);
18     background-repeat: repeat;
19     font-family: Arial;
20     font-size: 14px;
21 }
22 a {
23     color: blue;
24     text-decoration: none;

```

8. ANEXOS

```
25     font-size: 14px;
26 }
27 </style>
28 </head>
29     <body>
30         <div>
31             
32             <h1>¿A qu&eacute; año acad&eacute;mico quieres acceder?</font></h1>
33             <h1>C&oacute;digo: <font color="blue"><?php echo $data[
cod_asignatura']?></font> Asignatura: <font color="blue"><?php echo $data[
nom_asignatura']?></font></font></h1>
34
35
36         </div>
37         <div style='height:20px;'></div>
38         <div>
39             <?php echo $output; ?>
40         </div>
41     </body>
42 </head>
43 </html>
```

vista de los exámenes

```
1
2 <!DOCTYPE html>
3 <html>
4 <head>
5 <meta charset="utf-8" />
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 <?php
8 foreach($css_files as $file): ?>
9 <link type="text/css" rel="stylesheet" href="<?php echo $file; ?>" />
10 <?php endforeach; ?>
11 <?php foreach($js_files as $file): ?>
12 <script src="<?php echo $file; ?>"></script>
13 <?php endforeach; ?>
14 <style type='text/css'>
15 body
16 {
17     background-image: url(<?php echo base_url('img/background.jpg'); ?>);
18
19     background-repeat: repeat;
20     font-family: Arial;
21     font-size: 14px;
22 }
23 a {
24     color: blue;
25     text-decoration: none;
26     font-size: 14px;
27 }
28 </style>
29 </head>
30 <body>
31     <div>
32         
```



```

33     <h1>Gesti&oacute;n de <font color="blue"><?php echo $data[ '
nom_formulario ' ]?></font></font></h1>
34     <h1>C&oacute;digo: <font color="blue"><?php echo $data[ '
cod_asignatura ' ]?></font> Asignatura: <font color="blue"><?php echo $data[ '
nom_asignatura ' ]?></font></font></h1>
35     <a href='<?php echo site_url('examples/selec_asig_management')?'>'>
seleccion de asignatura</a> |
36     <a href='<?php echo site_url('examples/exam_management')?'>'>Exámenes
</a>
37     </div>
38     <div style='height:20px;'></div>
39     <div>
40         <?php echo $output; ?>
41     </div>
42 </body>
43 </head>
44 </html>

```

vista de las características del examen

```

1
2 <!DOCTYPE html>
3 <html>
4 <head>
5 <meta charset="utf-8" />
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 <?php
8 foreach($css_files as $file): ?>
9 <link type="text/css" rel="stylesheet" href="<?php echo $file; ?>" />
10 <?php endforeach; ?>
11 <?php foreach($js_files as $file): ?>
12 <script src="<?php echo $file; ?>"></script>
13 <?php endforeach; ?>
14 <style type='text/css'>
15 body
16 {
17     background-image: url(<?php echo base_url('img/background.jpg'); ?>);
18     background-repeat: repeat;
19     font-family: Arial;
20     font-size: 14px;
21 }
22 a {
23     color: blue;
24     text-decoration: none;
25     font-size: 14px;
26 }
27 </style>
28 </head>
29 <body>
30     <div>
31         
32         <h1 >C&oacute;digo: <font color="blue"><?php echo $data[ 'cod_asignatura '
]?'></font> Asignatura: <font color="blue"><?php echo $data[ 'nom_asignatura '
]?'></font>
33         <h1>Estas editando el examen <font color="blue"><?php echo $data[ '
nom_examen' ]?'></font> con fecha <font color="blue"><?php echo $data[ '

```

```
34 fecha_examen']?></font></h1>
35
36
37     <a href='<?php echo site_url('examples/exam_management')?'>'>Cambiar de
examen</a> |
38     <a href='<?php echo site_url('examples/model_management')?'>'>Modelos</a>
|
39     <a href='<?php echo site_url('examples/preg_management')?'>'>Preguntas</a>
|
40     <a href='<?php echo site_url('examples/exam_preg_alumn_management')?'>'>
Soluciones de cada alumno</a>
41
42 </div>
43 <div style='height:20px;'></div>
44 <div>
45     <?php echo $output; ?>
46 </div>
47 </body>
48 </head>
49 </html>
```

8.3 Aplicación cliente

8.3.1 Pantalla de login

MainActivity

```
1
2 package com.example.ramon.clienteandroid;
3 import android.app.ProgressDialog;
4 import android.content.Context;
5 import android.content.Intent;
6 import android.os.AsyncTask;
7 import android.support.v7.app.AppCompatActivity;
8 import android.os.Bundle;
9 import android.util.Log;
10 import android.view.View;
11 import android.widget.Button;
12 import android.widget.EditText;
13 import android.widget.Toast;
14
15 import org.w3c.dom.Text;
16
17 import java.io.IOException;
18 import java.io.InputStream;
19 import java.io.PrintStream;
20 import java.net.InetAddress;
21 import java.net.Socket;
22 import java.net.UnknownHostException;
23
24 public class MainActivity extends AppCompatActivity {
25
26 private Button button_acce;
27 private EditText editText3;
```

```

28 private EditText editText4;
29 private Context context = this;
30 private String contrasena;
31 String [] datos;
32
33 private static final int SERVERPORT = 5000;
34 private static final String ADDRESS = "192.168.112.2";
35
36
37 public static class Hash {
38     private static final char[] CONSTS_HEX = { '0','1','2','3','4','5','6','7','8',
39         '9','a','b','c','d','e','f' };
40     public static String getHash(String txt) {
41         try {
42             java.security.MessageDigest md = java.security.MessageDigest.
43             getInstance("MD5");
44             byte[] array = md.digest(txt.getBytes());
45             StringBuffer sb = new StringBuffer();
46             for (int i = 0; i < array.length; ++i) {
47                 int bajo = (int)(array[i] & 0xf);
48                 int alto = (int)((array[i] & 0xf0) >> 4);
49                 sb.append(CONSTS_HEX[alto]);
50                 sb.append(CONSTS_HEX[bajo]);
51             }
52             return sb.toString();
53         } catch (java.security.NoSuchAlgorithmException e) {
54             System.out.println(e.getMessage());
55         }
56         return null;
57     }
58 }
59
60 public static String md5(String txt) {
61     return Hash.getHash(txt);
62 }
63
64 @Override
65 protected void onCreate(Bundle savedInstanceState) {
66     super.onCreate(savedInstanceState);
67     setContentView(R.layout.activity_main);
68
69     editText3 = ((EditText) findViewById(R.id.editText3));
70     editText4 = ((EditText) findViewById(R.id.editText4));
71     boton_acce = ((Button) findViewById(R.id.button3));
72
73     //enviar credenciales al servidor
74     boton_acce.setOnClickListener(
75         new View.OnClickListener() {
76             public void onClick(View view) {
77                 if (editText3.getText().toString().length() == 6) {
78                     contrasena = editText4.getText().toString();
79                     contrasena = Hash.md5(contrasena);
80                     MyATaskCliente myATasUSER = new MyATaskCliente();
81                     myATasUSER.execute("MAc10"+editText3.getText().toString()+
82                     contrasena);

```

```
82         } else {
83             Toast.makeText(context, "Escriba usuario de 6 caracteres y
login en la parte superoir", Toast.LENGTH_LONG).show();
84         }
85     }
86     });
87 }
88
89 //Clase para interactuar con el servidor
90
91 class MyATaskCliente extends AsyncTask<String,Void,String>{
92
93     //Ventana que bloqueara la pantalla del movil hasta recibir respuesta del
servidor
94     ProgressDialog progressDialog;
95
96     // muestra una ventana emergente
97     @Override
98     protected void onPreExecute ()
99     {
100         super.onPreExecute ();
101         progressDialog = new ProgressDialog(context);
102         progressDialog.setCanceledOnTouchOutside(false);
103         progressDialog.setTitle("Connecting to server");
104         progressDialog.setMessage("Please wait ...");
105         progressDialog.show();
106     }
107
108     //Se conecta al servidor y trata resultado
109     @Override
110     protected String doInBackground(String... values){
111
112         try {
113             //Se conecta al servidor
114             InetAddress serverAddr = InetAddress.getByName(ADDRESS);
115             Log.i("I/TCP Client", "Connecting...");
116             Socket socket = new Socket(serverAddr, SERVERPORT);
117             Log.i("I/TCP Client", "Connected to server");
118
119             //envia peticion de cliente
120             Log.i("I/TCP Client", "Send data to server");
121             PrintStream output = new PrintStream(socket.getOutputStream());
122             String request = values[0];
123             output.println(request);
124
125             //recibe respuesta del servidor y formatea a String
126             Log.i("I/TCP Client", "Received data to server");
127             InputStream stream = socket.getInputStream();
128             byte[] lenBytes = new byte[256];
129             stream.read(lenBytes,0,256);
130             String received = new String(lenBytes,"UTF-8").trim();
131             Log.i("I/TCP Client", "Received " + received);
132             Log.i("I/TCP Client", "");
133             //cierra conexion
134             socket.close();
135             return received;
136         } catch (UnknownHostException ex) {
```

```

137         Log.e("E/TCP Client", "" + ex.getMessage());
138         return ex.getMessage();
139     } catch (IOException ex) {
140         Log.e("E/TCP Client", "" + ex.getMessage());
141         return ex.getMessage();
142     }
143 }
144
145 //Ocultar ventana emergente y muestra resultado en pantalla
146 @Override
147 protected void onPostExecute(String value) {
148     datos = value.split("#");
149     progressDialog.dismiss();
150     if (datos[0].length() == 2) {
151         //abrir nuevo layout
152         Intent ListSong = new Intent(getApplicationContext(),
Main2Activity.class);
153         startActivity(ListSong);
154         //Compartir variables con el segundo layout
155         Intent intent = new Intent(MainActivity.this, Main2Activity.class
);
156         intent.putExtra("nombre_usuario", editText3.getText().toString());
157
158         intent.putExtra("nombre_alumno", datos[1]);
159         startActivity(intent);
160     }
161 }
162 }

```

layout

```

1
2 <?xml version="1.0" encoding="utf-8"?>
3 <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.
android.com/apk/res/android"
4 xmlns:app="http://schemas.android.com/apk/res-auto"
5 xmlns:tools="http://schemas.android.com/tools"
6 android:layout_width="match_parent"
7 android:layout_height="match_parent"
8
9 tools:context="com.example.ramon.clienteandroid.MainActivity">
10
11
12 <TextView
13     android:id="@+id/textView2"
14     android:layout_width="288dp"
15     android:layout_height="27dp"
16     android:layout_marginBottom="8dp"
17     android:layout_marginEnd="8dp"
18     android:layout_marginLeft="8dp"
19     android:layout_marginRight="8dp"
20     android:layout_marginStart="8dp"
21     android:layout_marginTop="20dp"
22     android:text="Nombre de Usuario"
23     android:textAppearance="@style/TextAppearance.AppCompat.Body2"

```

```
24 app:layout_constraintBottom_toBottomOf="parent"
25 app:layout_constraintHorizontal_bias="0.495"
26 app:layout_constraintLeft_toLeftOf="parent"
27 app:layout_constraintRight_toRightOf="parent"
28 app:layout_constraintTop_toTopOf="parent"
29 app:layout_constraintVertical_bias="0.117" />
30
31 <EditText
32 android:id="@+id/editText3"
33 android:layout_width="289dp"
34 android:layout_height="44dp"
35 android:layout_marginBottom="8dp"
36 android:layout_marginEnd="8dp"
37 android:layout_marginLeft="8dp"
38 android:layout_marginRight="8dp"
39 android:layout_marginStart="8dp"
40 android:layout_marginTop="8dp"
41 android:ems="10"
42 android:inputType="textPersonName"
43 app:layout_constraintBottom_toBottomOf="parent"
44 app:layout_constraintLeft_toLeftOf="parent"
45 app:layout_constraintRight_toRightOf="parent"
46 app:layout_constraintTop_toBottomOf="@+id/textView2"
47 app:layout_constraintVertical_bias="0.002" />
48
49 <TextView
50 android:id="@+id/textView4"
51 android:layout_width="298dp"
52 android:layout_height="29dp"
53 android:layout_marginBottom="8dp"
54 android:layout_marginEnd="8dp"
55 android:layout_marginLeft="8dp"
56 android:layout_marginRight="8dp"
57 android:layout_marginStart="8dp"
58 android:layout_marginTop="8dp"
59 android:text="Contraseña"
60 android:textAppearance="@style/TextAppearance.AppCompat.Body2"
61 app:layout_constraintBottom_toBottomOf="parent"
62 app:layout_constraintHorizontal_bias="0.556"
63 app:layout_constraintLeft_toLeftOf="parent"
64 app:layout_constraintRight_toRightOf="parent"
65 app:layout_constraintTop_toBottomOf="@+id/editText3"
66 app:layout_constraintVertical_bias="0.0" />
67
68 <EditText
69 android:id="@+id/editText4"
70 android:layout_width="287dp"
71 android:layout_height="52dp"
72 android:layout_marginBottom="8dp"
73 android:layout_marginEnd="8dp"
74 android:layout_marginLeft="8dp"
75 android:layout_marginRight="8dp"
76 android:layout_marginStart="8dp"
77 android:layout_marginTop="8dp"
78 android:ems="10"
79 android:inputType="textPassword"
80 app:layout_constraintBottom_toBottomOf="parent"
```

```

81     app:layout_constraintLeft_toLeftOf="parent"
82     app:layout_constraintRight_toRightOf="parent"
83     app:layout_constraintTop_toBottomOf="@+id/textView4"
84     app:layout_constraintVertical_bias="0.0" />
85
86     <Button
87         android:id="@+id/button3"
88         style="@android:style/Widget.Button.Inset"
89         android:layout_width="208dp"
90         android:layout_height="65dp"
91         android:layout_marginBottom="9dp"
92         android:layout_marginEnd="8dp"
93         android:layout_marginLeft="8dp"
94         android:layout_marginRight="8dp"
95         android:layout_marginStart="8dp"
96         android:layout_marginTop="64dp"
97         android:elevation="13dp"
98         android:text="ACCEDER"
99         android:textAppearance="@style/TextAppearance.AppCompat.Display1"
100        app:layout_constraintBottom_toBottomOf="parent"
101        app:layout_constraintHorizontal_bias="0.497"
102        app:layout_constraintLeft_toLeftOf="parent"
103        app:layout_constraintRight_toRightOf="parent"
104        app:layout_constraintTop_toBottomOf="@+id/editText4"
105        app:layout_constraintVertical_bias="0.0" />
106
107
108 </android.support.constraint.ConstraintLayout>

```

8.3.2 Pantalla de listado de asignaturas

MainActivity

```

1  package com.example.ramon.clienteandroid;
2  import android.app.ProgressDialog;
3  import android.content.Context;
4  import android.content.Intent;
5  import android.os.AsyncTask;
6  import android.support.v7.app.AppCompatActivity;
7  import android.os.Bundle;
8  import android.util.Log;
9  import android.view.View;
10 import android.widget.AdapterView;
11 import android.widget.AdapterView;
12 import android.widget.ArrayAdapter;
13 import android.widget.Button;
14 import android.widget.EditText;
15 import android.widget.ListAdapter;
16 import android.widget.ListView;
17 import android.widget.Toast;
18 import java.io.IOException;
19 import java.io.InputStream;
20 import java.io.OutputStream;
21 import java.net.InetAddress;
22 import java.net.Socket;
23 import java.net.UnknownHostException;
24 import java.util.ArrayList;

```

```

24
25
26 public class Main2Activity extends AppCompatActivity {
27
28     private EditText editText3;
29     private ListView lista;
30     private ArrayAdapter<String> adaptador;
31     private ArrayList<String> arrayList;
32     private Context context = this;
33     String nombre_usuario;
34     String nombre_alumno;
35     ArrayList<String> ejemplo = new ArrayList<String>();
36
37     private static final int SERVERPORT = 5000;
38     private static final String ADDRESS = "192.168.112.2";
39
40     @Override
41     protected void onCreate(Bundle savedInstanceState) {
42
43         super.onCreate(savedInstanceState);
44         setContentView(R.layout.activity_main2);
45
46         // Objetos de pantalla
47         lista = (ListView) findViewById(R.id.listView);
48         editText3 = ((EditText) findViewById(R.id.editText6));
49         arrayList = new ArrayList<String>();
50
51         //Recibir variables de otro layout
52         nombre_usuario = getIntent().getExtras().getString("nombre_usuario");
53         nombre_alumno = getIntent().getExtras().getString("nombre_alumno");
54
55         editText3.setText(nombre_usuario+" "+nombre_alumno);
56
57         //Enviar petición al servidor
58         MyATaskCliente myATaskYW = new MyATaskCliente();
59         myATaskYW.execute("MAC20"+nombre_usuario);
60     }
61
62     class MyATaskCliente extends AsyncTask<String,Void,String> {
63
64         ProgressDialog progressDialog;
65
66         @Override
67         protected void onPreExecute()
68         {
69             super.onPreExecute();
70             progressDialog = new ProgressDialog(context);
71             progressDialog.setCanceledOnTouchOutside(false);
72             progressDialog.setTitle("Connecting to server");
73             progressDialog.setMessage("Please wait...");
74             progressDialog.show();
75         }
76
77         @Override
78         protected String doInBackground(String... values){
79             try {
80                 InetAddress serverAddr = InetAddress.getByName(ADDRESS);

```



```

81         Log.i("I/TCP Client", "Connecting...");
82         Socket socket = new Socket(serverAddr, SERVERPORT);
83         Log.i("I/TCP Client", "Connected to server");
84
85         Log.i("I/TCP Client", "Send data to server");
86         PrintStream output = new PrintStream(socket.getOutputStream());
87         String request = values[0];
88         output.println(request);
89
90         Log.i("I/TCP Client", "Received data to server");
91         InputStream stream = socket.getInputStream();
92         byte[] lenBytes = new byte[256];
93         int i;
94         String received = "";
95         while ((stream.read(lenBytes,0,256)) != -1) {
96             String receivedaux = new String(lenBytes, "UTF-8");
97             i = receivedaux.indexOf(' ');
98             if (i != -1) received += receivedaux.substring(0,i+1);
99             else received += receivedaux;
100        }
101
102
103         Log.i("I/TCP Client", "Received " + received);
104         Log.i("I/TCP Client", "");
105         socket.close();
106         return received;
107     } catch (UnknownHostException ex) {
108         Log.e("E/TCP Client", "" + ex.getMessage());
109         return ex.getMessage();
110     } catch (IOException ex) {
111         Log.e("E/TCP Client", "" + ex.getMessage());
112         return ex.getMessage();
113     }
114 }
115
116 @Override
117 protected void onPostExecute(String value) {
118     progressDialog.dismiss();
119
120     int Index, fromIndex = 0;
121     while (true) {
122         Index = value.indexOf(' ', fromIndex);
123         if (Index == -1) {
124             ejemplo.add(value.substring(fromIndex));
125             break;
126         }
127         ejemplo.add(value.substring(fromIndex, Index));
128         fromIndex = Index + 1;
129     }
130
131     adaptador = new ArrayAdapter(context, android.R.layout.
simple_list_item_1, ejemplo);
132     lista.setAdapter(adaptador);
133     lista.setOnItemClickListener(new AdapterView.OnItemClickListener() {
134         @Override
135         public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {

```

8. ANEXOS

```
136
137         Intent intent = new Intent(Main2Activity.this, Main26Activity
        .class);
138
139         //Enviar parametros a la siguiente pantalla
140         intent.putExtra("posicion", ejemplo.get(position));
141         intent.putExtra("nombre_usuario", nombre_usuario);
142         intent.putExtra("nombre_alumno", nombre_alumno);
143
144         startActivity(intent);
145     }
146 });
147 }
148 }
149 }
```

layout

```
1
2 <?xml version="1.0" encoding="utf-8"?>
3 <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.
    android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     tools:context="com.example.ramon.clienteandroid.Main2Activity">
9
10 <EditText
11     android:id="@+id/editText6"
12     android:layout_width="0dp"
13     android:layout_height="42dp"
14     android:layout_marginBottom="8dp"
15     android:layout_marginEnd="8dp"
16     android:layout_marginLeft="8dp"
17     android:layout_marginRight="8dp"
18     android:layout_marginStart="8dp"
19     android:layout_marginTop="8dp"
20     android:editable="false"
21     android:ems="10"
22     android:enabled="false"
23     android:inputType="textPersonName"
24     android:text="Usuario"
25     android:textColor="#00F"
26     android:focusable="false"
27     app:layout_constraintBottom_toBottomOf="parent"
28     app:layout_constraintHorizontal_bias="0.0"
29     app:layout_constraintLeft_toLeftOf="parent"
30     app:layout_constraintRight_toRightOf="parent"
31     app:layout_constraintTop_toTopOf="parent"
32     app:layout_constraintVertical_bias="0.013" />
33
34 <ListView
35     android:id="@+id/listView"
36     android:layout_width="wrap_content"
37     android:layout_height="wrap_content"
```

```

38     android:layout_marginTop="8dp"
39     android:focusable="false"
40     app:layout_constraintTop_toBottomOf="@+id/editText6"
41     tools:layout_editor_absoluteX="8dp" />
42
43 </android.support.constraint.ConstraintLayout>

```

8.3.3 Pantalla de selección del examen del año actual

MainActivity

```

1  package com.example.ramon.clienteandroid;
2
3  import android.app.AlertDialog;
4  import android.content.Context;
5  import android.content.Intent;
6  import android.os.AsyncTask;
7  import android.support.v7.app.AppCompatActivity;
8  import android.os.Bundle;
9  import android.util.Log;
10 import android.view.View;
11 import android.widget.AdapterView;
12 import android.widget.AdapterView.OnItemClickListener;
13 import android.widget.EditText;
14 import android.widget.ListView;
15 import android.widget.Spinner;
16
17 import java.io.IOException;
18 import java.io.InputStream;
19 import java.io.OutputStream;
20 import java.net.InetAddress;
21 import java.net.Socket;
22 import java.net.UnknownHostException;
23 import java.util.ArrayList;
24 import java.util.List;
25 import java.util.StringTokenizer;
26
27 public class Main22Activity extends AppCompatActivity {
28
29     private EditText editText, editText2, editText3;
30     private Context context = this;
31     private ListView lista;
32     private ArrayAdapter<String> adaptador;
33     String nombre_asign;
34     String nombre_usuario;
35     String nombre_alumno;
36     String[] datos;
37     ArrayList<String> ejemplo = new ArrayList<String>();
38     ArrayList<String> ejemplo2 = new ArrayList<String>();
39     ArrayList<String> ejemplo3 = new ArrayList<String>();
40
41     private static final int SERVERPORT = 5000;
42     private static final String ADDRESS = "192.168.112.2";
43
44     @Override
45     protected void onCreate(Bundle savedInstanceState) {

```

```

46     super. onCreate (savedInstanceState);
47     setContentView (R.layout.activity_main22);
48
49     lista = (ListView) findViewById (R.id.listView);
50     editText = ((EditText) findViewById (R.id.editText2));
51     editText2 = ((EditText) findViewById (R.id.editText5));
52     editText3 = ((EditText) findViewById (R.id.editText15));
53
54     nombre_asign = getIntent ().getExtras ().getString ("posicion");
55     nombre_usuario = getIntent ().getExtras ().getString ("nombre_usuario");
56     nombre_alumno = getIntent ().getExtras ().getString ("nombre_alumno");
57
58     editText.setText (nombre_asign);
59     editText3.setText (nombre_usuario+" "+nombre_alumno);
60     //extraemos el codigo de la asignatura
61     String asignatura=nombre_asign.substring (0,5);
62
63     Main22Activity.MyATaskCliente myATasUSER = new Main22Activity.
MyATaskCliente ();
64     myATasUSER.execute ("Mac22#"+asignatura+"#" +nombre_usuario);
65 }
66 class MyATaskCliente extends AsyncTask<String,Void,String> {
67
68     ProgressDialog progressDialog;
69
70     @Override
71     protected void onPreExecute ()
72     {
73         super.onPreExecute ();
74         progressDialog = new ProgressDialog (context);
75         progressDialog.setCanceledOnTouchOutside (false);
76         progressDialog.setTitle ("Connecting to server");
77         progressDialog.setMessage ("Please wait...");
78         progressDialog.show ();
79     }
80
81     @Override
82     protected String doInBackground (String... values) {
83
84         try {
85             InetAddress serverAddr = InetAddress.getByName (ADDRESS);
86             Log.i ("I/TCP Client", "Connecting...");
87             Socket socket = new Socket (serverAddr, SERVERPORT);
88             Log.i ("I/TCP Client", "Connected to server");
89
90             Log.i ("I/TCP Client", "Send data to server");
91             PrintStream output = new PrintStream (socket.getOutputStream ());
92             String request = values [0];
93             output.println (request);
94             Log.i ("I/TCP Client", "Received data to server");
95             InputStream stream = socket.getInputStream ();
96             byte [] lenBytes = new byte [256];
97             int i;
98             String received = "";
99             while ((stream.read (lenBytes,0,256)) != -1) {
100                 String receivedaux = new String (lenBytes, "UTF-8");
101                 i = receivedaux.indexOf (' ');

```

```

102         if (i != -1) received += receivedaux.substring(0,i+1);
103         else         received += receivedaux;
104     }
105
106     Log.i("I/TCP Client", "Received " + received);
107     Log.i("I/TCP Client", "");
108
109     socket.close();
110     return received;
111 } catch (UnknownHostException ex) {
112     Log.e("E/TCP Client", "" + ex.getMessage());
113     return ex.getMessage();
114 } catch (IOException ex) {
115     Log.e("E/TCP Client", "" + ex.getMessage());
116     return ex.getMessage();
117 }
118 }
119
120 @Override
121 protected void onPostExecute(String value) {
122
123     progressDialog.dismiss();
124     datos = value.split("#");
125
126     if (datos[0].length() > 1) {
127         editText2.setText(datos[0].substring(0,datos[0].length()-1)); //
128         PROFESORES
129     } else {
130         editText2.setText("No hay profesor asociado a la asignatura");
131     }
132
133     String valor;
134     if (datos[1].length() > 2) {
135         valor=datos[1].substring(1,datos[1].length()-1);
136         datos[2]=datos[2].substring(1,datos[2].length());
137     } else {
138         valor = "No has realizado ningún examen";
139     }
140
141     int i = 0, Index, fromIndex = 0,Indexdate=0,fromIndexdate=0;
142     if (!valor.equals("No has realizado ningún examen")) {
143         while (true) {
144             Index = valor.indexOf(', ', fromIndex);
145             Indexdate = datos[2].indexOf(', ', fromIndexdate);
146             if (Index == -1) {
147                 ejemplo.add(valor.substring(fromIndex));
148                 ejemplo2.add(valor.substring(fromIndex) + "- Fecha examen
149                 : " + datos[2].substring(fromIndexdate));
150                 ejemplo3.add(datos[2].substring(fromIndexdate));
151                 break;
152             }
153             ejemplo.add(valor.substring(fromIndex, Index));
154             ejemplo2.add(valor.substring(fromIndex, Index) + "- Fecha
155             examen:" + datos[2].substring(fromIndexdate, Indexdate));
156             ejemplo3.add(datos[2].substring(fromIndexdate, Indexdate));
157             fromIndex = Index + 1;
158             fromIndexdate = Indexdate + 1;

```

8. ANEXOS

```
156     }
157     } else {
158         ejemplo2.add("No has realizado ningún examen");
159     }
160
161     adaptador = new ArrayAdapter(context, android.R.layout.
simple_list_item_1, ejemplo2);
162     lista.setAdapter(adaptador);
163     if (!valor.equals("No has realizado ningún examen")) {
164         lista.setOnItemClickListener(new AdapterView.OnItemClickListener
() {
165             @Override
166             public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {
167
168                 Intent intent = new Intent(Main22Activity.this,
Main23Activity.class);
169
170                 intent.putExtra("posicion", ejemplo.get(position));
171                 intent.putExtra("fecha", ejemplo3.get(position));
172                 intent.putExtra("nombre_usuario", nombre_usuario);
173                 intent.putExtra("nombre_alumno", nombre_alumno);
174                 intent.putExtra("asignatura", nombre_asign. substring(0,5)
);
175
176                 startActivity(intent);
177             }
178         });
179     }
180 }
181 }
182 }
183 }
```

layout

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.
android.com/apk/res/android"
3 xmlns:app="http://schemas.android.com/apk/res-auto"
4 xmlns:tools="http://schemas.android.com/tools"
5 android:layout_width="match_parent"
6 android:layout_height="match_parent"
7 tools:context="com.example.ramon.clienteandroid.Main22Activity">
8
9 <TextView
10     android:id="@+id/textView"
11     android:layout_width="wrap_content"
12     android:layout_height="wrap_content"
13     android:layout_marginBottom="9dp"
14     android:layout_marginEnd="8dp"
15     android:layout_marginLeft="8dp"
16     android:layout_marginRight="8dp"
17     android:layout_marginStart="8dp"
18     android:layout_marginTop="8dp"
19     android:text="PROFESORES">
```

```
20     android:textAppearance="@style/TextAppearance.AppCompat.Body2"
21     app:layout_constraintBottom_toBottomOf="parent"
22     app:layout_constraintHorizontal_bias="0.0"
23     app:layout_constraintLeft_toLeftOf="parent"
24     app:layout_constraintRight_toRightOf="parent"
25     app:layout_constraintTop_toBottomOf="@+id/editText2"
26     app:layout_constraintVertical_bias="0.0" />
27
28 <EditText
29     android:id="@+id/editText2"
30     android:layout_width="0dp"
31     android:layout_height="40dp"
32     android:layout_marginBottom="8dp"
33     android:layout_marginEnd="8dp"
34     android:layout_marginLeft="8dp"
35     android:layout_marginRight="8dp"
36     android:layout_marginStart="8dp"
37     android:layout_marginTop="8dp"
38     android:editable="false"
39     android:ems="10"
40     android:enabled="false"
41     android:inputType="textPersonName"
42     android:textColor="#05F"
43     android:focusable="false"
44     app:layout_constraintBottom_toBottomOf="parent"
45     app:layout_constraintHorizontal_bias="0.0"
46     app:layout_constraintLeft_toLeftOf="parent"
47     app:layout_constraintRight_toRightOf="parent"
48     app:layout_constraintTop_toBottomOf="@+id/textView5"
49     app:layout_constraintVertical_bias="0.004" />
50
51 <TextView
52     android:id="@+id/textView3"
53     android:layout_width="wrap_content"
54     android:layout_height="wrap_content"
55     android:layout_marginBottom="8dp"
56     android:layout_marginEnd="8dp"
57     android:layout_marginLeft="8dp"
58     android:layout_marginRight="8dp"
59     android:layout_marginStart="8dp"
60     android:layout_marginTop="8dp"
61     android:text="EXAMENES"
62     android:textAppearance="@style/TextAppearance.AppCompat.Body2"
63     app:layout_constraintBottom_toBottomOf="parent"
64     app:layout_constraintHorizontal_bias="0.0"
65     app:layout_constraintLeft_toLeftOf="parent"
66     app:layout_constraintRight_toRightOf="parent"
67     app:layout_constraintTop_toBottomOf="@+id/editText5"
68     app:layout_constraintVertical_bias="0.0" />
69
70 <TextView
71     android:id="@+id/textView5"
72     android:layout_width="wrap_content"
73     android:layout_height="wrap_content"
74     android:layout_marginBottom="8dp"
75     android:layout_marginEnd="6dp"
76     android:layout_marginLeft="8dp"
```

```
77     android:layout_marginRight="6dp"
78     android:layout_marginStart="8dp"
79     android:layout_marginTop="8dp"
80     android:text="ASIGNATURA"
81     android:textAppearance="@style/TextAppearance.AppCompat.Body2"
82     app:layout_constraintBottom_toBottomOf="parent"
83     app:layout_constraintHorizontal_bias="0.0"
84     app:layout_constraintLeft_toLeftOf="parent"
85     app:layout_constraintRight_toRightOf="parent"
86     app:layout_constraintTop_toBottomOf="@+id/editText15"
87     app:layout_constraintVertical_bias="0.0" />
88
89 <EditText
90     android:id="@+id/editText5"
91     android:layout_width="0dp"
92     android:layout_height="43dp"
93     android:layout_marginBottom="8dp"
94     android:layout_marginEnd="8dp"
95     android:layout_marginLeft="8dp"
96     android:layout_marginRight="8dp"
97     android:layout_marginStart="8dp"
98     android:layout_marginTop="8dp"
99     android:editable="false"
100    android:ems="10"
101    android:enabled="false"
102    android:focusable="false"
103    android:inputType="textPersonName"
104    app:layout_constraintBottom_toBottomOf="parent"
105    app:layout_constraintHorizontal_bias="0.0"
106    app:layout_constraintLeft_toLeftOf="parent"
107    app:layout_constraintRight_toRightOf="parent"
108    app:layout_constraintTop_toBottomOf="@+id/textView"
109    app:layout_constraintVertical_bias="0.0" />
110
111 <EditText
112     android:id="@+id/editText15"
113     android:layout_width="0dp"
114     android:layout_height="wrap_content"
115     android:layout_marginBottom="9dp"
116     android:layout_marginEnd="8dp"
117     android:layout_marginLeft="8dp"
118     android:layout_marginRight="8dp"
119     android:layout_marginStart="8dp"
120     android:layout_marginTop="8dp"
121     android:editable="false"
122     android:ems="10"
123     android:focusable="false"
124     android:enabled="false"
125     android:inputType="textPersonName"
126     android:textColor="#00F"
127     app:layout_constraintBottom_toBottomOf="parent"
128     app:layout_constraintHorizontal_bias="0.0"
129     app:layout_constraintLeft_toLeftOf="parent"
130     app:layout_constraintRight_toRightOf="parent"
131     app:layout_constraintTop_toTopOf="parent"
132     app:layout_constraintVertical_bias="0.013" />
133
```



```

134 <ListView
135     android:id="@+id/listView"
136     android:layout_width="wrap_content"
137     android:layout_height="wrap_content"
138     android:layout_marginLeft="8dp"
139     android:layout_marginStart="8dp"
140     android:layout_marginTop="7dp"
141     android:focusable="false"
142     app:layout_constraintLeft_toLeftOf="parent"
143     app:layout_constraintTop_toBottomOf="@+id/textView3" />
144
145 </android.support.constraint.ConstraintLayout>

```

8.3.4 Pantalla de selección de preguntas

MainActivity

```

1  package com.example.ramon.clienteandroid;
2
3  import android.app.AlertDialog;
4  import android.content.Context;
5  import android.content.Intent;
6  import android.os.AsyncTask;
7  import android.support.v7.app.AppCompatActivity;
8  import android.os.Bundle;
9  import android.util.Log;
10 import android.view.View;
11 import android.widget.AdapterView;
12 import android.widget.AdapterView.OnItemClickListener;
13 import android.widget.ArrayAdapter;
14 import android.widget.EditText;
15 import android.widget.ListView;
16
17 import java.io.IOException;
18 import java.io.InputStream;
19 import java.io.OutputStream;
20 import java.net.InetAddress;
21 import java.net.Socket;
22 import java.net.UnknownHostException;
23 import java.util.ArrayList;
24 import java.util.StringTokenizer;
25
26 public class Main23Activity extends AppCompatActivity {
27
28     private EditText editText, editText1, editText2, editText3, editText4;
29     private ListView lista;
30     private Context context = this;
31     private ArrayAdapter<String> adaptador;
32     String nombre_usuario, nombre_alumno, fecha;
33     String[] datos;
34     ArrayList<String> ejemplo = new ArrayList<String>();
35     ArrayList<String> ejemplo2 = new ArrayList<String>();
36
37     private static final int SERVERPORT = 5000;
38     private static final String ADDRESS = "192.168.112.2";
39

```

```

40  @Override
41  protected void onCreate(Bundle savedInstanceState) {
42      super.onCreate(savedInstanceState);
43      setContentView(R.layout.activity_main23);
44
45      editText = ((EditText) findViewById(R.id.editText));
46      editText1 = ((EditText) findViewById(R.id.editText7));
47      editText2 = ((EditText) findViewById(R.id.editText8));
48      editText3 = ((EditText) findViewById(R.id.editText16));
49      editText4 = ((EditText) findViewById(R.id.editText29));
50
51      lista = (ListView) findViewById(R.id.listView);
52
53      final String nom_exa = getIntent().getExtras().getString("posicion");
54      final String asign = getIntent().getExtras().getString("asignatura");
55
56      nombre_usuario = getIntent().getExtras().getString("nombre_usuario");
57      nombre_alumno = getIntent().getExtras().getString("nombre_alumno");
58      fecha = getIntent().getExtras().getString("fecha");
59
60      editText.setText(nom_exa);
61      editText1.setText(fecha.trim());
62      editText3.setText(nombre_usuario+" "+nombre_alumno);
63
64      Main23Activity.MyATaskCliente myATasUSER = new Main23Activity.
MyATaskCliente();
65      myATasUSER.execute("MAC23#" + nom_exa + "#" + asign + "#" + fecha.trim() + "#" +
nombre_usuario);
66  }
67
68
69  class MyATaskCliente extends AsyncTask<String,Void,String> {
70
71      ProgressDialog progressDialog;
72
73      private String REM_HTMLTags(String Data)
74      {
75          Data = Data.replaceAll("<p>", "");
76          Data = Data.replaceAll("</p>", "");
77          return Data;
78      }
79
80      @Override
81      protected void onPreExecute()
82      {
83          super.onPreExecute();
84          progressDialog = new ProgressDialog(context);
85          progressDialog.setCanceledOnTouchOutside(false);
86          progressDialog.setTitle("Connecting to server");
87          progressDialog.setMessage("Please wait...");
88          progressDialog.show();
89      }
90
91      @Override
92      protected String doInBackground(String... values){
93
94          try {

```

```

95     InetAddress serverAddr = InetAddress.getByName(ADDRESS);
96     Log.i("I/TCP Client", "Connecting...");
97     Socket socket = new Socket(serverAddr, SERVERPORT);
98     Log.i("I/TCP Client", "Connected to server");
99
100    Log.i("I/TCP Client", "Send data to server");
101    PrintStream output = new PrintStream(socket.getOutputStream());
102    String request = values[0];
103    output.println(request);
104
105    Log.i("I/TCP Client", "Received data to server");
106    InputStream stream = socket.getInputStream();
107    byte[] lenBytes = new byte[256];
108    int i;
109    String received = "";
110    while ((stream.read(lenBytes,0,256)) != -1) {
111        String receivedaux = new String(lenBytes, "UTF-8");
112        i = receivedaux.indexOf(' ');
113        if (i != -1) received += receivedaux.substring(0,i+1);
114        else received += receivedaux;
115    }
116
117    Log.i("I/TCP Client", "Received " + received);
118    Log.i("I/TCP Client", "");
119
120    socket.close();
121    return received;
122 } catch (UnknownHostException ex) {
123     Log.e("E/TCP Client", "" + ex.getMessage());
124     return ex.getMessage();
125 } catch (IOException ex) {
126     Log.e("E/TCP Client", "" + ex.getMessage());
127     return ex.getMessage();
128 }
129 }
130
131
132 @Override
133 protected void onPostExecute(String value) {
134     progressDialog.dismiss();
135     datos = value.split("#");
136
137     if(datos[0].length()>2) {
138         datos[0] = datos[0].substring(0, datos[0].length() - 1);
139     } else {
140         datos[0] ="no hay preguntas asociadas al examen";
141     }
142
143     if(datos[1].length() >= 2) datos[1]=datos[1].substring(1,datos[1].
length()-1);
144     if(datos[1].equals("NULL")) datos[1]="no hay fecha de revisión del
examen";
145
146     if(datos[2].length()>=2) {
147         datos[2]=REM_HTMLTags(datos[2]);
148         datos[2]=datos[2].substring(1,datos[2].length());
149     }

```

8. ANEXOS

```
150         datos[2]=datos [2]. trim ();
151         if (datos [2]. equals ("NULL"))   datos[2]="no hay comentarios del examen
";
152
153
154         editText2.setText (datos [1]); //fecha de revision
155         editText4.setText (datos [2]); //comentarios generales
156
157         String valor=datos [0];
158
159         int i = 1, Index, fromIndex = 0;
160         while (true) {
161             Index = valor.indexOf ('',',', fromIndex);
162             if (Index == -1) {
163                 ejemplo.add (valor.substring (fromIndex));
164                 ejemplo2.add ("Pregunta #" + i); i++;
165                 break;
166             }
167             ejemplo.add (valor.substring (fromIndex, Index));
168             ejemplo2.add ("Pregunta #" + i); i++;
169             fromIndex = Index + 1;
170         }
171
172         adaptador = new ArrayAdapter (context, android.R.layout.
simple_list_item_1, ejemplo2);
173         lista.setAdapter (adaptador);
174         if (valor != "no hay preguntas asociadas al examen") {
175             lista.setOnItemClickListener (new AdapterView.OnItemClickListener
() {
176                 @Override
177                 public void onItemClick (AdapterView<?> parent, View view, int
position, long id) {
178
179                     Intent intent = new Intent (Main23Activity.this,
Main24Activity.class);
180
181                     intent.putExtra ("posicion", ejemplo.get (position));
182                     intent.putExtra ("nombre_examen", editText.getText ().
toString ());
183                     intent.putExtra ("nombre_usuario", nombre_usuario);
184                     intent.putExtra ("nombre_alumno", nombre_alumno);
185                     intent.putExtra ("fecha_examen", fecha);
186
187                     startActivity (intent);
188                 }
189             });
190         }
191     }
192 }
193 }
```

layout

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.
android.com/apk/res/android"
```

```
3 xmlns:app="http://schemas.android.com/apk/res-auto"
4 xmlns:tools="http://schemas.android.com/tools"
5 android:layout_width="match_parent"
6 android:layout_height="match_parent"
7 tools:context="com.example.ramon.clienteandroid.Main23Activity">
8
9 <RelativeLayout
10 android:layout_width="0dp"
11 android:layout_height="4dp"
12 tools:layout_editor_absoluteX="8dp"
13 tools:layout_editor_absoluteY="8dp">
14
15 </RelativeLayout>
16
17 <EditText
18     android:id="@+id/editText16"
19     android:layout_width="0dp"
20     android:layout_height="wrap_content"
21     android:layout_marginBottom="10dp"
22     android:layout_marginEnd="8dp"
23     android:layout_marginLeft="8dp"
24     android:layout_marginRight="8dp"
25     android:layout_marginStart="8dp"
26     android:layout_marginTop="8dp"
27     android:editable="false"
28     android:ems="10"
29     android:enabled="false"
30     android:inputType="textPersonName"
31     android:textColor="#00F"
32     android:focusable="false"
33     app:layout_constraintBottom_toBottomOf="parent"
34     app:layout_constraintHorizontal_bias="0.0"
35     app:layout_constraintLeft_toLeftOf="parent"
36     app:layout_constraintRight_toRightOf="parent"
37     app:layout_constraintTop_toTopOf="parent"
38     app:layout_constraintVertical_bias="0.013" />
39
40 <TextView
41     android:id="@+id/textView6"
42     android:layout_width="wrap_content"
43     android:layout_height="wrap_content"
44     android:layout_marginBottom="8dp"
45     android:layout_marginEnd="8dp"
46     android:layout_marginLeft="8dp"
47     android:layout_marginRight="8dp"
48
49     android:layout_marginStart="8dp"
50     android:layout_marginTop="8dp"
51     android:text="PREGUNTAS"
52     android:textAppearance="@style/TextAppearance.AppCompat.Body2"
53     app:layout_constraintBottom_toBottomOf="parent"
54     app:layout_constraintHorizontal_bias="0.0"
55     app:layout_constraintLeft_toLeftOf="parent"
56     app:layout_constraintRight_toRightOf="parent"
57     app:layout_constraintTop_toBottomOf="@+id/editText29"
58     app:layout_constraintVertical_bias="0.0" />
59
```

```
60 <TextView
61     android:id="@+id/textView9"
62     android:layout_width="wrap_content"
63     android:layout_height="wrap_content"
64     android:layout_marginBottom="8dp"
65     android:layout_marginEnd="8dp"
66     android:layout_marginLeft="8dp"
67     android:layout_marginRight="8dp"
68     android:layout_marginStart="8dp"
69     android:layout_marginTop="8dp"
70
71     android:text="EXAMEN"
72     android:textAppearance="@style/TextAppearance.AppCompat.Body2"
73     app:layout_constraintBottom_toBottomOf="parent"
74     app:layout_constraintHorizontal_bias="0.0"
75     app:layout_constraintLeft_toLeftOf="parent"
76     app:layout_constraintRight_toRightOf="parent"
77     app:layout_constraintTop_toBottomOf="@+id/editText16"
78     app:layout_constraintVertical_bias="0.0" />
79
80 <EditText
81     android:id="@+id/editText"
82     android:layout_width="0dp"
83     android:layout_height="wrap_content"
84     android:layout_marginBottom="8dp"
85     android:layout_marginEnd="8dp"
86     android:layout_marginLeft="8dp"
87     android:layout_marginRight="8dp"
88     android:layout_marginStart="8dp"
89     android:layout_marginTop="8dp"
90     android:editable="false"
91     android:ems="10"
92     android:enabled="false"
93     android:inputType="textPersonName"
94     android:textColor="#05F"
95     android:focusable="false"
96     app:layout_constraintBottom_toBottomOf="parent"
97     app:layout_constraintHorizontal_bias="0.0"
98     app:layout_constraintLeft_toLeftOf="parent"
99     app:layout_constraintRight_toRightOf="parent"
100    app:layout_constraintTop_toBottomOf="@+id/textView9"
101    app:layout_constraintVertical_bias="0.0" />
102
103 <TextView
104     android:id="@+id/textView7"
105     android:layout_width="wrap_content"
106     android:layout_height="wrap_content"
107     android:layout_marginBottom="8dp"
108     android:layout_marginEnd="8dp"
109     android:layout_marginLeft="8dp"
110     android:layout_marginRight="8dp"
111
112     android:layout_marginStart="8dp"
113     android:layout_marginTop="8dp"
114     android:text="FECHA DE REVISIÓN"
115     android:textAppearance="@style/TextAppearance.AppCompat.Body2"
116     app:layout_constraintBottom_toBottomOf="parent"
```

```
117     app:layout_constraintHorizontal_bias="0.0"
118     app:layout_constraintLeft_toRightOf="@+id/editText7"
119     app:layout_constraintRight_toRightOf="parent"
120     app:layout_constraintTop_toBottomOf="@+id/editText"
121     app:layout_constraintVertical_bias="0.002" />
122
123 <TextView
124     android:id="@+id/textView8"
125     android:layout_width="wrap_content"
126     android:layout_height="wrap_content"
127     android:layout_marginBottom="8dp"
128     android:layout_marginEnd="8dp"
129     android:layout_marginLeft="8dp"
130     android:layout_marginRight="8dp"
131
132     android:layout_marginStart="8dp"
133     android:layout_marginTop="8dp"
134     android:text="FECHA DEL EXAMEN"
135     android:textAppearance="@style/TextAppearance.AppCompat.Body2"
136     app:layout_constraintBottom_toBottomOf="parent"
137     app:layout_constraintHorizontal_bias="0.0"
138     app:layout_constraintLeft_toLeftOf="parent"
139     app:layout_constraintRight_toRightOf="parent"
140     app:layout_constraintTop_toBottomOf="@+id/editText"
141     app:layout_constraintVertical_bias="0.0" />
142
143 <EditText
144     android:id="@+id/editText7"
145     android:layout_width="195dp"
146     android:layout_height="wrap_content"
147     android:layout_marginBottom="8dp"
148     android:layout_marginEnd="8dp"
149     android:layout_marginLeft="8dp"
150     android:layout_marginRight="8dp"
151     android:layout_marginStart="8dp"
152     android:layout_marginTop="8dp"
153     android:editable="false"
154     android:ems="10"
155     android:enabled="false"
156     android:focusable="false"
157     android:inputType="textPersonName"
158     app:layout_constraintBottom_toBottomOf="parent"
159     app:layout_constraintHorizontal_bias="0.0"
160     app:layout_constraintLeft_toLeftOf="parent"
161     app:layout_constraintRight_toRightOf="parent"
162     app:layout_constraintTop_toBottomOf="@+id/textView8"
163     app:layout_constraintVertical_bias="0.0" />
164
165 <EditText
166     android:id="@+id/editText8"
167     android:layout_width="195dp"
168     android:layout_height="wrap_content"
169     android:layout_marginBottom="9dp"
170     android:layout_marginEnd="8dp"
171     android:layout_marginLeft="8dp"
172     android:layout_marginRight="8dp"
173     android:layout_marginStart="8dp"
```

```
174     android:layout_marginTop="8dp"
175     android:editable="false"
176     android:textColor="#F00"
177     android:ems="10"
178     android:enabled="false"
179     android:focusable="false"
180     android:inputType="textPersonName"
181     app:layout_constraintBottom_toBottomOf="parent"
182     app:layout_constraintHorizontal_bias="0.1"
183     app:layout_constraintLeft_toRightOf="@+id/editText7"
184     app:layout_constraintRight_toRightOf="parent"
185     app:layout_constraintTop_toBottomOf="@+id/textView7"
186     app:layout_constraintVertical_bias="0.0" />
187
188 <ListView
189     android:id="@+id/listView"
190     android:layout_width="wrap_content"
191     android:layout_height="wrap_content"
192     android:layout_marginTop="8dp"
193     android:focusable="false"
194     app:layout_constraintTop_toBottomOf="@+id/textView6"
195     tools:layout_editor_absoluteX="0dp" />
196
197 <TextView
198     android:id="@+id/textView25"
199     android:layout_width="wrap_content"
200     android:layout_height="wrap_content"
201     android:layout_marginBottom="8dp"
202     android:layout_marginEnd="8dp"
203     android:layout_marginLeft="8dp"
204     android:layout_marginRight="8dp"
205     android:layout_marginStart="8dp"
206     android:layout_marginTop="8dp"
207     android:focusable="false"
208     android:text="Comentarios generales"
209     android:textAppearance="@style/TextAppearance.AppCompat.Body2"
210     app:layout_constraintBottom_toBottomOf="parent"
211     app:layout_constraintHorizontal_bias="0.0"
212     app:layout_constraintLeft_toLeftOf="parent"
213     app:layout_constraintRight_toRightOf="parent"
214     app:layout_constraintTop_toBottomOf="@+id/editText7"
215     app:layout_constraintVertical_bias="0.0" />
216
217 <EditText
218     android:id="@+id/editText29"
219     android:layout_width="395dp"
220     android:layout_height="137dp"
221     android:layout_marginBottom="8dp"
222     android:layout_marginLeft="8dp"
223     android:layout_marginStart="8dp"
224     android:layout_marginTop="8dp"
225     android:editable="false"
226     android:ems="10"
227     android:enabled="false"
228     android:focusable="false"
229     android:inputType="textMultiLine"
230     app:layout_constraintBottom_toBottomOf="parent"
```



```

231     app:layout_constraintLeft_toLeftOf="parent"
232     app:layout_constraintTop_toBottomOf="@+id/textView25"
233     app:layout_constraintVertical_bias="0.0" />
234
235 </android.support.constraint.ConstraintLayout>

```

8.3.5 Pantalla de soluciones de una pregunta

MainActivity

```

1  package com.example.ramon.clienteandroid;
2
3  import android.app.ProgressDialog;
4  import android.content.ActivityNotFoundException;
5  import android.content.Context;
6  import android.content.Intent;
7  import android.net.Uri;
8  import android.os.AsyncTask;
9  import android.support.v7.app.AppCompatActivity;
10 import android.os.Bundle;
11 import android.util.Log;
12 import android.view.View;
13 import android.webkit.WebView;
14 import android.widget.AdapterView;
15 import android.widget.AdapterView.OnItemClickListener;
16 import android.widget.Button;
17 import android.widget.EditText;
18
19 import java.io.IOException;
20 import java.io.InputStream;
21 import java.io.OutputStream;
22 import java.net.InetAddress;
23 import java.net.Socket;
24 import java.net.UnknownHostException;
25 import java.util.StringTokenizer;
26
27 public class Main24Activity extends AppCompatActivity {
28
29     private EditText editText, editText1, editText2, editText3, editText4, editText5,
30         editText6, editText7, editText8;
31     private Button boton1, boton2, boton3, boton4;
32     private Context context = this;
33     String nombre_usuario, nombre_alumno, URL_solucion, URL_sol_alumn, URL_preg,
34         URL_preg_sol;
35     String[] separada;
36
37     private static final int SERVERPORT = 5000;
38     private static final String ADDRESS = "192.168.112.2";
39
40     @Override
41     protected void onCreate(Bundle savedInstanceState) {
42         super.onCreate(savedInstanceState);
43         setContentView(R.layout.activity_main24);
44
45         editText = ((EditText) findViewById(R.id.editText12));
46         editText1 = ((EditText) findViewById(R.id.editText10));

```

```

45     editText2 = ((EditText) findViewById(R.id.editText11));
46     editText3 = ((EditText) findViewById(R.id.editText9));
47     editText4 = ((EditText) findViewById(R.id.editText13));
48     editText5 = ((EditText) findViewById(R.id.editText18));
49     editText6 = ((EditText) findViewById(R.id.editText14));
50     editText7 = ((EditText) findViewById(R.id.editText28));
51     editText8 = ((EditText) findViewById(R.id.editText31));
52     boton1=((Button) findViewById(R.id.button4));
53     boton2=((Button) findViewById(R.id.button));
54     boton3=((Button) findViewById(R.id.button2));
55     boton4=((Button) findViewById(R.id.button9));
56
57     nombre_usuario =getIntent().getExtras().getString("nombre_usuario");
58     nombre_alumno =getIntent().getExtras().getString("nombre_alumno");
59     final String preg =getIntent().getExtras().getString("posicion");
60     final String nom_exa =getIntent().getExtras().getString("nombre_examen");
61     final String fecha_examen =getIntent().getExtras().getString("
fecha_examen");
62
63     URL_preg=preg;
64     editText6.setText(nombre_usuario+" "+nombre_alumno);
65     editText1.setText(preg.substring(6,preg.length()-4));
66     editText.setText(nom_exa);
67
68
69     Main24Activity.MyATaskCliente myATasUSER = new Main24Activity.
MyATaskCliente();
70     myATasUSER.execute("Mac24#" +nombre_usuario+"#" +nom_exa+"#" +preg+"#" +
fecha_examen);
71
72 }
73
74 class MyATaskCliente extends AsyncTask<String,Void,String> {
75     ProgressDialog progressDialog;
76
77     private String REM_HTMLTags(String Data)
78     {
79         Data = Data.replaceAll("<p>","");
80         Data = Data.replaceAll("</p>","");
81         Data = Data.replaceAll("&acute;","á");
82         Data = Data.replaceAll("&oacute;","ó");
83         Data = Data.replaceAll("&eacute;","é");
84         Data = Data.replaceAll("&iacute;","í");
85         Data = Data.replaceAll("&uacute;","ú");
86         return Data;
87     }
88
89     @Override
90     protected void onPreExecute()
91     {
92         super.onPreExecute();
93         progressDialog = new ProgressDialog(context);
94         progressDialog.setCanceledOnTouchOutside(false);
95         progressDialog.setTitle("Connecting to server");
96         progressDialog.setMessage("Please wait...");
97         progressDialog.show();
98     }

```

```

99
100     @Override
101     protected String doInBackground(String... values) {
102
103         try {
104             InetAddress serverAddr = InetAddress.getByName(ADDRESS);
105             Log.i("I/TCP Client", "Connecting...");
106             Socket socket = new Socket(serverAddr, SERVERPORT);
107             Log.i("I/TCP Client", "Connected to server");
108
109             Log.i("I/TCP Client", "Send data to server");
110             PrintStream output = new PrintStream(socket.getOutputStream());
111             String request = values[0];
112             output.println(request);
113
114             Log.i("I/TCP Client", "Received data to server");
115             InputStream stream = socket.getInputStream();
116             byte[] lenBytes = new byte[256];
117             int i;
118             String received = "";
119             while ((stream.read(lenBytes, 0, 256)) != -1) {
120                 String receivedaux = new String(lenBytes, "UTF-8");
121                 i = receivedaux.indexOf(' ');
122                 if (i != -1) received += receivedaux.substring(0, i+1);
123                 else received += receivedaux;
124             }
125
126             Log.i("I/TCP Client", "Received " + received);
127             Log.i("I/TCP Client", "");
128
129             socket.close();
130             return received;
131         } catch (UnknownHostException ex) {
132             Log.e("E/TCP Client", "" + ex.getMessage());
133             return ex.getMessage();
134         } catch (IOException ex) {
135             Log.e("E/TCP Client", "" + ex.getMessage());
136             return ex.getMessage();
137         }
138     }
139
140     @Override
141     protected void onPostExecute(String value) {
142         progressDialog.dismiss();
143         separada = value.split("#");
144
145         // QUITAR COMAS
146         separada[0] = separada[0].substring(0, separada[0].length() - 1);
147         separada[1] = separada[1].substring(1, separada[1].length() - 1);
148         separada[2] = separada[2].substring(1, separada[2].length() - 1);
149         separada[3] = separada[3].substring(1, separada[3].length() - 1);
150         separada[4] = separada[4].substring(1, separada[4].length() - 1);
151         separada[5] = separada[5].substring(1, separada[5].length());
152         separada[5] = separada[5].trim();
153
154         if (separada[0].equals("NULL")) separada[0] = "no hay profesor corrector
";

```

```
155
156     if (!separada [1].equals ("NULL")) {
157         separada [1]=REM_HTMLTags (separada [1]);
158     } else {
159         separada [1]= "no hay comentarios individuales";
160     }
161
162     if (!separada [2].equals ("NULL")) {
163         URL_solucion=separada [2];
164         separada [2]=separada [2].substring (6,separada [2].length () -4);
165     } else {
166         separada [2]= "no hay solución";
167     }
168
169     if (!separada [3].equals ("NULL")) {
170         URL_sol_alumn=separada [3];
171         separada [3]=separada [3].substring (6,separada [3].length () -4);
172     } else {
173         separada [3]= "no hay corrección";
174     }
175
176     if (!separada [4].equals ("NULL")) {
177         separada [4]=REM_HTMLTags (separada [4]);
178     } else {
179         separada [4]= "no hay comentarios generales";
180     }
181
182     if (separada [5].equals ("NULL")) {
183         separada [5]= "no hay solución del profesor";
184
185     } else {
186
187         URL_preg_sol=separada [5];
188         separada [5]=separada [5].substring (6,separada [5].length () -4);
189
190     }
191
192     editText2.setText (separada [0]);
193     editText5.setText (separada [1]);
194     editText3.setText (separada [2]);
195     editText4.setText (separada [3]);
196     editText7.setText (separada [4]);
197     editText8.setText (separada [5]);
198
199     boton1.setOnClickListener (
200     new View.OnClickListener () {
201         public void onClick (View view) {
202             Intent browserIntent = new Intent (Intent.ACTION_VIEW, Uri.
203             parse ("http://192.168.112.2/app_examen_uib/assets/uploads/files/"+URL_preg));
204             startActivity (browserIntent);
205         }
206     });
207
208     boton2.setOnClickListener (
209     new View.OnClickListener () {
210         public void onClick (View view) {
```

```

211         Intent browserIntent = new Intent(Intent.ACTION_VIEW, Uri.
parse ("http://192.168.112.2/app_examen_uib/assets/uploads/files/"+
URL_solucion));
212             startActivity (browserIntent);
213         }
214     }
215 );
216
217     boton3.setOnClickListener (
218     new View.OnClickListener () {
219         public void onClick (View view) {
220             Intent browserIntent = new Intent (Intent.ACTION_VIEW, Uri.
parse ("http://192.168.112.2/app_examen_uib/assets/uploads/files/"+
URL_sol_alumn));
221                 startActivity (browserIntent);
222             }
223         }
224     );
225
226     boton4.setOnClickListener (
227     new View.OnClickListener () {
228         public void onClick (View view) {
229             Intent browserIntent = new Intent (Intent.ACTION_VIEW, Uri.
parse ("http://192.168.112.2/app_examen_uib/assets/uploads/files/"+
URL_preg_sol));
230                 startActivity (browserIntent);
231             }
232         }
233     );
234 }
235 }
236 }

```

layout

```

1
2 <?xml version="1.0" encoding="utf-8"?>
3 <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.
android.com/apk/res/android"
4 xmlns:app="http://schemas.android.com/apk/res-auto"
5 xmlns:tools="http://schemas.android.com/tools"
6 android:layout_width="match_parent"
7 android:layout_height="match_parent"
8 tools:context="com.example.ramon.clienteandroid.Main24Activity">
9
10 <TextView
11 android:id="@+id/textView24"
12 android:layout_width="wrap_content"
13 android:layout_height="wrap_content"
14 android:layout_marginBottom="128dp"
15 android:layout_marginEnd="8dp"
16 android:layout_marginLeft="8dp"
17 android:layout_marginRight="8dp"
18 android:layout_marginStart="8dp"
19 android:layout_marginTop="8dp"
20 android:text="Comentarios generales de la pregunta"

```

```
21 android:textAppearance="@style/TextAppearance.AppCompat.Body2"
22 app:layout_constraintBottom_toBottomOf="parent"
23 app:layout_constraintHorizontal_bias="0.0"
24 app:layout_constraintLeft_toLeftOf="parent"
25 app:layout_constraintRight_toRightOf="parent"
26 app:layout_constraintTop_toBottomOf="@+id/editText13"
27 app:layout_constraintVertical_bias="0.0" />
28
29 <RelativeLayout
30 android:layout_width="0dp"
31 android:layout_height="0dp"
32 tools:layout_editor_absoluteX="8dp"
33 tools:layout_editor_absoluteY="8dp">
34
35 </RelativeLayout>
36
37 <EditText
38     android:id="@+id/editText14"
39     android:layout_width="0dp"
40     android:layout_height="42dp"
41     android:layout_marginBottom="9dp"
42     android:layout_marginEnd="8dp"
43     android:layout_marginLeft="8dp"
44     android:layout_marginRight="8dp"
45     android:layout_marginStart="8dp"
46     android:layout_marginTop="8dp"
47     android:editable="false"
48     android:ems="10"
49     android:enabled="false"
50     android:inputType="textPersonName"
51     android:text="Usuario"
52     android:textColor="#00F"
53     app:layout_constraintBottom_toBottomOf="parent"
54     app:layout_constraintHorizontal_bias="0.0"
55     app:layout_constraintLeft_toLeftOf="parent"
56     app:layout_constraintRight_toRightOf="parent"
57     app:layout_constraintTop_toTopOf="parent"
58     app:layout_constraintVertical_bias="0.013" />
59
60 <EditText
61     android:id="@+id/editText13"
62     android:layout_width="258dp"
63     android:layout_height="wrap_content"
64     android:layout_marginBottom="8dp"
65     android:layout_marginEnd="8dp"
66     android:layout_marginLeft="8dp"
67     android:layout_marginRight="8dp"
68     android:layout_marginStart="8dp"
69     android:layout_marginTop="8dp"
70     android:editable="false"
71     android:ems="10"
72     android:enabled="false"
73     android:inputType="textPersonName"
74     android:textColor="#0BF"
75     app:layout_constraintBottom_toBottomOf="parent"
76     app:layout_constraintHorizontal_bias="0.0"
77     app:layout_constraintLeft_toLeftOf="parent"
```

```
78     app:layout_constraintRight_toRightOf="parent"
79     app:layout_constraintTop_toBottomOf="@+id/textView14"
80     app:layout_constraintVertical_bias="0.0" />
81
82 <TextView
83     android:id="@+id/textView10"
84     android:layout_width="wrap_content"
85     android:layout_height="wrap_content"
86     android:layout_marginEnd="8dp"
87     android:layout_marginLeft="8dp"
88     android:layout_marginRight="8dp"
89     android:layout_marginStart="8dp"
90     android:layout_marginTop="8dp"
91     android:text="Examen"
92     android:textAppearance="@style/TextAppearance.AppCompat.Body2"
93     app:layout_constraintBottom_toBottomOf="parent"
94     app:layout_constraintHorizontal_bias="0.0"
95     app:layout_constraintLeft_toLeftOf="parent"
96     app:layout_constraintRight_toRightOf="parent"
97     app:layout_constraintTop_toBottomOf="@+id/editText14"
98     app:layout_constraintVertical_bias="0.0" />
99
100 <EditText
101     android:id="@+id/editText9"
102     android:layout_width="258dp"
103     android:layout_height="wrap_content"
104     android:layout_marginBottom="8dp"
105     android:layout_marginEnd="8dp"
106     android:layout_marginLeft="8dp"
107     android:layout_marginRight="8dp"
108     android:layout_marginStart="8dp"
109     android:layout_marginTop="8dp"
110     android:editable="false"
111     android:ems="10"
112     android:enabled="false"
113     android:inputType="textPersonName"
114     android:textColor="#0BF"
115     app:layout_constraintBottom_toBottomOf="parent"
116     app:layout_constraintHorizontal_bias="0.0"
117     app:layout_constraintLeft_toLeftOf="parent"
118     app:layout_constraintRight_toRightOf="parent"
119     app:layout_constraintTop_toBottomOf="@+id/textView13"
120     app:layout_constraintVertical_bias="0.0" />
121
122 <EditText
123     android:id="@+id/editText12"
124     android:layout_width="163dp"
125     android:layout_height="42dp"
126     android:layout_marginBottom="8dp"
127     android:layout_marginLeft="8dp"
128     android:layout_marginStart="8dp"
129     android:layout_marginTop="8dp"
130     android:editable="false"
131     android:ems="10"
132     android:enabled="false"
133     android:inputType="textPersonName"
134     android:textColor="#05F"
```

```
135     app:layout_constraintBottom_toBottomOf="parent"
136     app:layout_constraintLeft_toLeftOf="parent"
137     app:layout_constraintTop_toBottomOf="@+id/textView10"
138     app:layout_constraintVertical_bias="0.0" />
139
140 <TextView
141     android:id="@+id/textView11"
142     android:layout_width="wrap_content"
143     android:layout_height="wrap_content"
144     android:layout_marginBottom="8dp"
145     android:layout_marginEnd="8dp"
146     android:layout_marginLeft="8dp"
147     android:layout_marginRight="8dp"
148     android:layout_marginStart="8dp"
149     android:layout_marginTop="8dp"
150     android:text="Pregunta"
151     android:textAppearance="@style/TextAppearance.AppCompat.Body2"
152     app:layout_constraintBottom_toBottomOf="parent"
153     app:layout_constraintHorizontal_bias="0.0"
154     app:layout_constraintLeft_toLeftOf="parent"
155     app:layout_constraintRight_toRightOf="parent"
156     app:layout_constraintTop_toBottomOf="@+id/editText12"
157     app:layout_constraintVertical_bias="0.0" />
158
159 <EditText
160     android:id="@+id/editText10"
161     android:layout_width="wrap_content"
162     android:layout_height="wrap_content"
163     android:layout_marginBottom="8dp"
164     android:layout_marginEnd="8dp"
165     android:layout_marginLeft="8dp"
166     android:layout_marginRight="8dp"
167     android:layout_marginStart="8dp"
168     android:layout_marginTop="8dp"
169     android:editable="false"
170     android:ems="10"
171     android:enabled="false"
172     android:inputType="textPersonName"
173     android:textColor="#0BF"
174     app:layout_constraintBottom_toBottomOf="parent"
175     app:layout_constraintHorizontal_bias="0.0"
176     app:layout_constraintLeft_toLeftOf="parent"
177     app:layout_constraintRight_toRightOf="parent"
178     app:layout_constraintTop_toBottomOf="@+id/textView11"
179     app:layout_constraintVertical_bias="0.0" />
180
181 <TextView
182     android:id="@+id/textView12"
183     android:layout_width="wrap_content"
184     android:layout_height="wrap_content"
185     android:layout_marginBottom="8dp"
186     android:layout_marginEnd="8dp"
187     android:layout_marginLeft="8dp"
188     android:layout_marginRight="8dp"
189     android:layout_marginStart="8dp"
190     android:layout_marginTop="8dp"
191     android:text="Profesor corrector"
```



```
192     android:textAppearance="@style/TextAppearance.AppCompat.Body2"
193     app:layout_constraintBottom_toBottomOf="parent"
194     app:layout_constraintHorizontal_bias="0.511"
195     app:layout_constraintLeft_toRightOf="@+id/textView10"
196     app:layout_constraintRight_toRightOf="parent"
197     app:layout_constraintTop_toBottomOf="@+id/editText14"
198     app:layout_constraintVertical_bias="0.0" />
199
200 <EditText
201     android:id="@+id/editText11"
202     android:layout_width="0dp"
203     android:layout_height="wrap_content"
204     android:layout_marginBottom="223dp"
205     android:layout_marginEnd="8dp"
206     android:layout_marginLeft="8dp"
207     android:layout_marginRight="8dp"
208     android:layout_marginStart="8dp"
209     android:layout_marginTop="8dp"
210     android:editable="false"
211     android:ems="10"
212     android:enabled="false"
213     android:inputType="textPersonName"
214     app:layout_constraintBottom_toBottomOf="parent"
215     app:layout_constraintHorizontal_bias="0.0"
216     app:layout_constraintLeft_toRightOf="@+id/editText12"
217     app:layout_constraintRight_toRightOf="parent"
218     app:layout_constraintTop_toBottomOf="@+id/textView12"
219     app:layout_constraintVertical_bias="0.0" />
220
221 <TextView
222     android:id="@+id/textView13"
223     android:layout_width="wrap_content"
224     android:layout_height="wrap_content"
225     android:layout_marginBottom="8dp"
226     android:layout_marginEnd="8dp"
227     android:layout_marginLeft="8dp"
228     android:layout_marginRight="8dp"
229     android:layout_marginStart="8dp"
230     android:text="Solución alumno"
231     android:textAppearance="@style/TextAppearance.AppCompat.Body2"
232     app:layout_constraintBottom_toBottomOf="parent"
233     app:layout_constraintHorizontal_bias="0.0"
234     app:layout_constraintLeft_toLeftOf="parent"
235     app:layout_constraintRight_toRightOf="parent"
236     android:layout_marginTop="8dp"
237     app:layout_constraintTop_toBottomOf="@+id/editText31"
238     app:layout_constraintVertical_bias="0.0" />
239
240 <TextView
241     android:id="@+id/textView14"
242     android:layout_width="wrap_content"
243     android:layout_height="wrap_content"
244     android:layout_marginBottom="8dp"
245     android:layout_marginEnd="8dp"
246     android:layout_marginLeft="8dp"
247     android:layout_marginRight="8dp"
248     android:layout_marginStart="8dp"
```

```
249     android:layout_marginTop="8dp"
250     android:text="Corrección alumno"
251     android:textAppearance="@style/TextAppearance.AppCompat.Body2"
252     app:layout_constraintBottom_toBottomOf="parent"
253     app:layout_constraintHorizontal_bias="0.0"
254     app:layout_constraintLeft_toLeftOf="parent"
255     app:layout_constraintRight_toRightOf="parent"
256     app:layout_constraintTop_toBottomOf="@+id/editText9"
257     app:layout_constraintVertical_bias="0.0" />
258
259 <TextView
260     android:id="@+id/textView18"
261     android:layout_width="wrap_content"
262     android:layout_height="wrap_content"
263     android:layout_marginBottom="8dp"
264     android:layout_marginEnd="8dp"
265     android:layout_marginLeft="8dp"
266     android:layout_marginRight="8dp"
267     android:layout_marginStart="8dp"
268     android:layout_marginTop="8dp"
269     android:text="Comentarios individuales"
270     android:textAppearance="@style/TextAppearance.AppCompat.Body2"
271     app:layout_constraintBottom_toBottomOf="parent"
272     app:layout_constraintHorizontal_bias="0.0"
273     app:layout_constraintLeft_toLeftOf="parent"
274     app:layout_constraintRight_toRightOf="parent"
275     app:layout_constraintTop_toBottomOf="@+id/editText28"
276     app:layout_constraintVertical_bias="0.015" />
277
278 <EditText
279     android:id="@+id/editText18"
280     android:layout_width="0dp"
281     android:layout_height="71dp"
282     android:layout_marginBottom="8dp"
283     android:layout_marginEnd="8dp"
284     android:layout_marginLeft="8dp"
285     android:layout_marginRight="8dp"
286     android:layout_marginStart="8dp"
287     android:layout_marginTop="8dp"
288     android:editable="false"
289     android:ems="10"
290     android:enabled="false"
291     android:inputType="textMultiLine"
292     app:layout_constraintBottom_toBottomOf="parent"
293     app:layout_constraintHorizontal_bias="0.0"
294     app:layout_constraintLeft_toLeftOf="parent"
295     app:layout_constraintRight_toRightOf="parent"
296     app:layout_constraintTop_toBottomOf="@+id/textView18"
297     app:layout_constraintVertical_bias="0.0" />
298
299 <Button
300     android:id="@+id/button"
301     android:layout_width="wrap_content"
302     android:layout_height="wrap_content"
303     android:layout_marginBottom="8dp"
304     android:layout_marginEnd="8dp"
305     android:layout_marginLeft="8dp"
```

```
306     android:layout_marginRight="8dp"
307     android:layout_marginStart="8dp"
308     android:layout_marginTop="8dp"
309     android:text="ABRIR"
310     app:layout_constraintBottom_toBottomOf="parent"
311     app:layout_constraintHorizontal_bias="0.0"
312     app:layout_constraintLeft_toRightOf="@+id/editText9"
313     app:layout_constraintRight_toRightOf="parent"
314     app:layout_constraintTop_toBottomOf="@+id/textView13"
315     app:layout_constraintVertical_bias="0.0" />
316
317 <Button
318     android:id="@+id/button2"
319     android:layout_width="wrap_content"
320     android:layout_height="wrap_content"
321     android:layout_marginBottom="8dp"
322     android:layout_marginEnd="8dp"
323     android:layout_marginLeft="8dp"
324     android:layout_marginRight="8dp"
325     android:layout_marginStart="8dp"
326     android:layout_marginTop="8dp"
327     android:text="Abrir"
328     app:layout_constraintBottom_toBottomOf="parent"
329     app:layout_constraintHorizontal_bias="0.0"
330     app:layout_constraintLeft_toRightOf="@+id/editText13"
331     app:layout_constraintRight_toRightOf="parent"
332     app:layout_constraintTop_toBottomOf="@+id/textView14"
333     app:layout_constraintVertical_bias="0.0" />
334
335 <Button
336     android:id="@+id/button4"
337     android:layout_width="wrap_content"
338     android:layout_height="wrap_content"
339     android:layout_marginBottom="9dp"
340     android:layout_marginEnd="8dp"
341     android:layout_marginLeft="8dp"
342     android:layout_marginRight="8dp"
343     android:layout_marginStart="8dp"
344     android:layout_marginTop="8dp"
345     android:text="ABRIR"
346     app:layout_constraintBottom_toBottomOf="parent"
347     app:layout_constraintHorizontal_bias="0.0"
348     app:layout_constraintLeft_toRightOf="@+id/editText10"
349     app:layout_constraintRight_toRightOf="parent"
350     app:layout_constraintTop_toBottomOf="@+id/textView11"
351     app:layout_constraintVertical_bias="0.0" />
352
353 <EditText
354     android:id="@+id/editText28"
355     android:layout_width="0dp"
356     android:layout_height="71dp"
357     android:layout_marginBottom="8dp"
358     android:layout_marginEnd="8dp"
359     android:layout_marginLeft="8dp"
360     android:layout_marginRight="8dp"
361     android:layout_marginStart="8dp"
362     android:layout_marginTop="8dp"
```

```
363     android:editable="false"
364     android:ems="10"
365     android:enabled="false"
366     android:inputType="textMultiLine"
367     app:layout_constraintBottom_toTopOf="@+id/editText18"
368     app:layout_constraintHorizontal_bias="0.0"
369     app:layout_constraintLeft_toLeftOf="parent"
370     app:layout_constraintRight_toRightOf="parent"
371     app:layout_constraintTop_toBottomOf="@+id/textView24"
372     app:layout_constraintVertical_bias="0.0" />
373
374 <TextView
375     android:id="@+id/textView28"
376     android:layout_width="wrap_content"
377     android:layout_height="wrap_content"
378     android:layout_marginBottom="8dp"
379     android:layout_marginLeft="8dp"
380     android:layout_marginRight="8dp"
381     android:layout_marginTop="8dp"
382     android:text="Pregunta resuelta"
383     android:textAppearance="@style/TextAppearance.AppCompat.Body2"
384     app:layout_constraintBottom_toBottomOf="parent"
385     app:layout_constraintHorizontal_bias="0.0"
386     app:layout_constraintLeft_toLeftOf="parent"
387     app:layout_constraintRight_toRightOf="parent"
388     app:layout_constraintTop_toBottomOf="@+id/editText10"
389     app:layout_constraintVertical_bias="0.0" />
390
391 <EditText
392     android:id="@+id/editText31"
393     android:layout_width="258dp"
394     android:layout_height="wrap_content"
395     android:layout_marginBottom="8dp"
396     android:layout_marginLeft="8dp"
397     android:layout_marginRight="8dp"
398     android:layout_marginTop="8dp"
399     android:editable="false"
400     android:ems="10"
401     android:enabled="false"
402     android:inputType="textPersonName"
403     android:textColor="#0BF"
404     app:layout_constraintBottom_toBottomOf="parent"
405     app:layout_constraintHorizontal_bias="0.0"
406     app:layout_constraintLeft_toLeftOf="parent"
407     app:layout_constraintRight_toRightOf="parent"
408     app:layout_constraintTop_toBottomOf="@+id/textView28"
409     app:layout_constraintVertical_bias="0.016" />
410
411 <Button
412     android:id="@+id/button9"
413     android:layout_width="wrap_content"
414     android:layout_height="wrap_content"
415     android:layout_marginBottom="8dp"
416     android:layout_marginLeft="8dp"
417     android:layout_marginRight="8dp"
418     android:layout_marginTop="8dp"
419     android:text="ABRIR"
```

```

420     app:layout_constraintBottom_toBottomOf="parent"
421     app:layout_constraintHorizontal_bias="0.012"
422     app:layout_constraintLeft_toRightOf="@+id/editText31"
423     app:layout_constraintRight_toRightOf="parent"
424     app:layout_constraintTop_toBottomOf="@+id/textView28"
425     app:layout_constraintVertical_bias="0.0" />
426
427 </android.support.constraint.ConstraintLayout>

```

8.3.6 Pantalla de selección de exámenes de años anteriores

MainActivity

```

1  package com.example.ramon.clienteandroid;
2
3  import android.app.AlertDialog;
4  import android.content.Context;
5  import android.content.Intent;
6  import android.os.AsyncTask;
7  import android.support.v7.app.AppCompatActivity;
8  import android.os.Bundle;
9  import android.util.Log;
10 import android.view.View;
11 import android.widget.AdapterView;
12 import android.widget.AdapterView.OnItemClickListener;
13 import android.widget.ArrayAdapter;
14 import android.widget.Button;
15 import android.widget.EditText;
16 import android.widget.ListAdapter;
17 import android.widget.ListView;
18
19 import java.io.IOException;
20 import java.io.InputStream;
21 import java.io.OutputStream;
22 import java.net.InetAddress;
23 import java.net.Socket;
24 import java.net.UnknownHostException;
25 import java.util.ArrayList;
26
27 public class Main25Activity extends AppCompatActivity {
28     private Context context = this;
29     private EditText editText;
30     String nombre_usuario, nombre_alumno, asignatura;
31     String [] datos;
32
33     private ListView lista;
34     private ArrayAdapter<String> adaptador;
35     ArrayList<String> ejemplo = new ArrayList<String>();
36     ArrayList<String> ejemplo2 = new ArrayList<String>();
37
38     private static final int SERVERPORT = 5000;
39     private static final String ADDRESS = "192.168.112.2";
40
41     @Override
42     protected void onCreate(Bundle savedInstanceState) {
43         super.onCreate(savedInstanceState);
44         setContentView(R.layout.activity_main25);

```

```

44
45     editText = ((EditText) findViewById(R.id.editText17));
46     lista = (ListView) findViewById(R.id.listView);
47
48     nombre_usuario = getIntent().getExtras().getString("nombre_usuario");
49     nombre_alumno = getIntent().getExtras().getString("nombre_alumno");
50     asignatura = getIntent().getExtras().getString("posicion");
51
52     editText.setText(nombre_usuario+ " "+nombre_alumno);
53
54     asignatura=asignatura.substring(0,5);
55     Main25Activity.MyATaskCliente myATasUSER = new Main25Activity.
MyATaskCliente();
56     myATasUSER.execute("MAC25#"+asignatura+"#" + nombre_usuario);
57 }
58
59
60 class MyATaskCliente extends AsyncTask<String, Void, String> {
61
62     ProgressDialog progressDialog;
63
64     @Override
65     protected void onPreExecute()
66     {
67         super.onPreExecute();
68         progressDialog = new ProgressDialog(context);
69         progressDialog.setCanceledOnTouchOutside(false);
70         progressDialog.setTitle("Connecting to server");
71         progressDialog.setMessage("Please wait...");
72         progressDialog.show();
73     }
74
75     @Override
76     protected String doInBackground(String... values){
77
78         try {
79             InetAddress serverAddr = InetAddress.getByName(ADDRESS);
80             Log.i("I/TCP Client", "Connecting...");
81             Socket socket = new Socket(serverAddr, SERVERPORT);
82             Log.i("I/TCP Client", "Connected to server");
83
84             Log.i("I/TCP Client", "Send data to server");
85             PrintStream output = new PrintStream(socket.getOutputStream());
86             String request = values[0];
87             output.println(request);
88             Log.i("I/TCP Client", "Received data to server");
89             InputStream stream = socket.getInputStream();
90             byte[] lenBytes = new byte[256];
91             int i;
92             String received = "";
93             while ((stream.read(lenBytes,0,256)) != -1) {
94                 String receivedaux = new String(lenBytes, "UTF-8");
95                 i = receivedaux.indexOf(' ');
96                 if (i != -1) received += receivedaux.substring(0,i+1);
97                 else         received += receivedaux;
98             }
99

```

```

100         Log.i("I/TCP Client", "Received " + received);
101         Log.i("I/TCP Client", "");
102
103         socket.close();
104         return received;
105     } catch (UnknownHostException ex) {
106         Log.e("E/TCP Client", "" + ex.getMessage());
107         return ex.getMessage();
108     } catch (IOException ex) {
109         Log.e("E/TCP Client", "" + ex.getMessage());
110         return ex.getMessage();
111     }
112 }
113
114 @Override
115 protected void onPostExecute(String value) {
116
117     progressDialog.dismiss();
118     datos = value.split("#");
119
120     String valor;
121     if (datos[0].length() > 2) {
122         valor=datos[0].substring(1,datos[0].length()-1);
123         datos[1]=datos[1].substring(1,datos[1].length());
124     } else {
125         valor = "No has realizado exámenes de años anteriores";
126     }
127
128     int i = 0, Index, fromIndex = 0,Indexdate=0,fromIndexdate=0;
129     while (true) {
130         Index = valor.indexOf(',',',', fromIndex);
131         Indexdate = datos[1].indexOf(',',',', fromIndexdate);
132         if (Index == -1) {
133             ejemplo.add(valor.substring(fromIndex));
134             ejemplo2.add(valor.substring(fromIndex) + "- Fecha examen:"+
135 datos[1].substring(fromIndexdate));
136             break;
137         }
138         ejemplo.add(valor.substring(fromIndex, Index));
139         ejemplo2.add(valor.substring(fromIndex, Index) + "- Fecha examen:
140 "+datos[1].substring(fromIndexdate, Indexdate));
141         fromIndex = Index + 1;
142         fromIndexdate=Indexdate+1;
143     }
144
145     adaptador = new ArrayAdapter(context, android.R.layout.
146 simple_list_item_1, ejemplo);
147     lista.setAdapter(adaptador);
148     if (!valor.equals("No has realizado exámenes de años anteriores")) {
149         lista.setOnItemClickListener(new AdapterView.OnItemClickListener
150 () {
151
152             @Override
153             public void onItemClick(AdapterView<?> parent, View view, int
154 position, long id) {
155
156                 Intent intent = new Intent(Main25Activity.this,
157 Main27Activity.class);

```

8. ANEXOS

```
151
152         intent.putExtra("posicion", ejemplo.get(position));
153         intent.putExtra("nombre_usuario", nombre_usuario);
154         intent.putExtra("nombre_alumno", nombre_alumno);
155         intent.putExtra("asignatura", asignatura);
156
157         startActivity(intent);
158     }
159 }
160 }
161 }
162 }
163 }
```

layout

```
1
2 <?xml version="1.0" encoding="utf-8"?>
3 <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.
  android.com/apk/res/android"
4 xmlns:app="http://schemas.android.com/apk/res-auto"
5 xmlns:tools="http://schemas.android.com/tools"
6 android:layout_width="match_parent"
7 android:layout_height="match_parent"
8 tools:context="com.example.ramon.clienteandroid.Main25Activity">
9
10 <EditText
11     android:id="@+id/editText17"
12     android:layout_width="0dp"
13     android:layout_height="wrap_content"
14     android:layout_marginBottom="8dp"
15     android:layout_marginEnd="8dp"
16     android:layout_marginLeft="8dp"
17     android:layout_marginRight="8dp"
18     android:layout_marginStart="8dp"
19     android:layout_marginTop="8dp"
20     android:editable="false"
21     android:ems="10"
22     android:enabled="false"
23     android:inputType="textPersonName"
24     android:text="Name"
25     android:textColor="#00F"
26     app:layout_constraintBottom_toBottomOf="parent"
27     app:layout_constraintHorizontal_bias="0.0"
28     app:layout_constraintLeft_toLeftOf="parent"
29     app:layout_constraintRight_toRightOf="parent"
30     app:layout_constraintTop_toTopOf="parent"
31     app:layout_constraintVertical_bias="0.013" />
32
33 <TextView
34     android:id="@+id/textView15"
35     android:layout_width="wrap_content"
36     android:layout_height="wrap_content"
37     android:layout_marginBottom="8dp"
38     android:layout_marginEnd="8dp"
39     android:layout_marginLeft="8dp"
```



```

40     android:layout_marginRight="8dp"
41     android:layout_marginStart="8dp"
42     android:layout_marginTop="8dp"
43     android:text="Exámenes de los años anteriores"
44     android:textAppearance="@style/TextAppearance.AppCompat.Body2"
45     app:layout_constraintBottom_toBottomOf="parent"
46     app:layout_constraintHorizontal_bias="0.0"
47     app:layout_constraintLeft_toLeftOf="parent"
48     app:layout_constraintRight_toRightOf="parent"
49     app:layout_constraintTop_toBottomOf="@+id/editText17"
50     app:layout_constraintVertical_bias="0.0" />
51
52 <ListView
53     android:id="@+id/listView"
54     android:layout_width="405dp"
55     android:layout_height="wrap_content"
56     android:layout_marginLeft="8dp"
57     android:layout_marginStart="8dp"
58     android:layout_marginTop="8dp"
59     app:layout_constraintLeft_toLeftOf="parent"
60     app:layout_constraintTop_toBottomOf="@+id/textView15" />
61 </android.support.constraint.ConstraintLayout>

```

8.3.7 Pantalla de selección del tipo de examen

MainActivity

```

1  package com.example.ramon.clienteandroid;
2
3  import android.content.Context;
4  import android.content.Intent;
5  import android.support.v7.app.AppCompatActivity;
6  import android.os.Bundle;
7  import android.util.Log;
8  import android.view.View;
9  import android.widget.Button;
10 import android.widget.EditText;
11 import android.widget.ListAdapter;
12 import android.widget.Toast;
13
14 import java.util.ArrayList;
15
16 public class Main26Activity extends AppCompatActivity {
17
18     private EditText editText, editText1;
19     private Button boton, boton1;
20     @Override
21     protected void onCreate(Bundle savedInstanceState) {
22         super.onCreate(savedInstanceState);
23         setContentView(R.layout.activity_main26);
24
25         editText = ((EditText) findViewById(R.id.editText19));
26         editText1 = ((EditText) findViewById(R.id.editText20));
27         boton = ((Button) findViewById(R.id.button6));
28         boton1 = ((Button) findViewById(R.id.button7));
29

```

8. ANEXOS

```
30     final String nombre_usuario =getIntent().getExtras().getString("
    nombre_usuario");
31     final String nombre_alumno =getIntent().getExtras().getString("
    nombre_alumno");
32     final String asign =getIntent().getExtras().getString("posicion");
33
34     editText.setText(nombre_usuario+" "+nombre_alumno);
35     editText1.setText(asign);
36
37     boton.setOnClickListener(//boton examenes años anteriores
38     new View.OnClickListener() {
39         public void onClick(View view) {
40             Intent intent = new Intent(Main26Activity.this, Main25Activity.
    class);
41             intent.putExtra("posicion", asign);
42             intent.putExtra("nombre_usuario", nombre_usuario);
43             intent.putExtra("nombre_alumno", nombre_alumno);
44             startActivity(intent);
45         }
46     });
47
48     boton1.setOnClickListener(//boton examenes acutuales
49     new View.OnClickListener() {
50         public void onClick(View view) {
51             Intent intent = new Intent(Main26Activity.this, Main22Activity.
    class);
52             intent.putExtra("posicion", asign);
53             intent.putExtra("nombre_usuario", nombre_usuario);
54             intent.putExtra("nombre_alumno", nombre_alumno);
55             startActivity(intent);
56         }
57     });
58 }
59 }
```

layout

```
1
2 <?xml version="1.0" encoding="utf-8"?>
3 <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.
    android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     tools:context="com.example.ramon.clienteandroid.Main26Activity">
9
10 <EditText
11     android:id="@+id/editText19"
12     android:layout_width="0dp"
13     android:layout_height="wrap_content"
14     android:layout_marginBottom="8dp"
15     android:layout_marginEnd="8dp"
16     android:layout_marginLeft="8dp"
17     android:layout_marginRight="8dp"
18     android:layout_marginStart="8dp"
```

```
19     android:layout_marginTop="8dp"
20     android:editable="false"
21     android:ems="10"
22     android:enabled="false"
23     android:inputType="textPersonName"
24     android:text="Name"
25     android:textColor="#00F"
26     app:layout_constraintBottom_toBottomOf="parent"
27     app:layout_constraintHorizontal_bias="0.0"
28     app:layout_constraintLeft_toLeftOf="parent"
29     app:layout_constraintRight_toRightOf="parent"
30     app:layout_constraintTop_toTopOf="parent"
31     app:layout_constraintVertical_bias="0.013" />
32
33 <TextView
34     android:id="@+id/textView16"
35     android:layout_width="wrap_content"
36     android:layout_height="wrap_content"
37     android:layout_marginBottom="8dp"
38     android:layout_marginEnd="8dp"
39     android:layout_marginLeft="8dp"
40     android:layout_marginRight="8dp"
41     android:layout_marginStart="8dp"
42     android:layout_marginTop="8dp"
43     android:text="Asignatura"
44     android:textAppearance="@style/TextAppearance.AppCompat.Body2"
45     app:layout_constraintBottom_toBottomOf="parent"
46     app:layout_constraintHorizontal_bias="0.0"
47     app:layout_constraintLeft_toLeftOf="parent"
48     app:layout_constraintRight_toRightOf="parent"
49     app:layout_constraintTop_toBottomOf="@+id/editText19"
50     app:layout_constraintVertical_bias="0.001" />
51
52 <EditText
53     android:id="@+id/editText20"
54     android:layout_width="0dp"
55     android:layout_height="wrap_content"
56     android:layout_marginBottom="8dp"
57     android:layout_marginEnd="8dp"
58     android:layout_marginLeft="8dp"
59     android:layout_marginRight="8dp"
60     android:layout_marginStart="8dp"
61     android:layout_marginTop="8dp"
62     android:editable="false"
63     android:ems="10"
64     android:enabled="false"
65     android:inputType="textPersonName"
66     android:text="Name"
67     app:layout_constraintBottom_toBottomOf="parent"
68     app:layout_constraintHorizontal_bias="0.044"
69     app:layout_constraintLeft_toLeftOf="parent"
70     app:layout_constraintRight_toRightOf="parent"
71     app:layout_constraintTop_toBottomOf="@+id/textView16"
72     app:layout_constraintVertical_bias="0.0" />
73
74 <Button
75     android:id="@+id/button6"
```

8. ANEXOS

```
76     android:layout_width="wrap_content"
77     android:layout_height="wrap_content"
78     android:layout_marginBottom="317dp"
79     android:layout_marginEnd="8dp"
80     android:layout_marginLeft="8dp"
81     android:layout_marginRight="8dp"
82     android:layout_marginStart="8dp"
83     android:layout_marginTop="8dp"
84     android:text="Acceso a examenes de años anteriores"
85     app:layout_constraintBottom_toBottomOf="parent"
86     app:layout_constraintHorizontal_bias="0.0"
87     app:layout_constraintLeft_toLeftOf="parent"
88     app:layout_constraintRight_toRightOf="parent"
89     app:layout_constraintTop_toBottomOf="@+id/button7"
90     app:layout_constraintVertical_bias="0.0" />
91
92 <Button
93     android:id="@+id/button7"
94     android:layout_width="wrap_content"
95     android:layout_height="wrap_content"
96     android:layout_marginBottom="9dp"
97     android:layout_marginEnd="8dp"
98     android:layout_marginLeft="8dp"
99     android:layout_marginRight="8dp"
100    android:layout_marginStart="8dp"
101    android:layout_marginTop="8dp"
102    android:text="Acceso a examenes actuales"
103    app:layout_constraintBottom_toBottomOf="parent"
104    app:layout_constraintHorizontal_bias="0.0"
105    app:layout_constraintLeft_toLeftOf="parent"
106    app:layout_constraintRight_toRightOf="parent"
107    app:layout_constraintTop_toBottomOf="@+id/editText20"
108    app:layout_constraintVertical_bias="0.0" />
109
110
111 </android.support.constraint.ConstraintLayout>
```

8.3.8 Pantalla de selección de preguntas de exámenes de años anteriores

MainActivity

```
1 package com.example.ramon.clienteandroid;
2
3 import android.app.ProgressDialog;
4 import android.content.Context;
5 import android.content.Intent;
6 import android.os.AsyncTask;
7 import android.support.v7.app.AppCompatActivity;
8 import android.os.Bundle;
9 import android.util.Log;
10 import android.view.View;
11 import android.widget.AdapterView;
12 import android.widget.AdapterView;
13 import android.widget.ArrayAdapter;
14 import android.widget.Button;
15 import android.widget.EditText;
16 import android.widget.ListAdapter;
```

```

16 import android.widget.ListView;
17
18 import java.io.IOException;
19 import java.io.InputStream;
20 import java.io.OutputStream;
21 import java.net.InetAddress;
22 import java.net.Socket;
23 import java.net.UnknownHostException;
24 import java.util.ArrayList;
25
26 public class Main27Activity extends AppCompatActivity {
27     private Context context = this;
28     private ListView lista;
29     private EditText editText, editText1, editText2, editText3;
30     private ArrayAdapter<String> adaptador;
31     String nombre_usuario, nombre_alumno, asignatura;
32     String[] datos;
33     ArrayList<String> ejemplo = new ArrayList<String>();
34     ArrayList<String> ejemplo2 = new ArrayList<String>();
35
36     private static final int SERVERPORT = 5000;
37     private static final String ADDRESS = "192.168.112.2";
38     @Override
39     protected void onCreate(Bundle savedInstanceState) {
40         super.onCreate(savedInstanceState);
41         setContentView(R.layout.activity_main27);
42
43         editText = ((EditText) findViewById(R.id.editText21));
44         editText1 = ((EditText) findViewById(R.id.editText22));
45         editText2 = ((EditText) findViewById(R.id.editText27));
46         editText3 = ((EditText) findViewById(R.id.editText30));
47         lista = (ListView) findViewById(R.id.listView);
48
49         nombre_usuario = getIntent().getExtras().getString("nombre_usuario");
50         nombre_alumno = getIntent().getExtras().getString("nombre_alumno");
51         final String examen = getIntent().getExtras().getString("posicion");
52         asignatura = getIntent().getExtras().getString("asignatura");
53
54         editText.setText(nombre_usuario+" "+nombre_alumno);
55         editText1.setText(examen);
56
57         Main27Activity.MyATaskCliente myATasUSER = new Main27Activity.
58         MyATaskCliente();
59         myATasUSER.execute("Mac27#" + examen + "#" + asignatura);
60     }
61
62     class MyATaskCliente extends AsyncTask<String, Void, String> {
63
64         ProgressDialog progressDialog;
65         private String REM_HTMLTags(String Data)
66         {
67             Data = Data.replaceAll("<p>", "");
68             Data = Data.replaceAll("</p>", "");
69             return Data;
70         }
71

```

```

72     @Override
73     protected void onPreExecute ()
74     {
75         super.onPreExecute ();
76         progressDialog = new ProgressDialog (context);
77         progressDialog.setCanceledOnTouchOutside (false);
78         progressDialog.setTitle ("Connecting to server");
79         progressDialog.setMessage ("Please wait ...");
80         progressDialog.show ();
81     }
82
83     @Override
84     protected String doInBackground (String ... values) {
85
86         try {
87             InetAddress serverAddr = InetAddress.getByName (ADDRESS);
88             Log.i ("I/TCP Client", "Connecting ...");
89             Socket socket = new Socket (serverAddr, SERVERPORT);
90             Log.i ("I/TCP Client", "Connected to server");
91
92             Log.i ("I/TCP Client", "Send data to server");
93             PrintStream output = new PrintStream (socket.getOutputStream ());
94             String request = values [0];
95             output.println (request);
96
97             Log.i ("I/TCP Client", "Received data to server");
98             InputStream stream = socket.getInputStream ();
99             byte [] lenBytes = new byte [256];
100            int i;
101            String received = "";
102            while ((stream.read (lenBytes, 0, 256)) != -1) {
103                String receivedaux = new String (lenBytes, "UTF-8");
104                i = receivedaux.indexOf (' ');
105                if (i != -1) received += receivedaux.substring (0, i+1);
106                else received += receivedaux;
107            }
108            Log.i ("I/TCP Client", "Received " + received);
109            Log.i ("I/TCP Client", "");
110            socket.close ();
111            return received;
112        } catch (UnknownHostException ex) {
113            Log.e ("E/TCP Client", "" + ex.getMessage ());
114            return ex.getMessage ();
115        } catch (IOException ex) {
116            Log.e ("E/TCP Client", "" + ex.getMessage ());
117            return ex.getMessage ();
118        }
119    }
120
121    @Override
122    protected void onPostExecute (String value) {
123        progressDialog.dismiss ();
124
125        datos = value.split ("#");
126
127        if (datos [1].length () > 2 & datos [1] != "null,") {
128            datos [1] = datos [1].substring (1, datos [1].length () - 1);

```

```

129         datos[1]=REM_HTMLTags(datos[1]);
130     } else {
131         datos[1] = "No hay comentario asociado al examen";
132     }
133     editText2.setText(datos[1]);
134
135     if (datos[2].length() > 2) {
136         datos[2] = datos[2].substring(1, datos[2].length());
137     } else {
138         datos[2] = "No hay fecha de realización del examen";
139     }
140     editText3.setText(datos[2]);
141
142     String valor = datos[0];
143     if (valor.length() > 2) {
144         valor = valor.substring(0, valor.length() - 1);
145     } else {
146         valor = "no hay preguntas asociadas al examen";
147     }
148
149     int i = 1, Index, fromIndex = 0;
150     while (true) {
151         Index = valor.indexOf(',', fromIndex);
152         if (Index == -1) {
153             ejemplo.add(valor.substring(fromIndex));
154             ejemplo2.add("Pregunta #" + i); i++;
155             break;
156         }
157         ejemplo.add(valor.substring(fromIndex, Index));
158         ejemplo2.add("Pregunta #" + i); i++;
159         fromIndex = Index + 1;
160     }
161
162     adaptador = new ArrayAdapter(context, android.R.layout.
simple_list_item_1, ejemplo2);
163     lista.setAdapter(adaptador);
164     if(valor != "no hay preguntas asociadas al examen") {
165         lista.setOnItemClickListener(new AdapterView.OnItemClickListener
() {
166             @Override
167             public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {
168
169                 Intent intent = new Intent(Main27Activity.this,
Main28Activity.class);
170
171                 intent.putExtra("posicion", ejemplo.get(position));
172                 intent.putExtra("nombre_usuario", nombre_usuario);
173                 intent.putExtra("nombre_alumno", nombre_alumno);
174                 intent.putExtra("asignatura", asignatura);
175                 intent.putExtra("fecha_examen", datos[2]);
176
177                 startActivity(intent);
178             }
179         });
180     }
181 }

```

8. ANEXOS

```
182 }
183 }
```

layout

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.
  android.com/apk/res/android"
3 xmlns:app="http://schemas.android.com/apk/res-auto"
4 xmlns:tools="http://schemas.android.com/tools"
5 android:layout_width="match_parent"
6 android:layout_height="match_parent"
7 tools:context="com.example.ramon.clienteandroid.Main27Activity">
8
9 <EditText
10     android:id="@+id/editText21"
11     android:layout_width="0dp"
12     android:layout_height="wrap_content"
13     android:layout_marginBottom="8dp"
14     android:layout_marginEnd="8dp"
15     android:layout_marginLeft="8dp"
16     android:layout_marginRight="8dp"
17     android:layout_marginStart="8dp"
18     android:layout_marginTop="8dp"
19     android:editable="false"
20     android:ems="10"
21     android:enabled="false"
22     android:inputType="textPersonName"
23     android:text="Name"
24     android:textColor="#00F"
25     android:focusable="false"
26     app:layout_constraintBottom_toBottomOf="parent"
27     app:layout_constraintHorizontal_bias="0.0"
28     app:layout_constraintLeft_toLeftOf="parent"
29     app:layout_constraintRight_toRightOf="parent"
30     app:layout_constraintTop_toTopOf="parent"
31     app:layout_constraintVertical_bias="0.013" />
32
33 <TextView
34     android:id="@+id/textView17"
35     android:layout_width="63dp"
36     android:layout_height="18dp"
37     android:layout_marginBottom="8dp"
38     android:layout_marginEnd="8dp"
39     android:layout_marginLeft="8dp"
40     android:layout_marginRight="8dp"
41     android:layout_marginStart="8dp"
42     android:layout_marginTop="8dp"
43     android:text="Examen"
44     android:textAppearance="@style/TextAppearance.AppCompat.Body2"
45     app:layout_constraintBottom_toBottomOf="parent"
46     app:layout_constraintHorizontal_bias="0.0"
47     app:layout_constraintLeft_toLeftOf="parent"
48     app:layout_constraintRight_toRightOf="parent"
49     app:layout_constraintTop_toBottomOf="@+id/editText21"
50     app:layout_constraintVertical_bias="0.0" />
```



```
51
52 <EditText
53     android:id="@+id/editText22"
54     android:layout_width="183dp"
55     android:layout_height="40dp"
56     android:layout_marginBottom="8dp"
57     android:layout_marginEnd="8dp"
58     android:layout_marginLeft="8dp"
59     android:layout_marginStart="8dp"
60     android:layout_marginTop="8dp"
61     android:editable="false"
62     android:ems="10"
63     android:enabled="false"
64     android:inputType="textPersonName"
65     android:text="Name"
66     android:focusable="false"
67     android:textColor="#05F"
68     app:layout_constraintBottom_toBottomOf="parent"
69     app:layout_constraintLeft_toLeftOf="parent"
70     app:layout_constraintTop_toBottomOf="@+id/textView17"
71     app:layout_constraintVertical_bias="0.0" />
72
73 <TextView
74     android:id="@+id/textView21"
75     android:layout_width="wrap_content"
76     android:layout_height="wrap_content"
77     android:layout_marginBottom="8dp"
78     android:layout_marginEnd="8dp"
79     android:layout_marginLeft="8dp"
80     android:layout_marginRight="8dp"
81     android:layout_marginStart="8dp"
82     android:layout_marginTop="8dp"
83     android:text="Preguntas"
84     android:textAppearance="@style/TextAppearance.AppCompat.Body2"
85
86     app:layout_constraintBottom_toBottomOf="parent"
87     app:layout_constraintHorizontal_bias="0.0"
88     app:layout_constraintLeft_toLeftOf="parent"
89     app:layout_constraintRight_toRightOf="parent"
90     app:layout_constraintTop_toBottomOf="@+id/editText27"
91     app:layout_constraintVertical_bias="0.0" />
92
93 <TextView
94     android:id="@+id/textView23"
95     android:layout_width="wrap_content"
96     android:layout_height="wrap_content"
97     android:layout_marginBottom="8dp"
98     android:layout_marginEnd="8dp"
99     android:layout_marginLeft="8dp"
100    android:layout_marginRight="8dp"
101    android:layout_marginStart="8dp"
102    android:layout_marginTop="8dp"
103    android:text="Comentarios"
104    android:textAppearance="@style/TextAppearance.AppCompat.Body2"
105    app:layout_constraintBottom_toBottomOf="parent"
106    app:layout_constraintHorizontal_bias="0.0"
107    app:layout_constraintLeft_toLeftOf="parent"
```

```
108     app:layout_constraintRight_toRightOf="parent"
109     app:layout_constraintTop_toBottomOf="@+id/editText22"
110     app:layout_constraintVertical_bias="0.0" />
111
112 <EditText
113     android:id="@+id/editText27"
114     android:layout_width="383dp"
115     android:layout_height="138dp"
116     android:layout_marginEnd="8dp"
117     android:layout_marginLeft="8dp"
118     android:layout_marginRight="8dp"
119     android:layout_marginStart="8dp"
120     android:layout_marginTop="8dp"
121     android:ems="10"
122     android:inputType="textMultiLine"
123     android:focusable="false"
124     app:layout_constraintHorizontal_bias="0.0"
125     app:layout_constraintLeft_toLeftOf="parent"
126     app:layout_constraintRight_toRightOf="parent"
127     app:layout_constraintTop_toBottomOf="@+id/textView23" />
128
129 <TextView
130     android:id="@+id/textView26"
131     android:layout_width="wrap_content"
132     android:layout_height="wrap_content"
133     android:layout_marginBottom="8dp"
134     android:layout_marginLeft="0dp"
135     android:layout_marginRight="8dp"
136     android:layout_marginTop="8dp"
137     android:text="Fecha del examen"
138     android:textAppearance="@style/TextAppearance.AppCompat.Body2"
139     app:layout_constraintBottom_toBottomOf="parent"
140     app:layout_constraintHorizontal_bias="0.0"
141     app:layout_constraintLeft_toLeftOf="@+id/editText30"
142     app:layout_constraintRight_toRightOf="parent"
143     app:layout_constraintTop_toBottomOf="@+id/editText21"
144     app:layout_constraintVertical_bias="0.0" />
145
146 <EditText
147     android:id="@+id/editText30"
148     android:layout_width="192dp"
149     android:layout_height="42dp"
150     android:layout_marginBottom="244dp"
151     android:layout_marginLeft="8dp"
152     android:layout_marginRight="27dp"
153     android:layout_marginTop="8dp"
154     android:editable="false"
155     android:ems="10"
156     android:enabled="false"
157     android:inputType="textPersonName"
158     android:text="Name"
159     android:textColor="#F00"
160     android:focusable="false"
161     app:layout_constraintBottom_toBottomOf="parent"
162     app:layout_constraintHorizontal_bias="0.0"
163     app:layout_constraintLeft_toRightOf="@+id/editText22"
164     app:layout_constraintRight_toRightOf="parent"
```

```

165     app:layout_constraintTop_toBottomOf="@+id/textView26"
166     app:layout_constraintVertical_bias="0.0" />
167
168 <ListView
169     android:id="@+id/listView"
170     android:layout_width="388dp"
171     android:layout_height="423dp"
172     android:layout_marginLeft="7dp"
173     android:layout_marginStart="8dp"
174     android:layout_marginTop="0dp"
175     android:focusable="false"
176     app:layout_constraintLeft_toLeftOf="parent"
177     app:layout_constraintTop_toBottomOf="@+id/textView21" />
178
179 </android.support.constraint.ConstraintLayout>

```

8.3.9 Pantalla de solución de un pregunta de años anteriores

MainActivity

```

1  package com.example.ramon.clienteandroid;
2
3  import android.app.ProgressDialog;
4  import android.content.Context;
5  import android.content.Intent;
6  import android.net.Uri;
7  import android.os.AsyncTask;
8  import android.support.v7.app.AppCompatActivity;
9  import android.os.Bundle;
10 import android.util.Log;
11 import android.view.View;
12 import android.widget.Button;
13 import android.widget.EditText;
14
15 import java.io.IOException;
16 import java.io.InputStream;
17 import java.io.OutputStream;
18 import java.net.InetAddress;
19 import java.net.Socket;
20 import java.net.UnknownHostException;
21
22 public class Main28Activity extends AppCompatActivity {
23     private EditText editText, editText1, editText2, editText3;
24     private Button boton, boton1;
25     String nombre_usuario, nombre_alumno, pregunta, asignatura, fecha_examen;
26     private Context context = this;
27     String[] datos;
28
29     private static final int SERVERPORT = 5000;
30     private static final String ADDRESS = "192.168.112.2";
31     @Override
32     protected void onCreate(Bundle savedInstanceState) {
33         super.onCreate(savedInstanceState);
34         setContentView(R.layout.activity_main28);
35
36         editText = ((EditText) findViewById(R.id.editText23));

```

```

37     editText1 = ((EditText) findViewById(R.id.editText24));
38     editText2 = ((EditText) findViewById(R.id.editText26));
39     editText3 = ((EditText) findViewById(R.id.editText25));
40     boton=((Button) findViewById(R.id.button5));
41     boton1=((Button) findViewById(R.id.button8));
42
43     nombre_usuario =getIntent().getExtras().getString("nombre_usuario");
44     nombre_alumno =getIntent().getExtras().getString("nombre_alumno");
45     pregunta =getIntent().getExtras().getString("posicion");
46     asignatura =getIntent().getExtras().getString("asignatura");
47     fecha_examen =getIntent().getExtras().getString("fecha_examen");
48
49     editText.setText(nombre_usuario+" "+nombre_alumno);
50     editText1.setText(pregunta.substring(6,pregunta.length()));
51
52     asignatura=asignatura.substring(0,5);
53     Main28Activity.MyTaskCliente myATasUSER = new Main28Activity.
MyTaskCliente();
54     myATasUSER.execute("MAC28#"+pregunta+"#"+asignatura+"#"+fecha_examen);
55 }
56
57
58
59 class MyTaskCliente extends AsyncTask<String,Void,String> {
60
61     ProgressDialog progressDialog;
62     private String REM_HTMLTags(String Data)
63     {
64         Data = Data.replaceAll("<p>","");
65         Data = Data.replaceAll("</p>","");
66         return Data;
67     }
68
69     @Override
70     protected void onPreExecute()
71     {
72         super.onPreExecute();
73         progressDialog = new ProgressDialog(context);
74         progressDialog.setCanceledOnTouchOutside(false);
75         progressDialog.setTitle("Connecting to server");
76         progressDialog.setMessage("Please wait...");
77         progressDialog.show();
78     }
79
80     @Override
81     protected String doInBackground(String... values){
82
83         try {
84             InetAddress serverAddr = InetAddress.getByName(ADDRESS);
85             Log.i("I/TCP Client", "Connecting...");
86             Socket socket = new Socket(serverAddr, SERVERPORT);
87             Log.i("I/TCP Client", "Connected to server");
88
89             Log.i("I/TCP Client", "Send data to server");
90             PrintStream output = new PrintStream(socket.getOutputStream());
91             String request = values[0];
92             output.println(request);

```

```

93     Log.i("I/TCP Client", "Received data to server");
94     InputStream stream = socket.getInputStream();
95     byte[] lenBytes = new byte[256];
96     int i;
97     String received = "";
98     while ((stream.read(lenBytes,0,256)) != -1) {
99         String receivedaux = new String(lenBytes, "UTF-8");
100        i = receivedaux.indexOf(' ');
101        if (i != -1) received += receivedaux.substring(0,i+1);
102        else received += receivedaux;
103    }
104    Log.i("I/TCP Client", "Received " + received);
105    Log.i("I/TCP Client", "");
106
107    socket.close();
108    return received;
109 } catch (UnknownHostException ex) {
110     Log.e("E/TCP Client", "" + ex.getMessage());
111     return ex.getMessage();
112 } catch (IOException ex) {
113     Log.e("E/TCP Client", "" + ex.getMessage());
114     return ex.getMessage();
115 }
116 }
117
118 @Override
119 protected void onPostExecute(String value){
120     progressDialog.dismiss();
121     datos = value.split("#");
122
123     if(datos[0].length()>7) {
124         editText2.setText(datos[0].substring(6, datos[0].length() - 1));
125     } else {
126         editText2.setText("no hay solución para esta pregunta");
127     }
128
129     if(datos[1].length()>2) {
130         datos[1]=datos[1].substring(1,datos[1].length()-1);
131         datos[1]=REM_HTMLTags(datos[1]);
132     } else {
133         datos[1]="no hay comentarios de la pregunta";
134     }
135     editText3.setText(datos[1]);
136
137
138     boton.setOnClickListener(
139     new View.OnClickListener() {
140     public void onClick(View view) {
141         Intent browserIntent = new Intent(Intent.ACTION_VIEW, Uri.
142 parse("http://192.168.112.2/app_examen_uib/assets/uploads/files/"+pregunta));
143         startActivity(browserIntent);
144     }
145     });
146
147     boton1.setOnClickListener(
148     new View.OnClickListener() {

```

8. ANEXOS

```
149         public void onClick(View view) {
150             Intent browserIntent = new Intent(Intent.ACTION_VIEW, Uri.
                parse("http://192.168.112.2/app_examen_uib/assets/uploads/files/"+datos[0].
                substring(0,datos[0].length()-1));
151                 startActivity(browserIntent);
152             }
153         }
154     );
155 }
156 }
157 }
```

layout

```
1
2 <?xml version="1.0" encoding="utf-8"?>
3 <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.
    android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     tools:context="com.example.ramon.clienteandroid.Main28Activity">
9
10 <TextView
11     android:id="@+id/textView19"
12     android:layout_width="69dp"
13     android:layout_height="17dp"
14     android:layout_marginBottom="8dp"
15     android:layout_marginLeft="8dp"
16     android:layout_marginStart="8dp"
17     android:layout_marginTop="8dp"
18     android:text="Pregunta"
19     android:textAppearance="@style/TextAppearance.AppCompat.Body2"
20     app:layout_constraintBottom_toBottomOf="parent"
21     app:layout_constraintLeft_toLeftOf="parent"
22     app:layout_constraintTop_toBottomOf="@+id/editText23"
23     app:layout_constraintVertical_bias="0.001" />
24
25 <EditText
26     android:id="@+id/editText23"
27     android:layout_width="0dp"
28     android:layout_height="wrap_content"
29     android:layout_marginBottom="8dp"
30     android:layout_marginEnd="8dp"
31     android:layout_marginLeft="8dp"
32     android:layout_marginRight="8dp"
33     android:layout_marginStart="8dp"
34     android:layout_marginTop="8dp"
35     android:editable="false"
36     android:ems="10"
37     android:enabled="false"
38     android:focusable="false"
39     android:inputType="textPersonName"
40     android:text="Name"
41     android:textColor="#00F"
```

```
42     app:layout_constraintBottom_toBottomOf="parent"
43     app:layout_constraintHorizontal_bias="0.0"
44     app:layout_constraintLeft_toLeftOf="parent"
45     app:layout_constraintRight_toRightOf="parent"
46     app:layout_constraintTop_toTopOf="parent"
47     app:layout_constraintVertical_bias="0.013" />
48
49 <EditText
50     android:id="@+id/editText24"
51     android:layout_width="wrap_content"
52     android:layout_height="wrap_content"
53     android:layout_marginBottom="8dp"
54     android:layout_marginEnd="8dp"
55     android:layout_marginLeft="8dp"
56     android:layout_marginRight="8dp"
57     android:layout_marginStart="8dp"
58     android:layout_marginTop="8dp"
59     android:editable="false"
60     android:ems="10"
61     android:enabled="false"
62     android:focusable="false"
63     android:inputType="textPersonName"
64     android:text="Name"
65     android:textColor="#0BF"
66     app:layout_constraintBottom_toBottomOf="parent"
67     app:layout_constraintHorizontal_bias="0.0"
68     app:layout_constraintLeft_toLeftOf="parent"
69     app:layout_constraintRight_toRightOf="parent"
70     app:layout_constraintTop_toBottomOf="@+id/textView19"
71     app:layout_constraintVertical_bias="0.0" />
72
73 <TextView
74     android:id="@+id/textView20"
75     android:layout_width="wrap_content"
76     android:layout_height="wrap_content"
77     android:layout_marginBottom="8dp"
78     android:layout_marginEnd="8dp"
79     android:layout_marginLeft="8dp"
80     android:layout_marginRight="8dp"
81     android:layout_marginStart="8dp"
82     android:layout_marginTop="8dp"
83     android:text="Solución"
84     android:textAppearance="@style/TextAppearance.AppCompat.Body2"
85     app:layout_constraintBottom_toBottomOf="parent"
86     app:layout_constraintHorizontal_bias="0.0"
87     app:layout_constraintLeft_toLeftOf="parent"
88     app:layout_constraintRight_toRightOf="parent"
89     app:layout_constraintTop_toBottomOf="@+id/editText24"
90     app:layout_constraintVertical_bias="0.0" />
91
92 <EditText
93     android:id="@+id/editText26"
94     android:layout_width="wrap_content"
95     android:layout_height="wrap_content"
96     android:layout_marginBottom="8dp"
97     android:layout_marginEnd="8dp"
98     android:layout_marginLeft="8dp"
```

```
99     android:layout_marginRight="8dp"
100     android:layout_marginStart="8dp"
101     android:layout_marginTop="8dp"
102     android:editable="false"
103     android:ems="10"
104     android:enabled="false"
105     android:focusable="false"
106     android:inputType="textPersonName"
107     android:text="Name"
108     android:textColor="#0BF"
109     app:layout_constraintBottom_toBottomOf="parent"
110     app:layout_constraintHorizontal_bias="0.0"
111     app:layout_constraintLeft_toLeftOf="parent"
112     app:layout_constraintRight_toRightOf="parent"
113     app:layout_constraintTop_toBottomOf="@+id/textView20"
114     app:layout_constraintVertical_bias="0.0" />
115
116 <Button
117     android:id="@+id/button5"
118     android:layout_width="wrap_content"
119     android:layout_height="wrap_content"
120     android:layout_marginBottom="8dp"
121     android:layout_marginEnd="8dp"
122     android:layout_marginLeft="8dp"
123     android:layout_marginRight="8dp"
124     android:layout_marginStart="8dp"
125     android:layout_marginTop="8dp"
126     android:text="Abrir"
127     app:layout_constraintBottom_toBottomOf="parent"
128     app:layout_constraintHorizontal_bias="0.0"
129     app:layout_constraintLeft_toRightOf="@+id/editText24"
130     app:layout_constraintRight_toRightOf="parent"
131     app:layout_constraintTop_toBottomOf="@+id/textView19"
132     app:layout_constraintVertical_bias="0.0" />
133
134 <Button
135     android:id="@+id/button8"
136     android:layout_width="wrap_content"
137     android:layout_height="wrap_content"
138     android:layout_marginBottom="8dp"
139     android:layout_marginEnd="8dp"
140     android:layout_marginLeft="8dp"
141     android:layout_marginRight="8dp"
142     android:layout_marginStart="8dp"
143     android:layout_marginTop="8dp"
144     android:text="Abrir"
145     app:layout_constraintBottom_toBottomOf="parent"
146     app:layout_constraintHorizontal_bias="0.0"
147     app:layout_constraintLeft_toRightOf="@+id/editText26"
148     app:layout_constraintRight_toRightOf="parent"
149     app:layout_constraintTop_toBottomOf="@+id/textView20"
150     app:layout_constraintVertical_bias="0.0" />
151
152 <EditText
153     android:id="@+id/editText25"
154     android:layout_width="562dp"
155     android:layout_height="415dp"
```



```

156     android:layout_marginBottom="8dp"
157     android:layout_marginEnd="8dp"
158     android:layout_marginLeft="8dp"
159     android:layout_marginRight="8dp"
160     android:layout_marginStart="8dp"
161     android:layout_marginTop="8dp"
162     android:editable="false"
163     android:ems="10"
164     android:enabled="false"
165     android:focusable="false"
166     android:inputType="textMultiLine"
167     app:layout_constraintBottom_toBottomOf="parent"
168     app:layout_constraintHorizontal_bias="0.0"
169     app:layout_constraintLeft_toLeftOf="parent"
170     app:layout_constraintRight_toRightOf="parent"
171     app:layout_constraintTop_toBottomOf="@+id/textView22"
172     app:layout_constraintVertical_bias="0.0" />
173
174 <TextView
175     android:id="@+id/textView22"
176     android:layout_width="wrap_content"
177     android:layout_height="wrap_content"
178     android:layout_marginBottom="8dp"
179     android:layout_marginEnd="8dp"
180     android:layout_marginLeft="8dp"
181     android:layout_marginRight="8dp"
182     android:layout_marginStart="8dp"
183     android:layout_marginTop="8dp"
184     android:text="Comentarios"
185     android:textAppearance="@style/TextAppearance.AppCompat.Body2"
186     app:layout_constraintBottom_toBottomOf="parent"
187     app:layout_constraintHorizontal_bias="0.0"
188     app:layout_constraintLeft_toLeftOf="parent"
189     app:layout_constraintRight_toRightOf="parent"
190     app:layout_constraintTop_toBottomOf="@+id/editText26"
191     app:layout_constraintVertical_bias="0.0" />
192 </android.support.constraint.ConstraintLayout>

```

8.4 Aplicación Servidor

```

1 package server;
2 import java.io.BufferedReader;
3 import java.io.IOException;
4 import java.io.InputStreamReader;
5 import java.io.OutputStream;
6 import java.net.ServerSocket;
7 import java.net.Socket;
8 import java.util.ArrayList;
9 import java.util.Collections;
10 import java.net.*;
11 import java.sql.*;
12 import java.util.List;
13 import java.util.logging.Level;
14 import java.util.logging.Logger;
15

```

```
16
17 public class MainServer {
18
19
20     private final static int PORT = 5000;
21     Connection co;
22     Statement stm;
23
24
25     private static String DataToSend(ArrayList Data)
26     {
27         String DataToSend = "";
28         int i;
29
30         if (Data.isEmpty() == false) {
31             for (i=0; i<Data.size()-1; i++) {
32                 DataToSend = DataToSend + Data.get(i) + ",";
33             }
34             DataToSend += Data.get(Data.size()-1);
35         }
36         return(DataToSend);
37     }
38
39     public static void main(String[] args) {
40
41         try {
42             //Socket de servidor para esperar peticiones de la red
43             ServerSocket serverSocket = new ServerSocket(PORT);
44             System.out.println("Servidor> Servidor iniciado");
45             System.out.println("Servidor> En espera de cliente...");
46             //Socket de cliente
47             Socket clientSocket;
48             while(true) {
49                 //en espera de conexion, si existe la acepta
50                 clientSocket = serverSocket.accept();
51                 //Para leer lo que envie el cliente
52                 BufferedReader input = new BufferedReader(new InputStreamReader(
clientSocket.getInputStream()));
53                 //para imprimir datos de salida
54                 PrintStream output = new PrintStream(clientSocket.getOutputStream
());
55                 //se lee peticion del cliente
56
57                 String request = input.readLine();
58                 System.out.println("Cliente> petición [" + request + "]);
59                 //se procesa la peticion y se espera resultado
60
61
62
63                 ArrayList str_BD = conexionSQL(request);
64                 //Se imprime en consola "servidor"
65                 System.out.println("Servidor> Resultado de petición");
66
67                 //System.out.println("Servidor> \" + str_BD + "\");
68                 //System.out.println("Servidor> " + str_BD + "");
69                 System.out.println("Servidor> " + DataToSend(str_BD));
70
```

```

71         //se imprime en cliente
72         output.flush(); //vacía contenido
73         // output.println(str_BD);
74         output.println(DataToSend(str_BD));
75
76         //cierra conexión
77         clientSocket.close();
78     }
79     } catch (IOException ex) {
80         System.err.println(ex.getMessage());
81     }
82 }
83
84
85 public static ArrayList conexionSQL(String request) {
86     String result = null;
87     ArrayList<String> respuesta = new ArrayList<String>();
88     String orden = request.substring(0,5);
89     try {
90         Class.forName("com.mysql.jdbc.Driver");
91         Connection co = DriverManager.getConnection("jdbc:mysql://localhost/
test_database","root","");
92         Statement stm = co.createStatement();
93
94         switch(orden) {
95
96             case "MAC10":
97
98                 String usuario = request.substring(5,11);
99                 String contraseña = request.substring(11,request.length());
100
101                 ResultSet login=stm.executeQuery("SELECT nombre, apellido1 ,
apellido2 FROM alumnos WHERE nom_usuario_alumn =' " + usuario + "' AND
password='"+contraseña+"'");
102                 if (login.next()) {
103                     result="OK#" +login.getString(1)+" "+login.getString(2)+" "+
login.getString(3);
104                     respuesta.add(result);
105                 }
106                 else {
107                     respuesta.add("Usuario no valido#");
108                 }
109
110                 break;
111
112             case "MAC20":
113                 String user = request.substring(5,11);
114                 ResultSet asign=stm.executeQuery("select * from vw_cursa WHERE
nom_usuario_alumn ='"+ user +"' and ano_academico>=year(now())");
115
116                 while (asign.next()) {
117                     result=asign.getString(4)+" "+asign.getString(5);
118                     respuesta.add(result);
119                 }
120
121                 break;
122

```

8. ANEXOS

```
123         case "MAc22":
124             String [] variable = request.split("#");
125             ResultSet prof=stm.executeQuery("select * from vw_imparte where
cod_asignatura='"+variable[1]+' and ano_academico>=year(now())");
126
127             while(prof.next()){
128                 result=prof.getString(3)+" "+prof.getString(4)+" "+prof.
getString(5);
129                 respuesta.add(result);
130             }
131
132             respuesta.add("#"); //separador
133             ResultSet exa=stm.executeQuery("select nom_examen from
vw_modelos_preguntas_alumnos where cod_asignatura='"+variable[1]+' and Year(
fecha_examen)>=Year(now()) and nom_usuario_alumn ='"+variable[2]+' group by
fecha_examen order by fecha_examen");
134
135             while(exa.next()){
136                 result=exa.getString(1);
137                 respuesta.add(result);
138             }
139
140             respuesta.add("#"); //separador
141             ResultSet fch_e=stm.executeQuery("select date_format(fecha_examen
, '%Y-%m-%d %H%i ') from vw_modelos_preguntas_alumnos where cod_asignatura='"+
+variable[1]+' and Year(fecha_examen)>=Year(now()) and nom_usuario_alumn ='
+variable[2]+' group by fecha_examen order by fecha_examen");
142
143             while(fch_e.next()){
144                 result=fch_e.getString(1);
145                 respuesta.add(result);
146             }
147
148             break;
149
150         case "MAc23": //examnes
151             String [] examen = request.split("#");
152             ResultSet enun=stm.executeQuery("select distinct enunciado from
vw_modelos_preguntas_alumnos where nom_examen='"+examen[1]+' and
cod_asignatura='"+examen[2]+' and fecha_examen ='"+examen[3]+' and
nom_usuario_alumn="'+examen[4]+' order by num_pregunta");
153
154             while(enun.next()){
155                 result=enun.getString(1);
156                 respuesta.add(result);
157             }
158
159             respuesta.add("#"); //separador
160             ResultSet fch_rev_=stm.executeQuery("select IFNULL(date_format(
fecha_revision , '%Y-%m-%d %H%i '), 'NULL') from vw_modelos_preguntas where
nom_examen='"+examen[1]+' and cod_asignatura='"+examen[2]+' and
fecha_examen ='"+examen[3]+' group by fecha_examen");
161
162             while(fch_rev_.next()){
163                 result=fch_rev_.getString(1);
164                 respuesta.add(result);
165             }
```

```

166         respuesta.add("#"); //separador
167         ResultSet com_generales=stm.executeQuery("select IFNULL(
168 coment_generales,'NULL') from vw_examenes where nom_examen='"+examen[1]+' '
and cod_asignatura='"+examen[2]+' ' and fecha_examen =' '+examen[3]+' 'group by
fecha_examen");

169
170         while (com_generales.next()) {
171             result=com_generales.getString(1);
172             respuesta.add(result);
173         }
174
175         break;
176
177         case "Mac24":
178             String[] separada = request.split("#");
179             ResultSet prof_correc=stm.executeQuery("select nombre, apellido1,
apellido2 from vw_modelos_preguntas_alumnos where nom_usuario_alumn='"+
separada[1]+' ' AND nom_examen='"+separada[2]+' ' AND enunciado =' '+separada
[3]+' ' and fecha_examen='"+separada[4]+' '");

180
181             while (prof_correc.next()) {
182                 result=prof_correc.getString(1)+" "+prof_correc.getString(2)+
" "+prof_correc.getString(3);
183                 respuesta.add(result);
184             }
185
186             respuesta.add("#"); //separador
187             ResultSet comentario=stm.executeQuery("select IFNULL(
coment_individuales,'NULL') from vw_modelos_preguntas_alumnos where
nom_usuario_alumn='"+separada[1]+' ' AND nom_examen='"+separada[2]+' ' AND
enunciado =' '+separada[3]+' ' and fecha_examen='"+separada[4]+' '");

188
189             while (comentario.next()) {
190                 result=comentario.getString(1);
191                 respuesta.add(result);
192             }
193
194             respuesta.add("#"); //separador
195             ResultSet sol_alumno=stm.executeQuery("select IFNULL(sol_alumno, '
NULL') from vw_modelos_preguntas_alumnos where nom_usuario_alumn='"+separada
[1]+' ' AND nom_examen='"+separada[2]+' ' AND enunciado =' '+separada[3]+' ' and
fecha_examen='"+separada[4]+' '");

196
197             while (sol_alumno.next()) {
198                 result=sol_alumno.getString(1);
199                 respuesta.add(result);
200             }
201
202             respuesta.add("#"); //separador
203             ResultSet correc_prof=stm.executeQuery("select IFNULL(
correc_profesor,'NULL') from vw_modelos_preguntas_alumnos where
nom_usuario_alumn='"+separada[1]+' ' AND nom_examen='"+separada[2]+' ' AND
enunciado =' '+separada[3]+' ' and fecha_examen='"+separada[4]+' '");

204
205             while (correc_prof.next()) {
206                 result=correc_prof.getString(1);

```

```

207         respuesta.add(result);
208     }
209
210     respuesta.add("#"); //separador
211     ResultSet comentario_general=stm.executeQuery("select IFNULL(
coment_generales,'NULL') from vw_modelos_preguntas where nom_examen='"+
separada[2]+" ' AND enunciado ='"+separada[3]+" ' and fecha_examen='"+separada
[4]+" ' group by fecha_examen");
212
213     while(comentario_general.next()){
214         result=comentario_general.getString(1);
215         respuesta.add(result);
216     }
217
218     respuesta.add("#"); //separador
219     ResultSet sol_profesor=stm.executeQuery("select IFNULL(
sol_profesor,'NULL') from vw_modelos_preguntas where nom_examen='"+separada
[2]+" ' AND enunciado ='"+separada[3]+" ' and fecha_examen='"+separada[4]+" '
group by fecha_examen");
220
221     while(sol_profesor.next()){
222         result=sol_profesor.getString(1);
223         respuesta.add(result);
224     }
225
226     break;
227
228     case "Mac25":
229     String [] variable_25 = request.split("#");
230     Statement ex_ant = co.createStatement();
231
232     ResultSet exam=ex_ant.executeQuery("select nom_examen from
vw_modelos_preguntas_alumnos where cod_asignatura='"+variable_25[1]+" ' and
Year(fecha_examen)<Year(now()) and nom_usuario_alumn ='"+variable_25[2]+" '
group by fecha_examen order by fecha_examen");
233
234     while(exam.next()){
235         result=exam.getString(1);
236         respuesta.add(result);
237     }
238
239
240     respuesta.add("#"); //separador
241     ResultSet fch_ant=stm.executeQuery("select date_format(
fecha_examen, '%Y-%m-%d %T%i ') from vw_modelos_preguntas_alumnos where
cod_asignatura='"+variable_25[1]+" ' and Year(fecha_examen)<Year(now()) and
nom_usuario_alumn ='"+variable_25[2]+" 'group by fecha_examen order by
fecha_examen");
242
243     while(fch_ant.next()){
244         result=fch_ant.getString(1);
245         respuesta.add(result);
246     }
247     break;
248
249     case "Mac27": //profesores y examnes
250     Statement pr_ex = co.createStatement();

```

```

251     String [] pr_exa_ant = request.split("#");
252     pr_exa_ant[2]=pr_exa_ant[2].substring(0,5);
253     ResultSet preg=pr_ex.executeQuery("select enunciado from
vw_modelos_preguntas where cod_asignatura='"+pr_exa_ant[2]+'and nom_examen='
"+pr_exa_ant[1]+'group by enunciado");

254
255     while(preg.next()){
256         result=preg.getString(1);
257         respuesta.add(result);
258     }
259
260     respuesta.add("#"); //separador
261     ResultSet com=pr_ex.executeQuery("select coment_generales from
vw_examenes where cod_asignatura='"+pr_exa_ant[2]+'and nom_examen=''+
pr_exa_ant[1]+'group by nom_examen");

262
263     while(com.next()){
264         result=com.getString(1);
265         respuesta.add(result);
266     }
267
268     respuesta.add("#"); //separador
269     ResultSet fch_examen=pr_ex.executeQuery("select date_format(
fecha_examen, '%Y-%m-%d %t%i ') from vw_modelos_preguntas where
cod_asignatura='"+pr_exa_ant[2]+'and nom_examen='"+pr_exa_ant[1]+'group by
fecha_examen");

270
271     while(fch_examen.next()){
272         result=fch_examen.getString(1);
273         respuesta.add(result);
274     }
275
276     break;

277
278     case "MAc28":
279     Statement sol_pr = co.createStatement();
280     String [] sol_pr_ant = request.split("#");
281     ResultSet sol_pregunta=sol_pr.executeQuery("select sol_profesor
from vw_modelos_preguntas where cod_asignatura='"+sol_pr_ant[2]+'and
enunciado='"+sol_pr_ant[1]+'and fecha_examen='"+sol_pr_ant[3]+'group by
sol_profesor");

282
283     while(sol_pregunta.next()){
284         result=sol_pregunta.getString(1);
285         respuesta.add(result);
286     }
287
288     respuesta.add("#"); //separador
289     ResultSet comentarios=sol_pr.executeQuery("select
coment_generales from vw_modelos_preguntas where cod_asignatura='"+sol_pr_ant
[2]+'and enunciado='"+sol_pr_ant[1]+'and fecha_examen='"+sol_pr_ant[3]+'
group by fecha_examen");

290
291     while(comentarios.next()){
292         result=comentarios.getString(1);
293         respuesta.add(result);
294     }

```

8. ANEXOS

```
295         break;  
296     }  
297 }  
298 catch(ClassNotFoundException ex) {  
299     ex.printStackTrace();  
300 }  
301 catch(SQLException e) {  
302     Logger.getLogger(MainServer.class.getName()).log(Level.SEVERE, null, e)  
303 ;  
304 }  
304 return respuesta ;  
305 }  
306 }
```


BIBLIOGRAFÍA

- [1] J. DiMarzio, *Beginning Android Programming with Android Studio*. WROX PRESS, 2016.
- [2] J. Valade, *PHP Y MYSQL para Dummies*, 2nd ed.
- [3] B. Eckel, *Piensa en Java*, 2nd ed. Prentice Hall, 2013.
- [4] “(8 de Julio de 2017) Web de Grocery CRUD.” [Online]. Available: <https://www.grocerycrud.com/>
- [5] “(5 de Mayo de 2017) Web de la Escola Politècnica Superior.” [Online]. Available: <http://eps.uib.es/>
- [6] “(8 de Febrero de 2017) Fundación de MariaDB.” [Online]. Available: <https://mariadb.org/>
- [7] “(8 de Febrero de 2017) MD5 en Java.” [Online]. Available: <https://stackoverflow.com/questions/415953/how-can-i-generate-an-md5-hash>
- [8] “(12 de Abril de 2017) Tutoriales Java.” [Online]. Available: <https://www.tutorialspoint.com/index.htm>