**Informe de Actividades**

**Sistemas Multimedia Basados en Internet**

**Versión: 1**

**Fecha de entrega: 25/10/2016**

**Ututui, George Valentin Cristian**

**Candela Ibáñez, Antonio**

Index

[1. Instalar Apache, MySQL y PHP 3](#_Toc465179026)

[1.1 Seguridad MySQL 3](#_Toc465179027)

[1.2 MySQLTuner 3](#_Toc465179028)

[1.3 PHP 3](#_Toc465179029)

[3. Especificación del sistema 4](#_Toc465179030)

[2.1 Arquitectura del Sistema 4](#_Toc465179031)

[a. Esquema ER 4](#_Toc465179032)

[2.3 Tablas del Sistema 5](#_Toc465179034)

[4. Tecnologías 9](#_Toc465179035)

# Instalar Apache, MySQL y PHP

Apache es un servidor web HTTP que es muy configurable ya que permite bases de datos de autenticación y también negociado de contenido. Para la instalación del mismo hemos utilizado el comando: *sudo apt install apache2.*

MySQL es la base de datos relacional open source mas utilizada y conocida del mundo que se utiliza sobre todo para entornos web, como es nuestro caso. Para la instalación del MySQL hemos utilizado el comando: *sudo apt install mysql\_server.*

## 1.1 Seguridad MySQL

A través del comando *sudo mysql\_secure\_installation* aumentamos la seguridad de nuestro MySQL de las siguientes maneras:

* Se ha aumentado el nivel de seguridad de la contraseña y se añadió una contraseña al usuario Root.
* Se ha configurado MySQL de forma que solo se pueda acceder al MySQL desde el localhost.
* Se han eliminado los usuarios anónimos que pueden acceder a MySQL y vienen por defecto.
* Así mismo se ha eliminado la tabla ‘test’ que viene de serie con MySQL

## MySQLTuner

MySQLTuner es un script que nos permite revisar la instalación de MySQL y hacer los ajustes que consideremos necesarios para aumentar el rendimiento y la escalabilidad. La configuración actual y los datos del estado se presentan en un formato breve y esquemático con sugerencias al hacer el comando *mysqltuner.*

En términos de Métricas de Rendimiento el Tuner nos propone habilitar la Cache de nuestra MySQL. Esto lo valoraremos a lo largo de esta semana y decidiremos si nos conviene o no habilitar la Caché. También hay que decir que entre otras cosas nos advierten de cosas relacionadas con InnoBD.

Para mejores recomendaciones de este Script se recomienda dejar en funcionamiento MySQL para ver cómo se comporta y funciona en base a nuestras exigencias de este sistema de gestión de bases de datos.

## PHP

Es un lenguaje muy conocido para el desarrollo web que tiene la característica de poderse incrustar en HTML. En la versión 16 de Ubuntu se emplea PHP7. Para instalar PHP en Ubuntu hemos utilizado el comando: *sudo apt-get install php.*

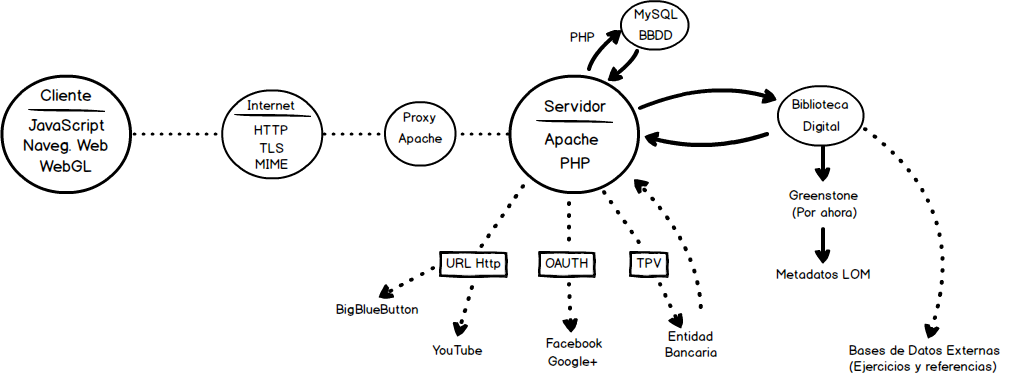
Lo siguiente fue instalar el módulo para crear el vínculo entre Apache y PHP: *sudo apt-get install libapache2-mod-php5.*

# Especificación del sistema

En los documentos de la asignatura de Proyectos Multimedia se ven reflejados los elementos y las tecnologías de seguridad y autentificación que utilizaremos para la creación de nuestro sistema. También se especifican todos los elementos del back-end, las tecnologías y como se relacionan cada uno de estos componentes entre sí. A continuación, vamos a presentar los elementos más importantes en las que hemos estado trabajando estás dos semanas.

## 2.1 Arquitectura del Sistema

Se puede observar como los dos principales ‘actores’ en este esquema son el Cliente y el Servidor. Con el servidor se relacionan la mayoría de los sistemas y tecnologías para ofrecer al Cliente el servicio que estamos intentando ofrecer.



## Esquema ER

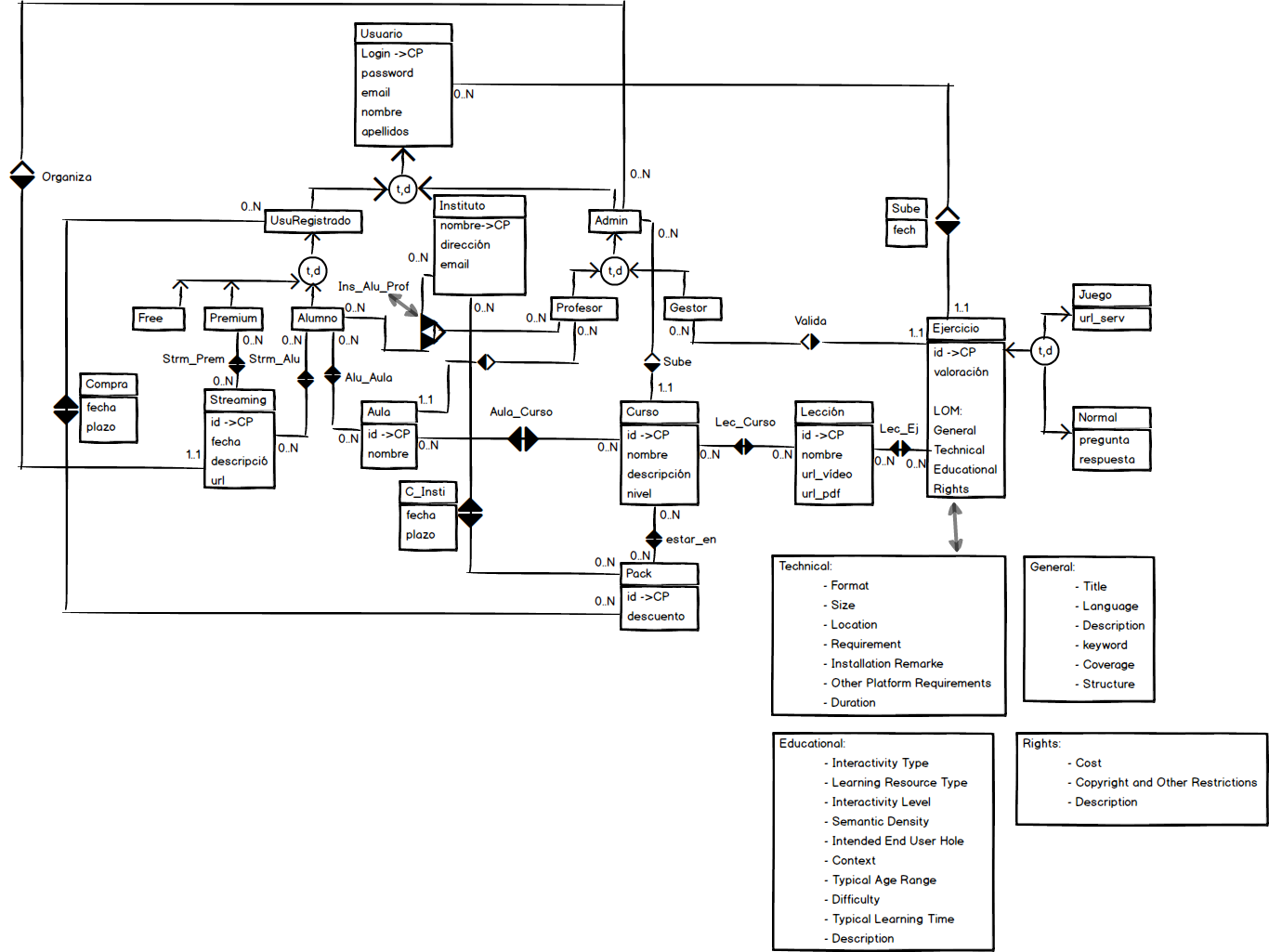
Mediante el modelo Entidad-Relación podemos representar las entidades más importantes que conforman nuestro sistema junto con sus relaciones y propiedades.

Ya que la foto del esquema ER es muy grande no se pueden apreciar los del mismo, por lo que junto a este documento esta adjunta una foto con el diagrama.

**Aclaraciones del modelo Entidad Relación:**

1. Una lección debe tener al menos un curso, pero en el modelo relacional no se puede representar una relación 0..N-1..N, cumpliremos está restricción internamente mediante código.
2. También especificamos mediante código el hecho de que un usuario Premium no puede acceder a los streaming de un profesor.
3. La entidad ‘Instituto’ también representa academias, por eso un profesor puede estar relacionado con un alumno desde su academia y su instituto en el caso de permanecer ambos a la vez.
4. El LOM representado en el diagrama serán los metadatos que debemos incluir al subir los ejercicios a nuestra biblioteca digital. Esto lo haremos para cumplir es estándar SCROM.

### 



## 2.3 Tablas del Sistema

Estás son las tablas resultantes del esquema ER que van a representar nuestro sistema en MySQL.

* Usuario(Login,password,email,nombre,apellidos)

CP : Login

* Usuario Registrado(Login)

CP : Login

C. Aj. : Login -> Usuario

* Admin(login)

CP : Login

C. Aj. : Login -> Usuario

* Free(Login)

CP: Login

C. Aj. : Login -> Usuario Registrado

* Premium(Login)

CP: Login

C. Aj. : Login -> Usuario Registrado

* Alumno(Login)

CP: Login

C. Aj. : Login -> Usuario Registrado

* Profesor(Login)

CP : Login

C. Aj. : Login -> Admin

* Gestor(Login)

CP : Login

C. Aj. : Login -> Admin

* Streaming(id,fecha,descripcion,url,login\_adm)

CP: id

C. Aj. : login\_adm -> Admin

VNN: login\_adm

* Instituto(nombre, direccion, email)

CP: Nombre

* Aula(id,nombre,login\_profe)

CP: id

C. Aj. : login\_profe -> Profesor

VNN: login\_profe

* Curso(id, nombre, descripcion, nivel, lodin\_adm)

CD: id

C. Aj. : login\_adm ->Admin

VNN: login\_adm

* Leccion(id,nombre,url\_video,url\_pdf)

CP: id

* Ejercicio(id, valoracion, login\_usu, login\_ges)

CP: id

C. Aj. : login\_usu -> Usuario

C. Aj. : login\_ges -> Gestor

VNN: login\_usu, login\_ges

* Juego(id,url\_serv)

CP: id

C. Aj. : id ->Ejercicio

* Normal(id, pregunta, respuesta)

CP: id

C. Aj. : id -> Ejercicio

* Pack(id, descuento)

CP: id

A continuación, vamos a presentar las tablas que surgen de relaciones Muchos a Muchos o tablas que han surgido por la necesidad de guardar un dato en concreto al relacionar dos entidades.

* Sube(login\_usu, id\_ej, fecha)

CP: login\_usu

C. Aj. : Login\_usu -> Usuario

C. Aj. : login\_ej -> Ejercicio

VNN: id\_ej

* Leccion\_ej(id\_lec, id\_ej)

CP(id\_lej, id\_ej)

C. Aj. : id\_lec -> Leccion

C. Aj. : id\_ej -> Ejercicio

* Curso\_lec(id\_curso, id\_lec)

CP(id\_curso, id\_lecc)

C. Aj. : id\_curso -> Curso

C. Aj. : id\_lec -> Leccion

* Comprar(fecha, plazo, login\_usu, id\_pack)

CP(id\_pack, login\_usu)

C. Aj. : id\_pack -> Pack

C. Aj. : login\_usu -> Usuario

* C\_Insti(fecha, plazo, nombre\_inti, id\_pack)

CP(id\_pack, nombre\_insti)

C. Aj. : id\_pack -> Pack

C. Aj. : nombre\_insti -> Instituto

* Estar\_en(id\_pack, id\_curso)

CP(id\_pack, id\_curso)

C. Aj. : id\_pack -> Pack

C. Aj. : id\_curso -> Curso

* Ins\_alu\_prof(nombre\_insti, login\_alu, login\_prof)

CP(nombre\_insti, login\_alu)

C. Aj. : nombre\_insti -> Instituto

C. Aj. : login\_alu -> Alumno

C. Aj. : login\_prof -> Profesor

VNN: login\_prof

* Alumn\_aula(login\_alumno, id\_aula)

CP(login\_alumno,id\_aula)

C. Aj. : login\_alumno -> Alumno

C. Aj. : id\_aula -> Aula

* Aula\_curso(id\_aula, id\_curso)

CP(id\_aula, id\_curso)

C. Aj. : id\_aula -> Aula

C. Aj. : id\_curso -> Curso

* Prem\_strm(login\_premi, id\_stream)

CP(login\_premi, id\_stream)

C. Aj. : login\_premi -> Premium

C. Aj. : id\_stream -> Streaming

* Alumn\_strm(login\_alu, id\_stream)

CP(login\_alu, id\_stream)

C. Aj. : lodin\_alu -> Alumno

C. Aj. : id\_stream -> Streming

# Tecnologías

* Nuestro servidor funcionará mediante Apache, javascript y php. Estudiamos la utilización de Node.js en un futuro.
* Usaremos una base de datos relacional SQL.
* Utilizaremos esclavos y discos duros extraíbles para el almacenamiento de copias de seguridad del contenido de nuestra página.
* Emplearemos SEASLOG para manejar la contabilidad de nuestra web.
* Para el tema de gráficos 3D, utilizaremos WebGL.
* Utilizaremos OAUTH para relacionar las cuentas de Facebook y Google+ con los perfiles de los usuarios.
* Los videos de nuestra plataforma estarán disponibles en Youtube. Los streamings se harán mediante la herramienta BigBlueButton.
* Utilizaremos comunicación REST y https.
* Seguiremos el estándar IEEE LOM para catalogar nuestros materiales educativos.
* Usaremos el estándar de empaquetamiento SCORM, por lo tanto deberemos adaptar nuestra base de datos para almacenar ficheros de este estilo. A su vez utilizaremos Reload como editor de SCORM.
* Para gestionar nuestra biblioteca digital utilizaremos Greenstone o DSpace.
* Emplearemos xPerience para poder obtener fácilmente las estadísticas que necesitemos obtener de nuestros ejercicios.
* También haremos uso de MERLOT para buscar materiales educativos en bases de datos ya creadas por otras instituciones.