

Mayo ' 17

# MELÓMANOS



Estíbaliz Busto Pérez de Mendiguren

Ampliación de Base de Datos

3º C

UNIVERSIDAD COMPLUTENSE  
FACULTAD DE INFORMÁTICA  
AMPLIACIÓN DE BASE DE DATOS

# MELÓMANOS

CONSTRUCCIÓN DE UN SITIO WEB DE MENSAJERÍA PARA MELÓMANOS

Madrid, 29 de mayo de 2017  
Facultad de Informática  
Grado Ingeniería Informática  
3º C



## CONTENIDO

Prólogo .....	IV
1. Introducción .....	1
2. Herramientas utilizadas.....	2
1) MySQL .....	2
2) PHP .....	2
3) jQuery.....	2
4) jQuery UI.....	3
5) CSS .....	3
6) JavaScript.....	3
7) HTML .....	3
3. Arquitectura .....	5
Tabla de usuarios.....	6
Tabla de mensajes .....	7
Tabla de grupos .....	7
Base de datos de melómanos .....	8
4. Conclusiones.....	9



## PRÓLOGO

El presente trabajo lleva el título de “Melómanos”. Esta memoria es una parte de la asignatura Ampliación de Bases de Datos impartida por el profesor D. Ramón González del Campo Rodríguez Barbero de la Universidad Complutense de Madrid (UCM).

El periodo de creación del código fuente y redacción de la presente memoria ha sido de ocho semanas (desde el 3 de abril hasta el 31 de mayo de 2017). Durante este tiempo he podido buscar información y entender diferentes códigos indispensables para realizar la presente práctica.

La práctica se ha llevado a cabo a través del estudio de diferentes diapositivas proporcionadas por el profesor anteriormente mencionado, que, desde un principio, ha estado dispuesto a ayudarme en todas mis dudas.

Me gustaría, por tanto, dar las gracias a mi supervisor por su excelente orientación para arrancar en el proceso de realización de mi primera construcción de un sitio web de mensajería para melómanos.

Espero que merezca la pena mi trabajo y esfuerzo realizado.

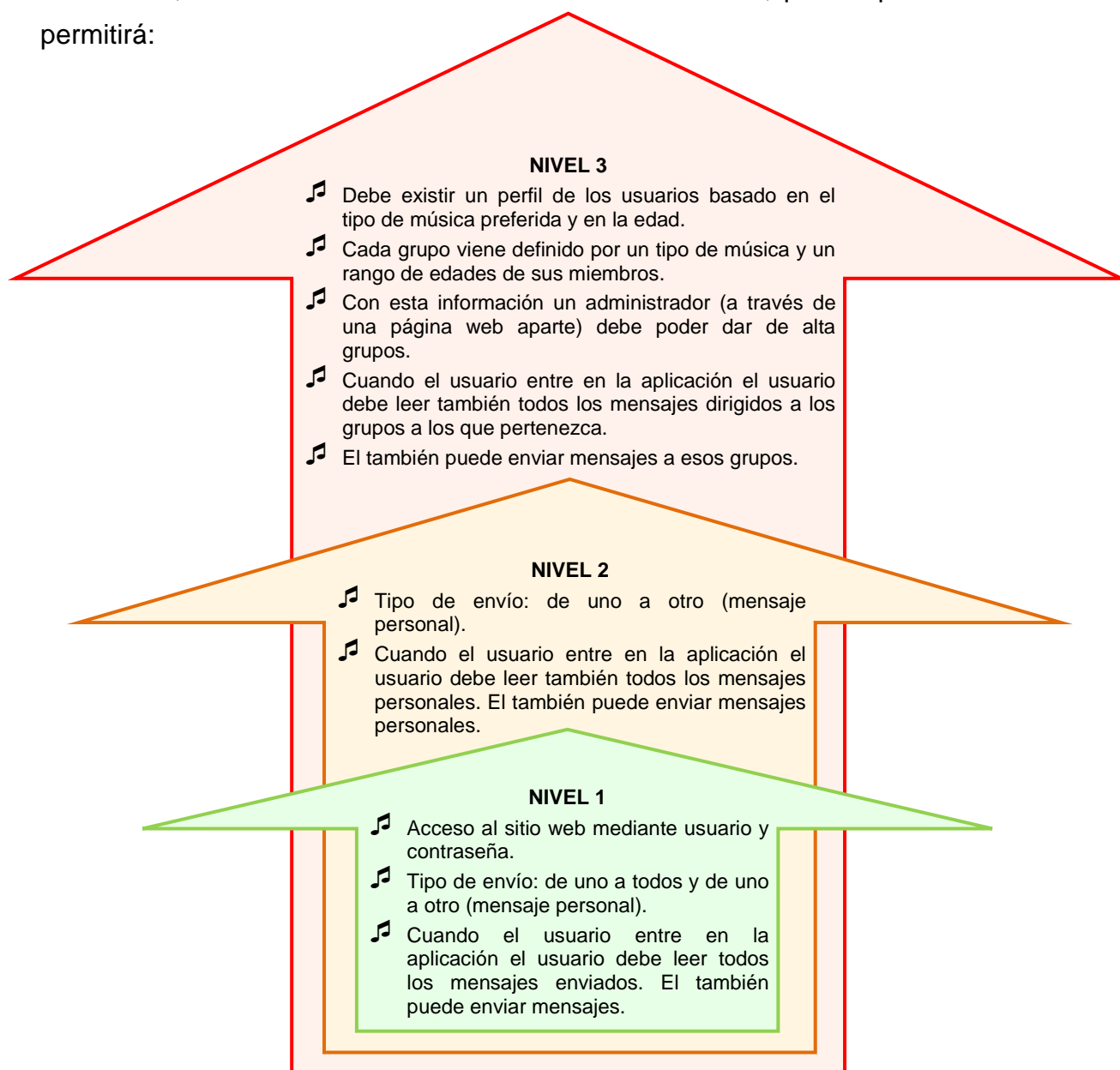
*Estíbaliz Busto Pérez de Mendiguren*



## 1. INTRODUCCIÓN

El objetivo de la práctica es la construcción de un sitio web de mensajería para melómanos. Este proyecto nos permite realizar varias acciones correspondientes a tres diferentes niveles de especificación, tal y como se explica en el enunciado de la práctica.

Siendo así, he decidido realizar los tres niveles definidos; por lo que mi sitio web permitirá:





## 2. HERRAMIENTAS UTILIZADAS

En este proyecto he utilizado las tecnologías siguientes basadas en web: MySQL, PHP, jQuery, jQuery UI, CSS, JavaScript y Html. He aquí, las ideas más relevantes de cada una de ellas:

### 1) MySQL

**Sistema de administración de bases de datos** (Database Management System, DBMS) para bases de datos relacionales.

Así, MySQL no es más que una aplicación que permite gestionar archivos llamados de bases de datos. Por lo que, tenemos una sintaxis que nos permite hacer peticiones (select, update, insert, delete...).



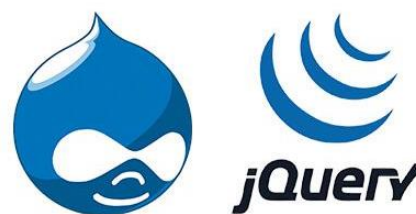
### 2) PHP



Lenguaje de **código abierto** muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. Este lenguaje, orientado a programación de servidores web, tiene facilidad para conectarse tanto con HTML como MySQL. No se utiliza objetos de PHP, sólo se utilizan **funciones**.

### 3) jQuery

**Biblioteca multiplataforma de JavaScript**, creada inicialmente por John Resig, permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX





a páginas web.

#### 4) jQuery UI

Biblioteca de componentes para el framework jQuery que le añaden un conjunto de **plug-ins, widgets y efectos visuales** para la creación de aplicaciones web.

Cada componente o módulo se desarrolla de acuerdo a la filosofía de jQuery (find something, manipulate it: encuentra algo, manipúlalo).



#### 5) CSS

Lenguaje utilizado en la presentación de documentos HTML. Un documento HTML viene siendo coloquialmente “una página web”. Entonces podemos decir que el lenguaje **CSS** sirve para organizar la **presentación y aspecto** de una página web.



#### 6) JavaScript



Lenguaje de programación que se utiliza principalmente para crear **páginas web dinámicas**. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.

#### 7) HTML

**HyperText Markup Language** (lenguaje de marcas de hipertexto), hace referencia al **lenguaje de marcado** para la elaboración de **páginas web**. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en



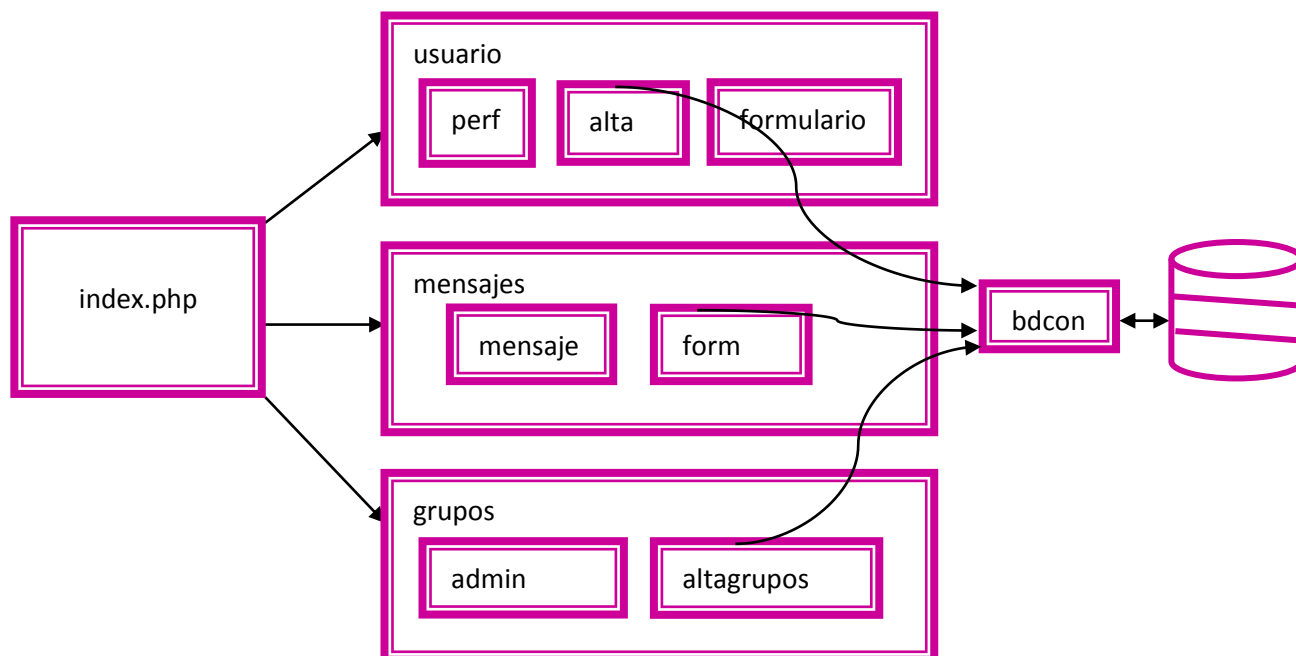
sus diferentes versiones, define una estructura básica y un código para la definición de contenido de una página web, como **texto, imágenes, videos, juegos**, entre otros.

Se considera el lenguaje web más importante y el estándar que se ha impuesto en la visualización de páginas web, que todos los navegadores actuales han adoptado.





### 3. ARQUITECTURA



Con **PHPy HTML** se hace una infraestructura que se mantiene el **index** y es el que se encarga de llamar al resto de archivos que tendrán el contenido. La gestión del usuario actual se realiza mediante el guardado en **cookies** en la variable de sesión de **php** y que se destruye al hacer *logout* o al salir del navegador.

Los mensajes se pueden escribir de tres maneras, para escribirlos se da al usuario a elegir entre la lista de personas, incluido el mismo y la lista de grupos a los que el pertenece. En el perfil puede consultar sus datos, aunque no los pueda editar y en la de ver mensajes tiene un listado por cada tipo de mensaje y por cada grupo dentro del tipo grupo.





Con el archivo **bdcon.php** he separado toda la infraestructura de llamada a la base de datos en un archivo aparte que se llama **bdcon** que es conexión con la *BD* que se encarga de hacer las peticiones.

Existen tres tipos:

### *Tabla de usuarios*

Las funciones que utilizo son: dar de alta un usuario (**altaUser**), comprobar si existe el usuario mediante el email (**existsUser**), verificar un usuario si su contraseña y nombre están bien (**verifyUser**), obtener los datos del usuario (**getUser**), obtener el nombre de un usuario para poder escribirle un mensaje (**getUserName**) y obtener la lista de usuarios a los que posteriormente le podré escribir un mensaje (**getListUsers**).

```
function altaUser($email,$pass,$nombre,$tipomusica,$edad){
    if(existsUser($email))
        return 1;
    $query = "INSERT INTO ".TABLE_USUARIOS." (nombre,email,password,tipomusica,edad) ";
    $query.="VALUES ('".$nombre."', '".$email."', '".$pass."', '".$tipomusica."', '".$edad."')";
    doQuery($query);
    if(verifyUser($email,$pass)==1)
        return 0;
    else
        return 2;
}

function existsUser($email){
    $query = "SELECT * FROM ".TABLE_USUARIOS." WHERE email='".$email."';";
    $count = doQuery($query);
    if($count->num_rows>0){
        return true;
    }else{
        return false;
    }
}

function verifyUser($email,$pass){
    $query = "SELECT * FROM ".TABLE_USUARIOS." WHERE email='".$email.'" AND password='".$pass."';";
    $count = doQuery($query);
    return $count->num_rows;
}

function getUser($email){
    $query = "SELECT * FROM ".TABLE_USUARIOS." WHERE email='".$email."';";
    return doQuery($query);
}

function getUserName($idaut){
    $query="SELECT nombre FROM ".TABLE_USUARIOS." WHERE idnum='".$idaut."';";
    return doQuery($query);
}

function getListUsers($idusu){
    //todos los usuarios menos yo
    $query="SELECT idnum,nombre FROM ".TABLE_USUARIOS." WHERE idnum<>".$idusu."';";
    return doQuery($query);
}
```



### Tabla de mensajes

---

Las funciones que utilizo consisten en poder enviar y escribir mensajes. Hago una función genérica (**enviarMensaje**), y a partir de ella creo las siguientes dependiendo del valor que le al “tipodestino” el mensaje será enviado a un de manera global, personal o grupal; por cada tipo de mensaje:

0- globales, 1-personales y 2-grupales (**enviarMensajeGlobal**, **enviarMensajePersonal**, **enviarMensajeGrupo**).

Además, para seleccionar y poder escribir mensajes utilizo las funciones: (**getMensajesGlobales**, **getMensajesPersonales**, **getMensajesGrupales**), también teniendo en cuenta el “tipodestino”.

```
function enviarMensajeGlobal($idaut,$titulo,$mensaje){
    return enviarMensaje($idaut,$titulo,$mensaje,0,"NULL","NULL");
}
function enviarMensajePersonal($idaut,$titulo,$mensaje,$iddestinario){
    return enviarMensaje($idaut,$titulo,$mensaje,1,"".$iddestinario,"","NULL");
}
function enviarMensajeGrupo($idaut,$titulo,$mensaje,$idgrupo){
    return enviarMensaje($idaut,$titulo,$mensaje,2,"NULL","".$idgrupo,"");
}
function enviarMensaje($idaut,$titulo,$mensaje,$tipodestino,$iddestinario,$idgrupo){
    $fecha = date('h:i m/d/Y', time());
    $query = "INSERT INTO ".TABLE_MENSAJES." ( titulo, contenido, fecha, idautor, tipodestino, iddestinario, idgrupo)";
    $query .= " VALUES ('".$titulo."','".$mensaje."','".$fecha."','".$idaut."','".$tipodestino."','".$iddestinario."','".$idgrupo."')";
    return doQuery($query);
}

function getMensajesGlobales(){
    $query = "SELECT * FROM ".TABLE_MENSAJES." WHERE tipodestino=0;";
    return doQuery($query);
}

function getMensajesPersonales($idusu){
    $query = "SELECT * FROM ".TABLE_MENSAJES." WHERE tipodestino=1 AND iddestinario='".$idusu."";
    return doQuery($query);
}

function getMensajesGrupales($idgrupo){
    $query = "SELECT * FROM ".TABLE_MENSAJES." WHERE tipodestino=2 AND idgrupo='".$idgrupo."";
    return doQuery($query);
}
```

### Tabla de grupos

---

Las siguientes funciones permiten: crear grupos (**createGroup**), obtener la lista de grupos existentes (**getListGroup**), eliminar un grupo (**deleteGroup**) y poder preguntar a qué grupos pertenece un usuario en concreto (**getUserGroup**).



```
function createGroup($nombre,$descripcion,$tipomusica,$edadmin,$edadmax){
    $query = "INSERT INTO ".TABLE_GRUPOS." (nombre, descripcion, tipomusica, edadmin, edadmax) ";
    $query = "VALUES ('".$nombre."', '".$descripcion."', '".$tipomusica."', '".$edadmin."', '".$edadmax."')";
    var_dump(doQuery($query));
}

function getListGroup(){
    $query = "SELECT * FROM ".TABLE_GRUPOS."";
    return doQuery($query);
}

function deleteGroup($idgrupo){
    $query = "DELETE FROM ".TABLE_GRUPOS." WHERE idgrupo='".$idgrupo.'";
    doQuery($query);
}

function getUserGroup($tipomusica,$edad){
    $query = "SELECT * FROM ".TABLE_GRUPOS." WHERE tipomusica='".$tipomusica.'" AND edadmin<='".$edad.'" AND edadmax>='".$edad.'";
    return doQuery($query);
}
?>
```

## Base de datos de melómanos

bdmelomanos.tbusuario	bdmelomanos.tbmensajes	bdmelomanos.tbgrupos
idnum : int(40)	idmen : int(11)	idgrupo : int(11)
nombre : varchar(100)	titulo : varchar(100)	nombre : varchar(100)
email : varchar(100)	contenido : text	descripcion : text
password : varchar(40)	fecha : text	tipomusica : varchar(100)
activo : int(11)	idautor : int(11)	edadmin : int(11)
edad : int(11)	tipodestino : int(11)	edadmax : int(11)
tipomusica : varchar(100)	iddestinatario : int(11)	
	idgrupo : int(11)	

Respecto a la interfaz utilizada es muy sencilla, el .css tiene lo básico que debe contener ya que decidí centrarme en las funcionalidades de la práctica porque esta era “libre”.



## CONCLUSIONES

La realización de la presente práctica me ha ofrecido la oportunidad de aprender a montar un **sitio web** de forma autodidacta, siempre teniendo en cuenta la ayuda proporcionada por mi profesor de la asignatura de Ampliación de Bases de Datos, D. Ramón González del Campo Rodríguez Barbero de la Universidad Complutense de Madrid (UCM).

Pese a tener algunas dificultades y ciertos problemas a la hora de realizar el código fuente, he sido capaz de superarlos y solucionarlos, gracias también a los recursos que hay en Internet sobre PHP, HTML, MySQL ...

Como futuros cambios, propondría poder **editar los mensajes** que escribe un usuario, así como tener más en cuenta la **interfaz**, ya que la he realizado de forma libre como indica el enunciado y creo que, si hubiera dispuesto de más tiempo, la hubiera podido mejorar.