

# Creando nuestro propio blog

¿En qué consiste nuestro blog? Básicamente lo que vamos a hacer es un sistema que muestre noticias, ordenadas por fecha cronológicamente descendente, a las cuales se les podrá dejar comentarios. ¡Comencemos!

## Diferentes niveles de accesos

Para poder llevar a cabo nuestro **blog** vamos a necesitar tres niveles diferentes de **usuarios**, el nivel base va a ser el **usuario anónimo**, o sea aquel usuario que nunca se registró en nuestro sitio, podrá leer todo el contenido del mismo pero no podrá dejar comentarios.

El siguiente nivel es el usuario común o **usuario registrado**, que podrá dejar comentarios en las noticias.

Y por último tenemos el **usuario administrador** que se va a encargar de publicar noticias y aprobar comentarios de los usuarios.

Normalmente en los **blogs** no hace falta ser un usuario registrado para dejar comentarios, pero va a ser útil que aprendan como se puede hacer un “sistemita” de **registro**, así que los permisos los incluimos de esta forma para fines prácticos.

## Diferentes características del contenido

El contenido del sitio va a estar separado por categorías por lo que las noticias no sólo se deberán mostrar cronológicamente sino que también se tendrán que poder filtrar por categorías.

## FrontEnd y BackEnd. ¿Qué son?

Por **FrontEnd** entendemos a lo que ve el usuario anónimo o usuario registrado, es la parte frontal o visible del sitio.

Por **BackEnd** nos referimos al **panel de control** para administrar el sitio. A donde vamos a publicar nuestras noticias, guardar borradores y aprobar comentario

## Creación de la base de datos

Básicamente vamos a tener cuatro tablas. Una de usuarios, otra de noticias, una tercera con las categorías que van a tener las noticias y finalmente una tabla con los comentarios a las noticias.

## Estructura de la base de datos de nuestro blog:

### Tabla: usuarios

Field	Type	Null	Key	Default	Extra
idUsuario	int(11)		PRI	NULL	auto_increment
usuario	varchar(30)				
password	char(50)				
email	varchar(50)				
tipo	enum('comun','admin')			comun	

Hacemos que **idUsuario** sea la primary key con un autoincrement para que se vaya sumando a medida que se van registrando nuevos usuarios, también ponemos que el campo usuario tenga un varchar de 30, ya que no sabemos qué tan largos pueden llegar a ser los nombres de usuarios elegidos por los mismos, lo que sí podemos hacer es limitárselo a 30.

Por otro lado el campo **password** va a tener siempre un ancho de **50 caracteres**, porque para mayor seguridad vamos a usar una codificación **md5** y los códigos que genera la misma tienen siempre la misma cantidad de caracteres.

El campo **email** es igual que el del usuario y el de tipo es con un **enum**, ya que solo vamos a tener dos tipos de usuarios.

Por otro lado tenemos la tabla de noticias:

### Tabla: noticias

Field	Type	Null	Key	Default	Extra
idNoticia	int(11)		PRI	NULL	auto_increment
titulo	varchar(50)				
copete	varchar(255)				
cuerpo	text				
idUsuario	int(11)				
idCategoria	int(11)				
fPublicacion	timestamp				
fCreacion	timestamp				
fModificacion	timestamp				

No hay mucho que explicar en esta, también tenemos una primary key en **idNoticia** para identificar a cada una, tenemos una relación 1 a muchos entre una noticia y un usuario, es decir, una noticia puede tener un sólo autor pero un autor puede tener muchas noticias. Lo mismo con las categorías, una noticia puede tener una sola categoría pero una categoría puede tener muchas noticias.

Y también tenemos tres campos de fechas, uno para la fecha de publicación de la noticia, otra para la fecha en que se creó y finalmente cuando se modificó.

Para la tabla de categorías vamos a tener:

### Tabla: categorías

Field	Type	Null	Key	Default	Extra
idCategoria	int(11)		PRI	NULL	auto_increment
valor	varchar(50)				

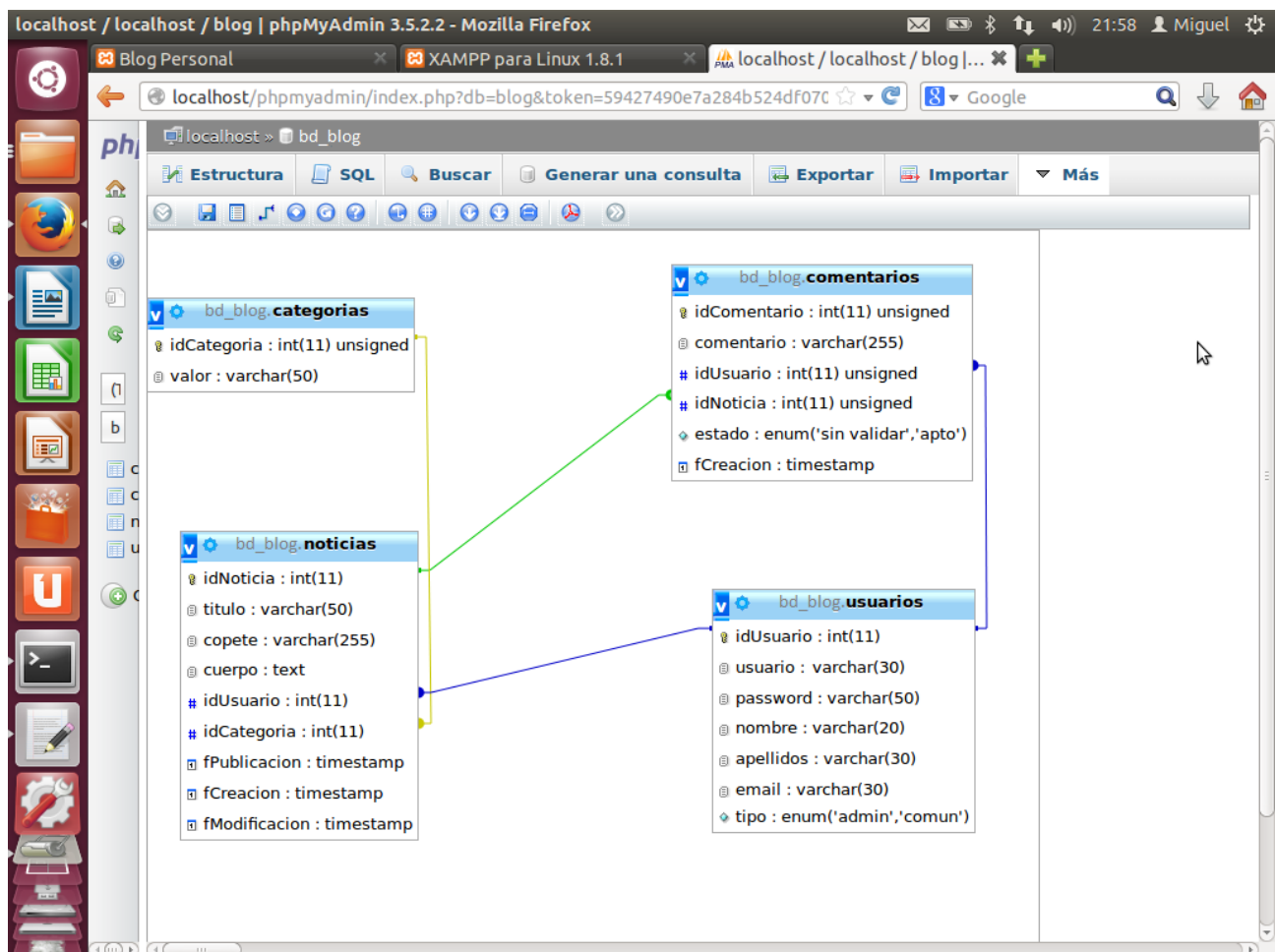
## Tabla: comentarios

Field	Type	Null	Key	Default	Extra
idComentario	int(11)		PRI	NULL	auto_increment
comentario	text				
idUsuario	int(11)				
idNoticia	int(11)				
estado	enum('sin validar','apto')			sin validar	
fCreacion	timestamp				

En esta tabla tenemos también una relación 1 a muchos con los usuarios, ya que un comentario puede tener un único usuario pero un usuario puede tener muchos comentarios, lo mismo para las noticias.

Por otro lado tenemos un estado, ya que apenas un usuario escribe un comentario el mismo está sin validar por un administrador, en este punto podemos optar por mostrar o no los comentarios no validados (igual todavía no estamos en esa disyuntiva) y los mismos aparecerán en un listado general para que un **admin** los apruebe o los borre.

Y por último tenemos la fecha de creación del mismo.



# Página del sitio (index.php)

Por un lado vamos a tener algunos archivos en la carpeta raíz, pero por otro vamos a tener una carpeta llamada "**admin**" que es donde va a estar nuestro **backend**. En esta carpeta también tendremos dos archivos adicionales que son necesarios antes de arrancar.

## Cómo vamos a organizar nuestros archivos

La estructura de archivos por le momento será:

```
+ admin } - config.php
         } - conexion.php
-----
```

```
- index.php
- registro_usuario.php
- login_usuario.php
```

El primer archivo que vamos a ver, **config.php**, es un archivo de configuración, en el cual guardaremos constantes que vamos a necesitar a lo largo del sitio. El misma va a contener:

admin/config.php

PLAIN TEXT

```
1.      <?php
2.
3.      // constantes de conexion de db
4. define( 'DB_SERVER', 'localhost' );
5. define( 'DB_NAME', 'bd_blog' );
6. define( 'DB_USER', 'jose' );
7. define( 'DB_PASS', 'josefa' );

8.      ?>
```

El mismo deberá estar completo con los datos de su base de datos.

El siguiente archivo, **conexion.php**, contiene una serie de sentencias para la conexión con la base de datos, ya que en todas o casi todas las páginas de nuestro sitio necesitaremos una conexión con la db, es práctico mantenerla dentro de un archivo.

admin/conexion.php

PLAIN TEXT

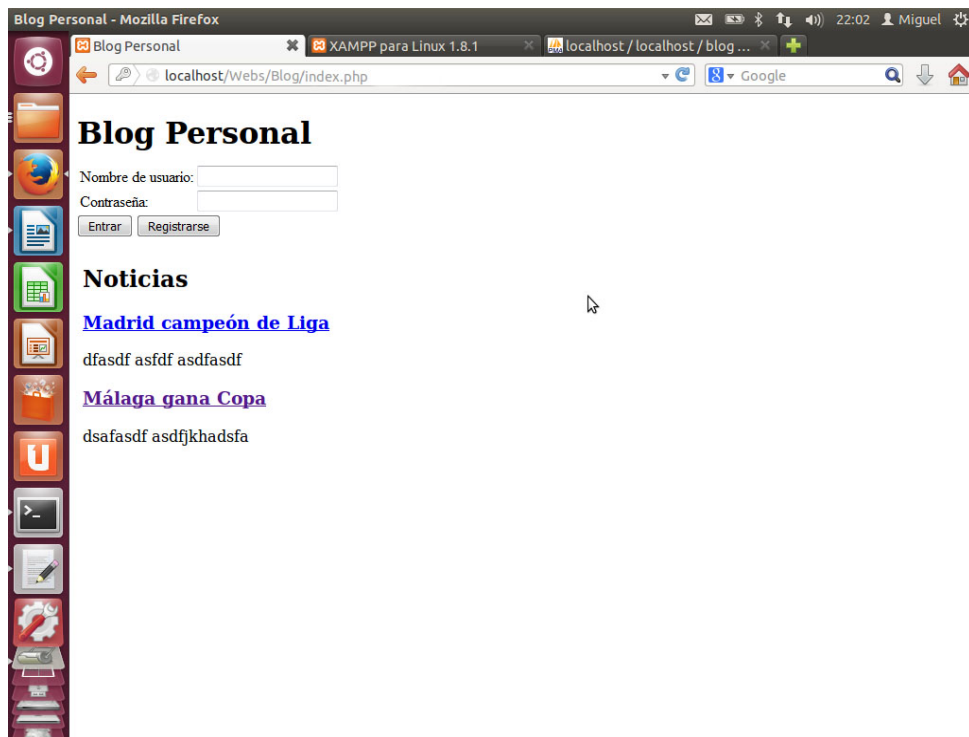
```
1.      <?php
2.
3. $conexion = mysql_connect(DB_SERVER, DB_USER, DB_PASS);
4. if(!$conexion)
5.         die("Imposible conectar. Error
    numero".mysql_errno()." : ".mysql_error());
6.
7. $bd = mysql_select_db(DB_NAME);
8. if(!$bd)
9.         die("No se pudo conectar con bd ".mysql_error());
10.
11.      ?>
```

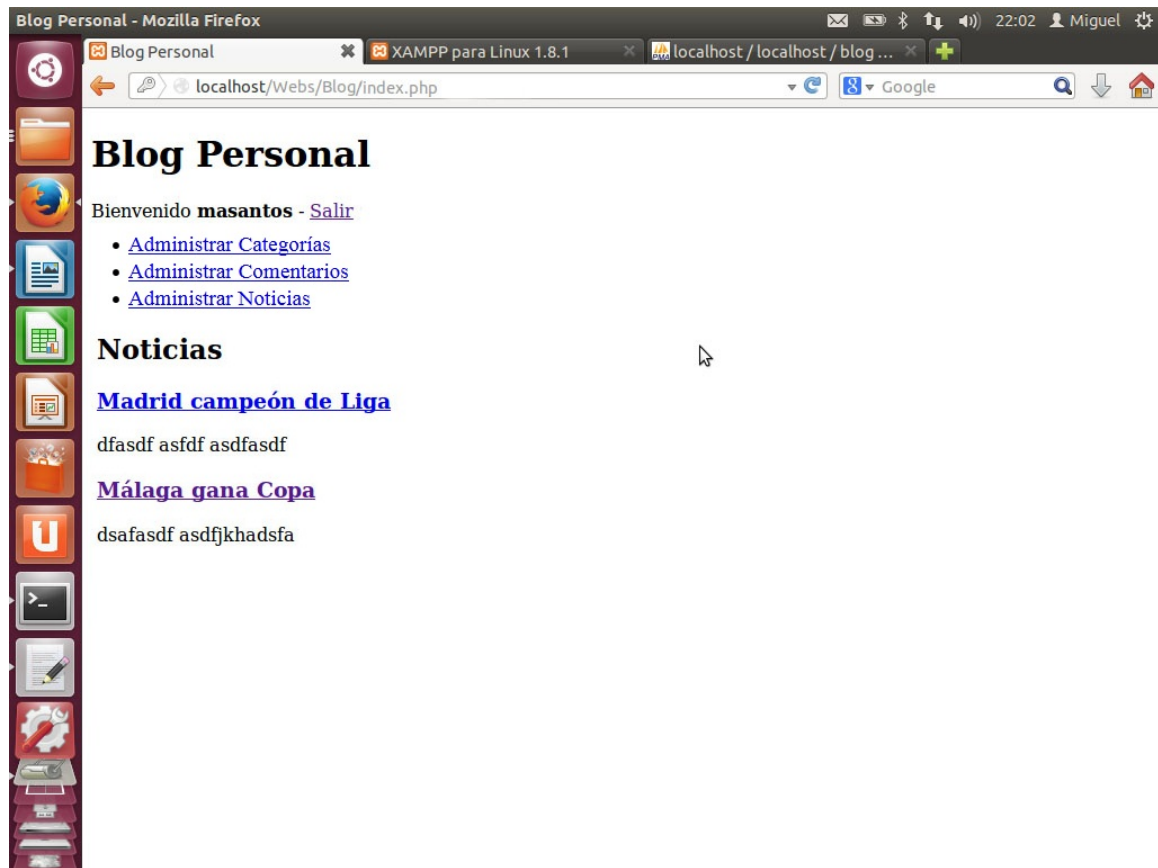
Fíjense que estamos usando las constantes que agregamos en el archivo **config.php**, esto es práctico a la hora de tener que cambiar de base de datos, o migrar el sitio, sólo tendremos que modificar el archivo de configuración.

Listo, ya tenemos nuestros dos archivos antes de comenzar el proceso de registro.

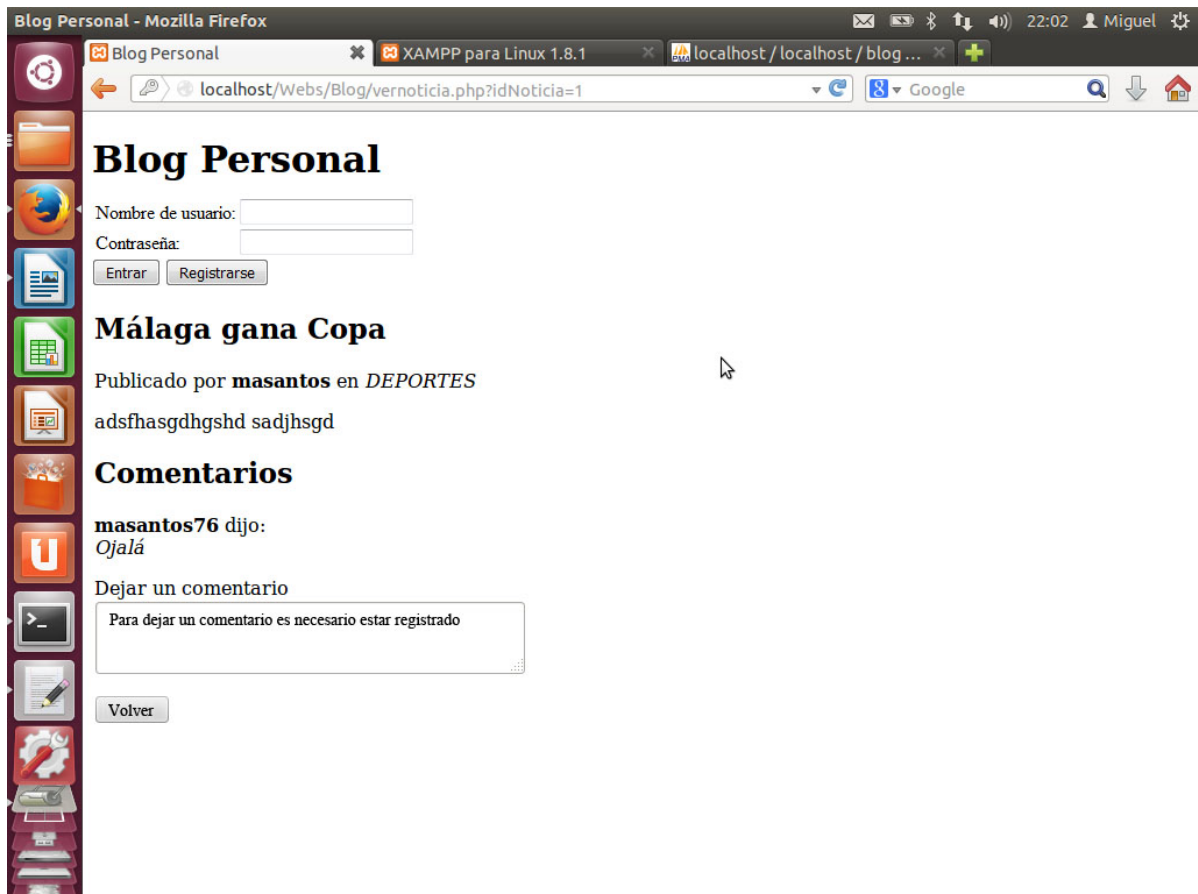
En el archivo **index.php** tendremos una presentación, donde se podrá optar por registrarse o ingresar y se podrán leer las noticias.

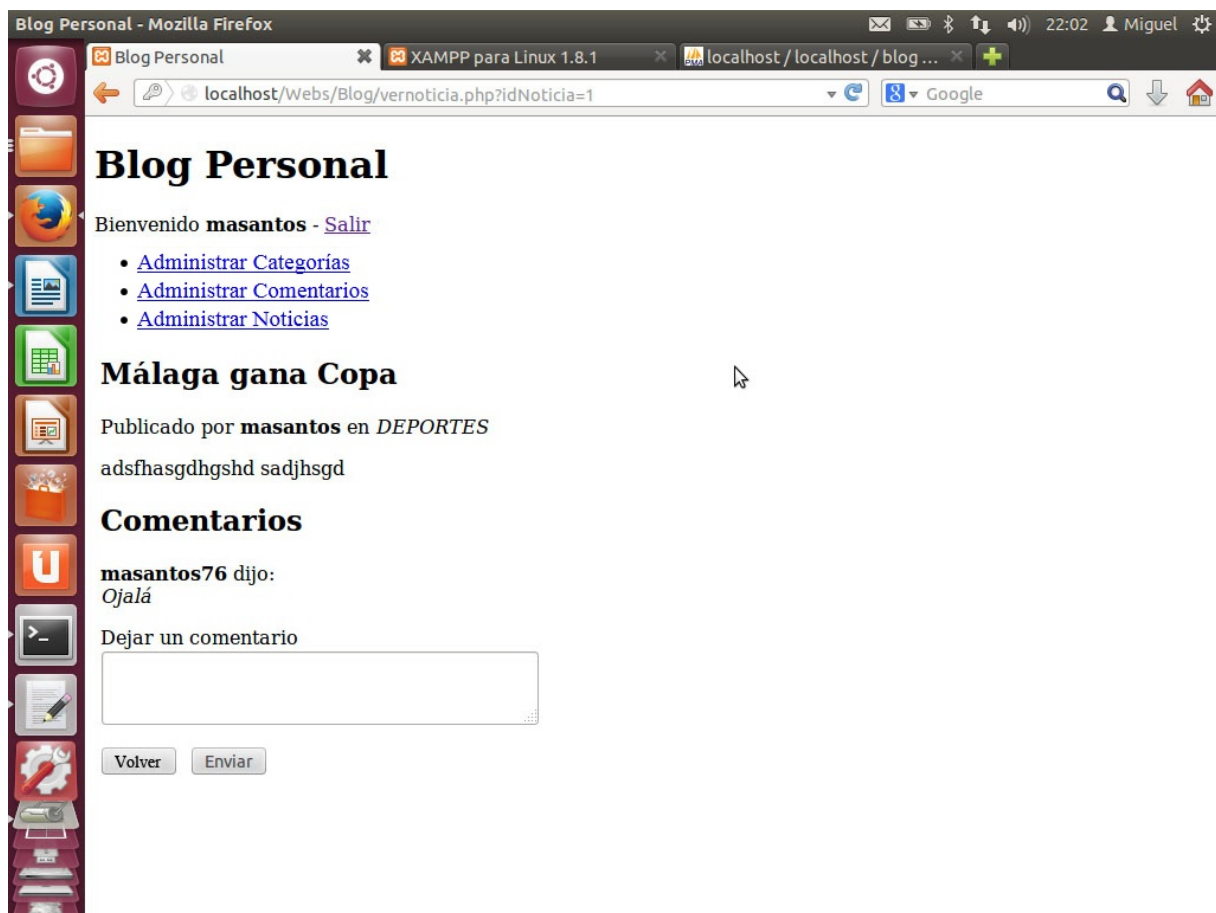
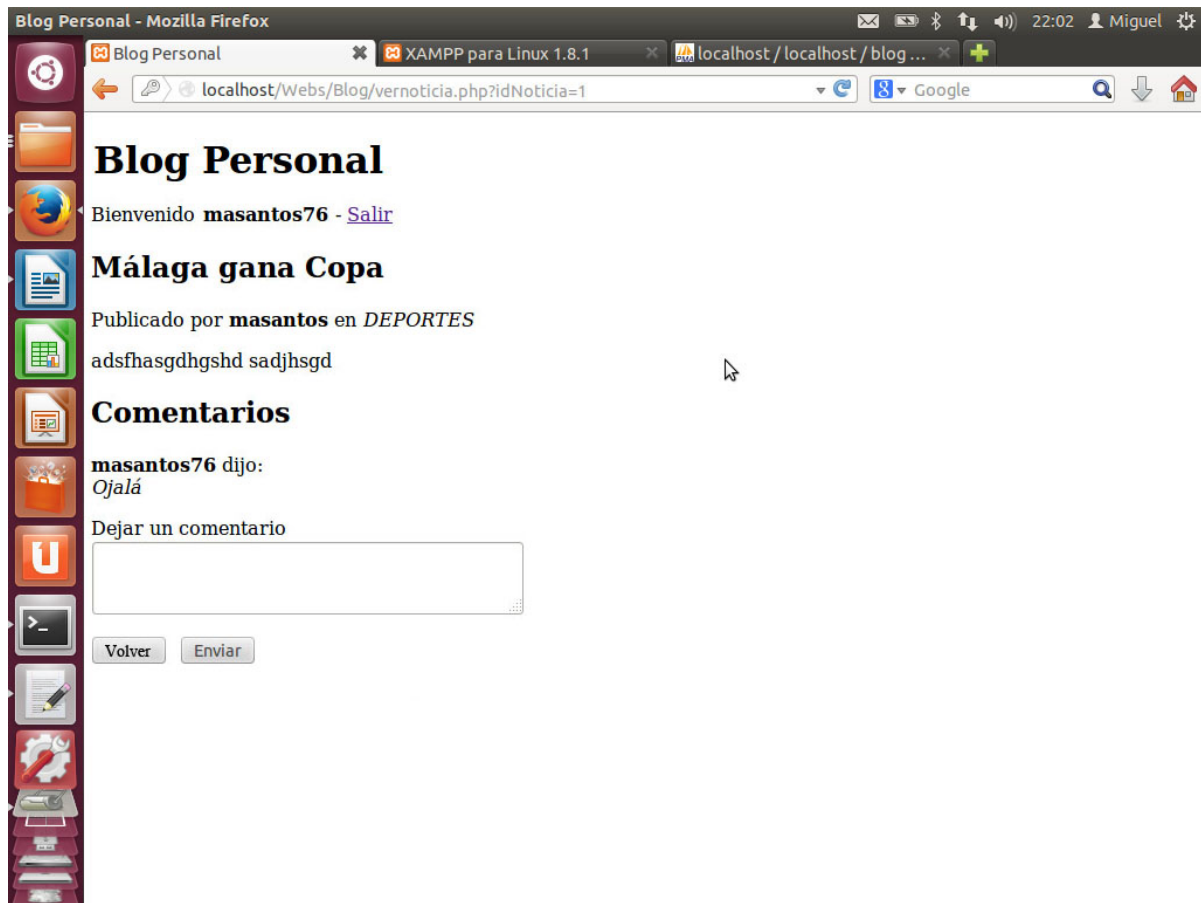
Dependiendo si se ha logueado o no, o si el logueado es común o administrador, la apariencia de **index.php** podrá ser:



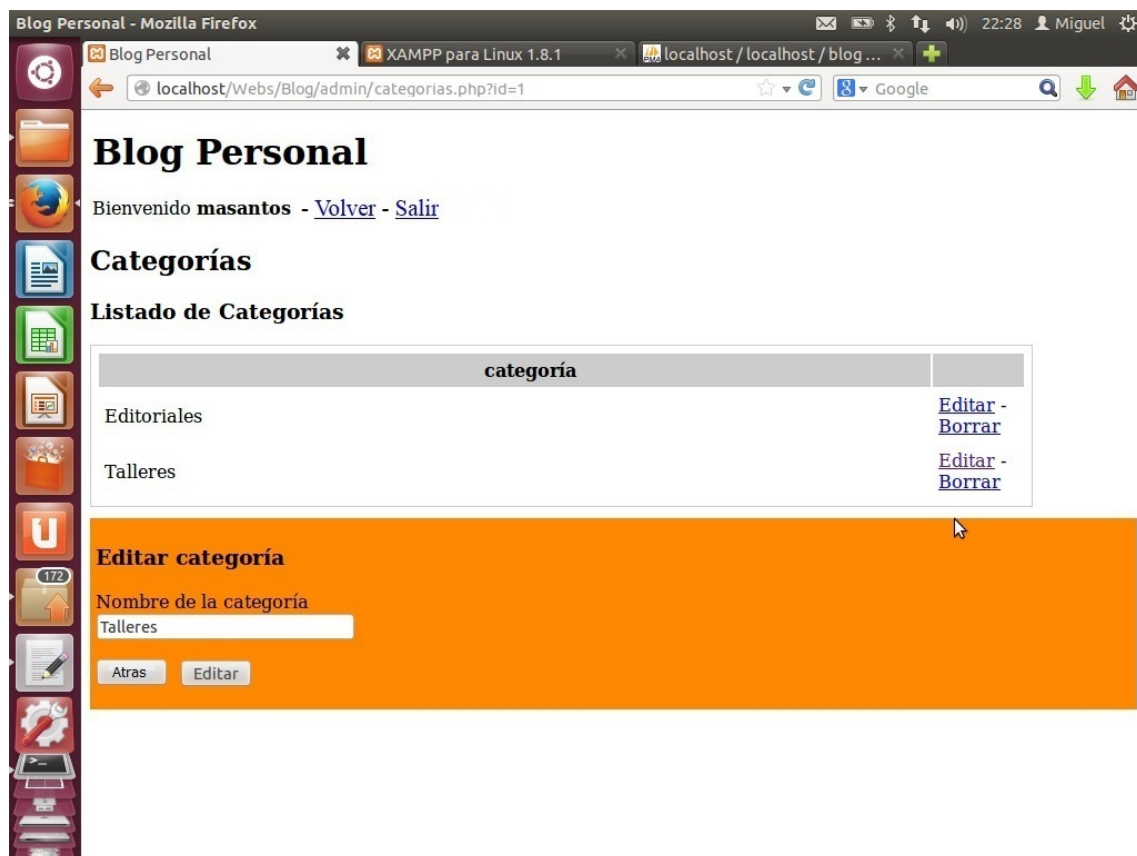
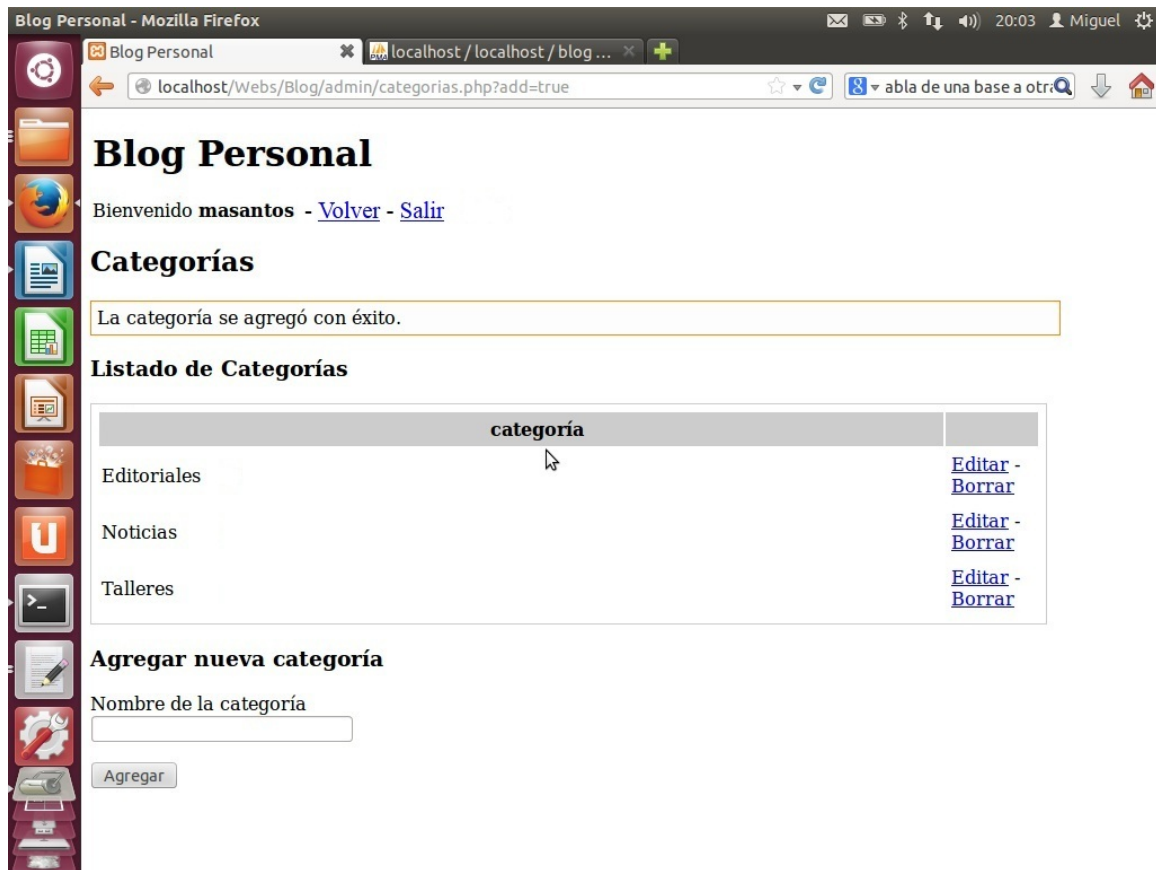


Si un usuario decide leer una noticia ( pulsa en algún enlace de alguna noticia ) iremos a **vernoticia.php**:





Ahora, toca la parte del administrador. Comenzaremos administrando las categorías. Para ello crearemos un archivo llamado **categorias.php** dentro de la carpeta admin con la siguiente apariencia:





Para la gestión de las noticias crearemos **noticias.php** dentro de la carpeta admin con funcionalidad equivalente a **categorias.php** y con la siguiente apariencia:

Observad, que cuándo se crea una noticia sólo damos opción en el formulario de cambiar la Fecha de publicación, ya que para la fecha de modificación pondremos en la base de datos en el campo fPublicacion de la tabla *noticias*, la propiedad *"ON UPDATE CURRENT\_TIMESTAMP"* y valor por defecto *"CURRENT\_TIMESTAMP"*, y para la fecha de creación cogeremos la fecha del sistema:

```
<?php
    $f_creacion =date("Y-m-d H:i:s");
<?
```

Y para la gestión de las comentarios crearemos **comentarios.php** dentro de la carpeta admin con la siguiente apariencia:

Panel de Control - Mozilla Firefox

Panel de Control

http://localhost/Webs/Blog/admin/comentarios.php

## Blog Personal

Bienvenido **masantos** - [Volver](#) - [Salir](#)

### Listado de Comentarios

Tipos de comentarios: Sin aprobar

### Comentarios sin aprobar

Comentarios	Opción
Otro comentario Dijo <b>masantos76</b> en <a href="#">Ferrer sucumbe ante el rocoso Nishikori</a>	<a href="#">Aprobar</a> - <a href="#">Editar</a> - <a href="#">Borrar</a>

Señalar, que además de mostrarse los comentarios, se enlaza también con la noticia (*vernoticia.php*)

Panel de Control - Mozilla Firefox

Panel de Control

http://localhost/Webs/Blog/admin/comentarios.php

## Blog Personal

Bienvenido **masantos** - [Volver](#) - [Salir](#)

### Listado de Comentarios

Tipos de comentarios: Todos

### Todos los Comentarios

Comentarios	Opción
Comentario en condiciones Dijo <b>masantos76</b> en <a href="#">Ferrer sucumbe ante el rocoso Nishikori</a>	<a href="#">Editar</a> - <a href="#">Borrar</a>
Otro comentario Dijo <b>masantos76</b> en <a href="#">Ferrer sucumbe ante el rocoso Nishikori</a>	<a href="#">Aprobar</a> - <a href="#">Editar</a> - <a href="#">Borrar</a>
Comentario bueno Dijo <b>masantos</b> en <a href="#">Ferrer sucumbe ante el rocoso Nishikori</a>	<a href="#">Editar</a> - <a href="#">Borrar</a>

Y para terminar, modificaremos en la base de datos, el campo “*tipo*” de la tabla **usuarios**, para que podamos almacenar tres tipos de usuarios: *comun*, *admin* y *superadmin*. De esta forma si un usuario de tipo “superadmin” entrase en el sistema, además de poder administrar las noticias, categorías y los comentarios, le aparecería además la opción de *Administrar Usuarios*.