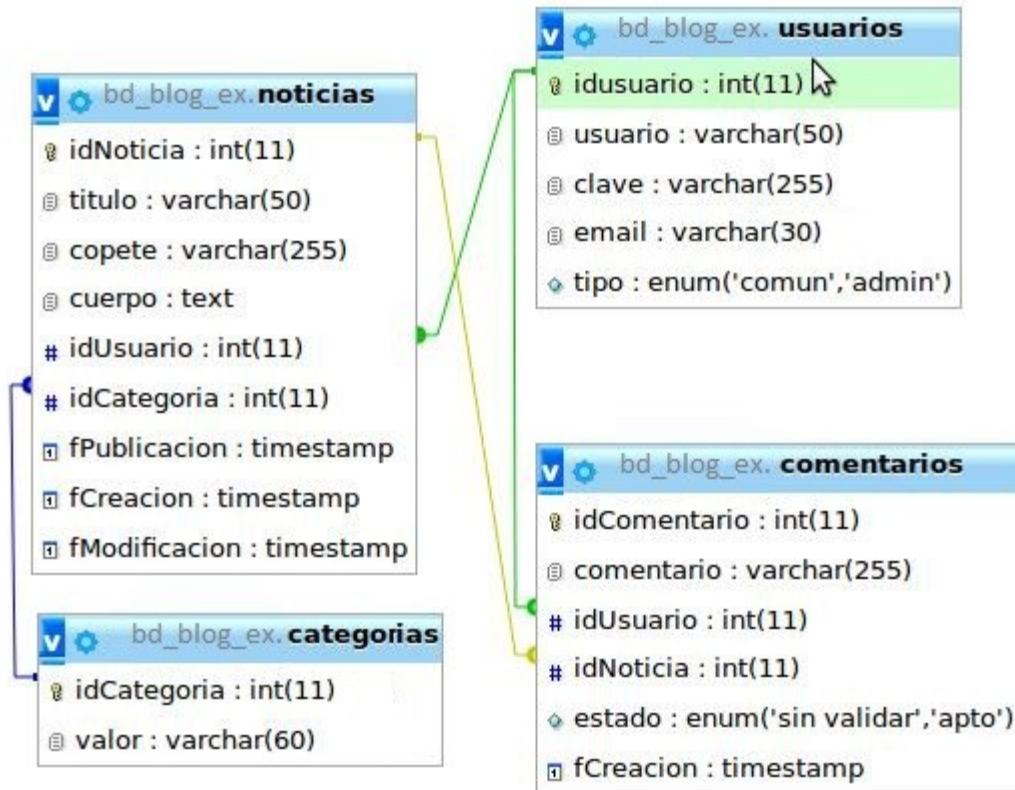


EXAMEN 2º DAW. Examen recuperación Servicios Web

Realizar un sitio web para llevar el control de un blog , teniendo en cuenta los siguientes aspectos:

- 1) La base de datos, así como la tablas y los atributos empleados deben ser EXACTAMENTE :



- 2) Rellene manualmente las tablas inventándose:

- Tres usuarios (dos “admin” y otro “comun”).
- Una noticia.
- Y tres comentarios de los usuarios “admin” creados, sobre la noticia creada (dos “sin validar” y uno “apto”).

- 3) En la página inicial del lugar (index.php), el usuario va a poder loguearse o registrarse. Tanto si un usuario común se loguea como si un usuario nuevo se registra correctamente, va a saltar a una página inicial (principal.php) de acceso restringido sin contenido (Sólo bienvenida y la opción de salir).

Si el usuario que se loguea es administrador, entonces va saltar a una página (admin/gest_comentarios.php), dónde el administrador va a poder aprobar y borrar comentarios.

Tras un periodo de inactividad superior a los 3 minutos, un usuario (normal o administrador) deberá volver a loguearse.

Panel de Control - Mozilla Firefox

Panel de Control

http://localhost/Blog/admin/gest_comentarios.php

GESTIÓN DE COMENTARIOS

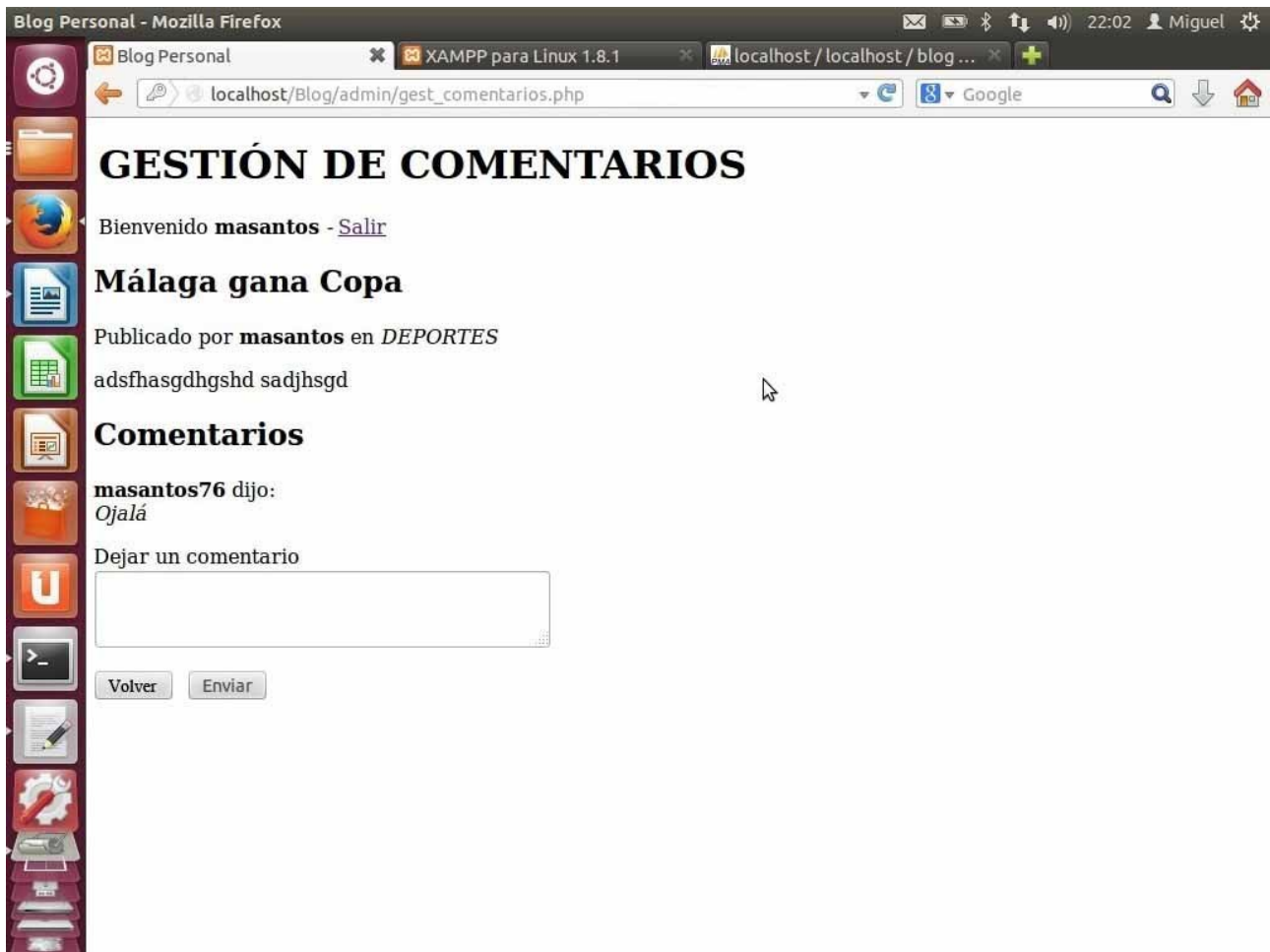
Bienvenido masantos - [Salir](#)

Todos los Comentarios

ID	Comentarios	Opción
2	Comentario en condiciones Dijo masantos en Ferrer sucumbe ante el rocoso Nishikori	Aprobar - Borrar
3	Otro comentario Dijo masantos en Ferrer sucumbe ante el rocoso Nishikori	Borrar
6	Comentario bueno Dijo masantos en Ferrer sucumbe ante el rocoso Nishikori	Aprobar - Borrar

EXAMEN 2º DAW. Examen recuperación Servicios Web

- 4) Si el administrador pincha en el enlace (Titular de la noticia) podrá leer la noticia y realizar un nuevo comentario:



- 5) Nuestra aplicación sólo podrá acceder a la BD local (bd_blog_ex) mediante los siguientes Servicios Webs:

- 1) Para el logueo, vamos a acceder a la BD mediante un servicio web de tipo POST y la URL de la petición será:

http://localhost/Blog/servicios_rest/login

Para su correcto funcionamiento se le han de pasar a este servicio dos parámetros con índices asociativos “usuario” y “clave”. En “usuario” irá el nombre del usuario y en “clave” la contraseña encriptada en MD5. Este servicio responderá con tres tipos distintos de JSON:

- a) {“mensaje_error” : “Error.....”}, si hay problemas en la conexión a la BD o al realizar la consulta.
- b) {“mensaje”: “Usuario no registrado en la BD”}, si el usuario logueado no se encuentra registrado.
- c) {“usuario”: [...] }, el array contendrá todos los datos del usuario logueado correctamente

EXAMEN 2º DAW. Examen recuperación Servicios Web

- 2) Mediante una petición POST, vamos a dar de alta un nuevo usuario. La URL de la petición será:

http://localhost/Blog/servicios_rest/insertarUsuario

Para su correcto funcionamiento se le han de pasar a este servicio tres parámetros con índices asociativos “usuario”, “clave” y “email”. En “usuario” irá el nombre de un usuario no repetido, en “clave” la contraseña encriptada en MD5 y en “email” un email no repetido y escrito sintácticamente correcto. Este servicio responderá con dos tipos distintos de JSON:

a) {“mensaje_error” : “Error....”}, si hay problemas en la conexión a la BD o al realizar la consulta.

b) {“mensaje”: “Usuario registrado correctamente en la BD”}, si el usuario se ha registrado correctamente.

- 3) Mediante una petición GET, obtener los datos de todos los comentarios (idComentario, comentario, idUsuario, idNoticia, estado, fechaCreacion). URL de la petición:

http://localhost/Blog/servicios_rest/comentarios

En caso de error en la conexión o en la consulta a la BD, el JSON devuelto será:

{ “mensaje_error” : “Error....” }

y en caso de éxito el JSON será:

{ “comentarios” : [[...],[...],.....] }

- 4) Mediante una petición GET, obtener los datos de todos los usuarios que tenga una de sus campos a un valor determinado. URL de la petición:

http://localhost/Blog/servicios_rest/usuarios/columna/valor

,dónde *columna* y *valor* van a ser dos parámetros. En caso de error en la conexión o en la consulta a la BD, el JSON devuelto será:

{ “mensaje_error” : “Error....” }

, si no existe en la tabla usuarios ningún registro donde esa columna tenga ese valor el JSON devuelto será:

{ “mensaje” => “No existe un usuario con la columna = valor” }

en otro caso devolverá un JSON con todos los datos de los usuarios que tenga esa columna a ese valor:

{ “usuarios” : [[...],[...],.....] }

- 5) Mediante una petición GET, obtener los datos de todos los comentarios de una noticia (idComentario, comentario, idUsuario, idNoticia, estado, fechaCreacion). URL de la petición:

http://localhost/Blog/servicios_rest/comentarios/id

,dónde *id* va a ser un parámetro. En caso de error en la conexión o en la consulta a la BD, el JSON devuelto será:

EXAMEN 2º DAW. Examen recuperación Servicios Web

```
{ "mensaje_error" : "Error...." }
```

y en caso de éxito el JSON será:

```
{ "comentarios" : [[...],[...],.....] }
```

- 6) Mediante una petición GET, obtener los datos de un usuario (idusuario, usuario, clave, email, tipo). URL de la petición:

http://localhost/Blog/servicios_rest/usuario/id

,dónde *id* va a ser un parámetro. En caso de error en la conexión o en la consulta a la BD, el JSON devuelto será:

```
{ "mensaje_error" : "Error...." }
```

y en caso de éxito el JSON será:

```
{ "usuario : [.....] }
```

- 7) Mediante una petición GET, obtener los datos de una noticia (idNoticia, título, copete,.....). URL de la petición:

http://localhost/Blog/servicios_rest/noticia/id

,dónde *id* va a ser un parámetro. En caso de error en la conexión o en la consulta a la BD, el JSON devuelto será:

```
{ "mensaje_error" : "Error...." }
```

y en caso de éxito el JSON será:

```
{ "noticia : [.....] }
```

- 8) Mediante una petición GET, obtener los datos de una categoría (idCategoria, valor). URL de la petición:

http://localhost/Blog/servicios_rest/categoria/id

,dónde *id* va a ser un parámetro. En caso de error en la conexión o en la consulta a la BD, el JSON devuelto será:

```
{ "mensaje_error" : "Error...." }
```

y en caso de éxito el JSON será:

```
{ "categoria : [.....] }
```

- 9) Mediante una petición GET, obtener los datos de un comentario (idComentario, comentario, idUsuario, idNoticia, estado, fechaCreacion). URL de la petición:

http://localhost/Blog/servicios_rest/comentario/id

,dónde *id* va a ser un parámetro. En caso de error en la conexión o en la consulta a la BD, el JSON devuelto será:

```
{ "mensaje_error" : "Error...." }
```

EXAMEN 2º DAW. Examen recuperación Servicios Web

y en caso de éxito el JSON será:

```
{ "categoria : [.....]}
```

10) Mediante una petición PUT, actualizar un comentario. URL de la petición:

```
http://localhost/Blog/servicios_rest/actualizarComentario/id
```

dónde *id* va a ser un parámetro. Y además habrá que pasarle el estado a actualizar en un array con índice asociativo tal y como éste:

```
array("estado"=>.....)
```

En caso de error en la conexión o en la consulta a la BD, el JSON devuelto será:

```
{ "mensaje_error" : "Error...."}
```

y en caso de éxito:

```
{ "mensaje": "El comentario con idComentario =X ha sido actualizado con éxito"}
```

11) Mediante una petición DELETE borrar un comentario. URL de la petición:

```
http://localhost/Blog/servicios_rest/borrarComentario/id
```

dónde *id* va a ser un parámetro. En caso de error en la conexión o en la consulta a la BD, el JSON devuelto será:

```
{ "mensaje_error" : "Error...."}
```

y en caso de éxito:

```
{ "mensaje": "El comentario con idComentario =X ha sido borrado con éxito"}
```

RECUERDA:

- La función predefinida ***filter_var(\$correo, FILTER_VALIDATE_EMAIL)*** que devuelve ***true*** si en ***\$correo*** hay una dirección de correo sintácticamente correcta y ***false*** en otro caso.