

Al iz vel otra vez...

Para esta evaluación nuevamente tomamos la red social Facebook como ejemplo.

Antes de comenzar:

- Copie la carpeta llamada facebook en la dirección c:\wamp\www
- Acceda desde un navegador web a la dirección <http://localhost/facebook>
- De no mostrar nada notificar al catedrático.
- Ingrese a <http://localhost/phpmyadmin> y cree una base de datos vacía llamada bd_facebook.
- Seleccione la base de datos creada y haga click en Import, luego seleccione el archivo bd_facebook.sql y luego Go.
- De mostrar algún error al importar la base de datos informar al catedrático.
- Puede visualizar el modelo de la base de datos utilizando la imagen adjunta.

A continuación se listan los requisitos:

La página contiene 3 secciones principales,

“¿Quién soy?”: Muestra los diferentes usuarios que puede ser, cada vez que de click sobre el icono del usuario se deberá mostrar la información en los campos ID y Usuario actual. La lista de tarjetas deberá llenarse obteniendo información de la base de datos utilizando la siguiente consulta:

Consulta que muestra todos los usuarios registrados.

```
SELECT codigo_usuario, nombre_usuario, correo, contrasena, url_imagen_perfil  
FROM tbl_usuarios
```

Además cada tarjeta se creara a partir del siguiente bloque html:

```
<div class="col-xs-4 col-sm-2 col-md-2 col-lg-2 card-container">  
  <div class="card-profile">  
    <button type="button" class="btn btn-primary btn-xs" style="position:absolute;"  
title="#NOMBRE#" onclick="seleccionarUsuario(#CODIGO#, '#NOMBRE#');">  
      <span class="glyphicon glyphicon-user" aria-hidden="true"></span>  
    </button>  
      
    <span class="profile-name">#NOMBRE#</span>  
  </div>  
</div>
```

Nota: Cada vez que se seleccione un usuario diferente se deben actualizar la sección de las publicaciones.

“Nueva publicación”: Muestra un pequeño formulario compuesto por un botón y un componente llamado Text Area el cual es definido con la etiqueta <textarea>. Este es similar a un textfield pero permite gestionar más contenido. Es necesario agregar la lógica al botón Publicar de tal forma que se envíe la información de la publicación y el usuario seleccionado y se almacene dicha información en la tabla tbl_publicaciones.

Una vez guardada la publicación se debe actualizar el listado de las publicaciones.

Para la fecha de la publicación utilizar directamente en el insert una función llamada curdate() que retornara la fecha actual.

Algo asi como INSERT INTO tabla (campos) values (valor, curdate(), ...)

“Publicaciones”: Esta sección mostrara todas las publicaciones del usuario seleccionado, la consulta para este resultado es:

```
SELECT codigo_publicacion, titulo_publicacion, texto_publicacion, fecha_publicacion
FROM tbl_publicaciones
WHERE codigo_usuario = #USUARIO_SELECCIONADO#
```

A partir de esta consulta se generarán la estructura html de todas las publicaciones del usuario seleccionado utilizando el siguiente bloque HTML:

```
<div class="col-xs-12 col-sm-12 col-md-12 col-lg-12 card-container">
  <div class="card-profile">
    <span class="profile-name">#TITULO PUBLICACION#</span>
    <small>#FECHA PUBLICACION#</small>
    <p>#TEXTO PUBLICACION#</small>
    <h5>Comentarios</h5>
    <### LISTADO DE TODOS LOS COMENTARIOS (Otro SELECT) ###>
  </div>
</div><br>
```

Dentro del bloque anterior se puede visualizar el texto **###LISTADO DE TODOS LOS COMENTARIOS###** cuyo objetivo es sustituirla por el listado de comentarios relacionados a cada publicación respectiva. Esta información debe ser generada mediante otra consulta la cual se define a continuación:

```
SELECT a.codigo_usuario, a.comentario, b.nombre_usuario, b.url_imagen_perfil
FROM tbl_comentarios a
INNER JOIN tbl_usuarios b
ON a.codigo_usuario = b.codigo_usuario
WHERE codigo_publicacion = #CODIGO PUBLICACION RESPECTIVA#
```

Cabe recalcar que esta consulta se ejecutara por cada publicación, es decir que esta consulta estará dentro del ciclo utilizado para recorrer el listado de publicaciones por lo tanto al final deben quedar dos ciclos anidados.

El bloque html para poder generar cada comentario se muestra a continuación:

```
<div class="card-comment">
  
  <span class="profile-name">#NOMBRE USUARIO#</span>
  <p class="comment">#TEXTO COMENTARIO#</p>
</div>
```

Otros requerimientos:

- Dar funcionalidad al botón Iniciar Sesión para que haga una petición AJAX y que envíe el correo y el password y que verifique si existe algún registro que coincida con dichos datos, en caso de si haber registros (utilizar la función cantidadRegistros(\$resultado) de la clase conexión) que muestre un mensaje que, si existe, mostrar un mensaje de error en caso de no haber registros. Dicho mensaje mostrarlo en algún div.
- Dar funcionalidad al formulario de registro de usuarios que se muestra en la ventana modal al dar click en el botón que aparece a la par del botón de iniciar sesión. El formulario se encuentra casi al final de la página html.

Nota:

- Al finalizar el examen asegúrese de guardar todos los archivos, comprima el proyecto y súbalo a la plataforma web de la clase.
- Evite hacer preguntas como: ¿Cómo se hacía esto?, ¿Porque no funciona?, ¿Dónde está el error?