

Al iz vel...

¿Cómo funcionan los sitios famosos como Facebook, Twitter, Instagram, entre otros? Son plataformas enormes y complejas, gestionadas por cientos de personas, sin embargo para ser lo que son ahora seres humanos las crearon comenzando con algo pequeño utilizando conceptos básicos.

Ahora es su turno, su misión el día de hoy además de la evaluación es aprender un poco más. Se le encomienda la misión de simular una lista de amigos “similar” a la red social Facebook siguiendo los criterios de este enunciado, probablemente después del examen no quiera entrar más a Facebook, o probablemente tendrá más curiosidad sobre el tema.

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 solo ciertos usuarios, específicamente el 1,2,13,14,15 y 18

```
SELECT codigo_usuario, nombre_usuario, correo, contrasena, url_imagen_perfil
FROM tbl_usuarios
WHERE codigo_usuario IN (1,2,13,14,15,18);
```

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 las otras secciones.

“Mis Aleros”: Muestra los diferentes usuarios que han sido marcados como amigos, esta lista de usuarios deberá ser consultada de la base de datos a partir de la siguiente consulta:

Consulta que muestra el listado de los amigos de un usuario específico, el usuario se debe especificar en la sección #USUARIO_SELECCIONADO#, haga una prueba ejecutando el script en phpmyadmin sustituyendo el valor por 1 y vera los amigos de Patricio.

```
SELECT a.codigo_usuario, a.nombre_usuario, a.correo, a.contrasena, a.url_imagen_perfil
FROM tbl_usuarios a
INNER JOIN tbl_amigos b
ON (a.codigo_usuario = b.codigo_usuario_amigo)
WHERE b.codigo_usuario = #USUARIO_SELECCIONADO#;
```

Además cada tarjeta de esta sección 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">
    
    <span class="profile-name">#NOMBRE#</span>
  </div>
</div>
```

“Seleccionar nuevos amigos”: Esta sección mostrara todos los usuarios que no son amigos de la persona seleccionada, la lista de estos usuarios se obtendrá utilizando la siguiente consulta:

Consulta que muestra todos los usuarios que no son amigos del usuario seleccionado y excluye también al usuario seleccionado.

```
SELECT a.codigo_usuario, a.nombre_usuario, a.correo, a.contrasena, a.url_imagen_perfil
FROM tbl_usuarios a
WHERE a.codigo_usuario NOT IN
(
  SELECT codigo_usuario_amigo
  FROM tbl_amigos
  WHERE codigo_usuario = #USUARIO_SELECCIONADO#
)
AND a.codigo_usuario != #USUARIO_SELECCIONADO#;
```

Cada vez que se de click sobre el botón con el signo más(+) de cada tarjeta se deberá hacer una petición AJAX que envíe el código del usuario seleccionado actualmente y el código del usuario que se agregara como amigo esta información se deberá guardar en la tabla tbl_amigos la cual definirá las amistades mediante los códigos relacionados, la tabla solo tiene dos campos, el código del usuario y el código del usuario amigo.

Cada tarjeta de esta sección deberá ser creada 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="Agregar"
onclick="agregarAmigo(#CODIGO#);">
      <span class="glyphicon glyphicon-plus" aria-hidden="true"></span>
    </button>
    
    <span class="profile-name">#NOMBRE#</span>
  </div>
</div>
```

Nota: Cada vez que se agregue un usuario nuevo se deben actualizar todas las secciones, utilizar la imagen loading.gif.

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?