

Técnicas de Programación 3 Trabajo Práctico Nro 2

Este trabajo consta de una aplicación web, la cual deberá programarse con PHP para cumplir con los siguientes procesos:

- Se debe poder activar una cuenta para poder ingresar a un panel
- Se debe poder ingresar al panel estando teniendo la cuenta activa
- Una vez ingresado al panel, se debe poder ver un log de eventos que se van registrando en ciertos momentos, y según el nivel del usuario logueado.
- Manejar sesiones de PHP para ingresar y salir del panel.



La **Base de datos tendrá esta estructura**. Deberás generarla y luego conectarla desde tus scripts que contengan las funciones correspondientes.

Tabla Usuarios:

- un identificador único, que sea Clave Primaria de la tabla, siendo autoincremental.
- Email [hasta 100 caracteres]. No puede ser nulo.
- Clave [hasta 25 caracteres]. No puede ser nulo.
- FechaCreacion [fecha y hora]. No puede ser nulo.
- Activo [usuario inactivo o usuario activo, 0 o 1 respectivamente]
- FechaActivacion [fecha y hora]. Puede ser nulo
- Nivel: 1 es usuario admin, 2 usuario comun. Tipo entero.

Deberá contener estos valores, los cuales debes ingresarlos mediante consultas o desde el PHPMYAdmin.

ID	EMAIL	CLAVE	FECHACREACION	ESTADO	FECHAActivACION	NIVEL
1	sue@gmail.com	123459	2017-07-08 00:00:00	0	NULL	1
2	pablo@gmail.com	789456	2017-07-08 00:00:00	0	Null	2
3	maria@gmail.com	123789	2017-07-08 00:00:00	0	Null	1

Esto indica que nuestra tabla contiene usuarios registrados pero inactivos.

Tabla Logs:

- un identificador único, que sea Clave Primaria de la tabla, siendo autoincremental.
- FechaLog [fecha y hora que ocurre el evento]. No puede ser nulo
- ID_Evento: un identificador del evento que está ocurriendo en ese momento [tipo entero]

- Email [hasta 100 caracteres]. No puede ser nulo.

Esta tabla de **Logs** se ira cargando a medida que surjan los eventos correspondientes.

Tabla **Eventos**:

- un identificador único, que sea Clave Primaria de la tabla, siendo autoincremental.
- Descripción [hasta 150 caracteres]. Puede ser nulo. Almacena el detalle del evento que sucede.
- Tipo: [entero pequeño] [1: warning , 2: información]

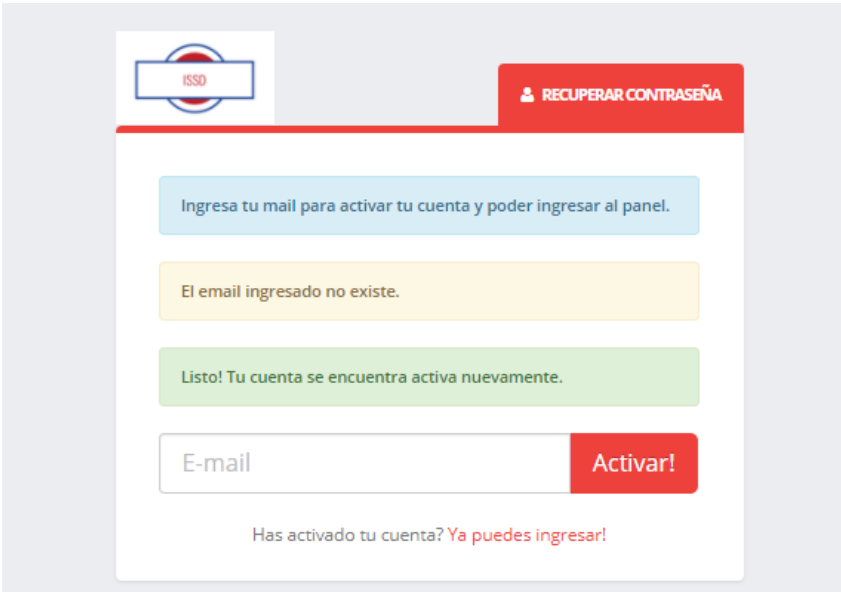
Deberá contener estos valores los cuales debes ingresarlos mediante consultas o desde el PHPMyAdmin.

ID	DETALLE	TIPO
1	Activación de cuenta	2
2	Email inexistente para activar	1
3	Acceso correcto	2
4	Datos incorrectos en login	1

Una vez estructurada la Base, vamos a los procesos de cada pantalla.

Activar Cuenta

Desde la **pantalla de activar_cuenta** se debe:



ISSD

RECUPERAR CONTRASEÑA

Ingresa tu mail para activar tu cuenta y poder ingresar al panel.

El email ingresado no existe.

Listo! Tu cuenta se encuentra activa nuevamente.

E-mail

Activar!

Has activado tu cuenta? [Ya puedes ingresar!](#)

 <p>Instituto Superior Santo Domingo</p>	<p>Trabajo Práctico Nro 2</p>	<p>TÉCNICAS DE PROGRAMACIÓN 3</p>
---	--------------------------------------	--

- permitir activar la clave de un mail ya registrado: es decir, al ingresar un mail, se debe validar que exista en nuestra base. Para esto se debe cumplir con estos procesos:
 - Se debe **validar** que el campo del mail llegue al server con datos, es decir **no** puede estar **vacio** **y** tenga una longitud de al menos **10 caracteres**.
 - Luego, se debe verificar que ese **mail exista** en la tabla correspondiente.

Si el mail existe, se debe procesar la **actualización del registro** de la tabla **Usuarios**:

- la clave para este usuario se actualizará con un nuevo valor: el año actual, encriptado con la función MD5(). [el año será extraído con una función de PHP],
- el campo de Activo ahora será 1, pues estamos activando el usuario.
- el campo FechaActivacion será la fecha y hora actual del servidor, usar la función de PHP que devuelve la fecha y hora actual]
- una vez actualizados estos datos, **se deberá insertar en la tabla de logs** estos datos:
 - ✓ fecha y hora actual usando la función de PHP,
 - ✓ usar el valor 1 en el campo "ID_Evento". Revisar la tabla de eventos, el registro id=1 es el "Activacion de cuenta"
 - ✓ el email en cuestión.
- se debe notificar al usuario con el mensaje en color verde dispuesto para este caso.

Si el mail NO existe:

- **se deberá insertar en la tabla de logs** estos datos:
 - ✓ fecha y hora actual usando la función de PHP,
 - ✓ usar el valor 2 en el campo "ID_Evento".
Revisar la tabla de eventos, el registro id=2 es el "Email inexistente para activar "
 - ✓ el email en cuestión.
- se debe notificar al usuario con el mensaje en color amarillo dispuesto para este caso.

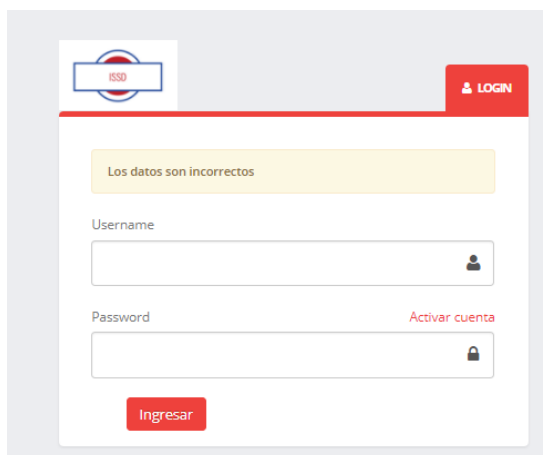
Ingreso al panel

Desde la pantalla de **login** se debe:

- Permitir el ingreso al panel de usuario, validando que el mail **no esté vacío y** tenga al menos **10 caracteres**, además que el **mail exista** en la base, y que se encuentre **activo**. Recordar que el proceso

anterior activa la cuenta con su clave igual al *año* actual, encriptada en MD5.

- Siendo los **datos ingresados correctos**, se deberá **insertar** en la **tabla de logs** estos datos:
 - ✓ fecha y hora actual usando la función de PHP,
 - ✓ usar el valor 3 en el campo "ID_Evento". Revisar la tabla de eventos, el registro id=3 es el " Acceso correcto "
 - ✓ el email en cuestión.
- Abrir **sesión de PHP** y almacenar en ella el valor del mail que está accediendo, y el nivel del usuario. Redireccionar al index.
- Si los **datos son incorrectos**, se deberá **insertar** en la **tabla de Logs** estos datos:
 - ✓ fecha y hora actual usando la función de PHP,
 - ✓ usar el valor 4 en el campo "ID_Evento". Revisar la tabla de eventos, el registro id=3 es el "Datos incorrectos en login"
 - ✓ el email en cuestión.
- Mostrar mensaje en amarillo sólo en caso que los datos ingresados sean incorrectos.



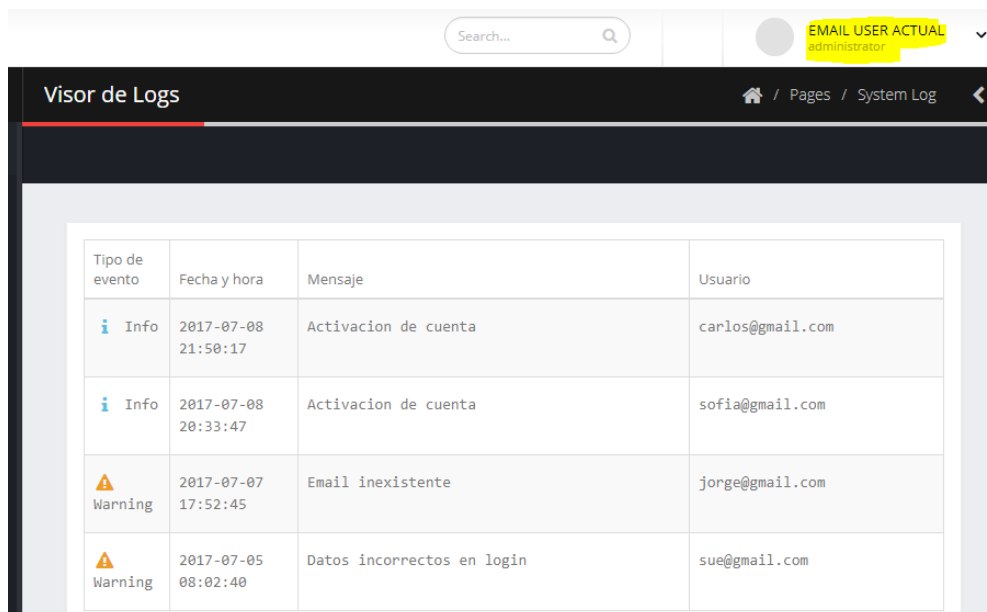
Trabajar usando una **función propia** para cada momento en que se deba **insertar en la tabla de Logs**. Esta función se llamará "**ErrorLogs()**" y deberá tomar los parámetros que consideres necesarios, para poder insertar en la tabla *Logs* los distintos momentos que se fueron mencionando. Dentro de la función deberás usar la sql correspondiente, la conexión, y todo lo que consideres necesario.

Dentro del panel

En la pantalla **index**:

- Validar que si no hay sesión activa, debe dirigir al navegante al login. Es decir, no puedo ingresar directamente a este script, sin haber pasado previamente por el login. [**Manejo de session**]

- Visualizar el **listado de Logs** almacenados en la tabla Logs, ordenados en forma descendente por la fecha, y mostrando la imagen correspondiente según el tipo de evento ocurrido. [1: warning , 2: informacion]. Ubicar los datos en cada celda correspondiente de la tabla.
- Programar para que si el usuario que ingresa es de tipo Admin, deberá ver TODOS los registros de logs: Warnings e Informaciones. Si el usuario *no* es de tipo Admin, deberá ver solamente los Warnings. Analizar la Sql para este comportamiento, puedes usar valores de Session.
- Visualizar el **mail del usuario activo** [valor de session], en la esquina superior derecha. Si es Admin que se vea la palabra "administrator", de lo contrario no.

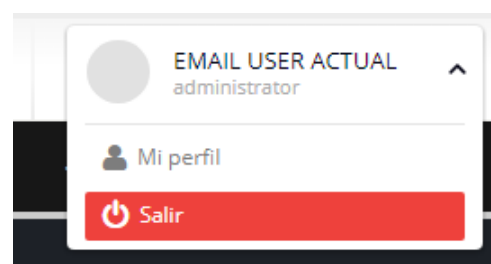


Tipo de evento	Fecha y hora	Mensaje	Usuario
Info	2017-07-08 21:50:17	Activacion de cuenta	carlos@gmail.com
Info	2017-07-08 20:33:47	Activacion de cuenta	sofia@gmail.com
Warning	2017-07-07 17:52:45	Email inexistente	jorge@gmail.com
Warning	2017-07-05 08:02:40	Datos incorrectos en login	sue@gmail.com

Salir del panel

Proceso de **Cerrar sesión**:

- el link de Salir deberá apuntar a un script que destruya la sesión de usuario actual y ubique al navegante en el login



Los scripts podrán seccionarse como deseen, pero debiendo hacerlo una vez al menos, usando **funciones de PHP que permitan la inclusión de scripts**.

El alumno contará como recurso todas las interfaces en HTML, debiendo programar lo necesario en ellas usando PHP. Las templates deberán respetarse.

Porcentajes de aprobación

Activar cuenta		30%
Validar mail	10%	
proceso actualización	10%	
uso de Logging()	5%	
Mensajes al usuario	5%	
Función Logging()		15%
Login:		20%
permitir acceso con restricciones	10%	
abrir sesión y ubicar en el index	5%	
uso de Logging()	5%	
Index:		25%
validar session activa	5%	
listado de Logs según nivel y session	15%	
usuario activo esquina derecha	5%	
Salir:		5%
proceso con session	5%	
Seccionado de scripts		5%

Porcentaje de aciertos	Calificación que le corresponde	Condición
0 a 29	1 (uno)	Desaprobado
30 a 49	2 (dos)	Desaprobado
50 a 59	3 (tres)	Desaprobado
60 a 64	4 (cuatro)	Aprobado
65 a 69	5 (cinco)	Aprobado
70 a 74	6 (seis)	Aprobado
75 a 79	7 (siete)	Aprobado
80 a 89	8 (ocho)	Aprobado
90 a 95	9 (nueve)	Aprobado
96 a 100	10 (diez)	Aprobado