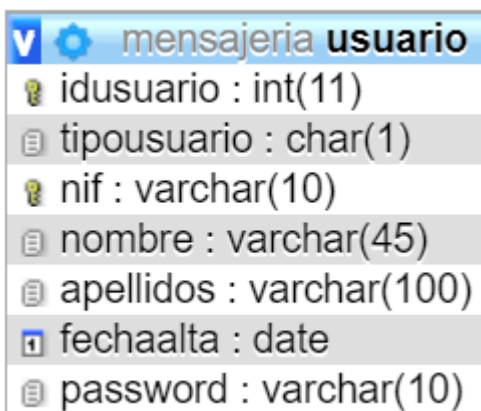


# EJERCICIO MENSAJERIA

## PARTE I: BBDD

Vamos a crear una nueva tabla (parecida a la tabla de pacientes o de invitados que ya tenemos) para crear un programa de mensajería.

En la tabla añadiremos un nuevo campo *password* varchar(12)



A screenshot of a database table definition window. The title bar shows a blue 'v' icon, a gear icon, and the text 'mensajeria usuario'. The table has the following fields:

Field Name	Field Type
idusuario	int(11)
tipousuario	char(1)
nif	varchar(10)
nombre	varchar(45)
apellidos	varchar(100)
fechaalta	date
password	varchar(10)

El campo *nif* char(9) nos permitirá identificarnos en la plataforma

Además crearemos una nueva tabla *mensajes* con los siguientes atributos:

- Idmensaje int PK AI
- Remitente int
- Destinatario int
- Mensaje varchar(300)
- Fechaalta: timestamp

v	mensajeria.mensaje
🔑	idmensaje : int(11)
#	remitente : int(11)
#	destinatario : int(11)
📄	mensaje : varchar(600)
📅	fechaalta : timestamp

## PARTE II: LOGIN

Confeccionaremos una pantalla de login para validar si el usuario y el password existen en la base de datos. Una vez realizado el login se mostrará la pantalla de menu.

El id del usuario (y el tipo de usuario) lo guardaremos en una variable de sesión o en una cookie para poder comprobar en las siguientes pantallas que el usuario se ha conectado previamente

Si entramos a esta pantalla con un login activo conectaremos directamente con la pantalla de menu

### PROGRAMA DE MENSAJERIA

NIF Usuario:

Password:

Login

## PARTE III: MENU

Si hemos realizado el login previo, se nos mostrará la siguiente pantalla de menú de opciones. En caso contrario, si se intenta acceder a esta pantalla sin login previo (no existirá la variable de sesión o la cookie) volveremos a la pantalla de login.

Para comprobar el login lo más usual es realizarlo directamente al entrar en la pantalla comprobando si existe la variable de sesión y, en caso que no exista, redirigir a login con un header location

### **PROGRAMA DE MENSAJERIA**

[Ver mensajes](#)

[Enviar mensajes](#)

[Logoff](#)

En esta pantalla se mostrarán 3 opciones

- Enlace con la pantalla de ver mensajes recibidos por el usuario conectado
- Enlace con la pantalla de enviar mensajes a otros usuarios
- Logoff de la plataforma: borrar variable de sesión o cookie y volver a la pantalla de login

## PARTE IV: CONSULTA MENSAJES

En esta pantalla se mostrarán solo los mensajes de la persona que se ha conectado.

Si intentamos acceder sin login previo se retornará a la pantalla de login

Para ello recuperaremos el id que tenemos guardado en la variable de sesión o en la cookie y lo utilizaremos para consultar los mensajes de la tabla 'mensajes'.

En la lista de mensajes se mostrará el mensaje y el nombre del remitente

Opcionalmente incorporar paginación para que los mensajes se muestren de 5 en 5.

### PROGRAMA DE MENSAJERIA

#### Mensajes recibidos

Autor: david alcolea 2017-05-02 17:45:04

Mensaje 3 de prueba

---

Autor: david alcolea 2017-05-02 18:23:05

Mensaje 5 de prueba

---

[Volver a menu](#)

## PARTE V: ENVIAR MENSAJES

Al entrar en la pantalla se mostrará la lista de personas de la base de datos con un check para cada una de ellas (los checkboxes deberán tener todos el mismo atributo name y/o class y como atributo value el id de cada usuario)

Se rellenará el textarea del mensaje y, al pulsar el botón 'envia mensaje' se grabará en la tabla mensajes tantas filas como usuarios se han seleccionado.

### PROGRAMA DE MENSAJERIA

Seleccionar destinatarios

Nombre	Apellidos	Mandar mensaje
Adolph	Suareznegger	<input type="checkbox"/>
david	alcolea	<input type="checkbox"/>
Doctor	maligno	<input type="checkbox"/>
faemino	y cansado	<input type="checkbox"/>

Enviar mensaje

[Volver a menu](#)

NOTA 1: para enviar el array de usuarios seleccionados CON AJAX y JQUERY:

En la función javascript de enviar mensaje recuperaremos los checks utilizando el bucle each de jquery:

```
$checkboxes=$('#checkbox')
```

```
$idArray=new Array() //creamos un array vacio para guardar los id de los usuarios seleccionados
```

```
$checkboxes.each(function() {  
    If ($(this).prop('checked')) {  
        $idArray.push($(this).val())  
    }  
});
```

Y en la llamada Ajax al servidor pasamos el array generado con el bucle anterior

```
data: {'destinatarios[]': $idArray},
```

Por último, en el fichero php que procesa la llamada Ajax para enviar los mensajes recogemos el array que nos llega por POST y lo recorremos con un foreach para enviar el mensaje a los usuarios seleccionados (INSERT en la tabla mensaje)

```
$destinatarios=$_POST['checkboxes'];
```

```
foreach ($destinatarios as $destinatario)
```

NOTA 2: para enviar el array de usuarios seleccionados CON AJAX y FETCH:

En la función javascript de enviar mensaje recuperaremos los checks utilizando el bucle each de jquery:

```
$checkboxes=document.querySelectorAll('input[type=checkbox]:checked')
```

```
$idArray=new Array() //creamos un array vacio para guardar los id de los usuarios seleccionados
```

```
for (d=0;d< checkboxes.length;d++) {  
    idArray.push(checkboxes [d].value)  
}
```

Y en la llamada Ajax al servidor pasamos el array generado con el bucle anterior

```
datos.append('destinatarios', idArray)
```

Por último, en el fichero php que procesa la llamada Ajax para enviar los mensajes recogemos la lista de usuarios que nos llega por POST y lo transformamos a array utilizando explode

```
$arrayDestinatarios = explode(',', $destinatarios);
```

Lo recorremos con un foreach para enviar el mensaje a los usuarios seleccionados (INSERT en la tabla mensaje)

```
foreach ($destinatarios as $destinatario)
```

NOTA 3: para enviar el array de usuarios seleccionados SIN AJAX:

Al montar la tabla de checkboxes todos ellos deberán tener el mismo atributo `name` con el formato `name='checkboxes[]'`

La tabla de usuarios con sus cheks deberá estar dentro de una etiqueta `<form>`, al igual que el textarea y el botón deberá ser de tipo submit

En el servidor se recogerán los checks y montaremos el foreach para realizar el INSERT en la tabla mensajes

```
$destinatarios=$_POST['checkboxes'];
```

```
foreach ($destinatarios as $destinatario)
```

## PARTE VI: BORRAR MENSAJES PROPIOS

Modificar la pantalla de consulta de mensajes para incluir un checkbox al lado de cada uno y un botón de borrar aquellos que estén seleccionados

# PROGRAMA DE MENSAJERIA

## Mensajes recibidos

☐ Autor: david alcolea 2017-05-02 17:45:04

Mensaje 3 de prueba

---

☐ Autor: david alcolea 2017-05-02 18:23:05

Mensaje 5 de prueba

---

Borrar mensajes

[Volver a menu](#)



## PARTE VII: BORRAR TODOS LOS MENSAJES

Dar de alta un tipo de usuario específico en la tabla de personas (usuario administrador) que pueda borrar cualquier mensaje de la base de datos

Para ello en la tabla de usuarios daremos de alta un nuevo atributo (columna) para identificar si un usuario es administrador

Si el usuario es administrador tendrá una opción adicional en el menú para borrar mensajes de cualquier usuario.

### **PROGRAMA DE MENSAJERIA**

[Ver mensajes](#)

[Enviar mensajes](#)

[Administrar mensajes](#)

[Logoff](#)

Para ello:

Confeccionamos una nueva pantalla de ver mensajes en donde aparezca una combo con todos los usuarios

Al seleccionar un usuario de la combo se mostrarán solo los mensajes que el usuario seleccionado ha enviado con el check y el botón de borrar mensajes

A esta pantalla solo podrá acceder el usuario administrador (comprobaremos que el id que guardamos en la variable de sesión o la cookie pertenezca a un usuario administrador consultando el mismo en la tabla de usuarios al entrar en la página)

# PROGRAMA DE MENSAJERIA

## Mensajes recibidos

Doctor maligno ▾

☐ Autor: jose luis torrente 2012-12-15

mensaje de prueba

---

☐ Autor: jose luis torrente 2012-12-15

mensaje de prueba

---

Borrar mensajes

[Volver a menu](#)