



TRABAJO: **Autónomo** ▾

TEMA: Informe crud que agregar_usuarios_claves_roles

Nombre: Stalin Carrión
Carrera: Desarrollo de Software
Docente: Ing. Jorge Beltrán
Fecha: 31 may 2025

Nivel: Cuarto ▾
Jornada: Matutina ▾
Código: 25251781
Calificación: _____

Inserción de Datos de Usuarios

Previamente, se había creado la tabla **usuarios** en la base de datos, la cual contiene los campos necesarios para almacenar información clave como: nombre de usuario, contraseña encriptada, rol del usuario y la cédula del empleado asociado.

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/> 1	id	int			No	Ninguna		AUTO_INCREMENT
<input type="checkbox"/> 2	usuario	varchar(50)	utf8mb4_0900_ai_ci		No	Ninguna		
<input type="checkbox"/> 3	clave	varchar(255)	utf8mb4_0900_ai_ci		No	Ninguna		
<input type="checkbox"/> 4	rol	enum('admin', 'empleado')	utf8mb4_0900_ai_ci		No	Ninguna		
<input type="checkbox"/> 5	ci_empleado	int			Sí	NULL		
<input type="checkbox"/> 6	fecha_user	timestamp			Sí	CURRENT_TIMESTAMP		DEFAULT_GENERATE

A partir de esta estructura, se desarrolló un formulario en el archivo **insertar.php**, ubicado en la carpeta **vista**, con el propósito de registrar nuevos usuarios en el sistema. El formulario permite seleccionar de una lista a los empleados ya registrados, asegurando así la integridad de los datos y evitando duplicidades.



Para automatizar la asignación del nombre de usuario, se incorporó un script que copia automáticamente la cédula del empleado seleccionado. Esto elimina errores comunes al ingresar manualmente el nombre de usuario.

```
1 document.addEventListener('DOMContentLoaded', function() {  
2     var ciEmpleadoSelect = document.getElementById('ci_empleado');  
3     var usuarioInput = document.getElementById('user');  
4     ciEmpleadoSelect.addEventListener('change', function() {  
5         usuarioInput.value = this.value;  
6     });  
7 });
```

El formulario también incluye un campo para la contraseña, la cual se encripta utilizando `password_hash`, una función nativa de PHP que garantiza la seguridad de la información sensible.

```
1 $hash = password_hash($clave, PASSWORD_DEFAULT);  
2 $stmt = $this->db->prepare("INSERT INTO usuarios (usuario, clave, rol, ci_empleado) VALUES (?, ?, ?, ?)");  
3 return $stmt->execute([$usuario, $hash, $rol, $ci_empleado]);
```

La interfaz resultante es intuitiva y funcional, facilitando el ingreso de nuevos usuarios al sistema.



+ Registrar Usuario

Ver Usuarios

Nombre de usuario

Contraseña

Rol del Usuario

Seleccione un rol

Empleado

Seleccione un empleado

Guardar

Limpiar

Listado de usuarios

+ Agregar usuario

ID	Usuario	Rol de usuario	Empleado	Fecha de registro	Acciones
9	1983657845	empleado	1983657845 - Hola Que hace	2025-06-19 11:04:38	<div>Editar</div> <div>Eliminar</div>
10	1657486151	empleado	1657486151 - Natalia Ceviche	2025-06-19 11:43:33	<div>Editar</div> <div>Eliminar</div>

Funcionamiento del Controlador y el Modelo

Al enviar el formulario, el flujo del proceso se gestiona desde el archivo

`UsuariosController.php`, que recibe la solicitud HTTP, generalmente de tipo POST. Este



controlador valida si la petición es POST y evalúa el valor del campo oculto **accion** para decidir entre dos operaciones: crear o actualizar un usuario.

- **Si la acción es crear**, el controlador recoge los datos del formulario (usuario, contraseña, rol, y cédula del empleado) y los envía al método **crearUsuario()** del modelo **Usuarios.php**. Este método encripta la contraseña con **password_hash** y ejecuta una consulta preparada para insertar los datos en la tabla **usuarios**.
- **Si la acción es actualizar**, se llama al método **actualizarUsuario()** del mismo modelo. Este método valida el campo **id**, aplica el hash a la nueva contraseña y actualiza los datos del usuario correspondiente en la base de datos.

En ambos casos, el controlador finaliza redirigiendo al usuario hacia el listado general, acompañado de un mensaje de confirmación o advertencia, según el resultado de la operación.

Conclusión

Gracias a la estructura previamente definida en la tabla **usuarios** y a la implementación de un formulario seguro y automatizado, se logró un flujo eficiente para la gestión de usuarios dentro del sistema. La separación entre vista, controlador y modelo no solo favorece la organización del código bajo el patrón MVC, sino que también garantiza un manejo ordenado de los datos,



Instituto Superior Tecnológico *Vida Nueva*



reduciendo riesgos de errores y fortaleciendo la seguridad de la aplicación mediante la encriptación de contraseñas.

Este enfoque modular permite escalar el sistema fácilmente en el futuro, agregando nuevas funcionalidades o ajustando validaciones sin comprometer la integridad del proyecto.

Sede Matriz: Guamaní
Av. Maldonado y 4ta transversal
Telf: 307 6032 / 098 208 1291

Campus Norte: Calderón
Av. Carapungo Oe7-312 y Landázuri
Telf: 202 2979 / 096 142 6472

www.vidanueva.edu.ec



Tecnológico Universitario Vida Nueva

