

## Web Api

La Gestión de **Usuarios, Grupos y Acciones** es similar, porque dependen para su persistencia de tablas MySql sin claves foráneas.

### Ingreso de datos

#### HTML

Se parte de un archivo HTML, que consta de un formulario para que el usuario ingrese los datos. El archivo HTML se encuentra vinculado a un archivo JS que se encarga de gestionar la recepción de los datos ingresados.

#### JS

A partir de un eventlistener, se espera la pulsación del botón submit del formulario para ejecutar las siguientes tareas:

1. Se recuperan los valores ingresados por el usuario.
2. Se incluyen estos valores en una petición JSON al archivo PHP encargado del backend.
3. Se comunica al usuario el resultado de la petición JSON y de la operación de inserción en la base de datos realizada en el backend.

En el mismo archivo JS también se gestiona el botón de regreso al menú principal.

#### PHP

Este archivo incluye las cabeceras necesarias para la recepción de la petición JSON y los datos necesarios para realizar la conexión con la base de datos.

Este archivo realiza las siguientes tareas:

1. Conexión con la base de datos.
2. Recepción de los parámetros que acompañan la petición JSON.
3. Inserción de los datos en la tabla correspondiente.
4. Envío de la respuesta JSON.
5. Cierre de la conexión.

### Borrado de datos

#### HTML

Se parte de un archivo HTML, que consta de un formulario para que el usuario ingrese el id perteneciente al registro a borrar. El archivo HTML se encuentra vinculado a un archivo JS que se encarga de gestionar la recepción del id ingresado.

## JS

A partir de un eventlistener, se espera la pulsación del botón submit del formulario para ejecutar las siguientes tareas:

1. Se recuperan los valores ingresados por el usuario.
2. Se incluyen estos valores en una petición JSON al archivo PHP encargado del backend.
3. Se comunica al usuario el resultado de la petición JSON y de la operación de borrado en la base de datos realizada en el backend.

En el mismo archivo JS también se gestiona el botón de regreso al menú principal.

## PHP

Este archivo incluye las cabeceras necesarias para la recepción de la petición JSON y los datos necesarios para realizar la conexión con la base de datos.

Este archivo realiza las siguientes tareas:

1. Conexión con la base de datos.
2. Recepción de los parámetros que acompañan la petición JSON.
3. Verificar la existencia del registro mediante una sentencia COUNT
4. Borrado de los datos en la tabla correspondiente.
5. Envío de la respuesta JSON.
6. Cierre de la conexión.

## Listado de datos

### HTML

Se parte de un archivo HTML, que consta de una etiqueta **div** reservada para contener una tabla que se producirá dinámicamente.

El archivo HTML se encuentra vinculado a un archivo JS que se encarga de solicitar los datos a la base de datos y producir una tabla con los mismos.

### JS

Realiza las siguientes tareas:

1. Realiza una petición JSON al archivo PHP encargado del backend.
2. Crea una tabla con los datos recuperados.

En el mismo archivo JS también se gestiona el botón de regreso al menú principal.

## **PHP**

Este archivo incluye las cabeceras necesarias para la recepción de la petición JSON y los datos necesarios para realizar la conexión con la base de datos.

Este archivo realiza las siguientes tareas:

1. Conexión con la base de datos.
2. Solicitud de todos los datos de la tabla correspondiente.
3. Envío de la respuesta JSON.
4. Cierre de la conexión.

## **Modificación de datos**

### **HTML**

El archivo HTML contiene dos formularios:

- El primero para que el usuario ingrese el id perteneciente al registro a modificar.
- El segundo muestra los datos del registro correspondiente y habilita su modificación.

El archivo HTML se encuentra vinculado a un archivo JS que se encarga de gestionar la recepción del id ingresado y la recepción de los datos modificados.

### **JS**

Realiza las siguientes tareas:

1. Se recupera el id ingresado por el usuario.
2. Se incluye este valor en una petición JSON al archivo PHP encargado del backend.
3. Se comunica al usuario el resultado de la petición JSON.
4. Se completan los inputs type del segundo formulario con los datos recuperados.
5. Se reciben los datos modificados por el usuario.
6. Realiza una petición JSON al archivo PHP encargado del backend.
7. Se comunica al usuario el resultado de la modificación de los datos en la base de datos.

En el mismo archivo JS también se gestiona el botón de regreso al menú principal.

## **PHP**

Se emplean dos archivos PHP:

El primero recupera los datos del registro correspondiente al id ingresado.

Este archivo incluye las cabeceras necesarias para la recepción de la petición JSON y los datos necesarios para realizar la conexión con la base de datos.

Este archivo realiza las siguientes tareas:

1. Conexión con la base de datos.
2. Solicitud de todos los datos de la tabla correspondiente.
3. Envío de la respuesta JSON.
4. Cierre de la conexión.

El segundo archivo realiza las siguientes tareas:

1. Conexión con la base de datos.
2. Recepción de los parámetros de la petición JSON.
3. Actualización de los datos en la tabla correspondiente.
4. Envío de la respuesta JSON.
5. Cierre de la conexión.

La Gestión de **Puestos** y **Permisos** es similar, porque son tablas intermedias que representan relaciones de uno a muchos por lo que contienen claves foráneas. Los datos que aportan los usuarios deben existir en las tablas Usuarios, Grupos y Acciones por lo que el ingreso o modificación de datos se realiza a partir de selects de datos ya existentes.

## Ingreso de datos

### HTML

Se parte de un archivo HTML, que consta de dos campos select que se completaran dinámicamente. El archivo HTML se encuentra vinculado a un archivo JS que se encarga de gestionar la recepción de los datos ingresados.

### JS

La carga completa de la pagina desencadena dos peticiones JSON para completar los campos select del formulario.

Pulsado el botón submit del formulario:

1. Se recuperan los valores ingresados por el usuario.
2. Se incluyen estos valores en una petición JSON al archivo PHP encargado del backend.
3. Se comunica al usuario el resultado de la petición JSON y de la operación de inserción en la base de datos realizada en el backend.

En el mismo archivo JS también se gestiona el botón de regreso al menú principal.

## **PHP**

Hay tres archivos PHP involucrados:

Los dos primeros responden a las peticiones JSON para cargar los campos select del segundo formulario.

El tercer formulario recibe los datos modificados por el usuario y los inserta en la base de datos.

## **Borrado de datos**

### **HTML**

Se parte de un archivo HTML, que consta de un formulario para que el usuario ingrese el id perteneciente al registro a borrar. El archivo HTML se encuentra vinculado a un archivo JS que se encarga de gestionar la recepción del id ingresado.

### **JS**

A partir de un eventlistener, se espera la pulsación del botón submit del formulario para ejecutar las siguientes tareas:

1. Se recuperan el valor ingresado por el usuario.
2. Se incluye este valor en una petición JSON al archivo PHP encargado del backend.
3. Se comunica al usuario el resultado de la petición JSON y de la operación de borrado en la base de datos realizada en el backend.

En el mismo archivo JS también se gestiona el botón de regreso al menú principal.

### **PHP**

Este archivo incluye las cabeceras necesarias para la recepción de la petición JSON y los datos necesarios para realizar la conexión con la base de datos.

Este archivo realiza las siguientes tareas:

1. Conexión con la base de datos.
2. Recepción de los parámetros que acompañan la petición JSON.
3. Verificar la existencia del registro mediante una sentencia COUNT
4. Borrado de los datos en la tabla correspondiente.
5. Envío de la respuesta JSON.
6. Cierre de la conexión.

## Listado de datos

### HTML

Se parte de un archivo HTML, que consta de una etiqueta **div** reservada para contener una tabla que se producirá dinámicamente.

El archivo HTML se encuentra vinculado a un archivo JS que se encarga de solicitar los datos a la base de datos y producir una tabla con los mismos.

### JS

Realiza las siguientes tareas:

1. Realiza una petición JSON al archivo PHP encargado del backend.
2. Crea una tabla con los datos recuperados.

En el mismo archivo JS también se gestiona el botón de regreso al menú principal.

### PHP

Este archivo incluye las cabeceras necesarias para la recepción de la petición JSON y los datos necesarios para realizar la conexión con la base de datos.

Este archivo realiza las siguientes tareas:

1. Conexión con la base de datos.
2. Solicitud de todos los datos de la tabla correspondiente.
3. Envío de la respuesta JSON.
4. Cierre de la conexión.

## Modificación de datos

### HTML

El archivo HTML contiene dos formularios:

- El primero para que el usuario ingrese el id perteneciente al registro a modificar.
- El segundo esta formado por dos campos select que se completan dinámicamente y habilita su modificación.

El archivo HTML se encuentra vinculado a un archivo JS que se encarga de gestionar la recepción del id ingresado y la recepción de los datos modificados.

### JS

Realiza las siguientes tareas:

1. Se recupera el id ingresado por el usuario.
2. Se incluye este valor en una petición JSON al archivo PHP encargado del backend.
3. Se comunica al usuario el resultado de la petición JSON.
4. Se crea una tabla con el registro recuperado.
5. Se realizan dos peticiones JSON para recuperar los datos de las tablas que completaran los selects.

6. Se completan los inputs type del segundo formulario con los datos recuperados.
7. Se reciben los datos modificados por el usuario.
8. Realiza una petición JSON al archivo PHP encargado del backend.
9. Se comunica al usuario el resultado de la modificación de los datos en la base de datos.

En el mismo archivo JS también se gestiona el botón de regreso al menú principal.

## **PHP**

Se emplean 4 archivos PHP:

El primero recupera los datos del registro correspondiente al id ingresado.

Este archivo incluye las cabeceras necesarias para la recepción de la petición JSON y los datos necesarios para realizar la conexión con la base de datos.

Este archivo realiza las siguientes tareas:

1. Conexión con la base de datos.
2. Solicitud de todos los datos de la tabla correspondiente.
3. Envío de la respuesta JSON.
4. Cierre de la conexión.

El segundo y tercer archivo realizan las siguientes tareas:

1. Conexión con la base de datos.
2. Recepción de los parámetros de la petición JSON.
3. Recuperación de los registros de las tablas que completan los select.
4. Envío de la respuesta JSON.
5. Cierre de la conexión.

El cuarto archivo realiza las siguientes tareas:

6. Conexión con la base de datos.
7. Recepción de los parámetros de la petición JSON.
8. Actualización del registro de la tabla correspondiente con los datos modificados.
9. Envío de la respuesta JSON.
10. Cierre de la conexión.

	CREATE	READ	UPDATE	DELETE
USUARIOS	insertar_usuario.php	listado_usuario.php	obtener_usuario.php actualizar_usuario.php	borrado_usuario.php
GRUPOS	insertar_grupo.php	listado_grupo.php	obtener_grupo.php actualizar_grupo.php	borrado_grupo.php
ACCIONES	insertar_accion.php	listado_accion.php	obtener_accion.php actualizar_accion.php	borrado_accion.php
PUESTOS	getUsuarioPuesto.php getGrupoPuesto.php insertar_puesto.php	listado_puesto.php	obtener_puesto.php actualizado_puesto.php	borrado_puesto.php
PERMISOS	getGrupoPermiso.php getAccionPermiso.php insertar_permiso.php	listado_permiso.php	obtener_permiso.php actualizaddo_permiso.php	Borrado.permiso.php