

## PRÁCTICA PHP

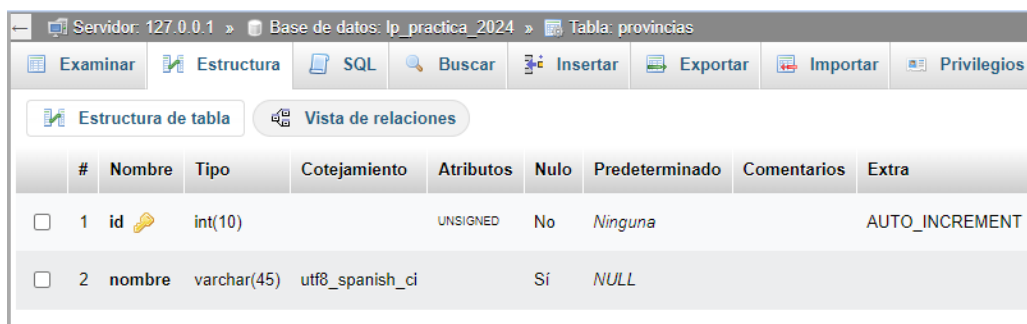
### Objetivo:

- Desarrollar una aplicación para gestionar usuarios y clientes, mediante un acceso restringido a través de sesiones.
- Implementar las operaciones CRUD en una Base de Datos MySQL a través de una interfaz Web, utilizando PHP como lenguaje de Scripting del lado del servidor.

### EJERCICIOS

- 1) Crear una base de datos con el nombre **“lp\_practica\_2024”**. La misma deberá contener las siguientes tablas:

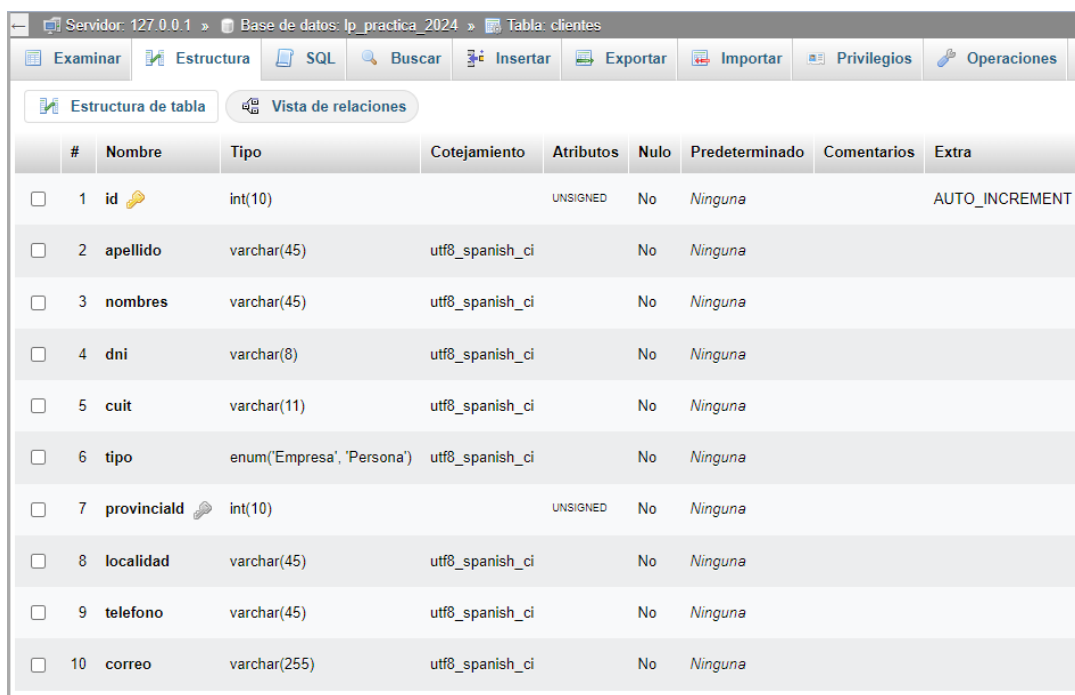
- a) La tabla **“provincias”**, la cual deberá respetar la siguiente estructura:



The screenshot shows the MySQL Table Structure for the 'provincias' table. The table has two columns: 'id' (int(10), UNSIGNED, No NULL, AUTO\_INCREMENT) and 'nombre' (varchar(45), utf8\_spanish\_ci, Sí NULL).

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
1	id	int(10)		UNSIGNED	No	Ninguna		AUTO_INCREMENT
2	nombre	varchar(45)	utf8_spanish_ci		Sí	NULL		

- b) La tabla **“clientes”**, la cual deberá respetar la siguiente estructura:



The screenshot shows the MySQL Table Structure for the 'clientes' table. The table has ten columns: 'id' (int(10), UNSIGNED, No NULL, AUTO\_INCREMENT), 'apellido' (varchar(45), utf8\_spanish\_ci, No NULL), 'nombres' (varchar(45), utf8\_spanish\_ci, No NULL), 'dni' (varchar(8), utf8\_spanish\_ci, No NULL), 'cuit' (varchar(11), utf8\_spanish\_ci, No NULL), 'tipo' (enum('Empresa', 'Persona'), utf8\_spanish\_ci, No NULL), 'provinciald' (int(10), UNSIGNED, No NULL), 'localidad' (varchar(45), utf8\_spanish\_ci, No NULL), 'telefono' (varchar(45), utf8\_spanish\_ci, No NULL), and 'correo' (varchar(255), utf8\_spanish\_ci, No NULL).

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
1	id	int(10)		UNSIGNED	No	Ninguna		AUTO_INCREMENT
2	apellido	varchar(45)	utf8_spanish_ci		No	Ninguna		
3	nombres	varchar(45)	utf8_spanish_ci		No	Ninguna		
4	dni	varchar(8)	utf8_spanish_ci		No	Ninguna		
5	cuit	varchar(11)	utf8_spanish_ci		No	Ninguna		
6	tipo	enum('Empresa', 'Persona')	utf8_spanish_ci		No	Ninguna		
7	provinciald	int(10)		UNSIGNED	No	Ninguna		
8	localidad	varchar(45)	utf8_spanish_ci		No	Ninguna		
9	telefono	varchar(45)	utf8_spanish_ci		No	Ninguna		
10	correo	varchar(255)	utf8_spanish_ci		No	Ninguna		

c) La tabla “**perfiles**”, la cual deberá respetar la siguiente estructura:

Servidor: 127.0.0.1 » Base de datos: lp\_practica\_2024 » Tabla: perfiles

Examinar

Estructura

SQL

Buscar

Insertar

Exportar

Importar

Privilegios

Estructura de tabla

Vista de relaciones

	#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/>	1	id	int(10)		UNSIGNED	No	Ninguna		AUTO_INCREMENT
<input type="checkbox"/>	2	nombre	varchar(45)	utf8_spanish_ci		No	Ninguna		

d) La tabla “**usuarios**”, la cual deberá respetar la siguiente estructura:

Servidor: 127.0.0.1 » Base de datos: lp\_practica\_2024 » Tabla: usuarios

Examinar

Estructura

SQL

Buscar

Insertar

Exportar

Importar

Privilegios

Estructura de tabla

Vista de relaciones

	#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/>	1	id	int(10)		UNSIGNED	No	Ninguna		AUTO_INCREMENT
<input type="checkbox"/>	2	apellido	varchar(45)	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/>	3	nombres	varchar(45)	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/>	4	cuenta	varchar(45)	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/>	5	correo	varchar(255)	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/>	6	clave	varchar(255)	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/>	7	perfilId	int(10)		UNSIGNED	No	Ninguna		
<input type="checkbox"/>	8	estado	tinyint(1)			No	Ninguna		
<input type="checkbox"/>	9	horaEntrada	time			Sí	NULL		
<input type="checkbox"/>	10	horaSalida	time			Sí	NULL		
<input type="checkbox"/>	11	fechaAlta	datetime			No	Ninguna		
<input type="checkbox"/>	12	resetear	tinyint(1)			No	Ninguna		

- 2) Defina la estructura de directorios del proyecto, teniendo en cuenta que desarrollará una aplicación Web con un patrón de diseño MVC, separando el *front-end* del *back-end* de la aplicación.
- 3) Realice una maquetación Web prototipo, en la cual se definan claramente las diferentes secciones que se reutilizarán posteriormente con el uso de plantillas en PHP.

- 4) Para todos los módulos de la aplicación diseñe sus correspondientes vistas en *app/resources/views*, las cuales se mostrarán a través de la plantilla *template.php*. Utilice *Bootstrap* como *framework* CSS. Mínimamente deberá diseñar las siguientes vistas:
- a) Interfaz de alta de registros para los módulos de *usuarios*, *perfiles*, y *clientes*.
  - b) Interfaz de consulta y edición para los módulos de *usuarios*, *perfiles*, y *clientes*.
  - c) Interfaz principal para los módulos de *usuarios*, *perfiles* y *clientes*. La vista de inicio de cada módulo debe organizar el listado de registros y las operaciones que se pueden realizar.
  - d) Interfaz de inicio de la aplicación. El contenido a mostrarse queda a criterio del alumno. Por ejemplo información sobre operaciones recientes, estadísticas sobre los módulos, etc.
  - e) Interfaz de autenticación para el ingreso al sistema, a través de usuario y contraseña.
  - f) Además, tenga en cuenta incorporar como vista nueva o parte de una existente, las interfaces para habilitar y deshabilitar una cuenta, y resetear la contraseña de una cuenta.
  - g) Tenga en cuentas las operaciones para eliminar clientes y usuarios. Proponga, diseñe e implemente la vista para que el usuario pueda hacer uso de las operaciones de baja. Tenga en cuenta que al ser una operación crítica, se deberá mostrar un vista previa y/o confirmación de la acción por parte del usuario.
- 5) Agregue al *front* la funcionalidad correspondiente para cada módulo, a través de *JavaScript*. Teniendo en cuenta lo siguiente:
- a) Todos los formularios de la aplicación deberán validar los datos ingresados por el usuario y mostrar los mensajes correspondientes.
  - b) Organice el código y las funcionalidades *JavaScript* usando una representación por dominio (una carpeta por cada módulo) y dentro de cada dominio por controlador y servicio.
  - c) Todas las operaciones que el usuario realice sobre la aplicación deberán tener una notificación de respuesta por parte de la aplicación. Por ejemplo, para las operaciones de ABM se mostrará información sobre el resultado de las mismas.
- 6) Aplique el patrón estructural DTO para encapsular la información que se intercambiará entre las diferentes capas del *back* de la aplicación. Genere un DTO por cada tabla en la base de datos.
- 7) Aplique el patrón estructural DAO para manejar las operaciones de acceso y manipulación de datos. Genere un DAO por cada tabla en la base de datos. Cada DAO deberá proveer los siguientes métodos:
- a) *load()*
  - b) *save()*
  - c) *update()*
  - d) *delete()*
  - e) *list()*



- (2) Cuenta
- (3) Correo
- (4) Perfil
- (5) Fecha de alta

- b)** Se puedan exportar en formato PDF el listado de clientes y usuarios. Los reportes deberán respetar el siguiente formato:
- i) Formato tamaño A4.
  - ii) Debe tener un título principal.
  - iii) Debe indicar la fecha de emisión del reporte.
  - iv) Los datos deben estar tabulados en formato de tabla.