

COMPETENCIA

CONSTRUIR EL SISTEMA QUE CUMPLA CON LOS
REQUISITOS DE LA SOLUCIÓN
INFORMÁTICA.

LUIS FELIPE LONDOÑO GUARIN

SERVICIO NACIONAL DE APRENDIZAJE (SENA)

**TECNOLOGIA EN ANALISIS Y DESARROLLO DE
SOFTWARE**

2024

TEMAS

1. Estructura PHP
2. [POST , GET] - [EMPTY , ISSET]
3. Conexión a Base de datos
4. Insert- Registro con formulario a base de datos
5. Select- Consulta General y detallada
6. Update-Modificar Datos desde la BD
7. Delete- Eliminar registros
8. Framework **LARAVEL**

POST Y GET

Los métodos HTTP GET y HTTP POST permiten enviar información al servidor, en PHP se administra mediante los arrays `$_GET` y `$_POST`

METODO GET

El **método GET** envía la información codificada del usuario en el *header* del **HTTP request**, directamente en la [URL](#). La página web y la información codificada se separan por un interrogante ?:

- El método GET envía la información en la propia URL, estando limitada a 2000 caracteres.
- La información es visible por lo que con este método nunca se envía **información sensible**.
- No se pueden enviar **datos binarios** (archivos, imágenes...).
- En **PHP** los datos se administran con el array asociativo **\$_GET**.

METODO POST

Con el método **HTTP POST** también se codifica la información, pero ésta se envía a través del **body** del **HTTP Request**, por lo que no aparece en la URL.

- El método POST no tiene **límite de cantidad de información** a enviar.
- La información proporcionada no es visible, por lo que se puede enviar **información sensible**.
- Se puede usar para enviar **texto normal** así como **datos binarios** (archivos, imágenes...).
- PHP proporciona el array asociativo **\$_POST** para acceder a la información enviada.

METODO **isset** y **empty**

A **isset** no le importa el valor de la variable que estamos probando, sólo le importa que no sea nulo. Esto viene bien cuando no nos importa el tipo de valor que tiene la variable, y sólo necesitamos saber si existe.

empty es un poco mejor pero también peligrosa. A esta sí le importa el valor de la variable, pero tiene sus preferencias en cuanto a decidir si está vacía o no.

CONEXIÓN A B.D

Variable de
conexión

Servidor

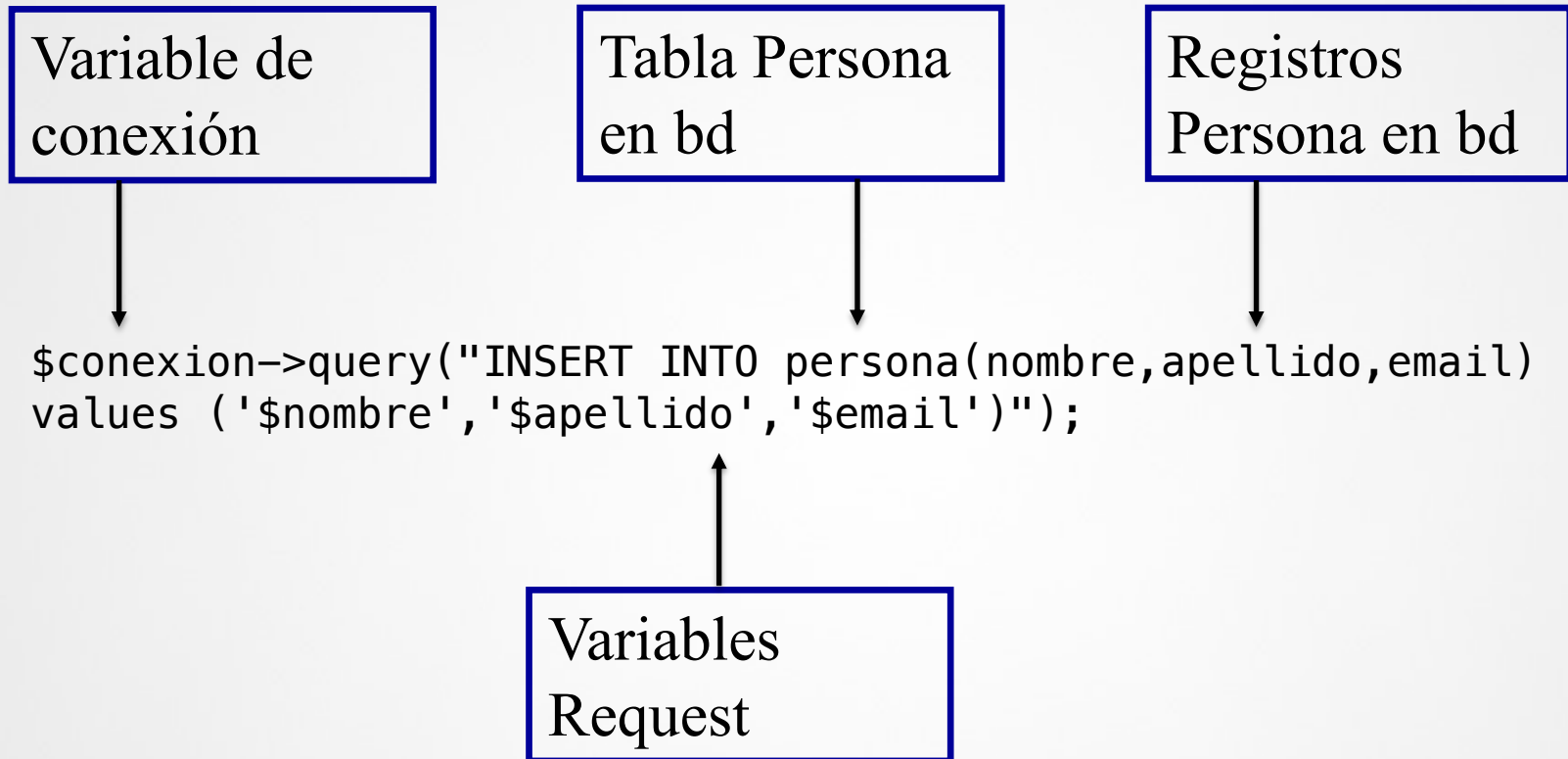
Usuario

Contraseña

```
$conexion=new mysqli("localhost","root","","colegio");
```

Base de Datos

INSERT



Formulario persona

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <form action="ingresar_persona.php" method="POST">
    Nombre: <input type="text" name="nombre">
    Apellido: <input type="text" name="apellido">
    Email: <input type="text" name="email">

    <input type="submit" value="Registrar">
  </form>
</body>
</html>
```

Conexión a BD e Insert

```
<?php
    $conexion = new mysqli('localhost','root','','adso');

    $nombre=$_POST['nombre'];
    $apellido=$_POST['apellido'];
    $email=$_POST['email'];

    $conexion->query("INSERT INTO persona(nombre,apellido,email)
    values ('$nombre','$apellido','$email')");

    echo "La persona fue ingresada correctamente";

?>
```