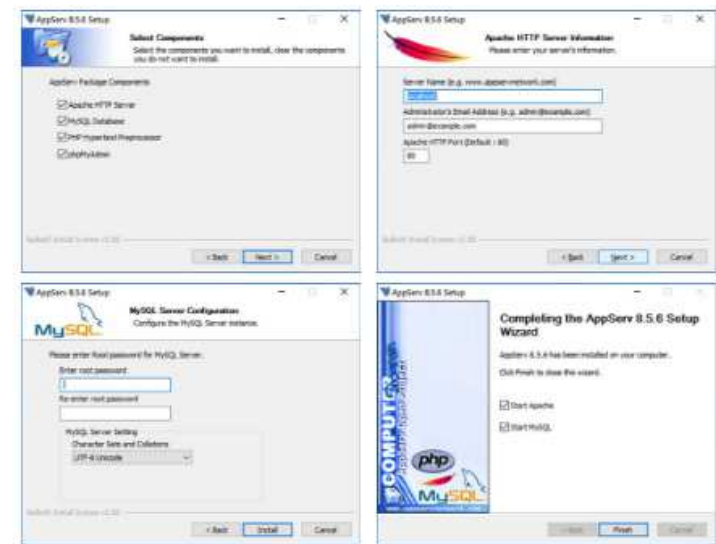


## OVI PARA DISEÑAR UN OBJETO VIRTUAL DE INFORMACIÓN - OVI

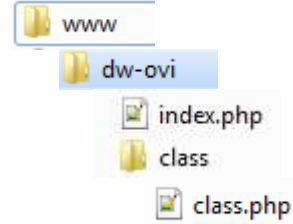
### Objetivo General

**Conectar base de datos implementando el lenguaje de programación PHP, usando base de datos MYSQL.  
Crear, consultar, modificar y eliminar registros.**

### Contenido informativo de la presentación en PPT para diseñar el OVI

Descripción	Sección de imágenes
<p>Para realizar la conexión a una base de datos MYSQL usando el lenguaje de programación PHP, es necesario tener instalado un servidor de aplicaciones ya puede ser APPSERVER (este es el que usaremos) o XAMPP entre otras muchas opciones.</p> <p>Con este servidor de aplicaciones, se configura la base de datos MYSQL y el lenguaje de programación PHP de acuerdo a la versión descargada.</p> <p>En la configuración, le pedirá que configure un usuario root para el acceso a la base de datos, usted tiene la opción de colocar una contraseña (Recomendable) o no hacerlo.</p> <p>También puede configurar el puerto por el cual el servidor de aplicaciones accederá a la aplicación que esté realizando.</p>	

Crear base de datos	Sección de imágenes
<p>Es necesario crear una base de datos para dar inicio a crear el proyecto PHP con la configuración necesaria para realizar la conexión.</p> <p>En este caso se creara una base de datos con una sola tabla llamada usuarios, que consta de 6 campos, los cuales son:</p> <ol style="list-style-type: none"> <li>1. cdusuario -&gt; hace referencia al número de documento de la persona.</li> <li>2. dstipo_documento -&gt; Describe el tipo de documento de la persona.</li> <li>3. dsnombre -&gt; Descripción del nombre de la persona.</li> <li>4. dsapellido -&gt; Descripción del apellido de la persona.</li> <li>5. dsclave -&gt; Describe la clave de acceso de la persona.</li> <li>6. snacrivo -&gt; Describe si la persona se encuentra activo (S) o no lo está (N).</li> </ol>	<pre>CREATE SCHEMA IF NOT EXISTS `dw_ovi` DEFAULT CHARACTER SET latin1 COLLATE latin1_spanish_ci ; USE `dw_ovi`;  CREATE TABLE IF NOT EXISTS `dw_ovi`.`usuarios` (   `cdusuario` INT(11) NOT NULL ,   `dstipo_documento` VARCHAR(2) NOT NULL ,   `dsnombre` VARCHAR(45) NOT NULL ,   `dsapellido` VARCHAR(45) NOT NULL ,   `dsclave` VARCHAR(45) NOT NULL ,   `snactivo` VARCHAR(1) NOT NULL DEFAULT 'S',   PRIMARY KEY (`cdusuario`)) ENGINE = InnoDB DEFAULT CHARACTER SET = latin1 COLLATE = latin1_spanish_ci;</pre>

Estructura de carpetas	Sección de imágenes
<p>Cuando se haya instalado el servidor de aplicaciones APPSERVER, el instalador crea una carpeta (www) por defecto en el disco local (C:), en esta carpeta es donde crearemos el proyecto, la estructura de carpetas es la siguiente:</p> <ul style="list-style-type: none"> <li>➤ Carpeta principal(Nombre del proyecto)</li> <li>➤ Dentro de la carpeta principal, va el archivo index.php que es el archivo principal del proyecto para que el navegador lo interprete.</li> <li>➤ Dentro de la carpeta principal, debe ir otra carpeta con el nombre class</li> <li>➤ Dentro de la carpeta class, va el archivo de configuración para la conexión a la base de datos con toda la gestión.</li> </ul> <p>En el archivo class.php, colocamos las etiquetas que corresponden al lenguaje</p>	 <pre>graph TD     www[www] --&gt; dw_ovi[dw-ovi]     dw_ovi --&gt; index_php[index.php]     dw_ovi --&gt; class[class]     class --&gt; class_php[class.php]</pre>

PHP y en ella crearemos una clase para configurar las credenciales de acceso a la base de datos. Los String de conexión son configurados con las palabras reservadas de PHP para la conexión con MYSQL, MYSQL\_CONNECT es donde viene la forma de acceder, usuario y contraseña. MYSQL\_SELECT\_DB es donde se selecciona la base de datos que vamos a trabajar.

Teniendo la conexión, procedemos a crear nuestra clase de trabajo, para ello lo haremos en el mismo archivo class.php, donde crearemos los métodos crear, modificar, consultar y eliminar usuario.

Cada método tiene unos parámetros que son necesarios para la realización de las sentencias SQL.

En el método crearUsuario encontramos la sentencia INSERT para crear un nuevo usuario, donde la palabra reservada MYSQL\_QUERY() que es el que toma la sentencia SQL y la conexión y la envía al servidor para que realice la inserción a la base de datos, tenemos un ECHO que nos imprime un mensaje y nos envía a la página principal index.php.

En el método consultarUsuarios encontramos la palabra reservada MYSQL\_FETCH\_ASSOC que es una de las formas que tiene para consultar y armar el retorno del resultado de la consulta, esta opción la usaremos ya que nos devuelve un arreglo de dos dimensiones, la primera dimensión es para referencia la posición del arreglo y la segunda dimensión es el nombre que proviene de la base de datos (campo de la tabla), este dato es retornado para que se muestre en la página principal index.php.

Los siguientes métodos tienen los parámetros necesarios para ejecutar las sentencias SQL, que son UPDATE (Modificar) y DELETE (Eliminar).

```
<?php

class Conectar {

    public static function conn() {
        $conexion = mysql_connect("localhost", "dw", "1234");
        mysql_query("SET NAMES 'utf-8'");
        mysql_select_db("dw_ovi");
        return $conexion;
    }
}

class Trabajo {

    private $usuarios;

    public function __construct() {
        $this->usuarios = array();
    }

    public function crearUsuario($dstipo_documento, $cdusuario, $denombre, $dsapellido, $dsclave, $snactivo) {
        $sql = "INSERT INTO usuarios (cdusuario,dstipo_documento,dnombre"
            . ",dsapellido,dclave,snactivo)"
            . "VALUES($cdusuario,'$dstipo_documento','$denombre','$dsapellido'"
            . ", '$dsclave','$snactivo')";

        $res = mysql_query($sql, Conectar::conn());
        echo "<script type='text/javascript'>"
            . " alert('Usuario creado!');"
            . "window.location='index.php';"
            . "</script>";
    }

    public function consultarUsuarios() {
        $sql = "SELECT * FROM usuarios";
        $res = mysql_query($sql, Conectar::conn());
        while ($reg = mysql_fetch_assoc($res)) {
            $this->usuarios[] = $reg;
        }
        return $this->usuarios;
    }
}
```

Teniendo la configuración necesaria, procederemos a crear la página principal index.php, para trabajar desde el formulario, donde encontraremos la forma de acceder a cada método creado en el archivo class.php.

En el archivo principal index.php, trabajaremos lenguaje HTML y PHP de los cuales se puede decir que son un complemento.

Se crea la estructura del lenguaje HTML donde crearemos un formulario con una tabla que tiene los campos creados en la base de datos:

1. Tipo de documento -> una lista desplegable.
2. Documento -> campo numérico para el registro del número de documento.
3. Nombre -> campo tipo texto para la captura del nombre.
4. Apellido -> campo tipo texto para la captura del apellido.
5. Clave -> Campo tipo password para la captura de la clave.
6. Activo -> Lista desplegable con las opciones S (activo) N (inactivo).
7. Tipo de acción -> Lista desplegable con las opciones CREAR, MODIFICAR Y ELIMINAR las cuales hacen referencia a los métodos a los cuales la aplicación va a acceder.

Para validar los datos y su obligatoriedad utilizamos HTML5 con la opción required="true".

Se debe crear una tabla para visualizar la información insertada, modificada y eliminada a través de la aplicación.

```
}

public function modificarUsuario($dstipo_documento, $cdusuario, $dsnombre,
    $dsapellido, $dsclave, $snactivo) {
    $sql = "UPDATE usuarios "
        . "SET dsnombre = '$dsnombre'"
        . ", dsapellido = '$dsapellido'"
        . ", dsclave = '$dsclave'"
        . ", snactivo = '$snactivo'"
        . " WHERE dstipo_documento = '$dstipo_documento'"
        . " AND cdusuario = '$cdusuario'";
    $res = mysql_query($sql, Conectar::conn());

    echo "<script type='text/javascript'>"
        . " alert('Usuario Modificado!');"
        . "window.location='index.php';"
        . "</script>";
}

public function eliminarUsuario($dstipo_documento, $cdusuario) {
    $sql = "DELETE FROM usuarios WHERE cdusuario = '$cdusuario' "
        . "AND dstipo_documento = '$dstipo_documento'";
    $res = mysql_query($sql, Conectar::conn());

    echo "<script type='text/javascript'>"
        . " alert('Usuario Eliminado!');"
        . "window.location='index.php';"
        . "</script>";
}
```

## Index.php

```
<!DOCTYPE html>
<?php
require_once("class/class.php");
$tra = new Trabajo();

if (isset($_POST["grabar"]) and isset($_POST["grabar"]) == "si") {
    echo $_POST["dstipo_accion"];

    if (isset($_POST["dstipo_accion"]) and isset($_POST["dstipo_accion"]) == "C")
        echo "Creando...";
    $tra->crearUsuario($_POST["dstipo_documento"], $_POST["cdusuario"],
        $_POST["dsnombre"], $_POST["dsapellido"], $_POST["dsclave"],
        $_POST["snactivo"]);
}

if ($_POST["dstipo_accion"] == "M") {
    echo "Modificando...";
    $tra->modificarUsuario($_POST["dstipo_documento"], $_POST["cdusuario"],
        $_POST["dsnombre"], $_POST["dsapellido"], $_POST["dsclave"],
        $_POST["snactivo"]);
}

if ($_POST["dstipo_accion"] == "E") {
    echo "Eliminando...";
    $tra->eliminarUsuario($_POST["dstipo_documento"], $_POST["cdusuario"]);
}
}
```

```
<html>
<head>
  <meta charset="UTF-8">
  <title>Gestion de usuarios</title>
</head>
<body>
  <form method="post" action="index.php">
    <h1>GESTION DE USUARIOS</h1>
    <table border="0">
      <tr>
        <td>Tipo documento</td>
        <td>
          <select name="dstipo_documento" required="true">
            <option value="">Seleccione opcion</option>
            <option value="CC">Cedula</option>
            <option value="TI">Tarjeta de identidad</option>
          </select>
        </td>
      </tr>
      <tr>
        <td>Documento:</td>
        <td>
          <input type="number" name="cdusuario" required="true"/>
        </td>
      </tr>
      <tr>
        <td>Nombre:</td>
        <td>
          <input type="text" name="dsnombre" max="40" required="true"/>
        </td>
      </tr>
      <tr>
        <td>Apellido:</td>
        <td>
          <input type="text" name="dsapellido" max="40" required="true"/>
        </td>
      </tr>
      <tr>
        <td>Clave:</td>
        <td>
          <input type="password" name="dsclave" max="4" required="true"/>
        </td>
      </tr>
      <tr>
        <td>Activo:</td>
        <td>
          <select name="sactivo" required="true">
            <option value="">Seleccione opcion</option>
            <option value="S">Si</option>
            <option value="N">No</option>
          </select>
        </td>
      </tr>
      <tr>
        <td>Tipo de acción:</td>
        <td>
          <select name="dstipo_accion" required="true">
            <option value="">Seleccione opcion</option>
          </select>
        </td>
      </tr>
    </table>
  </form>
</body>
</html>
```

```

        <option value="">Selecione opcion</option>
        <option value="C">Crear</option>
        <option value="M">Modificar</option>
        <option value="E">Eliminar</option>
    </select>
</td>
</tr>
</table>
<input type="hidden" name="grabar" value="si"/>
<input type="submit" value="Crear Usuario" />
</form>
<hr/>
<h1>Usuarios creados</h1>
<table>
    <tr>
        <td width="150px">Tipo documento</td>
        <td width="150px">Documento</td>
        <td width="150px">Nombre</td>
        <td width="150px">Apellido</td>
        <td width="150px">Activo</td>
    </tr>
<?php
$reg = $tra->consultarUsuarios();
echo sizeof($reg) . " registros";
for ($i = 0; $i < sizeof($reg); $i++) {
    ?>
    <tr>
        <td width="150px"><?php echo $reg[$i]["dstipo_documento"]; ?></td>
        <td width="150px"><?php echo $reg[$i]["cdusuario"]; ?></td>
        <td width="150px"><?php echo $reg[$i]["dsnombre"]; ?></td>
        <td width="150px"><?php echo $reg[$i]["dsapellido"]; ?></td>
        <td width="150px"><?php echo $reg[$i]["sactivo"]; ?></td>
    </tr>
    <?php
}
?>
</table>

</body>
</html>

```

## Visualización

Formulario completo, con los campos y la creación de la tabla que consultara los registros creados, modificados y eliminados.

Campos diligenciados con la opción crear en el campo tipo de acción, que hace referencia a crear un nuevo registro.

Los mensajes de aviso según la opción escogida.

Campo diligenciados con la opción modificar en el campo tipo de acción, que hace referencia a modificar un registro existente.

Campo diligenciados con la opción Eliminar en el campo tipo de acción, que hace referencia a eliminar un registro existente.

## Sección de imágenes

### GESTION DE USUARIOS

Tipo documento

Documento:

Nombre:

Apellido:

Clave:

Activo:

Tipo de acción:

### Usuarios creados

0 registros

Tipo documento	Documento	Nombre	Apellido	Activo
----------------	-----------	--------	----------	--------

### GESTION DE USUARIOS

Tipo documento

Documento:

Nombre:

Apellido:

Clave:

Activo:

Tipo de acción:

localhost dice:

Usuario creado!

Aceptar

## Usuarios creados

1 registros

Tipo documento	Documento	Nombre	Apellido	Activo
CC	1013616358	Diego	Panqueva	S

## GESTION DE USUARIOS

Tipo documento

Documento:

Nombre:

Apellido:

Clave:

Activo:

Tipo de acción:

Crear Usuario

## Usuarios creados

1 registros

Tipo documento	Documento	Nombre	Apellido	Activo
CC	1013616358	Diego Alejandro	Panqueva Benitez	S

## Usuarios creados

0 registros

Tipo documento	Documento	Nombre	Apellido	Activo
----------------	-----------	--------	----------	--------



Sección de texto	Sección de imágenes
<p><b>Referencias consultadas</b></p> <p>ITSON (2009) Inducción al proceso de diseño y desarrollo de objetos de aprendizaje. Recuperado de <a href="http://biblioteca.itson.mx/oa/formacion_profesores/inducccion_oa/index.html">http://biblioteca.itson.mx/oa/formacion_profesores/inducccion_oa/index.html</a></p> <p>Ministerio de Educación Nacional. República de Colombia (2009) ¿Qué es un objeto de aprendizaje? Recuperado de <a href="http://aprendeenlinea.udea.edu.co/lms/men/oac1.html">http://aprendeenlinea.udea.edu.co/lms/men/oac1.html</a></p> <p>Mager, R (2002) Pautas de Mager para el diseño de objetivos de aprendizaje. Recuperado de <a href="http://eduteka.icesi.edu.co/articulos/Tema15">http://eduteka.icesi.edu.co/articulos/Tema15</a></p> <p>Barreto, M. (2016) Procedimiento gestión de recursos educativos digitales. Bogotá, D.C. VIMEP – UNAD.</p>	

