

Lo primero que debemos tener en cuenta son las herramientas básicas necesarias para comenzar a desarrollar la aplicación en nuestro computador, entre ellas tenemos:

1. Servidor Web. (apache, xampp).
2. PHP.
3. Manejador de base de datos (mysql, postgres).

Estas herramientas se pueden instalar de manera individual y luego configurarla de tal modo que se interrelacionen entre sí. Aunque hoy en día tenemos disponible instaladores de paquetes que se encargan de realizar estos procedimientos así como la respectiva configuración de manera automática. Entre ellos tenemos:

Para Windows:

Xampp, Wampp, Ijj (Asp).

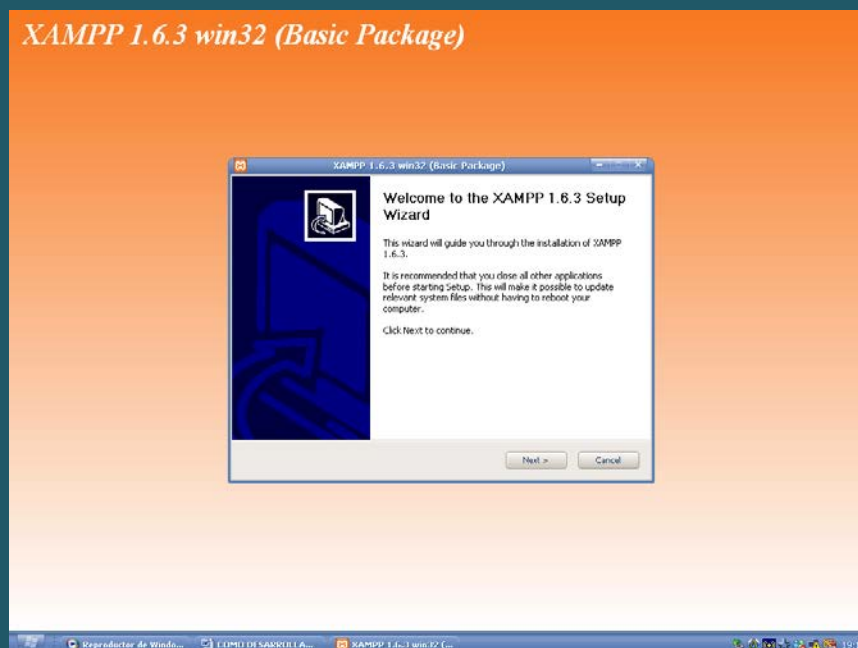
Para Linux.

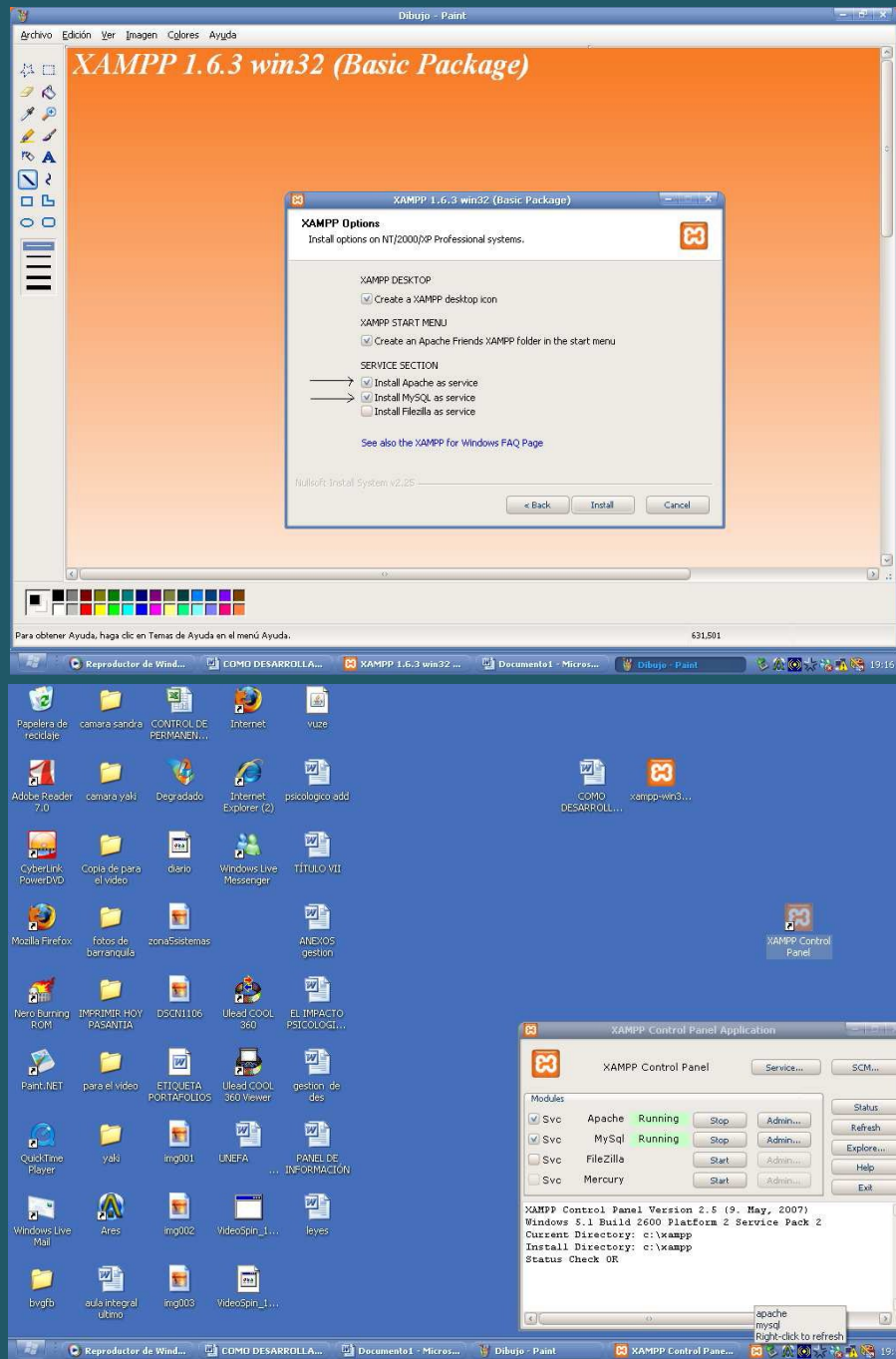
Lampp.

Los programas desarrollados en la plataforma Web se realizan con muchas paginas interrelacionadas a través de llamados al servidor. Ejemplo para un pequeño proceso:

UsuarioListar.PHP ➡ UsuarioAgregar.PHP ➡ UsuarioNuevoGuardar.PHP ➡ UsuarioListar.PHP

Vamos a partir con la instalación del Xampp en Windows:





Luego de haber instalado el Xampp. Podemos comprobar si el apache esta funcionando correctamente. Abriendo el navegador de su preferencia. Aunque es bueno recomendar a todos los usuarios el uso del Mozilla Firefox y aún más a los desarrolladores de software ya que este navegador posee características adicionales que podemos aprovechar en su debido momento.

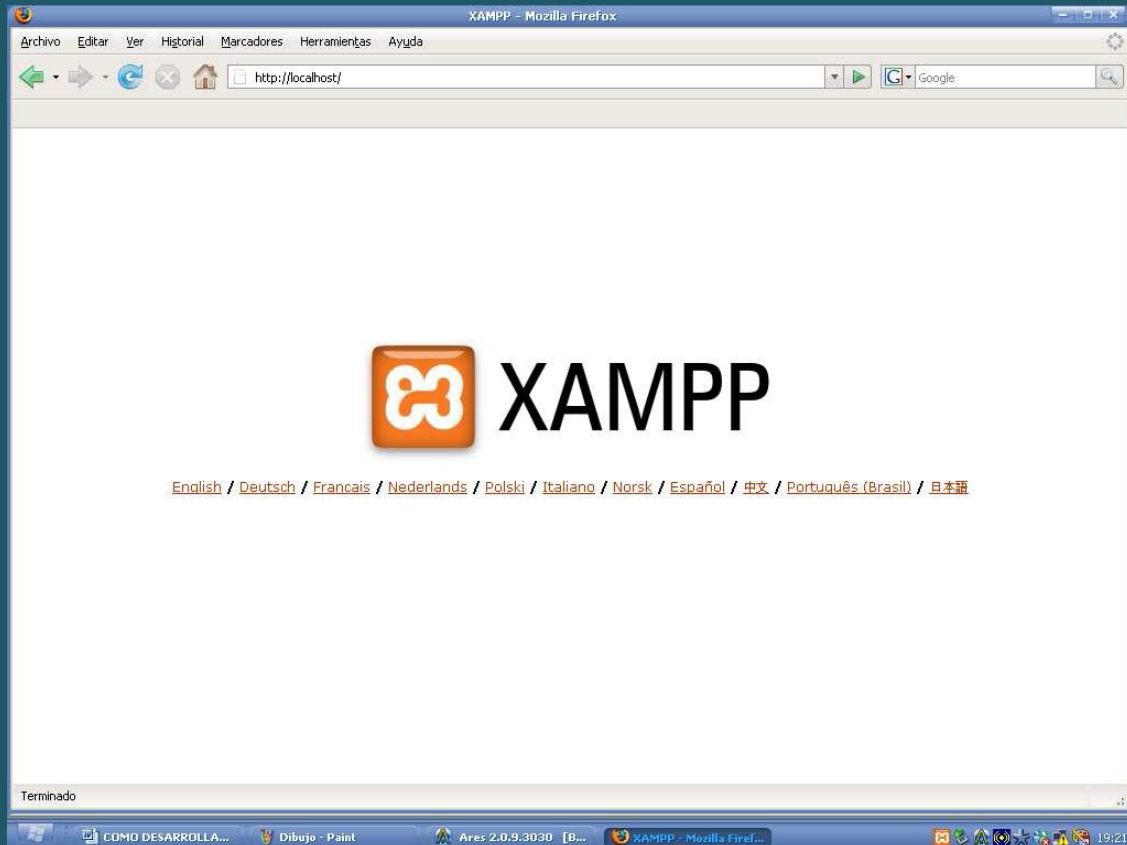
Procedemos a abrir el navegador (Mozilla firefox de ahora en adelante). Y en

la barra de navegación escribimos lo siguiente:

<http://localhost> o <http://172.64.54.7> (dirección IP) de la maquina en la cuál estoy instalando el apache.

En este caso las dos direcciones son las mismas ya que por ahora solo estamos trabajando de manera local.

En caso de tener instalado el apache y que el mismo se encuentre activado (running) debe aparecer la siguiente pantalla, que no es más que la bienvenida al software instalado:



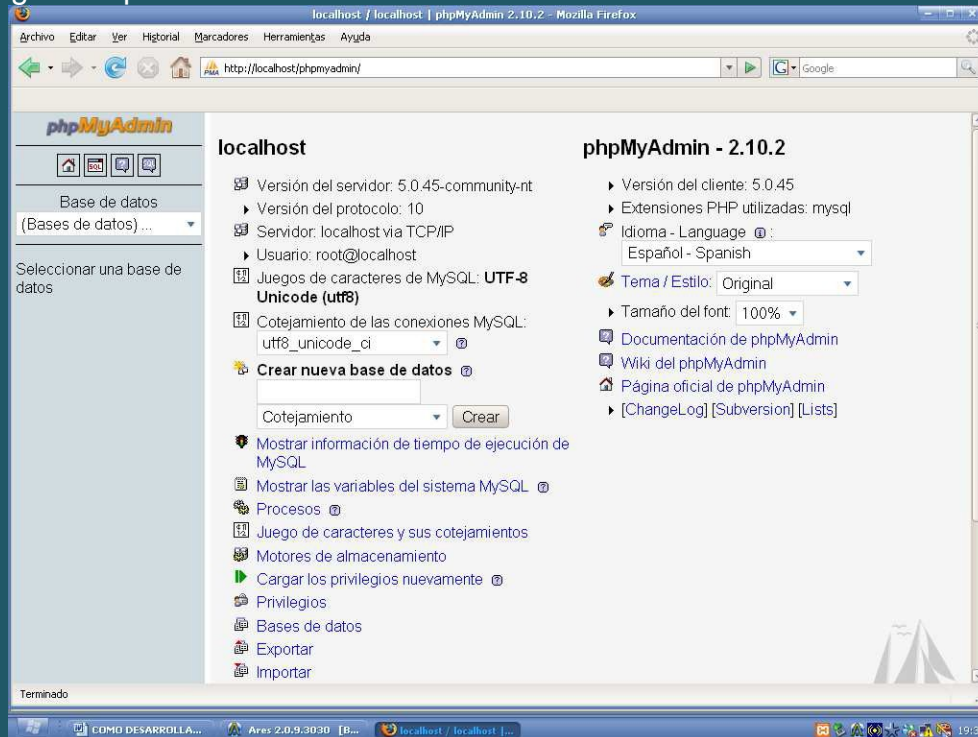
Al instalar el xampp se creara el directorio `c:/xampp/htdocs/` el cual es el directorio root donde deberan estar colocados las paginas web.

Muy bien. Ahora debemos diseñar la base de datos del sistema a desarrollar. Para esta guía rápida solo vamos a trabajar con 2 tablas que nos servirán para conocer el entorno básico de la programación basada y concebida en ambiente Web.

En este momento se debe probar si se instalo correctamente el mysql. A través de administrador de base de datos PHPmyadmin. En el url del navegador se debe ingresar la siguiente dirección

<http://localhost/PHPmyadmin>

En este caso debe aparecer la pantalla del PHPmyadmin tal como se observa en la siguiente pantalla:



Los siguientes tipos de campos son los más comunes:

1. Char. Textos de cantidad de caracteres fijos. Ejemplo: sexo (1) es f o m. nombre no funciona correctamente con este tipo ya que el nombre es variable. Ejemplo: Daniel abarca 5 caracteres pero ana tiene solo 3.
2. VarChar: este campo es idóneo para campos de cantidad de caracteres variables. Ejemplo Nombre, Apellido, Dirección. Tiene una limitante de 255 caracteres.
3. Texto: este campo abarca hasta más de 25 mil caracteres. Es usado solo si el campo será muy grande. Ejemplo Observaciones.
4. Int, TinyInt, BigInt. Este es un tipo de campo numérico.
5. Date: campo de fechas. Este campo siempre será guardado en formato inglés: año-mes-día.
6. Time: campo para la hora.

Para todos los campos existen otras características de definición. Tenemos:

- Cotejamiento: es el identificador de caracteres. Los más usados son latin1 y utf8.
- Atributos: unsigned.
- Nulo: establece si acepta o no un valor nulo en ese campo.

- Extra: permite determinar si el campo contendrá un valor auto numérico.
- Clave: el tipo de clave (primaria, foránea, única).

Estandarización de nomenclatura de programación método “zona5sistemas”.
(Nombre variables, campos, etc.).

Una nomenclatura estandarizada facilita la comprensión del código fuente del sistema, tanto para el desarrollador como a las diversas personas que necesiten interactuar con el código.

En la definición de la base de datos:		
Elemento	Estándar	Ejemplo
<i>base de datos</i>	BDNombreBaseDatos	BDBecas2008
<i>Tablas</i>	TNombreTabla	TDatosPersonales
<i>Campos</i>	NombreCampo	CedulaUsuario
<i>Llaves</i>	NombreTabla_TipoKey_NombreCampo (Primaria-Foránea-Única)	TUsuario_Pkey_Cedula TUsuario_Fkey_Cedula TUsuario_Ukey_Cedula

Definición del código fuente.		
Elemento	Estándar	Ejemplo
Nombre Páginas	NombreAccion.PHP	GuardarUsuario.PHP
Variable	\$NombreVariable	\$FechaEstimada
Nombre Input (Elementos de ingreso de datos del usuario al sistema):	NombreInput	CedulaFuncionario

Puntos a tomar en cuenta:

- Se debe evitar la minimización de palabras en la creación de variables o nombres de elementos en la programación. Todos los nombres deben ser escritos de forma completa.
- Los campos de las tablas deben llevar luego de su nombre la extensión de la tabla a la que pertenece. Ejemplo:
TEmpleado -> CedulaEmpleado -> NombreEmpleado.
- No se permiten ningún tipo de carácter como %, _, -, ni espacios en blanco en la construcción de variables.
- La primera letra de cada variable debe ser mayúscula y si esta compuesta por dos palabras juntas la segunda palabra debe empezar por mayúscula también.
- La base de datos debe comenzar con BD mayúscula seguida del nombre propio de la base de datos con la primera letra de la palabra en mayúscula.
- Todas las tablas de la base de datos debe contener la letra T en mayúscula seguida del nombre de la tabla con la primera letra en mayúscula igualmente.
- Todos los input del sistema debe tener por nombre el mismo que lo identifica con la base de datos en caso de ser relacionado, y si este no fuese el caso debe llevar las normas anteriores.

Ahora desarrollaremos una base de datos:

BDUniversidad.

TAlumno

CedulaAlumno (Int - 8 - unsigned - NotNull - Clave primaria)

PrimerNombreAlumno (VarChar - 15)

TelefonoAlumno (char 14) ejemplo "0412-900-53-13"

SexoAlumno (Char 1)

FechaNacimientoAlumno (Date)

ObservacionAlumno (Text)

TipoAlumno

TTipoAlumno

CodigoTipoAlumno (- 8 - unsigned - NotNull - Clave primaria - autonumerico)

DescripcionTipoAlumno (VarChar 20).

Ya que tenemos la base de datos lista, procedemos con la elaboración del sistema.

La primera página que desarrollaremos es la que permite la conexión de la página con la base de datos. La misma la llamaremos.

Conex.PHP

```
<?
```

```
if(!@$Link=mysql_connect("localhost","usuario","contraseña")){  
die("Error Al Tratar De Conectar");  
}
```

```
if(!@mysql_select_db("BDUniversidad")){  
die ("Error Al Tratar De Conectar Con La Base De Datos");  
}
```

```
}
```

```
?>
```

Elementos nuevos:

- If: Significa una condición. Si pasa algo ejecuta una instrucción.
- ! : Representa la negación.
- @ : evita un mensaje de error por defecto en caso de que alguna de las dos instrucciones no se cumplan.
- \$Link: al tener \$ significa que es una variable y la misma guardara la conexión establecida Con la base de datos.

- `mysql_connect`: instrucción propia del PHP que permite conectarse con el MySQL, necesita de 3 parámetros.
 - a. La ubicación de la base de datos. La misma puede ser localhost en caso de la aplicación resida en el mismo equipo que el sistema a desarrollar, o bien puede ser una dirección IP o una dirección Web.
 - b. El usuario: representa el login con el cual nos conectamos a MySQL.
 - c. La Clave: contraseña que utiliza el login con el que nos vamos a conectar.

- `Die`: es una instrucción que permite mostrar un mensaje y en caso de que se ejecute detiene la lectura del código por parte del navegador. Se usa principalmente para mostrar errores que si llegara a ocurrir sería innecesario continuar mostrando más inconvenientes.
- `mysql_select_db`: instrucción que permite seleccionar una base de datos de MySQL.
- Toda línea de instrucción escrita en PHP debe finalizar con “ ; ”
- Los If condicionales tiene la siguiente estructura:

```
If (comparación) {en caso de entrar en la condición} else {en caso contrario}
```

Página 2. index.PHP (Página solamente HTML)

```
<HTML>
```

```
<head>
```

```
<title></title>
```

```
</head>
```

```
<body>
```

```
<a href="AlumnoListar.PHP">ir a alumnos</a>
```

```
</body>
```

```
</HTML>
```

Elementos nuevos:

- index: esta página la llamaremos index para que el navegador por defecto la abra la primera vez que ingresamos. Si no existiera se mostraría el listado de páginas para que el usuario seleccione la que desee.
- Link.

```
<a href="pagina que voy a abrir en caso de accionar click">lo que ve el  
usuario en pantalla</a>
```

Página 3. el listado de alumnos registrados en la tabla TAlumno. Y la llamaremos AlumnoListar.PHP

```
<HTML>
```

```
<head>
```

```
<title></title>
```

```
</head>
```

```
<body>
```

```
<?
```

```
include("Conex.PHP");
$Sql="select * from TAlumno Where 1;

if(!@$Resultado=mysql_query($Sql,$Link)){
die("Error");
}

<table>
    <tr>
        <td> Modificar</td>
        <td> Eliminar</td>
        <td>Cédula</td>
        <td>Nombre</td>
        <td>Sexo</td>
    </tr>
<?
while($Row=mysql_fetch_array($Resultado)){
?>
<tr>
    <td ><a href="ModificarAlumno.PHP?CedulaAlumno=<?
echo($Row["CedulaAlumno"]);?>">Modificar</a> </td>
    <td ><a href="EliminarAlumno.PHP?CedulaAlumno=<?
echo($Row["CedulaAlumno"]);?>">Eliminar</a> </td>
    <td ><? echo($Row["CedulaAlumno"]);?></td>
    <td ><? echo($Row["PrimerNombreAlumno"]);?></td>
    <td ><? echo($Row["SexoAlumno"]);?></td>
</tr>
<? }?>
</table>
<div><a href="AgregarAlumno.PHP">Agregar un alumno nuevo</a></div>
</body>
</HTML>
<? mysql_close();?>
```

Elementos nuevos:

- Include: permite incluir el código de otra página externa a la que estamos creando. Esto permite escribir códigos comunes en todas las páginas una sola vez y solo hacer el llamado cuando se necesite, mejorando así su actualización y lectura.
- \$Sql: variable creada para la interacción con la base de datos.
- mysql_query(\$Sql,\$Link): esta función propia del PHP permite ejecutar una consulta SQL directamente con el MySQL. Recibe dos parámetros. El primero representa la SQL formulada anteriormente y el segundo parámetro es la variable de conexión con la base de datos.
- mysql_fetch_array: función del PHP que organiza de manera separada todos los registros encontrados luego de ejecutar la consulta pudiendo mostrar cada campo con su valor de manera individual.
- Echo: igual a la instrucción die con la diferencia que este tipo de mensaje no detiene la lectura del código.
- Echo (\$Row["CedulaAlumno"]); permite imprimir por pantalla el contenido de la cedula del alumno encontrada.
- Un Link puede ser muy útil en la creación de una pagina, ya que el permite o tiene la posibilidad de enviar variables de una pagina actual a la pagina solicitada. Se recuerda que una variable funciona en la pagina actual y al llamarse una página deja de existir. En el ejemplo tenemos lo siguiente:

```
<a href="PaginaSolicitada.PHP?Variable=Valor"></a>
```

Al recibir la variable en la página solicitada pudieras realizar el siguiente código.
Echo (\$_GET["Variable"]); y la misma imprimiría el valor de la variable de paso.

Pagina 4 AgregarAlumno.PHP

```
<HTML>
<head>
</head>
<body>
<form name="form1" method="post" action="GuardarAlumno.PHP">
<table>
  <tr>
    <td>Cedula</td>
    <td><input type="text" name="CedulaAlumno"></td>
  </tr>
  <tr>
    <td>Nombre</td>
    <td><input type="text" name="PrimerNombreAlumno"></td>
  </tr>
  <tr>
    <td><input type="submit" value="Guardar"></td>
  </tr>
</table>
</form>
</body>
</HTML>
```

Elementos nuevos:

- Se puede observar que es una página solo HTML y de igual manera tiene la extensión PHP.
- Se incluye el formulario. El cual lleva consigo un name identificador y un method que determinara la forma como pasa los datos ingresados en el formulario. Tenemos 2 tipos get o post.
- El método POST es el mas usado en el envío de datos ingresados en el formulario, lo hace de manera oculta hacia el servidor y el formulario que recibe obtendrá el arreglo print_r(\$_POST);

- El método GET es una forma alternativa de envío de datos donde el formulario que recibe obtendrá el arreglo `print_r($_GET)`; y el envío se ve en la barra de navegación del explorador. Este método es necesario ya que el envío hecho por los links siempre serán por este método.

Página 5.

GuardarAlumno.PHP

<?

```
Include("Conex.PHP")
```

```
:
```

```
$Sql="insert                into                TAlumno  
(CedulaAlumno,PrimerNombreAlumno)values($_POST["CedulaAlumno"],  
$_POST["PrimerNombreAlumno"]);
```

```
If($Resultado=mysql_query($Sql,$Link)){  
    Header("location:ListadoAlumno.PHP");
```

```
}else{
```

```
    Echo("Error al tratar de guardar");
```

```
}
```

?>

Elementos Nuevos:

- `mysql_query`: función propia del PHP que permite la ejecución de una consulta sql con el manejador de base de datos.
- `Header`: función del PHP que en el momento que se ejecuta envía una petición o llamado de una página al servidor. Se puede considerar como un link automático.

Página 6. EliminarAlumno.PHP

```
<?  
Include("Conex.PHP")  
:  
$Sql="delete from TAlumno where  
CedulaAlumno='".$_$_POST["CedulaAlumno"]."'";  
If($Resultado=mysql_query($Sql,$Link)){  
    Header("location:ListadoAlumno.PHP");  
}  
}else{  
    Echo("Error al eliminar el registro");  
}  
  
?>
```

Sentencias SQL.

Sintaxis básica:

```
Select NombreCampo1, NombreCampo2 from TAlumno Where Cedula=1
```

Interpretando su lectura puede decirse que significa: seleccióname el Campo1 y Campo2 proveniente de la Tabla Alumno donde la Cedula sea igual a uno.

Esta SQL permite realizar un listado de los registros de la tabla Alumnos.

Preguntas frecuentes:

- ¿Es necesario tener Internet para trabajar con PHP? No.
Por qué: PHP tiene la ventaja que puede ser utilizada en una maquina local, así como en una intranet. Lo que sí es indispensable es convertir la computadora en servidor mediante la instalación del apache.
- Es necesario tener instalado PHP en las maquinas clientes? No.
Por qué: el código PHP se ejecuta del lado del servidor, transformando el mismo en código HTML y este es enviado al cliente donde los navegadores puede interpretar ese código.

