

# PWD-Programación Web Dinámica



## PHP - HTML



**Universidad Nacional del Comahue**  
**Facultad de Informática**  
**Departamento de Programación**

# Interacción PHP – HTML

- PHP puede generar HTML

Ejemplos:

```
<html>
  <head>
    <title>Ejemplo de PHP</title>
  </head>
  <body>
    <?php
      $datos = "Hola Mundo";
      echo '<h2>Este es un ejemplo de generación de HTML </h2>';
      echo '<input type="text" value="" '.$datos.' "/> <br />';
    ?>
  </body>
</html>
```

# Interacción PHP – HTML

- HTML puede pasar información a PHP a través de formularios.
- Cuando se envía un formulario a un script PHP, las variables de dicho formulario pasan a estar automáticamente disponibles en el script destino.

# PHP – Saliendo de HTML

- Veamos un ejemplo para enviar datos ....

```
<form id="form1" name="form1" method="post" action="destino.php">
```

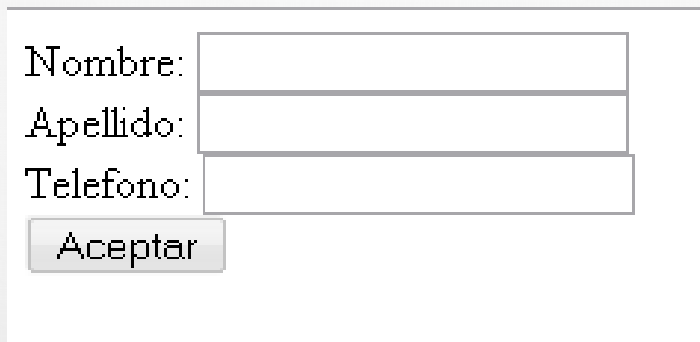
```
Nombre: <input name="nombre_form" type="text" id="nombre_form" /> <br />
```

```
Apellido: <input name="apellido_form" type="text" id="apellido_form" /> <br />
```

```
Telefono: <input name="tel_form" type="text" id="tel_form" /> <br />
```

```
<input type="submit" name="Submit" value="Aceptar" />
```

```
</form>
```



Nombre:

Apellido:

Telefono:



# PHP – Enviando datos al Servidor

- En la etiqueta FORM debemos fijarnos en dos atributos:
  - ACTION: la pagina a donde enviará el formulario (nuestra pagina en PHP)
  - METHOD: es la forma en la que enviará los valores. Pueden ser:
    - GET
    - POST

# PHP – Enviando datos al Servidor

- Get vs Post:
  - GET indica en la misma petición los datos que se van a enviar, o sea en la url.
  - Ejemplo: MiPagina.php?parametro=valor
  - POST inserta las variables después de realizar la petición.

# PHP – Enviando datos al Servidor

- Deberíamos usar **GET**:
  - Si no nos importa que los parámetros se vean en la URL.
  - Si el formulario HTML va a transferir variables de poco tamaño.
  - Si queremos que el usuario pueda cambiar el contenido de la web a través de los parámetros de la URL.

# PHP – Enviando datos al Servidor

- Deberíamos usar **POST**:
  - Si no queremos que las variables se muestren en la URL. Sobre todo si son datos delicados, como usuarios, contraseñas, etc.
  - Si el formulario transfiere mucha información, o variables que tengan un tamaño considerable.



# PHP – Recibiendo datos en el Servidor

- Para recibir la información enviada al servidor vamos a utilizar los arreglos de parámetros `$_GET` y `$_POST` dependiendo del método usado para enviar los parámetros
- Son asociativos y accedemos a cada parámetro por el atributo “name”

# PHP – Recibiendo datos en el Servidor

## *Ejemplo con POST:*

```
<h1>Procesa</h1>
```

```
<?php
```

```
    if ($_POST){
        $nombre = $_POST['nombre_form'] ;
        $apellido = $_POST['apellido_form'] ;
        $tel = $_POST['tel_form'];

        echo "<h3>Nombre: $nombre </h3><br />";
        echo "Apellido: $apellido <br />";
        echo "Tel&eacute;fono: $tel <br />";
    }else{
        echo "No se recibieron datos<br />";
    }
    ?>
```

# PHP – Recibiendo datos en el Servidor

## *Ejemplo con GET:*

```
<h1>Procesa</h1>
```

```
<?php
```

```
    if ($_GET){  
        $nombre = $_GET ['nombre_form'] ;  
        $apellido = $_GET ['apellido_form'] ;  
        $tel = $_GET ['tel_form'];  
  
        echo "<h3>Nombre: $nombre </h3><br />";  
        echo "Apellido: $apellido <br />";  
        echo "Teléfono: $tel <br />";  
    }else{  
        echo "No se recibieron datos<br />";  
    }  
}
```

```
?>
```

# Referencias

- [www.php.net/manual](http://www.php.net/manual)
- [http://www.w3schools.com/php/php\\_forms.asp](http://www.w3schools.com/php/php_forms.asp)