

# Bases de Datos

Tema 05. Prototipo de una aplicación de base de datos



**Marta Elena Zorrilla Pantaleón**

**Rafael Duque Medina**

DPTO. DE MATEMÁTICAS, ESTADÍSTICA Y  
COMPUTACIÓN

Este tema se publica bajo Licencia:

[Creative Commons BY-NC-SA 3.0](https://creativecommons.org/licenses/by-nc-sa/3.0/)

# Tabla de contenidos

---

- ▶ **Introducción**
  - ▶ **Base de datos utilizada**
  - ▶ **Estructura de la aplicación**
  - ▶ **Material complementario: formularios con PHP**
-

# Bibliografía

---

- ▶ Schafer, Steven M. HTML, XHTML y CSS. Anaya Multimedia, 2010.
- ▶ Cosentino, Christopher. Guía esencial de PHP. Prentice Hall, 2001.
- ▶ Charre Ojeda, Francisco. PHP 5. Guía práctica para usuarios. Anaya Multimedia, 2005.

# Introducción

---

- ▶ A continuación se presenta el prototipo de una aplicación que gestiona la información relativa a una empresa de envíos de pizzas a domicilio
- ▶ La aplicación está disponible mediante acceso web
- ▶ Las tecnologías que han sido utilizadas son las siguientes:
  - ▶ HTML/CSS: Lenguajes para mostrar, publicar y dar formato a la información en la web
  - ▶ PHP: Lenguaje para acceder y manipular la información de la base de datos
  - ▶ SQL Server 2008: Sistema Gestor de Base de Datos

# Base de datos

---

- ▶ La base de datos utilizada almacena toda la información relacionada con la pizzería (clientes, empleados, pedidos, etc.)
- ▶ La pizzería no ofrece pizzas establecidas previamente, como es lo habitual en la mayoría de estos negocios donde se ofertan pizzas cuatro quesos, margarita, etc.
- ▶ En esta pizzería los clientes deben registrarse en una aplicación web y tiene que configurar sus propias pizzas mediante la selección de los ingredientes y la base

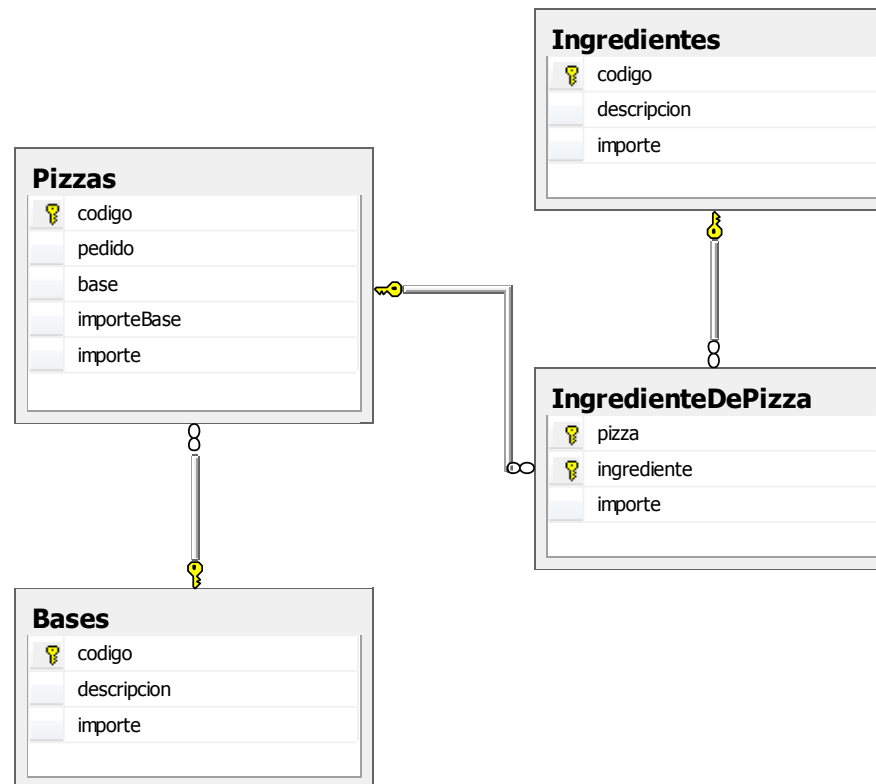
# Base de datos

---

- ▶ Los clientes pueden solicitar a la pizzería un pedido que debe constar de una o más pizzas
- ▶ Un mismo cliente puede realizar varios pedidos y en cada uno de ellos utilizar un medio de pago distinto (pago en efectivo, tarjeta de crédito, etc.)
- ▶ En todo momento es necesario conocer en qué estado se encuentra el pedido de un cliente (en cocina, el motorista lo está transportando, ya ha sido entregado, etc. )

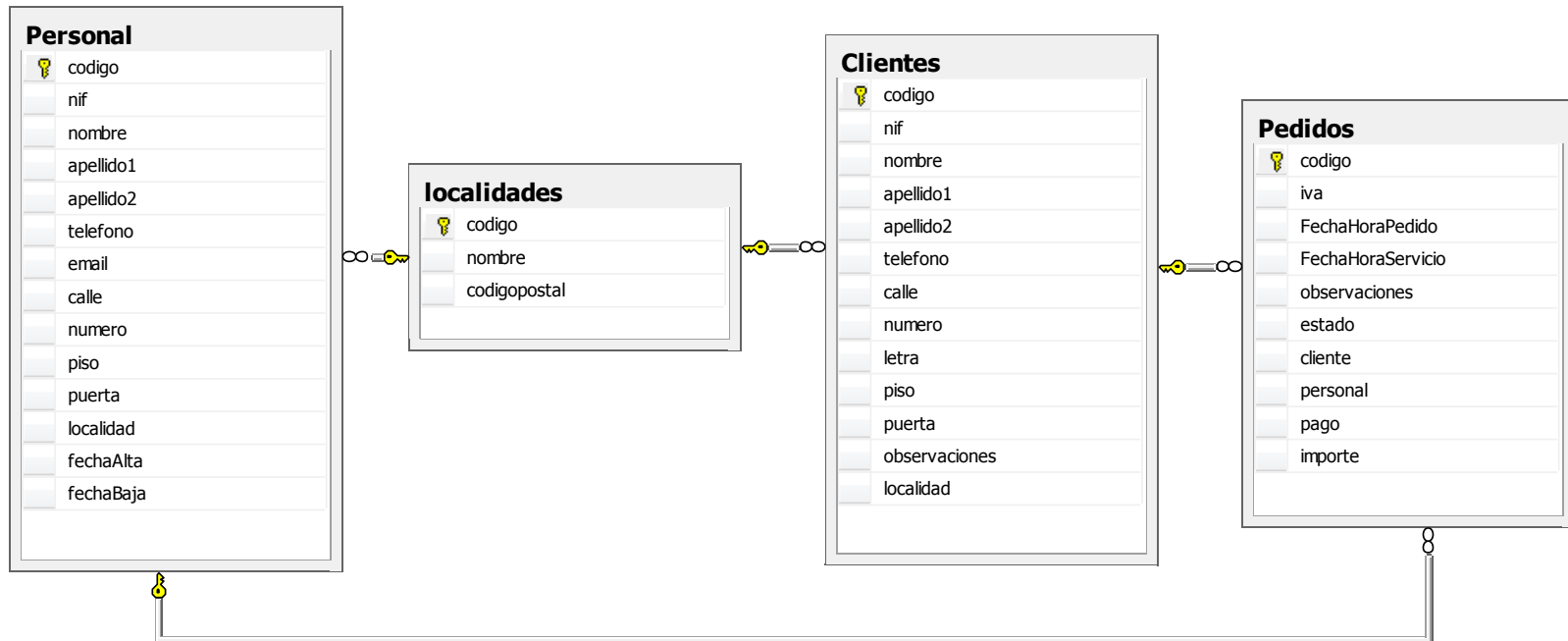
# Base de datos

- ▶ Un pizza podrá estar formada por varios ingredientes, pero tendrá una única base



# Base de datos

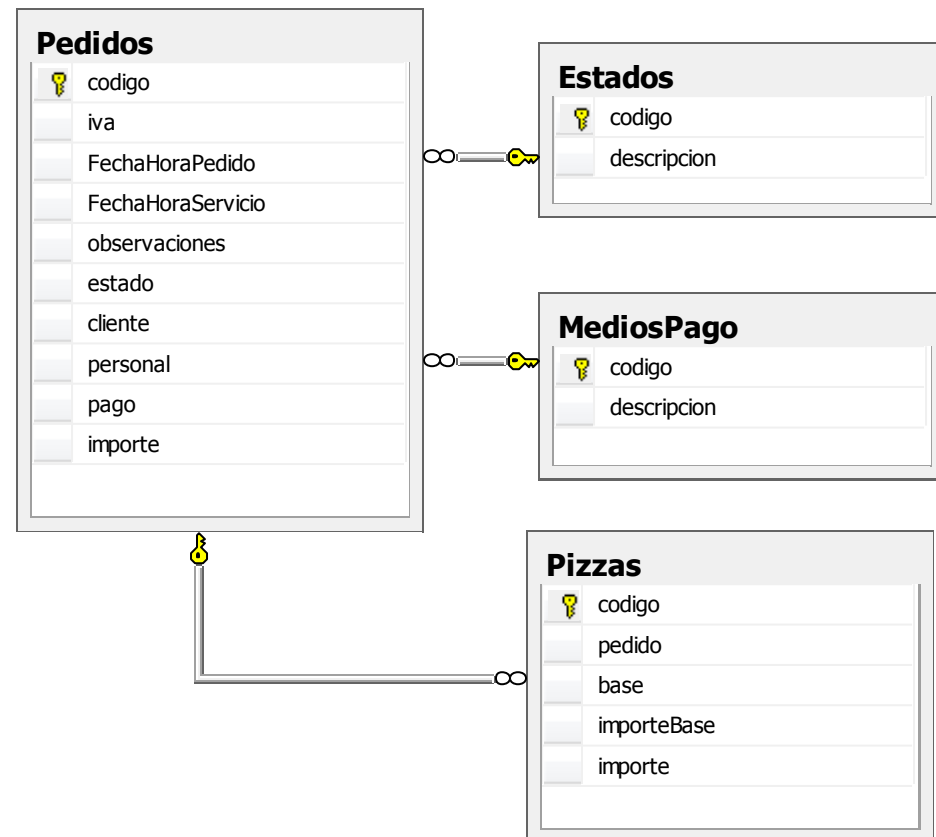
- ▶ La base de datos almacena la información de los pedidos
- ▶ Es necesario saber qué cliente solicita cada pedido y el trabajador (personal) de la empresa que gestiona el pedido





# Base de datos

- ▶ Un mismo pedido puede incluir más de una pizza
- ▶ De cada pedido es preciso conocer en qué estado se encuentra
- ▶ También es necesario conocer el medio de pago utilizado para abonar el importe del pedido



# Sistema web

---

- ▶ Se ha diseñado un sistema web que permite al usuario gestionar la información de todas las tablas de la base de datos
- ▶ Este sistema web está diseñado como una página de inicio que da acceso mediante hipervínculos a los formularios e informes programados
- ▶ Presenta dos tipos de formularios:
  - ▶ Maestros: formularios que permiten al usuario manipular la información de una única tabla de la base de datos. En nuestro caso sólo se ha programado la gestión de clientes
  - ▶ Maestro-detalle: formularios que permiten al usuario manipular la información de varias tablas

# Sistema web: Formulario Maestro

- ▶ Esta página permite gestionar la tabla cliente
- ▶ Inicialmente muestra la información de todos los clientes visualizados de 10 en 10. El funcionamiento de la misma se describe en la imagen

HOME BASES INGREDIENTES ESTADOS CLIENTES PAGO PERSONAL LOCALIDADES

Rellenando alguno de estos campos y pulsando en Buscar, filtramos los clientes por los parámetros que hemos especificado

Dar de alta un nuevo cliente

Filtrado por: DNI: Localidad: Cualquiera Buscar

Código	DNI	Nombre	1r Apellido	2º Apellido	Teléfono	Calle	Nº	Letra	Piso	Puerta	Observaciones	Localidad
1	07659222R	Jaime	Muñoz	Herrera	678342001	Avenida Mayor	17	j	6	a		REO
2	07612231F	Juan	Del Castillo	Gonzalez	611342521	Plaza Castro	21	i	7	a		REO
3	03459231O	Ramiro	Polo	Garcia	942735201	Calle Sardinero	231	h	2	b		PSM
4	07659221P	Gema	Delgado	Tirado	678342209	Calle Romanones	111	g	2	d		PSM
5	07655231I	Tomas	Fernandez	Cortes	600342521	Calle Castilla	31	f	5	b		HEL
6	12359231U	Jose	Gonzalez	Claro	678725021	Avenida Remos	9	e	4	a		HEL
7	08959231H	Carla	Rivera	Vera	658142521	Calle Goya	8	d	9	e		AST
8	07459231R	Davinia	Castilla	Aguirre	678312521	Calle Lierganes	1	c	2	d		AST
9	07652131D	Aurora	Molina	Moraga	678343821	Avenida Mercader	2	b	1	a		MAL
10	07659211A	Macario	Polaino	Romero	678342521	Calle Galdos	5	a	5	c		MAL

Mostrar anteriores

Mostrar siguientes

Este enlace permite dar de alta un nuevo cliente

Haciendo clic en el botón que muestra el código del cliente, accedemos a un formulario donde podemos manipular los datos de ese cliente

Con estos botones navegamos a través de todos los clientes

# Sistema web: Formulario Maestro ( y 2)

- ▶ Cuando se pulsa sobre el código del cliente (ver imagen previa) aparece este formulario que permite modificar la información del cliente seleccionado o borrar el registro si fuese necesario

HOME	BASES	INGREDIENTES	ESTADOS	CLIENTES	PAGO	PERSONAL	LOCALIDADES
Los campos con (*) son obligatorios.							
DNI(*):	<input type="text" value="07655231I"/>		1r Apellido(*):	<input type="text" value="Fernandez"/>		2º Apellido :	<input type="text" value="Cortes"/>
Nombre(*):	<input type="text" value="Tomas"/>		Calle(*):	<input type="text" value="Calle Castilla"/>		Número(*):	<input type="text" value="31"/>
Telefono(*):	<input type="text" value="600342521"/>		Piso(*):	<input type="text" value="5"/>		Puerta(*):	<input type="text" value="b"/>
Letra(*):	<input type="text" value="f"/>		Localidad (*):	<input type="text" value="Helguera (Reocin)"/>			
Observaciones :	<input type="text"/>						
			<input type="button" value="Modificar"/>	<input type="button" value="Eliminar registro"/>			

Botón para modificar los datos del cliente según lo que se observa en pantalla

Botón para eliminar el cliente mostrado en la pantalla

# Sistema web: Formulario maestro-detalle

- ▶ Este formulario permite modificar la información de tres tablas: pedidos, pizzas e ingredientes de pizza para que sea tratado todo como una unidad

En primer lugar identificamos al cliente

Damos de alta un nuevo pedido con una o varias pizzas y sus respectivos ingredientes

HOME BASES INGREDIENTES ESTADOS CLIENTES PAGO PERSONAL LOCALIDADES

Introduce los siguientes datos para dar de alta a un nuevo pedido:  
Los campos con (\*) son obligatorios.

**Datos del cliente:** DNI: (introducir los dígitos y la letra del dni sin espacios ni guiones)  Validar

Código	DNI	Nombre	1r Apellido	2º Apellido	Teléfono	Calle	Nº	Letra	Piso	Puerta	Observaciones	Localidad
1	07659222R	Jaime	Muñoz	Herrera	678342001	Avenida Mayor	17	j	6	a		REO

**Datos del pedido:** Código del personal (\*):  Tipo de pago (\*): American Express Observaciones:

**Datos de la pizza:** Base (\*): Fina

Ingredientes (\*):  
 Aceitunas  Anchoas  Cebolla  Champiñones  
 Chorizo  Jamon  Queso Picon

Finalizar Pedido Guardar y añadir Pizza

# Organización de los módulos programados

---

- ▶ La aplicación se ha estructurado en tres módulos:
  - ▶ Carpeta CSS: Incluye el archivo style.css con el formato que tendrá la interfaz de usuario (color de los textos, tipo de letra utilizado, etc.)
  - ▶ Carpeta principal: Incluye varios archivos:
    - ▶ El archivo header.php incluye el menú de la página web con los hipervínculos para navegar entre páginas
    - ▶ El archivo funciones.php incluye las funciones para acceder a la base de datos y manipular la información de todas las tablas (todas las funciones están comentadas para facilitar su uso)
    - ▶ El archivo generaFacturaPDF.php incluye una función para generar en fichero PDF la factura de un pedido
  - ▶ Carpeta tcpdf: Es una librería con soporte adicional para crear archivos pdf (es invocada por generaFacturaPDF.php)

# Instalación de la aplicación

---

- ▶ Para instalar y ejecutar correctamente la aplicación hay que seguir los siguientes tres pasos:
  1. Instalación del paquete XAMPP y SQL Server 2008 (ver las instrucciones en las diapositivas correspondientes)
  2. Descomprimir el contenido del archivo *pizzeria.zip* que contiene los archivos de la aplicación en la carpeta *htdocs* creada por el paquete XAMPP durante su instalación
  3. Ejecutar el script *pizzeria.sql* en SQL Server 2008 para crear la base de datos e insertar la información en las tablas
  4. Es necesario actualizar los scripts PHP modificando la función que gestiona la conexión a la base de datos ya que será necesario especificar los datos (servidor, login, password, etc.) que posibilitan la conexión
  5. Arrancar la aplicación abriendo un navegador y escribiendo <http://localhost/pizzeria/index.php>

# Instalación de la aplicación (y 2)

- ▶ La función *conectar* incluida en el archivo *funciones.php* gestiona la conexión a la base de datos
- ▶ En caso de utilizar la autenticación integrada de Windows únicamente hay que especificar el servidor donde se ubica el SGBD y la base de datos a la que queremos conectarnos

```
function conectar(){
    // variables a modificar: $serverName y $dbName
    $serverName = "localhost";
    $connectionOptions = array( "Database"=>"pizzeria");

    $conn = sqlsrv_connect($serverName, $connectionOptions);
    if($conn){
        // echo "Conexion establecida.<br>";
    }else{
        echo"La conexion no pudo ser establecida.";
        die(print_r(sqlsrv_errors(), true));
    }// if - else
    return $conn;
}// conectar
```

Servidor

Base de  
datos



# Instalación de la aplicación (y 3)

- ▶ En caso de utilizar la autenticación de SQL Server hay que especificar el servidor donde se ubica el SGBD, la base de datos a la que queremos conectarnos, el login y la password del usuario

```
function conectar(){
    // variables a modificar: $serverName y $dbName
    $serverName = "vm.ciencias.unican.es";
    $connectionOptions = array( "Database"=>"pizzeria", "UID"=>"javier", "PWD"=>"javipassword");

    $conn = sqlsrv_connect($serverName, $connectionOptions);
    if($conn){
        // echo "Conexion establecida.<br>";
    }else{
        echo"La conexion no pudo ser establecida.";
        die(print_r(sqlsrv_errors(), true));
    }// if - else
    return $conn;
}// conectar
```

Servidor

Base de datos, login y password

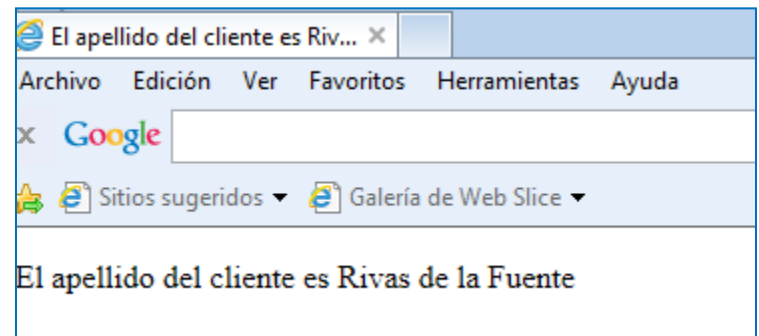
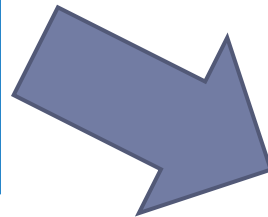
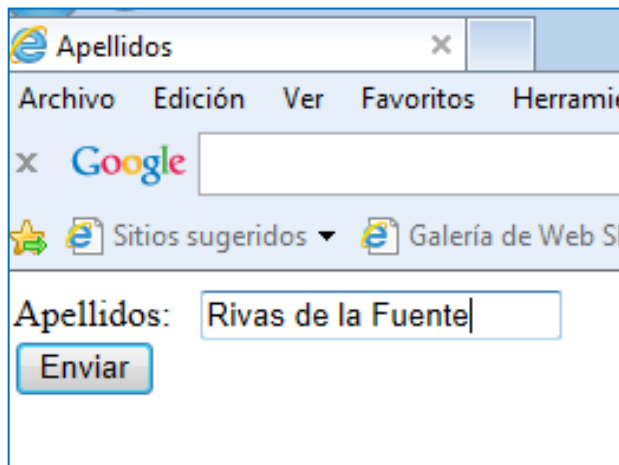
# Material complementario: formularios con PHP

---

- ▶ Esta aplicación sirve de modelo para crear otra aplicación web de estas características ya que ilustra cómo utilizar formularios, mostrar información y llevar a cabo el acceso y manipulación a información en una base de datos
- ▶ Sin embargo, HTML y PHP permiten utilizar formularios adicionales a los que se han usado en esta aplicación, los formularios más comunes se repasan a continuación

# Acceso a formularios con PHP

- ▶ Los **formularios web** permiten introducir información que será almacenada o buscada en la **base de datos**



# Acceso a formularios con PHP

---

## ▶ **¿Cómo implementar formularios con PHP?**

- ▶ Dentro de la etiqueta form de html que permite definir formularios hay que especificar dos atributos:
  - ▶ **Action:** Especifica el nombre del **script php** que gestionará la información del formulario
  - ▶ **Method:** Toma valor **post** para evitar que la información que se suministra mediante el formulario aparezca como parte de la URL
- ▶ Usamos el botón submit que permite al usuario notificar al servidor que ha finalizado de introducir la información en el formulario

### Archivo cliente.php

```
<html>
<body>
<form action="servidor.php" method="post">
  Nombre: <input type="text" size="20" name="nombre"/>
  <input type="submit" value="enviar"/>
</form>
</body>
</html>
```

# Acceso a formularios con PHP

---

## ▶ ¿Cómo implementar formularios?

- ▶ Dentro del script implementado accedemos y gestionamos la información introducida por el usuario
- ▶ En este ejemplo se visualiza en el navegador el valor introducido por el usuario

### Archivo servidor.php

```
<HTML>
<BODY>
<?PHP
  echo "su nombre es: ";
  echo $_POST["nombre"];
?>
</BODY>
</HTML>
```

**\$POST**: Colección  
de elementos  
enviados desde  
método post

# Formularios tipo TEXT

---

- ▶ Los **formularios** de tipo **text** permiten al cliente introducir **cadenas de texto**

.....

Introduzca su apellido:

```
<input type="text" name="apellido" size="20"/>
```

.....

....

```
<?PHP
```

```
echo "su apellido es: ";
```

```
echo $_POST["apellido"];
```

```
?>
```

...

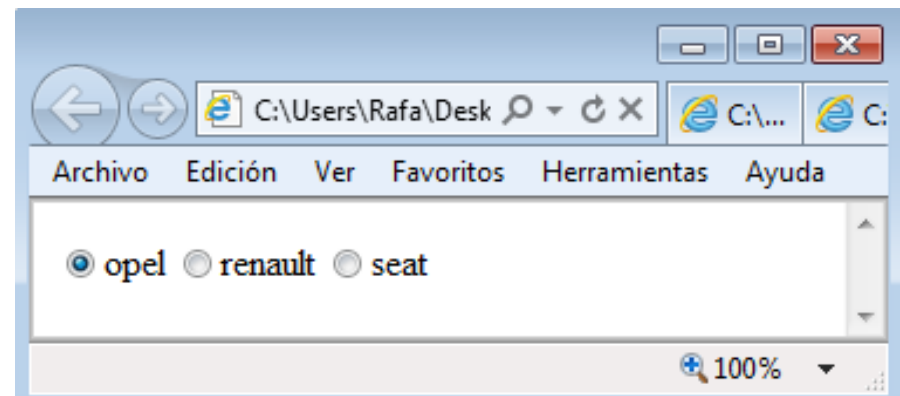
1r Apellido(*):	<input type="text" value="Polo"/>
Calle(*):	<input type="text" value="Calle Sardinero"/>
Piso(*):	<input type="text" value="2"/>

# Formularios tipo RADIO

- ▶ Los formularios de tipo **radio** permiten al cliente seleccionar una opción de entre varias

```
<INPUT TYPE="radio" NAME="coche" VALUE="opel">opel  
<INPUT TYPE="radio" NAME="coche" VALUE="renault">renault  
<INPUT TYPE="radio" NAME="coche" VALUE="seat">seat
```

```
<?PHP  
    echo "Su coche es:";  
    echo $_POST["coche"];  
?>
```

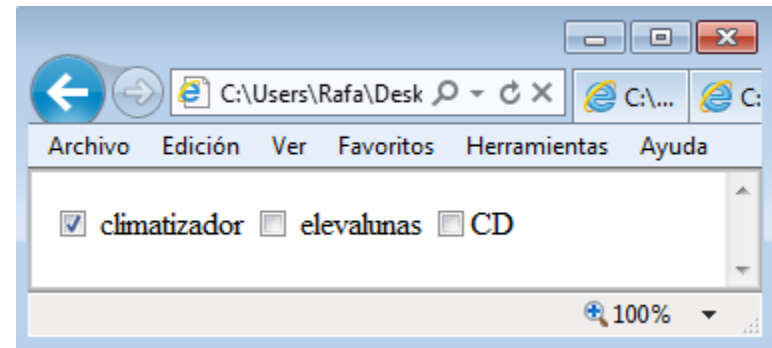


# Formularios tipo CHECKBOX

- ▶ Los formularios de tipo **checkbox** permiten al cliente seleccionar distintas opciones

```
<INPUT TYPE="checkbox" NAME="opciones[]" VALUE="climatizador" CHECKED>
    climatizador
<INPUT TYPE="checkbox" NAME="opciones[]" VALUE="elevelunas"> elevelunas
<INPUT TYPE="checkbox" NAME="opciones[]" VALUE="CD">CD
```

```
<?PHP
    $auxiliar=$_POST["opciones"];
    $n=count($auxiliar);
    echo "Estas son las opciones";
    echo "<br>";
    for ($i=0; $i<$n; $i++) {
        echo $auxiliar[$i];
        echo "<br>"; } ?>
```





# Formularios tipo PASSWORD

---

- ▶ El formulario **password** tiene un funcionamiento análogo al formulario **text**
- ▶ En este caso la información que introduce el usuario no se muestra en el navegador por motivos de seguridad

Introduzca la password de acceso al sistema:

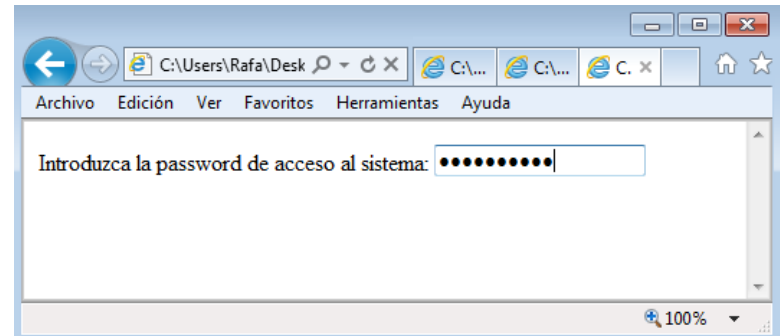
```
<INPUT TYPE="password" NAME="password" />
```

```
<?PHP
```

```
echo "la password introducida es: ";
```

```
echo $_POST["password"];
```

```
?>
```

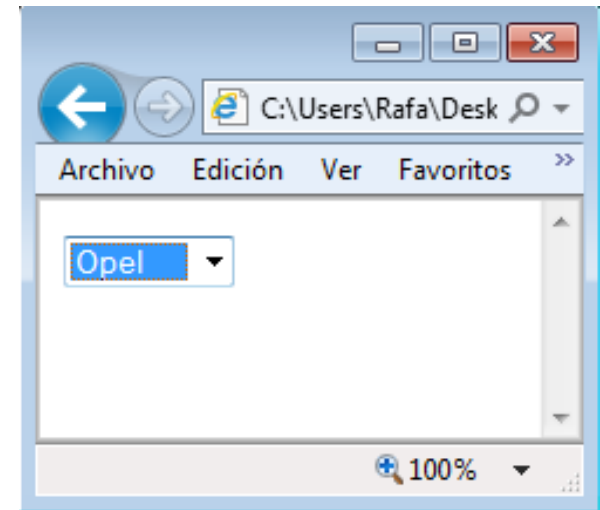


# Formularios tipo SELECT SIMPLE

- ▶ Este tipo de formularios sólo muestran un elemento al usuario que usa una lista desplegable para ver el resto

```
<SELECT NAME="coche">  
  <OPTION VALUE="Opel" >Opel  
  <OPTION VALUE="Volvo">Volvo  
  <OPTION VALUE="Renault">Renault  
</SELECT>
```

```
<?PHP  
  echo "Su coche es";  
  echo $_POST["coche"]; ?>
```

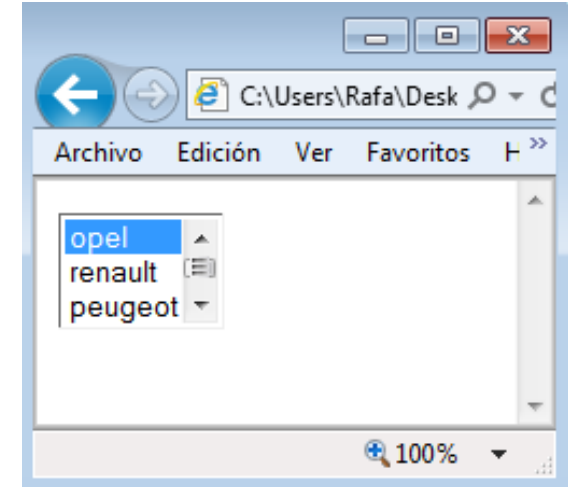


# Formularios y PHP: SELECT MULTIPLE

- ▶ Este tipo de formularios muestran un listado con varios elementos

```
<SELECT MULTIPLE SIZE="3" NAME="coches[]">  
  <OPTION VALUE="opel" SELECTED >opel  
  <OPTION VALUE="renault">renault  
  <OPTION VALUE="peugeot">peugeot  
  <OPTION VALUE="seat">seat  
</SELECT>
```

```
<?PHP  
  $coche=$_POST ["coches"];  
  $n=count($coche);  
  for ($i=0; $i<$n; $i++) {  
    echo $coche[$i];  
    echo "<br>"; } ?>
```



# Formularios tipo TEXTAREA

---

- ▶ El formulario **textarea** tiene un funcionamiento análogo al formulario **text**, pero con capacidad para varias líneas

```
<TEXTAREA COLS="15" ROWS="5" NAME="sugerencias">
Escriba aquí sus sugerencias
</TEXTAREA>
```

```
<?PHP
echo "la sugerencias son: ";
echo $_POST["sugerencias"];
?>
```

