```
        TEMAS PREFERIDOS:
        <input type="radio" name="aventuras" value="a"> Acción
        <input type="radio" name="historica" value="h"> Histórica
        <input type="radio" name="ficcion" value="fi"> Ficción <br>
        DESCUENTO:
        <input name ="descuento" value="15%" size="5" readonly> <br>
        SUBIR FOTOGRAFIA: <br>
        <input type="file" name="fichero"
               accept="application/x-zip-compressed"> <br>
        <input type="hidden" name="fecha" value=""> <br>
        <input type="submit" value="enviar">
        <input type="reset" value="reset">
        <input type="button" value="salir"
               onclick="window.close();">
</form>
```

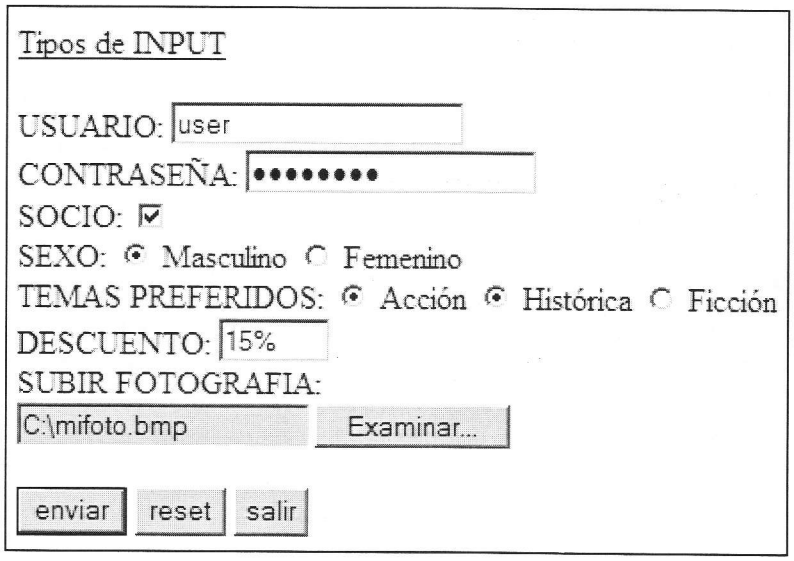

Figura 2.26: Controles de tipo <input>

El control de tipo text tiene el valor por defecto "user".

El control de tipo password sustituye los caracteres que introduce el usuario por símbolos para ocultarlos.

El control de tipo checkbox o casilla de verificación, está marcado por defecto mediante el atributo **checked**, pudiendo ser desmarcado por el usuario.

El primer conjunto botones tipo radio tienen el mismo **name ="genero"**, lo cual los hace mutuamente excluyentes, por tanto, el usuario solo puede marcar uno de ellos.

El segundo conjunto de botones tipo radio tienen diferentes nombres, por tanto, es posible seleccionar varios a la vez.

El control que contiene el descuento tiene el atributo **readonly** que impide al usuario modificar el valor.

El control de tipo file inserta un botón, para realizar una búsqueda en el disco del fichero que se enviará con el formulario. Hay que definir el atributo **enctype**.

El control de tipo hidden no se ve en el formulario, por tanto, no es percibido por el usuario, su finalidad es la de enviar ciertos datos al servidor que no son útiles para el cliente. Por ejemplo, cuando enviamos el formulario anterior podemos enviar la fecha y hora junto con el resto de información, sin que el usuario lo vea expresamente.

Para obtener al fecha y hora del sistema, sería preciso crear un script con lenguaje cliente (como Javascript) y modificar el atributo **value** del control hidden.

Por otro lado, el hecho de que el control sea oculto para el usuario no significa que se envíe de forma segura, todos los datos son enviados sin cifrar salvo que utilicemos un protocolo seguro como HTTPS.

El control de tipo submit envía los datos del formulario a la URI especificada en el atributo **action**.

El control de tipo reset recupera los valores por defecto de todos los controles.

El control de tipo button define un botón genérico, para asociarle una determinada acción con un script, por ejemplo, cerrar la ventana del formulario.

Actividad 2.22:

Crea la página manejador.html con el texto "PEDIDO RECIBIDO" (simulando la respuesta del servidor). Copia el código anterior en una página nueva llamada form.html y comprueba cómo funcionan los controles.

<button>

Definición: sirve para insertar un botón como los creados por **<input>** pero con más posibilidades de formato (por ejemplo, incluir imágenes dentro del botón).

Aparición: etiqueta inicial y final obligatorias.

Atributos: **%attrs, type, name, value, disabled, tabindex, acceskey, onfocus, onblur.**

- **type**: indica el tipo de botón (submit | reset | button).

Ejemplo:

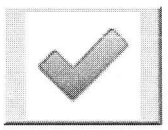

Fig. 2.27: Botón

```
<button type="submit">
  <img src="imagenes/enviar.jpg">
</img></button>
```

Actividad 2.23:

Inserta un botón de enviar con una imagen sonriente en form.html.

<select>

Definición: sirve para insertar un menú de formulario con varias opciones.

Aparición: etiquetas inicial y final obligatorias.

Atributos: `%attrs, name, size, multiple, disabled, tabindex, onfocus, onblur, onchange`

- `size`: especifica el nº de filas del menú que se ven al mismo tiempo.
- `multiple`: este booleano permite una selección múltiple de opciones.

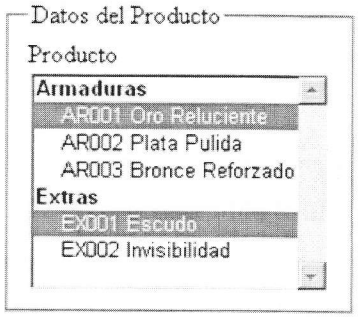

Figura 2.28: Control <select>

<option>

Definición: sirve para indicar cada opción de un menú de formulario.

Aparición: etiqueta final opcional.

Atributos: `%attrs, selected, disabled, label, value.`

- `selected`: este booleano indica que la opcion está preseleccionada.
- `label`: especifica un rótulo para la opción de menú.

<optgroup>

Definición: sirve para definir grupos de opciones.

Aparición: etiquetas inicial y final obligatorias.

Atributos: `%attrs, disabled, label.`

<textarea>

Definición: sirve para insertar un cuadro de entrada de texto con muchas líneas.

Aparición: etiquetas inicial y final obligatorias.

Atributos: **`%attrs, name, rows, cols, disabled, readonly, tabindex, accesskey, onfocus, onblur, onselect, onchange`**.

- **`rows`**: indica el nº de líneas de texto visibles.
- **`cols`**: indica el ancho del cuadro en caracteres.

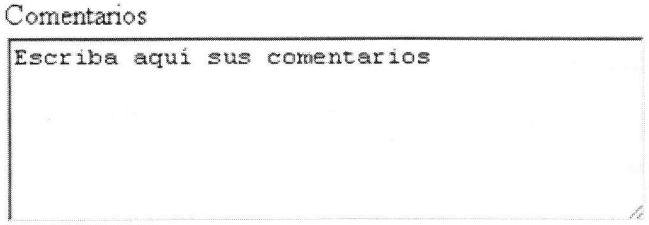

Figura 2.29: Control <textarea>

<label>

Definición: sirve para asignar un título a un control.

Aparición: etiquetas inicial y final obligatorias.

Atributos: **`%attrs, for, accesskey, onfocus, onblur`**.

- **`for`**: especifica el nombre de otro control para el que servirá de rótulo.
  (Coincide con el atributo **`id`** del control relacionado).

<fieldset>

Definición: sirve para agrupar los controles y sus títulos por temas.

Aparición: etiquetas inicial y final obligatorias.

Atributos: **`%attrs.`**

<legend>

Definición: sirve para asignar un rótulo a un fieldset.

Aparición: etiquetas inicial y final obligatorias.

Atributos: **`%attrs, accesskey`**.

Ejemplo conjunto: pedido.html

```
<body>
 <h1>INDUSTRIAS STARFUERTE</h1>
 <table width="1000"><tr><td>
 <p>
 <form action="manejador.html" method="post">
  <fieldset>
    <legend>FORMULARIO DE PEDIDO</legend>
    <table width=700 cellpadding=5><tr>
    <tr><td><strong> > Paso 1. Identificación </strong></td></tr>
    <td><fieldset>
         <legend>Datos del Cliente</legend><br>
         <label for="email">Correo Electrónico</label>
         <input type="text" id="email" size=20 maxlength=40>
         <br><label for="pass">Contraseña</label>
         <input type="password" id="pass" size=10 maxlength=30>
         <br>(Longitud mínima 6 caracteres)
        </fieldset>
    </td>
    <td><fieldset>
         <legend>Tipo de Cliente</legend><br>
         <input type="radio" id="particular" name="tipo" checked>
         <label for ="particular">Particular</label><br>
         <input type="radio" id="asociacion" name="tipo">
         <label for ="asociacion">Asociación</label><br>
         <input type="radio" id="empresa" name="tipo">
         <label for ="empresa">Empresa</label>
        </fieldset>
    </td></table>

    <table width=980 cellpadding=5>
    <tr><td><strong> > Paso 2. Pedido <br> </strong></td></tr>
    <tr>
    <td>
        <fieldset>
        <legend>Datos del Producto</legend>
         Producto<br>
         <select id="producto" name="producto" multiple="si" size=8>
          <optgroup label="Armaduras">
          <option value="AR001" selected="selected">
           AR001 Oro Reluciente
          </option>
          <option value="AR002">
           AR002 Plata Pulida
          </option>
          <option value="AR003">
           AR003 Bronce Reforzado
          </option>
          </optgroup>
         <optgroup label="Extras">
          <option value="EX001" selected="selected">
           EX001 Escudo
```

```
            </option>
            <option value="EX002">EX002 Invisibilidad<option>
           </optgroup>
          </select>
          </fieldset>
      </td>
       <td><fieldset>
          <legend>Información de interés</legend>
          Comentarios<br>
          <textarea rows=6 cols=60>Escriba aquí sus comentarios
       </textarea>
          <br><input type="checkbox" id="publi" value="si" checked>
          <label for "publi">Acepto recibir información publicitaria
          </label>
          </fieldset>
       </td>
      </tr>
      </table>

      <table width=980 cellpadding=5>
      <tr>
       <td><fieldset>
            <legend>Datos de Envío</legend><br>
            <label for "dire">Dirección</label>
            <input type="text" id="dire" size=40>
            <br><label for "ciudad">Ciudad</label>
            <input type="text" id="ciudad" size=20>
            <label for "cp">Codigo Postal</label>
            <input type="text" id="cp" size=5>
           </fieldset>
       </td>
       <td><fieldset>
            <legend>Datos Bancarios</legend><br>
            <label for "entidad">Nombre de la Entidad</label>
            <input type="text" id="entidad" size=20><br>
            <label for "cuenta">N° de cuenta</label>
            <input type="text" id="entidad" size=20>
           </fieldset>
       </td>
      </tr>
      <tr>
       <td><button type="submit">ENVIAR<br>PEDIDO</button></td>
       <td><button type="reset">RESET</button></td>
      </tr>
      </table>

    </fieldset>
    </p>
   </form>
   </table>
  </body>
```

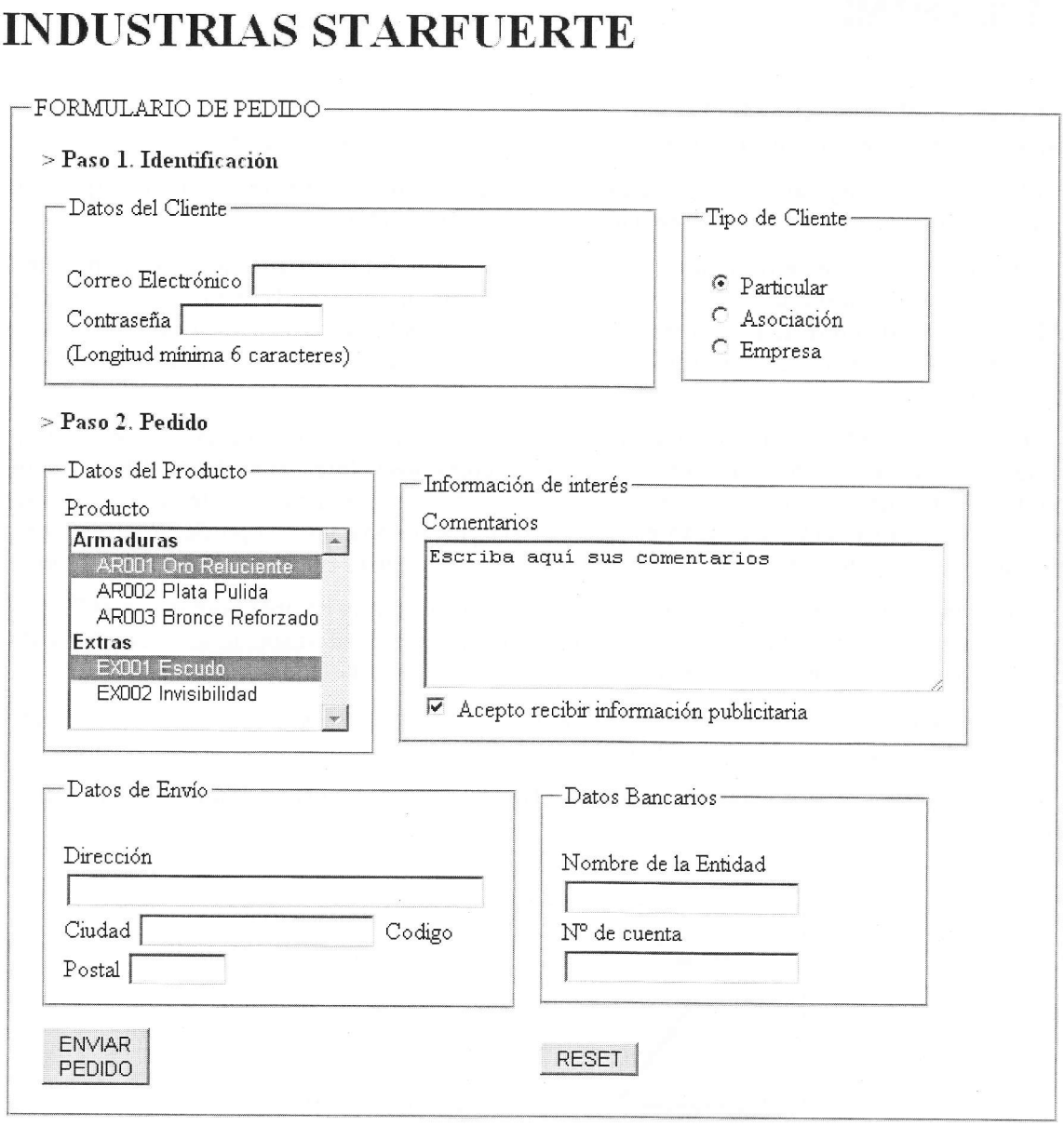

Figura 2.30: Formulario completo "pedido.html"

Mediante `<fieldset>` podemos agrupar los controles por grupos, lo que facilita al usuario la tarea de rellenar el formulario.

2.6. XHTML

Como ya sabemos XHTML 1.0 vio la luz por primera vez de forma oficial en 1998 y se convirtió en recomendación para construir páginas web en enero del año 2000.

El objetivo era mejorar las deficiencias que tenía el HTML de aquel momento, incorporando las ventajas que aportaba un nuevo lenguaje recién creado, el XML.

XML es un lenguaje derivado de SGML como todos, pero mucho más sencillo, cuya principal finalidad es la de transmitir datos con una cierta estructura a través de la Web. Los documentos XML no aportan información sobre como van a ser tratados esos datos, ni cómo se van a presentar y esta separación constituye, en sí misma, una gran ventaja para su desarrollo y mantenimiento.

Por otro lado, XML es un lenguaje más estricto en su sintaxis, ya que exige por ejemplo, escribir todos los elementos y sus atributos en minúsculas, los valores de atributos deben ir entre comillas, todo elemento debe tener su marca de cierre, el anidamiento de elementos debe ser correcto sin solapamiento, etc. Esto es una ventaja cuando hay un equipo de desarrolladores trabajando en el mismo sitio Web, así mismo permite que el parser (intérprete) sea más simple.

Por último indicar, que también era imprescindible mantener la compatibilidad con HTML 4, porque estaba muy extendido y los navegadores que no se habían adaptado al nuevo XHTML debían ser capaces de interpretar estos documentos como si se tratase de documentos HTML.

En resumen, XHTML es un lenguaje formado por los mismos elementos que HTML pero que se ajusta a las normas sintácticas de XML.

(Por esta razón, se dice habitualmente que XHTML es una reconstrucción de HTML usando XML).

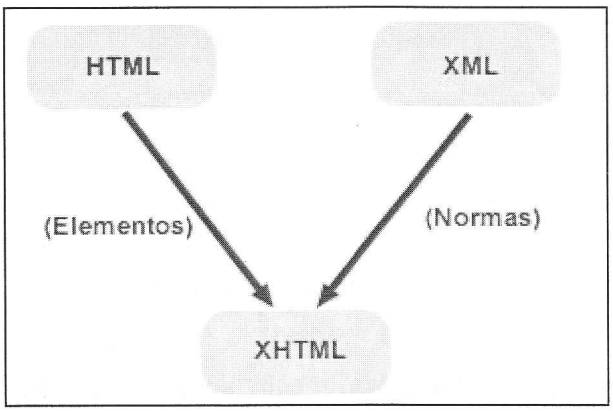

Figura 2.31: Origen del lenguaje XHTML

Otra ventaja de XHTML es la posibilidad de incorporar elementos procedentes de otros espacios de nombres como MathML y SVG. Esto permitirá construir documentos con contenido científico o gráficos vectoriales, de una forma mucho más simple y eficaz.

2.6.1. Estructura y elementos del documento

La estructura es exactamente la misma que la de un documento HTML, por tanto, tiene el conocido aspecto:

```
<! Declaración de DTD >
<html>
  <head>
    Contenido de cabecera
  </head>
  <body>
    Contenido del cuerpo
  </body>
</html
```

La principal diferencia está en la declaración de DTD y en el elemento `<html>`.

Para empezar la declaración de DTD en XHTML es obligatoria mientras que en HTML no lo era, pero seguimos teniendo tres definiciones equivalentes a las de HTML:

Strict, Transitional y Frameset.

DTD Strict
· Si no vamos a utilizar elementos de estilo (considerados obsoletos) dentro del documento, podemos usar la declaración estricta: `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">`
DTD Transitional
· Si queremos permitir el uso de elementos de estilo, utilizaremos la declaración de transición: `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">`
DTD Frameset
· Si queremos usar marcos en el documento, utilizaremos la declaración de frameset: `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">`

En cuanto al elemento `<html>` es también obligatorio y tiene dos atributos nuevos: `xmlns` y `xml:lang`.

· `xmlns="http://www.w3.org/1999/xhtml"` se trata del espacio de nombres, que se tratará más adelante en la unidad correspondiente a XML.

El valor que toma este atributo en este contexto es la URI fija indicada.

· `xml:lang="es"` que indica el idioma al estilo xml.

Ejemplo:

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
```

Además el elemento `<html>` debe llevar obligatoriamente los elementos `<head>`, `<body>` y a su vez `<head>` debe tener un `<title>`.

2.6.2. Normas en XHTML

Para que un documento XHTML esté bien formado debe cumplir una serie de normas sintácticas heredades de XML:

- Todas las etiquetas y todos los atributos debe ir en minúsculas.

Incorrecto	Correcto

`<Head>`
`  <TITLE>Bienvenidos</TITLE>`
`<Head>`
	`<head>`
	`  <title> Bienvenidos </title>`
	`</head>`

- Todos los elementos deben cerrarse, incluso los elementos vacíos.
  (Con éstos se puede utilizar la forma abreviada de cierre, que consiste en añadir la barra "/" al final, en lugar de poner la etiqueta de cierre)

Incorrecto	Correcto
` ` `<img src="grafico.gif"` `  alt="gráfico">`	` ` `<img src="grafico.gif"` `  alt="gráfico" />`