# Práctica de Formularios en HTML5

## HTML

Crea una web que contenga un formulario con los siguientes campos:

* **Nombre** (longitud máxima de 20 caracteres)
* **Primer apellido** (longitud máxima de 20 caracteres)
* **Segundo apellido** (longitud máxima de 20 caracteres)
* **Fecha de nacimiento**
* **Sexo**
  + **Hombre**
  + **Mujer**
* **Imagen de perfil** (únicamente acepta ficheros que son imágenes)
* **Teléfono móvil** (El valor sugerido es +34555334455)
* **Teléfono fijo** (El valor sugerido es +34955553344)
* **Correo electrónico** (El valor sugerido es asd@asdf.es)
* **Contraseña**
* **Checkbox que acepta las condiciones de uso.**
* **Checkbox que acepta la recepción de las novedades de la web.**
* **Imagen Anterior** (id=btnAnterior) Utiliza la etiqueta <img>
* **Imagen Siguiente**(id=btnSiguiente) Utiliza la etiqueta <img>
* **Botón submit “Enviar”.**

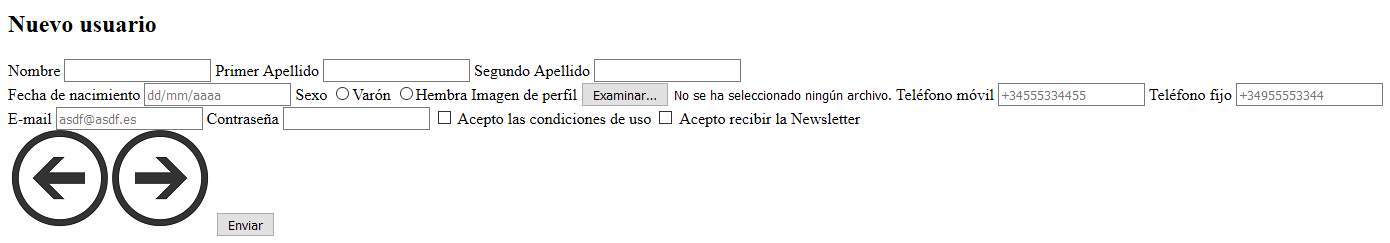
Los campos en rojo es obligatorio que sean rellenados.

El contenido del campo teléfono móvil debe respetar la siguiente expresión regular: ^(\(?\+34\)?\s?|0034\s?)[67](?:[\.\-\s]?\d\d){4}$|^[67](?:[\.\-\s]?\d\d){4}$

El contenido del campo teléfono fijo debe respetar la siguiente expresión regular: ^(\(?\+34\)?\s?|0034\s?)9(?:[\.\-\s]?\d\d){4}$|^9(?:[\.\-\s]?\d\d){4}$

Cuando los teléfonos introducidos no se correspondan con las expresiones regulares aparecerá el siguiente mensaje: “Ha de tener 9 dígitos, se aceptan los prefijos de España (+34, 0034)”

Formulario sin estilos:



## Estilos

Vamos a darle un estilo moderno al formulario. Las tendencias actuales de diseño evitan formularios largos con muchos campos en una única pantalla. La alternativa es intentar hacer formularios más responsables en las que tengamos varias pantallas a través de las cuales el usuario pueda ir navegando y rellenando los datos del formulario.

A continuación tienes las capturas de pantalla para ver cómo están organizados los controles del formulario. Fíjate que en todas ellas se ven unos controles de formulario fijos: los botones de Anterior y Siguiente y el botón Enviar.

Los estilos a aplicar en el formulario son:

La cabecera h2 “Nuevo usuario” tendrá la siguiente fuente: Palatino, 'Palatino Linotype', 'Palatino LT STD', 'Book Antiqua', Georgia, serif;

Las label tendrán a Helvetica como fuente.

Todos los input, sean del tipo que sean, tendrán un margen inferior de 10px, un borde con radio de 5px, y el estilo de borde será: outset. La fuente para los input será la misma que para h2: Palatino, 'Palatino Linotype', 'Palatino LT STD', 'Book Antiqua', Georgia, serif;

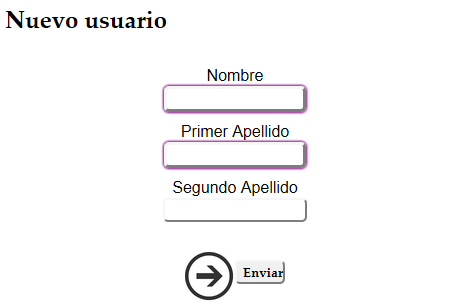
Cuando un input tenga un valor válido, el texto del input aparecerá en negrita.

Cuando un input tenga un valor inválido, el borde tendrá una anchura de 3px, y la sombra de caja tendrá una altura y anchura de 0px, el radio de la sombra es de 1.5px y el radio de propagación de 1px. El color de la sombra de caja será: purple.

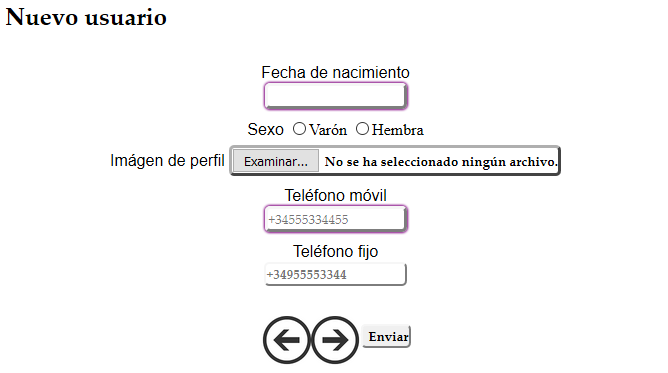
Cuando se realice el paso de una fase a otra se tiene que ver un efecto de animación/transición en el que la fase que desaparece y la fase que aparece lo hace con una duración de 1 segundo y utilizando la función ease. NOTA: Utiliza la propiedad opacity de CSS3.

En esta práctica vamos a realizar 3 fases o etapas de visualización del formulario:

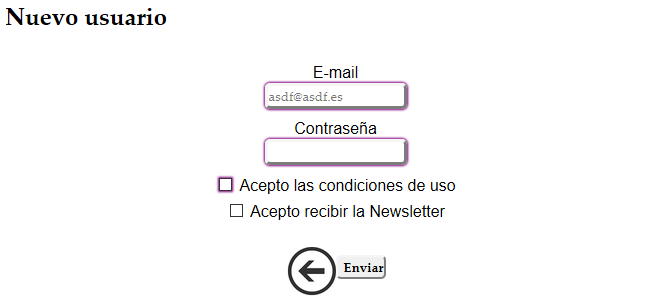
Primera etapa:



Segunda etapa:



Tercera etapa:



## Javascript

La funcionalidad que vamos a darle al formulario consiste en ir pasando de fase en fase al pulsar en los botones Anterior y Siguiente.

Para ello vamos a crear un “namespace” que se va a llamar miForm. Dentro de este “namespace” vamos a definir 3 propiedades y 3 métodos:

**Propiedades**

* primerafase: Identifica la primera fase (1) de las distintas que tiene el formulario.
* ultimafase: Identifica la última fase (3) de las distintas que tiene el formulario.
* faseActual: Va a mantener la fase actual que está viendo el usuario en cada momento.

**Métodos**

* verFaseAnterior(): Permite ocultar la fase que se está viendo actualmente y visualizar la fase anterior a la que estábamos. Ha de controlar que se muestren únicamente los botones Anterior/Siguiente que se puedan utilizar.
* verFaseSiguiente():Permite ocultar la fase que se está viendo actualmente y visualizar la fase siguiente a la que estábamos. Ha de controlar que se muestren únicamente los botones Anterior/Siguiente que se puedan utilizar.
* asignarEventos(): Asigna el evento click a los botones Anterior y Siguiente.