



**IPAP**

INSTITUTO PROVINCIAL DE  
ADMINISTRACIÓN PÚBLICA  
DEL CHACO

# NAVEGACIÓN ACCESIBLE



**Pauta 2.4 Navegable:** Proporcionar medios para ayudar a los usuarios a navegar, encontrar contenido y determinar dónde se encuentran.

Para lograr una navegación accesible, los criterios de conformidad de esta pauta hacen referencia a:

- Identificar el contenido.
- Describir adecuadamente el propósito de los enlaces.
- Saltar contenido repetido.
- Orden adecuado del foco y visibilidad del mismo.
- Información sobre la localización del sitio.



**IPAP**

INSTITUTO PROVINCIAL DE  
ADMINISTRACIÓN PÚBLICA  
DEL CHACO

# Título de la página



**Criterio de Conformidad 2.4.2 Titulado de Páginas:** Las páginas web tienen títulos que describen su temática o propósito. Este criterio está clasificado como nivel A.

Todos los documentos HTML deben tener un elemento TITLE en la sección HEAD.

```
<!DOCTYPE html>  
<html lang="es">  
  
  <head>  
    <meta charset="UTF-8">  
    <title>Título de mi página</title>  
  </head>  
  
  <body>  
  </body>  
  
</html>
```



accesibilidad web



Todo

Imágenes

Vídeos

Noticias

Maps

Más ▾

Herramientas de búsqueda

Cerca de 3.200.000 resultados (0,35 segundos)

## accesibilidad web

### Definiciones web

La Organización Mundial de la Salud recoge en sus informes un total de 600 millones de personas con discapacidad. El acceso de estas personas a la tecnología debe tenerse en cuenta en la construcción de una sociedad igualitaria. ...

[http://es.wikipedia.org/wiki/Accesibilidad\\_web](http://es.wikipedia.org/wiki/Accesibilidad_web)

*Comentarios*

### Accesibilidad Web: ¿Qué es?

[dlsi.ua.es/accesibilidadweb](http://dlsi.ua.es/accesibilidadweb) ▾

Todo tipo de información sobre **accesibilidad** en la **Web**: errores de **accesibilidad**, ejemplos de páginas inaccesibles, noticias, software, hardware, ayudas ...

[Las pautas de accesibilidad al ...](#) · [Introducción](#) · [Definición](#) · [WCAG 2.0](#)

Has visitado esta página 2 veces. Fecha de la última visita: 09/08/16.

### Introducción a la Accesibilidad Web - W3C

[www.w3c.es/Traducciones/es/WAI/intro/accessibility](http://www.w3c.es/Traducciones/es/WAI/intro/accessibility) ▾

La **accesibilidad Web** significa que personas con algún tipo de discapacidad van a poder hacer uso de la Web. En concreto, al hablar de **accesibilidad Web** se ...

[¿Qué es la Accesibilidad Web?](#) · [¿Por qué la Accesibilidad Web ...](#)

### Guía Breve de Accesibilidad Web - W3C



**IPAP**  
INSTITUTO PROVINCIAL DE  
ADMINISTRACIÓN PÚBLICA  
DEL CHACO

# Enlaces



## Atributo aria-label y aria-labelledby de WAI-ARIA

### aria-label

```
<a href="enlaces.html" aria-label="Leer más sobre enlaces  
accesibles"> Leer más </a>
```



# IPAP

INSTITUTO PROVINCIAL DE  
ADMINISTRACIÓN PÚBLICA  
DEL CHACO

“aria-label” sustituye al texto del enlace mientras que *title* ofrece información adicional que es leída junto con el texto del enlace.

Incluir el atributo *aria-label* con una descripción adecuada nos permite cumplir con el **Criterio de Conformidad 2.4.9** Propósito de los enlaces solo con el texto del enlace.

Este Criterio de Conformidad esta clasificado con un nivel AAA.

Técnica de referencia de las WCAG 2.0: [ARIA8: Using aria-label for link purpose](#)





## aria-labelledby

El atributo “aria- labelledby” es similar a “aria-label”, pero en vez de incluir directamente en él el texto descriptivo del enlace, indicamos el id o ids del elemento o elementos de la página que funcionan como descripción del enlace.

Hay que tener cuidado con su uso. aria-label pueden ser desatendida si se usa con aria-labelledby y anula otras formas nativas de etiquetado como ALT en las imágenes o <label> en los campos de formulario.



```
<p>Descargar <span id="documento">Archivo de ventas 2015 </span>:  
  <a aria-labelledby="documento pdf" href="[url]" id="pdf">PDF</a> |  
  <a aria-labelledby="documento doc" href="[url]" id="doc">Word</a> |  
  <a aria-labelledby="documento ppt" href="[url]" id="ppt">PPT</a>  
</p>
```

Técnica de referencia de las WCAG 2.0: [ARIA7: Using aria-labelledby for link purpose](#)



## Navegación a Saltos

Cuando navegamos en un sitio web y vamos saltando de una página a otra es muy normal que encontremos unas partes comunes, como la cabecera y el menú principal, que se repiten de una página a otra.

Para una persona que usa un lector de pantallas, la consulta de una página web es lineal, de arriba hacia abajo y supone pasar por todo el contenido de la página web.



¿Cómo se puede evitar este problema para mejorar la navegación?

Las Pautas de Accesibilidad al Contenido en la Web 2.0, proporcionan la solución en el criterio de conformidad 2.4.1 Evitar bloques, que dice: Existe un mecanismo para evitar los bloques de contenido que se repiten en múltiples páginas web. Este criterio está clasificado como nivel A.

Vamos a ver un ejemplo real de aplicación de este criterio.



La página del [Web Accessibility Initiative \(WAI\)](#) presenta un enlace con el texto “Skip to Content” , “saltar al contenido”, al principio de la página.

Si revisamos el código fuente, vemos que el enlace conduce a un destino en la propia página etiquetado con “skip” en el atributo href.

Este destino está situado un poco más abajo en el contenido principal de la página que comienza con el encabezado h1 “Web Accessibility Initiative”, es decir, este punto de la página.



**IPAP**

INSTITUTO PROVINCIAL DE  
ADMINISTRACIÓN PÚBLICA  
DEL CHACO

Navegación por teclado.  
Problemas con el foco.



La pauta 2.1 Accesible por teclado contiene Criterios de Conformidad que hacen referencia a la operabilidad de las páginas mediante el teclado.

**2.1.1 Teclado:** Toda la funcionalidad del contenido es operable a través de una interfaz de teclado sin que se requiera una determinada velocidad para cada pulsación individual de las teclas, excepto cuando la función interna requiere de una entrada que depende del trayecto de los movimientos del usuario y no sólo de los puntos inicial y final. (Nivel A)

**2.1.3 Teclado (sin excepciones):** Toda la funcionalidad del contenido se puede operar a través de una interfaz de teclado sin requerir una determinada velocidad en la pulsación de las teclas. (Nivel AAA)



## Problemas habituales con el foco:

- El foco no está visible;
- El orden del foco no tiene sentido;
- Las páginas tienen trampas para el foco;
- Recibir el foco provoca automáticamente un cambio de contexto sin al menos avisar previamente al usuario.





## El foco no está visible

Cuando un enlace o un campo de un formulario obtiene el foco, éste presenta un contorno punteado o con algún color que lo haga notar, o bien, en el caso de un textarea, por ejemplo, muestra el cursor de texto parpadeante en su interior.

Estos recursos nos permiten reconocer visualmente dónde se encuentra el foco en todo momento. Sin embargo, muchos sitios anulan este efecto por razones meramente estéticas, dificultando mucho la navegación por teclado.



## El orden del foco no tiene sentido

Como dice el [criterio de conformidad 2.4.3](#) de nivel A , los componentes que pueden recibir el foco lo debe hacer en un orden que preserve su significado y operabilidad.

Para conseguirlo hay que organizar el contenido en el código HTML de forma que el orden por defecto sea el correcto.



## Trampas para el foco

El tercer problema relacionado con el foco es que hay páginas en las cuales nos encontramos con trampas para el foco, zonas de la página a las que podemos llegar mediante el teclado, como una ventana modal, una película Flash o los controles de reproducción de un vídeo, pero de las cuales solo podemos salir, para regresar a otra parte de la página, mediante el ratón.

Criterio de conformidad de referencia de las WCAG 2.0:  
[2.1.2 Sin trampas para el foco del teclado \(Nivel A\)](#)



## Recibir el foco provoca un cambio de contexto

El cuarto problema es que haya componentes de la página que cuando reciben el foco provoquen automáticamente un cambio de contexto sin advertir al menos previamente al usuario.

Criterios de conformidad de referencia de las WCAG 2.0:

- [3.2.1 Al recibir el foco \(Nivel A\)](#)
- [3.2.2 Al recibir entradas \(Nivel A\)](#)
- [3.2.5 Cambios a petición \(Nivel AAA\)](#)



# Localización de contenido dentro del sitio



## Múltiples vías

**Criterio de Conformidad 2.4.5 Múltiples vías:** se proporciona más de un camino para localizar una página web dentro de un conjunto de páginas web, excepto cuando la página es el resultado, o un paso intermedio, de un proceso. (Nivel AA)

Por lo tanto, las WCAG 2.0 indican que debemos aplicar dos o más técnicas para ofrecer múltiples vías:

- Mapa del sitio.
- Función de búsqueda.
- Enlaces a páginas relacionadas.
- Tabla de contenidos.
- Otras técnicas para sitios pequeños.



## Mapa de sitio

Un mapa del sitio facilita la comprensión del contenido y la organización del sitio web, y es una alternativa a la navegación mediante menús y barras de navegación.

Técnica de referencia de las WCAG 2.0: [“G63. Providing a site map”](#)

### **¿Qué requisitos debe cumplir un mapa de sitio?**

- Debe estar enlazado desde todas las páginas.
- Desde el mapa web se tiene que poder acceder a todas las páginas del sitio.



- Sus elementos deben ser, por lo tanto, enlaces.
- La presentación y la estructura del mapa debe reflejar la organización del sitio, por ello se debe maquetar con los elementos HTML adecuados para reflejar dicha organización, como son los encabezados y las listas.
- Se debe revisar que el orden de lectura y de tabulación por los elementos del mapa sea correcto.
- Debe actualizarse cada vez que se actualiza el sitio.





# IPAP

INSTITUTO PROVINCIAL DE  
ADMINISTRACIÓN PÚBLICA  
DEL CHACO

## ...: MAPA DE NAVEGACION ...:





## Función de búsqueda

Otra técnica es proporcionar una función de búsqueda en la página. De esta manera, los usuarios pueden buscar el contenido sin necesidad de comprender la estructura del sitio o navegar por el mismo. Es especialmente útil en sitios de gran tamaño.

Técnica de referencia de las WCAG 2.0\_

[“G161: Providing a searchfunction to help users find content”](#)



# IPAP

INSTITUTO PROVINCIAL DE  
ADMINISTRACIÓN PÚBLICA  
DEL CHACO

## Enlaces a páginas relacionadas

Otra técnica es proporcionar enlaces a páginas relacionadas. De esta manera los usuarios pueden seguir estos enlaces para encontrar información adicional.

Técnica de las WCAG 2.0 de referencia:

[“G125: Providing links to navigate to related Web pages”](#)



## Tabla de contenidos

Una tabla de contenidos es un índice con las secciones y subsecciones de una página (o conjunto de páginas) que proporciona un acceso directo a una sección específica.

No hay que confundir una tabla de contenidos con un menú de navegación o un mapa de sitio.

Técnica de referencia de las WCAG 2.0: ["G64: Providing a Table of Contents"](#)



## Otras técnicas para sitios pequeños

Y por último, si el sitio es pequeño, se pueden aplicar otras Técnicas:

- el menú de navegación incluye el acceso a todas las páginas del sitio;
- todas las páginas están enlazadas desde la página de inicio.

Técnicas de referencia de las WCAG 2.0:

- [G126: Providing a list of links to all other Web pages](#)
- [G185: Linking to all of the pages on the site from the home page](#)



## Ubicación del Usuario

Un requisito de accesibilidad que hará que la navegación por tu sitio sea más accesible es facilitar la ubicación del usuario, proporcionarle información sobre dónde se encuentra dentro de un conjunto de páginas y que, de este modo, pueda contestar en todo momento a la pregunta "¿Dónde estoy?".

Existen diferentes técnicas para lograrlo:



# IPAP

INSTITUTO PROVINCIAL DE  
ADMINISTRACIÓN PÚBLICA  
DEL CHACO

## Migas de pan

Las migas de pan ayudan a:

- comprender cómo se estructura el contenido de un sitio
- navegar a páginas visitadas anteriormente
- identificar la ubicación actual de una página dentro de un conjunto de páginas o dentro del sitio

Técnica de referencia de las WCAG 2.0: ["G65: Providing a breadcrumb trail"](#)



## Requisitos que deben cumplir las migas de pan

- Deben estar situadas siempre en el mismo lugar en todas las páginas.
- Hay que maquetarlas mediante listas, aunque visualmente se vean en forma lineal.
- Se deben emplear separadores visuales entre las mismas.
- Deben ser enlaces, salvo la ubicación actual.
- Deben indicar el camino usado para llegar a la página actual.

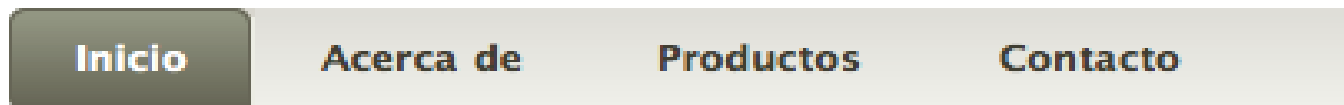
Inicio > página1 > página2 > página actual





## Indicar la situación actual en la navegación

Otra técnica para facilitar la ubicación del usuario es indicar la situación actual en menús, barras de navegación, pestañas o pasos en un proceso.



Técnica de referencia de las WCAG 2.0:

[“G128: Indicating current location within navigation bars”](#)



# IPAP

INSTITUTO PROVINCIAL DE  
ADMINISTRACIÓN PÚBLICA  
DEL CHACO

Home

Portfolio

Services

About Us

Contact



## Blue is a wonderful Colour

This is how an overlaid tabbed navigation menu looks like. This type of tabbed menus increase the visual appeal of any tabbed menu layouts. Its not that hard to code a menu like this. This is how an overlaid tabbed navigation menu looks like. This type of tabbed menus increase the visual appeal of any tabbed menu layouts. Its not that hard to code a menu like this. This is how an overlaid tabbed navigation menu looks like.

[Go to Tutorial](#)



## Consistencia

Por último, otro aspecto que hace más accesible la navegación es la consistencia, tanto en la presentación y maquetación de los mecanismos de navegación, como en la identificación de los componentes que tienen la misma funcionalidad dentro de un conjunto de páginas

3.2.3 - Navegación coherente [Nivel AA]

3.2.4 - Identificación coherente [Nivel AA]



## Navegación coherente

Si los mecanismos de navegación que se repiten en diferentes páginas de un conjunto (como el buscador, el menú de navegación, las migas de pan, el enlace saltar al contenido, etc.) se presentan en el mismo orden en las diferentes páginas del conjunto, los usuarios pueden predecir dónde estarán en cada página y localizarlos con más facilidad.

Técnica de referencia de las WCAG 2.0: ["G61: Presenting repeated components in the same relative order each time they appear"](#)



## Identificación coherente

Identificación coherente quiere decir, en los enlaces, que aquellos que tienen un mismo destino tienen el mismo texto de enlace.

También se aplica al texto alternativo de los elementos no textuales.

Como así también es aplicable a las etiquetas de los campos de formulario: los mismos controles de formulario deben estar etiquetados igual.



**IPAP**

INSTITUTO PROVINCIAL DE  
ADMINISTRACIÓN PÚBLICA  
DEL CHACO

# JavaScript y Accesibilidad



# IPAP

INSTITUTO PROVINCIAL DE  
ADMINISTRACIÓN PÚBLICA  
DEL CHACO

Javascript es una tecnología compatible con la accesibilidad y, por tanto, compatible con los productos de apoyo.

Para poder usar JavaScript en nuestras páginas de forma accesible hay que seguir una serie de pautas.

- Independiente del dispositivo.
- Implementar javascript de forma no intrusiva, separando por tanto el contenido, la presentación y el comportamiento de la página.
- mejora progresiva



## Independiente del dispositivo

Para que los scripts y elementos de programación puedan ser operados desde cualquier dispositivo, es necesario usar correctamente los manejadores de eventos javascript.

Los manejadores de eventos nos permiten ejecutar unas instrucciones cuando un evento es disparado, por ejemplo, ejecutar una validación del formulario antes de que se envíe dicho formulario:

```
<form onsubmit="validarForm()" id="miForm" ...>
```

O con javascript no intrusivo:

```
document.getElementById('miForm').onsubmit =validarForm
```





## Manejadores Lógicos

No dependen de dispositivos concretos y reconocen eventos tanto del ratón como del teclado:

- onsubmit
- onreset
- onfocus
- onblur
- onload
- onselect



## Manejadores específicos del mouse

Estos manejadores de eventos tienen su equivalente para teclado:

- onmousedown = onkeydown
- onmouseup = onkeyup
- onmouseover = onfocus
- onmouseout = onblur

Se pueden ver varios ejemplos de uso de estos manejadores en las técnicas [SCR2](#) y [SCR20](#) de las WCAG2.0



```
<a href="menu.php"
```

```
onmouseover="swapImageOn('menu')"  
onfocus="swapImageOn('menu')"
```

```
onmouseout="swapImageOff('menu_off')"  
onblur="swapImageOff('menu_off')">
```

```

```

```
</a>
```



## Evento “onclick”

Puede parecer un manejador de evento de ratón, pero es el evento de acción por defecto de enlaces y botones y se activa por teclado y por ratón.

Cuando usamos “onclick” con otros elementos que no sean enlaces y botones ( `<div>` , `<img>` , `<td>` , etc.), para que el código asociado al evento pueda ser invocado también por teclado, entonces deberá utilizarse junto a su evento equivalente de teclado “onkeypress”.



## Atributo tabindex:

- **tabindex="0"** para que se pueda acceder a dicho elemento por el teclado, mediante el tabulador;
- **tabindex="-1"** cuando se quiere que el elemento pueda obtener el foco mediante JavaScript con `.focus()`.
- **tabindex="x"** donde x es un numero entero. Define un orden de tabulación explícito: 1,2,3,etc.

```

```



## JavaScript no intrusivo

Javascript no intrusivo nos va a permitir llevar a cabo la separación del comportamiento (de la programación de la página) de lo que es el contenido y la presentación.

Ejemplo de JS intrusivo:

```
<form onsubmit="validarForm()" id="miForm" ...>
```



La forma no intrusiva y correcta es separar el comportamiento incluyendo toda la programación en ficheros JS.

*En el fichero JS*

```
document.getElementById('miForm').onsubmit =validarForm
```

*En la página HTML*

```
<form id="miForm" ...>
```



## Mejora Progresiva

La “mejora progresiva” se basa en el principio básico de diseño y programación que venimos viendo, la separación del contenido, la presentación y el comportamiento, que implementamos con javascript no intrusivo.

¿Cómo se hace?

Ejemplo

Para ejemplificar cómo se hace vamos a seguir el ejemplo disponible en la técnica [SCR24](#)





Esta sería la manera incorrecta de hacerlo:

```
<a href="javascript:;" onclick="window.open('ayuda.html')">  
Ayuda  
</a>
```

La forma correcta es implementar el enlace para que no dependa de javascript:

```
<script type="text/javascript" src="popup.js"></script>  
...  
<a href="ayuda.html" id="newwin">Ayuda</a>
```



# IPAP

INSTITUTO PROVINCIAL DE  
ADMINISTRACIÓN PÚBLICA  
DEL CHACO

```
window.onload = addHandlers;
```

```
function addHandlers(){  
    var objAnchor =  
    document.getElementById('newwin');  
    if (objAnchor){  
        objAnchor.firstChild.data =  
        objAnchor.firstChild.data + ' (se abre en ventana nueva)';  
        objAnchor.onclick = function(event){return  
        launchWindow(this, event);};  
        objAnchor.onkeypress = function(event){return  
        launchWindow(this, event);};  
    }  
}
```



```
function launchWindow(objAnchor, objEvent){
    var iKeyCode, bSuccess=false;
    if (objEvent && objEvent.type == 'keypress'){
        if (objEvent.keyCode)
            iKeyCode = objEvent.keyCode;
        else if (objEvent.which)
            iKeyCode = objEvent.which;
        if (iKeyCode != 13 && iKeyCode != 32)
            return true;
    }
    bSuccess = window.open(objAnchor.href);
    if (!bSuccess)
        return true;
    return false;
}
```



## Degradación elegante

Parte de implementar la solución teniendo en cuenta a los usuarios con los dispositivos y navegadores más modernos, y pensar después hacia atrás, en dar una solución alternativa a los usuarios con navegadores más antiguos, para ofrecerles una funcionalidad mínima y suficiente.

Ejemplo de aplicación del principio de degradación elegante:

¿Cómo solucionaría el enfoque de “degradación elegante” nuestro enlace para abrir un pop-up en aquellas circunstancias en las cuales no se soporta javascript?



```
<a href="javascript:;"  
onclick="window.open('ayuda.html')">Ayuda</a>
```

```
<noscript>
```

```
<p>Su navegador no admite javascript o lo tiene  
Deshabilitado.
```

```
Acceda desde aquí a la <a href="ayuda.html">página  
de Ayuda</a></p>
```

```
</noscript>
```



**IPAP**  
INSTITUTO PROVINCIAL DE  
ADMINISTRACIÓN PÚBLICA  
DEL CHACO

# Formularios



## Etiquetar correctamente los controles del formulario

El elemento LABEL permite indicar semánticamente qué texto funciona como etiqueta de un campo de formulario.

```
<label for="nombre">Nombre:</label>
```

```
<input type="text" name="nombre" id="nombre" />
```

Técnica de referencia de las WCAG 2.0:

[H44: Using label elements to associate text labels with form controls](#)



## Elementos con los que debe usarse LABEL

El elemento LABEL debe usarse con los elementos: “textarea”, “select”, y los “input” de tipo: “text”, “checkbox”, “radio”, “file” y “password”.

## Elementos con los que NO debe usarse LABEL

Por el contrario, el elemento LABEL no debe usarse con los “input” de tipo “image”, que se etiquetan utilizando el atributo ALT. Por ejemplo:

```
<input type="image" name="enviar" src="boton.gif" alt="Enviar" />
```





LABEL tampoco debe usarse en los “input” de tipo “submit” o “reset”, los cuales se etiquetan con su atributo VALUE.  
Por ejemplo:

```
<input type="submit" name="enviar" value="Enviar" />
```

Tampoco debe usarse con el elemento “button”, en el cual se usa su contenido como etiqueta.

Por ejemplo:

```
<button type="submit" name="enviar">Enviar</button>
```



## ¿Dónde situamos la etiqueta respecto al campo?

**Antes del campo**, bien encima, bien a la izquierda, con los elementos “textarea”, “select”, y los “input” de tipo “text”, “file” y “password”.

### Datos del producto

Nombre del producto

Descripción del producto

### Alta en el servicio

Nombre

Apellidos

DNI

Contraseña

Darme de alta



**Detras del campo con los “input” de tipo “radio” y “checkbox”.**



Técnica de referencia de las WCAG 2.0:

[G162: Positioning labelsto maximize predictability of relationships](#)



## ¿ Y si no podemos usar LABEL?

Las WCAG 2.0 nos permiten el uso del atributo *title* para etiquetar los controles de formularios e identificar su propósito.

```
<fieldset>  
  <legend>Fecha de nacimiento (dd/mm/aaaa):</legend>  
  <input type="text" id="dia" name="dia" title="Día (dos dígitos)" /> /  
  <input type="text" id="mes" name="mes" title="Mes (dos dígitos)" /> /  
  <input type="text" id="año" name="año" title="Año (cuatro dígitos)" />  
</fieldset>
```



Otro ejemplo habitual es usar TITLE para etiquetar el campo del buscador de la página.

```
<input type="text" name="buscar"  
      title="Escribe aquí lo que quieres buscar" />  
<input type="submit" value="Buscar" />
```



## ¿Cómo prevenir los errores de los usuarios en los formularios?

### Identificar los campos obligatorios

```
<label for="nombre">Nombre (obligatorio):</label>  
<input type="text" name="nombre" id="nombre" />
```

Adicionalmente podemos añadir al campo la propiedad "aria-required" de WAI-ARIA

```
<input type="text" name="nombre" id="nombre" aria-  
required="true" />
```



## Informar del formato requerido

```
<label for="fecha">Fecha (dd/mm/aaaa):</label>
```

```
<input type="text" aria-describedby="ayuda" ... />
```

```
<p id="ayuda">Por ejemplo: 12/03/2014</p>
```



## **Mejorar la accesibilidad de los mensajes de validación**

### Identificar y describir los errores mediante texto

Aunque se admiten opciones como ventanas de alerta o información contextual, lo preferible es:

- Informando mediante el texto, al comienzo del formulario, los errores existentes;
- Describiendo el problema e identificando los campos con error;
- Indicando o sugiriendo valores posibles.