GOBIERNO DE ESPAÑA

MINISTERIO DE INDUSTRIA, TURISMO Y COMERCIO

SECRETARÍA DE ESTADO DE TELECOMUNICACIONES Y PARA LA SOCIEDAD DE LA INFORMACIÓN

INTECO

Instituto Nacional de Tecnologías de la Comunicación

GUÍA DE ACCESIBILIDAD Y SEGURIDAD EN TRÁMITES ONLINE

Centro de Referencia en Accesibilidad y Estándares Web

Mayo 2009

ÍNDICE

1. OBJETIVO DE LA GUÍA

Para facilitar a los usuarios la tramitación de diferentes servicios, cada vez es más común la inclusión de los denominados **trámites online** en los propios sitios Web, permitiendo así que los usuarios puedan realizar sus operaciones desde cualquier terminal con acceso a Internet. Estos trámites, a su vez, suelen disponer de mecanismos de **seguridad** que garantizan la integridad y la confidencialidad de la información.

No obstante, el empleo de dispositivos de seguridad en trámites online sin tener en cuenta los requisitos de accesibilidad, puede ocasionar una importante barrera de acceso a determinados usuarios, por lo que se hace indispensable tener presentes los aspectos relacionados con la **accesibilidad** si se desea desarrollar un sitio Web accesible.

El objetivo de esta guía es la de ayudar a los desarrolladores a la hora de incluir mecanismos de tramitación seguros en los sitios Web que desarrollan, de tal forma que ésta sea accesible según los requisitos que dicta la norma UNE 139803:2004, permitiendo así el acceso a ellos a todos los usuarios independientemente de sus propias limitaciones o las derivadas de su entorno.

2. ELEMENTOS QUE PARTICIPAN EN UN TRÁMITE ONLINE

Para comenzar esta guía y con la finalidad de que su seguimiento sea lo más intuitivo y fácil posible, primeramente, se va a resaltar en una clasificación, cada uno de los elementos que pueden participar en una tramitación online, posteriormente se analizará cada uno de ellos, destacando sus principales aspectos relacionados con la accesibilidad y la seguridad.

2.1. SOPORTE DE LA TRAMITACIÓN

Toda tramitación posee un soporte de comunicación de datos, en el caso de los trámites online dicho soporte con el que interactúa el usuario es el **formulario**.

Figura 1. Formulario

Se deberá garantizar la accesibilidad de los propios formularios y de las medidas de seguridad que incorporen mediante:

- Asociación de etiquetas con sus respectivos controles.
- Inclusión de información adicional en los mismos.
- Agrupación de controles de formularios.
- Validación de los datos introducidos.

2.2. MECANISMOS DE ACCESO SEGURO

Los mecanismos de acceso seguro garantizan la confidencialidad de la información, permitiendo la correcta autenticación de los usuarios. Otras veces su uso es únicamente para prevenir que los "robots" acceda al sistema.

Los diferentes elementos que se van a analizar en esta guía son los siguientes:

- Captchas.
- Firma digital / DNI electrónico.
- Mecanismos de autenticación mediante códigos.

2.3. OTROS ASPECTOS

Además se tendrán en cuenta otros aspectos como:

- Certificaciones de seguridad.
- Duración del tiempo de sesión en los trámites online.

3. ACCESIBILIDAD Y SEGURIDAD EN UN FORMULARIO

Como se indicó anteriormente el soporte utilizado para la trasmisión de datos en una tramitación online es el formulario.

A continuación se procede a describir las diferentes características de accesibilidad y seguridad que deben incorporar los formularios.

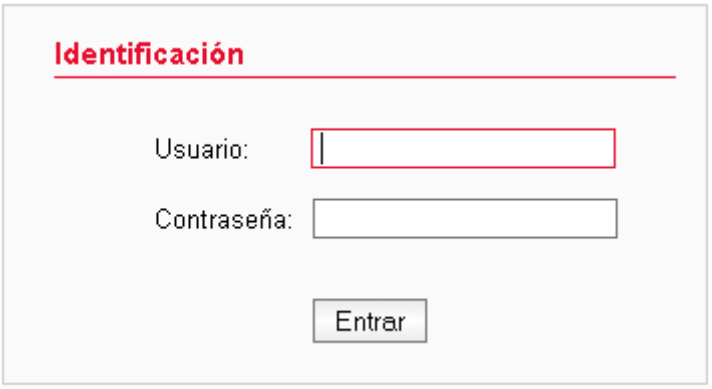

Figura 2. Formulario de identificación

3.1. ESTRUCTURACIÓN DE LAS ETIQUETAS Y CONTROLES

Para realizar una asociación correcta de las etiquetas de formularios con sus respectivos controles se deben asociar tanto implícita como explícitamente.

3.1.1. Asociación implícita

Hasta que las aplicaciones de usuario soporten la asociación explícita, se deben asociar implícitamente las etiquetas con sus controles; para ello, la etiqueta debe preceder inmediatamente a su control en la estructura del documento (en el caso de las casillas de verificación o botones de selección, el control precedería a la etiqueta).

Para realizar la asociación implícita se debe identificar una etiqueta para cada control de formulario mediante el elemento `LABEL`

Ejemplo de código:

```
<label for="nombre">Nombre: <input type="text" id="nombre" />
</label>
```

3.1.2. Asociación explícita

La asociación explícita elimina la dependencia en la colocación de la etiqueta y su control, ya que los agentes de usuario los asociarán automáticamente.

Para asociar explícitamente la etiqueta con su control de formulario se debe utilizar el atributo *id* en los controles del formulario y el atributo *for* en las etiquetas. El contenido de ambos atributos debe ser igual para cada par etiqueta-control.

Ejemplo de código:

```
<label for="nombre">Nombre: <input type="text" id="nombre" />
</label>
```

3.2. INFORMACIÓN ADICIONAL

A continuación se detalla como incluir información adicional en los formularios respetando las características de accesibilidad que deben poseer los mismos.

3.2.1. Campos obligatorios

En muchas ocasiones se desea indicar a los usuarios la obligatoriedad de rellenar determinados campos en un formulario.

Una manera adecuada y accesible para identificar estos campos obligatorios es mediante el uso de asteriscos. Se debe colocar antes del formulario una nota explicativa indicando que los campos con un asterisco son obligatorios.

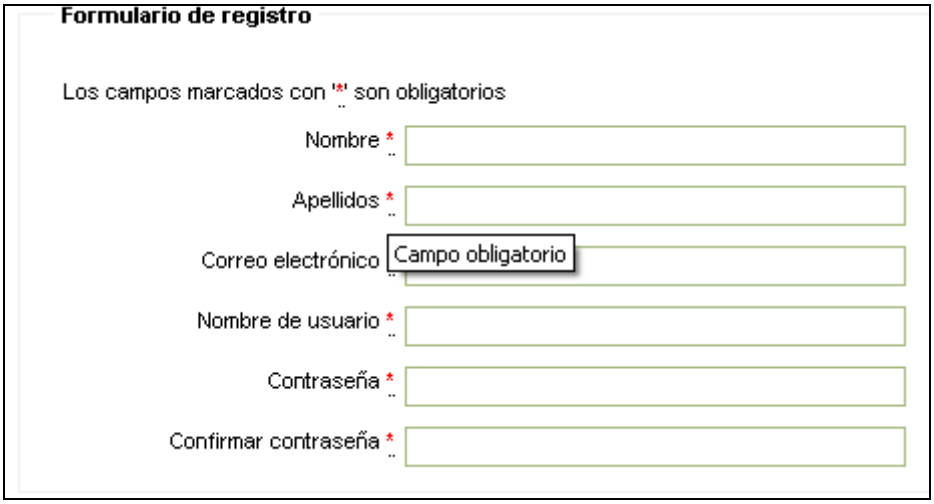

Figura 3. Campos obligatorios en un formulario de registro

En la etiqueta de cada campo obligatorio se colocará un asterisco. Dicho asterisco se podría marcar como una abreviatura cuyo atributo *title* será "campo obligatorio" por ejemplo.

Ejemplo de código:

```
<form action="formulario.html" method="post">
  <p>Nota: Los campos marcados con un * son obligatorios</p>
```

```
  <label for="nombre">
    <abbr title="campo obligatorio">*</abbr> Nombre:
    <input type="text" id="nombre" name="nombre" />
  </label>
  <input type="submit" value="Enviar" />
</form>
```

3.2.2. Otro tipo de información

Si se desea mostrar alguna otra indicación que requiera una explicación sobre el formulario es importante posicionar esa información antes del propio formulario, de forma que todos los usuarios localicen dicha información antes de rellenar el mismo.

No obstante si esta información es muy extensa es posible que se quiera acompañar una leyenda al formulario.

3.2.2.1. Leyenda

En el caso de que se desee incluir información sobre algún campo del formulario mediante el uso de una leyenda se debe realizar de la siguiente forma:

En el caso de que la leyenda sea muy amplia, lo mejor es colocarla después del formulario, y hacer referencias desde el campo correspondiente mediante enlaces a dicha leyenda. Finalmente se deberá incluir un enlace de retorno al campo de origen para que el usuario pueda seguir rellenando en el orden adecuado el formulario.

3.2.3. Agrupación de formularios

Tal y como indica la UNE 139803:2004 se debe agrupar la información cuando sea natural y apropiado, por lo tanto, cuando existan controles de formulario relacionados que se puedan agrupar en unidades lógicas, se debe utilizar el elemento `FIELDSET` y aplicar una etiqueta a esas unidades con el elemento `LEGEND`.

Ejemplo de código:

```
<form action="http://ejemplo.com/nuevousuario" method="post">
   <fieldset>
   <legend>Datos personales</legend>
   <label for="nombre">Nombre: </label>
   <input type="text" id="nombre" name="nombre" />
   <label for="apellidos">Apellidos: </label>
   <input type="text" id="apellidos" name="apellidos" />

   ...más datos personales...

   </fieldset>

   <fieldset>
   <legend>Historial médico</legend>
```

```
   ...datos del historial médico...

   </fieldset>
</form>
```

Cuando se deseen incluir listas largas de selección en los menús (en las cuales puede resultar difícil orientarse), se deberán agrupar los elementos `SELECT` (definidos mediante el elemento `OPTION`) en una jerarquía con el elemento `OPTGROUP`. Se podrá especificar una etiqueta para el grupo de opciones mediante el atributo *`label`* en el elemento `OPTGROUP`.

Ejemplo de código:

```
<form action="http://ejemplo.com" method="post">
 <p>Seleccione el requisito deseado:
 <select name="Prioridades accesibilidad según UNE 139803:2004">

    <optgroup label="Prioridad 1">
      <option value="4.2.1">Legibilidad sin CSS</option>
      <option value="4.2.2">Dependencia de color</option>
      <option value="4.5.1">Enlaces descriptivos</option>
    </optgroup>

    <optgroup label="Prioridad 2">
      <option value="4.1.4">Inclusión de metainformación
</option>
      <option value="4.3.5">Estructura de encabezados adecuada
</option>
      <option value="4.4.8">Asociación implícita de formularios
</option>
    </optgroup>

  </select>
  </p>
<input type="submit" value="Enviar" />
</form>
```

3.3. VALIDACIÓN DE LOS CAMPOS

La validación de los datos introducidos en un formulario se debe realizar de forma que no dependa únicamente de JavaScript. Normalmente se realizará la validación del lado del servidor. De esta forma, cuando no se disponga de JavaScript, el usuario enviará los datos del formulario sin validar y el servidor realizará la validación devolviendo el resultado de la misma con los errores que haya detectado, si los hubiese.

Por otra parte, se debe informar al usuario de todos los errores de validación que se produzcan. Dichos avisos se deberán mostrar de manera accesible antes del formulario de forma que no pasen inadvertidos para el usuario.

A continuación se muestra un ejemplo de validación en el lado del servidor implementada en código PHP.

Ejemplo de código:

```
<?php

// Primero se comprueba si se ha enviado el formulario o no

if ($_POST['submit']=="Acceder") {
      // Se comprueba si el contenido de los campos es correcto
      $errores = existen_errores($_POST);
      if (!$errores) {
            // Si no hay errores, es que los datos son
correctos, se permite el acceso al usuario
            echo '<p>Acceso permitido. Bienvenido.</p>';
      } else {
            // Imprime en pantalla los errores encontrados y
muestra de nuevo el formulario
            echo '<p>Se han producido los siguientes
errores:</p>';
            echo '<ul>';
            if ($errores['login_vacio'] != "") echo
'<li>'.$errores['login_vacio'].'</li>';
            if ($errores['password_vacio'] != "") echo
'<li>'.$errores['password_vacio'].'</li>';
            if ($errores['password_no_valido'] != "") echo
'<li>'.$errores['password_no_valido'].'</li>';
            echo '</ul>';
      }
}

?>

<form action="form.php" method="post">
<fieldset>
<legend>Identificación</legend>
<div>
      <p><label for="login">Usuario: <input type="text"
name="login" id="login" /></label></p>
      <p><label for="password">Contraseña: <input
type="password" name="password" id="password" /></label></p>
      <p><input type="submit" value="Acceder" name="submit"
/></p>
</div>
</fieldset>
</form>

[…]

<?php

function existen_errores($_POST) {
$errores = false;
if ($_POST['login'] == "") {
```

```
      // Comprueba si se ha introducido un usuario
      $errores['login_vacio'] = "El campo Usuario se encuentra
vacío.";
}

if ($_POST['password'] == "") {
      // Comprueba si se ha introducido una contraseña
      $errores['password_vacio'] = "El campo Contraseña se
encuentra vacío.";
}

// En caso de que se haya introducido usuario y contraseña
comprueba si son correctos (se utilizan valores fijos en el
ejemplo para hacerlo más sencillo)
if ( !($_POST['login'] == "administrador" && $_POST['password']
== "secreto") ) {
      $errores['password_no_valido'] = "La combinación de
usuario y contraseña introducida no es correcta.";
}

// Devuelve la variable que contiene los errores encontrados, o
'false' si no se encontró ninguno
return $errores;
}
```

4. ACCESIBILIDAD EN MECANISMOS DE ACCESO

4.1. CAPTCHAS

Actualmente las técnicas Captcha utilizadas en los formularios de acceso no garantizan ni la accesibilidad ni la seguridad necesaria. Por otra parte, estas técnicas no se pueden considerar test de Turing totalmente fiables.

A continuación se definen las técnicas más habituales de **Captcha** describiendo las principales ventajas e inconvenientes.

4.1.1. Captcha gráfico

Se trata del elemento más habitual y consiste en una imagen en la que se muestran una serie de caracteres alfanuméricos manipulados para dificultar su percepción.

Figura 4. Ejemplo de Captcha gráfico

Esta tecnología aporta seguridad al sistema, si bien las técnicas de reconocimiento de caracteres OCR actuales permiten evitar en muchos casos esta barrera. Para mejorar la seguridad de estos mecanismos se debe dificultar la legibilidad con lo que se empeora el grado de accesibilidad.

Por otra parte, en relación a la accesibilidad se ha de tener en cuenta que todo elemento no textual debe contener un equivalente en formato texto que proporcione toda la información que presente la imagen. Sin embargo para que dicho mecanismo siga siendo seguro, la alternativa no puede contener el conjunto de caracteres que se muestran en el Captcha, con lo que se impide el acceso a usuarios que no puedan visualizar la imagen.

4.1.2. Captcha auditivo

Se trata de un archivo de sonido equivalente al Captcha gráfico. Este archivo indica una serie de caracteres alfanuméricos al reproducirse mediante sonidos distorsionados o con ruido de fondo.

Sin embargo, estos mecanismos conllevan el mismo problema que sus equivalentes gráficos: se pueden evitar mediante técnicas de reconocimiento de voz y precisan de una alternativa textual que proporcione la misma información y que por seguridad no puede incluirse en los documentos de registro de usuarios.

De la misma forma que para los Captchas gráficos, el grado de accesibilidad y seguridad de este mecanismo es inversamente proporcional, por lo que se debe buscar un equilibrio entre ambos aspectos.

Estos dos mecanismos (Captcha gráfico y auditivo) se pueden emplear de forma complementaria facilitando al usuario la elección de uno de ellos. A pesar de poder facilitar el acceso de este modo, puede haber usuarios a los que se impida acceder, como usuarios sordociegos.

4.1.3. Captcha Lógico

Consiste en preguntas sencillas que no puedan ser respondidas por una máquina, por ejemplo: *"¿De que color es el caballo blanco de Santiago?"* ó *"¿Cuántos dedos hay en una mano?"*.

Este mecanismo es el que facilita el acceso a un mayor número de usuarios y en consecuencia el mejor desde el punto de vista de la accesibilidad, siempre y cuando las preguntas que se realicen no sean difíciles de comprender o requieran de conocimientos especiales. Sin embargo, proporciona menos seguridad que otros mecanismos.

La seguridad de este mecanismo es mayor cuantas más preguntas posibles pueda mostrar el sistema, aún así este número es siempre limitado.

4.1.4. Conclusión

Ninguno de los mecanismos citados anteriormente resuelve por sí mismo el problema de incorporar a la vez un mecanismo de seguridad como son los Captcha y al mismo tiempo permitir el acceso universal.

Sin embargo se ha de intentar lograr un equilibrio entre ambos requerimientos. Para ello se pueden proporcionar varias alternativas equivalentes de Captcha que abarquen las diferentes dificultades de acceso y a su vez incorporar un método alternativo para aquellos usuarios que aún así no puedan acceder a estos mecanismos.

Este método alternativo puede consistir en una dirección de correo electrónico de contacto para dichos usuarios, un envío de mensaje de texto por móvil, un número de teléfono de contacto con teleoperadores, etc.

Para más información sobre este tema se recomienda visitar el siguiente artículo del W3C: http://www.w3.org/TR/turingtest

4.2. FIRMA DIGITAL Y DNI ELECTRÓNICO

En el desarrollo actual de aplicaciones seguras, se hace imprescindible asegurar la **autenticidad** del usuario, **privacidad** de los datos intercambiados, y la **fiabilidad** de las transacciones ejecutadas.

En este sentido surgen soluciones como la **firma digital** o el **DNI electrónico**. Estas soluciones se basan en la utilización de sistemas criptográficos para alcanzar los objetivos de seguridad. Los citados sistemas criptográficos se materializan en certificados (claves) alojados en el ordenador personal del usuario que desea realizar la operación.

Figura 5. DNI Electrónico

El problema surge dado que para **la lectura de estos certificados** por parte de un portal Web se debe utilizar **tecnología de ejecución dinámica**, de mayor potencia que el lenguaje HTML estándar. Tal tecnología consiste en scripts, applets, objetos ActiveX u otros elementos que puedan ser ejecutados en el ordenador personal del usuario en el momento de la operación (para leer el certificado).

La ejecución de estos elementos choca con los requisitos de accesibilidad de la Norma UNE 139803:2004 que hace referencia a que las páginas deben poder utilizarse aunque los scripts y objetos de programación estén desconectados o no sean soportados.

Aquellos usuarios que tengan desactivada la ejecución de este tipo de tecnologías (o no dispongan de ellas) en sus ordenadores personales no podrán acceder a la aplicación. Esto supone, aplicando los criterios de accesibilidad expuestos, un incumplimiento de los mismos. Por lo que evita que la aplicación, y por extensión el portal, sean accesibles.

Respecto a este hecho se entiende que existe una barrera tecnológica que no es posible superar, dado que **no hay forma de leer las claves de un usuario prescindiendo de este tipo de tecnologías**.

No obstante, se recomienda, en cualquier caso, que frente a la posibilidad de realizar la citada operación online se permitan otras **alternativas** como por ejemplo posibilitar la realización presencial de la operación, incluir un formulario PDF accesible firmado digitalmente, posibilitar la tramitación telefónica, por correo postal, etc.

4.3. MECANISMOS DE AUTENTICACIÓN MEDIANTE CÓDIGOS

Los mecanismos de validación de códigos permiten autenticar adecuadamente a los usuarios, garantizando la confidencialidad de la información a la que acceden.

El usuario, gracias a ellos, introduce una contraseña de acceso. En ocasiones para fortalecer la seguridad, dicha contraseña se genera a partir de una tarjeta de claves o a través de un código que incluye el propio mecanismo.

Normalmente en estos mecanismos se incluyen teclados virtuales en la introducción de la contraseña para evitar que un posible keylogger pueda almacenar esta información.

Actualmente hay muchos organismos que hacen uso de estos mecanismos en sus sitios Web.

Figura 6. Mecanismo de autenticación mediante códigos

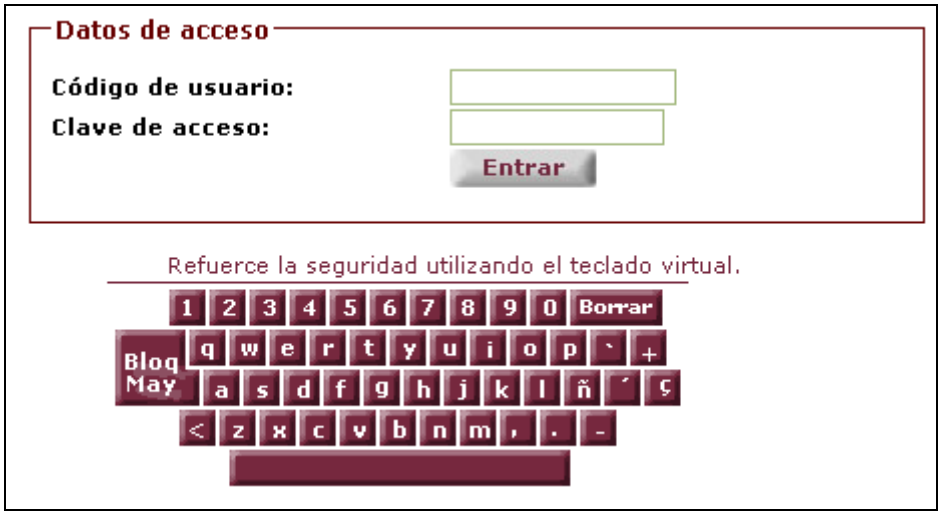

Figura 7. Mecanismo de autenticación mediante códigos

No obstante, en muchas ocasiones, estos mecanismos no son accesibles, ya que se encuentran desarrollados mediante JavaScript intrusivo y no se posibilita ninguna alternativa, por lo que no se garantiza el acceso a todos los usuarios.

Teniendo en cuenta que la finalidad principal de estas funcionalidades es expresamente para aumentar la seguridad de los usuarios, se recomienda que se implemente de la manera más accesible posible, respetando los requisitos de la Norma UNE 139803:2004 e incluyendo otro tipo de alternativas adicionalmente en el caso de que sea necesario.

5. OTROS ASPECTOS

5.1. CERTIFICACIONES DE SEGURIDAD

Generalmente, en trámites donde se requiera un mínimo de seguridad, cuando se desea que la información esté encriptada es habitual incluir certificados para usar Web vía SSL, implementando HTTPS en la comunicación para asegurar el tráfico entre cliente y servidor.

Una de las ventajas de usar este tipo de certificados públicos, es que no requieren instalación de ningún servicio adicional, la conexión con los clientes es transparente, pero eso sí, requieren de un costo de mantenimiento.

Figura 8. Conexión segura mediante https

Habitualmente, los navegadores suelen avisar a los usuarios, cuando se navega por documentos mediante HTTPS, incluyendo un icono de un candado en la barra de estado. También es posible configurar algunos navegadores para que avisen a los usuarios mediante una alerta cuando van acceder a este tipo de documentos. Por otra parte, en algunas versiones de lectores de pantallas, como JAWS, es posible configurarlo para que avise a los usuarios de este suceso.

Figura 9. Icono de acceso seguro en la barra de estado del Mozilla Firefox

No obstante, y para garantizar que todos los usuarios tengan acceso a esa información tan importante, se recomienda que se indique en los contenidos de los documentos cuando se va a acceder a documentos seguros. De esta forma, los usuarios conocerán con certeza cuando sí y cuando no tendrán una garantía de seguridad de la transmisión de los datos que incluyan en sus trámites.

5.2. TIEMPOS DE SESIÓN

Otra de las cuestiones que hay que tener en cuenta por temas de seguridad es la de controlar el tiempo en el que está activa la sesión. Por ejemplo, para evitar que una persona olvide "desconectarse" y otra aproveche su usuario cuando no esté, pudiendo suplantar su identidad o acceder a información confidencial.

No obstante, desde el punto de vista de la accesibilidad, estos tiempos de sesión en ocasiones no son lo suficientemente apropiados para que todos los usuarios sean capaces de rellenar los trámites sin que caduque la sesión. Por ello, se deben utilizar tiempos de sesión que, no siendo demasiado excesivos para mantener el nivel de seguridad, permitan que usuarios que no puedan acceder a la página de forma “rápida” dispongan del tiempo suficiente para completar el trámite.

6. RECOMENDACIONES DE ACCESIBILIDAD EN TRÁMITES ONLINE SEGUROS

Por último, para realizar un resumen de los contenidos incluidos a lo largo de la guía de accesibilidad y seguridad en trámites online, se muestran a continuación una serie de recomendaciones que se deben tener en cuenta cuando se implemente un trámite online seguro:

- Los formularios deben poseer características de accesibilidad adecuadas; **asociando implícita** y **explícitamente** las etiquetas con sus controles de formulario, colocando y marcando adecuadamente la **información adicional** en el mismo, **agrupándose** correctamente, etc.

- La **validación** de la información incluida en los campos del **formulario** no debe depender de JavaScript, por lo que se debe realizar **desde el servidor**, aunque es posible utilizar JavaScript como complemento para mejorar la experiencia del usuario.

- Si se incorporan **captchas**, debe ofrecerse la posibilidad de que **todos** los **usuarios** sean capaces de **interpretarlos**.

- Se deben proporcionar alternativas a la firma digital y al DNI electrónico.

- Los mecanismos de **validación por códigos** deben respetar los requisitos de **accesibilidad**.

- Es recomendable **avisar al usuario** cuando **va a acceder** a documentos mediante una **conexión segura.**

- Se recomienda que el **tiempo de sesión** sea el **suficiente** para que todos los usuarios sean capaces de rellenar los trámites satisfactoriamente.