UNIDAD DE TRABAJO 4

diseño de interfaces web



Inés Gómez Hernáez

tarea 1. creación de sitio web

[1. ACCESO AL SITIO WEB 2](#_Toc156734642)

[2. JUSTIFICACIÓN DEL HTML 2](#_Toc156734643)

[2.1. INDEX.HTML 2](#_Toc156734644)

[2.2. TIPOS-DE-ROBOTS.HTML 3](#_Toc156734645)

[2.3. ROBOTS-DE-FICCIÓN.HTML 3](#_Toc156734646)

[2.4. METODOLOGÍA BEM 4](#_Toc156734647)

[2.5. VALIDACIÓN HTML 5](#_Toc156734648)

[3. JUSTIFICACIÓN DEL CSS 8](#_Toc156734649)

[3.1. VARIABLES DE COLOR 8](#_Toc156734650)

[3.2. RESETEAR VALORES 8](#_Toc156734651)

[3.3. ESTILOS GENERALES DEL PROYECTO 8](#_Toc156734652)

[3.3.1. BODY 8](#_Toc156734653)

[3.3.2. IMG (IMAGENES) 9](#_Toc156734654)

[3.3.3. FIGURE, FIGCAPTION 9](#_Toc156734655)

[3.3.4. A (ENLACES) 9](#_Toc156734656)

[3.3.5. P (PARRAFOS) 9](#_Toc156734657)

[3.3.6. UL, LI (LISTAS NO ORDENADAS) 9](#_Toc156734658)

[3.3.7. H1, H2 (ENCABEZADOS) 10](#_Toc156734659)

[3.3.8. BLOCKQUOTE, CITE (CITAS) 10](#_Toc156734660)

[3.3.9. SECTION (SECCIONES) 10](#_Toc156734661)

[3.3.10. IFRAMES (VIDEOS) 11](#_Toc156734662)

[3.4. BLOQUES PRINCIPALES 11](#_Toc156734663)

[3.4.1. HEADER 11](#_Toc156734664)

[3.4.2. MAIN 12](#_Toc156734665)

[3.4.3. FOOTER 13](#_Toc156734666)

[3.4.4. NAV 13](#_Toc156734667)

[3.5. REGLAS PARA DISEÑO RESPONSIVE 13](#_Toc156734668)

[3.6. ESTILOS ESPECIFICOS DE CADA PÁGINA 14](#_Toc156734669)

[3.6.1. INDEX.HTML 14](#_Toc156734670)

[3.6.2. TIPOS-DE-ROBOTS.HTML 15](#_Toc156734671)

[3.6.3. ROBOTS-DE-FICTION.HTML 15](#_Toc156734672)

[3.7. VALIDACIÓN CSS 16](#_Toc156734673)

[4. VALIDACIÓN ACCESIBILIDAD 16](#_Toc156734674)

[4.1. RESULTADO DE LA VALIDACIÓN 16](#_Toc156734675)

[4.2. CORRECCIÓN DE ERRORES 18](#_Toc156734676)

[4.3. CORRECCIONES TRAS LAS VALIDACIONES 19](#_Toc156734677)

tarea 1. CREACIÓN DE SITIO WEB

# ACCESO AL SITIO WEB

El sitio web de la tarea ha sido publicado mediante Netlify y es accesible desde las siguiente URL:

<https://timely-khapse-1ebdad.netlify.app>

El código completo del sitio web se encuentra disponible en:

# JUSTIFICACIÓN DEL HTML

## ESTRUCTURA SEMÁNTICA

El HTML de la página join.html se ha estructurado utilizando etiquetas semánticas para mejorar la comprensión del contenido y facilitar la navegación del usuario. Además, mantiene coherencia con el resto de páginas del sitio web

**Encabezado (<header>):**

El encabezado del documento contiene la navegación principal del sitio, identificado por la etiqueta <nav>, donde se encuentran los enlaces a diferentes secciones como "Sobre Petinder", "Camiseta" y "Hazte socio". Esto permite que los usuarios puedan moverse fácilmente entre las distintas partes del sitio.

**Cuerpo Principal (<main>):**

El cuerpo principal del documento, representado por la etiqueta <main>, contiene el contenido principal de la página. En este caso, presenta la información y el formulario para que los usuarios puedan hacerse socios de Petinder.

**Formulario (<form>):**

El formulario está marcado correctamente con la etiqueta <form>, lo que indica su propósito y permite que los navegadores y las tecnologías de asistencia lo interpreten correctamente. Dentro del formulario, los campos de entrada (<input>, <select>, <textarea>) están etiquetados adecuadamente con la etiqueta <label> para asociarlas con sus descripciones correspondientes.

**Secciones Semánticas (<section>, <fieldset>, <legend>):**

El contenido se organiza en secciones semánticas utilizando etiquetas como <section>, <fieldset> y <legend>. Por ejemplo, los datos personales y la dirección tienen sus propios fieldsets con leyendas descriptivas para proporcionar una estructura clara y comprensible al formulario.

**Pie de Página (<footer>):**

El pie de página del documento contiene información adicional y enlaces legales relacionados con el sitio. Esto incluye el año de copyright, los términos y condiciones, la política de privacidad, las cookies y las preguntas frecuentes.

## FUNCIONALIDAD DEL FORMULARIO

En los formularios, se pueden utilizan de atributos y elementos que brindan funcionalidad y ayudan a validar el contenido introducido por el usuario, sin tener que recurrir a código JavaScript. En el diseño del formulario he incluido 3 atributos que son clave para este fin: title, maxlength, required y pattern.

Cada uno de estos atributos desempeña un papel fundamental en la experiencia del usuario al interactuar con el formulario, proporcionando información contextual, indicaciones visuales y requisitos de entrada obligatorios.

**Atributo title**

Este atributo se utiliza para proporcionar información adicional sobre el propósito o la restricción de un campo. En este formulario, se usa, por ejemplo, en el campo DNI para indicar al usuario que debe ingresar un DNI válido, que consta de 8 dígitos seguidos de una letra:

<li class="form-item required">

<label for="dni">DNI</label>

<input

class="item-control"

type="text"

name="dni"

id="dni"

placeholder="DNI"

value=""

size="9"

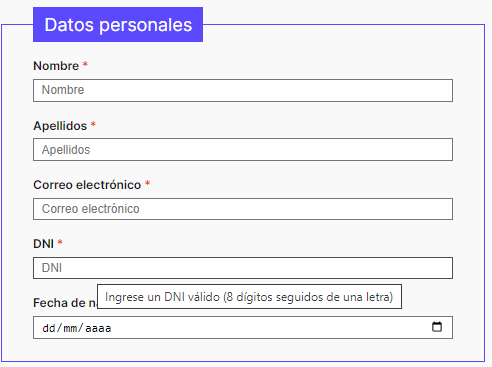
pattern="[0-9]{8}[A-Za-z]"

title="Ingrese un DNI válido (8 dígitos seguidos de una letra)"

required

/>

Al posicionarnos sobre este campo en el formulario aparece un tooltip con el texto indicado:



También se ha empleado este atributo en el campo de fecha.

**Atributo maxlength**

Este atributo se utiliza para especificar la longitud máxima del campo de entrada. En este formulario, se utiliza en los campos de Nombre, Apellidos, Correo electrónico, DNI, Calle y numero, Cuidad, Provincia y Código Postal.

<li class="form-item required">

<label for="codigoPostal">Codigo postal</label>

<input

class="item-control"

type="text"

name="codigoPostal"

id="codigoPostal"

placeholder="Codigo postal"

value=""

maxlength="10"

required

/>

</li>

El empleo de este atributo garantiza que los usuarios no ingresen más información de la necesaria en un campo, especialmente en casos donde hay limitaciones de espacio o donde se espera un formato específico. Por ejemplo, al utilizar maxlength en un campo DNI, se garantizar que no se ingresen más caracteres de los esperados en ese formato específico.

**Atributo required**

Este atributo se utiliza para especificar que un campo debe ser completado antes de enviar el formulario. En este formulario, se utiliza en todos los campo que deben ser obligatorios, por ejemplo, en el compo nombre:

<input

class="item-control"

type="text"

name="nombre"

id="nombre"

placeholder="Nombre"

value=""

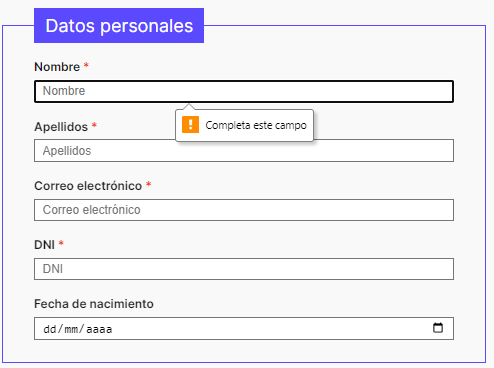
size="50"

pattern="[a-zA-ZáéíóúÁÉÍÓÚ\s]+"

required

/>

Con este atributo se garantiza que no se envíe un formulario sin completar los campos obligatorios. Si se trata de enviar sin completar los campos obligatorios se mostrará un mensaje indicando que falta de completar el campo:



**Atributo pattern**

Su función es especificar una expresión regular que el valor del campo debe cumplir para ser considerado válido. En este formulario se ha utilizado para validar el formato de los campos Nombre, Apellidos, Correo electrónico, DNI y Fecha de nacimiento.

Por ejemplo, para validar el campo de DNI, se utiliza una expresión regular que coincida con el formato típico de un DNI español.

**[0-9]{8}[A-Za-z]**

Esta expresión regular indica:

[0-9]{8}: dígitos del 0 al 9. El {8} indica que deben aparecer exactamente 8 dígitos.

[A-Za-z]: Este segmento coincide con una letra, ya sea mayúscula o minúscula.

<li class="form-item required">

<label for="dni">DNI</label>

<input

class="item-control"

type="text"

name="dni"

id="dni"

placeholder="DNI"

value=""

size="9"

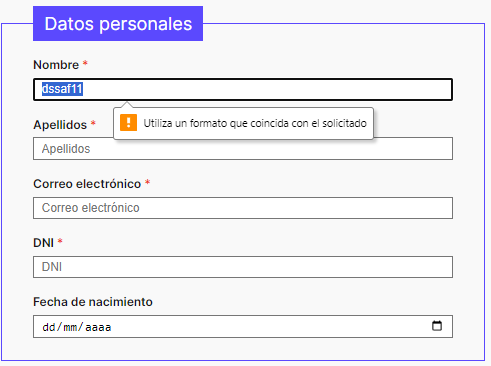
pattern="[0-9]{8}[A-Za-z]"

title="Ingrese un DNI válido (8 dígitos seguidos de una letra)"

required

/>

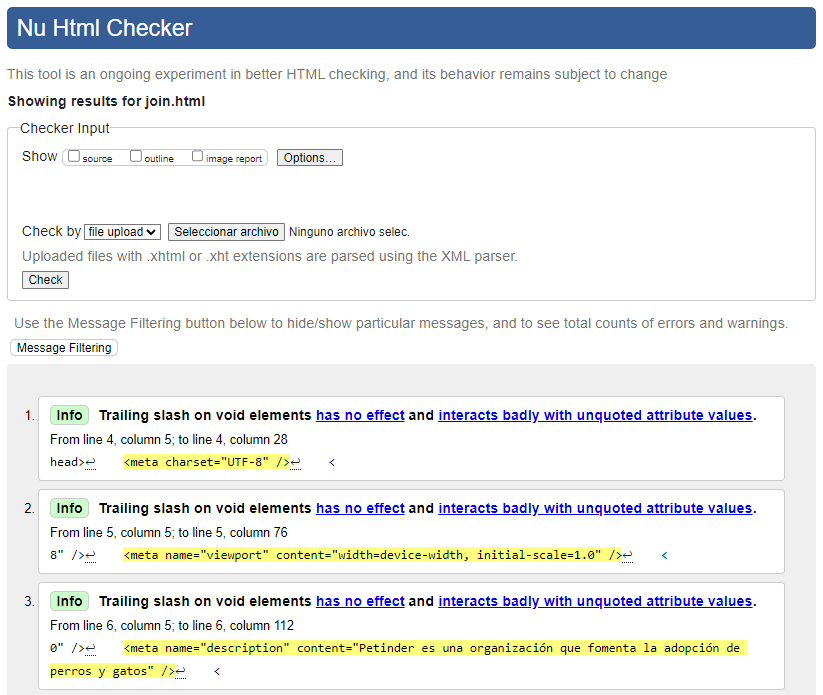
Con este atributo se garantiza que no se envíe un formulario con datos que no cumplan con el formato establecido. Si se trata de enviar el formulario y alguno de los campos no cumple con la expresión regular indicada se mostrará un mensaje indicando el error:



## VALIDACIÓN HTML

Por último, he realizado la validación del código HTML de las 3 páginas del sitio web utilizando la aplicación web <validator.w3.org>.

Los resultados obtenidos han sido los siguientes. Como se observa no se obtienen errores, solo avisos informativos:



# JUSTIFICACIÓN DEL CSS

En este apartado, se detallan los diferentes estilos aplicados y las razones detrás de su elección:

**Formulario**

He creado un estilo que se aplica sobre la etiqueta <form>. A esta etiqueta le he aplicado un margen un margen superior de 40px y un ancho máximo de 650px para proporcionar un espaciado adecuado y limitar su extensión horizontal en pantallas grandes.

form {

  margin-top: 40px;

  max-width: 650px;

}

**Fieldset**

Se ha añadido un borde sólido a la etiqueta <fieldset> para resaltar visualmente los campos relacionados. Además, se ha incluido un margen inferior de 25px para separar visualmente los fieldsets adyacentes y mejorar la legibilidad del formulario.

fieldset {

  border: 1px solid var(--primary-color);

  margin-bottom: 25px;

}

**Leyendas**

A las leyendas dentro del fieldset se les ha añadido el color azul en el fondo, el texto en blanco y un tamaño de fuente más grande para resaltar su importancia. Además se le han aplicado margins y padding para mejorar la legibilidad.

legend {

  margin-left: 30px;

  margin-bottom: 15px;

  padding: 5px 15px;

  background-color: var(--primary-color);

  color: white;

  font-size:1.5rem;

}

**Títulos y Párrafos**

Se han ajustado los estilos predeterminados de los párrafos y títulos para garantizar una coherencia visual con el diseño del formulario. Para ello, se ha establecido un margen superior de 5px para los títulos (<h2>) dentro del formulario, mientras que se ha eliminado el margen superior de los párrafos (<p>). Además, se ha seleccionado un tamaño de fuente óptimo que mejora significativamente la legibilidad del contenido.

form h2{

  margin: 5px 0;

}

form p{

  margin-top: 0;

}

**Botones**

Se definen los estilos para los botones dentro del formulario. Se ponen con bordes redondeados, el color azul de fondo y el texto blanco. Se añade un efecto de cambio de color al pasar el cursor sobre el botón para mejorar la retroalimentación visual para el usuario, indicando la interactividad del elemento.

**Elementos del Formulario**

Se ha creado la clase “form-item” para definir los estilos de los diferentes elementos del formulario, que incluyen la label y el campo del formulario. Se proporciona un padding para crear espacio visual entre los elementos y dar una apariencia más ordenada al formulario.

.form-item {

  padding: 0 30px 15px 30px;

}

**Campos de entrada**

Se ha creado la clase “item-control” para dar estilo a los input, los radio-button y los select del formulario. Se aplica un padding y un tamaño de fuente legible y se establece el ancho y el alto.

/\* Estilos para los los campos del formulario \*/

.form-item .item-control {

  box-sizing: border-box;

  padding-left: 10px;

  padding-right: 10px;

  margin: 5px 0;

  height: 30px;

  width: 100%;

  font-size: 1rem;

}

**Etiquetas y Campos Requeridos**

Las etiquetas (<label>) de los elementos del formulario se han resaltado con negrita para mejorar su visibilidad y jerarquía visual dentro del formulario.

Además, se ha añadido un asterisco rojo junto a las etiquetas de los campos requeridos para indicar claramente al usuario los campos obligatorios del formulario. Para ello, se ha creado la clase “required” y se añade el símbolo “\*” después de todas las label que estén contenidas dentro de un elemento con esa clase.

/\* Estilos para las label dentro de cada elemento del formulario \*/

.form-item label {

  font-weight: 600;

}

/\* Estilo para añadir el simbolo "\*" junto a la label de los elementos requeridos del formulario \*/

.form-item.required label::after {

  color: #e32;

  content: " \*";

  display: inline;

}

**Texto de Información Requerida**

Se ha creado la clase “info-required” para poder dar estilo al texto informativo sobre los campos requeridos. Este texto se ha resaltado con un color rojo para llamar la atención del usuario sobre la importancia de completar estos campos.

## VALIDACIÓN CSS

Por último, he realizado la validación del código CSS del sitio web utilizando la aplicación web <https://jigsaw.w3.org/css-validator/>

Los resultados obtenidos han sido los siguientes.



Como se observa el CSS pasa la validación sin problemas.

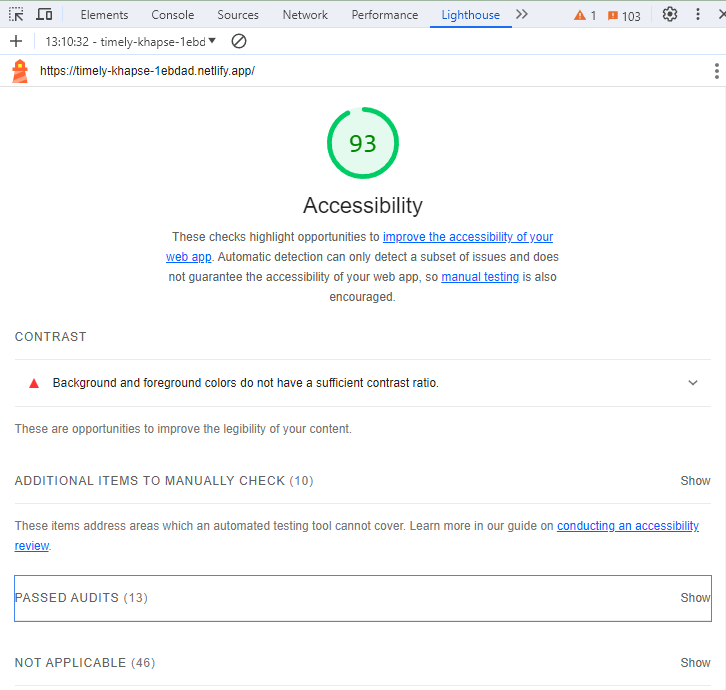
# VALIDACIÓN ACCESIBILIDAD

El ultimo paso de diseño del sitio web de robots será comprobar si la web cumple las condiciones mínimas de accesibilidad.

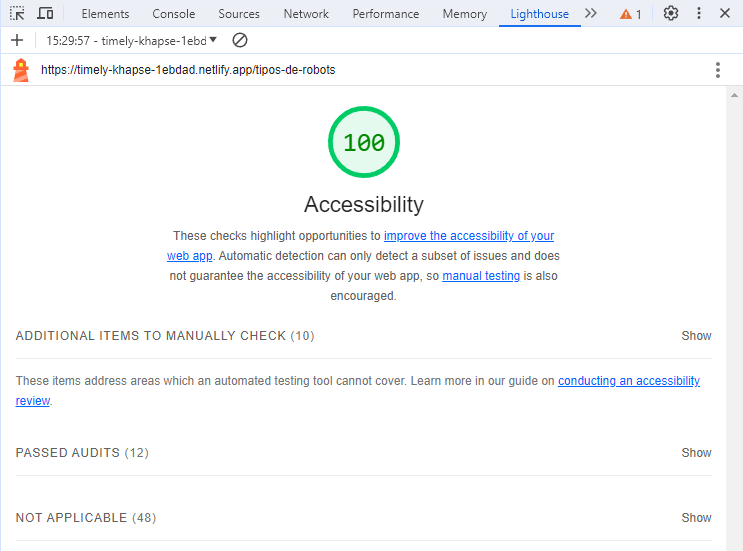
## RESULTADO DE LA VALIDACIÓN

La validación de la accesibilidad la he realizado utilizado la herramienta Lighthouse del navegador Chrome sobre cada una de las páginas. Los resultados obtenidos son:

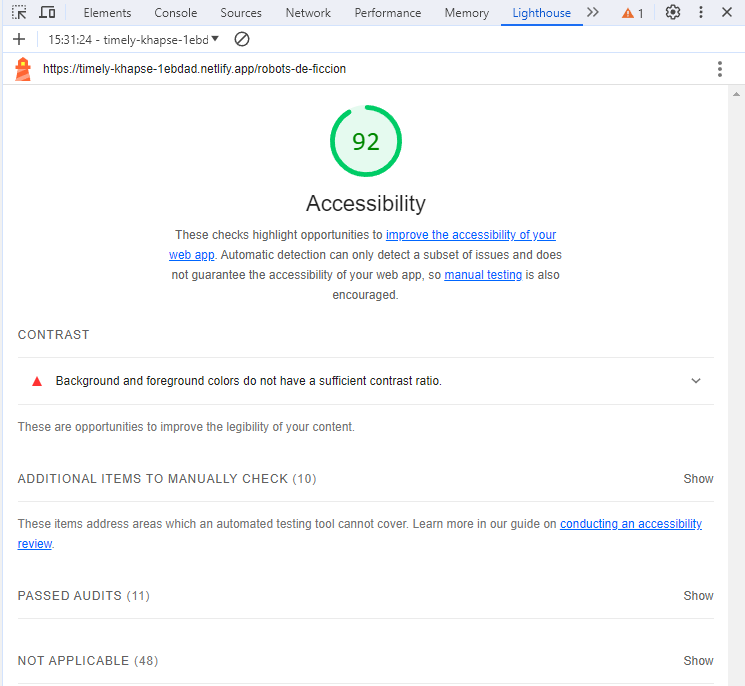
**INDEX.HTML**



**ROBOTS-DE-FICCIÓN.HTML**



**TIPOS-DE-ROBOTS.HTML**



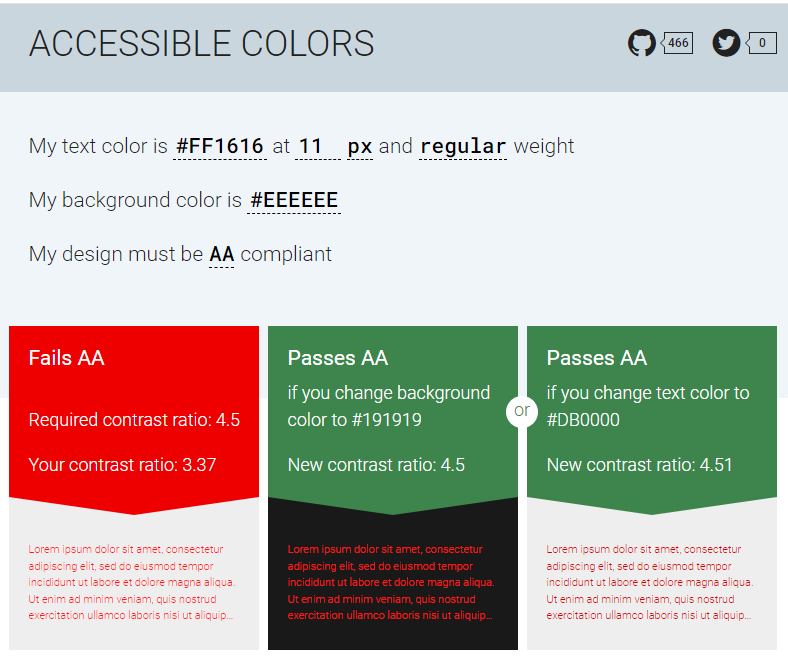
Como puede verse en las páginas anteriores, la página de robots-de-ficción cumple con los criterios mínimos de accesibilidad, pero las otras dos páginas tienen problemas de contraste. Concretamente en el contraste entre el color rojo empleado con respecto al fondo.

## CORRECCIÓN DE ERRORES

Para corregir el error he empleado la herramienta web [Accesible Colors](https://accessible-colors.com/) para seleccionar un color mas adecuado. He probado el caso mas desfavorable, que es el del color de la información del artículo de la página index.html. Este texto tiene un tamaño bastante pequeño y se muestra sobre un fondo gris.



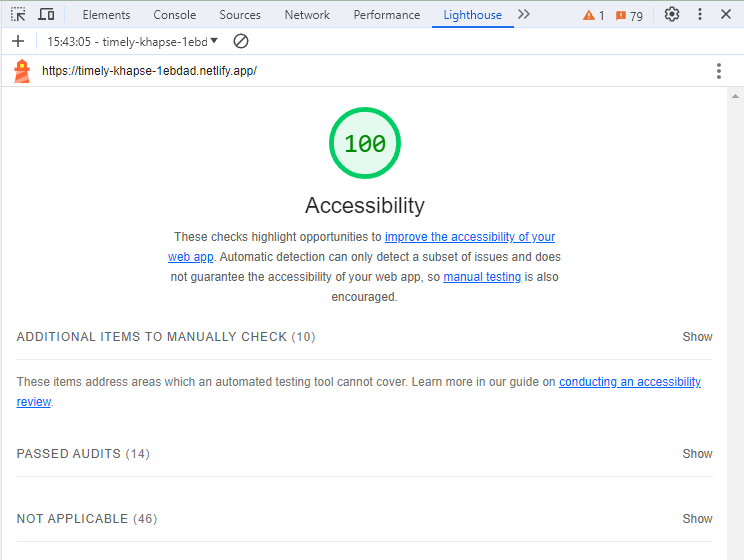
El resultado es que debería utilizar el color #DB0000 para sustituir al actual #FF1616:



## CORRECCIONES TRAS LAS VALIDACIONES

Una vez solventados los problemas detectados en el sitio web, se procede a validar de nuevo las páginas. Esta vez el resultado obtenido en todas ellas es del 100% en accesibilidad.

**INDEX.HTML**



**TIPOS-DE-ROBOTS.HTML**

