

HTML y CSS

Actividad 3 (P3): Informe de Características del Sitio

Nombre y Apellido: Huimin Tao

Fecha: 15/01/2024

ÍNDICE

1. Breve Descripción del Sitio Web	3
1.1 Servicio de publicación	3
1.2 Organización de archivos	3
2. Validaciones de las Páginas.....	4
2.1 index.html.....	4
2.2 blog.html.....	4
2.3 entrada.html	5
2.4 videos.html	5
2.5 estilos.css	5
3. Explicaciones de las Páginas	6
3.1 index.html	6
3.2 blog.html	10
3.3 entrada.html	11
3.4 videos.html	12
3.5 estilos.css	13

1. Breve Descripción del Sitio Web

1.1 Servicio de publicación

He utilizado el servicio de publicación proporcionado por la aplicación WebFTP del campus para alojar mi proyecto, y el URL que conduce a la primera página (índex.html) del prototipo que he desarrollado es :

https://protected-content.ftp.uoc.edu/20231_M4.252_html/aula2/htao%40uoc.edu/PEC3_HUIMIN_TAO/index.html

1.2 Organización de archivos

A continuación está la estructura de la organización de archivos :

(1) Carpeta Principal: “PEC3_HUIMIN_TAO”:

Dentro de esta carpeta principal, hay diferentes archivos y subcarpetas que constituyen la estructura del sitio web.

(2) Archivos HTML:

- **index.html**: la página de inicio del sitio.
- **blog.html**: la página dedicada al blog.
- **entrada.html**: la página del artículo individual del blog.
- **videos.html**: la página que contenga la lista de vídeos.

(3) Subcarpetas:

- **Carpeta "css"**: Contiene el archivo **estilos.css**, que define los estilos aplicados a lo largo del sitio web.
- **Carpeta "img"**: Contiene subcarpetas para imágenes organizadas por sección:
 - **Subcarpeta "favicon"** : Se encuentran los dos tamaños distintos de la imagen de favicon: “favicon.png” es la imagen original proporcionada por la profesora, mientras que “favicon-resized.png” ha sido reducido de tamaño (32 * 32) para acelerar la carga del sitio sin sacrificar la calidad visual, garantizando así una experiencia de usuario fluido.

- Subcarpeta "logo": Se encuentra los dos logos diferentes para el *header* (logoVital360.svg) y *footer* (logoVital360-n.sv) respectivamente.
- Subcarpeta "img-index": Contiene imágenes específicas de la página de inicio.
- Subcarpeta "img-blog": Contiene imágenes relacionadas con el blog.
- Subcarpeta "img-redes_sociales": Incluye los íconos de redes sociales para el *footer*.

2. Validaciones de las Páginas

2.1 index.html

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for uploaded file **index.html**

Checker Input

Show ☐ source ☐ outline ☐ image report [Options...](#)

Check by [file upload](#) [Seleccionar archivo](#) Ninguno archivo selec.

Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

[Check](#)

Document checking completed. No errors or warnings to show.

Used the HTML parser.

Total execution time 17 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 23.12.19

2.2 blog.html

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for uploaded file **blog.html**

Checker Input

Show ☐ source ☐ outline ☐ image report [Options...](#)

Check by [file upload](#) [Seleccionar archivo](#) Ninguno archivo selec.

Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

[Check](#)

Document checking completed. No errors or warnings to show.

Used the HTML parser.

Total execution time 25 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 23.11.28

2.3 entrada.html

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for uploaded file **entrada.html**

Checker Input

Show ☐ source ☐ outline ☐ image report

Check by Ninguno archivo selec.

Uploaded files with .html or .xht extensions are parsed using the XML parser.

Document checking completed. No errors or warnings to show.

Used the HTML parser.

Total execution time 16 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 23.11.28

2.4 videos.html

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for uploaded file **videos.html**

Checker Input

Show ☐ source ☐ outline ☐ image report

Check by Ninguno archivo selec.

Uploaded files with .html or .xht extensions are parsed using the XML parser.

Document checking completed. No errors or warnings to show.

Used the HTML parser.

Total execution time 10 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 23.11.28

2.5 estilos.css.

 The W3C CSS Validation Service
W3C CSS Validator results for estilos.css (CSS level 3 + SVG)

Jump to: [Warnings \(12\)](#) [Validated CSS](#)

W3C CSS Validator results for **estilos.css** (CSS level 3 + SVG)

Congratulations! No Error Found.

3. Explicaciones de las Páginas

3.1 index.html

- **Doctype:** <!DOCTYPE html> indica al navegador que el tipo de documento es HTML5.
- **html:** <html lang="es"> define el idioma del documento como español, es importante para la **accesibilidad** y los motores de búsqueda.
- **head:** Contiene metadatos y enlaces a hoja de estilo y recursos externos que no se muestran directamente en la ventana del navegador:
 - **meta charset:** Define la codificación de caracteres UTF-8, que incluye la mayoría de los caracteres utilizados en diversos idiomas.
 - **meta viewport:** Garantiza que el sitio web es *responsive* y se ve bien en todos los dispositivos.
 - **title:** El título que aparecerá en la pestaña del navegador. He puesto un título único de “Vital360 - Cuidado Personal, Salud y Bienestar para tu Vida”, ayuda a SEO del sitio.
 - **meta description:** Se trata de una breve descripción de la página, que los motores de búsqueda pueden usar en los resultados de búsqueda.
- **.link:** enlazar el icono de “favicon”, he reducido su tamaño a 32 * 32 píxeles para acelerar la carga del sitio sin sacrificar la calidad visual, garantizando así una experiencia de usuario fluido.
- **link :** Establece una conexión con el servidor de fuentes de Google y enlaza a una hoja de estilos para fuentes específicas de Google, puede mejorar la velocidad de carga de la página.
- **link:** Enlaza el archivo CSS externo que define los estilos del sitio.
- **body:** contiene el contenido del sitio web, está compuesto por tres elementos : <header>, <main> y <footer>:
 - **header:** Define el encabezado de la página, incluyendo el menú de navegación t el logo. Es una etiqueta semántica para mejorar la **accesibilidad** y el SEO del sitio.

- **<div class =“hero-container”>**: es un contenedor genérico, no aporta significado semántico al contenido, es puramente estructural, lo he utilizado para envolver todo el contenido de la página. Facilita la estilización mediante CSS.
 - **<a>**: es una etiqueta de anclaje que crea un hipervínculo a la imagen de logo, el atributo “href” especifica el destino del enlace, el atributo “title” proporciona información adicional sobre un elemento cuando el cursor del ratón se sitúa sobre el elemento, mejora la **accesibilidad**, también he añadido un atributo de clase para poner estilos a los enlaces.
 - ****: es la etiqueta de imagen, he puesto el atributo “alt” para la **accesibilidad** de web, ya que los usuarios que dependen de lectores de pantalla escucharán este texto como una descripción de lo que hay en la imagen, además, si la imagen no se carga debido a un error de carga, el texto alternativo se mostrará en su lugar. Aparte, también ayuda a SEO.
 - **<nav>** : es la barra de navegación que contiene una lista de enlaces. Es una etiqueta semántica para mejorar la **accesibilidad** y el SEO del sitio.
 - **ul-li**: define una lista desordenada en HTML, representa un elemento de lista individual dentro de lista desordenada . Esta lista se está utilizando como un contenedor para los elementos de navegación de la página web.
 - **<a>**: es la etiqueta para crear enlaces del menú de navegación.
- **main**: Contiene el contenido principal del sitio, contiene 3 secciones, la sección 1 es el espacio destacado con imagen y el lema, la sección 2 es el formulario de suscripción al boletín de noticias y la última sección son cuatro tarjetas con temas diferentes.
 - **<section>** : para estructurar el <main>, he utilizado 3 etiquetas de <section> para agrupar contenido temáticamente relacionado. Es una etiqueta que mejora la **accesibilidad**.
 - Aquí abajo son los elementos de la primera sección:
 - **<div class= “container hero-image”>** : aquí he puesto dos clases al contenedor genérico “div” para crear distintos estilos en CSS. La clase “container” sugiere un estilo específico que limitará el estilo del contenedor y la

clase “hero-image” estaría relacionada con estilos para una imagen destacada.

- **<h1>......</h1>**: la etiqueta **<h1>** define el encabezado más importante de la página. He puesto dos elementos de “span” con clases que sugiere controlar el comportamiento del texto en diferentes tamaños de pantalla (escritorio y móvil).
- Aquí abajo son los elementos de la segunda sección:
 - **<div class=“contenido” >** : este contenedor genérico controla el comportamiento de la clase de “contenido”.
 - **<h2>**: Un encabezado de segundo nivel que introduce el contenido de esta sección.
 - **<div class=“two-col”>div class=“intro”>...</div><form>...</form></div>**: el contenedor genérico con clase “two-col” divide el contenido en dos columnas, una columna para introducir el contenedor con clase de “intro” y otra columna para el formulario.
 - **<p > y **: párrafo con el contenido la información sobre Vital360 y texto en negrita (****).
 - **<form action=“#” method=“post” class=“intro”>**: Un formulario con el método POST para enviar datos. La acción # indica que no hay un destino específico definido para el envío del formulario.
 - Dentro del formulario, hay una lista “**ul**” con dos elementos de lista “**li**”, cada uno conteniendo una etiqueta **<label>** y un campo de entrada **input>** . Estos campos son para ingresar el nombre y el correo electrónico. La etiqueta **<label>** está asociada con el campo de entrada **<input>** a través del tributo “for”. El valor de “for” coincide con el valor del atributo “id” de la etiqueta **<input>** correspondiente.
 - **<input type=... id=... name=... placeholder=... required>** :
 - **type=“text” / “email” / “checkbox”**: el campo está configurado para recibir texto, correo o verificación respectivamente.

- **id** : asegura una asociación adecuada con su `<label>`
- **name**: es importante tanto en el *client-side* como en el *server-side*, le dice al navegador qué nombre asignar a cada dato, y en el servidor, permite al servidor manejar cada dato por su nombre.
- **placeholder**: proporciona un ejemplo que se espera en el campo
- **required**: indica que el campo es obligatorio
- **``** : "<mailto:info@vital360.com>" es un esquema especial de URL que crea un enlace para enviar un correo electrónico. Al hacer clic en este enlace, el cliente de correo electrónico predeterminado del usuario se abrirá para enviar un correo a la dirección especificada, que en este caso es "info@vital360.com".
- **``** : he utilizado la etiqueta `` para aplicar estilos personalizados al checkbox.
- **`<button type="submit">`** : un botón para enviar el formulario.
- Aquí abajo son los elementos de la tercera sección: he utilizado una `` de cuatro `` para maquetar las 4 tarjetas, en cada `` tiene dos partes `<figure class="right">` y `<div class="card-body">`:
 - **`<figure><figcaption></figure>`**: Se usa para encapsular una imagen y su correspondiente *credit*. `` se utiliza para enlazar la imagen de la tarjeta, y `<figcaption>` define una leyenda para el contenido del elemento `<figure>`, en este caso, se atribuye la fuente de la imagen.
 - **`<div class="card-body">`**: es un contenedor genérico que contiene el cuerpo de contenido de la tarjeta, concretamente, contiene la etiqueta `<h3>`, el párrafo `<p>` de breve introducción y el botón de "me interesa".
- **footer**: en el `<footer>`, he puesto una etiqueta de `<div class="container">` para envolver los dos bloques de contenido, el primer bloque es logo y copyright, y el segundo bloque es menú de redes sociales y páginas legales.

- `<p>2023-2024HUIMIN TAO</p>`: para insertar una barra vertical con CSS, he puesto la etiqueta ``.
- Para el segundo bloque, he utilizado dos `` para contener las tres `` de redes sociales y tres `` de las páginas legales.
- `` : para este enlace externo, he utilizado el atributo `target="_blank"` para que el enlace se abra en una nueva pestaña. El atributo `title` proporciona información adicional sobre este enlace cuando el cursor se sitúa sobre él.
- `<i lang="en">cookies</i>` : la etiqueta `"i"` se muestra en cursiva, y especifica que el lenguaje del contenido en inglés mediante el atributo `lang="en"`. Esto es útil para la accesibilidad, ya que ayuda a los lectores de pantalla y otras tecnologías asistivas a manejar correctamente el cambio de idioma.

3.2 blog.html

En la parte de "Document Type Declaration", "HTML Element" y "Head Section", he mantenido las mismas etiquetas que en index.html, con la única variación en el contenido de la etiqueta de `<meta name="descripción"...>` y la etiqueta de `<title>` para asegurar su unicidad.

Respecto a las estructuras, he conservado la misma disposición que en index.html, manteniendo los elementos semánticos de `<header>`, `<main>` y `<footer>`. Tanto la parte de `<header>` como la parte de `<footer>` son idénticos a las de index.html. A continuación, se detallan las etiquetas de `<main>` que difieren de index.html:

- La etiqueta `<main>` se divide en 3 partes en total, un `<header>` que sirve como el titular del blog; un `<div class="blog-content">` destinado a albergar los siete artículos y un `<footer>` para incluir los elementos de paginación.
- **<article>**: He usado la etiqueta `<article>` para los siete artículos, ya que este contenido puede separarse del resto de la página sin perder su significado. La etiqueta `<article>` es semántica y mejora la **accesibilidad** y el SEO del sitio. Estructuralmente, cada etiqueta `<article>` alberga dos contenedores `<div>`. La primera de estas divisiones encapsula el contenido principal del artículo, mientras que la segunda se encarga de presentar la imagen relacionada, los créditos de la misma y la categoría a la que pertenece el artículo. Para el primer artículo que se extiende

a lo largo de dos fracciones de la pantalla de desktop, he puesto una **clase de “span-2”** para diferenciarse de los demás artículos.

- **<p>Por Sofía Martín - <time datetime="2023-12-25">25 de diciembre de 2023</time></p>** : He utilizado la etiqueta `<time>` maquetar la fecha de metadatos, de esta manera mejora la **accesibilidad** de la página.

3.3 entrada.html

En la parte de "Document Type Declaration", "HTML Element" y "Head Section", he mantenido las mismas etiquetas que en index.html, con la única variación en el contenido de la etiqueta de `<meta name="descripción"...>` y la etiqueta de `<title>` para asegurar su unicidad.

Respecto a las estructuras, he conservado la misma disposición que en index.html, manteniendo los elementos semánticos de `<header>`, `<main>` y `<footer>`. Tanto la parte de `<header>` como la parte de `<footer>` son idénticos a las de index.html. A continuación, se detallan las etiquetas de `<main>` que difieren de index.html:

- La etiqueta `<main>` se divide en 3 partes en total, un `<header>` que sirve como el menú de las migas de pan; un `<div class="wrapper-barra-lateral contenido">` destinado a albergar el artículo principal (`<article class="article-entrada">`) y el artículo reciente (`<div class="barra-lateral">`) y un `<footer>` para la lista de categorías.
- **<p>Por Daniel López - <time datetime="2023-12-30">30 de diciembre de 2023</time> - Fitness</p>** : He utilizado la etiqueta `<time>` maquetar la fecha de metadatos, de esta manera mejora la **accesibilidad** de la página. Para la palabra “fitness”, lo he metido dentro del `` para aplicar el estilo en CSS y el atributo `lang="en"` indica que el lenguaje del texto es inglés.
- **<h1> / <h2> / <h3> /<h4>**: He utilizado la etiqueta `<h1>` para el título del artículo principal, las etiquetas `<h2>`, para los subtítulos del artículo principal, las etiquetas `<h3>` para el título “Artículos recientes”, que son secciones secundarias y `<h4>` para los títulos propios de los tres artículos recientes.
- **sitio web de la <abbr title="Organización Mundial de la Salud">OMS</abbr>**: La etiqueta `<abbr>` se usa para una abreviatura, y su atributo **"title"** “proporciona el significado completo de la abreviatura cuando el usuario pasa el cursor sobre ella. En este caso, "OMS" es la abreviatura de "Organización Mundial de la Salud".

- **<cite lang="en">The Lancet Psychiatry</cite>** : he utilizado la etiqueta **<cite>** para citar el nombre de una fuente de referencia, en este caso, la revista "The Lancet Psychiatry". El atributo **lang="en"** indica que el idioma de esta fuente es el inglés.

- **<blockquote>**

...

<p><cite class="info-auxiliar">Dra. Ana Márquez, especialista en medicina

Deportiva</cite></p>

</blockquote>. : he usado la etiqueta **<blockquote>** para indicar una sección que es una cita de otra fuente, con el texto de la cita en un **** y la fuente de la cita en un **<cite>** dentro de un **<p>**.

- Para las cuatro **categorías**, he utilizado una lista de **** con los cuatro ****, y cada **** enlaza a la página externa, en este caso es página vacía.

3.4 videos.html

En la parte de "Document Type Declaration", "HTML Element" y "Head Section", he mantenido las mismas etiquetas que en index.html, con la única variación en el contenido de la etiqueta de **<meta name="descripción"...>** y la etiqueta de **<title>** para asegurar su unicidad.

Respecto a las estructuras, he conservado la misma disposición que en index.html, manteniendo los elementos semánticos de **<header>**, **<main>** y **<footer>**. Tanto la parte de **<header>** como la parte de **<footer>** son idénticos a las de index.html. A continuación, se detallan las etiquetas de **<main>** que difieren de index.html:

La etiqueta **<main>** se divide en 2 partes en total: un **<header>** que sirve como el titular de la página y un contenedor **<div class="video-content">** para albergar los 6 videos.

- **<div class = "tarjeta-video">** : actúa como un contenedor principal que encapsula dos secciones internas. La primera, **<div class="tarjeta-description">**, aloja el título (**<h2>**) y un párrafo introductorio breve del video (**<p>**), proporcionando un contexto inicial al usuario. La segunda sección, **<div class="video-fluid">**, está diseñada para contener el video, presentado a través de un elemento **<iframe>**.

- **<iframe>** : Es una etiqueta para incrustar otro documento HTML dentro del documento HTML actual, en este caso se usa para incrustar un video de Youtube.

3.5 estilos.css

(1) Colores y tipografía del tema

Para los colores y tipografía del tema, he utilizado el selector “:root” para definir variables globales.

(2) Box-model alternativo

He usado “*”, “*::before”, “*::after” para aplicar “box-sizing”, “margin” y “padding” a todos los elementos. Esto asegura un comportamiento consistente del box model en las páginas, es crucial para la compatibilidad entre diferentes navegadores.

(3) Estilos generales de tipografía y colores

He aplicado el enfoque "Mobile First" para desarrollar páginas web responsivas, definiendo breakpoints específicos para dispositivos móviles, tabletas y ordenadores de escritorio en 576px, 768px y 992px, respectivamente. En lo que respecta a los estilos de tipografía y colores generales, he empleado selectores para elementos fundamentales como body, html, y los encabezados h1 a h4, además de nav, y figcaption. También he personalizado estilos específicos para botones, pies de foto, textos auxiliares, migas de pan, así como para las secciones de encabezado (header) y pie de página (footer). Para los elementos , <a> y <abbr>, he optado por eliminar los estilos predeterminados, buscando así un diseño más limpio y consistente.

(4) Reglas del .container y .contenido

Este código CSS se centra en hacer que un contenedor con la clase .container sea adaptable a diferentes tamaños de pantalla. Inicialmente, .container tiene un ancho del 100% con un *padding* a ambos lados de 15px, lo que lo hace óptimo para dispositivos móviles. En pantallas más grandes, específicamente a partir de 992px de ancho (como en desktops), se ajusta el *padding* del .container a 40px a cada lado para dar más espacio. Además, se define un estilo para los elementos con la clase .contenido dentro del .container, limitando su ancho máximo a 960px y centrando estos contenidos horizontalmente. Esto evita que el contenido se extienda demasiado en pantallas grandes, manteniendo así la legibilidad.

(5) Header y menú de navegación

Esta parte de CSS define el estilo de un encabezado (header) y un menú de navegación (navbar). En “header .container”, se utiliza *Flexbox* para distribuir los elementos de manera “space-between” horizontalmente y alinearlos al centro verticalmente.

El header con la clase .header está diseñado para ser fijo (sticky) en la parte superior de la pantalla y tiene un sombreado sutil para resaltar su presencia. La elección de utilizar “sticky” en lugar de “fixed” para el posicionamiento fijo del elemento se debe a su capacidad de mantenerse dentro del flujo normal de la página, evitando así cualquier interferencia con el posicionamiento de otros elementos. También he puesto “z-index” muy alto para que se muestre siempre encima de todos los restos elementos.

El menú de navegación, representado por “.navbar ul”, también usa *Flexbox* para alinear sus elementos horizontalmente y a la derecha. Además, se añade una barra vertical (" | ") después del primer elemento de la lista para separar visualmente las opciones del menú.

Por último, se encuentra una media query diseñada específicamente para pantallas de escritorio. En este punto, se realiza un ajuste en la posición del elemento .navbar, desplazándolo 40px hacia la derecha. Esto se debe a que en la vista de escritorio se han definido márgenes más amplios en comparación con dispositivos móviles. Para garantizar la alineación adecuada del menú con el resto de los elementos, es necesario realizar esta modificación en su posición.

(6) Footer

Esta parte de CSS se encarga de estilizar el pie de página (footer) de un sitio web. Inicialmente, se utiliza *Flexbox* para organizar los contenidos en columnas, centrando el contenido verticalmente con espaciado entre elementos (gap). También se establecen espaciados superior e inferior (padding-top y padding-bottom).

Se definen estilos específicos para dos bloques de contenido dentro del pie de página, como “.footer-bloque-1” (logo y copyright), “.footer-bloque-2” (menú de redes sociales y páginas legales) , y .legal (las páginas legales). Se utiliza *flex-direction* para ajustar la dirección de los elementos y *align-items* para alinearlos verticalmente. Además, se agrega un separador vertical entre los primeros elementos de .footer-bloque-1 (derechos de autor).

En una *media query* para pantallas de *desktop*, se realiza un ajuste en la disposición del contenido del pie de página. “.container” del footer cambia a una disposición horizontal (*flex-direction: row*) y se ajustan los estilos para alinear los elementos de manera diferente dentro de

los bloques “.footer-bloque-1”, “.footer-bloque-2” y “.legal”. También., Se añaden separadores verticales en la lista “.legal”entre elementos, excepto el último.

(7) Reglas de enlaces

Esta parte de css define el estilo de varios tipos de enlaces para todas las páginas:

- Enlaces en la barra de navegación (navbar) tienen colores de texto y cambio de color al pasar el cursor (hover).
- Enlaces en el pie de página (footer) siguen una paleta de colores diferente, con cambio de color al pasar el cursor.
- Imágenes en enlaces dentro de un menú social cambian ligeramente su opacidad al pasar el cursor.
- Enlaces con la clase “.enlace-texto” tienen un efecto de subrayado punteado y cambio de color al pasar el cursor.
- Botones (con la clase “.button”) cambian su color de fondo al pasar el cursor.
- Enlaces con clases específicas (“titular-entrada”, “breadcrumbs”, “category-list ul li a”) cambian de color al pasar el cursor.
- “.category”: Estos enlaces se muestran como bloques en línea con colores de fondo y texto. Al pasar el cursor, el color de fondo cambia.
- “.pagination”: Establece que el elemento de paginación debe mostrarse como un bloque en línea, lo que significa que estará en la misma línea que otros elementos en su contenedor. “.pagination a” establece el estilo de los enlaces dentro de la paginación. “.paginación a:active, .pagination:hover” define el estilo de los enlaces dentro de una paginación cuando se activan o cuando se pasa el cursor por encima.
- “.category-list ul li”: se cambia tanto el color de fondo como el color de texto al pasar el cursor.

(8) Reglas de la página de index.html

Esta parte de código CSS define el estilo particular de la página “index.html”:

- “.hero-image”: Define la imagen de fondo, su tamaño y posición, y ajusta el contenido verticalmente al centro. He establecido también que el hero-image ocupe 100% de anchura de

Su contenedor y la altura como 50% de la altura de la ventana del navegador, lo que la hace ajustable en diferentes tamaños de pantalla. He utilizado también el Flexbox para centrar verticalmente el lema superpuesto.

- ".hero-image h1": He agregado una sombra al texto para mejorar la legibilidad.
- ".hero-image h1 span.break-mobile": aquí, he puesto un "display:block" para asegurarme de que una parte del lema se desplace a la línea siguiente en la vista de dispositivos móviles.
- "label": he puesto "display: block" para que los elementos de <label> se muestren en una línea separada.
- input[type="text"], input[type="email"]: Establece el estilo de los campos de entrada de texto y correo, incluyendo el diseño de su "placeholder" utilizando "::placeholder".
- ". custom-checkbox", ".checkbox-style::before", ".custom-checkbox:checked + .checkbox-style::before" : estos códigos crea un estilo personalizado para casillas de verificación. Oculta la casilla de verificación estándar, muestra un cuadro personalizado en su lugar y agregar un marca de verificación cuando la casilla está marcada.
- ".button": define el estilo para botones con clase de ".button", se muestran en línea, tienen un fondo de color, un espacio superior, un relleno interno, una altura fija, sin borde y un cursor interactivo.
- ".card-img-top": Establece que la imagen contenida en estos elementos debe ocupar el 100% del ancho disponible, mantener su relación de aspecto (altura automática) y ajustarse para cubrir completamente el contenedor, lo que significa que la imagen se recorta si es necesario para llenar el espacio sin deformarse.
- Para tabletas (768px o más), se realiza lo siguiente:
 - El texto dentro del elemento h1 con la clase "break-mobile" se muestra en línea .
 - Los elementos con la clase "contenido" se muestran como una cuadrícula de 4 filas.
 - El contenido dentro de las tarjetas se alinea verticalmente al centro.
 - Las tarjetas se muestran en fila, con espacio entre ellas

- El contenido con la clase de “.right” de las tarjetas se reordenan a la derecha de las tarjetas y rotan ligeramente, mientras que el contenido con la clase de “.left” se reordenan a la izquierda de las tarjetas y rotan ligeramente también.
- Las imágenes en las tarjetas tienen un ancho máximo de 500px para que no se toque las paredes laterales en caso de rotación.
- Para escritorios (992px o más), se realiza lo siguiente:
 - El lema tiene el mismo ancho que el resto contenido con clase de “.contenido”, ya que se tiene que alinear con los restos de los elementos de la página.
 - El elemento span dentro de h1 con la clase "break-desktop" se muestra en bloque.
 - Los elementos con clase de “.two-col” se muestran en una disposición de grid. Esto crea una disposición de dos columnas en la que los elementos se distribuyen uniformemente en el espacio disponible, con un espacio de separación de 50px entre ellas.

(9) Reglas de la página de blog.html

Esta parte de códigos CSS proporciona reglas específicas de la página de blog.html:

- “.titular-blog”: Define márgenes superiores e inferiores para los titulares del blog.
- “.meta p”: Establece un margen superior para los párrafos dentro de la clase “meta”.
- “.tarjeta-contenido h2”: Aplica un margen inferior a los encabezados h2 dentro de las tarjetas de contenido.
- “.tarjeta-contenido a.enlace-texto”: Establece que los enlaces de texto dentro de las tarjetas de contenido sean bloques en línea y define un margen superior.
- “.blog-content article”: Establece un estilo para los artículos dentro de “.blog-content”, con bordes, márgenes y un diseño de Flexbox.
- “.tarjeta-contenido”: Define el orden y el relleno para las tarjetas de contenido.
- “.tarjeta-imagen”: En la clase “.tarjeta-imagen”, he configurado la posición como relativa. Esto es importante ya que permite que cualquier elemento interno pueda ser posicionado de manera absoluta en relación con la tarjeta. Para asegurar que la tarjeta ocupe el 100% del ancho de su contenedor padre, he establecido su ancho al 100%. Además, con el objetivo de mantener

consistentemente la relación de aspecto de la imagen en 1000:667, he empleado un truco: estableciendo la altura (height) a 0 y aplicando un relleno superior (padding-top) del 66.7%.

- “tarjeta-imagen .caption” / “tarjeta-imagen .category” : he aplicado estilos con posicionamiento absoluto en relación con la “tarjeta-imagen” a las leyendas y categorías.
- “.tarjeta-[1-7]”: Asigna imágenes de fondo específicas a diferentes tarjetas.
- “.tarjeta-contenido .enlace-texto::after”: Añade un contenido de “>” al enlace de “leer más”.
- “.pagination”: define el estilo de la paginación. Usa Flexbox para un diseño flexible y centrado, con un espacio de entre elementos. Añade un relleno superior y un relleno inferior.
- “.pagination a”: Aplica un relleno vertical y horizontal a los enlaces dentro de la paginación.
- Para tabletas (768px o más), se realiza lo siguiente:
 - El elemento con clase de “.blog-content” se muestra en una grid de dos columnas y cuatro filas, con un espacio de entre ellas. Los artículos <article> tienen un margen inferior.
 - Para desktops (992px o más), se realiza lo siguiente:
la cuadrícula cambia a cuatro columnas y dos filas. Además, se introduce una clase “.span-2” que permite al elemento abarcar dos columnas.

(10) Reglas de la página de entrada.html

Esta parte de códigos define los estilos propios para la página de “entrada.html”:

- “.migas-de-pan”: Establece márgenes superior e inferior.
- “.titular-article”: Usa un diseño de grid con dos filas, el tamaño de cada fila se determina automáticamente en función del tamaño del contenido. He añadido también un borde medio transparente y un margen inferior.
- “.titular-article img”: Las imágenes dentro de la clase “titular-articule” se muestran en bloque, ocupan 100% del ancho manteniendo su proporción y se recortan para cubrir el área del contenedor.
- “.titular-article .titular-entrada”: este subelemento de “titular-article” tiene un fondo de color y un relleno para el texto.

- `"span.color-azul"` : he cambiado el color del texto con este selector.
- `"p.intro-article, blockquote .quote" / ".article-entrada p:not(.intro-article):not(blockquote .quote),.article-entrada ul, .article-entrada ol"` : para conseguir el efecto de *mockup*, he puesto tamaño más grande para la introducción de artículo ("`p.intro-article`") y las citas ("`blockquote .quote`") , pero para el resto de texto lo he puesto un menor tamaño de fuente para diferenciar claramente estos elementos.
- `"ol li, ul.lista-inherit li"` : Para las listas en este diseño, inicialmente configuré el `"list-style-type"` de todas ellas en `"none"` para eliminar los estilos predeterminados. Sin embargo, siguiendo las especificaciones del *mockup*, es necesario que ciertas listas retomen su estilo predeterminado. Por esta razón, he asignado el valor `"inherit"` a la propiedad `"list-style-type"` en los elementos `"ol li"` y `"ul.lista-inherit li"`. Esto garantiza que estos elementos de lista hereden el estilo de lista (como viñetas o números) de su elemento contenedor padre, en lugar de utilizar el estilo `"none"` previamente establecido.
- `". articulo-reciente .wrapper"`: Se establece un diseño de grid con tres filas de igual tamaño con un espacio de separación entre las filas.
- `".wrapper li"`: Aplica un borde de 1px sólido con color y opacidad específicos a los elementos `` dentro de cualquier contenedor con la clase `.wrapper.`, en este caso, se trata de las tres tarjetas de artículos recientes.
- `".tarjeta-imagen .category"`: He aplicado el estilo para los elementos con la clase de `".category"` dentro de contenedores `".tarjeta-imagen"`. Los posiciona absolutamente en la esquina superior izquierda y reduce el tamaño de la fuente.
- `".category-list ul li"` : he estilizado los elementos `` dentro de una `` en `.category-list`. Los muestra en línea, con un borde, un margen superior, relleno y el cursor cambia a puntero.
- `".category-list ul"`: he definido un diseño de Flexbox para `` en la clase de `".category-list"`, con elementos en fila que se envuelven según sea necesario y un espacio entre elementos.
- Para tabletas (768px o más) , se realiza lo siguiente:
 - Para los elementos con clases `".articulo-reciente .wrapper"`, he establecido un diseño de grid con tres columnas de igual tamaño (1fr cada una) y una sola fila.
- Para desktops (992px o más), se realiza lo siguiente:

- Para elementos con la clase “.wrapper-barra-lateral”, activa un diseño de grid con dos columnas, siendo una dos veces más ancha que la otra (2fr y 1fr), y establece un espacio de separación de entre ellas.

(11) Reglas de la página de videos.html

Esta parte de códigos define los estilos específicos de la página de videos.html:

- “.video-content”: Aplica un relleno inferior. Para pantallas de tabletas, se convierte en una cuadrícula de dos columnas, y para pantallas de desktop, en una cuadrícula de tres columnas, con un espacio de entre ellas.
- “.video-content p”: Establece márgenes superior e inferior para los párrafos dentro del contenido de video.
- “.tarjeta-description”: Aplica relleno a izquierda y derecha, y un relleno superior para los títulos h2 dentro de la tarjeta.
- “.tarjeta-video”: He utilizado Flexbox con disposición en columna, sin envolver. He aplicado un borde semitransparente y he añadido un margen inferior.
- “.video-fluid”: He establecido la posición relativa con contenido oculto que desborde. He configurado su “orden” para ajustar su disposición. Para asegurar que el video ocupe el 100% del ancho de su contenedor padre, he establecido su ancho al 100%. Además, con el objetivo de mantener consistentemente la relación del video en 560:315, he empleado un truco: estableciendo la altura (height) a 0 y aplicando un relleno superior (padding-top) del 56.25%.
- “.video-fluid-item”: he configurado la posición como absoluta, asegurando que se sitúe exactamente en la parte superior de su contenedor. Permite que el elemento ocupe el 100% del espacio disponible, extendiéndose de borde a borde, lo que garantiza una cobertura total sin dejar márgenes o bordes visibles.