DISEÑO DE INTERFACES WEB UD-2

USO DE ESTILOS

ÍNDICE

[VENTAJAS DEL USO DE CSS 4](#_Toc148079636)

[PREPROCESADORES EN ESTILOS 4](#_Toc148079637)

[DEFINICIÓN DE CSS 4](#_Toc148079638)

[HOJAS DE ESTILO EN FICHERO EXTERNO 5](#_Toc148079639)

[PRIORIDAD EN CSS 5](#_Toc148079640)

[ESTILO CON @import url() 5](#_Toc148079641)

[ETIQUETA !important EN LOS ESTILOS 6](#_Toc148079642)

[PROPIEDADES DEL NAVEGADOR EN CSS 6](#_Toc148079643)

[ESTILOS DE USUARIO CON ETIQUETA !important 6](#_Toc148079644)

[PREFERENCIA EN CASCADA 6](#_Toc148079645)

[REGLAS CSS 7](#_Toc148079646)

[A. SELECTORES ETIQUETAS 7](#_Toc148079647)

[B. SELECTORES CLASES 7](#_Toc148079648)

[C. SELECTORES IDENTIFICADORES 8](#_Toc148079649)

[AGRUPACIÓN Y ANIDAMIENTO DE SELECTORES 8](#_Toc148079650)

[SELECTOR UNIVERSAL 9](#_Toc148079651)

[SELECTOR DE DESCENDIENTES 9](#_Toc148079652)

[SELECTOR DE HIJOS 10](#_Toc148079653)

[SELECTOR POR ATRIBUTOS 10](#_Toc148079654)

[SELECTOR POR ATRIBUTOS CON VALOR CONCRETO 10](#_Toc148079655)

[SELECTOR POR ATRIBUTOS CON AL MENOS EL VALOR INDICADO ~=”VALOR” 11](#_Toc148079656)

[SELECTOR POR LOCALIZACIÓN DE PÁGINA (lang) 11](#_Toc148079657)

[HERENCIA EN CSS 12](#_Toc148079658)

[BUENAS PRÁCTICAS AL ESCRIBIR CSS 12](#_Toc148079659)

[ATRIBUTOS. MODELO DE CAJAS 13](#_Toc148079660)

[ELEMENTOS: TEXTOS 14](#_Toc148079661)

[ELEMENTOS: FONDO Y TABLAS 15](#_Toc148079662)

[ELEMENTOS: VISIBILIDAD 15](#_Toc148079663)

[ELEMENTOS: LISTAS 15](#_Toc148079664)

[PSEUDO-CLASES Y PSEUDO-ELEMENTOS 16](#_Toc148079665)

[ESPECIFICIDAD EN CASO DE CONFLICTOS 16](#_Toc148079666)

[POSICIONAMIENTO 16](#_Toc148079667)

[A. TIPOS DE ELEMENTOS (SEGÚN DISPLAY) 16](#_Toc148079668)

[B. POSITION 17](#_Toc148079669)

[C. POSICIONAMIENTO FLOTANTE 17](#_Toc148079670)

[D. SOLAPAMIENTO DE ELEMENTOS (Z-INDEX) 18](#_Toc148079671)

[FORMULARIOS 18](#_Toc148079672)

[A. MOSTRAR UN BOTÓN COMO ENLACE 18](#_Toc148079673)

[B. MEJORAS EN LOS CAMPOS DE TEXTO 19](#_Toc148079674)

[C. LABELS ALINEADAS 19](#_Toc148079675)

[D. LABELS ALINEADAS EN LA MISMA LÍNEA 20](#_Toc148079676)

[PRECEDENCIA DE ESTILOS 20](#_Toc148079677)

[DEFINICIÓN DE HOJAS DE ESTILO 21](#_Toc148079678)

[HERRAMIENTAS Y TEST DE VERIFICACIÓN 21](#_Toc148079679)

# INTRODUCCIÓN

CSS nos sirve para hacer estilados en un HTML sin necesidad de utilizar marcas de estilo como <b>, <em>, etc. facilitando el diseño web.

## VENTAJAS DEL USO DE CSS

* Separación entre contenido y presentación.
* Uso en múltiples páginas del mismo archivo.
* Ahorro de espacio en memoria.
* Separación de trabajos de diseño y contenido. Especialización.
* Legibilidad a los motores de búsqueda.
* Adaptación a otros formatos de dispositivos.
* Adaptación para accesibilidad.
* Posibilidad de funcionalidades más potentes.

## PREPROCESADORES EN ESTILOS

Podemos utilizar otros lenguajes que nos den más funcionalidad que CSS, por ejemplo, SASS, que permite usar variables. Estos ficheros se deben convertir posteriormente a CSS.

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Estos preprocesadores permiten crear un estilo de lenguaje tipo script que es mucho más potente que el CSS tipo, aun así esto en proyectos pequeños o medianos no es útil, pero en proyectos grandes sí que facilita muchísimo el trabajo e incluso lo hace más legible en esas situaciones.

# DEFINICIÓN DE CSS

Definición de CSS:

* Mediante atributo style de HTML
* Etiqueta <style> dentro de <head>
* En un fichero externo .css

Varias formas de definir estilos:

* Usando el atributo style dentro de una etiqueta.
* Ej: <p style=“background: black;”>párrafo</p>
* Presenta alta prioridad frente a otras reglas. Sin embargo es una desventaja porque
* Imposibilita la reutilización
* Dificulta el mantenimiento CSS.
* Etiqueta <style> dentro de la cabecera <head>. Esta etiqueta tiene varios atributos opcionales:
* **Type (requerido)**: para indicar que se aplica un estilo CSS
* **Media**: para indicar el tipo de dispositivo.
* **Title**: nombre que se le da al estilo

# HOJAS DE ESTILO EN FICHERO EXTERNO

La ventaja de utilizar hojas de estilo externas es que se pueden reutilizar en varios documentos.

Puede ser enlazada a un documento mediante la etiqueta <link> que se coloca en la cabecera.

El fichero css debe contener únicamente reglas de estilo.

# PRIORIDAD EN CSS

El estilo que se aplica es el último que se lee, por tanto el que está en la propia etiqueta tiene alta prioridad por ser el último que se lee.

Si tenemos un estilo en cabecera y con la etiqueta <sytle> sobre una propiedad y otro estilo en un fichero .css sobre la misma propiedad, dependerá de la posición de las líneas de código en la cabecera, la última que se lea será la que se aplique (si no hay estilo en la propia etiqueta).

Texto

Descripción generada automáticamente Texto

Descripción generada automáticamente

# ESTILO CON @import url()

Otra forma de conectar el .css con el .html es a través de @import url(“archivo.css) dentro de la etiqueta <style> de cabecera:

Texto

Descripción generada automáticamente

# ETIQUETA !important EN LOS ESTILOS

Si queremos que un estilo permanezca sobre otros de mayor prioridad, le ponemos !important en su definición:

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

Texto

Descripción generada automáticamente

# PROPIEDADES DEL NAVEGADOR EN CSS

Los atributos que no indicamos en nuestras hojas de estilo, se toman de los valores por defecto del navegador.

# ESTILOS DE USUARIO CON ETIQUETA !important

Algunos navegadores pemiten, mediante extensiones, cargar estilos de usuario, y si estos tienen !important, son los más prioritarios de todos, por facilitar estilos de accesibilidad por ejemplo.

# PREFERENCIA EN CASCADA

Diagrama

Descripción generada automáticamente

# REGLAS CSS

Las normas de estilo se definen con reglas, que están formadas por un selector y una declaración:

Diagrama

Descripción generada automáticamente

Los selectores pueden ser:

* Etiquetas
* Clases
* Identificadores

## SELECTORES ETIQUETAS

Usando las propias etiquetas HTML como selectores.

* Selector { atributo: valor}
* El atributo hace referencia a la característica que se quiere modificar de la etiqueta (p.e. color).
* El valor hace referencia a la instancia del atributo.
* Por ejemplo: h1 {color:blue}
* Se pueden definir atributos para varios selectores y selectores con varios atributos.
* Mediante las clases se pueden definir estilos abstractos que no estén asociados a una etiqueta HTML.
* Permiten aplicar estilos a etiquetas HTML, con el mismo efecto que usando selectores de etiquetas, pero también a cualquier otro elemento de la página.
* Hay diferentes tipos de clases:
  + Asociadas directamente a una etiqueta HTML
  + Genéricas, que se pueden aplicar a cualquier etiqueta.

## SELECTORES CLASES

* Nombreetiqueta.nombreclase {atributo:valor}

Ejemplos:

* + - h1.verde {color:green}
    - h1.azul {color:blue}
* Para indicar que estilo aplicar a cada etiqueta:
  + - <h1 class=“nombreclase”>texto</h1>

Las clases más genéricas no se aplican a ninguna etiqueta HTML por lo que en su descripción no se especifica el nombre de ninguna etiqueta.

* .nombreclase {atributo:valor}
* .verde {color:green}

Una clase así definida se puede aplicar a cualquier elemento de la página.

## SELECTORES IDENTIFICADORES

Los selectores basados en identificadores son muy similares a los basados en clases y con una sintaxis muy parecida.

La diferencia es que los identificadores solo se pueden usar en un único elemento

* Nombreetiqueta#nombreidentificador {atributo:valor}
* #nombreidentificador {atributo:valor}
* #verde {color:green}

<p id=“verde”>Párrafo en color verde</p>

Los identificadores se suelen usar cuando identificadores como cabecera, contenidos y piepagina definen el estilo de tres zonas de una página web.

No tiene sentido que esos identificadores se repitan en varios elementos.

# AGRUPACIÓN Y ANIDAMIENTO DE SELECTORES

Los selectores se pueden agrupar y anidar para conseguir estilos CSS.

* **Agrupamientos**

Hace referencia a la manera en la que se pueden escribir las reglas de estilo para conseguir un CSS más claro y fácil de entender.

Se puede aplicar el mismo estilo a un conjunto de selectores al mismo tiempo.

Selector1, selector2 {atributo1:valor1; atributo2:valor2}

* Los selectores se pueden **anidar** con el fin de conseguir estilos más concretos y definidos.

En CSS se llaman selectores contextuales.

* **Selector anidado común**: se usa para crear reglas sobre elementos que están rodeados de otros elementos.

Ej: h1 i b {color:blue}

* **Anidamiento de selectores hijos**: si lo que se desea es restringir que las etiquetas, además de estar en el mismo contexto, estén seguidas unas de otras

Ej: h1>b {color:blue}

* **Anidamiento de selectores adyacentes**: se usa cuando se quiere aplicar un estilo a un elemento que tiene adyacente a otro elemento en el mismo nivel de anidamiento en HTML.

Texto, Carta

Descripción generada automáticamente

* **Anidamiento de selectores hermanos**: se usa cuando se quiere aplicar un estilo a un elemento que tiene como hermano a otro elemento HTML.

Forma

Descripción generada automáticamente

# SELECTOR UNIVERSAL

Con el “\*” representamos a todo del documento html.

Se utiliza para aplicar propiedades “no heredables” a todos los elementos. Por ejemplo, para resetear algunas propiedades:



# SELECTOR DE DESCENDIENTES

El selector tiene dos etiquetas, la que queremos aplicarle el estilo, y la que la contiene, de prefijo. Por ejemplo:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación, Sitio web

Descripción generada automáticamenteTexto

Descripción generada automáticamente con confianza baja

# SELECTOR DE HIJOS

El selector tiene dos etiquetas, la que queremos aplicarle el estilo, y la etiqueta padre, separados por un “>” Por ejemplo:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteTexto

Descripción generada automáticamente con confianza baja

# SELECTOR POR ATRIBUTOS

El selector incluye el atributo entre [] que debe tener la etiqueta para que se aplique el estilo indicado.

Texto

Descripción generada automáticamente

Pantalla de un reloj

Descripción generada automáticamente con confianza media

Texto

Descripción generada automáticamente

# SELECTOR POR ATRIBUTOS CON VALOR CONCRETO

Se puede acotar más la regla indicando el valor que debe tener el atributo para aplicar el estilo:

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

# SELECTOR POR ATRIBUTOS CON AL MENOS EL VALOR INDICADO ~=”VALOR”

Texto

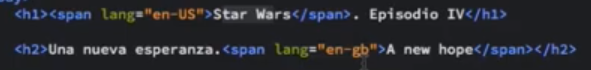
Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación, Sitio web

Descripción generada automáticamenteTexto

Descripción generada automáticamente

# SELECTOR POR LOCALIZACIÓN DE PÁGINA (lang)

Aplica el estilo a las etiquetas marcadas con el idioma indicado en el atributo “lang”.

Interfaz de usuario gráfica

Descripción generada automáticamenteImagen que contiene botella, firmar

Descripción generada automáticamente

# HERENCIA EN CSS

Los estilos se heredan en etiquetas anidadas, pero no todas las propiedades se heredan:

* font-size -> sí
* colores de los bordes, márgenes, color de fondo -> no suelen ser heredables.

Por tanto, se recomienda usar de selector de la regla al elemento superior que agrupe a todos sobre los que queremos aplicar el estilo, por ejemplo body, y los que queramos excluir, le podemos aplicar un estilo particular en su etiqueta.

# BUENAS PRÁCTICAS AL ESCRIBIR CSS

Recomendaciones a la hora de escribir CSS para que las modificaciones posteriores seán más fáciles. No están definidas en el estándar W3C, pero ayudan en el desarrollo.

* Lo selectores se nombran en minúsculas, nunca empezando por caracteres especiales o numéricos.
* El nombre de los selectores debe ser específico y claro, para que tenga una mayor capacidad expresiva.
* El nombre de las clases e identificadores no debe describir una característica visual, como color, tamaño o posición. Si especificamos un nombre definiendo un color y cambia el color de la clase también se debería cambiar el nombre del selector.
* Los nombres deben seguir más una visión semántica que estructural.
* Comentarios: /\*texto del comentario\*/
* Separa las palabras mediante guiones o mayúsculas.
* No hacer uso excesivo de clases: utilizar en su lugar selectores contextuales o anidados.
* Agrupar estilos según selector siempre que sea posible:
  + table
  + table.empleados
  + Etc
* Al principio de un CSS es aconsejable definir los selectores de etiquetas.

# ATRIBUTOS. MODELO DE CAJAS

Diagrama, Esquemático

Descripción generada automáticamente

* Unidades de medida:
  + Pulgadas
  + Centímetros
  + Milímetros
  + Puntos
  + Picas
* Atributos de posición:
  + Top
  + Left
  + Right
  + Bottom
* Atributos margin:
  + Margin-left
  + Margin-right
  + Margin-top
  + Margin-bottom
* Atributos padding: se puede traducir como relleno. Indica la distancia entre el borde y los elementos que se encuentran en el interior.
  + Padding-top
  + Padding-right
  + Padding-bottom
  + Padding-left
* Atributos border: definen el estilo y color del borde de la caja <https://www.w3schools.com/css/css_border.asp>
* Border-top
* Border-bottom
* Border-right
* Border-left

Algunos de las palabras claves son:

* None
* Dotted
* Dashed
* Solid
* Double
* Groove
* Ridge
* Inset
* Outset
* Border-radius: se usa para realizar bordes redondeados.

Texto, Pizarra

Descripción generada automáticamente

# ELEMENTOS: TEXTOS

Una vez vistos los atributos asociados a las clases estos son los atributos relacionados con la apariencia de textos, listas, tablas, enlaces e imágenes.

* Fuentes:
  + color: nombres en inglés y valores RGB.
  + font-size: unidades | xx-small | x-small | small | medium | large | x-large | xx-large
  + font-family. Serif | sans-serif | cursive | fantasy | monospace. Indica la tipografia del texto.
  + font-weight: normal | bold | bolder | lighter | 100 | 200 | … | 900. Para definir la anchura de los caracteres (efecto negrita) Normal=400, bold=700
  + font-style: normal | italic | oblique. Estilo de la fuente. Oblique es similar a italic.
* Párrafos:
  + line-height: normal | unidades. El alto de una línea y por tanto el espaciado entre líneas.
  + text-decoration: none | underline | overline | line-through. Para establecer la decoración de un texto, si está subrayado, sobre-rayado o tachado.
  + text-align:alineación del texto.
  + text-indent: Establece tabulación primera línea.
  + text-transform: capitalize | uppercase | lowercase | none text-transform | none. Permite transformar el texto haciendo que tenga la primera letra en mayúsuculas, de todas las palabras, todo en mayúsculas o minúsculas.

# ELEMENTOS: FONDO Y TABLAS

* Fondo:
  + background-color.
  + background-image.
  + background-repeat
* Tablas:
  + caption-side: top | bottom. Posición del título.
  + table-layout: auto | fixed.
  + border-collapse: collapse | separate. Selección del modelo de los bordes.
  + border-spacing: unidades. Espaciado entre los bordes de celdas adyacentes.
  + empty-cells: show | hide. Visibilidad de los bordes de celda sin contenido.

# ELEMENTOS: VISIBILIDAD

* Visibilidad:
  + overflow: visible | hidden | scroll | auto. Comportamiento del contenido si se desborda en la caja.
  + clip: rect (top, right, bottom, left) | auto. Especifica la región visible del elemento mediante las dimensiones de un rectángulo que hace de ventana de visualización.
  + visibility: visible | hidden | collapse. Visibilidad de las cajas.
  + display: muestra una caja con diferentes estilos. El más común es none, que se diferencia de visibility: hidden en que en este caso las cajas de alrededor se reorganizan cuando se oculta.

# ELEMENTOS: LISTAS

* Listas:
  + list-style-type: disc | circle | square…. Estilo aplicable a las listas.
  + list-style-image: url(http://...) | none. Imagen aplicable a los elementos de las listas.
  + list-style-position: inside | outside

# PSEUDO-CLASES Y PSEUDO-ELEMENTOS

Definen diferentes estados de un elemento. Por ejemplo:

* Enlaces
  + Normales: a:link {atributos}
  + Visitados: a:visited {atributos}
  + Activos: a:active {atributos}. Los enlaces están activos en el preciso momento en que se pulsa sobre ellos.
  + Hover: h1:hover {atributos}. Cuando el ratón está encima.
  + Foco: p:focus {...}

# ESPECIFICIDAD EN CASO DE CONFLICTOS

Interfaz de usuario gráfica, Diagrama

Descripción generada automáticamente

# POSICIONAMIENTO

## TIPOS DE ELEMENTOS (SEGÚN DISPLAY)

El atributo display puede tomar los siguientes valores:

* + **block**: elemento de bloque, ocupa la totalidad de la linea
  + **inline**: elemento de línea, ocupa el contenido del elemento
  + **inline-block**: elemento de línea pero admite atributos width y height.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

## POSITION

Position puede tener como valores:

* **Static**: valor predeterminado. No provoca ningún posicionamiento especial de los elementos y por tanto, los atributos top, left, right y bottom no se tendrán en cuenta.
* **Relative**: se ve influenciado por los elementos anteriores pero top y left definen la distancia respecto al último elemento.
* **Absolute**: permite posicionar cajas de manera absoluta de manera definida mediante los valores top, left, bottom y right. Los elementos no se ven afectados por el lugar de otros elementos. Si la posición absolute es respecto al contenedor, este debe ser relative.
* **Fixed**: Posiciona el elemento según posicionamiento absoluto, pero su posición final será siempre fija. Admiten valores top y left.
* **Sticky**: Es una mezcla de relative y fixed manteniendose fijo cuando el scroll baja.

## POSICIONAMIENTO FLOTANTE

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

Los elementos que se encuentran alrededor de una caja flotante adaptan sus contenidos para que fluyan alrededor del elemento posicionado.

Interfaz de usuario gráfica, Aplicación, Tabla, Word

Descripción generada automáticamente

Diagrama

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Aplicación

Descripción generada automáticamente con confianza baja

## SOLAPAMIENTO DE ELEMENTOS (Z-INDEX)

El atributo z-index permite definir el nivel de profundidad de una caja.

Su valor es un número entero.

El valor 0 suele tomarse como el nivel más bajo. Cuanto más alto sea el valor más cerca se mostrará la capa.

Z-index solo tiene efecto si aparece position

# FORMULARIOS

## MOSTRAR UN BOTÓN COMO ENLACE

Los botones de formulario también se pueden modificar para que parezcan enlaces.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

## MEJORAS EN LOS CAMPOS DE TEXTO

Añadiendo un pequeño padding a cada elemento <input>, se mejora notablemente el aspecto del formulario.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

## LABELS ALINEADAS

Los elementos <input> y <label> de los formularios son elementos en línea, por lo que el aspecto que muestran los formularios por defecto, es similar al de la siguiente imagen.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteInterfaz de usuario gráfica

Descripción generada automáticamente

## LABELS ALINEADAS EN LA MISMA LÍNEA

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente

# PRECEDENCIA DE ESTILOS

La precedencia de estilos es una manera de indicar que un estilo definido prevalece por encima de otro definido en la misma o en CSS diferentes.

Esto es necesario cuando hay dos o más estilos que actúan sobre los mismos atributos pero con diferente valor.

La precedencia de estilos va asociado con el concepto de especificidad, que se refiere al peso de cada uno de los elementos de una hoja de estilo.

Un cálculo sencillo para calcular la especificidad de una regla es sumar los puntos según los selectores que contenga.

* 1 punto: a un selector de etiqueta
* 10 puntos: selector de clase
* 100 puntos: selector de identificador.
* puntos: atributo de estilo (atributo style).

Además de la especifidad, se puede utilizar la declaración !important. Se situa en el lado del valor del atributo, antes del ;.

# DEFINICIÓN DE HOJAS DE ESTILO

El atributo rel se usa para definir la relación con el fichero enlazado:

* rel=stylesheet especifica un estilo persistente o preferido.
  + Persistente es aquel que se aplica si están activas las hojas de estilo.
  + Preferido es uno que se aplica automáticamente. La combinación de rel=stylesheet y un atributo title especifica un atributo preferido. No se puede especificar más de un estilo preferido.
* rel=“alternate stylesheet” define un estilo alternativo que el usuario podría elegir para reemplazar la hoja de estilo preferido.

Un estilo también puede definirse mediante múltiples hojas de estilo.

Otra alternativa es usar la regla @import incluida dentro de las etiquetas style.

El funcionamiento es igual que el anterior aunque @import no está soportado por todos los navegadores.

# HERRAMIENTAS Y TEST DE VERIFICACIÓN

W3C proporciona herramientas para validar código y hojas de estilo, comprobando si éstas son correctas según las gramáticas publicadas.

Si la validación no encuentra errores, sus autores podrán incluir un icono como el siguiente indicando que los desarrolladores se han preocupado por crear un sitio web interoperable y acorde al estándar.