

# **NIELSEN Y HTML**

## **Principios de usabilidad de Nielsen.**

1. **Visibilidad del estado del sistema:** el sistema debe mantener a los usuarios informados sobre lo que está ocurriendo.
2. **Relación entre el sistema y el mundo real:** el diseño debe hablar el idioma de los usuarios.
3. **Control y libertad del usuario:** permitir al usuario deshacer o cancelar acciones.
4. **Consistencia y estándares:** usar los mismos patrones en los elementos del diseño.
5. **Prevención de errores:** medidas para evitar errores.
6. **Reconocimiento en lugar de recuerdo:** objetos, acciones y opciones visibles y fácilmente accesibles.
7. **Flexibilidad y eficiencia de uso:** distintas maneras de llegar al contenido, aceleradores y acciones personalizadas.
8. **Diseño estético y minimalista:** eliminar información irrelevante.
9. **Ayuda a los usuarios a reconocer errores:** lenguaje claro y simple en los mensajes de error.
10. **Ayuda y documentación:** proporcionar información de ayuda cuando sea necesaria.

## **HTML 5 – Etiquetas básicas.**

- **Estructura.**  
Ejemplo: <header>, <nav>, <section>, <article>, <aside>, <footer>.
- **Texto.**  
Ejemplo: <h1>, <p>.
- **Enlaces.**  
Ejemplo: <a href = "">.
- **Listas.**  
Ejemplo: <ul>, <ol>, <li>.
- **Imágenes.**  
Ejemplo: <img src = ""/>.
- **Tablas.**  
Ejemplo: <table>, <tr>, <th>, <td>.
- **Formularios.**  
Ejemplo: <form action="" method = "POST">, <label>, <input>.

## HTML 5

- Última versión de **HTML**.

### HTML (HyperText Markup Language)

- **Lenguaje de marcas** diseñado para la creación de páginas web.

### HTML5. Comentarios.

- Se escriben con: `<!-- -->`.

### HTML5. Algunos aspectos a conocer.

- Los **elementos** que forman las páginas **HTML** están identificados por **MARCAS** o **ETIQUETAS**.
- Hay una etiqueta de apertura y una de cierre.

### Elementos vacíos.

- Son elementos sin contenido (**void elements**).
- No tienen una etiqueta de finalización.  
Ejemplo: `<br>`.
- Se pueden poner con auto cierre.  
Ejemplo: `<br/>`.

### Entidades.

- **Entidades de caracteres:** se usan cuando se quiere poner un **carácter especial**.  
Ejemplo: (`<`, `>`, `"` y `&`).
- Empiezan con un símbolo `"&"` y terminan con el símbolo `;"`.  
Ejemplo: queremos poner `"<"`, entonces escribimos `"&lt;"`;

### HTML5. Estructura y sintaxis.

- **DOCTYPE html:** identifica el tipo de documento que va a procesar (**HTML5**).
- **html:** elemento raíz y dentro de él se encuentra todo el contenido de la página.
- **head:** información no visible pero que sí afecta a cómo esta va a ser presentada.
- **body:** contenido visible de la página.

### HTML5. Elementos de HTML.

**Atributos:** proporcionan información adicional sobre **elementos HTML**.

### Características de los atributos:

- Tienen un espacio entre el **atributo** y el **nombre del elemento**.
- El nombre del atributo va seguido por un signo de igual (=).
- Suelen llevar **comillas de apertura** y de **cierre**.
- Todos los **elementos HTML** pueden tener **atributos**.
- Siempre se especifican en la **etiqueta de inicio**.
- Los valores `"true"` y `"false"` no están permitidos en los **atributos booleanos**.

### **Atributos Globales:**

- **Class:** agrupa elementos.
- **Contenteditable:** el contenido del elemento es editable.
- **Hidden:** el elemento no se representa en el navegador.
- **Id:** identifica el elemento.
- **Lang:** determina el idioma.
- **Spellcheck:** el elemento es analizado ortográficamente y gramaticalmente.
- **Style:** asigna un estilo al componente.
- **TabIndex:** cuando se pulsa tabulador determina el orden de selección de los elementos.
- **Title:** especifica información extra.
- **Translate:** traduce.

### **HTML5. Elementos <body>.**

**Elementos en bloque:** se posicionan uno sobre otro hacia abajo en la página.

### **Características:**

- Forma un bloque y se posiciona de forma vertical con un nuevo salto de línea.
- Pueden contener otros elementos de bloque o elementos de línea, pero no puede estar dentro de un elemento de línea.
- La anchura es la máxima que puede tomar dentro de su elemento contenedor (padre). Si no está dentro de ningún elemento, su anchura es la del navegador y la altura cambia en base al contenido que posea (con CSS se puede cambiar).

### **Ejemplos:**

- Address:
- Article:
- Aside:
- Blockquote:
- Details:
- Dialog:
- Div:
- Dl:
- Dt:
- Dd:
- Figure:
- Figcaption:
- Footer:
- Header:

- Hr:
- Li:
- Main:
- Nav:
- P:
- Pre:
- Section:
- Form:
- Fieldset:
- Table:

**Elementos en línea:** se posicionan lado a lado en la misma línea.

**Características:**

- Se posiciona horizontalmente en línea con los otros elementos.
- La anchura y altura se define en base al contenido que posea.
- Solo puede contener elementos de tipo inline.
- No se puede aplicar una anchura y una altura fija por medio de **CSS**.

**Ejemplos:**

- A:
- Abbr:
- B:
- Br:
- Cite:
- Code:
- Del:
- Dfn:
- Em:
- I:
- Iframe:
- Ins:
- Kbd:
- Mark:
- Q:

- S:
- Samp:
- Small:
- Span:
- Strong:
- Sub:
- Sup:
- Time:
- Var:

### **HTML5. Formularios.**

- Es un **bloque** de un documento **HTML** compuesto por un conjunto de elementos de entrada de información.
- Se puede usar **Javascript** en ellos.

### **HTML5. Elementos de los formularios.**

- **Form:** delimita una sección de una página **HTML** que contiene los componentes interactivos (**formulario**).
- **Input:** permite proporcionar un componente interactivo para que el usuario introduzca información para su envío a un servidor web.

**Principales valores del atributo "type" del elemento <input>:** button, checkbox, color, date, datetime-local, email, file, hidden, image, month, number, password, radio, range, reset, search, submit, tel, text, time, url, week.

**Otros atributos del elemento <input>:** list, name, placeholder, readonly, required, value,

- **Button:** permite crear un **botón**.
- **Select:** permite al usuario elegir una **opción** de un conjunto de ítems mostrando como una **lista de opciones** o como una **lista desplegable**. Las **opciones** se establecen con el componente **<option>** y se pueden **agrupar** con el componente **<optgroup>**.
- **Atributos significativos:**
  - **multiple:** indica si admite selección múltiple.
  - **size:** indica el número de líneas que se mostrarán simultáneamente.
- **Fieldset:** permite agrupar un conjunto de campos de un formulario. Mediante el elemento **<legend>** se puede asignar un texto que haga del título del marco.
- **Atributos significativos:**
  - **disabled:** permite deshabilitar todos los elementos agrupados.
  - **form:** el valor del atributo id del formulario al que está asociado.
  - **name:** asigna un nombre al conjunto de campos.

- **Label:** identifica una etiqueta de un campo de entrada de datos del formulario.
- **Meter:** representa un valor dentro de un rango.
- **Atributos significativos:**
  - **value:** indica el valor actual.
  - **min:** indica el valor mínimo.
  - **max:** indica el valor máximo.
  - **low:** indica el umbral por debajo del cual se considera que el valor es bajo.
  - **high:** indica el umbral por debajo del cual se considera que el valor es alto.
  - **optimum:** indica que el valor es óptimo.
- **Progress:** muestra un indicador visual del grado de avance de un proceso.
- **Atributos significativos:**
  - **value:** indica el valor actual.
  - **max:** indica el valor máximo.
- **Textarea:** proporciona un área para la introducción de texto.
- **Atributos significativos:**
  - **cols:** determina el ancho del componente.
  - **maxlength:** indica el número máximo de caracteres que admite.
  - **minlength:** indica el número mínimo de caracteres que admite.
  - **placeholder:** texto mostrado como ayuda que desaparece al introducir un valor.
  - **readonly:** indica que el componente sólo es de lectura.
  - **required:** hace obligatorio introducir valor en el componente.
  - **rows:** número de líneas visibles.
  - **wrap:** indica el método de ajuste del texto para el envío del servidor.

## **HTML5. Validaciones en formularios.**

### **Tipos de validación.**

- **Validación interna:** utiliza características de validación de **formularios HTML**. Esta validación **no** requiere **JavaScript**.
- **Validación externa:** se codifica utilizando **JavaScript**.

### **Expresiones regulares.**

- Es un **patrón** formado por una **cadena de caracteres** que define una **regla**.

**No lo resumo porque es mucho y se vio en clase.**

## CSS

- El **CSS (Cascading Style Sheets)** describe cómo se tienen que mostrar visualmente los elementos de **HTML**. Básicamente aporta la capa **visual** a los **elementos** del **HTML**.

### CSS. Evolución y estado actual.

- **Han existido:** **CSS1**, **CSS2**, **CSS3** y **CSS4**, aunque esta última **no es oficial**.

### CSS. Uso de CSS dentro de HTML.

1. **Como un elemento CSS externo:** las reglas se almacenan en un **documento externo** con extensión **.css** y se establece una relación entre el documento **HTML**.
2. **Como una declaración CSS interno:** definir la hoja de estilos dentro del propio documento **HTML** usando la **etiqueta <style>** dentro de la **sección <head>**.
3. **Como una declaración directa en las etiquetas:** definir el estilo directamente en las propias **etiquetas** de **HTML**, usando el **atributo style**.

### CSS 3. Estructura y sintaxis.

- **Reglas CSS:** son **declaraciones** sobre un estilo de **uno o varios elementos** y está compuesta por **dos** partes diferenciadas: un **selector** y una **declaración**.
- **Selector:** pueden ser **uno o varios** los **elementos** a los que se aplica la regla **CSS**. Cualquier **elemento** de **HTML** puede ser un **selector**.
- **Declaración:** está compuesta por dos partes diferenciadas, propiedad y valor, separadas por el carácter **dos puntos (:)** y es donde especifica el estilo del selector o elemento.
- **Propiedad:** permite modificar el **aspecto** de una **característica** de un **selector**.
- **Valor:** indica la **asignación** que se hace a la **propiedad**.

### Tipos de selectores (el propio nombre dice que hacen).

- Selector de **tipo**.
- Selector de **clase**.
- Selector de **id**.
- Selector **universal (todos)**.
- Selector de **atributo**.

### Tipos de combinadores (el propio nombre dice que hacen).

Los combinadores son **operadores** que se utilizan entre **selectores**.

- Combinador de **hermanos adyacentes**.
- Combinador **general de hermanos**.
- Combinador de **hijo**.
- Combinador de **descendientes**.

### **CSS3. Propiedades y valores.**

Un **selector** puede contener **una** o **varias propiedades** y sus **valores** respectivos.

- Propiedades de **texto**.
- Propiedades de **tipo letra**.
- Propiedades de **fondos**.
- Propiedades de **tamaño**.
- Propiedades de **posicionamiento**.

### **CSS3. Comentarios.**

- Los comentarios son necesarios en cualquier tipo de lenguaje para agregar significado al **código** o para realizar **depuraciones**.  
Ejemplo: **/\*\*/**.

### **CSS 3. Agrupamientos.**

- Para aplicar la misma declaración a distintos selectores se agrupan por el carácter **coma (,)**.

### **CSS. Prioridades y orden.**

- El número de **reglas** que se pueden aplicar a un documento **HTML** es **ilimitado** y estas pueden entrar en **conflicto**.

### **Hay tres conceptos fundamentales en la aplicación de los estilos:**

- **Herencia**: provoca que los **valores** de algunas de las **propiedades** aplicadas a un **elemento padre** o **contenedor**, se apliquen también a los **elementos contenidos** por él (**hijos**).
- **Especificidad**.
- Los estilos **CSS** se aplican en **cascada**.