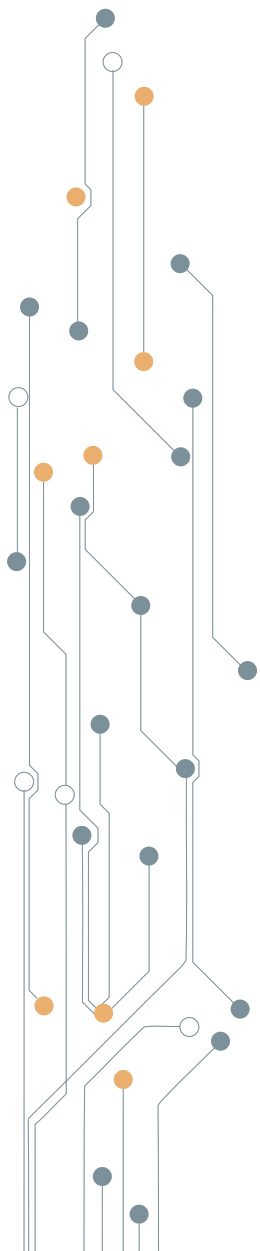




# Principios de diseño de páginas web

# Índice



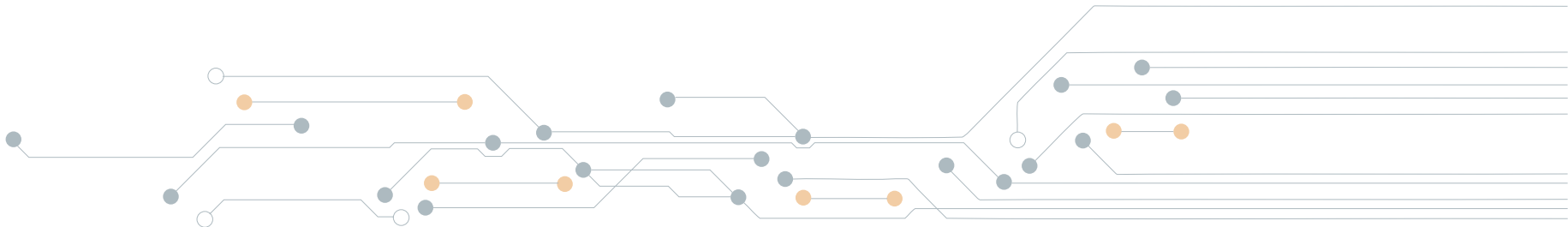
1   Principios de diseño de páginas web	3
1.1   Compatibilidad	3
1.2   Utilidad	4
1.3   Interoperabilidad	6
1.4   Acceso universal	7

# 1. Principios de diseño de páginas web

Antes de comenzar en materia de desarrollo propiamente dicho, vamos a ver algunos conceptos de diseño que debemos conocer previamente. Son directivas o consejos de cómo debemos enfocar el desarrollo de una web.

## 1.1 | Compatibilidad

- **Dar soporte al contenido existente:** los navegadores deben ser compatibles con cualquier versión de HTML, CSS o JavaScript. No todo el contenido que encontramos en la Web está desarrollado con las últimas tecnologías.
- **Degradar de forma elegante (degrade gracefully):** cuando diseñamos páginas web con las últimas tecnologías y estándares, debemos tener en cuenta cómo se van a comportar en navegadores antiguos y dar una alternativa de visualización adecuada para estos dispositivos.
- **No reinventar la rueda:** se debe fomentar la reutilización de elementos. Si existe algo que ya funciona correctamente, úsalo. Si desarrollas algo nuevo, hazlo de manera que pueda ser reutilizado en el futuro.
- **Construyendo sobre lo que ya existe:** si ya hay cierta práctica extendida, es preferible adoptarla y mejorarla que prohibirla o sustituirla por algo radicalmente diferente.
- **Evolución, no revolución:** es preferible evolucionar progresivamente un diseño cambiando cosas sobre lo ya construido, que cambiar un diseño por otro.



## 1.2 | Utilidad

- **Resolución de problemas reales:** cualquier modificación o ampliación de un diseño debe responder a una necesidad real. No se deben añadir funcionalidades innecesarias ya que podemos construir páginas o aplicaciones web con una excesiva complejidad de uso.
- **Prioridad de los usuarios en caso de conflicto:** en relación con el diseño de una especificación como la de HTML, CSS o JavaScript, encontramos a varios grupos de personas: los usuarios de la web, los autores del contenido, los desarrolladores de los navegadores, los redactores de las especificaciones y, por último, los puramente teóricos.

Si existiese un conflicto de intereses entre alguno de estos grupos, en relación a una funcionalidad concreta, este se debería resolver según el orden anteriormente citado.

Por ejemplo, si hubiese un conflicto en relación a la representación visual de una tabla entre usuarios de la web y los desarrolladores de los navegadores web, el conflicto se debería resolver en favor de los usuarios.



- **Seguro por diseño:** la seguridad en una página web, debe ser tomada en consideración desde el propio diseño. Por ejemplo: accesos seguros mediante una página de login (mediante HTTPS), control de acceso de usuarios, roles, permisos, etc.

No solo debemos dar soluciones de seguridad con herramientas de implementación.

- **Separación de responsabilidades:** HTML nos permite crear páginas web con un contenido y presentarlo al usuario de una forma atractiva. Sin embargo, debemos tener en cuenta que separar el contenido de la presentación es una buena estrategia a seguir. Lo podemos separar mediante los siguientes elementos:

- **HTML:** documento en el que escribimos el contenido de una página.
- **CSS:** estilos que podemos aplicar sobre un documento HTML para definir cómo se va a visualizar su contenido en un navegador.

De esta forma, podemos representar un mismo contenido de formas diferentes, aplicando distintas hojas de estilo CSS.

También podemos reutilizar estilos aplicados a un documento HTML, a cualquier otro documento HTML que necesitemos.

- **Consistencia del árbol DOM:** el árbol DOM (Document Object Model) es una estructura de datos en la que el navegador almacena la información que lee de una página HTML.

Esta estructura de datos debe ser implementada de igual forma entre los distintos navegadores. Solo de esta forma se garantiza que un script contenido en una página HTML puede obtener la misma



## 1.3 | Interoperabilidad

- **Biendefinida** por comportamiento: se debe definir con precisión, el funcionamiento de cada requisito de la especificación.
- **Evitar complejidad innecesaria:** de todas las soluciones correctas a un problema, la solución que se debe escoger en todos los casos es la más sencilla. De esta forma, será más sencillo realizar desarrollos evolutivos por nuestra parte, o por la de cualquier otro desarrollador.
- **Control de errores:** definir un correcto control de errores es un requisito importante para desarrollar aplicaciones interoperables. En ningún caso, la aplicación debe parar bruscamente su ejecución.

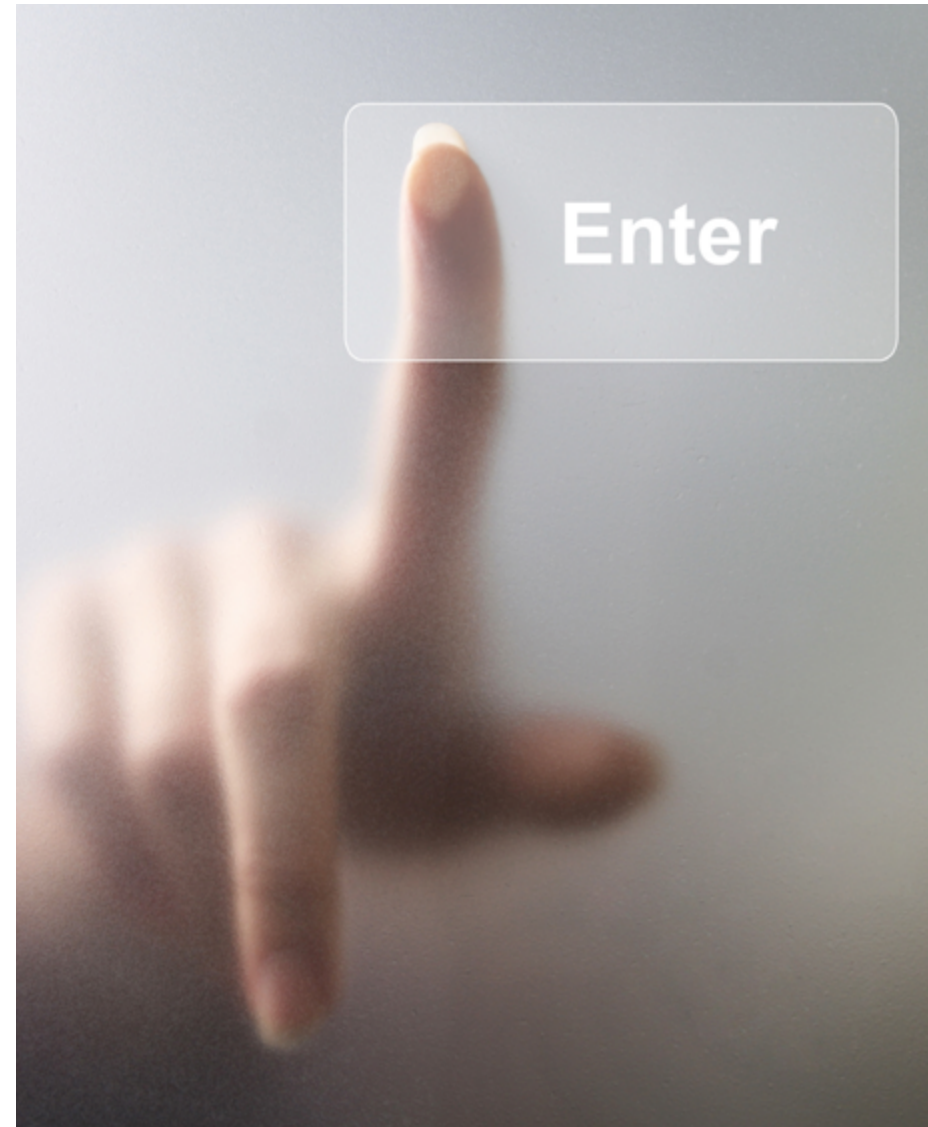
El usuario debe recibir notificaciones elegantes y con un contenido fácilmente interpretable por él, pero nunca detalles técnicos del error.





## 1.4 | Acceso universal

- **Independencia del medio:** las páginas deben poder ser aceptadas por cualquier navegador, independientemente de la plataforma o dispositivo en el que van a ser visualizadas.
- **Apoyo a diferentes idiomas:** en un diseño, debemos tener en cuenta a que público va destinado y tener en cuenta el idioma en el que se relacionan dichos grupos. Tenemos que desarrollar permitiendo que el usuario al que va dirigido el contenido pueda verlo en su idioma.
- **Accesibilidad:** se deben diseñar funcionalidades teniendo en cuenta que pueden ser utilizadas por personas con discapacidad. Por ejemplo, permitir estilos de alto contraste, añadir textos alternativos para las imágenes (por ejemplo, si usamos un lector de páginas), ser cuidadoso en la selección de los colores (hay un alto porcentaje de personas con daltonismo).



*Telefonica*

---

EDUCACIÓN DIGITAL