1. Accesibilidad Web desde una perspectiva de Mejora Progresiva. 2

1.1 Mejora Progresiva y Tolerancia a Fallos 2

1.2 Accesibilidad Web Universal 4

1.2.1 Diseño Universal e Inclusivo 4

1.2.2 Largo camino del desarrollo de los estándares de la web 8

1.2.3 Un paso más que la Accesibilidad Web normal 14

1.3 Accesibilidad Web en Capas 17

2. El futuro de la Accesibilidad Web: AOM (Accessibility Object Model) 19

## 1. Accesibilidad Web desde una perspectiva de Mejora Progresiva.

### 1.1 Mejora Progresiva y Tolerancia a Fallos

Mejora progresiva (Progressive Enhancement) es una estrategia particular de diseño web que acentúa la accesibilidad, margen de beneficio semántico, y tecnologías externas del estilo y el scripting, en una manera adecuada que permite que cada uno tenga acceso al contenido y a la funcionalidad básica de una página web, usando cualquier navegador web o conexión a Internet, mientras que también permite a otros con un mayor ancho de banda o un navegador web más avanzado experimentar una versión mejorada de la página. (Champeon Steven, 2003)

El concepto anterior fue acuñado por primera vez por Steven Champeon, de la empresa de diseño web *hesketh.com*, en una serie de artículos y presentaciones para Webmonkey y la conferencia interactiva Sxsw realizada entre marzo y junio de 2003.[[1]](#footnote-1)

Según otro experto en el área, Aaron Gustafson (2011), mejora progresiva se basa en un principio: tolerancia a fallos (Fault Tolerance). La tolerancia a fallos es la capacidad de que un sistema puede continuar operando cuando encuentra un error inesperado. Hoy en día, sobre todo en el contexto de la web, no importa que eres diseñador profesional o consumidor casual, estás beneficiándose de la tolerancia de fallos todo el tiempo.

La filosofía de tolerancia a fallos no solo se integra en los protocolos que encaminan una solicitud desde su navegador web al servidor que está tratando de llegar, sino también se muestran en la semántica de los lenguajes que han convertido completamente el desarrollo de la web: HTML y CSS. Según lo prescrito por las especificaciones de los dos lenguajes, los navegadores web deben ignorar todo lo que no se entienden. Ese simple requisito hace posible la mejora progresiva.

Otro aspecto interesante del diseño de la tolerancia de fallos es adaptable en la evolución. Ambos HTML y CSS fueron diseñados para ser compatibles con versiones posteriores (Forward Compatible[[2]](#footnote-2)), lo que significa que todo lo que escribamos hoy funcionará mañana y el próximo año y dentro de diez años. En cierto sentido, es un estado perfecto: los dos son diseñados para prosperar evitando preocuparse por el cambio del entorno de navegación.

Estos lenguajes fueron diseñados para evolucionar con el tiempo, por lo que a los navegadores web se les instruyó que obedecieran las reglas de tolerancia a fallos e ignoraran cualquier cosa que no entendieran. Esto permita a estos lenguajes a crecer y adaptarse con tiempo, sin llegar a un punto en el que el contenido que almacenan y el estilo de la web ya no puedan leerse o que corra el riesgo de provocar la falla de un navegador. Por ejemplo, con la tolerancia a fallos, podemos navegar por un sitio web construido por HTML5 en Lynx y experimentar con las características de CSS3 sin preocuparnos por romper Internet Explorer 6.

Comprender la tolerancia a fallos es la clave para entender la mejora progresiva. La tolerancia a errores es la razón por la cual la mejora progresiva funciona y hace posible asegurar que todo el contenido entregado en la web esté accesible y disponible para todos.

### 1.2 Accesibilidad Web Universal

#### 1.2.1 Diseño Universal e Inclusivo

El término "Diseño Universal" fue acuñado por el arquitecto Ronald Mace para describir el concepto de diseñar todos tipos de productos (edificios, electrodomésticos, muebles, etc.) y el entorno construido, que sea estético y utilizable en la mayor medida posible para todos, independientemente de su edad, capacidad o estado en la vida.[[3]](#footnote-3)

El movimiento de Diseño Universal se expandió su influencia rápidamente en muchas áreas. Ya no se limite su uso en arquitecta, ha marcado un gran impacto en el Diseño Inclusivo en el diseño de software, que fue promovido principalmente por Microsoft desde el inicio de este siglo. Otro tema popular que se discuten bastante recientemente como por el el entorno de trabajo inclusivo (Inclusive Workplace) es también herencia de este concepto.

Un informe de Microsoft muestra que 57%(74.2 millones) de los usuarios de PC benefician del uso de las tecnologías de accesibilidad.[[4]](#footnote-4)

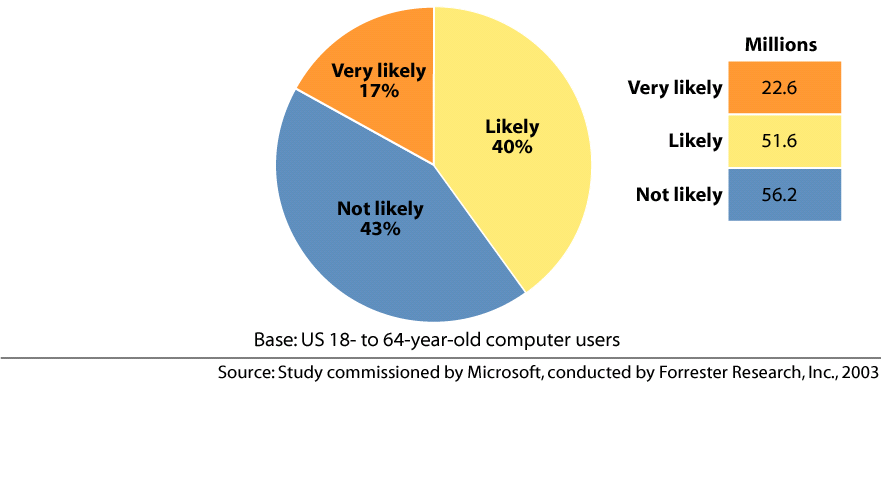


Figura 1. La mayoría de los usuarios de PC benefician del uso de las tecnologías de accesibilidad. (Microsoft, 2003)

“En un mundo con 7.4 billones de personas, nuestro objetivo es crear productos que son físicamente, cognitivamente y emocionalmente apropiados para cada uno de ellas. Reconocer la diversidad de los seres humanos es un fuente principal para un mejor diseño.” (Microsoft, 2016)

El Diseño Inclusivo o Universal no solo abrió los productos y las experiencias de Microsoft a personas con una gama más amplia de capacidades sino también refleja el estado verdadero de los seres humanos. Todos estamos creciendo, cambiando, y adaptando al mundo alrededor de nosotros cada día, por eso el diseño debe reaccionar a esta realidad de diversidad humana.

En el tiempo contemporáneo, los principios del Diseño Universal fue extendida por The Center for Universal Design at North Carolina State University a los siguientes[[5]](#footnote-5):

1. **Igualdad de uso:** el diseño debe ser fácil de usar y adecuado para todas las personas independientemente de sus capacidades y habilidades.
2. **Flexibilidad:** el diseño debe poder adecuarse a un amplio rango de preferencias y habilidades individuales.
3. **Simple e intuitivo:** el diseño debe ser fácil de entender independientemente de la experiencia, los conocimientos, las habilidades o el nivel de concentración del usuario.
4. **Información fácil de percibir:** el diseño debe ser capaz de intercambiar información con usuario, independientemente de las condiciones ambientales o las capacidades sensoriales del mismo.
5. **Tolerante a errores:** el diseño debe minimizar las acciones accidentales o fortuitas que puedan tener consecuencias fatales o no deseadas.
6. **Escaso esfuerzo físico:** el diseño debe poder ser usado eficazmente y con el mínimo esfuerzo posible.
7. **Dimensiones apropiadas:** los tamaños y espacios deben ser apropiados para el alcance, manipulación y uso por parte del usuario, independientemente de su tamaño, posición, y movilidad.

La mejora progresiva ha tomado también los principios del Diseño Universal como base fundamental a la hora de desarrollar una Accesibilidad Universal, respecto a la diversidad de los usuarios, en este caso, los usuarios de la web, navegador principalmente y más adelante otros dispositivo como el móvil. Tolerancia a fallos como mencionado anteriormente es su enfoque principal.

En el entorno del desarrollo de la web, este concepto también fue aceptado eventualmente con el establecimiento de los estándares web en los últimos años. W3C[[6]](#footnote-6) fue creada con la misión principal para llevar a la World Wide Web a su máxima potencia mediante el desarrollo de protocolos y pautas que garanticen el crecimiento a largo plazo de la Web. Bajo esta misión, están una variedad de aspectos importantes, todos ellos favorecen la profundización de la visión del W3C de One Web.[[7]](#footnote-7)

En el aspecto de diseño, está bajo dos principios:

**Web for All (La web para todos)**

El valor social de la Web es facilitar la comunicación humana, el comercio y las oportunidades de compartir conocimiento. Uno de los principales objetivos del W3C es poner estos beneficios a disposición de todas las personas, independientemente de su hardware, software, infraestructura de red, idioma nativo, cultura, ubicación geográfica o capacidad física o mental.

**Web on Everything (La web en todos tipos de terminales)**

La cantidad de diferentes tipos de dispositivos que pueden acceder a la Web ha crecido enormemente. Teléfonos móviles, teléfonos inteligentes, asistentes digitales personales, sistemas de televisión interactiva, sistemas de respuesta de voz, quioscos e incluso ciertos electrodomésticos pueden acceder a la Web.

En relación con la accesibilidad web, es también muy importante extender su concepto a un sentido universal para adoptarse en un nuevo contexto de gran desarrollo de la World Wide Web.

#### 1.2.2 Largo camino del desarrollo de los estándares de la web

##### De HTML a HTML5

HTML fue creado por Tim Berners-Lee en 1991, y las versiones 1 a 4 de HTML se desarrollan durante la década de 1990 por W3C. En estos primeros días de uso general de Internet, HTML muestra de manera eficiente en la gran mayoría del contenido web, ya que en este momento la web se consistía principalmente en los sitios estáticos no interactivos.

Pero desde final de los años 90, las versiones anteriores de HTML empiezan a mostrar dificultades en manejar los sitios interactivos y de multimedia de la última generación. Para obtener los mejores resultados, los desarrolladores tenían que pedir apoyo a plugins de terceros.

El año 2000, W3C recomienda XHTML 1.0, un lenguaje de marcado basado en XML que refleja / amplía HTML. En este periodo, el desarrollo del estándar web está enfocado al XML y XHTML. El año 2004, el desarrollo de HTML fue cerrado por W3C.

No obstante, el mismo año, un grupo de desarrolladores de HTML de Apple, Mozilla Foundation y Opera Software crean una comunidad, llamado Web Hypertext Application Technology Working Group (WHATWG[[8]](#footnote-8)), con el objetivo de desarrollar una nueva versión de HTML para reaccionar a la web dinámica moderna, manteniendo compatibilidad con el código HTML existente. En un tiempo corto, WHATWG se populariza entre los programadores de navegador, y poco a poco llegó a ser reconocido por la mayoría de los navegadores. El término Web 2.0 también se llega al público general en este año, los sitios web dinámico y con multimedia empezó a florecer.

El año 2006, W3C también anunció su soporte a este proyecto. El HTML5 fue lanzado por WHATWG el año 2008. W3C y WHATWG han decidido separar su desarrollo de HTML5. W3C trabaja en el estándar definitivo de HTML5 mientras WHATWG enfoca en el desarrollo de un “estándar vivo”, lo cual se mejora con el tiempo y la evolución.

Al final, el 2014, W3C publicó oficialmente su recomendación de HTML5.[[9]](#footnote-9) (Más sobre HTML5, véase 2.2 de la tesis)

##### Breve historia de JavaScript y la “Guerra de Navegadores”.

JavaScript es un lenguaje de programación interpretado que fue creado por Brendan Eich de Netscape el año 1995 para ejecutar en su navegador web Netscape Navigator. Gracias al uso de JavaScript, hoy en día el contenido de la web ya no se limita en formato de texto, y los usuarios pueden tener una gran diversidad de interacciones con la web.

Al lanzar la versión 1.0 de JavaScript, Netscape Navigator logró a dominar el mercado de los navegadores web. Pero luego esta situación fue cambiada con la entrada al escenario del Microsoft, con su lanzamiento del sistema operativo Windows 95 OSR2, lo cual llevó empaquetada su propio navegador web Internet Explorer 3. Las tecnologías de script de Microsoft, incluidas VBScript y JScript, se lanzaron en 1996. JScript, una implementación de ingeniería inversa del JavaScript de Netscape, formaba parte de Internet Explorer 3. Con el lanzamiento de JScript, Microsoft llegó a alcanzar y adelantar Netscape en esta fase de la “Guerra de Navegadores”.

Frente a la competición del Microsoft, Netscape y Sun (otra compañía creadora del lenguaje JavaScript, también la de Java) entregaron JavaScript al ECMA (European Computer Manufacturers Association) para crear una especificación estándar, que los proveedores de navegadores podrían implementar en función del trabajo realizado en Netscape. Esto llevó al lanzamiento oficial de la especificación del lenguaje ECMAScript, la primera edición de la norma ECMA-262 fue publicada en junio de 1997. Los lenguajes de uso populares hoy como JavaScript, JScript o ActionScript son implementaciones de ECMAScript.

En las versiones tempranas de JavaScript también ofreció el DOM (Modelo de Objetos del Documento) de un nivel primario (o el llamado DOM Level 0) para definir de manera abstracta el contenido de los documentos, o sea un conjunto estándar de objetos para representar documentos HTML, XHTML y XML. El DOM permite el acceso dinámico a través de la programación, puede acceder, añadir y cambiar dinámicamente contenido estructurado en documentos con lenguajes de ECMAScript (JavaScript).

El origen del DOM viene del DHTML (HTML Dinámico), el DHTML no indica que han creado un lenguaje nuevo o han incorporado una nueva tecnología, sino que es un término para representar la combinación del uso de HTML, CSS y JavaScript en el diseño web. Con la continuación de “Guerra de Navegadores” entre Netscape y Microsoft, el año 1997, ambos lanzaron su nueva versión de navegador web, aumentando el soporte al DHTML, el predecesor del DOM, afortunadamente, el DOM de los dos no son compatibles. Esto dirigió a una situación ridícula: los programadores a la hora de trabajar con DOM, son forzados a considerar en qué navegador web se ejecutaría, en muchos casos, se tenía que escribir dos veces el código.

El octubre de 1998, la especificación estándar completa de DOM, también conocido como “DOM Level 1”, fue recomendada por W3C. Gracias a este esfuerzo del W3C, hasta este momento, el DOM estandarizado es soportado con mejor funcionalidad por las principales compañías proveedoras de navegador web incluso las anteriormente mencionadas.

(Más sobre ECMAScript y DOM, véase 2.4)

##### Antes de la Mejora Progresiva

Con la expansión rápida de la web, desde el nacimiento del primer navegador web, primero HTML, logró el intercambio de texto e imágenes, luego los contenidos audiovisuales y más allá unas interacciones también fue incluidas. Mientras las tecnologías de la web seguían avanzando con un enorme desarrollo, los diseñadores se peleaban para incluir las herramientas de última moda en sus páginas web, el tema de la tolerancia a fallos fue totalmente descuidado. Existían sitios web que fueron construidos con páginas totalmente maquetadas por imágenes en formato JPEG, otros se convirtieron en santuarios para Flash, pocos de estos sitios web son usables y con menor cantidad los que son accesibles.

Desde entonces, se aumentaba el interés hacia un diseño web de tolerancia a fallo, bajo este concepto, solía usar en esta etapa una estrategia conocida como “Degradación Agraciada”. Lo que se intentaba con la Degradación Agraciada era degradar la página para que pudiera ser vista o ser de alguna manera presentable en versiones anteriores del navegador, donde se suponía no existían las tecnologías que necesitaba el diseño de la página web. En la práctica, la "Degradación Agraciada" ha sido suplantada por la actitud de que el usuario simplemente "debe actualizar"[[10]](#footnote-10). Esta actitud ha ganado terreno debido a apremios de tiempo y presupuesto, dificultad para hacer pruebas en navegadores alternativos, así como también de creer que "los navegadores Web son libres". (González, Oscar et al. 1997).

Sin embargo, no siempre es posible actualizar el navegador debido a las políticas del departamento TI de una empresa, por ejemplo. Un hardware antiguo, o por alguna otra razón. La actitud "debe actualizar" no toma en cuenta la deliberada elección de los usuarios y la amplia variedad de plataformas en los navegadores; muchos de los cuales se ejecutan sobre distintos dispositivos portátiles o en entornos donde es limitado el ancho de banda, color o sonido, o el tamaño de la pantalla y así sucesivamente. Características que los hacen distintos al típico navegador gráfico de escritorio. (Champeon, Steven, 2003)

La Degradación Agraciada, en su uso en el contexto de la web, suele dar una mejor experiencia, una comida completa a los navegadores más recientes y avanzados mientras ofrecer casi nada a las personas obligadas a usar un navegador de versión antigua y con menor capacidad. Es una filosofía que se toma en consideración prioritariamente la compatibilidad hacia delante (Forward Compatibility), mientras, en este caso la compatibilidad hacia atrás o la retrocompatibilidad (Backward Compatibility) no es el tema que hay que preocuparse. (Aaron Gustafson, 2011)

Simplemente porque una persona está navegando con un navegador antiguo o con otras condiciones inferiores, no puede disfrutar la accesibilidad web como el resto, esto no es una buena estrategia de tolerancia a fallos. Es por la hipótesis dudosa planteada en la “Degradación Agraciada”, surgió la demanda de una estrategia alternativa del diseño web, esto tiene una mejor presentación en el argumento de Steve Champeon[[11]](#footnote-11).

##### ¿Fue HTML5 quien ha matado Flash?

El 25 de julio, 2017, finalmente anunció Adobe en su blog oficial: “Dejaremos de actualizar y distribuir Flash Player al final de 2020 y alentaremos a los creadores de contenido a migrar cualquier contenido Flash existente a los nuevos formatos abiertos.”[[12]](#footnote-12)

A menudo se dice que “HTML5 ha matado a Flash”, pero ¿realmente lo hizo? Se mencionan al mismo tiempo en artículos y publicaciones con mucha frecuencia (como la famosa carta de Steve Jobs[[13]](#footnote-13)), y muchos han hecho comparaciones entre los dos, la verdad es que no son equivalentes directos. Aunque en el manejo del contenido multimedia de la web, la combinación de HTML, CSS y JavaScript fue un modelo alternativo de la plataforma de Adobe por una década, en aquella época, como mencionaba anteriormente, con la ausencia de especificaciones estándares, los programadores web son obligados a buscar ayuda de plugins externos, en este contexto, Adobe Flash. No fue hasta los últimos años, especialmente con el lanzamiento de las nuevas API (Interfaz de programación de aplicaciones) de contenido multimedia, HTML5 logra una mayor aceptación en este escenario.

Además en el sentido de la Accesibilidad Web, la queja sobre el cerrado de Flash es un lugar común para los programadores, como bien explicaba en la carta de “Mis opiniones sobre Flash” de Steve Jobs: Flash es un producto 100% propietario, solo está disponible desde Adobe, y Adobe tiene la autoridad exclusiva para su mejora futura, fijación de precios, etc. Aunque los productos Flash de Adobe están ampliamente disponibles, esto no significa que estén abiertos, ya que están controlados por Adobe y solo están disponibles en Adobe. En casi cualquier definición, Flash es un sistema cerrado. Al mismo tiempo, Steve Jobs y Apple abrazan a HTML5, CSS y JavaScript, porque todos ellos son estándares webs abiertos.

Es por Flash es una herramienta cerrada no accesible, lo cual está en contradicción al espíritus de la web, poco a poco fue abandonado entre los diseñadores y desarrolladores web, y es por ser estándar abierto, HTML5 llega su etapa de florecimiento.

#### 1.2.3 Un paso más que la Accesibilidad Web normal

Para muchos de los diseñadores web, asegurar de que los sitios web o las aplicaciones sean accesibles es una cosa de consideración segundaria o algo que puede mejorar después del proceso de diseño. Hoy en día, se hablan con abundante frecuencia la idea de “Diseño Centrado al Usuario[[14]](#footnote-14)”, mientras cuando se trata del tema de Accesibilidad, es en muchas ocasiones reconocido como referencia del sinónimo del “Lector de Pantalla”.

Sin duda alguna, las personas con impedimentos visuales suelen utilizar un lector de pantalla para consumir contenido, pero también pueden usar un dispositivo de retroalimentación táctil braille o una impresora braille. Probablemente también puede utilicen un teclado; o pueden usar una pantalla táctil junto con señales de audio; incluso pueden usar una cámara para permitirles "leer" contenido a través del reconocimiento óptico de caracteres (OCR) y texto a voz.

Es bastante llamativa la discapacidad visual puede afectar a un porcentaje decente de la población (注释), incluso nos obliga a enfrentar la verdad de que todos envejecemos y podemos ser atacado por unas enfermedades de este tipo también, pero esto es solamente una parte de las rompecabezas de la Accesibilidad Web.

Así argumenta Aaron Gustafson (2017):

*“Fundamentalmente, la mejora progresiva es un tema de accesibilidad, pero no en un sentido limitado como su uso normal con mayor frecuencia. El término ‘accesibilidad’ se usa tradicionalmente para denotar hacer disponible el contenido para las personas con ‘necesidades especiales’ (personas con motilidad limitada, discapacidades cognitivas o impedimentos visuales); la mejora progresiva lleva esto un paso más allá al reconocer que todos tenemos necesidades especiales. Nuestras necesidades especiales también pueden cambiar con el tiempo y en diferentes contextos. Cuando cargo un sitio web en mi teléfono, por ejemplo, mi resolución de pantalla me limita visualmente (especialmente si uso un navegador que alienta el zoom) y tengo una capacidad limitada para interactuar con botones y enlaces porque estoy navegando con las yemas de los dedos, que son mucho más grandes y menos precisas que un cursor de mouse.”*

Todos nos beneficiamos cuando los diseñadores consideran la accesibilidad en todo el proceso de su diseño. Todos tenemos necesidades especiales. La Accesibilidad Web en sentido universal, se trata de reconocer ese hecho y tomar medidas para abordarlo.

### 1.3 Accesibilidad Web en Capas

En el mundo de la web, el contenido es el rey, el objetivo principal de la web es facilitar el intercambio de información. Primero que todo, es imprescindible identificar la información núcleo en la construcción de una página web o una aplicación, por ejemplo, para un sitio web de noticia, su función principal es publicar noticias, otros servicios como rompecabezas interactivas o notificación a tiempo real son también muy interesante, pero a lo mejor no comparte la misma importancia que garantizar el acceso de la noticia. (Jeremy Keith, 2016)

Una web accesible, en un sentido universal desde la perspectiva de mejora progresiva, es facilitar la transmisión del contenido núcleo con el fallo de unas tecnologías específicas, ya que cada año se escucha numerosos avances tecnológicos de la web, pero a la hora de aplicarlas, hay que tener en cuenta la diversidad de los usuarios. La accesibilidad web en este sentido, es la capacidad de adaptarse en una diversidad de navegadores, o sea, los medios que usan los usuarios para acceder a la web, y en su consecuencia, garantizar su experiencia de navegación.

Una gran ventaja de la filosofía de mejora progresiva es reconocer estas infraestructuras o sea las tecnologías que soportan la presentación del contenido o sea el interfaz como diferentes capas, cada capa cumple su función determinada. Con el funcionamiento de las capas, garantizará la experiencia de usuario de mayor medida posible. Por lo que se clasifica el tipo de experiencia del usuario en los niveles 5 siguientes:

* **Texto:** el nivel básica, no hay tecnología específica para figurar esta capa, aunque sí que requiere una capacidad competente del ***copywriter*** (redactor publicitario). Contenido textual mejor redactado puede facilitar el acceso del usuario.
* **HTML:** este nivel está asociado con las semánticas del lenguaje de marcas: HTML (lenguaje de marcas de hipertexto), en este caso, son los elementos o atributos que se usa en una página web para ofrecer información adicional o complementaria del contenido.
* **CSS:** Según Aaron Gustafson (2017), este nivel de experiencia constituye el factor clave para los contenidos audiovisuales, sobre todo en el uso de CSS en las imágenes, audios y videos en línea (inline)[[15]](#footnote-15).
* **JavaScript:** es el nivel que mantiene la interacción, en consideración a los estándares de la web, la experiencia de este nivel está totalmente dependiendo al JavaScript.
* **Accesibilidad:** es el nivel más avanzada, cuyo objetivo es una mejor semántica y práctica junto con the Web Accessibility Initiative’s Accessible Rich Internet Applications (WAI-ARIA).

Es como dijo el arquitecto finlandés Eliel Saarinen[[16]](#footnote-16): “Una silla en una habitación, una habitación en una casa, una casa en un ambiente, un ambiente en un plan de la ciudad.”

Esta parte de la tesis fue inspirada de la filosofía de mejora progresiva y el diseño adaptable, la tomamos como la base teórica para desarrollar el esquema de la Accesibilidad Audiovisual Estratificada. Básicamente, es reconocer como capas o distintos niveles la infraestructura de una página web o una aplicación en el proceso de diseño, a través de mantener la tolerancia a fallos de la misma, garantizará los contenidos audiovisuales sea accesibles y usables, y a su vez, una mejor experiencia del usuario.

## 2. El futuro de la Accesibilidad Web: AOM (Accessibility Object Model)

1. “The Future of Accessibility for Custom Elements,” Rob Dodson talks internets, October 2, 2017, <http://robdodson.me/the-future-of-accessibility-for-custom-elements/>.

Se presentará un panorama de las posibilidades o tendencia de desarrollo de la Accesibilidad Audiovisual, tanto en su concepto universal como en áreas más específicos. Se dividen en las 5 capas, del nivel básico al más avanzado, son las siguientes:

1. Contenido audiovisual en si mismo
2. HTML (HTML 5 API)
3. CSS (@media)
4. Javascript (EMCAscript 6 y DOM)
5. Accessibility

y como una dimensión extendida, una última capa en su uso de desarrollar las aplicaciones web (Web App): (Adaptive Streaming)

1. PWA (Progressive Web Apps)

1. Champeon, Steven (2003) [realce progresivo de <http://www.webmonkey.com/03/21/index3a.html> y el futuro del diseño del Web (Webmonkey)] 14 de abril de 2006 recuperado. [↑](#footnote-ref-1)
2. “Really Undoing Html.css,” Thoughts From Eric (blog), 15 de septiembre, 2004, <http://meyerweb.com/eric/thoughts/2004/09/15/emreallyem-undoing-htmlcss/> . Un ejemplo de la compatibilidad de versión posterior de HTML y CSS. [↑](#footnote-ref-2)
3. “Center for Universal Design NCSU - About the Center - Ronald L. Mace,” 25 de diciembre, 2017, <https://projects.ncsu.edu/ncsu/design/cud/about_us/usronmace.htm>. [↑](#footnote-ref-3)
4. Microsoft (2003): La amplia gama de capacidades y su impacto en la tecnología informática. <http://download.microsoft.com/download/0/1/f/01f506eb-2d1e-42a6-bc7b-1f33d25fd40f/researchreport.doc> [↑](#footnote-ref-4)
5. “The Center for Universal Design - Universal Design Principles,” 25 de diciembre, 2017, <https://projects.ncsu.edu/ncsu/design/cud/about_ud/udprinciplestext.htm>. [↑](#footnote-ref-5)
6. El World Wide Web Consortium (W3C) es una comunidad internacional donde las organizaciones miembros, personales a tiempo completo y el público trabajan juntos para desarrollar estándares web. Es dirigido por el inventor de la web Tim Berners-Lee y el CEO Jeffrey Jaffe. [↑](#footnote-ref-6)
7. “Misión de W3C”, 9 de diciembre, 2017, <https://www.w3.org/Consortium/mission>. [↑](#footnote-ref-7)
8. “Web Hypertext Application Technology Working Group,” Wikipedia, la enciclopedia libre, 4 de noviembre, 2015, <https://es.wikipedia.org/w/index.php?title=Web_Hypertext_Application_Technology_Working_Group&oldid=86456573>. [↑](#footnote-ref-8)
9. Tabla de desarrollo cronológico de HTML5. <https://www.fasthosts.co.uk/blog/websites/look-back-html5> 14 de diciembre de 2017. [↑](#footnote-ref-9)
10. Un ejemplo del evento de la enlace, muestra que en la Degradación Agraciada, se supone que la función de imprimir es disponible en todos los navegadores. “Graceful Degradation versus Progressive Enhancement - W3C Wiki,” revisado 19 December 2017, <https://www.w3.org/wiki/Graceful_degradation_versus_progressive_enhancement>. [↑](#footnote-ref-10)
11. “Inclusive Web Design,” entrado en 19 December 2017, <http://www.hesketh.com/publications/inclusive_web_design_for_the_future/>. [↑](#footnote-ref-11)
12. “Flash & The Future of Interactive Content,” Adobe Blog (blog), 25 de julio, 2017, <https://theblog.adobe.com/adobe-flash-update/>. [↑](#footnote-ref-12)
13. “Thoughts on Flash - Apple,” revisado en 20 de diciembre 2017, <https://www.apple.com/hotnews/thoughts-on-flash/>. [↑](#footnote-ref-13)
14. El Diseño Centrado en el Usuario (UCD) es una filosofía de diseño que tiene por objeto la creación de productos que resuelvan necesidades concretas de sus usuarios finales, consiguiendo la mayor satisfacción y mejor experiencia de uso posible con el mínimo esfuerzo de su parte. 8 de octubre, 2017, <https://es.wikipedia.org/wiki/Dise%C3%B1o_centrado_en_el_usuario>

    El término “Diseño Centrado al Usuario” (UCD por sus siglas en inglés) se originó en el laboratorio de Donald Norman en la Universidad de California San Diego (UCSD) en los 80's y se convirtió en un término muy usado después de la publicación del libro *User-Centered System Design: New Perspectives on Human-Computer Interaction* (Norman & Draper, 1986).

    Una de las ideas esenciales del UCD es considerar el diseño como un proceso de varias fases de desarrollo y sus principios deben ser aplicados en todo el proceso del diseño, desde la fase de planificación hasta la fase de monitorización y análisis. [↑](#footnote-ref-14)
15. Los elementos audiovisuales no son técnicamente insertado en una página HTML, sino enlazado a ella. Estos elementos puede ser etiquetado en HTML como <img>, <audio> y <video> (las dos últimas son nuevas en HTML 5) creando un espacio para ellos. Son reconocidos como elementos en línea (relativamente con elementos en bloque) pero no indica insertar una línea, sino que otros elementos como Texto puede envolver alrededor de ella. Puede ajustar su ubicación o mejor dicho, su alineación a través de CSS. [↑](#footnote-ref-15)
16. “Eliel Saarinen,” Wikipedia, 10 de diciembre, 2017, <https://en.wikipedia.org/w/index.php?title=Eliel_Saarinen&oldid=814678509>. [↑](#footnote-ref-16)