

UNIVERSIDAD DE ALCALÁ



Escuela Politécnica Superior

**MÁSTER UNIVERSITARIO EN INGENIERÍA DEL
SOFTWARE PARA LA WEB**

Trabajo Fin de Máster

**ESTUDIO DE EXPERIENCIA DE USUARIO INMERSIVA
APLICADA A PÁGINAS WEB**

Juan Antonio Muñoz Gómez

2021/2022

UNIVERSIDAD DE ALCALÁ

Escuela Politécnica Superior

MÁSTER UNIVERSITARIO EN

INGENIERÍA DEL SOFTWARE PARA LA WEB

Trabajo Fin de Máster

“ESTUDIO DE EXPERIENCIA DE USUARIO INMERSIVA APLICADA A PÁGINAS WEB”

Autor: **JUAN ANTONIO MUÑOZ GÓMEZ**

Director: **LUIS DE MARCOS ORTEGA**

Tribunal:

Presidente:

Vocal 1º: Vocal 2º:

Calificación:

Fecha: de de

A mi familia, en especial a mis padres y abuelos. A mis amigos y personas cercanas. A todos ellos, gracias por acompañarme en este camino. De corazón.

ÍNDICE RESUMIDO

| | |
|--|----|
| 1. INTRODUCCIÓN..... | 8 |
| 2. OBJETIVOS DEL PROYECTO | 6 |
| 3. ESTADO DEL ARTE | 8 |
| 4. USABILIDAD WEB | 22 |
| 5. ASPECTOS PRÁCTICOS DE LA UX INMERSIVA | 30 |
| 6. RESUMEN Y CONCLUSIÓN..... | 69 |
| 7. BIBLIOGRAFÍA Y REFERENCIAS | 73 |
| APÉNDICE A. GLOSARIO | 82 |

ÍNDICE DETALLADO

| | |
|--|-----------|
| 1. INTRODUCCIÓN | 8 |
| 2. OBJETIVOS DEL PROYECTO | 6 |
| 3. ESTADO DEL ARTE | 8 |
| 3.1. USABILIDAD EN LA WEB | 9 |
| 3.2. UX INMERSIVA: CASOS REALES | 11 |
| 3.2.1. <i>Frame-By-Frame Scroll: Apple</i> | 12 |
| 3.2.2. <i>Efecto Parallax: Samsung</i> | 13 |
| 3.2.3. <i>3D Mapping: Marseille LP5</i> | 14 |
| 3.2.4. <i>Objeto 3D: SKOLKOVO PARK</i> | 15 |
| 3.2.5. <i>Otros ejemplos: Kopke1638</i> | 16 |
| 4. USABILIDAD WEB | 22 |
| 4.1. JAKOB NIELSEN: PRINCIPIOS DE LA USABILIDAD | 24 |
| 4.2. MÉTRICAS DE USABILIDAD | 27 |
| 4.2.1. <i>Evaluación heurística</i> | 27 |
| 4.2.2. <i>Recorrido Cognitivo</i> | 27 |
| 4.2.3. <i>Recorrido de usabilidad pluralista</i> | 28 |
| 4.2.4. <i>Inspección formal de usabilidad</i> | 28 |
| 5. ASPECTOS PRÁCTICOS DE LA UX INMERSIVA | 30 |
| 5.1. DISEÑO DE INTERFACES DE USUARIO: UI/UX | 30 |
| 5.1.1. <i>Importancia de UX</i> | 31 |
| 5.2. UX INMERSIVA: ¿QUÉ ES? | 34 |
| 5.3. RECURSOS INMERSIVOS | 36 |
| 5.3.1. <i>Animaciones</i> | 36 |
| 5.3.2. <i>Elementos 3D</i> | 40 |
| 5.3.3. <i>Textos</i> | 44 |
| 5.3.4. <i>Controles periféricos</i> | 44 |
| 5.3.5. <i>Microinteracciones</i> | 46 |
| 5.3.6. <i>Temas</i> | 47 |
| 5.4. EJEMPLOS PRÁCTICOS | 54 |
| 5.4.1. <i>Parallax + Mouse Following</i> | 54 |
| 5.4.2. <i>Parallax + scrolling</i> | 55 |
| 5.4.3. <i>Horizontal scrolling</i> | 57 |
| 5.4.4. <i>Click Counter</i> | 58 |
| 5.4.5. <i>Drag & Drop</i> | 58 |
| 5.4.6. <i>Objetos 3D</i> | 59 |
| 5.4.7. <i>3D Mapping</i> | 61 |
| 5.4.8. <i>Temáticas</i> | 65 |
| 6. RESUMEN Y CONCLUSIÓN | 69 |
| 7. BIBLIOGRAFÍA Y REFERENCIAS | 73 |
| APÉNDICE A. GLOSARIO | 82 |

ÍNDICE DE FIGURAS

| | |
|--|----|
| FIGURA 1. FRAMES DE LA ANIMACIÓN “ <i>FLIP-REVEAL-GUTS</i> ” DEL PRODUCTO AIRPOD | 12 |
| FIGURA 2. ESTADO FINAL DEL PRODUCTO SAMSUNG Z FLIP4 | 14 |
| FIGURA 3. ESTADO FINAL DEL PRODUCTO SAMSUNG Z FLIP4 | 14 |
| FIGURA 4. MAPA TRIDIMENSIONAL DE LA PÁGINA MARSEILLE LP5..... | 15 |
| FIGURA 5. PARCELA 3D DEL SKOLKOVO PARK | 16 |
| FIGURA 6. RECURSO "DRAG & DROP" EN LA WEB KOPKE1638..... | 17 |
| FIGURA 7. ESTADO INICIAL DEL MENSAJE "MAJOR PRODUCER" | 18 |
| FIGURA 8. ESTADO FINAL DEL MENSAJE "MAJOR PRODUCER" | 18 |
| FIGURA 9. MAPA TRIDIMENSIONAL EN LA WEB KOPKE1638 | 19 |
| FIGURA 10. TEXTO IMPLEMENTADO EN 3D EN LA WEB KOPKE1638 | 20 |
| FIGURA 11. SHOWCASING DE LA APLICACIÓN EN LA PARTE DERECHA DE LA IMAGEN..... | 40 |
| FIGURA 12. FORMULARIO CON ESTILO ESQUEUMÓRFICO [66] | 48 |
| FIGURA 13. BOTONES CON ESTILO ESQUEUMÓRFICO [67] | 48 |
| FIGURA 14. APLICACIÓN MÓVIL CON ESTILO NEUMÓRFICO [70]..... | 49 |
| FIGURA 15. BOTONES CON ESTILO NEUMÓRFICO [71]..... | 50 |
| FIGURA 16. PÁGINA WEB CON ESTILO MINIMALISTA [74]..... | 51 |
| FIGURA 17. PÁGINA WEB CON ESTILO GLASSMÓRFICO [76]..... | 52 |
| FIGURA 18. APLICACIÓN MÓVIL CON ESTILO CLAYMÓRFICO..... | 53 |
| FIGURA 19. CASO PRÁCTICO OBJETO 3D - PROYECTO <i>MOLANG 3D</i> DE SPLINE..... | 59 |
| FIGURA 20. EJEMPLO PRÁCTICO DE MAPBOX - VISTA DE CERCA DE LA CIUDAD DE SALAMANCA..... | 62 |
| FIGURA 21. EJEMPLO PRÁCTICO DE MAPBOX - MARCADORES | 64 |
| FIGURA 22. EJEMPLOS PRÁCTICOS DE RELIEVE Y OQUEDAD CONSEGUIDOS CON NEUMORFISMO..... | 66 |
| FIGURA 23. EJEMPLO PRÁCTICO DE CLAYMORFISMO..... | 67 |
| FIGURA 24. EJEMPLO PRÁCTICO DE GLASSMORFISMO | 67 |

1. INTRODUCCIÓN



Todos los primates, incluyéndonos por supuesto a los seres humanos, somos criaturas altamente “visuales”. Solemos guiarnos en nuestro día a día por señales ópticas que nos avisan de posibles peligros o hallazgos en temas tan vitales como comida, refugio o incluso en nuestras relaciones sociales [1]. La información masiva a la que estamos bombardeados ha provocado una evolución en esta faceta de la que se puede sacar provecho en el mundo digital, y en concreto en el amplio mundo de internet y las páginas web. Hoy en día el consumo de tecnologías relacionadas con el desarrollo web está en su máximo apogeo. La gran mayoría de usuarios acceden a múltiples páginas web a lo largo del día, haciendo que reciban mucha, quizás demasiada, información en periodos de tiempo breves. Este inconveniente visto desde el lado empresarial saca a la luz la problemática de que los usuarios no dedican el suficiente tiempo y concentración para asimilar los productos, retenerlos en memoria y prestarles atención, y por tanto, no generen los activos que la empresa necesita: ingresos y publicidad.

El gran problema de las interfaces web actualmente es hacer páginas que retengan a los usuarios el suficiente tiempo como para que cale el mensaje que se quiere dar [2] [3]. El auge de nuevos movimientos de marketing (personalización, automatización, psicología del color, mensajes en forma de *banners/beros*, temáticas u otro tipo de experiencias, entre muchas otras) [4] [5] donde se recogen campos como el diseño gráfico, la arquitectura web, el desarrollo 3D o por supuesto experiencias únicas de control e interacción con elementos de la página, han hecho que la inmersividad web sea una tendencia en el concepto de desarrollo de la usabilidad orientada a la web.

La manera de aplicar estos movimientos de marketing a las páginas web se hace principalmente a través de la capa de presentación, denominada comúnmente como *frontend*. Esta capa se encarga de estilizar la página de manera que se pueda presentar la información al usuario de forma agradable para su interacción. Para lograr que sea agradable se han de aplicar principios relacionados con la *usabilidad*, donde se estudie el comportamiento de los consumidores con los diferentes elementos implementados en la página. [6]

Esta capa de presentación ha evolucionado desde los albores de internet hasta el resultado que hoy podemos ver en nuestros navegadores. Partiendo del final de la década de los 90, cuando se popularizó el uso de internet en masa por parte del público consumidor, se puede ver una clarísima diferencia con las interfaces actuales. La complejidad cada vez mayor de las hojas de estilo y lenguajes de programación, principalmente en las posibilidades que estos ofrecen a nivel computacional, han hecho que tanto la organización de la información como el formato de presentación sean el objetivo principal de aquellas compañías que han visto su mercado intensivamente sobredimensionado por la corriente de la globalización. Influenciados inconscientemente por las tendencias predominantes del día a día, los consumidores han ido detallando la hoja de ruta de las interfaces a través del tiempo. Por supuesto, gran parte de la ‘culpa’ la tienen la evolución de las tecnologías y la necesidad de focalizar el mensaje debido a la sobrecarga de información a la que son bombardeados los usuarios segundo tras segundo. Por eso es tan importante la manera en la que esa información se presenta al usuario y es asumida e interiorizada, siempre con el objetivo de obtener ambas partes, usuarios y compañías, beneficios de la interacción con las plataformas web de manera eficiente y cómoda.



En este trabajo se quiere realizar un estudio de estas tendencias englobadas dentro del campo de la usabilidad, y en concreto de la *experiencia de usuario inmersiva*, concepto que es cada vez más requerido por los equipos de desarrollo de usabilidad y de frontend. Se realizará un estudio de los diferentes recursos utilizados actualmente por algunas compañías, seguido de un marco teórico de los conceptos clave de este movimiento como son la usabilidad y la experiencia de usuario inmersiva. Para este último también se definirán tecnologías disponibles con el objetivo de lograr la inmersividad, y se tratará de poner en práctica algunas de las técnicas con el fin de realizar un ejemplo práctico de las mismas. Para finalizar, se pondrán en valor algunas de las posibles aplicaciones en las que se pueden ver reflejadas las diferentes técnicas, como pueden ser historias inmersivas o la joven promesa del cada vez más creciente metaverso. Se sigue este orden para en primer lugar conocer casos prácticos de los que empezar a sacar información. A continuación, se procede a ampliar los conocimientos del campo de la inmersividad web, para finalmente realizar una explicación detallada de los recursos que se contemplan para lograr la inmersividad, incorporando ejemplos prácticos de los mismos.

2. OBJETIVOS DEL PROYECTO



El desarrollo de páginas web ha sufrido una clara evolución en los últimos tiempos. El nacimiento de librerías, frameworks y plugins ha hecho muy accesible la confección de elementos amigables a la par que personalizables, lo que da como resultado infinitas posibilidades de elementos, comportamientos y sobre todo de maneras de captar la atención del usuario.

El principal objetivo de este trabajo es analizar **cómo interactúa el usuario con la página y sus elementos**, saber cuál es la razón de que el usuario gaste más tiempo en nuestra página que en otra de la competencia, o viceversa. En definitiva, razonar por qué el comportamiento humano contestará con más atención a ciertos elementos o interacciones. Algunas de estas técnicas a analizar recogen elementos 3D, disposición de elementos a través de la navegación de la plataforma, o efectos visuales como animaciones, *parallax scrolling* o *cardboarding showcases*. Otro tipo de herramientas en las que se apoyan pueden ser formularios personalizados para el usuario, fomentados por el uso de *psicología* de los mensajes y combinaciones de colores o focalización en la cantidad información desplegada. Por último pero no menos importante, ha de tenerse muy en cuenta la posibilidad de interactuar con los objetos de la página mediante control de eventos provocados por el usuario (clicks, drags & drops o scrolls entre otros). Será objetivo de este documento analizar la respuesta de los usuarios frente a estos recursos.

Todas estas técnicas suelen verse complementadas y puestas en valor gracias a los diferentes estilos predominantes en la última década. Ejemplo de esto es la tendencia del *esqueumorfismo* con la que se apela a los recuerdos del usuario mediante interfaces similares a objetos convencionales, o la tendencia del *neumorfismo*, estilo donde predominan colores pastel sobre elementos suavizados, consiguiendo de esta manera construir interfaces amigables para el usuario. Analizar estas tendencias en conjunto con los elementos será clave para entender el comportamiento de los usuarios en las páginas web.

A fin de cuentas, se quiere realizar un **estudio de las corrientes de UX más llamativas para los usuarios**, poniendo en manifiesto su importancia y su proceso de elección, justificación y posterior desarrollo en el marco web, concretamente en la capa de visualización denominada *frontend*. Para lograr esto se pretende recoger la suficiente información sobre por qué estas técnicas están en auge y cómo son beneficiosas para captar la atención de los usuarios. El objetivo es ayudar a sintetizar directrices y tendencias en la parte del frontend para que desarrolladores aprendan a mejorar sus capacidades y oferten un producto más llamativo al usuario.

La importancia de estas técnicas está ligada más a los aspectos de marketing y dirección de ventas en las empresas que al desarrollo en sí de los productos, pero, en este documento, se quiere dar importancia al proceso de implementación de estos elementos sin obviar las ventajas que traen al producto al que van ligadas. Por tanto, no se va a priorizar la explicación del alcance de una tendencia o recurso como tal, si no la posible aplicabilidad de ese artefacto viablemente en un proyecto web, de forma que el resultado obtenido dé el éxito buscado en todas las fases del proyecto, desde la etapa de creación del prototipo hasta la de producción y despliegue.



Tras todo lo expuesto, el objetivo principal es **analizar la importancia de la experiencia de usuario inmersiva dentro del campo de la usabilidad web**, desarrollando principalmente los siguientes sub-objetivos por el camino:

- Estudio del comportamiento del usuario a través de la interacción con los elementos de la página
- Definición y catalogación de la usabilidad y su envergadura en el diseño frontend
- Recopilación de tendencias y tecnologías para la creación de elementos inmersivos
- Analizar los resultados de la aplicación de elementos inmersivos en casos prácticos

3. ESTADO DEL ARTE



En este primer capítulo nombrado *Estado del Arte* se pretende ofrecer un resumen de conocimientos aplicados a la práctica relacionados con la inmersividad, aportando una breve base teórica y justificación en función de aquellas tendencias que están presentes en el mercado actualmente. Se explicará qué es la “Usabilidad” y la “Experiencia de Usuario Inmersiva”, qué importancia tienen y cuál es su potencia de uso en el mercado actualmente. A continuación, se listarán ejemplos reales de qué técnicas están siendo utilizadas y cómo se está planteando la utilización de las mismas.

Con estas definiciones y ejemplo ya podremos enmarcar el camino que ha de seguir este análisis de la experiencia de usuario inmersiva en páginas web.



3.1. Usabilidad en la web

Sin entrar mucho en detalle en definiciones y tecnicismos – eso tocará más adelante – es menester comentar la importancia de la usabilidad en el ámbito del desarrollo web. En resumen, cuando mencionamos el término usabilidad hablamos de interfaces amigables y bien estructuradas. Capas de presentación que, siguiendo unos patrones y estándares establecidos, bien por organizaciones o bien simplemente aceptados por la comunidad, han de facilitar al máximo el entendimiento y la navegación de todos los tipos de usuarios posibles sin tener en cuenta sus cualidades ni sus conocimientos previos.

Antes de avanzar en materia, se quiere dejar claro que siempre que se habla de “ámbito web”, “páginas web”, o cualquier otro homólogo, se pretende hacer referencia también a interfaces móviles o asociadas a otro tipo de dispositivos donde puedan ser desplegados elementos web, ya sea de manera nativa o híbrida. Ejemplo de este otro tipo de dispositivos pueden ser pantallas encontradas en el campo de IoT, como cajeros, altavoces inteligentes como *Amazon Echo Show* o *Google Nest*, coches, GPS o incluso frigoríficos. En definitiva, cualquier dispositivo que posea una pantalla y pueda hacer uso de navegación y/o despliegue de tecnologías web.

¿Por qué es tan importante la usabilidad en la web? ¿Merece la pena dedicar tanto tiempo a algo que quizás solo le demos importancia nosotros como diseñadores/desarrolladores? Quizás la respuesta sorprenda, pero sí, es importante, muy importante. Tanto que, si hiciéramos dos páginas web con exactamente el mismo contenido, pero una no siguiera ninguno de los estándares de usabilidad y la otra sí, nos daríamos cuenta en seguida de que la primera de ellas sería inmensamente más difícil de utilizar que la segunda. La correcta organización y presentación de los contenidos es vital para que el usuario interactúe con la página. Gracias a ello, se conseguirá que el contenido cale de manera más eficaz en el consumidor del mismo.

Moviéndonos a un entorno asociado con la popularmente denominada “industria digital”, se divulga que aquellos elementos que conforman la página web (*UI*) de una manera *usable*, han de generar una experiencia general que el usuario ha de sentir como agradable, cómoda y fácil de recordar y por tanto de utilizar a primera vista. El término escogido para esta definición es **Experiencia de Usuario (UX)**. Término que muchas veces se ve difuminado por el ámbito “marketiniano” de los despachos y las reuniones de equipos de trabajo y clientes. No obstante, esa experiencia de usuario que se busca, muchas veces, se ve equilibrada con la cantidad de trabajo dedicado por parte del equipo desarrollador. La reutilización de elementos como por ejemplo ofrecen frameworks, librerías, plugins y otro tipo de conglomerados de herramientas, obtenibles tanto públicamente (repositorios, frameworks públicos...) como desarrollados por el propio equipo, es una técnica que ahorra mucho tiempo en fase de implementación. Normalmente no se necesita crear algo novedoso, algo tan innovador para el usuario que el mero hecho de que se acceda a la página simplemente lo retenga en ella. Simplemente se requiere de un proceso de implementación eficaz, desplegar algo que los desarrolladores ya tienen a mano y conocen cómo llevarlo a cabo y personalizarlo en un tiempo asequible. Quizás es porque el equipo de desarrollo no elige el contenido de la plataforma, si no que elige la manera en la que muestra contenido que ellos no pueden escoger: publicaciones, mensajes, fotos, productos, etcétera. Pero, cuando se da la maravillosa situación de que sí se puede elegir el contenido, y de verdad se pretende hacer algo innovador para mostrarlo, se crea una simbiosis entre desarrolladores y diseñadores que da como resultado creaciones auténticamente sorprendentes.



En ese momento, la experiencia de usuario asciende al siguiente nivel. Estructuras de elementos más complejas, interacciones innovadoras, métodos lúdicos de descubrimiento del producto. En definitiva, lo que se conoce como “saberse vender”, y muy bien de hecho. Es la definición del marketing de productos extrapolada al desarrollo de interfaces web. Un producto, una idea, un mensaje, pero distinto al resto de productos, al resto de ideas, al resto de mensajes.

La personalización de todos los elementos de la interfaz, lo que podríamos englobar dentro del término *User Interface*, puede llegar a ofrecer un proceso de interacción (*User Experience*) tan novedoso que quede fuera de lo común, de lo que estamos acostumbrados a ver. Surge por tanto una experiencia de usuario especial, una experiencia donde el usuario se sumerja en una historia que el equipo de diseño y desarrollo ha querido contar. Una experiencia **immersiva** para el usuario.



3.2. UX inmersiva: Casos Reales

¿Qué es exactamente la experiencia de usuario (UX) inmersiva? Haciendo referencia a la RAE, dicese de *inmersiva* « que hace vivir al espectador una realidad virtual como si fuera auténtica ». Siguiendo esta tesis, entendemos que no estamos tratando de páginas web comunes. Y, como no puede ser de otra forma, páginas no comunes han de estar formadas por elementos no comunes. Hágase hincapié en la definición de *común* por medio de la RAE nuevamente: « Ordinario, vulgar, frecuente y muy sabido ». De esta definición se puede sacar en claro que estamos ante elementos no ordinarios, no frecuentes. Elementos que no se ven en la inmensa mayoría de los espacios web que acceden los usuarios. ¿Por qué estos elementos son tan raros de encontrar? Bien, procedamos por pasos.

Para comenzar, ha de dejarse claro que este tipo de elementos llevan una doble complejidad. Han de saber usarse en el momento y lugar adecuado, tarea que reside en los equipos de diseño, y han de saber implementarse de manera correcta, eficaz y sin perder rendimiento en el manejo de la web, tarea por supuesto de los desarrolladores informáticos. Estos motivos hacen que haya que invertir muchos recursos en algo que, quizás, los usuarios no dan demasiada importancia. Puede suceder que no sea relevante para la navegación web, o simplemente que estorbe más que facilite la comercialización y acceso al producto por parte de los usuarios.

Probablemente, el mayor de los hándicaps que traen estos elementos es la complejidad para llevarlos a cabo en la fase de implementación. La gran mayoría de desarrolladores web no están acostumbrados a trabajar de una manera tan artística como se requiere en este tipo de interfaces, si no de una más funcional y resolutive. El hecho de que este tipo de frontends sean un trabajo de sastre, meticuloso y laborioso, hace que la gran mayoría de soluciones adoptadas no incluyan elementos inmersivos.

Volvamos a la materia. Reincidiendo en la pregunta: ¿Qué es exactamente la **UX inmersiva**? Bueno, pues es aquella rama de la experiencia de usuario que recoge el uso de elementos inmersivos. Por decirlo así, es la sensación de sumergimiento virtual de los usuarios causada por elementos especiales que se rigen por unas normas de orden enmarcadas en la usabilidad web. La verdad es que es un campo emergente, no ampliamente documentado en la actualidad. Etiquetando de “exóticos” entre la inmensidad de la nube aquellos espacios con repertorio catalogable como inmersivo, podríamos recoger las pautas de usabilidad (apartado **4. Usabilidad Web**) marcadas por J. Nielsen y aplicarlas a cada una de las secciones. Pero el proceso no es tan trivial, ni mucho menos. Si bien siguen siendo páginas web, y por supuesto han de cumplir con rigurosidad los estándares de usabilidad (suponiendo que quieran maximizar el número de usuarios y por tanto de potenciales clientes en su plataforma), a veces algunas de las heurísticas quedan aparcadas a un lado para potenciar visualmente el mensaje a transmitir. El objetivo principal de la UX inmersiva es por tanto transmitir el mensaje de una manera no tan convencional, de una manera visual por encima de la corriente textual más “ordinaria”. La cuestión ahora no es qué es la usabilidad o qué es la UX si no, ¿Cómo conseguimos la inmersividad?

Los **elementos inmersivos** serán aquellos recursos que nos ayudarán a embeber al usuario dentro del mundo virtual desplegado en nuestra plataforma. Se puede decir por tanto que son los culpables de esta rama de la UX dentro de la usabilidad. Elementos interactivos tales como objetos *clickables* y *arrastrables* en pantalla, imágenes que sigan el movimiento del ratón, objetos o componentes mostrados en 3D, despliegue de mapas en vistas picadas, modificación de controles ya asimilados por el usuario como scrolls, clicks o incluso modificación de elementos también conocidos como menús, botones, etcétera. En conclusión, elementos

que muestren una idea de manera visual e interactiva por encima de la vertiente textual ya conocida y utilizada en la práctica totalidad de las páginas que accedemos en nuestro día a día.

Haciendo una búsqueda de estos elementos, se ha llegado a la conclusión de que grandes compañías como Apple, Samsung o Google entre otras utilizan este tipo de elementos para publicitar sus productos. Pero, como no podía ser de otra manera, se ha querido indagar en páginas no tan conocidas para ver si los elementos de inmersión utilizados coinciden con los de las grandes compañías, o sin embargo optan por otro tipo de soluciones más accesibles para el resultado que esperan.

A continuación se citan ejemplos que se han considerado claros para entender un poco mejor algunos de los elementos de inmersión más comunes.

3.2.1. Frame-By-Frame Scroll: Apple

El primer caso que ha de ser comentado es la página web del gigante mundial Apple. Si entramos en su página o en cualquiera de las subpáginas de los productos, se notará que hay una dedicación especial a la hora de mostrarlos y hacerlos llegar al *target* de usuarios. Poniendo como ejemplo de inmersividad la página del producto AirPods [7], lo primero que llama la atención es el efecto al hacer scroll, técnica asociada al desplazamiento del usuario por la página actual. Analizando qué es lo que está sucediendo gracias al visor de elementos, se puede detallar un doble comportamiento del scroll. Se mantendrá un comportamiento de movimiento vertical entre secciones, donde se incluirá por medio del atributo HTML “*data-anim-scroll-group*” una modificación del comportamiento a la hora de hacer scroll in en pantalla. Además de este, se crea una conducta nueva por medio de JavaScript que hace que cuando una sección está ocupando el completo de la pantalla (*full-size*) se despliegue una animación muy visual del producto, que en realidad no es nada más que frames (fotogramas) de un video, donde se añaden textos con el comportamiento modificado por el mismo atributo mencionado antes.

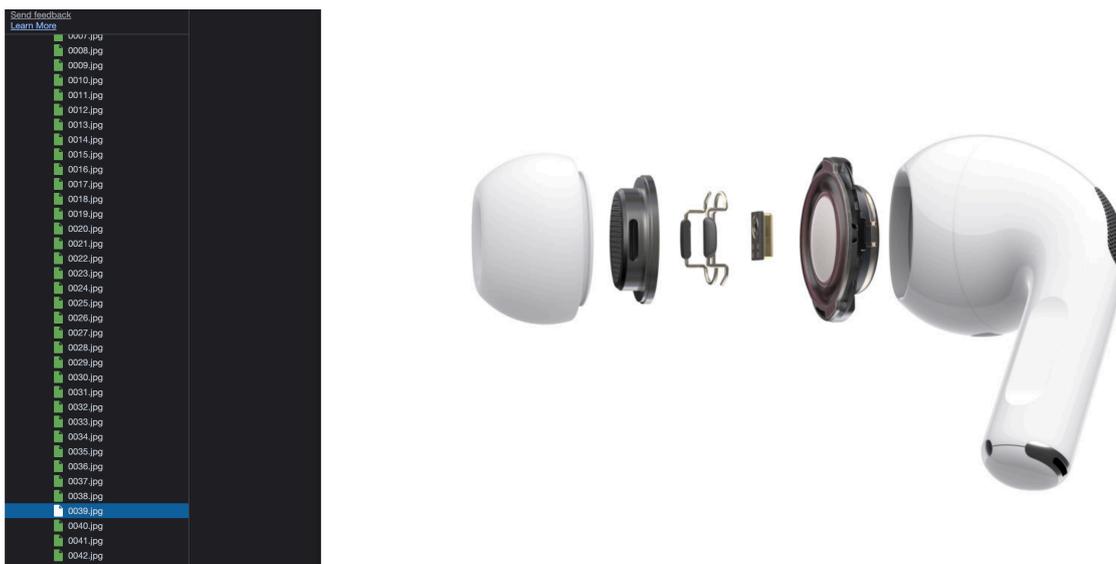


Figura 1. Frames de la animación “*Flip-Reveal-Guts*” del producto AirPods



Esta técnica de simular un desplazamiento inmersivo con una sucesión de frames (*5.3.1.1 Scrolling*), emulando la técnica con la que se creaban los primeros dibujos animados, es utilizada en todos los productos de Apple. Siendo concisos, el vídeo que hace el departamento de diseño gráfico o marketing, es transformado a imágenes. Estas imágenes se almacenan en el repositorio de la página y, gracias al control del **DOM** que permite JavaScript, se despliegan los distintos frames en función de la distancia que haya sido recorrida con el scroll. Pero, ¿merece la pena todo este trabajo? ¿Conseguiremos llamar la atención del usuario? Pues la verdad es que sí, merece mucho la pena. Romper barreras de interacción hace que no se dependa tanto de barreras idiomáticas: el mundo de las imágenes no requiere traducción [8] También hace mantener a los usuarios en nuestra página, ideal para un control de analíticas tanto para nosotros como para sponsors o espacios publicitarios que haya en la página. No hay que olvidar que una buena experiencia de usuario servirá, además de captar la atención de este usuario, para que este mismo difunda nuestra idea y por tanto nuevos posibles clientes accedan a consultar, bien por curiosidad o bien por necesidad, el producto del que han oído hablar. Es bien sabido que la publicidad es una de las armas más potentes de Apple, y así se refleja en su página web.

¿Qué es lo que se puede aprender de Apple? Su estrategia de contenido es abrumadora, sorprendiendo a los usuarios nuevos y quizás a los no tan nuevos. Esa facilidad de vender de manera visual una idea es la definición ideal de inmersividad. Además, la fragmentación de los mensajes es excelente para que calen los mensajes que la compañía quiere que sean retenidos consciente o inconscientemente [9].

3.2.2. Efecto Parallax: Samsung

La página de Samsung, a fecha de escritura de este documento, incorpora la campaña publicitaria del nuevo producto de la compañía *Samsung Z Flip4* [10], cuarta generación del dispositivo móvil plegable del gigante coreano. Anecdóticamente, este producto es un reto para la usabilidad, accesibilidad y responsividad de las interfaces por sus múltiples posibilidades de uso debido a su pantalla plegable, pero ese no es el tema que ha de tratarse. Volviendo a lo que nos importa para este documento, se quieren relatar las técnicas que utiliza Samsung para mostrar su producto al público.

Una de las técnicas que más destacan en esta web es la diferencia de desplazamiento a la hora de hacer scroll por la página. Se conoce esta técnica como *Efecto Parallax* (*5.3.1.1 Scrolling*). Se identifica este efecto al comprobar como distintas capas de la página quedan en distinta posición entre sí teniendo en cuenta sus posiciones inicial (se muestran en pantalla) y final (salen de la pantalla). En la imagen a continuación se puede ver el estado inicial del texto “Flexibilidad para tus fotografías”, teniendo un pequeño margen con respecto a la foto que tiene debajo. El efecto en esta ocasión se aplica en el eje vertical.



CÁMARA ●

Flexibilidad para tus habilidades fotográficas

Figura 2. Estado final del producto Samsung Z Flip4

Se compara la imagen de arriba con la de abajo, donde se puede apreciar que la foto ha recorrido mayor distancia al hacer scroll en la página, solapándose con el texto mencionado anteriormente. Este efecto da dinamismo a los diferentes elementos de la página, simulando distintas capas con menor o mayor proximidad.

fotográficas



Figura 3. Estado final del producto Samsung Z Flip4

3.2.3. 3D Mapping: Marseille LP5

El uso de elementos 3D añade un dinamismo y una potencia visual digna de ser mencionada en este estado del arte. En este caso, el uso de la tridimensionalidad se ve materializado en mapas (*5.3.2.1 3D Mapping*). Pudiéndose aplicar tanto al relieve como al despliegue de edificios, es una herramienta muy potente para llamar la atención de los usuarios en la página web. En esta ocasión se ha querido documentar el mapa [11] realizado por el estudio *PHASE 5* para mostrar puntos turísticos de la ciudad de Marsella, Francia.

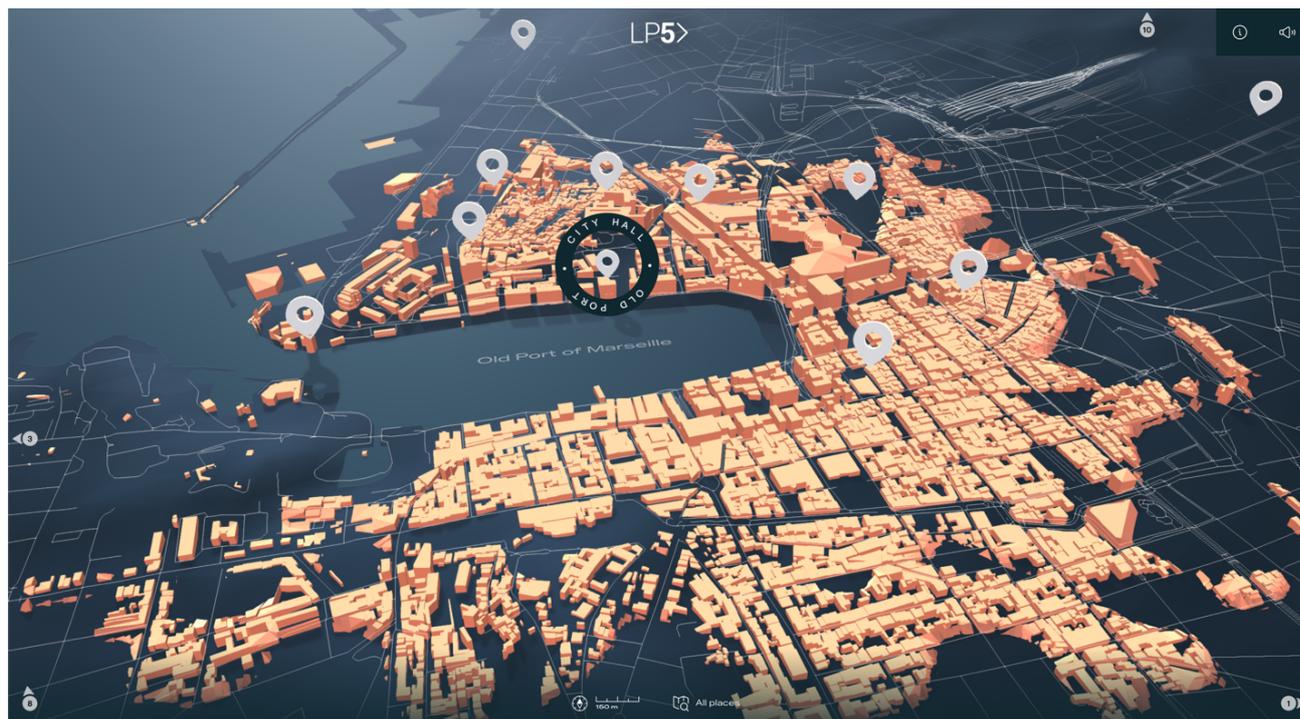


Figura 4. Mapa tridimensional de la página Marseille LP5

Una imagen vale más que mil palabras, y en esta ocasión queda demostrado cómo la presentación de la lista de puntos turísticos hace jugar al usuario, viajar por el mapa, en definitiva: Descubrir la ciudad. Es una representación ideal de la venta de un producto (la ciudad de Marsella) utilizando la inmersividad dentro de la web (mapeado tridimensional). En esta ocasión, se ha utilizado la herramienta *Mapbox*, un plugin potente de renderizado de mapas. Esto se puede consultar en la pestaña *Sources* del inspeccionador de elementos, en concreto en el fichero *chunk-vendors.194f113a.js*.

Considero conveniente matizar otro aspecto crucial para que la idea resulte tan atractiva: el contraste entre el azul marino del fondo y el naranja fosforito de los edificios hace que nos fijemos en los edificios, en las calles, inclusive en los marcadores, que verdaderamente se quiere que prestemos atención.

3.2.4. Objeto 3D: SKOLKOVO PARK

La utilización de objetos 3D (*5.3.2 Elementos 3D*). en páginas web se ha visto claramente incrementada en los últimos años. Son múltiples los motivos del auge de esta tecnología: La extravagancia de los mismos, La potencia, dedicación y profesionalidad (incluso elegancia si me lo permiten) que demuestra su correcto uso, y por supuesto la capacidad de renderización por parte de los navegadores que manejan los clientes.

Una llama peculiarmente la atención es Skolkovo Park [12], una página que muestra por medio de una proyección 3D las instalaciones que ofrece la marca moscovita para que subempresas hagan uso de ellas: campo de golf, bloques residenciales, facilidades sanitarias y educativas, y sobre todo, oficinas. Esta ciudad en miniatura incluye hasta tiendas, restaurantes e incluso carril bici. Una maravilla sin duda, aunque, para maravilla, la manera que han escogido de mostrar al público sus instalaciones.



El objeto 3D representa la parcela completa, donde, gracias al menú de la parte superior, podremos acercarnos a los distintos puntos de interés que se enumeran en el párrafo de arriba. Por supuesto, se nos deja libertad para rotar, cambiar el ángulo y la distancia de la cámara con respecto al objeto 3D.

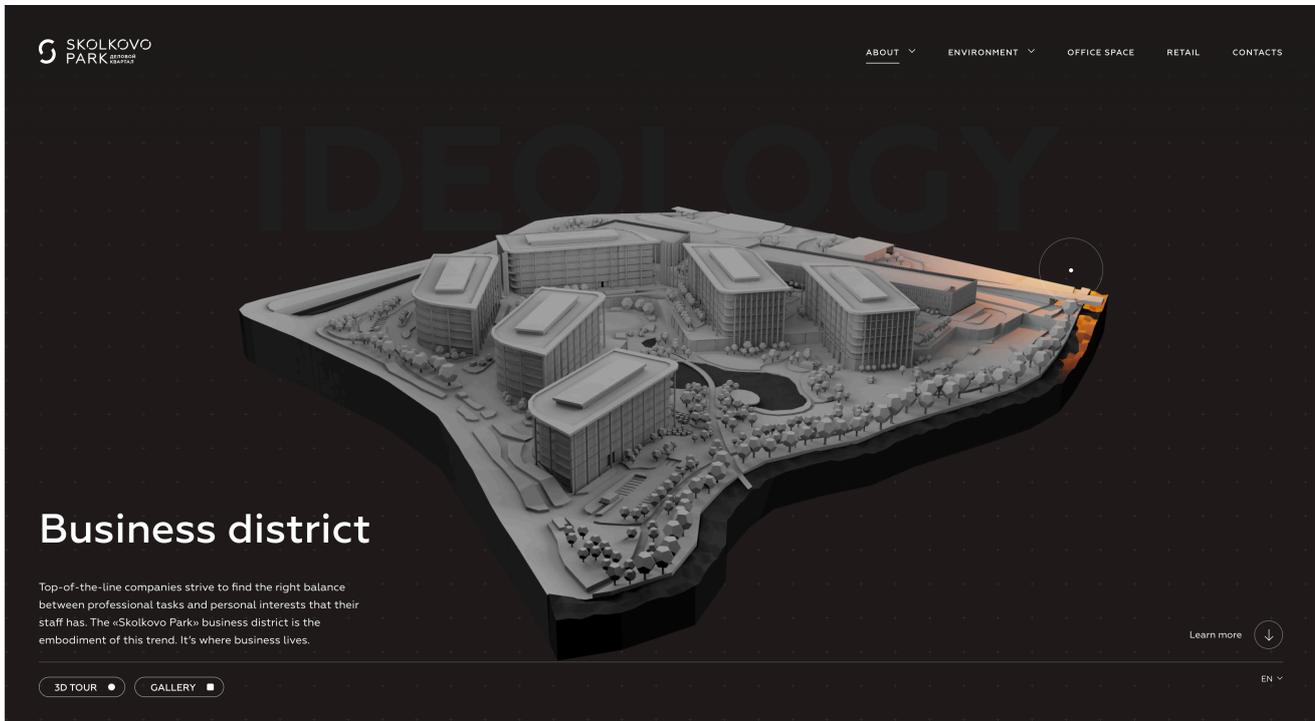


Figura 5. Parcela 3D del Skolkovo Park

Efectivamente una manera potente y elegante de hacer llegar a los usuarios lo que ofrecen. Además, el contraste naranja en el objeto que acompaña al cursor cuando el usuario lo mueve facilita el uso y el entendimiento de cómo funciona el control del mismo.

3.2.5. Otros ejemplos: Kopke1638

Hacer llamativa una página web de una bodega portuguesa creada en el siglo XVII por un Hamburgués es, sin duda, un reto digno de ser mencionado. Kopke1638 [13] muestra su historia, sus productos y sus instalaciones por medio de diferentes técnicas, algunas de ellas ya mencionadas en este apartado **3. Estado del arte**, pero otras de ellas no.

Para empezar, encontramos un *hero* donde se nos pide una interacción que no había sido mencionada todavía en el documento. Se nos pide agarrar un objeto y arrastrarlo hacia otra posición, o lo que es lo mismo, *Drag & Drop*. Junto con esta técnica, se utiliza un recurso que ya se había visto en este documento: separar un vídeo frame a frame. En este caso, el despliegue de cada uno de los frames se realiza en función de la posición del objeto a arrastrar.

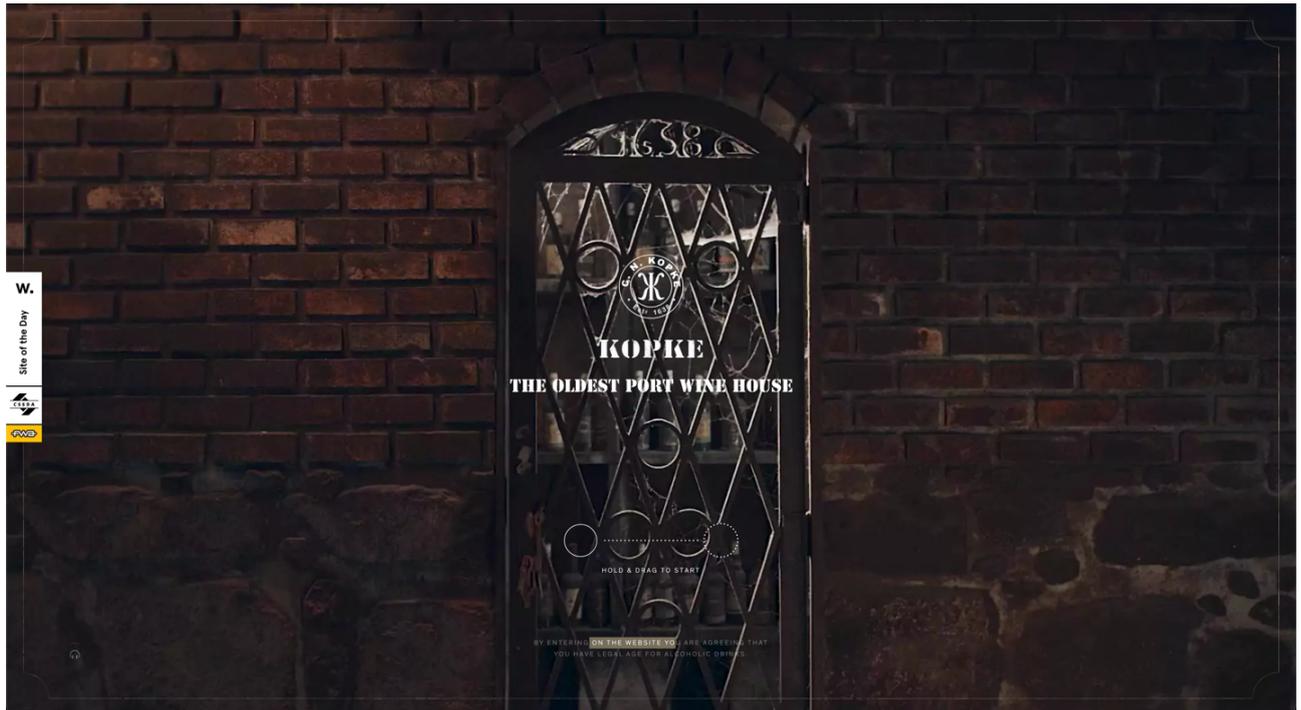


Figura 6. Recurso "Drag & Drop" en la web Kopke1638

Una vez hecho esto, accederemos dentro de la página, donde podremos movernos entre las distintas secciones bien de manera lineal, o bien mediante el menú. En la navegación por la página encontramos otra nueva técnica. La alteración del comportamiento del scroll para que el desplazamiento del contenido se haga de manera horizontal. Este efecto refleja una secuencia en los actos, como si de una historia se tratara, como si estuviéramos en un museo y cambiáramos de panel en la pared que tenemos en frente. En este desplazamiento horizontal se puede apreciar cómo aparece un viejo amigo, el *efecto parallax*. El uso de este efecto se puede ver en el texto "Major Producer". Para visualizar mejor el desfase entre el texto y el resto de elementos, se recomienda comparar su posición con respecto a las imágenes de las siguientes fotos:

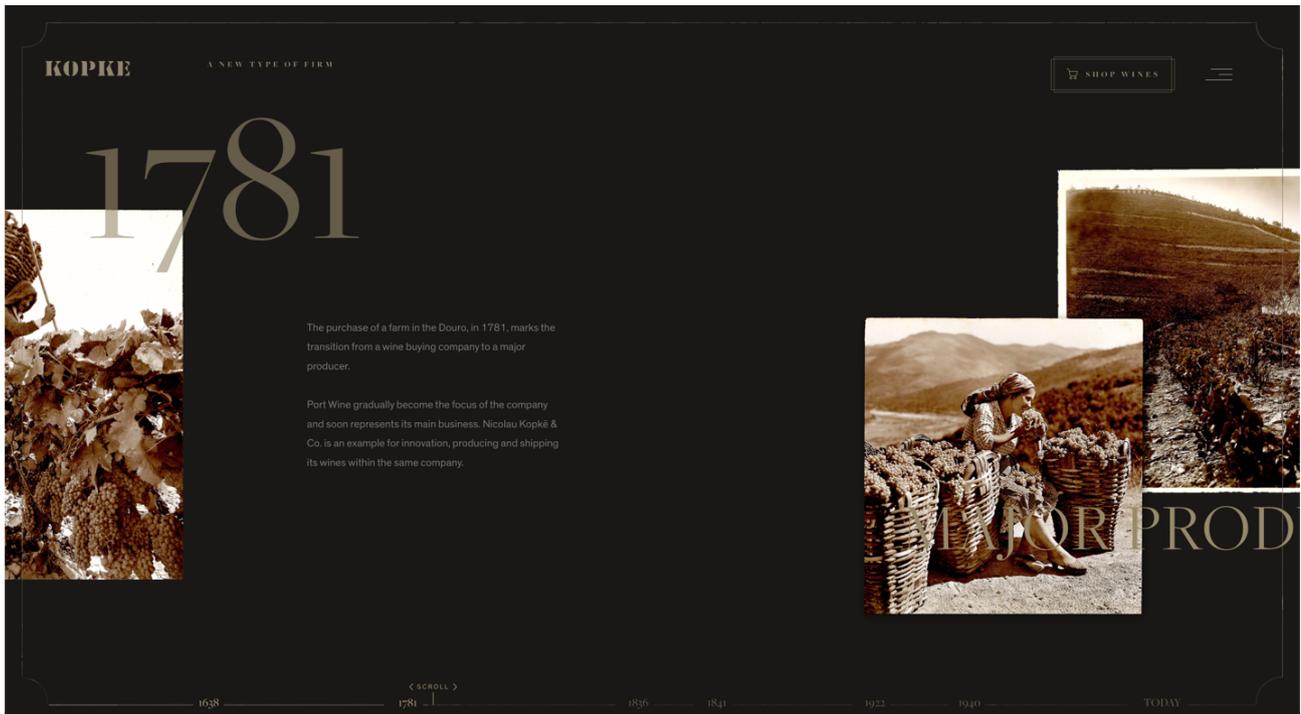


Figura 7. Estado inicial del mensaje "Major Producer"



Figura 8. Estado final del mensaje "Major Producer"



Otro de los recursos que aparecen en esta web, en concreto en la sección “Terroir”, es el uso de mapas tridimensionales, soportado de nuevo mediante la herramienta *Mapbox*.

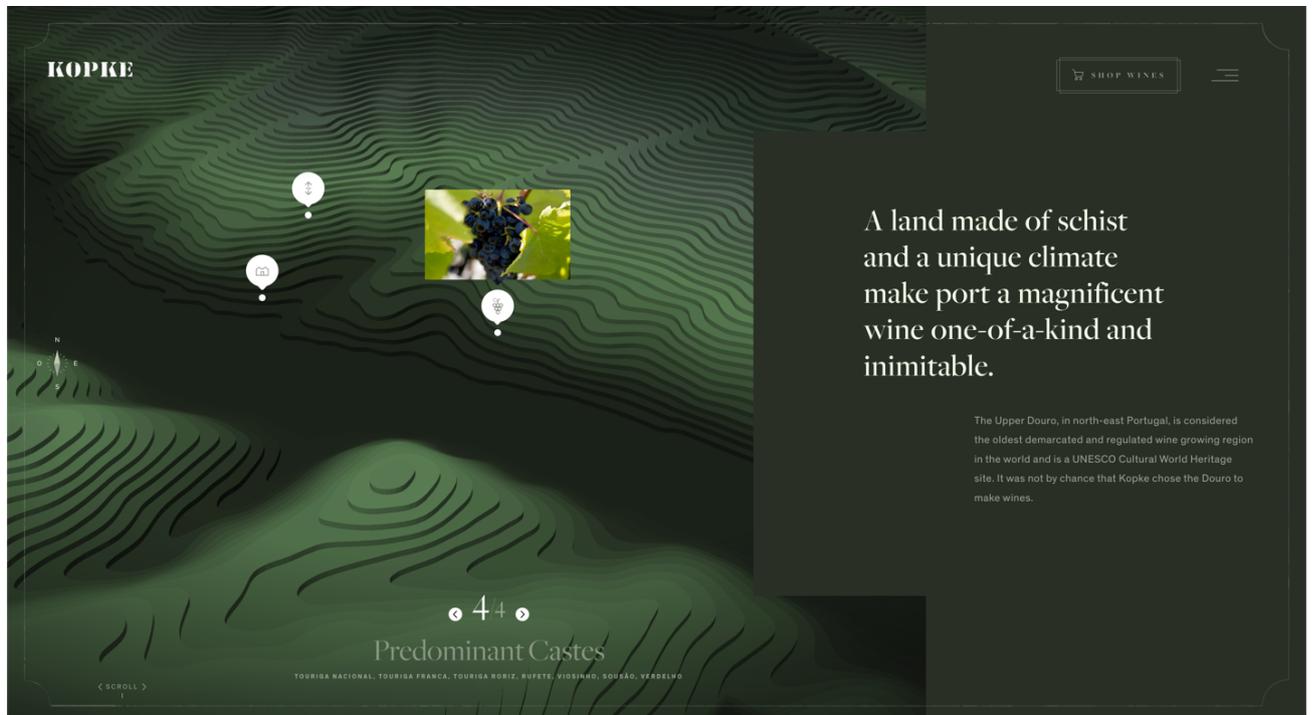


Figura 9. Mapa tridimensional en la web Kopke1638

En la sección “Quinta São Luiz” se utiliza una proyección 3D del texto que dice “Tourism” por medio de un contenedor *WebGL*. En la imagen de abajo se aprecia cómo las primeras letras están realizando la salida de la escena, por lo que van rotando y perdiendo opacidad. Las letras finales sin embargo, se mantienen en una posición de perfil, mostrando un efecto 2D como si fuera un texto plano normal, aunque es un objeto 3D. Al fin y al cabo, una ilusión llamativa sobre un elemento que el usuario ya conoce (texto plano).



Figura 10. Texto implementado en 3D en la web Kopke1638

4. USABILIDAD WEB



Se considera oportuno hacer una breve introducción a la usabilidad explicando por qué es tan importante este término para entender la experiencia de usuario y su creciente variante relacionada con las experiencias web inmersivas.

La **usabilidad** (dentro del campo del desarrollo web) es la disciplina que estudia la forma de diseñar espacios web para que los usuarios puedan interactuar con ellos de la forma más fácil, cómoda e intuitiva posible [14]. El término “usabilidad” se refiere por tanto al desarrollo de artefactos amigables para aquellos que las van a usar, consiguiendo que realicen sus objetivos de manera cómoda, con menor esfuerzo y mayor rapidez, o lo que es lo mismo, mayor eficiencia.

Históricamente, se empieza a hablar de facilidad de uso cuando Louis XV pide realizar un sillón confortable para sustituir su trono en Versalles. Con el paso del tiempo, son numerosas las industrias o inventores que han adoptado esta idea de facilidad de uso: MODEL A – Henry Ford (1927), SEARS TOPERATOR WASHING MACHINE - Henry Dreyfuss (1933) o DISNEYLAND – Walt Disney (1950) entre otros. En 1972 Harlan Crowder, trabajador de IBM, acuña el término “*user friendly*”, algo así como “amigable para el usuario” en castellano [15]. Es en la década de los 90 cuando surge el término “usabilidad” para sustituir al término “user friendly”. Con el paso del tiempo, la complejidad que engloba su definición ha sufrido devaluaciones de las que, a veces más a veces menos, ha conseguido recuperarse y adaptarse a nuevos tiempos. Originalmente, el término se orienta a tres puntos de vista principales [16]:

Vista orientada a producto

Donde la usabilidad tendrá el trabajo de analizar los atributos ergonómicos del producto. Se deberá tener en cuenta las barreras o comodidades en el proceso de creación o despliegue del producto, así como el impacto del mismo.

Vista orientada a usuario

Ha de evaluarse la capacidad de uso del producto para los diferentes usuarios. Es menester ver si el producto está adaptado a distintas capacidades de uso de los perfiles de sujetos que lo utilizarán, y si el uso del producto genera un esfuerzo mental notorio en los usuarios.

Vista de rendimiento del usuario

Sabiendo que el usuario es capaz de usar el producto, hay que analizar a continuación que es la manera óptima de utilizarlo. La aceptabilidad en el mundo real y la facilidad de uso son los principales temas a tratar en este punto de vista.



El gran obstáculo para estas pruebas es que están influenciadas por el contexto en el que se llevan a cabo. Es por esto por lo que la usabilidad y las técnicas que la evalúan evolucionan para tener en cuenta el contexto de ejecución, el contexto del usuario e incluso el contexto organizacional. Estas evoluciones han hecho refinar el concepto de usabilidad hasta el que tenemos en la actualidad, completando aquellas definiciones “tradicionales” con ideas o matices nuevos. En este artículo [17] de la *Revista Ibérica de Sistemas e Tecnologías de la Informação*, los autores de la publicación Gabriel Elías Chanchí G. , María Clara Gómez Álvarez y Wilmar Yesid Campo M. dan la definición de usabilidad « De acuerdo a Preece (1994) la usabilidad está asociada a aquellos sistemas fáciles de usar y fáciles de aprender, por lo que el concepto de usabilidad no es propio del área de los sistemas informáticos. Así mismo, según Nielsen (2012) la usabilidad es entendida como un atributo de calidad que mide la facilidad de la interacción entre el usuario y la interfaz. Del mismo modo, según la ISO 9241-11, la usabilidad es entendida como el grado en el que un producto puede ser utilizado por usuarios específicos para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un determinado contexto de uso. De estos tres atributos, la eficacia y la eficiencia son entendidos como atributos objetivos, mientras que el atributo satisfacción es el más subjetivo puesto que está asociado al confort o presencia de actitudes positivas durante la interacción (Enriquez & Casas, 2013; Hassan, 2015). En un test de usabilidad, la eficacia puede ser determinada de acuerdo con las tareas que un usuario puede cumplir en un tiempo determinado, la eficiencia se puede obtener a través del tiempo que emplea un usuario en realizar una determinada tarea y la satisfacción puede ser estimada mediante el uso de cuestionarios de percepción (Enriquez & Casas, 2013 ; Delgado, Girón, Chanchí, & Márceles, 2018). »

Se entiende por tanto que, dentro del proceso de desarrollo web, esta parte está asociada a la parte que comúnmente es conocida como *Capa de Presentación* por su término en castellano, o *Frontend* por su homólogo en inglés.



4.1. Jakob Nielsen: Principios de la usabilidad

Jakob Nielsen, Ph.D. en interacción persona-ordenador (abreviado en castellano *IPO* o *IHC*), es considerado como uno de los principales padres de la usabilidad y defensores del usuario. Director de Nielsen Norman Group, compañía que fundó con su socio y exvicepresidente de investigación de Apple el Dr. Donald A. Norman, es autor de más de 70 patentes estadounidenses relacionadas con la usabilidad web. Autor de numerosos libros, artículos y vídeos divulgativos sobre la usabilidad en la computación e internet, destaca el libro *Ingeniería de la Usabilidad* (1993) [18] donde lista cómo aplicar en el ciclo de vida de la ingeniería de la usabilidad diez (10) principios que sirven como guía para el desarrollo y análisis de interfaces siguiendo los estándares de la usabilidad. Estos principios son conocidos por el nombre de *heurísticas*, y dictaminan lo siguiente:

1. Visibilidad del estado del sistema

El sistema debe mantener siempre informados a los usuarios sobre lo que está ocurriendo, mediante una retroalimentación adecuada dentro de tiempo razonable. Los usuarios han de poder entender el estado del sistema, por lo que la retroalimentación ha de seguir una línea clara, intuitiva y cómoda de entender.

2. Correspondencia entre el sistema y el mundo real

El sistema debe hablar el idioma de los usuarios, con palabras, frases y conceptos que les resulten familiares, en lugar de que los términos orientados al sistema. Seguir las convenciones del mundo real, haciendo que la información aparezca en un orden natural y orden natural y lógico. En el caso de que se usen imágenes o iconos, estos han de representar aquellos elementos a los que se hace referencia.

3. Control y libertad del usuario

Los usuarios suelen elegir funciones del sistema por error y necesitarán una "salida de emergencia" claramente marcada para abandonar el estado no deseado sin tener que pasar por un extenso diálogo. Comúnmente soporta deshacer y rehacer.

4. Coherencia y normas

Los usuarios no deberían tener que preguntarse si diferentes palabras, situaciones o acciones significan lo mismo. Siga las convenciones de la plataforma las convenciones de la plataforma. Se recomienda seguir algún tipo de convenio, bien de *iure* o de *facto*, con el objetivo de encontrar denominativos únicos para cada uno de los elementos, consiguiendo diferenciarlos y por tanto que no compartan significado.



5. Prevención de errores

Aún mejor que unos buenos mensajes de error es un diseño cuidadoso que evite que se produzca un problema en primer lugar. Elimine las condiciones propensas a errores o compruébelas y presente a los usuarios una opción de confirmación antes de que realicen la acción.

6. Reconocer en lugar de recordar

Minimizar la carga de memoria del usuario haciendo visibles los objetos, las acciones y las opciones. El usuario no debería tener que recordar información de una parte del diálogo a otra. Las instrucciones de uso del sistema deben ser visibles o fácilmente recuperables siempre que sea apropiado.

7. Flexibilidad y eficacia de uso

Los aceleradores, que no son vistos por el usuario novato, a menudo pueden acelerar la interacción para el usuario experto, de modo que el sistema puede atender tanto a los usuarios inexpertos como a los experimentados. Permiten a los usuarios adaptar las acciones frecuentes. Pueden ser manifestados en forma de menús, atajos (*shortcuts*) o reiteración de enlaces en distintas secciones de la página.

8. Diseño estético y minimalista

Los diálogos no deben contener información irrelevante o raramente necesaria. Cada unidad extra de información en un diálogo compite con las unidades de información relevantes y disminuye su visibilidad relativa. Un mensaje sencillo y directo ayuda a que cale en los usuarios de mucha mejor manera y por tanto sea entendido con mayor facilidad.

9. Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de los errores

Los mensajes de error deben expresarse en un lenguaje sencillo (sin códigos), indicar con precisión el problema y sugerir constructivamente una solución. Por supuesto, han de referenciar el momento y punto exacto donde se ha producido el error y por qué se ha producido.

10. Ayuda y documentación

Aunque es mejor que el sistema pueda utilizarse sin documentación, puede ser necesario proporcionar ayuda y documentación. Esta información debe ser fácil de buscar, centrarse en la tarea del usuario, enumerar pasos concretos a realizar, y no ser demasiado extensa.

Para poner en práctica estas heurísticas, muchas empresas tienen departamentos encargados de usabilidad y diseño de la página web. Cada vez más profesionales del sector (diseñadores gráficos, desarrolladores web, ingenieros informáticos, arquitectos software...) siguen estos principios que, si bien pueden parecer más propios del ámbito de ventas y marketing, son esenciales para que la página que se accedida por los usuarios cumpla su función de mostrar el mensaje concisa y eficazmente. No obstante, el propio autor de este



“dogma” inculcado en la ingeniería de la usabilidad defiende con fiereza la necesidad de testear las interfaces con usuarios reales y/o expertos. Según Nielsen, se han de parametrizar estas cinco (5) principales características necesarias para que una interfaz sea usable [19]:

1. Facilidad de aprender

El usuario ha de ser capaz de interactuar con el sistema sin haber trabajado previamente con él. Una manera eficaz de comprobarlo es utilizando como métrica el tiempo que el usuario tarda en realizar ciertas acciones o tareas previamente establecidas por el equipo evaluador.

2. Facilidad de recordar

Los usuarios que hayan interactuado con el sistema deben saber cómo interactuar con él de nuevo. Una mayor facilidad para recordar la navegación provoca una mayor rapidez en el desarrollo de tareas y por tanto una mayor satisfacción para el sujeto. De nuevo, midiendo el tiempo que tarda el usuario (no nuevo) en realizar las acciones se puede obtener una métrica. Otra forma de evaluación es, mediante preguntas, conocer el grado de retención del usuario con respecto a pequeñas acciones o subtareas que puedan ser realizadas en la interfaz.

3. Eficiente

La productividad de la realización de tareas ha de ser un requisito vital a la hora de implementar una interfaz. Nuevamente, se puede medir el tiempo que tarda el usuario en hacer una tarea, diferenciando por supuestos aquellos que sean nóveles con los experimentados.

4. Sin errores (o pocos errores no-catastróficos)

Idealmente, el usuario no debería incurrir ni provocar ningún error durante el uso de la interfaz. No obstante, será casi imposible contemplar todos los posibles casos de fallo en la navegación de un usuario. Por este motivo es igual de importante que el sistema pueda recuperarse a un estado de no error para que el usuario siga con el proceso de navegación satisfactoriamente. Una medida muy útil para establecer el nivel de errores es, simplemente, contar los errores que se producen y saberlos diferenciar entre recuperables, irreversibles o catastróficos.

5. Satisfactoria para el sujeto

Subjetivamente, el usuario debe mostrarse satisfecho con el proceso y el resultado. Objetivamente se puede llegar a medir con encuestas donde se recoja el grado de satisfacción del usuario, incorporando comentarios de aspectos positivos y negativos, así como la consideración de posibles mejoras.



4.2. Métricas de usabilidad

En 1994 es publicado un artículo donde el propio Nielsen recomienda distintas formas de inspección en el ciclo de vida de la usabilidad web [20]. Estos métodos de inspección prometían resultados de usabilidad que mantenían costes bajos y resultados completos. La principal ventaja de estas metodologías era la dependencia de la revisión de las interfaces por parte de expertos en lugar de observar empíricamente el comportamiento de usuarios. Se propusieron varios enfoques que, en su mayor o menor medida, han ido evolucionando a la par que las interfaces de usuario.

A continuación se listan diferentes formas de tasar la usabilidad de un sistema. Se mencionan aquellas que se han adaptado de manera favorable con el paso del tiempo, teniendo en cuenta el abismo en la evolución de la usabilidad y de las plataformas web [21].

4.2.1. Evaluación heurística

Documentada por Nielsen y Molich por primera vez en 1990. De manera informal se puede analizar la usabilidad mediante el decálogo de heurísticas citado anteriormente [22]. Se recomienda contar con expertos en la materia de la interacción persona-ordenador y también de la interfaz a evaluar, aunque ninguno de los dos requisitos es imprescindible para obtener una evaluación válida. Han de saber interpretar todas aquellas señales que se produzcan entre el diálogo del usuario y el sistema, enmarcarlas en una de las diez heurísticas y por supuesto ofrecer soluciones a las mismas que sean fáciles de implementar y se mantengan en el tiempo. El objetivo principal de esta evaluación, que al final es el resultado que deseamos con la misma, es medir la calidad de la interfaz en función de su capacidad de ser aprendido, usado y recordado por un determinado grupo de usuarios en un determinado contexto de uso. Es preferible que haya varios evaluadores mejor que uno solo. Dentro de los evaluadores, se obtendrán distintos resultados si son novatos (usuarios no expertos en usabilidad), expertos, y por último expertos dobles (expertos en usabilidad y con conocimientos previos en la interfaz).

La principal ventaja de este método es la facilidad de llevarlo a cabo, el bajo coste y la rapidez con la que se realiza. El problema reside en que hay algunos problemas que no serán detectados, o incluso se detectarán falsas alarmas que harán perder tiempo al equipo que aplique los cambios en la siguiente etapa del ciclo de vida.

4.2.2. Recorrido Cognitivo

Método de inspección de usabilidad que evalúa el diseño de una interfaz de usuario por su facilidad de aprendizaje exploratorio. Suele orientarse para una tarea (o serie de tareas) en contraparte de la evaluación heurística. El método de evaluación de Recorrido Cognitivo basa su evaluación en la preferencia de muchos usuarios de aprender el sistema de una manera práctica por encima de un estudio de la documentación del mismo. El objetivo reside en identificar los usuarios que han de ser evaluados, dictaminar a continuación una serie de tareas a realizar y su método de resolución y finalmente dejar que los usuarios realicen las tareas de la forma que ellos consideran correcta. Haciendo la comparativa podremos obtener información donde



sabremos si el sujeto logró o no el objetivo, si no lo logró donde ocurrió el fallo, o cuales son aquellos elementos que pueden incentivar errores en la interacción del usuario con el sistema. [23]

Se ha demostrado [24] que el método puede ser aplicado por noveles, aunque necesitarán de desarrollar sus conocimientos siguiendo un proceso de curva de aprendizaje. En cualquiera de los casos que se aplique (noveles o expertos), ha demostrado ser un método muy eficaz, otorgando resultados complejos y con gran feedback del que partir para implementar las mejoras.

4.2.3. Recorrido de usabilidad pluralista

Adapta el método de recorrido cognitivo tradicional incorporando usuarios representativos, desarrolladores de productos, miembros del equipo del producto y expertos en usabilidad en el proceso. La ejecución de esta métrica pasaría por los siguientes pasos: se junta a un grupo de usuarios y se les presentan las distintas interfaces disponibles. Los usuarios han de escribir qué pantallas recorrerían, indicando las actuadores que generarían el proceso de navegación entre las mismas (botones, slides, scrolls...). A continuación, los usuarios debaten en una mesa redonda las distintas soluciones. De este intercambio de opiniones los evaluadores sacarán los resultados.

Las ventajas y desventajas de este método parten de la puesta en común de las ideas. Es obvio que juntando a usuarios con el equipo de desarrollo el feedback será obtenido mediante un canal directo, claro, eficaz y sin ruido de transmisión. Pero esto puede acarrear problemas de comprensión, coacción de opiniones o incluso disputas sobre si se tiene o no razón por parte del equipo de desarrollo hacia los usuarios.

4.2.4. Inspección formal de usabilidad

Las inspecciones formales de usabilidad toman la metodología de inspección del software y la adaptan a la evaluación de usabilidad. Las heurísticas son utilizadas como una ayuda para los no profesionales de la usabilidad en la búsqueda de defectos. Nuevamente, se indica a los usuarios que realicen una tarea siguiendo el camino que ellos consideren oportuno. Una vez terminado, los evaluadores recorren meticulosamente los pasos realizados para cada tarea, teniendo en cuenta los propósitos y objetivos de los usuarios (de forma similar a los recorridos cognitivos), si bien el énfasis radica menos en la teoría cognitiva y más en el hallazgo de errores. El objetivo es identificar el máximo número de defectos en la interfaz de la forma más eficiente posible. El proceso de revisión incluye modelos de desempeño de tareas y heurística aplicado al análisis del recorrido cognitivo del usuario.

5. ASPECTOS PRÁCTICOS DE LA UX INMERSIVA



5.1. Diseño de interfaces de usuario: UI/UX

Se denomina **interfaz de usuario (UI)** la interacción de un sistema y un usuario de manera bilateral a través de comandos o técnicas para operar dicho sistema, introducción de datos y utilización de los contenidos que ofrece el sistema. Las interfaces de usuario engloban desde sistemas como ordenadores, dispositivos móviles, juegos, o incluso aquellos dispositivos incorporados en IoT y otros aspectos de la vida como cajeros, pantallas de coches, etc. La **experiencia de usuario (UX)** se refiere a la experiencia relacionada con la percepción (emoción y pensamiento), reacción y comportamiento que el usuario siente y piensa a través del uso directo o indirecto de un sistema, producto, contenido o servicio [25]. La UX es un concepto relacionado con la interacción persona-ordenador que se aplica no sólo en el desarrollo de software y hardware, sino también en todo tipo de servicios, productos y procesos. El término **UI/UX** recoge la definición de una interfaz a través de la cual una persona puede interactuar con un sistema o aplicación en un entorno informático, pudiendo clasificar las interfaces entre interfaces software (aplicación, página web, menú) e interfaces hardware (enchufe, ranura, puertos de periféricos) [26].

Con respecto a una aplicación web, se puede interpretar como UI todos aquellos elementos que conforman la interfaz. Por dar ejemplos, se pueden enumerar listas, menús, inputs, etiquetas de texto o imágenes. La manera en la que estos elementos son colocados en la página web para que el usuario pueda acceder a ellos e interactuar sería por tanto la UX. Por tanto, el diseño de esa experiencia de usuario es el proceso en el que se crean productos útiles, fáciles de usar y que proporcionen satisfacción al usuario tanto en el proceso como en el resultado obtenido.

El proceso de creación de la experiencia de usuario de una interfaz se rige por una serie de pasos muy necesarios a la par que complejos. Son necesarios expertos en marketing e ingeniería social, expertos en diseño gráfico y expertos en desarrollo de interfaces. Las tareas de cada uno de ellos son vitales para la correcta puesta en marcha de la interfaz. De esas fases, a nosotros como desarrolladores web nos interesa la tercera de ellas, teniendo en cuenta el paso de la segunda fase a la última.

Desde la Fase de Diseño a la Fase de Desarrollo

El equipo de marketing ya ha decidido qué mensaje contar y como contarlo. El equipo de diseño ha materializado ese mensaje, siendo común la utilización de recursos visualmente potentes en el caso de proyectos inmersivos. Es trabajo del equipo de diseño que dejen todo documentado para que el desarrollo se lleve a cabo exitosamente: animaciones, interacciones con el usuario, tamaño de las imágenes y textos, etcétera. En este punto, el equipo de desarrollo ha de disponer del boceto final de la interfaz. Que sea el boceto final significa que debe haber sido testada, evaluada y refinada para que no haya cambios en fase de implementación. A partir de aquí, es tarea del equipo de desarrolladores saber cómo implementar los requisitos determinados por el resto del personal participante en el proceso. Se puede optar por utilizar frameworks, plugins o APIs, o bien desarrollándolo de manera propia para la plataforma.



5.1.1. Importancia de UX

Una buena experiencia de usuario hará que se ofrezca el camino más sencillo para que el usuario consiga acceder al producto que busca. Las ventajas que ofrece un buen diseño de la experiencia de usuario sobrepasan masivamente a las desventajas del proceso. Un mejor planteamiento de UX hará que los costes de refinamiento de la interfaz sean menores, por lo que el ciclo de vida será más sostenible y menos costoso. Además, la satisfacción tanto de los usuarios como de los desarrolladores (menos coste) se incrementará. Por el contrario, es conveniente destacar que el resultado obtenido de las métricas de experiencia de usuario puede no ser 100% representativo, si no que nos dará una aproximación en armonía a la dedicación que se haya puesto en las pruebas realizadas. En resumen, una buena UX se traduce positivamente en estos aspectos [27]:

- Aumento de la productividad
- Aumento de las ventas y los ingresos
- Aumento de la satisfacción del cliente
- Mejora en el posicionamiento (SEO)
- Reducción del coste de la asistencia y la formación
- Reducción del coste y tiempo de desarrollo
- Reducción del coste de mantenimiento

¿Cómo se materializa esto? ¿De verdad influye la experiencia de uso en aquellas personas que interactúan con nuestro sistema? La verdad es que medir algo no tangible puede ser un proceso ambiguo tanto para el evaluado como para el evaluador. Para demostrar la importancia de la experiencia de usuario en la sostenibilidad de la usabilidad de una plataforma web, se van a hacer uso de ciertas enumeraciones desglosadas en el libro *Laws of UX. Using Psychology to Design Better Products & Services* (Jon Yablosnky) [28].

Ley de Jakob

El usuario pasa la gran parte de su tiempo en otros espacios web, y prefiere que tu sitio web funcione de la misma manera que los otros sitios que conoce.

Psicológicamente, la mente del ser humano tiende a ser influenciada por la familiaridad de lo que le rodea. Nos sentimos cómodos con aquello que conocemos, aquello que dominamos. Y nos da respeto, miedo o desconfianza aquello que es nuevo para nosotros. La manera más efectiva y fácil de que un usuario consiga es sus metas, es que siga el recorrido que ya conoce. El resultado de esto será una doble satisfacción: El usuario no tendrá que aprender a hacer las cosas de cero. Ya sabe el camino, se siente confiado en lo que hace. Además, sabe cuál ha de ser el resultado obtenible. Tiene confianza en lo que ha hecho y en cómo lo ha hecho. En definitiva, en palabras del propio Jakob Nielsen: *Reconocer antes que recordar*.

Una de las principales ventajas es que, al aprovechar modelos mentales ya existentes (comportamiento, respuestas comunes...), podemos crear experiencias de usuario superiores que las de productos de nuestros competidores. Pero, si son necesarios aplicar algunos cambios, se ha de minimizar las diferencias, permitiendo a los usuarios usar una versión lo más familiar posible [29].



Supongamos ahora que el usuario ha de aprender a realizar una tarea nueva. Deberá memorizar una serie de pasos para llegar al objetivo, el cual deberá mostrar si se ha realizado el proceso satisfactoriamente o no. ¿Será fácil para el usuario realizar ese nuevo proceso?

Ley de Miller

Una persona promedio solo puede retener 7 (± 2) ítems en su memoria de trabajo.

El aprendizaje de uso de una plataforma se verá influenciado por la carga cognitiva que soporte la mente del usuario. Una sobrecarga de información con respecto a funcionalidades generará distracciones que dificultarán al usuario llegar a su objetivo. Dentro de la sabiduría popular de la sociedad es bien conocido que, muchas veces, *Menos es más*. Sabiendo cuándo y cómo desplegar una información concreta se conseguirá guiar al usuario hacia nuestro producto o servicio.

¿De dónde viene el número 7? En 1956, George Miller afirmó que el lapso de la memoria inmediata y el juicio absoluto estaban limitados a alrededor de 7 piezas de información. La principal unidad de información es el bit, que es la cantidad de datos necesarios para elegir entre dos alternativas igualmente probables. Asimismo, 4 bits de información es una decisión entre 16 alternativas binarias (4 decisiones binarias sucesivas). Este es el punto donde la confusión crea un juicio incorrecto de la capacidad de información del canal o, en otras palabras, la cantidad de bits que se pueden transmitir de forma fiable a través de un canal, en un tiempo determinado. Aunque este número sea muy importante, no hay que usarlo a la ligera para justificar limitaciones de diseño innecesarias. La correcta aplicación de esta ley reside en la agrupación de la información en pequeños montones para así procesarla, entenderla y memorizarla fácilmente [30].

No obstante, si la totalidad de la información mostrada en pantalla es necesaria y no puede ser modificada u ocultada, existen otras maneras de facilitar el trabajo al usuario y saberle guiar por nuestra plataforma.

Ley de Fitts

El tiempo necesario para lograr un objetivo es relativo a la distancia y al tamaño del usuario respecto a dicho objetivo.

¿Por qué sucede este comportamiento? En 1954, el psicólogo Paul Fitts, al examinar el sistema motor humano demostró que el tiempo necesario para moverse hacia un objetivo depende de la distancia hasta él, pero se relaciona inversamente con su tamaño. Según su ley, los movimientos rápidos y los objetivos pequeños dan como resultado mayores tasas de error, debido a la compensación entre velocidad y precisión. Aunque existen múltiples variantes de la ley de Fitts, todas abarcan esta idea. Por ejemplo, esta ley influyó en la convención de hacer que los botones interactivos sean grandes (especialmente en dispositivos móviles operados con los dedos): los botones más pequeños son más difíciles (y requieren más tiempo) para hacer click. Asimismo, la distancia entre la tarea/área de atención de un usuario y el botón relacionado con la tarea debe mantenerse lo más corta posible. [31]

Saber ordenar la información en pantalla es una ardua tarea del proceso de diseño de usabilidad. Son numerosas las técnicas a las que se pueden recurrir para llamar la atención de un usuario: Animaciones,



contraste de colores y formas, tipo de fuente, etcétera. Siendo concisos y sin menospreciar el valor que pueden otorgar todas estas modificaciones, existe una manera de hacer que algo destaque por encima de lo de más: hacerlo grande. Y si además de hacerlo grande, lo colocamos en un punto céntrico de la visión del usuario, llamará todavía más la atención. Escatimar en tamaño es muchas veces uno de los problemas más fáciles de solucionar pero que menos se tienen en cuenta por parte de métricas como evaluaciones heurísticas.

Ley de Hicks

El tiempo necesario para tomar una decisión aumenta con el número y complejidad de las elecciones posibles.

Además, reduciendo el número de posibilidades en pantalla, obtendremos una mayor limpieza visual. Se conseguirá más espacio, pudiendo jugar con los espacios en pantalla para resaltar los elementos que han de ser interactuados. Por supuesto, a más espacio, más posibilidad de jugar con el tamaño de los artefactos, y gracias al párrafo anterior ya sabemos de la importancia de resaltar esta característica.

Otra de las técnicas que se recomiendan para aplicar correctamente esta ley es dividir tareas complejas en pasos más pequeños para disminuir la carga cognitiva del usuario. También se recomienda ser meticulosos a la hora de resaltar las opciones recomendadas, intentando que no se sienta forzado a escoger estas opciones. Es muy importante no simplificar todo hasta el extremo, pues se puede llegar al punto de la abstracción donde el manejo sería igual o más difícil para el usuario. [32]



5.2. UX inmersiva: ¿Qué es?

La experiencia de usuario inmersiva sucede cuando una persona se ve llamada por un efecto o elemento que, de una manera visualmente atractiva y rompedora, muestra un mensaje que podría haber sido mostrado de manera tradicional. Dentro de la rama de la Experiencia de Usuario, la UX inmersiva promete ofrecer experiencias llamativas para el usuario. El objetivo principal es, que sin forzar al usuario a aprender un sistema complejo, sienta la necesidad de explorar una página novedosa en comparación a lo que conoce. La página web ha de mostrar el contenido de una manera no convencional para el usuario, pero sin alterar la facilidad de uso de la plataforma.

Realizar este tipo de experiencias es un trabajo de sastre: Ha de tenerse muy claro que es lo que se quiere mostrar, cómo se ha de mostrar, y por supuesto la implementación de esta idea teórica ha de estar meticulosamente llevada a cabo. La base principal de esta rama es suplir aquellas carencias que se pierden con la inmersividad fomentando la exploración del usuario a lo largo de la web. La disposición de elementos especiales puede hacer que el usuario no preste atención o no este acostumbrado al proceso de navegación al que se enfrenta, pero es menester saber guiar al usuario por medio de los artefactos inmersivos.

Las características por las que se puede reconocer una interfaz que ofrezca este tipo de experiencias se listan a continuación:

- La unicidad del sistema parte de la sensación de integración y coherencia entre los componentes causantes de la experiencia.
- El clímax de la atención del usuario se consigue y sumergiéndolo en la interfaz, de manera que dicha atención sea indivisible.
- Permitir al usuario un “Descubrimiento Activo” de los diferentes aspectos que ofrece la página produce una satisfacción desarrollada por el reto cognitivo de la exploración.
- Generar afecto en el usuario es la guinda para que esté incitado a volver a nuestra página. Crear recuerdos y momentos agradables a través de la usabilidad captará sus ganas de interactuar de nuevo con el sistema.
- La sensación de libertad a la vez que la gratificación intrínseca en la exploración de la web provoca un sentimiento de participación en la página. Ese sentimiento de portavocía de la experiencia y de reconocimiento de la satisfacción generará sensaciones intrínsecas agradables que el usuario puede recordar agradablemente.

El hecho de que sean interfaces más complejas no evade el deber de mantener una buena usabilidad en toda la página. Si bien no todos aquellos requisitos pueden ser cumplidos a rajatabla, se deberán suplir esas carencias fortaleciendo los puntos fuertes que caracterizan las interfaces inmersivas. La utilización de artefactos visualmente más llamativos que los métodos tradicionales facilitará esa suplencia de funciones. Volviendo a aquellas leyes de la UX enumeradas anteriormente, se han de interpretar aquellas recomendaciones para que encajen en la nueva definición que se quiere aportar.

Mencionando argumentos externos a los atributos estáticos de los elementos, se puede recurrir a la atención del usuario por medio de una herramienta muy potente: el movimiento. El dinamismo que ofrece la



monitorización y modificación de los controles utilizables favorece la adopción de los estándares mencionados anteriormente. Pero esta faceta por sí sola no conseguirá ofrecer el resultado final. Enmarcando esas alteraciones del movimiento junto con **técnicas** y **elementos inmersivos**, se definen aquellos recursos que compondrán el universo virtual que interactuará bilateralmente con el usuario. Todos ellos se documentarán en el apartado *5.3 Recursos inmersivos*.



5.3. Recursos inmersivos

Son múltiples las posibilidades que ofrecen las herramientas actuales para lograr algún efecto de inmersión web. El repertorio de soluciones [33] [34] recurre principalmente de la modificación de diferentes elementos web en la parte de software por parte de tecnologías dedicadas a la web, o de controles de componentes hardware como ratón, teclado o pantallas táctiles. Pero, otra de las técnicas que pueden simular un entorno interactivo es el uso de estilos o temáticas que acompañen y/o simulen los artefactos de interacción o inmersión. Elegir un estilo y tema idóneos subrayará aquellos elementos que han de destacar en la interfaz.

Se ha tratado de listar el mayor número de recursos posibles, teniendo en cuenta la escasa documentación que existe actualmente en lo que a esta rama de la usabilidad web se refiere. Se toman como referencia ejemplos prácticos, incluyendo todo tipo de comentarios con respecto al ámbito de su aplicabilidad digital siempre que su uso sea viable en la interfaz de una página web. Con intención de mencionar el origen de algunos de ellos, varios recursos provienen de la unión o adaptación de materias como puede ser el ámbito web con el mundo de los videojuegos, recogido bajo el término de la gamificación. En definitiva, saber innovar el campo de la usabilidad web con aquellos elementos que han triunfado en otras aplicaciones de las tecnologías.

5.3.1. Animaciones

En los albores de internet, las animaciones eran herramientas utilizadas por los desarrolladores por aquel tiempo noveles con el objetivo de llamar la atención de los usuarios. Sin embargo, nada más lejos de la realidad, este hecho no daba los resultado deseados. En gran medida provocado por el breve rendimiento de los navegadores y ordenadores (además de la inexperiencia de los desarrolladores por supuesto), se puede hasta afirmar que el resultado era el contrario al que se buscaba.

La evolución de la computación, liderada por las grandes hazañas de rendimiento y diseño logradas por iOS y Android ha mejorado radicalmente la experiencia de usuario. Gran muestra de esto es la simbiosis lograda entre la facilidad de uso de las herramientas que permiten generar las animaciones (para de los desarrolladores), y la facilidad de uso de las animaciones generadas (para de los usuarios). Desde este punto en adelante, el uso de animaciones se ha visto incentivado por la evolución constante de las tecnologías aplicadas al desarrollo web. Pero, ¿Es mejor desarrollarlas en JavaScript, o por el contrario implementarlas en CSS puro? La verdad es que, estas dos potentísimas herramientas sirven para realizar las animaciones de manera funcionalmente correcta. Pero, el uso de JavaScript permite explorar nuevas soluciones que con el simple uso de CSS no se contemplan [35].

Se podría afirmar que son “infinitas” las soluciones basadas en JavaScript. Para iniciar a los desarrolladores en esta materia de control de los elementos de la web, es común la formación en la potente herramienta JQuery, completada en ocasiones con otras librerías enfocadas a la moción como puede ser Velocity.js. Pero, para lograr la inmersividad que buscamos, se ha de dar un paso al siguiente escalón: los **Vectores**. Pasemos a comentar algo más acerca de este nuevo peldaño en nuestro camino.

Un vector de gráficos escalable, abreviado por sus siglas en inglés **SVG** es un elemento del DOM que, siguiendo una sintaxis XML directamente aplicable en HTML, genera una forma personalizada a través de figuras geométricas creadas por puntos [36]. La compresión de estas figuras al tratar simplemente los puntos



de las aristas y otro tipo de configuraciones menores como el color del borde o de relleno es mucho más eficaz que trabajar con imágenes tratadas pixel a pixel. A mayores, ofrece la posibilidad de reescalar el gráfico sin perder calidad en el proceso [36]. Por supuesto, trabajar con puntos permite que, al modificar su posición dentro del gráfico de una manera lógica y coordinada, se consiga simular un efecto de moción que no es más que la traslación de aquellos de una parte del vector. Sin duda, una potente solución para entornos dinámicos y responsivos. Una vez que conocemos la potencia de los SVG, podemos interactuar mediante JavaScript con otros elementos del DOM para crear el paisaje idóneo donde desplegar el efecto buscado.

Junto con la alteración de otros atributos como el color, donde podremos jugar con degradados, sombras y otro tipo de efectos ópticos, dispondremos de unas potentes herramientas para dar nuestros primeros pasos en la inmersividad. No obstante, también se puede recurrir a trabajos ya realizados por otros desarrolladores anteriormente. No debe darnos miedo importar plugins, APIs o librerías a nuestro proyecto. Un correcto uso de estas, previa documentación, dará grandes resultados en un muy breve periodo de tiempo. Sin duda, un gran apunte en la relación calidad-esfuerzo del proceso de implementación.

Vivus [37]

Si se desea imitar el efecto de escritura a mano en tiempo real este plugin es lo que buscas. Permite animar cualquier SVG para emular este efecto. Transformando texto al formato de vector se conseguirá un efecto llamativo ideal para mantener al usuario a la espera de que se complete el efecto.

SnapSVG [38]

Desarrollado por el autor de la herramienta *Raphäel*, ofrece las mismas posibilidades que este pero para navegadores modernos. La principal característica que hace destacar a Snap en el mercado es una potente API para desplegar SVGs desde JavaScript de una forma intuitiva y cómoda para el desarrollador. Además permite la animación y manipulación de vectores ya creados. Sin duda una herramienta a tener en cuenta cuando se trabaje con este tipo de recursos.

Aunque los vectores ofrecen una gran capacidad de trabajo, no son la única técnica de animaciones que podemos aplicar en nuestra plataforma. Aquí se listan otros ejemplos:

5.3.1.1. Scrolling

Una gran manera de aplicar animaciones en nuestra página es mostrándolas mientras el usuario va descubriendo el contenido de la misma. El proceso de descubrir la página literalmente se basa en la navegación dentro de la misma, comúnmente vertical, producida por el efecto de *scroll*. Este efecto es producido por el movimiento 2D de los elementos en la ventana. Gracias a JavaScript, se puede comprobar la posición desplazada respecto al origen de la página. Como se puede ver a continuación, son múltiples las aplicaciones que tiene el control de esta dinámica:



Framing-By-Framing

Se puede controlar el procesamiento de un vídeo fotograma a fotograma. Para ello, se deberá incorporar en un directorio del repositorio cada uno de los frames. Se recomienda darles un nombre formado por un prefijo seguido del número de frame (por ejemplo *Prueba_001*, siendo “Prueba” el prefijo y “001” el número de frame). Para lograr el efecto de Framing-By-Framing se deberá mantener el elemento en pantalla mientras la animación se está llevando a cabo. Para ello se puede hacer uso de la propiedad CSS *position: absolute* que fija la posición del elemento con respecto a los márgenes de la pantalla. Se puede ejecutar esta propiedad justo en el límite del scroll donde empieza la animación, y desactivarla del mismo modo en el punto que termina.

Una vez fijado el elemento, sabiendo cual va a ser el principio y el fin, se puede calcular la diferencia de distancia que recorrerá el usuario hasta llegar al último fotograma. Sabiendo esta diferencia, se puede asignar cada uno de los frames a un mismo número de elementos de distancia (px, vh, vw). Por ejemplo, si tenemos 35 frames y queremos que la animación ocupe desde el pixel 1300 hasta el 2900, tendremos 1600px de desplazamiento, por lo que se deberá cambiar de fotograma cada $1600/35 \approx 45,7\text{px}$.

Por supuesto, ha de testearse de manera que el “vídeo” animado quede fluido y no se noten los cortes entre un fotograma y el anterior o siguiente. Este proceso nos dictaminará si el resultado es válido o no. En caso de no serlo, puede solventarse bien añadiendo más fotogramas intermedios, o bien reduciendo la distancia que ha de “scrollear” el usuario.

Lottie Animation [39]

Creados por la empresa Airbnb, los archivos Lottie son una tendencia [40] en auge desde que llegaron al público. Su increíble rendimiento ha llevado a esta técnica a sustituir al ya archiconocido formato GIF, e incluso al formato vectorial en muchas ocasiones. Soportado por Android, iOS y React entre otros, esta librería desarrollada por Airbnb funciona de manera similar a un SVG, creando, trasladando y eliminando puntos, de manera que el resultado es liviano a la par que escalable. Sin duda una combinación de ventajas que muestra grandes resultados. La diferencia con un vector es que el vector se aplica en formato HTML, mientras que la carga de Lotties para nuestra página ha de realizarse mediante un fichero que almacene las coordenadas (comúnmente JSON, aunque puede optarse por otras soluciones como YAML).

Son ya varios los repositorios de *Lotties*: LottieFiles, IconScout, Motion Elements, etc. Aunque si lo que queremos es crear nuestro propio archivo, podemos crear por medio de herramientas como Affter Effects las animaciones, y posteriormente exportarlas al formato requerido para su implementación. Por supuesto, también existen herramientas para transformar SVGs en Lotties.



Horizontal Scrolling

Esta técnica hace que la navegación natural de la página (o de una sección) se realice de manera horizontal en vez de vertical. Este simple efecto dará un aroma de frescura e innovación a nuestro sitio. Es muy común utilizar este recurso para contar una historia, pues el dinamismo de los elementos y la frescura de su aparición emula una línea del tiempo virtual y sumerge al usuario en la experiencia. Esta técnica basada en el descubrimiento de la página a través de una historia se conoce como *storytelling*. Normalmente se usa este efecto visual apoyado sobre el movimiento de scroll denominándose ingenuamente *scrollytelling*, aunque no es exclusiva de esta forma de interactuar.

5.3.1.2. Efecto Parallax

Con el *efecto parallax* conseguiremos un efecto de desfase entre el fondo y las diferentes secciones. Esto se consigue aplicando a los elementos una velocidad de desplazamiento distinta con respecto a la que debería tomar. El punto fuerte de esta técnica se ve reflejado en el efecto de profundidad y movimiento a la hora de navegar por la página. Este principio es muy útil para romper los esquemas de los consumidores. Se puede jugar con varias capas de profundidad para conseguir efectos más llamativos o realistas.

El efecto de paralaje se logra controlando la posición de los elementos con respecto a un evento, como bien puede ser a medida que se hace scroll, de la misma manera que en el ejemplo anterior solo que en vez de modificar el elemento principal del cuerpo (comúnmente con la etiqueta HTML `<main>`), habría que modificar en función del tramo desplazado las coordenadas donde se ubican los elementos afectados. Otro tipo de controlador que podemos usar para aplicar la moción a los elementos es el movimiento del ratón (sección *Mouse following* dentro del apartado **5.3.4.1. Interacciones de Ratón**). Aun pudiendo hacerlo a mano, se da la opción de importar librerías ya creadas especializadas en esta materia. Son varias las que permiten lograr este efecto, aunque la que mayor comunidad y documentación tiene es la librería *Parallax.js* [41]. Es recurrente el uso de parallax para lograr el *scrollytelling* anteriormente mencionado, pudiendo ser en la dimensión vertical (comportamiento normal) o en la horizontal, como veíamos en el caso anterior.

5.3.1.3. Cardboarding: Device Showcasing

En los últimos años se ha creado la tendencia de desplegar imágenes de la plataforma funcional en la propia página de la plataforma. Sí, es algo extraño, intento explicarlo: Es común incorporar capturas del funcionamiento de la aplicación, mostrando así confianza en el producto para el posible consumidor. Pero este tipo de capturas no se muestran tal cuál en la página sino que se suelen enmarcar en dispositivos hardware. De esta manera, se apela a lo que el usuario ya conoce: un dispositivo (ordenador, móvil, tablet o cualquier otro) con la aplicación funcionando.

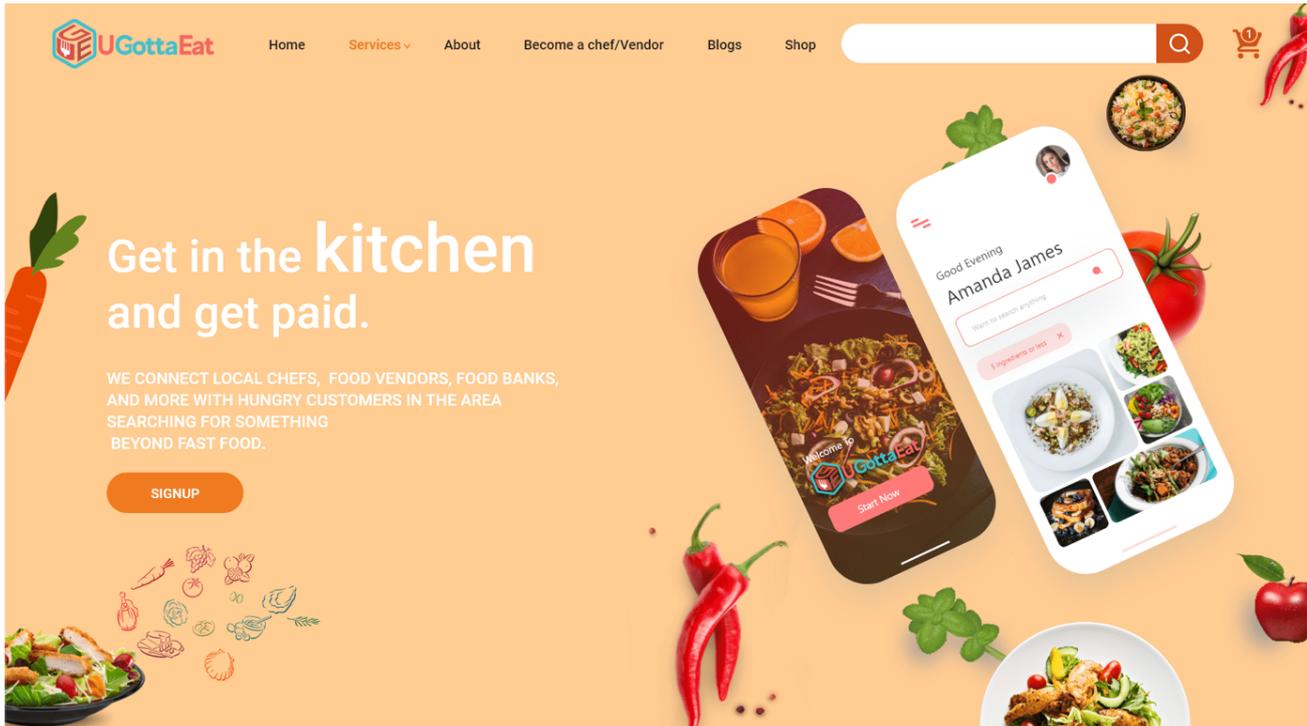


Figura 11. Showcasing de la aplicación en la parte derecha de la imagen

Pero hay ocasiones en las que, refinando esta muestra del producto, se consiguen efectos totalmente sorprendentes. Jugando con animaciones, se puede dar vida a la aplicación dentro del dispositivo. Del mismo modo, haciendo que el objeto sea tridimensional, se puede controlar su movimiento con diferentes eventos como movimiento del ratón, scrolls o clicks. Sin duda, una forma robusta de mostrar la viabilidad de nuestra plataforma y nuestro producto.

5.3.2. Elementos 3D

A medida que los gráficos tridimensionales basados en la denominada *Web 3D*, relacionados en muchas ocasiones con la famosa *Realidad Virtual* y otro tipo de técnicas de innovación, continúan siendo más asequibles, son más los entornos de desarrollo que han ido adoptando esta tecnología para su implementación en plataformas digitales [33]. Es una afirmación que el uso de elementos 3D en una página web da un resultado exótico al resultado de la misma [42]. Son muchas las maneras de aplicar este tipo de recursos para crear artefactos personalizados para nuestro entorno, aunque todos los caminos derivan en el mismo camino. La evolución de los navegadores ha permitido que la evaluación del entorno y su posterior modificación por medio del ya mencionado JavaScript sea una tarea cada vez más trivial para los desarrolladores.

Para poder hablar de la Web 3D, se debe hacer mención del término **WebGL**. Originalmente basada en OpenGL, la idea era crear objetos en un entorno tridimensional renderizado en tiempo real en el entorno de despliegue del cliente (navegador del usuario). WebGL es por tanto una librería de gráficos tridimensional que permite a los navegadores renderizar esos objetos de una manera eficiente y estándar para todos aquellos



navegadores o plataformas que requieran de esta interfaz de despliegue. Para crear estos objetos, ha de procesarse el entorno en base a una serie de componentes esenciales: [43]

- **Canvas:** Es el lienzo donde se renderizará la escena. Este espacio un elemento HTML5 que es accesible por JavaScript a través del DOM.
- **Objetos:** Entidades 3D compuestas por polígonos trilaterales que se muestran en la escena. Como poliedros que son, se han de tener en cuenta algoritmos de carga, omisión de renderización de partes no visualizadas y texturización del cuerpo.
- **Luces:** Efectos de iluminación y ensombrecimiento de los poliedros mostrados en el lienzo.
- **Cámara:** Ángulo desde el que se verán los objetos proyectados en el lienzo.

Actualmente son muchas las maneras de crear e incorporar objetos tridimensionales a nuestras páginas web [44]. Se listan a continuación varias de estas posibilidades.

5.3.2.1. Three.js [45]

La evolución del uso WebGL ha generado en el mercado una multitud de herramientas orientadas a la programación de objetos 3D. GLGE [46], SceneJS [47] o CubicVR [48] son algunas de las librerías opensource que se pueden encontrar en internet. Aunque estas librerías ofrecen una implementación a alto nivel y son amigables a la hora de utilizarlas, no son la elección favorita del grueso actual del público.

La librería *Three.js* (o *ThreeJS*) es una potente librería con gran cantidad de documentación y comunidad en la que apoyarse a la hora de trabajar con ella. Haciendo uso de Ray Casting, facilita el trabajo de WebGL con el DOM de manera que se trabaja con la proyección 2D generada por el objeto 3D. De esta manera, se hace menos carga en la GPU y se consigue mayor rendimiento [49].

Otra de sus grandes facetas es la posibilidad de incorporación de modelos tridimensionales de otras herramientas como *Blender*, *ZBrush* o *Maya*. De esta manera, se puede dividir el trabajo relacionado con el diseño de los objetos y escenarios con el de la programación de los mismos, gran utilidad para la división de tareas en el entorno de trabajo. Simplemente será necesario el esqueleto y las texturas del objeto para incorporarlo a nuestro lienzo, y finalmente podremos interactuar con el objeto en la web.

5.3.2.2. Spline [50]

Basada en la tecnología ThreeJS, la plataforma online *Spline* permite diseñar y exportar figuras 3D animadas totalmente funcionales para la incorporación en entornos web de una manera muy simple y fácil de realizar. Esta herramienta ofrecida como SaaS a sus usuarios no tiene la potencia que pueden tener gigantes como Blender o la suite de Adobe, pero ofrece una solución rápida y sencilla con la que exportar los artefactos 3D a código rápidamente.

5.3.2.3. 3D Mapping

Una corriente muy utilizada para lograr la inmersión de los usuarios es la representación del mundo real en el entorno digital. Las oportunidades para captar la atención que ofrece la técnica de mapeado tridimensional son abismales. Apelar al descubrimiento de su entorno a la vez que a la exploración de lo ya conocido es una virtud que pocos recursos pueden ofrecer. Son múltiples las herramientas que permiten



crear mapas 3D. Incluso se podría hacer un objeto 3D de manera tradicional, aunque no es lo recomendado, existiendo herramientas como las a continuación mencionadas

Mapbox [51]

La herramienta **Mapbox** permite crear una plataforma de mapas con su propio sistema de navegación, añadiendo la característica de la extrusión del terreno y los edificios tridimensionalmente. Junto con características como marcación de puntos de interés, ambiente y climatología y animaciones de *travelling* (ángulo, posición y rotación) se convierte en una herramienta muy potente. Además con la posibilidad de personalización de las texturas, de la libertad de navegación que se le deja al usuario, y otro tipo de medidas como demográficas o hidrográficas entre otras, sitúan a la herramienta en una posición muy aventajada en el ámbito del desarrollo de mapas 3D.

ArcGIS [52]

Calificado como un sistema de información geográfica para trabajar con mapas, se puede considerar más bien como un sistema de análisis y visualización de información mapas. Las ventajas de este recurso sobre Mapbox residen en la potencia y complejidad de los datos que se pueden manejar. Sin embargo, visualmente puede no ser tan llamativo como Mapbox. Todo dependerá de si en nuestra página necesitamos que sea más visual, o que los datos sean más complejos.

Google Maps Platform [53]

La popularidad de la herramienta Google Maps ha favorecido que el gigante estadounidense publique una API con la que acceder a métodos de personalización de los mapas que gestiona. Son claras las ventajas que ofrece esta alternativa por encima de Mapbox y ArcGIS. Lo principal a destacar es todo el sistema de información compartida del que dispone Google gracias a la inmensidad de su dominio, ya no solo en la enumeración de locales, si no de reseñas, de población y aforo actual tanto del local como del recorrido (tráfico), control de inconvenientes como incendios, accidentes u obras, etcétera. En definitiva, una amplia red de comunicación que puede ser un gran apoyo para nuestra plataforma en caso de necesitar optar a toda esa información.

5.3.2.3. Realidad virtual: VR/AR

Siempre que se habla de inmersividad una de las primeras facetas que nos viene a la mente es la realidad virtual y la realidad aumentada. La integración de este tipo de tecnologías en cualquiera de los navegadores ha supuesto que la web tradicional evolucionara hacia el siguiente escalón del mundo de la “interacción a domicilio”. Lo único que necesitamos para que esto funcione es que nuestro dispositivo disponga de una cámara o webcam incorporada o conectada.

La diferencia entre estos dos conceptos reside en el escenario donde se despliegan: En la **realidad virtual (VR)** se crea un nuevo entorno, un nuevo mundo totalmente virtual e inmersivo. La ventaja de esta vertiente



es la abstracción de la realidad del usuario ofreciéndole que se sumerja en el universo digital. La inmersión se completa con la posibilidad de interacción con los distintos elementos desplegados en el mundo virtual. Por el contrario, la **realidad aumentada (AR)** ofrece desplegar artefactos interactivos en el entorno del usuario. Se puede denominar como la conjunción entre el mundo real y el mundo virtual. [54]. Existe una tercera vertiente denominada como **realidad mixta (MR)** donde se mezclan los dos aspectos, haciendo posible la interacción con los elementos virtuales desplegados en un entorno real, tomando las ventajas más importantes de cada una de las vertientes [55].

El auge de esta tecnología ha sido gracias al avance de los estándares **WebGR** orientados a la detección de superficies y renderizado de las escenas 3D. Aplicando esta tecnología con modelos 3D propios de WebGL, podremos desplegar los artefactos tridimensionales en el entorno. Sobra decir lo importante que es que un usuario pueda interactuar con su entorno por medio de nuestro producto. Tanto a nivel cognitivo como a nivel psicológico, la satisfacción de conocer aquello que te rodea en este mismo momento y de enseñárselo a los demás, es uno de los pilares de la rama de la experiencia de usuario inmersiva.

Son múltiples los casos de éxito de este tipo de tecnologías, si bien son dos (2) los claros titanes a mencionar [56]. El primero de los casos es el mundialmente conocido **Pokémon Go**. El famoso juego desarrollado por *Niantic* ofrece la posibilidad de explorar nuestro entorno, nuestra ciudad, nuestras calles, y a partir de esta exploración disfrutar de lo que “Mundo Pokémon” nos puede ofrecer. En concreto, se utilizan otros dos recursos además de la realidad aumentada. Es el mapa del juego quien nos dice donde están las criaturas a capturar, mapa con el que podremos interactuar. Por supuesto, las criaturas han de mostrarse en pantalla de alguna manera. Este despliegue se hace por medio de objetos tridimensionales animados. Además de las criaturas, tenemos otros objetos 3D como pueden ser las “pokeballs” con las que capturaremos a los monstruos, o las “pokeparadas”, de las que podremos obtener las pokeballs.

El segundo de los casos a mencionar ofrece una forma lúdica de descubrir la fauna de nuestro planeta. **Google animal 3D** permite desplegar gran variedad de animales en el entorno donde estemos. En esta lista se incluyen también animales extintos como dinosaurios, o incluso en colaboración con la NASA se pueden consultar también los planetas del sistema solar.

La propia Google ofrece herramientas para implementar AR y VR a nuestros proyectos, pero no es la única solución en el mercado.

5.3.2.4. Fotorrealismo

Entre todas las aplicaciones de los modelos 3D, uno de los usos más curiosos reside en el escaneo y posterior proyección de objetos reales en formato digital. Gracias a herramientas de escaneo y/o realización de modelos fotorrealistas, se abren nuevas ramas de usabilidad de los objetos tridimensionales en la web. Gracias al framework *ATON* [57], cualquier usuario es capaz de recoger un elemento de su entorno en formato tridimensional de una manera totalmente gratuita.

La aplicabilidad de esta tecnología reside principalmente en la recogida y muestra de datos, tanto de carácter documentativo como de carácter lúdico. Museos, exposiciones o incluso trabajos de investigación pueden verse apoyados por esta aplicación de la Web3D.



5.3.3. Textos

Si lo que necesitamos es ofrecer un mensaje textual llamativo, son muchas las opciones que se pueden encontrar en la web para lograr efectos inmersivos. En este caso, no se puede hablar tanto de interacción del usuario con este tipo de elementos ya que por lo general un usuario no interactuará con un artefacto como este a no ser que se le añada algún comportamiento especial por JavaScript, como puede ser un hipervínculo u otro tipo de *trigger*.

Blotter.js [58]

La propia herramienta se define como una *API para dibujar efectos no convencionales para textos en la web*. Apoyado en el sistema de sombreado de GL conocido por el nombre de *GLSL*, esta herramienta ofrece efectos tales como distorsiones, barridos, granulado o división de colores en canales. Sin duda una herramienta muy a tener en cuenta para darle una vuelta de tuerca a los textos de nuestra página.

Typed.js [59]

Typed es una librería que simula la escritura del texto tal y cómo se realiza en el ordenador: carácter a carácter. Ideal para tutoriales, *showcase* de código o para menciones y citas, es una herramienta muy liviana a la par que eficiente.

Otra alternativa para la realización de estas animaciones orientadas a textos es tratar esta tarea como si de cualquier otro objeto se tratara. De esta manera, se pueden aplicar las distintas técnicas de animación orientadas a SVGs, Framing-by-Framing, Lotties, o incluso animaciones 3D con las que se puede dar vida a los cuerpos, en este caso con forma de letras u otro tipo de caracteres.

5.3.4. Controles periféricos

Una de las formas más eficaces de hacer sentir al usuario que está manejando la web, que está inmerso en esta, es controlar los movimientos que él mismo realiza. Para ello es necesario monitorizar los *inputs* que realiza el usuario mientras navega. Podemos clasificar (con un pequeño matiz) estos inputs en dos principales grupos: inputs de ratón e inputs de teclado, aunque hay más grupos que serán tratados en este apartado. El matiz reside en la experiencia de usuario aplicada a dispositivos táctiles, donde la pantalla hará la función de los dos. No obstante, para mantener la organización de esta clasificación, se mantendrán separados ambos grupos, y se pide que sea el propio lector quien tenga en cuenta las funciones que una pantalla táctil suple cuando no existen otro tipo de entradas.

5.3.4.1. Interacciones de Ratón

Se ha de tener en cuenta que el ratón cuenta con cuatro (4) entradas principales, de las cuales tendremos en cuenta dos (2) ya que una es externa al comportamiento de nuestra página y la última ya se ha mencionado previamente. Las entradas que se pueden recoger del ratón son el Click Izquierdo, el Click Derecho, la Rueda



del Ratón y el propio movimiento. La tercera de ellas ya ha sido mencionada en el apartado **5.3.1.1 Scrolling**, por lo que no será mencionada aquí. La segunda entrada despliega el menú de opciones correspondiente a la aplicación, interacción que queda fuera de nuestro entorno.

Por tanto, son dos las interacciones que podemos recoger del usuario para generar efectos que generen interés en él. A continuación se muestran algunas aplicaciones de estos recursos.

Click Counter

Esta interacción consiste en el número de clicks que se ha hecho en un elemento, en una zona de la pantalla o incluso en la página entera. Son múltiples los usos que se me ocurren de esta técnica: desde un saco de boxeo que mida la “intensidad del golpe”, una carrera donde el personaje se desplace con cada click, o incluso simular el zarandeo de un árbol para conseguir que caigan objetos de él. Como vemos, el uso de esta técnica se puede acompañar de animaciones para lograr que el usuario se sumerja todavía más en la tarea.

Drag & Drop

Juntando el evento de clicar con la detección de la posición del ratón se pueden conseguir interacciones tan llamativas como esta. *Drag & Drop*, o en español “arrastrar y soltar” es un movimiento que la gran mayoría de usuarios ya tiene mentalizado, pero al que apodemos aplicar un nuevo significado. Normalmente los usuarios están acostumbrados a realizar este movimiento con tareas relacionadas con archivos: subir archivos a la nube, moverlos de carpeta, etcétera. En esta ocasión, se puede apelar a un entorno más lúdico donde se necesite que el usuario arrastre un componente desde un punto a otro para seguir con la navegación.

Mouse Following

La posibilidad de monitorizar la posición del cursor en todo momento es una gran virtud que ofrece JavaScript para captar la atención del sujeto. Las posibilidades de aplicación de esta materia son muchas. Se pueden crear objetos que sigan al cursor, desde simples círculos de un color opaco hasta efectos de ondas que deformen la página, cuadros de texto muestren un mensaje en forma de texto [60] o incluso formas que cambien los colores de aquello que tocan [61] Nuevamente en este ejemplo de *Pluto App* se aprecia el seguimiento del movimiento por parte de los objetos tridimensionales siempre y cuando se acceda a la sección.

Como bien se ha visto, juntando esta técnica con elementos 3D o animaciones se puede conseguir que se desplieguen eventos en función de aquellos movimientos específicos que realice el usuario. Todo movimiento que nosotros queramos que sea un pistoletazo de salida para que ocurra algo, podremos controlarlo gracias a este recurso.



5.3.4.2. Interacciones de Teclado

Del mismo modo que se pueden recoger las entradas del ratón, se pueden manejar también las entradas de teclado. No hay mucho más que explicar: gracias a JavaScript podemos controlar qué teclas se están tocando y desplegar eventos en función de estas.

Las utilidades de esta táctica de interacción se pueden ver reflejados por ejemplo en un entorno más cerca de la gamificación que de otra cosa. Supongamos una compañía de coches o aviones que tiene el modelo 3D como se muestra en el segundo apartado de esta entrada relacionada con las tendencias de UI/UX en *Codica* [33]. Bien, pues aplicando una vista en tercera persona a estos dispositivos se puede simular la conducción de los vehículos como si de un videojuego se tratara con las interacciones de diferentes teclas.

Uno de los mayores ejemplos de control de teclas, quizás por su simplicidad y potencia, es el famoso *Dinosaur Game* de Google Chrome. En él se muestra un genial uso de la interacción por teclado permitiendo que el dinosaurio realice mientras avanza sus dos movimientos característicos: saltar y agacharse.

5.3.4.3. Interfaces de voz

Las interfaces de voz de usuario, abreviado **VUI** en inglés, permiten a los usuarios relacionarse por medio de un canal sonoro con los dispositivos móviles. Por medio de comandos de voz se pueden dar órdenes que, con un correcto procesamiento, pueden ser interpretados por la plataforma para realizar distintas acciones acordes a las peticiones. Uno de los principales proveedores de este procesamiento es VoiceXML, un procesador ampliamente utilizado y con gran potencia de interpretación [62].

Asistente virtual

Una de las aplicaciones de las VUI es la incorporación de un asistente virtual a nuestra plataforma. Un *asistente virtual* es un software interactivo controlado por comandos de voz que tiene el objetivo de ayudar al usuario en diversas tareas. La potencia de estos asistentes es la capacidad de personalizarlos gracias a pequeñas configuraciones de tareas denominadas como *skills*. El desarrollo de *skills* personalizadas basadas en tecnología web es un campo que está siendo investigado e incentivado por parte de compañías y particulares. [63]

5.3.5. Microinteracciones

Las microinteracciones son detalles de diseño que no afectan a la arquitectura ni al contenido de la aplicación. Son pequeños adornos que se incorporan a la interfaz para deleite del usuario mientras interactúa con ella. Hay que tener cuidado a la hora de aplicarlos, pues recargar una interfaz con una cantidad basta de microdetalles puede crear ruido visual que haga que la aplicación pierda usabilidad.

Para diseñar una microinteracción se ha de identificar y comprender el comportamiento del usuario y los problemas o dudas que puede tener al utilizar nuestra página web. Es muy importante mantener las microinteracciones naturales, haciendo que destaquen pero que pierdan el centro de atención en un periodo óptimo de tiempo para que el usuario pueda continuar con el proceso de interacción. [64].



Son ejemplos de microinteracciones eventos que suceden al hacer hover o click en artefactos de la web, como cambios de color, pequeñas animaciones (botón de like, botón de compra, botón de enviar email...). En definitiva, pequeñas muestras de que lo que ha realizado el usuario ha salido satisfactoriamente.

Uno de los mejores casos para mostrar qué es una microinteracción se puede encontrar en el botón de “me gusta” que ofrece la plataforma Facebook para interactuar con las publicaciones que encontramos en nuestro muro. El despliegue de un submenú al hacer hover durante un breve periodo de tiempo, muestra al usuario seis (6) emoticonos disponibles para mostrar diferentes formas de satisfacción del usuario con la publicación. Estos emoticonos están animados y desplegados en un canvas que se genera dinámicamente por medio de JavaScript. A la hora de seleccionar uno, se mostrará otra microinteracción y, una vez indicado cuál es el emoticono seleccionado, se desplegará otra animación, esta vez en el sitio donde hicimos hover anteriormente.

Sin duda son miles las aplicaciones de este recurso: despliegue de menús, animación de iconos, de botones, de listas... pero hay que saber aplicarlas con cautela.

5.3.6. Temas

Una buena manera de subrayar todos aquellos elementos inmersivos es acompañarlos de un buen marco donde mostrarse. Captar la atención del usuario no solo se basa en hacer algo llamativo, sino también en evitar que el resto de los elementos no lo sean. Existen múltiples temáticas en tendencia en los últimos años, y es igual de importante el uso de las mismas que de los elementos que se mostrarán en pantalla. Del mismo modo que es importante para la empresa usar un diseño centrado en el usuario, así lo es también para el propio consumidor de la página. Mostrar un estilo confiable, yendo al grano en lo que el usuario quiere, y por supuesto de una manera cómoda es una manera de captar usuarios y dar una buena imagen tanto de nuestro producto como de la manera en la que se está vendiendo a los adquirentes.

Se quieren mencionar varios ejemplos de estilos que facilitan la inmersión de los usuarios, haciendo hincapié en cómo lo logran y cuál es el efecto conseguido.

Esqueumorfismo

Esta técnica de diseño incluye en los elementos de la página estructuras y ornamentos que tradicionalmente estaban asociados a este tipo de objetos [65]. El objetivo es llevar todo aquello del mundo real al mundo digital, aunque no sea funcional. Por ejemplo, en un icono de una cámara, mostrarla como si fuera una polaroid. Del mismo modo, otro ejemplo sería añadir relieve a los botones de Off u On como si de pulsadores físicos se tratara.

La utilización de este tipo de interfaces es óptima en páginas web donde se quiera apelar a los recuerdos del usuario. Páginas enfocadas a personas con una cierta edad, que hayan crecido con la tecnología analógica, se verán más afectadas por elementos esqueumórficos que usuarios más jóvenes.



Figura 12. Formulario con estilo esquemórfico [66]



Figura 13. Botones con estilo esquemórfico [67]

Neumorfismo

El neumorfismo es un estilo que, más allá de enfocarse en el contraste entre el mundo real y el mundo digital, se centra en la manera de mostrar los elementos siguiendo una paleta de colores pastel sobre un fondo unicromático. La línea de estilo que sigue el neumorfismo es suave, donde se difuminan bordes como si de una textura de goma se tratara. Para conseguir esto se juega con bajo contraste y un correcto juego de sombreados [68].

Es una controversia si este estilo deriva o no del esquemorfismo [69]. La verdad es que, sin duda, mantiene elementos no necesarios en las interfaces, pero es muy discutible que los mantenga por parecerse a elementos antiguos como el anterior estilo ya mencionado.

El uso de esta tendencia de interfaces puede ser problemático para personas con capacidades visuales reducidas, por lo que se recomienda encarecidamente que se valore cuando usarlo.

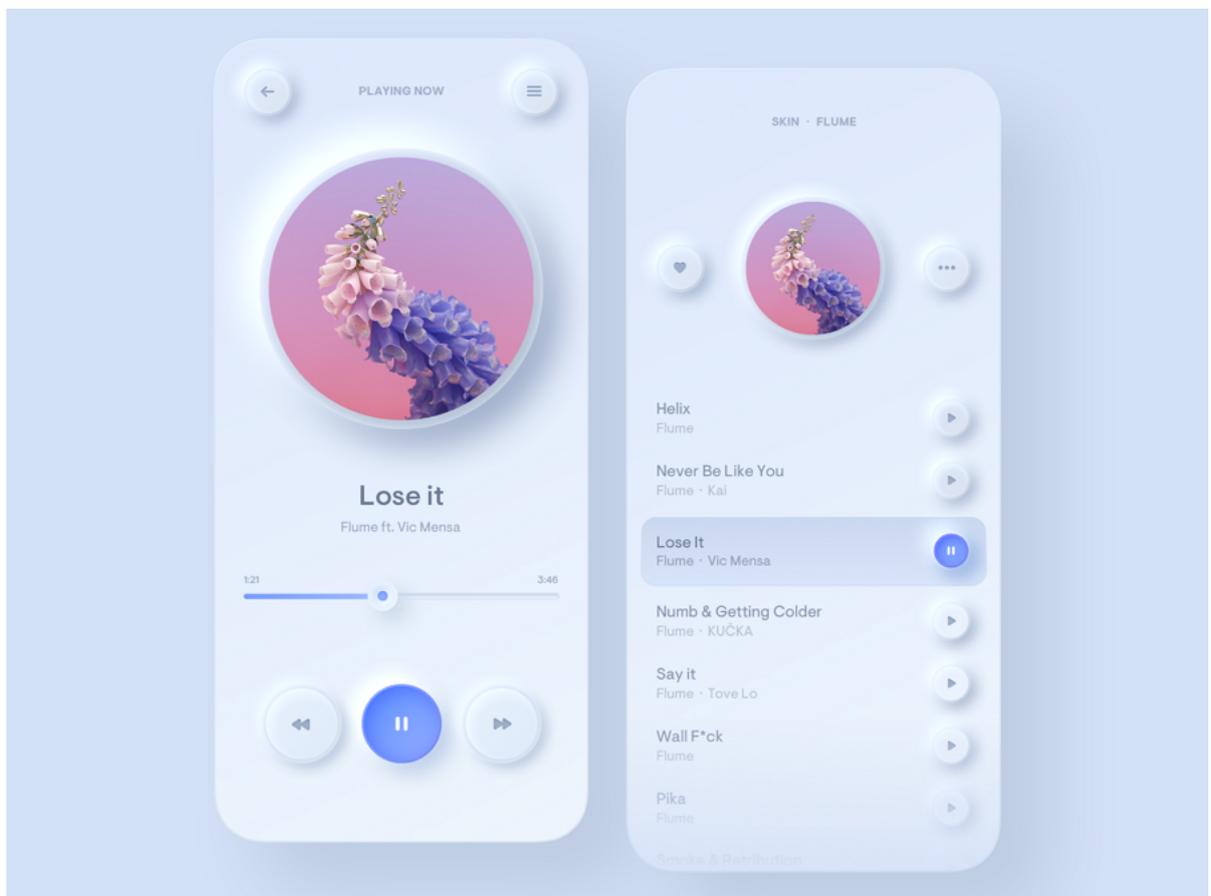


Figura 14. Aplicación móvil con estilo neumórfico [70]



Figura 15. Botones con estilo neumórfico [71]

Minimalismo

Sirva como descripción de este estilo la expresión “Menos es más”. El objetivo ya no es ofrecer un producto con muchas funciones, si no el producto más eficiente en su función específica. Basándose en la ya mencionada *ley de hicks*, este estilo reduce al máximo las opciones que un usuario puede tomar, haciendo que el camino hasta lograr su objetivo sea el más sencillo posible. Demasiadas opciones pueden hacer que un usuario se sienta abrumado y, en última instancia, abandone la tarea que está realizando.

El objetivo del minimalismo por tanto es ofrecer solo el contenido esencial apoyándose en el contraste entre elementos, espacios en blancos, diseño plano y muchas indicaciones en base a fotografías e ilustraciones gráficas tales como figuras y patrones. Los mensajes han de ser claros y concisos, utilizándose colores uniformes a la par que contrastados con el resto de elementos. Por supuesto, el uso de tipografías diferentes, diferenciado el estilo normal de la aplicación en negrita o cursiva ha de ser notorio en la interfaz, haciendo que tengan un significado propio en el mensaje que se muestra al usuario. [72] [73]



Figura 16. Página web con estilo minimalista [74]

Glassmorfismo

El efecto visual del glassmorfismo reside en la imitación de cristales traslúcidos en los diseños de la interfaz. Se potencia el efecto de los cristales gracias al uso de colores llamativos y bordes redondeados, comúnmente animados. De esta forma se consigue llamar al usuario a fijarse en ese elemento, por lo que si colocamos algo encima será el anzuelo perfecto para su atención. [75]

Para la correcta implementación de esta técnica es conveniente mencionar la importancia del *blur* en los elementos “cristalizados”: difuminando el fondo, conseguiremos que el usuario no se centre en ese elemento, si no en el que tiene encima. De esa manera podremos jugar con los colores y difuminados manejando el comportamiento de quien acceda a nuestra página. La conjunción de estas características es muy simple: habiendo dos elementos, uno encima del otro, se pone un color llamativo (en el de “abajo” siguiendo el eje Z) para que el usuario se vea llamado a mirar a la agrupación de elementos, pero se difumina para que, una vez fijada la vista, se preste atención a la parte nítida, que es la parte que queremos promocionar y vender (la de “arriba” siguiendo el eje Z)

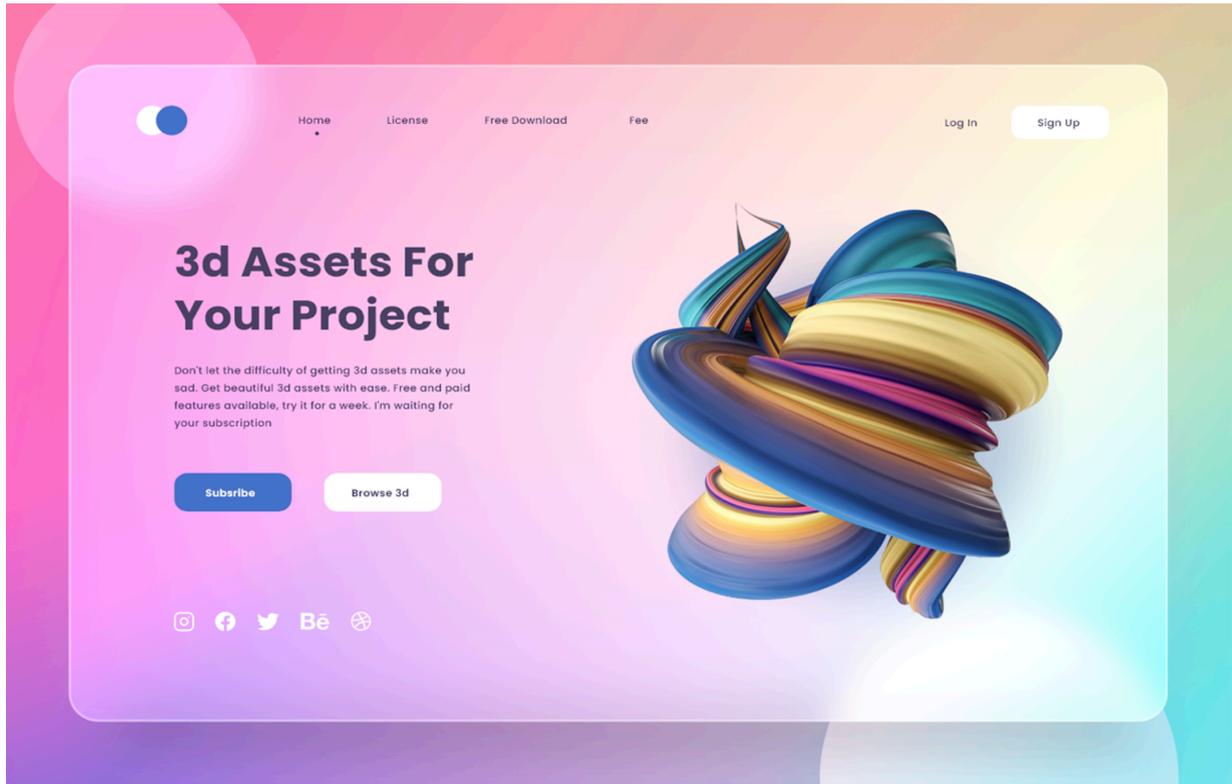


Figura 17. Página web con estilo glassmórfico [76]

Claymorfismo

La idea de esta tendencia de estilo es imitar el efecto de las superficies de arcilla en los elementos de la web. Naciendo del neumorfismo, se diferencia de este en la manera de aplicar los fondos y las sombras. Normalmente se suele optar por imitar efectos tridimensionales en los objetos por medio de degradados de color y juegos de luces y sombras.

Este efecto es muy útil cuando se quiere optar por un entorno 3D pero no se tienen los recursos para implementarlo. Con esta técnica, se pueden simular objetos tridimensionales sin la necesidad de renderizarlos en tiempo real. Además, la gran ventaja de esta técnica respecto al neumorfismo es que solventa los problemas de accesibilidad que esta generaba [77]. Sin duda una manera de hacer divertida e interactiva la navegación por la página.



Figura 18. Aplicación móvil con estilo claymórfico

Otros estilos

Existen muchas otras guías de estilo que, si bien no son “temáticas” como tal, son recursos igual de válidos para incorporarlos en una página web. *Neon Glowing/Holographic* [78] (recrear el efecto brillante de neones y elementos holográficos), *Brutalismo* [33] (Contrastes de colores muy brillantes y textos muy grandes y con tipografías duras), *Dark Themes* [34] (Recolorización de nuestra página con temas oscuros para una mejor usabilidad en entornos con poca luz, o simplemente por gusto del usuario) o por último *Funky Design* [79] (Utilización de “sketches” dibujados para representar los elementos, con el objetivo de generar menos estrés al usuario). En definitiva, Muchas otras oportunidades de crear una identidad propia a nuestro sitio web.



5.4. Ejemplos prácticos

Con el objetivo de probar la versatilidad de las tecnologías mencionadas en el documento, se van a realizar una serie de ejemplos comparando diferentes posibilidades de implementación para llegar así a las soluciones requeridas. Para ello, se documentará brevemente el código utilizado, indicando las ventajas y desventajas de cada método. Se considera oportuno combinar en caso de que así sea conveniente varias técnicas para un mayor aprovechamiento de los ejemplos y de las posibilidades que ofrecen las herramientas.

5.4.1. Parallax + Mouse Following

El primero de los ejemplos realizados consiste en la aplicación del efecto de paralaje utilizando una de las técnicas mencionadas anteriormente, el seguimiento del puntero del ratón. Haciendo un breve recordatorio, este efecto se consigue desfasando el movimiento de diferentes capas de la interfaz (textos, imágenes, botones o cualquier otro tipo de elemento) para así conseguir profundidad en la navegación.

La forma de implementar este comportamiento es ejecutando una función en bucle desde que carga la página (o desde otro evento de inicio, como puede ser pasar por encima de una sección), comprobando la posición del ratón en cada instante y modificando consecuentemente la posición y/o rotación de los distintos elementos de la página. Para conseguir esto ha de aplicarse un factor constante distinto para cada capa. Para las capas delanteras (cercanas al usuario, por decirlo así) el factor de movimiento ha de ser mayor, por lo que la constante utilizada debe ser también mayor. Por el contrario, las capas del fondo deben moverse poco, por lo que la constante será menor. La mejor manera de asimilar este concepto es imaginarse un paisaje en el campo. Mientras nosotros nos movamos, los árboles y arbustos cerca de nosotros se moverán mucho respecto a su posición inicial. Sin embargo, las montañas del fondo apenas se verán desplazadas.

El código JavaScript (con JQuery importado) a continuación muestra cómo se captura el movimiento del ratón y cómo se aplica a los elementos para conseguir las distintas velocidades de movimiento.

```
$(document).mousemove(function parallax(e) {
  // Take mouse coords
  let _mX = e.clientX;
  let _mY = e.clientY;

  // Take center of proportions always because window size could have change
  let _width = window.innerWidth/2;
  let _height = window.innerHeight/2;

  // Starting on center (50% 50%) lets make glasses run away from mouse pointer
  let _depth = `${50 - (_mX - _width) * 0.01}% ${50 - (_mY - _height) * 0.01}%`;

  // Updating element position
  $("#parallax").css('background-position', _depth);
}
```



Donde el factor de desplazamiento en este caso es 0.01 para los dos ejes. Se puede calcular una variable `_depth` para cada elemento si fuera necesario, modificando este valor a gusto del desarrollador. Finalmente, para aplicar los cambios simplemente hay que actualizar la posición absoluta o relativa del elemento, en este caso con `id="parallax"`.

Este ejemplo en concreto tiene como eje de movimiento el centro de la pantalla, tanto vertical como horizontalmente. Pero esto no es un requisito condicionante para el funcionamiento del efecto. Se puede aplicar solo movimiento en un eje y no necesariamente en ambos. También se puede tomar como centro cualquier otro punto de la pantalla, como una esquina, o incluso las coordenadas de otro elemento.

La conclusión que saco de esta técnica es que el desarrollador implemente él mismo el código JavaScript, ya que le ahorrará muchos quebraderos de cabeza a la hora de controlar el movimiento con respecto a un elemento en concreto y no es una función que requiera de mucho esfuerzo codificar. No obstante, utilizando la librería **ParallaxJS** se puede controlar también el movimiento, aunque hará falta entender la documentación para su uso. Se detallan ejemplos con esta librería en el siguiente apartado.

5.4.2. Parallax + scrolling

La siguiente prueba, de nuevo relacionada con el efecto de paralaje, tomará como objetivo el porcentaje de página navegada en vez de la posición del ratón como en el caso anterior. La ejecución de este caso es muy similar al anterior solo que el movimiento que altera la posición de los efectos se hará con el control de desplazamiento del dispositivo (rueda del ratón, trackpad en portátiles o slide de dedos en dispositivos táctiles). Por eso, para entender el proceso, se recomienda que sea el eje donde nos estamos desplazando al hacer scroll, aunque se puede hacer en ambos o incluso en el eje inverso. Por supuesto todo depende del caso donde se aplique y de la experiencia que queramos conseguir. Lo primordial en cualquiera de los casos es detectar el movimiento de scroll en cada momento, tal y como permite JQuery con el siguiente fragmento de código:

```
$(document).on({ scroll: function () {  
    ...  
    }  
});
```

El código restante para lograrlo es bastante parecido al caso anterior. Primeramente se debe controlar la posición con respecto al eje. En este caso se debe tener en cuenta la posición del elemento en pantalla desde la hora que se muestra en ella haciendo el scroll. Una vez en ella, podremos mover la posición del elemento tomando como referencia base el valor 1, es decir, >1 o <1 dependiendo si queremos que tenga más o menos velocidad que el scroll respectivamente. En el caso a continuación se aplica velocidad $\times 1.2$, por lo que se desplazará más rápido.

```
$("#parallax").css("transform", 'translateY(${distance * 1.2}px)');
```

Con el fragmento `var from_top = $(document).scrollTop()` sabremos qué distancia se ha navegado mediante el scroll o, lo que es equivalente, la posición actual de la barra de desplazamiento respecto al total de



su recorrido posible. A partir de ahí, simplemente tendremos que controlar las coordenadas en las que el elemento aparece y desaparece de pantalla y aplicar el desplazamiento en función de ellas. Esto se hace para activar el efecto solo cuando el elemento va a mostrarse o se muestra ya en pantalla y no durante toda la página, evitando así una sobrecarga de cálculos que puede verse reflejado negativamente en el rendimiento. Por supuesto, no solo el atributo de posición es modificable. Se puede alterar su rotación, su zoom, su blur, su opacidad... en definitiva, cualquier propiedad de estilo que se nos ocurra, incluso el fondo, técnica que nos será útil para conseguir otro efecto tal y como se explica en el apartado 5.4.2.1. *Framing by framing*.

5.4.2.1. Librerías

El uso de librerías ya diseñadas es una gran ventaja a la hora de escribir código, sobre todo en aspectos como el tiempo requerido y el rendimiento de trabajo (el refinamiento de estas herramientas normalmente es superior al del código que implementemos nosotros mismos, sobre todo por la gran comunidad que da soporte y reportes para correcciones y mejoras). Una de las librerías que podemos utilizar es **RellaxJS** [80]. Con ella, simplemente debemos añadir la clase `class="rellax"` (o la que se quiera realmente, como se verá más adelante) y a continuación añadir el atributo `data-rellax-speed="1"`, siendo este valor modificable por la velocidad que queramos en el elemento. Ahora simplemente nos queda importar a nuestro código el script desde el CDN (`src="https://cdn.rawgit.com/dixonandmoe/rellax/master/rellax.min.js"`) y ya podremos utilizar la funcionalidad. El método de utilización se realiza mediante la llamada a la función homónima, donde se indique la clase que se va a ver afectada por el efecto, y un objeto en formato JSON donde se indique si el desplazamiento es vertical y/u horizontal.

```
</script>
  var rellax = new Rellax('.rellax', { horizontal: true, vertical: false });
</script>
```

Otra librería a la que se puede recurrir, esta vez con una comunidad incluso mayor que en el caso anterior, es **ParallaxJS** [41]. Se puede leer la documentación en su Github, al cual se puede acceder directamente desde el enlace citado en 7. *Bibliografía y Referencias*. Su uso se basa también en la inserción de atributos en los componentes HTML. Simplemente habría que añadir las modificaciones a gusto del resultado que deseemos (por supuesto siguiendo la documentación) y finalmente controlar el/los elemento(s) padre donde van a ubicarse los objetos afectados por el efecto de paralaje. Primeramente habría que importar el código JavaScript, que podremos descargar desde la misma página de la documentación, y a continuación llamar a la función pasando como parámetro el elemento. En el siguiente código, se recupera el elemento utilizando JavaScript + JQuery:

```
var elem = $('#elem');
var parallaxInstance = new Parallax(elem);
```

Donde, en el Código HTML, se habría creado un elemento padre con `id="elem"` donde se alojan los elementos con los atributos que utiliza la librería. Todas las capas necesitan mínimo del atributo `data-depth="x"` para poder aplicar la jerarquía de profundidad en los elementos. El movimiento se ve influenciado por la profundidad, por lo que es obligatoria la incorporación del atributo.

```
<div id="elem">
```



```
<div data-depth="0.5">Primera capa</div>
<div data-depth="1.0">Segunda capa </div>
</div>
```

Comparando las distintas maneras de implementación, se considera que el uso de cada una de ellas depende de la magnitud del uso del efecto en la página. Si solo se quiere aplicar en un elemento puntual, la mejor opción es hacer un pequeño JavaScript de no más de 15~20 líneas donde realice esta funcionalidad. Si se quiere trabajar de manera más seria con el efecto, labrando un resultado que requiera de una mayor organización en la profundidad y traslación de los elementos, lo mejor es usar alguna de las dos librerías. Ambas cumplen bien con la función de paralaje mientras se hace scrolling, aunque, si queremos aplicar paralaje con otro tipo de inputs a mayores, la mejor opción es el uso de ParallaxJS por encima de RellaxJS.

5.4.2.1. Framing by framing

La implementación de esta técnica se basa en el mismo bloque de código que el efecto visto en el apartado **5.4.2 Parallax + Scrolling**. La diferencia reside en que en vez de cambiar las propiedades de traslación, rotación o similares del objeto, en esta ocasión ha de modificarse la url de la fuente de la imagen (o del background, pero técnicamente se puede considerar lo mismo). El único requisito para lograr el efecto es, como bien se menciona en el apartado relacionado con esta técnica, un buen desmenuzamiento del vídeo y un ratio de actualización acorde con la distancia de scroll y la longitud del vídeo.

5.4.3. Horizontal scrolling

Sin abultar más que 5 líneas de JavaScript podremos lograr este efecto tan influyente en la inmersividad. Simplemente se ha de recuperar el evento de la *rueda* del ratón (o del *double-finger scroll* si estamos en un trackpad o un dispositivo móvil), evitar que haga su comportamiento normal, y darle el nuevo comportamiento desplazando la pantalla en su equivalente del eje horizontal. Suponiendo un elemento `<main>` en nuestro archivo HTML, el fichero en JavaScript quedaría así:

```
const scrollContainer = document.querySelector("main");

scrollContainer.addEventListener("wheel", (evt) => {
  evt.preventDefault();
  scrollContainer.scrollLeft += evt.deltaY;
});
```

Es importante darse cuenta de que no es que estemos implementando un objeto con ancho mayor al 100%, si no que, estamos transformando el movimiento natural que hace el usuario en el eje vertical al eje horizontal. Por supuesto hay que hacer una página que supere la dimensión del ancho de la pantalla, pero al igual que en una página normal el ancho permanecería al 100%, en este caso el alto permanecerá al 100%. Dicho de otra manera, si tengo una página normal que me permite desplazarme 5 veces el tamaño de mi pantalla, tengo ancho=100% y alto=500%. Pues, el homólogo en horizontal scrolling tendría alto=100% y ancho=500%.



5.4.4. Click Counter

El uso de esta técnica se puede compaginar con animaciones o cambios en los atributos de estilo en los elementos afectados. Por medio de una simple línea de JavaScript seremos capaces de realizar esta tarea tan trivial pero efectiva. En el caso mostrado a continuación, el elemento ha de tener el identificador `click-id`.

```
document.getElementById("click-id").innerHTML = clicks;
```

Esta línea de código ha de colocarse en un bucle para que sea controlada todo el tiempo. JQuery ofrece una solución donde, por cada click, se activará el evento y podremos ejecutar la función que queramos. En esta nueva solución, podremos colocar el contador dentro de esta función:

```
$("#click-id").on("click", function(e){  
    ...  
});
```

Y una vez se llegue al número de clicks deseado, que ocurra el evento en cuestión. En definitiva, podemos elegir si controlar los clicks de toda la página o solo de un elemento.

5.4.5. Drag & Drop

Para conseguir este efecto se ha de construir una “capa de protección” alrededor del recorrido que se debe realizar con el ratón, haciendo que esa sea el área segura por donde se puede arrastrar el elemento. Controlando cuando se ha hecho click, se puede proceder al siguiente paso que es detectar que el ratón no salga de dicho área hasta llegar a la meta, donde puede dejar de hacer click. Se deberá controlar si se sale de la zona de seguridad o si se suelta el click, mostrando que no se ha realizado correctamente el proceso, con un mensaje o reiniciando el estado por ejemplo. La capa de protección se puede crear en JavaScript recuperando los vértices del área por donde pueda navegar el objeto arrastrable. En el caso más simple, un rectángulo, los puntos de control serían las 4 esquinas. Por tanto, una coordenada (X,Y) debe estar comprendida tal que $X_1 < X < X_2$, $Y_1 < Y < Y_2$.

El siguiente paso sería controlar la posición del ratón con el mismo código que usábamos en **5.4.1. Parallax + Mouse Following**:

```
$(document).mousemove(function parallax(e) {  
    let _mX = e.clientX;  
    let _mY = e.clientY;  
  
    ...  
}
```

Para continuar debemos detectar a continuación el click en el elemento tal y como vemos en el siguiente código. Mientras se mantenga el click, si el ratón se mueve (dentro del área de la capa de protección) se debe mover el elemento arrastrable a la posición del ratón.



```
$('#melem').on("mousedown",function(){
    // If (Mouse_coords are in safe_area) {
    //     Update element position
    // }
});

$('#elem').on("mouseup",function(){
    // Reset
});
```

No obstante, con alguna librería como *Interact.js* [81] el proceso de creación de un elemento *draggable*, del recorrido y de la meta es mucho más sencillo. Todo depende de la complejidad del movimiento a realizar por el usuario en el recorrido del artefacto.

5.4.6. Objetos 3D

En este apartado se quiere comprobar si la implementación de objetos tridimensionales se puede realizar de manera trivial, o sin embargo requiere de un proceso más complejo. Para ello, primeramente crearemos el objeto en Spline, y posteriormente lo importaremos y trataremos mediante la librería ThreeJS.

5.4.6.1. Spline

El uso de esta herramienta es debido a la facilidad de creación de los objetos tridimensionales con respecto a aplicaciones estáticas (no web) como pueden ser Blender o ZBrush entre otras. Por supuesto este tipo de aplicaciones dan resultados mucho más profesionales, pero como esa tarea no ha de ser realizada por los desarrolladores web, si no por un equipo de expertos en modelaje y animación tridimensional, se omite el uso de estos programas para este ejemplo. Volviendo a nuestro ejemplo, para crear nuestro objeto, debemos acceder a la página de **Spline** [50] y entrar en el editor. Nos pedirá una cuenta, con posibilidad de suscripción de pago para mejores características. No necesitamos pagar nada para realizar el ejemplo, así que continuamos con el proceso de creación. En mi caso, como este trabajo es más indicado para ser realizado por un diseñador, voy a optar por coger un proyecto ya creado. En concreto, he importado el proyecto *Molang 3D*.



Figura 19. Caso práctico Objeto 3D - proyecto *Molang 3D* de Spline



5.4.6.2. ThreeJS

A continuación importaremos el archivo a nuestra página web. Para ello he creado un proyecto en el framework **React** [82] donde gracias a la librería **GLTFLoader** se pueden cargar los ficheros necesarios. Para desplegar el objeto se requiere mínimo de su esqueleto poligonal (formato .gltf) y de sus texturas (formato de imagen. Normalmente .png para que, aun siendo un formato compresor, no pierda calidad ni los divisores transparentes entre texturas de diferentes cuerpos). Por supuesto hay que importar también la librería de **ThreeJS** [45] para renderizar el objeto 3D.

```
import * as THREE from 'three';
import { GLTFLoader } from 'three/examples/jsm/loaders/GLTFLoader.js';
```

Una vez importado el código necesario para la correcta muestra del objeto, se cita a continuación cuál es el código para referenciar y mostrar nuestro objeto 3D. Lo primero es crear una escena de ThreeJS. A continuación, por medio de la librería GLTFLoader, crearemos el objeto que se renderizará en la web.

```
const test_threeJS = new SceneInit('myThreeJsCanvas');
test_threeJS.initialize();
test_threeJS.animate();

let loadedModel;
const glftLoader = new GLTFLoader();

glftLoader.load('./assets/shiba/scene.gltf', (gltfScene) => {
  loadedModel = gltfScene;

  gltfScene.scene.position.y = 3;

  test.scene.add(gltfScene.scene);
});
```

Por último, queda aplicar el comportamiento que queramos darle al objeto. Podemos aplicarle una rotación o desplazamiento infinitos, o por ejemplo podemos hacer que siga al ratón con el mismo código que vimos en el apartado **5.4.1. Parallax + Mouse Following**. También, para mayor dinamismo, si un objeto está compuesto de varios objetos, podemos importarlos individualmente y darles movimientos con pequeñas diferencias, algo así como el efecto de paralaje, pero con los objetos 3D. Ese desfase ayudará a conseguir una inmersión aun mayor que la que conseguirá un movimiento errático periódico.

Esta tecnología ofrece una gran potencia y comodidad de uso, sobre todo cuando es utilizada con un framework como puede ser React. El uso con plantillas planas de HTML + CSS + JavaScript (sin ningún tipo de framework) puede endurecer el tiempo y coste de desarrollo, pero es perfectamente posible la incorporación de modelos 3D en ellas. En definitiva, se recomienda utilizar estos recursos con librerías ya preparadas para su implementación, aunque no se cierra la puerta a otras alternativas.



5.4.7. 3D Mapping

En esta ocasión se ha pretendido utilizar las tres herramientas de creación de mapas tridimensionales. El principal problema ha sido que, para acceder a funcionalidades especiales con las que conseguir inmersividad, se necesitaba el acceso a los planes de pago tanto para ArcGIS como para Google Maps. En este sentido, se ha desarrollado un ejemplo en Mapbox, una herramienta gratuita donde se podrá analizar la facilidad de implementación del recurso de mapas tridimensionales.

5.4.7.1. Mapbox

Para comenzar con el desarrollo de un sitio web que implemente Mapbox [51], debemos ir a su editor **Mapbox Studio** [83]. Una vez en él, debemos dar el estilo que queramos al mapa. En mi caso, las configuraciones que he hecho (a parte obviamente del color de los elementos) es el índice de extrusión de los edificios, la separación por líneas de nivel del terreno y el enmascaramiento de elementos que no quería que se reflejaran, como carreteras secundarias o caminos. Cuando hayamos terminado de editar podremos pedir a la herramienta que nos dé el identificador de nuestro mapa. También necesitaremos el token de acceso de nuestra cuenta para poder utilizarlo.

El siguiente paso es implementar en la página web el canvas donde se va a desplegar el mapa. Para ello se necesita importar los ficheros de Mapbox desde su CDN, y crear un elemento `<div/>` con `id="map"` (puede ser cualquier id, luego se indicará a la herramienta cual ha de ser el contenedor, pero por comodidad se decide usar *map*).

```
<head>
  ...
  <!-- mapbox -->
  <script src='https://api.mapbox.com/mapbox-gl-js/v2.8.2/mapbox-gl.js'></script>
  <link href='https://api.mapbox.com/mapbox-gl-js/v2.8.2/mapbox-gl.css' rel='stylesheet' />
</head>

<body>
  <div id="map"></div>
</body>
```

A continuación debemos dar el comportamiento y límites que queramos al mapa. El método para realizarlo consiste en añadir los atributos correspondientes (ver documentación para conocerlos todos) a través de una de las funciones importadas con el fichero *mapbox-gl.js*. De esta manera crearemos una variable con la que podremos trabajar en el resto del código. En este caso la configuración que se ha elegido es la siguiente: ángulo (pitch) 0 con un posible ángulo de 60, el centro inicial del mapa, el máximo y mínimo zoom, la dirección en la que se mira (bearing) esté un poco inclinada hacia el noreste, que no se pueda hacer zoom por doble click, solamente por la rueda del ratón, que no se pueda hacer scroll ni zoom en la página, que no se permita cambiar el bearing ni el pitch haciendo drag, y finalmente una optimización de antialiasing para los polígonos.



```
const map = new mapboxgl.Map({
  container: 'map', // container ID
  style: 'mapbox://styles/juananmgz/xxxxxxxxxxxxxxxxxxxxxxxx', // style URL
  pitch: 0,
  maxPitch: 60,
  center: [-2.169, 43.285],
  zoom: 10,
  minZoom: 10,
  maxZoom: 18,
  bearing: 60,
  doubleClickZoom: false,
  boxZoom: false,
  scrollZoom: false,
  dragRotate: false,
  optimizeForTerrain: true,
  dragPan: false,
});
```

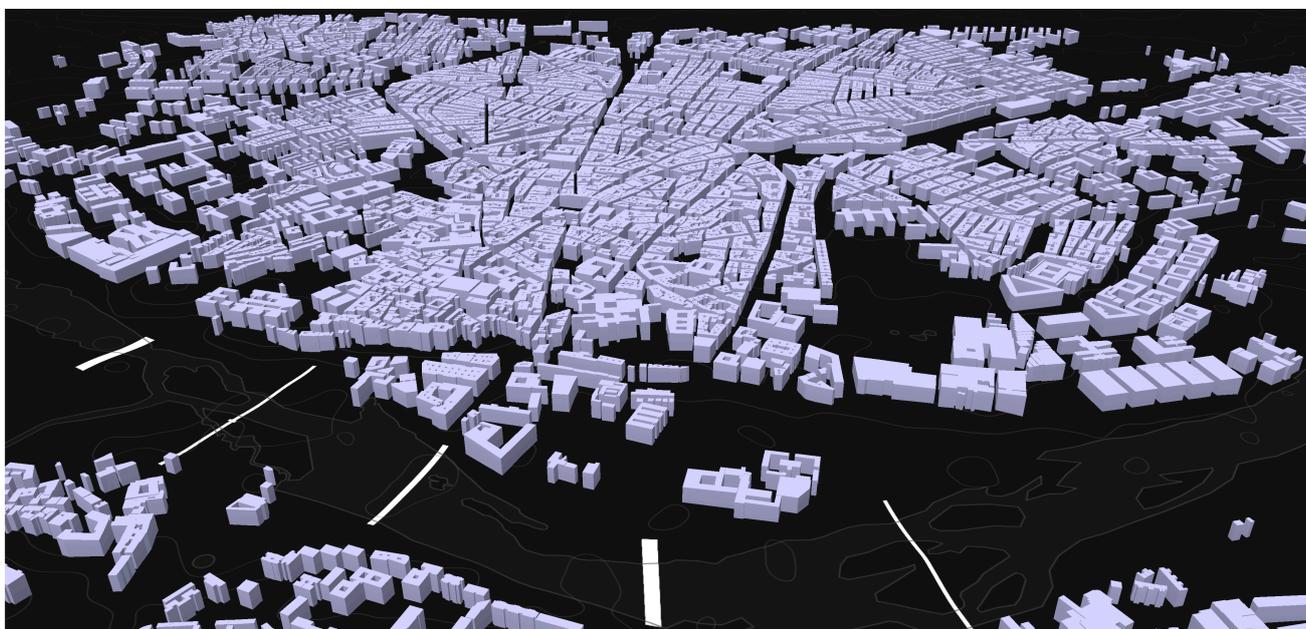


Figura 20. Ejemplo práctico de Mapbox - Vista de cerca de la ciudad de Salamanca

El próximo paso es añadir **marcadores** al mapa. El aspecto de estos se puede modificar en el fichero JavaScript como veremos a continuación, aunque se puede crear una hoja CSS e imponer un estilo que encaje más, o incluso que tenga un comportamiento más complejo. Para añadir los marcadores se debe crear un array con la información de cada uno de los puntos de la siguiente manera:

```
const markers = [
  {
    "type": "Feature",
    "properties": {
      "title": "San Sebastián (Donostia)",
      "description": "San Sebastián Vasque City"
    },
    "geometry": {
      "coordinates": [
```



```

        -1.98224,
        43.32213
      ],
      "type": "Point"
    },
    "id": "18a834791689251695b158fb95399a49"
  },
  {
    ...
  },
  ...
]

```

Para añadirlos al mapa se ha de crear una capa donde se representen los puntos. Para ello haremos uso de la variable creada anteriormente:

```

map.on('load', () => {
  // Adding markers to mapinfo.
  map.addSource('points', {
    type: 'geojson',
    data: {
      "features": markers,
      "type": "FeatureCollection"
    }
  });

  // Markers Layer.
  map.addLayer({
    'id': 'circle',
    'type': 'circle',
    'source': 'points',
    'paint': {
      'circle-color': '#000000',
      'circle-opacity': 0.7,
      'circle-radius': 8,
      'circle-stroke-width': 2,
      'circle-stroke-color': '#ffa500',
    }
  });

  ...
});

```

El resultado de aplicar los marcadores queda reflejado en la vista cenital que conseguimos al hacer zoom out en la página. En la siguiente imagen se pueden ver cuatro marcadores con forma de círculos naranjas.

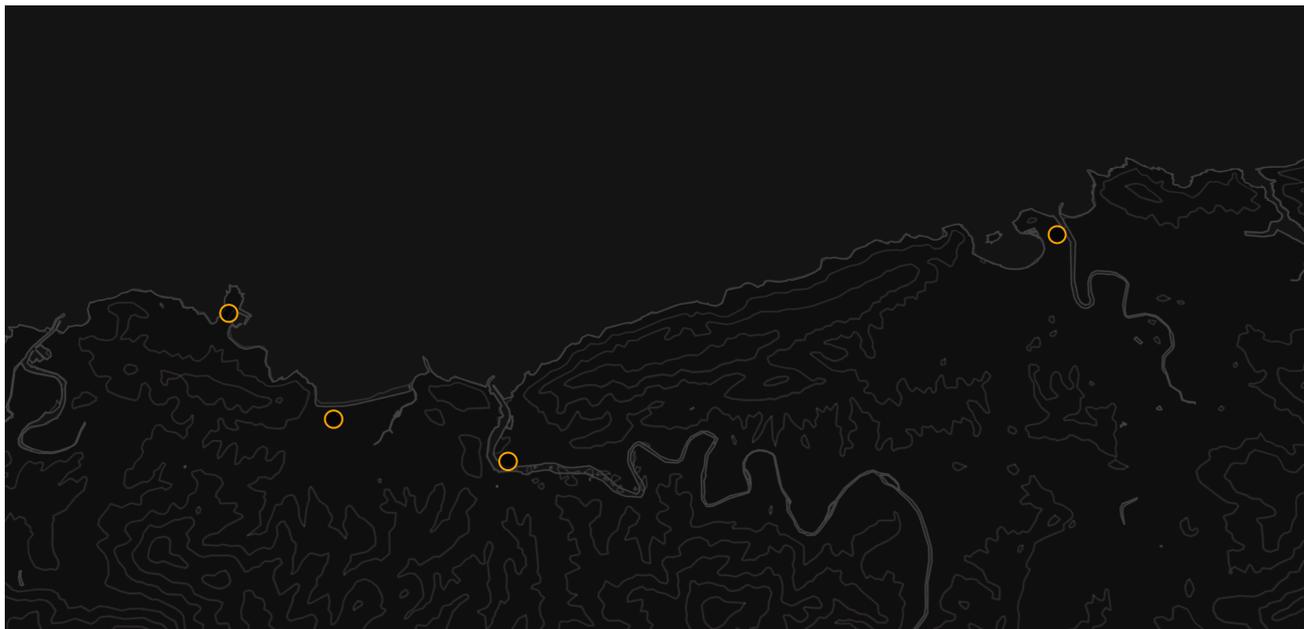


Figura 21. Ejemplo práctico de Mapbox - Marcadores

Para añadir comportamiento a los marcadores, podemos hacer que al clicaros se desplace al usuario hasta la posición del marcador. Podemos jugar con el ángulo para que una vez se vaya acercando, la vista en plano cenital coja ángulo hasta convertirse en una vista picada (como en la *Figura 19*). Para ello debemos añadir el siguiente código dentro de la función `addSource`, justo donde están los tres puntos en el código anterior.

```
// Center the map on the coordinates of any clicked circle from the 'circle' layer.
map.on('click', 'circle', (e) => {
  markerTransitionHasEnded = false;
  goal_coords = e.features[0].geometry.coordinates;

  map.flyTo({
    center:e.features[0].geometry.coordinates,
    zoom: ZOOM_ON_TRANSITION,
    pitch: 60,
    essential: true,
    bearing: 0,
    speed: 0.33,
  });

  map.setMaxBounds()
  map.scrollZoom.disable();
  map.dragPan.disable();
});

// Change the cursor to a pointer when the it enters a feature in the 'circle' layer.
map.on('mouseenter', 'circle', () => {
  map.getCanvas().style.cursor = 'pointer';
});

// Change it back to a pointer when it leaves.
map.on('mouseleave', 'circle', () => {
  map.getCanvas().style.cursor = '';
});
```



En resumen, el código hace que cuando se pase por un círculo cambie el estilo del ratón para que el usuario sepa que es un elemento clicable, y por supuesto que vuelva a cambiar cuando deja de estar encima. También controla el evento de clicar sobre un elemento 'circle', es decir, sobre un marcador. El comportamiento hace que se bloquee la posibilidad de que el usuario se mueva mientras se le desliza hasta la posición que ha pedido.

5.4.7.2. Conclusión

El uso de Mapbox es de gran facilidad y con una curva de aprendizaje no muy elevada. Si bien es verdad que hay que conocer cada parámetro a la perfección, es una muy potente herramienta con una interfaz de control muy bien lograda para su uso. Además, la fácil personalización del mapa a través del estudio de edición es un gran punto a su favor a la hora de adaptarse a la estética requerida en el proyecto.

Aunque no se puede comparar directamente con las otras dos herramientas ya que no se han probado, si se quiere resaltar que, si estas herramientas son tan conocidas y usadas, es porque tienen una comunidad que las apoya y confía en ellas. Como toda herramienta, puede tener sus puntos fuertes y flacos, pero con seguridad son grandes tecnologías por las que optar.

5.4.8. Temáticas

La aplicación de las diferentes temáticas no es, o mejor dicho no debe ser elección del equipo desarrollador. Se debe pasar por un proceso de diseño de interfaz y de experiencia de usuario, en el que la decisión sobre qué temática aplicar ha de debatirse y posteriormente refinar antes de empezar con la implementación. Una vez finalizado el resultado del prototipo, el equipo desarrollador ha de implementarlo eficazmente. A continuación veremos cómo aplicar estilos a las cajas y botones para conseguir el acabado deseado.

5.4.8.1. Neumorfismo

Para lograr el efecto de neumorfismo debemos jugar con las sombras interiores y exteriores para simular la profundidad identitaria de este estilo. La localización de los claros u oscuros depende de si se quiere conseguir un relieve positivo o negativo, es decir: un saliente o una oquedad. En el siguiente ejemplo se muestra un relieve.

```
background: linear-gradient(145deg, #f6f6f6, #cfcfcf);
box-shadow: 17px 17px 34px #b6b6b6,
            -17px -17px 34px #ffffff;
```

Por el contrario, el código a continuación muestra una oquedad. La diferencia es el atributo "inset" en la aplicación de las sombras, lo que hace las sombras queden en el interior del elemento.



```
background: #e6e6e6;  
box-shadow: inset 5px 5px 12px #cacaca,  
            inset -5px -5px 12px #ffffff;
```

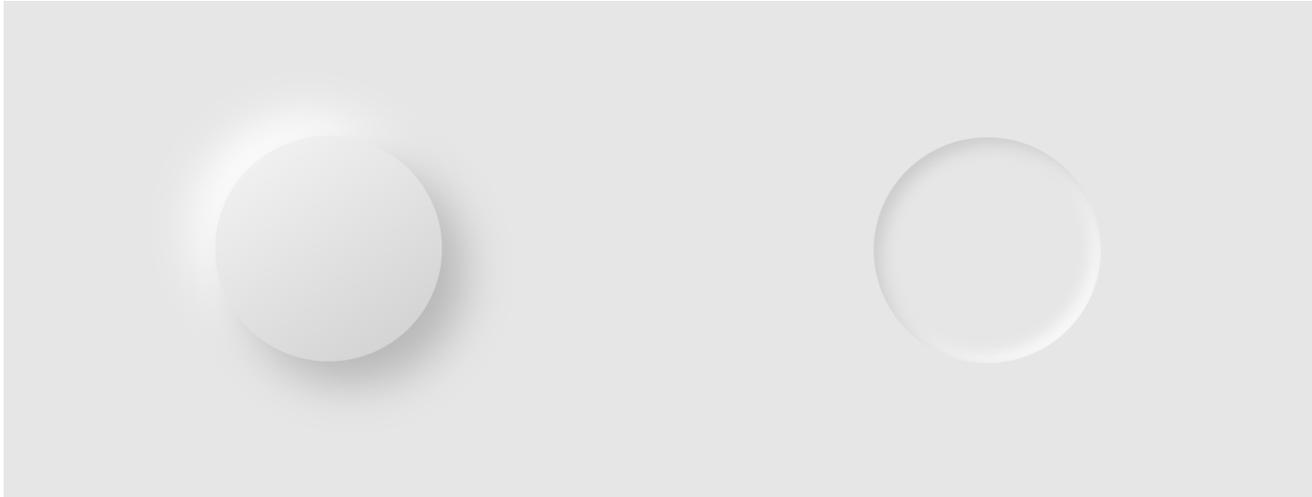


Figura 22. Ejemplos prácticos de relieve y oquedad conseguidos con neumorfismo

Para probar estos ejemplos, se recomienda entrar en el generador neumorphism.io [84], donde se podrán crear diferentes combinaciones que serán útiles en caso de aplicar esta técnica.

5.4.8.2. Claymorfismo

De manera similar, se muestra un ejemplo práctico de cómo crear el efecto claymórfico mediante hojas de estilo CSS. El código necesario es el siguiente:

```
backdrop-filter: blur(5px);  
background-color: rgba(255, 255, 255, 1);  
border-radius: 26px;  
box-shadow: 35px 35px 68px 0px rgba(145, 192, 255, 0.5), inset -8px -8px 16px 0px  
            rgba(145, 192, 255, 0.6), inset 0px 11px 28px 0px rgb(255, 255, 255);
```

Se puede utilizar el generador de claymorfismo [85] para probar distintas configuraciones. El efecto se puede ver en la siguiente imagen.

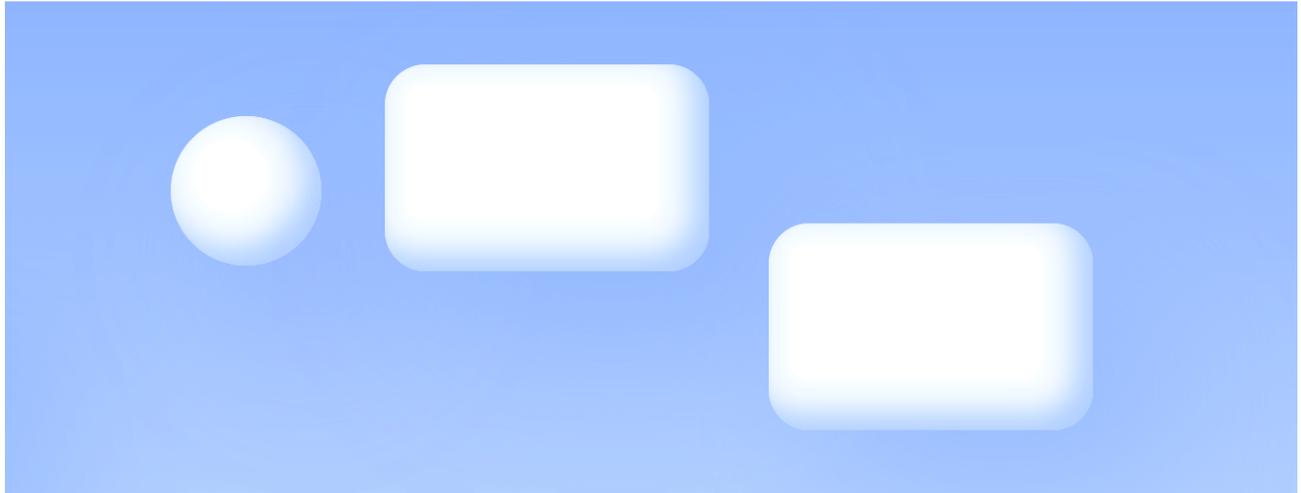


Figura 23. Ejemplo práctico de claymorfismo

5.4.8.3. Glassmorfismo

Por último, se incorpora un breve fragmento de código con los parámetros necesarios para crear el estilo de glassmorfismo. De nuevo, se puede optar por usar un generador [86] para comprobar el funcionamiento de este estilo e investigar nuevas combinaciones en los parámetros. Este es el código utilizado para el ejemplo de la foto:

```
background: rgba( 255, 255, 255, 0.25 );  
box-shadow: 0 8px 32px 0 rgba( 31, 38, 135, 0.37 );  
backdrop-filter: blur( 4px );  
-webkit-backdrop-filter: blur( 4px );  
border-radius: 10px;border: 1px solid rgba( 255, 255, 255, 0.18 );
```



Figura 24. Ejemplo práctico de glassmorfismo

6. RESUMEN Y CONCLUSIÓN



Que el impacto de la Experiencia de Usuario en el cliente afecta directamente a la percepción de la marca, a la capacidad de atención del usuario y a la conversión de los visitantes en clientes es un hecho que ha quedado reflejado a lo largo del documento. La importancia de una buena usabilidad ya es en sí un pilar fundamental en el desarrollo del frontend de páginas web. Pero, más allá de esta globalización del término ‘usabilidad’, una de las principales tendencias que hay en la actualidad y que demuestra una gran relevancia en la capa de visualización es la inmersividad.

Como bien hemos visto a lo largo del documento, existen multitud de recursos para sumergir al usuario en una experiencia innovadora, en la que sienta atracción por el producto y por la manera de mostrarlo. Resumiendo lo ya mencionado, una de las posibilidades tecnológicas que más se han desarrollado en los últimos años son las animaciones. La evolución de los navegadores y de los recursos físicos donde se ejecutan (computadoras/móviles/tablets/otros dispositivos IoT) hacen que las mociones de los elementos se lleven a cabo de manera fluida, lo que permite refinar y dar un acabado profesional a los resultados de las mismas. Otra de las opciones que han visto incrementado su uso de una manera abrumadora son los elementos 3D. Gracias a la estandarización de WebGL y del soporte que se le ha dado en los navegadores, y por supuesto de la mejora de prestaciones y recursos (al igual que en el caso de las animaciones) han hecho que sean cada vez más los sitios web que deciden incorporar elementos, interacciones u efectos tridimensionales para captar la atención de los posibles clientes. Por supuesto, el comportamiento de estos artefactos vanguardistas no viene embebido en el conocimiento de los usuarios. O no desde los albores de internet. Su uso ha sido condicionado por la evolución del resto de tecnologías en las que se basan, basadas en la usabilidad derivada del campo de la interacción persona-ordenador. Su método de interacción por tanto, si bien no se puede definir como trivial, sí es más fácil de entender y asimilar si se conocen los conceptos en los que se basan. Prueba de esto son las interacciones desplegadas al hacer movimientos como arrastrar y soltar (drag & drop) o scroll. Entendiendo por ejemplo el funcionamiento del scroll, de cómo una página normal se comporta con esta interacción, se puede modificar el comportamiento de este tipo de entrada, en este caso por la rueda del ratón, para poder ampliar las posibilidades de inmersión dentro del sitio web.

La modificación de estas entradas triviales se puede implementar en nuestras páginas para añadir comportamientos nuevos, como bien puede ser las recibidas por medio del ratón, traducido como movimiento o como clicks, o bien del teclado, traducido como teclas o bien combinaciones de las mismas. El uso de estos inputs puede servir como disparadores para ejecutar las técnicas que mencionábamos antes, o bien para desplegar otro tipo de técnicas menores. Pero, no todos los elementos han de ser dinámicamente vistosos, de tamaño colosal o de vivos colores para que destaquen. Se puede lograr la inmersividad de muchas otras maneras. Ejemplo de esto es la conversión de texto a formato vectorial, para poder animar su entrada, salida o incluso su contenido. Del mismo modo, se pueden convertir otros objetos en vectores, o bien trabajar con las transformaciones intrínsecas de las hojas de estilo y de scripts para crear microinteracciones que den dinamismo y detalle a cada uno de los rincones que puedan ser visitados a la hora de navegar por la web.



El uso de estos elementos por si solos no se traduce en una página web más profesional, refinada y con la que logremos la experiencia de inmersividad. Es más, un uso incorrecto de estos elementos puede lograr el efecto contrario. Para conseguir encauzar la implementación de los mismos de una manera amigable, se recomienda el uso de temáticas con las que acompañar los distintos artefactos. Algunas de estas temáticas, como el minimalismo, son ampliamente conocidas incluso fuera del ámbito web. Pero otras como el neumorfismo, el glassmorfismo o el claymorfismo son quizás menos conocidas, y por tanto es menester informarse bien sobre ellas y sobre cómo pueden ayudarnos a la hora de incorporar elementos con fines de inmersión.

Igual que es importante una implementación impoluta de las técnicas mencionadas anteriormente, es trascendental la correcta colocación de los elementos inmersivos en la interfaz. Partiendo de prototipos generados por expertos en usabilidad, gran parte de ese trabajo ya estará realizado y refinado por múltiples iteraciones en el proceso de creación validado por métricas teóricas y por usuarios de prueba de manera más práctica. Pero este punto de partida ha de ser complementado en el proceso con las técnicas de desarrollo del equipo de informáticos encargados de transformar el prototipo en una maqueta funcionalmente completa. Por eso, como desarrolladores, se debe tener en cuenta nuestra función dentro del proceso de creación de las interfaces de usuario, para poder crear así una simbiosis entre los equipos de marketing, de diseño gráfico/usabilidad y de implementación que se vea reflejada en el resultado.

Una mala implementación de la experiencia de usuario puede hacer que el usuario se sienta frustrado, lo que se verá reflejado negativamente en los beneficios de la página. La utilización de elementos inmersivos puede hacer que las interfaces sean más complejas, y por tanto puedan acarrear más errores en su utilización. El resultado de un mal diseño de experiencia de usuario se ve reflejado en: [87]

- Costes extras
- Creación de cuentas innecesarias
- Procesos complicados o demasiado largos
- Errores y Crashes
- Falta de confianza en el sitio por parte de los usuarios

En definitiva, en la pérdida de ingresos y exceso de gastos por culpa de errores directos o indirectos en el proceso de navegación de la página web. No obstante, un buen planteamiento y trabajo en equipo con los departamentos ofrecerá resultados cuanto menos prometedores, amortiguando el impacto de estos hándicaps o llegando incluso a desvanecerlos. Pero, para lograr que la improvisación tenga la calidad necesaria, se debe elegir la tecnología de implementación idónea.

Son varias las posibilidades de implementación de los recursos tecnológicamente hablando. En este trabajo se ha comprobado como utilizando librerías y frameworks se pueden conseguir grandes resultados de manera rápida, aunque no son páginas profesionales, sino más bien ejemplos prácticos de aplicabilidad. Esto demuestra que el acceso a estas tecnologías está al alcance de cualquier persona que esté mínimamente formada en tecnologías web. Principalmente, estas herramientas se basan en funciones JavaScript previamente refinadas donde, a través de interfaces, se pueden llamar a los métodos para que se modifiquen los elementos deseados por medio del DOM. No son implementaciones complejas, pero es necesario tener una visión más allá del ámbito programático. Analizando con una visión creativa las herramientas, podremos combinarlas y obtener productos más refinados.



La principal conclusión que se saca de este análisis es que el proceso de desarrollo de una interfaz ha de ser un trabajo meticuloso desde la conceptualización de la idea hasta la implementación, pasando por las distintas etapas de refinamiento del/los prototipo/s. Por supuesto, la manera de mostrar esa idea debe conjugar con las capacidades de implementación, tanto tecnológicas como del equipo de desarrollo. Si casan todos estos elementos, se podrá conseguir una página impecable y con mucho potencial para vender la idea. El uso de elementos inmersivos añade dificultad al proceso, pudiendo ser crítica su utilización y desastroso el resultado, pero, utilizados con cautela y conocimiento, son una potentísima herramienta para que el mensaje del producto y de nuestra marca cale en el consumidor.

En contraparte, la potencia de la experiencia de usuario inmersiva también se puede ver reflejada en grandes ventajas: retención del usuario en la página, tanto para analíticas como para publicidad, interiorización del mensaje, producto y marca, apelación a interacción lúdica o en base a recuerdos, motivación de exploración y difusión del contenido de la página, etcétera. En definitiva, llamar la atención del usuario y sacar beneficios tanto directos (ventas) como indirectos (analíticas o publicidad).

Se ha conseguido por tanto realizar una búsqueda de elementos inmersivos utilizados en páginas web, de ver su influencia y las técnicas que utilizan. También se ha podido en base a ese aprendizaje ampliar los conocimientos sobre la usabilidad, la experiencia de usuario y por supuesto las técnicas para lograr la experiencia de usuario inmersiva. Por último, se han conseguido aplicar estas técnicas a ejemplos prácticos, viendo cuales son los límites, ventajas y desventajas de las tecnologías seleccionadas.

La experiencia de usuario es un campo en constante evolución incentivado por las corrientes y tecnologías que van surgiendo cada vez más y más rápido. La necesidad de destacar nuestra idea por encima de las demás, hace que se necesiten métodos para llamar la atención al usuario. La verdadera potencia de la inmersividad está solo por llegar. Son muchos los campos de aplicación que se verán influenciados por estas vertientes. Una de las más prometedoras lanzaderas de la inmersividad es el idílico metaverso, donde poder interactuar con todo tipo de objetos y entes. Entornos tridimensionales, animaciones o controles por voz o movimiento son recursos que ahora mismo disponemos, pero no tan refinadamente como se espera. Por ello, es una afirmación que todavía queda mucho proceso de evolución y refinamiento. Y es en este proceso, en este instante de tiempo y en ese lugar, donde los desarrolladores web han de apostar por la inmersividad en todas aquellas páginas donde se quiera mostrar su mensaje creativamente.

7. BIBLIOGRAFÍA Y REFERENCIAS



- [1] Jon H Kaas, Pooja Balam, «Current research on the organization and function of the visual system in primates, vol. 6, pp. 1-4,» 2014. [En línea]. Available: <https://doi.org/10.2147/EB.S64016>.
- [2] Osama Sohaib, Walayat Hussain, M. Khalid Badini, «Standards, User Experience (UX) and the Web Accessibility,» May 2011. [En línea]. Available: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.436.2892&rep=rep1&type=pdf#page=601>.
- [3] James Spillane, «6 Incredible Benefits of Immersive Experience in Business Marketing,» Junio 2017. [En línea]. Available: <https://www.business2community.com/small-business/6-incredible-benefits-immersive-experience-business-marketing-01856859#MpGJbFSEV3gvrUf3.97>.
- [4] N. Wright, «What are the top UX trends that will drive customer engagement this year?,» 21 January 2022. [En línea]. Available: <https://www.thedrum.com/opinion/2022/01/21/what-are-the-top-ux-trends-will-drive-customer-engagement-year>.
- [5] Sol González, «2022 Marketing Trends: Branding, Design, and User Experience,» 06 Abril 2022. [En línea]. Available: <https://www.cyberclick.net/numericalblogen/2022-marketing-trends-branding-design-and-user-experience>.
- [6] Susana Graciela Pérez Ibarra, José Rolando Quispe, Felipe Fernando Mullicundo, Daniel Alberto Lamas, «HERRAMIENTAS Y TECNOLOGÍAS PARA EL DESARROLLO WEB DESDE EL FRONTEND AL BACKEND,» 16 Abril 2021. [En línea]. Available: <http://sedici.unlp.edu.ar/bitstream/handle/10915/120476/Ponencia.pdf-PDFA.pdf?sequence=1&isAllowed=y>.
- [7] Apple, «Apple AirPods,» [En línea]. Available: <https://www.apple.com/es/airpods-pro/>.
- [8] C. V. T. p. l. negocios, «Ventajas de usar experiencia inmersiva en el marketing del negocio,» [En línea]. Available: <https://ticnegocios.camaravalencia.com/servicios/tendencias/ventajas-usar-experiencia-inmersiva-marketing-del-negocio/>.
- [9] Bernd B, «8 things I learned analyzing Apples product pages,» 6 Mayo 2020. [En línea]. Available: <https://uxplanet.org/8-things-i-learned-analyzing-apples-product-pages-9a5284681b37>.
- [10] Samsung, «Galaxy Z Flip4,» [En línea]. Available: <https://www.samsung.com/es/smartphones/galaxy-z-flip4/>.
- [11] L. P. 5, «Marseille,» [En línea]. Available: <https://marseille.laphase5.com/en>.
- [12] S. f. Business, «Skolkovo Park,» [En línea]. Available: <https://skolkovoforbusiness.ru/en/features/#home>.



- [13] Kopke, «Kopke - The oldest Port Wine House,» [En línea]. Available: <https://kopke1638.com/>.
- [14] Yusef Hassan Montero , «Introducción a la Usabilidad,» Noviembre 2002. [En línea]. Available: https://www.nosolousabilidad.com/articulos/introduccion_usabilidad.htm.
- [15] Cliff Kuang, Robert Fabricant , USER FRIENDLY. How the hidden rules of design are changing the way we live, work and play, 2019.
- [16] Nigel Bevana, Jurek Kirakowskib, Jonathan Maissel, «What is Usability?,» Septiembre 1991. [En línea]. Available: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.630.1555&rep=rep1&type=pdf>.
- [17] Gabriel Elías Chanchí G., María Clara Gómez Álvarez, Wilmar Yesid Campo M. , «Propuesta de una herramienta de inspección según los atributos de usabilidad de Nielsen,» Octubre 2019. [En línea]. Available: <https://www.proquest.com/openview/b15b1d1ae0fd63a1471d3fbd3faf6cd9/1?pq-origsite=gscholar&cbl=1006393>.
- [18] Jakob Nielsen, «Usability Engineering,» 1993. [En línea]. Available: <https://www.nngroup.com/books/usability-engineering/>.
- [19] Jakob Nielsen, «Usability Metrics: Tracking Interface Improvements,» Noviembre 1996. [En línea]. Available: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8740869>.
- [20] Jakob Nielsen, «Usability Inspection Methods,» Abril 1994. [En línea]. Available: <https://dl.acm.org/doi/pdf/10.1145/259963.260531>.
- [21] Tasha Hollingsed, David G. Novick , «Usability inspection methods after 15 years of research and practice,» Octubre 2007. [En línea]. Available: <https://dl.acm.org/doi/pdf/10.1145/1297144.1297200>.
- [22] María Paula González, Jesús Lorés, Afra Pascual, «Evaluacion Heuristica,» Universitat de Lleida, 2001. [En línea]. Available: <https://d1wqtxts1xzle7.cloudfront.net/56797831/Evaluacion-Heuristica-with-cover-page-v2.pdf?Expires=1662549430&Signature=d954ax1w6M9Kg-Ldl281Lvpjgvv~6ebXx0ERVqdzIKIoM4cKfL7yMWL9E4p7pEFvExiyLDKrpELDst9KpEexholGRvrkv821BpPfmNA21706eVerZ-pE4gWtmn0Ow~39QriXhv>.
- [23] C. Wharton, J. Bradford, J. Jeffries, M. Franzke, «Cognitive Walkthrough,» 22 Octubre 2012. [En línea]. Available: <https://ext.eurocontrol.int/ehp/?q=node/1622>.
- [24] Marilyn Hughes Blackmon, Peter G. Polson, Muneo Kitajima, Clayton Lewis , «Cognitive walkthrough for the web,» Abril 2002. [En línea]. Available: <https://dl.acm.org/doi/pdf/10.1145/503376.503459>.
- [25] Si-Jung Kim, Do-Eun Cho., Si-Jung Kim, Do-Eun Cho., Technology Trends for UX/UI of Smart Contents,, The Korea Contents Association Review, 14 (2016), 29-33, 2016.



- [26] S. Kim, A Study on the UX/UI of Smart Mobile Remote Control App in the Development Trend of NUI: Journal of Korea Design Knowledge, 29(2014), 83-92, 2014.
- [27] A. N. H. G. P. K. S. Niranjnamurthy M, «Research Study on Importance of Usability Testing/ User Experience (UX) Testing, pp. 83-84,» Octubre 2014. [En línea]. Available: https://www.academia.edu/8682851/Research_Study_on_Importance_of_Usability_Testing_User_Experience_UX_Testing.
- [28] J. Yablosnky, Laws of UX. Using Psychology to Design Better Products & Services, 2020.
- [29] Laws of UX, «Jakob's Law,» [En línea]. Available: <https://lawsofux.com/en/jakobs-law/>.
- [30] Laws of UX, «Miller's Law,» [En línea]. Available: <https://lawsofux.com/en/millers-law/>.
- [31] Laws of UX, «Fitts's Law,» [En línea]. Available: <https://lawsofux.com/en/fittss-law/>.
- [32] Laws of UX, «Hick's Law,» [En línea]. Available: <https://lawsofux.com/en/hicks-law/>.
- [33] Anastassia, Inessa, «16 Latest UX/UI Design Trends That You Should Know in 2022,» Abril 2022. [En línea]. Available: <https://www.codica.com/blog/latest-ui-ux-design-trends/>.
- [34] Jenna Romano, Avital Santo, «The State of UX Design in 2022,» 27 Diciembre 2021. [En línea]. Available: https://www.wix.com/blog/2021/12/ux-design-trends/?utm_source=google&utm_medium=cpc&utm_campaign=13708482663^124757113672&experiment_id=^^530755701311^^_DSA&gclid=Cj0KCQjw9ZGYBhCEARIsAEUXITUqPP9D05b0VGnOt0M9Ywv_w2-SYiCXWur-Q7BF0Q523_nZmhcNBcUaAr5YEALw_wcB.
- [35] J. Shapiro, «Web Animation using JavaScript: Develop & Design. Chapter 1,» [En línea]. Available: <https://books.google.es/books?hl=es&lr=&id=4Wo0BwAAQBAJ&oi=fnd&pg=PT13&dq=animation+ux+web&ots=cSoxt6ROul&sig=uRphd1XqZgFf4-hlF1f0bntlu8E#v=onepage&q=animation%20ux%20web&f=false>.
- [36] J. David Eisenberg, Amelia Bellamy-Royds, «SVG Essentials: Producing Scalable Vector Graphics with XML,» Octubre 2014. [En línea]. Available: https://books.google.pt/books?hl=es&lr=&id=FI4PBQAAQBAJ&oi=fnd&pg=PT210&dq=svg&ots=IVDXCjMRz3&sig=NHmkIrkJujw46FHqA0SeQNsXtHo&redir_esc=y#v=onepage&q=svg&f=false.
- [37] Vivus, «Vivus, bringing your SVGs to life,» [En línea]. Available: <https://maxwellito.github.io/vivus/>.
- [38] Snap.svg, «The JavaScript SVG library for the modern web,» [En línea]. Available: <http://snapsvg.io/>.
- [39] AirBNB, «Lottie Files: Lightweight, scalable animations,» [En línea]. Available: <https://lottiefiles.com/>.



- [40] Josh Howarth, «8 Huge UI Design Trends For 2022-2024,» 26 Enero 2022. [En línea]. Available: <https://explodingtopics.com/blog/ui-design-trends> .
- [41] Matthew Wagerfield, Claudio Guglieri, «ParallaxJS,» [En línea]. Available: <https://matthew.wagerfield.com/parallax/>.
- [42] J. Jankowski, «A taskonomy of 3D web use,» Junio 2011. [En línea]. Available: https://dl.acm.org/doi/abs/10.1145/2010425.2010443?casa_token=thGPPdwO4h4AAAAA:Kc5NNrSJuHEQPgZTWrwHIF2qkM3X2iCa1IwoKCKZE5BII_WdzEzjSvmnbfyiONrl6wyVrrWYbQA.
- [43] Diego Cantor, Brandon Jones, «WebGL Beginner's Guide,» Enero 2012. [En línea]. Available: https://books.google.es/books?hl=es&lr=&id=uEmzeCKd8ZEC&oi=fnd&pg=PT15&dq=webgl&ots=rtOK1DvX7c&sig=W5rmfAQu_GMJ6y0ec9obeE8WiYU#v=onepage&q=webgl&f=false.
- [44] Ahmed Kamal, Carlos Andujar, «A platform for developing and fine tuning adaptive 3D navigation techniques for the immersive web,» Noviembre 2021. [En línea]. Available: <https://dl.acm.org/doi/pdf/10.1145/3485444.3487643>.
- [45] ThreeJS, «ThreeJS,» [En línea]. Available: <https://threejs.org/>.
- [46] P. Brunt, «GLGE: WebGL for the lazy,» 2010. [En línea]. Available: <http://www.glge.org/>.
- [47] Daybrush, «Scene.JS,» [En línea]. Available: <https://scene.js.org/>.
- [48] Charles J. Cliffe, «CubicVR 3D Engine,» [En línea]. Available: <https://github.com/cjcliffe/CubicVR.js/>.
- [49] Pedro Gutiérrez, «Introducción a Three.js, la librería 3D número uno para HTML5,» 25 Noviembre 2013. [En línea]. Available: <https://www.genbeta.com/desarrollo/introduccion-a-three-js-la-libreria-3d-numero-uno-para-html5>.
- [50] Spline, «Spline - Design tool for 3D web browser experiences,» 2022. [En línea]. Available: <https://spline.design/>.
- [51] Mapbox, «Maps and location for developers,» [En línea]. Available: <https://www.mapbox.com/>.
- [52] ArcGIS, «ArcGIS API for JavaScript,» Junio 2022. [En línea]. Available: <https://developers.arcgis.com/javascript/latest/>.
- [53] Google Maps Platform, «Crea apps increíbles con el conocimiento de Google del mundo real,» [En línea]. Available: <https://developers.google.com/maps>.
- [54] Invelon Technologies, «Realidad Virtual vs. Realidad Aumentada: ¿en qué se diferencian?,» 24 Enero 2019. [En línea]. Available: <https://invelon.com/diferencias-realidad-virtual-vr-realidad-aumentada->



ar/?gclid=CjwKCAjwu5yYBhAjEiwAKXk_eF-L_3d9IEDnYVbMthDfEWqSAma-
xdQkO6cRddl16C3wWg7ybPNRexoCmMwQAvD_BwE.

- [55] Maximilian Speicher, Brian D. Hall, Michael Nebeling , «What is Mixed Reality?,» Mayo 2019. [En línea]. Available:
https://dl.acm.org/doi/abs/10.1145/3290605.3300767?casa_token=wWUPFW2fJkAAAAA:Az8oiVNsdFTvo10S7WcWRzsubipxahARyzt7a8tL-x3VnD6jKJcAlrHKIUWNFMMSK85JVE0Yj_3R-.
- [56] Runroom, «El presente de la AR y VR es tu propia web,» 20 Mayo 2020. [En línea]. Available:
<https://www.runroom.com/realworld/el-presente-de-la-ar-y-vr-es-tu-propia-web>.
- [57] Bruno Fanini 1,*ORCID,Daniele Ferdani, Emanuel Demetrescu, Simone Berto, Enzo d'Annibale, «ATON: An Open-Source Framework for Creating Immersive, Collaborative and Liquid Web-Apps for Cultural Heritage,» 13 Noviembre 2021. [En línea]. Available: <https://www.mdpi.com/2076-3417/11/22/11062/html>.
- [58] Blotter, «A JavaScript API for drawing unconventional text effects on the web,» [En línea]. Available: <https://blotter.js.org>.
- [59] mattboldt, «Typed: A library that types,» 2021. [En línea]. Available: <https://github.com/mattboldt/typed.js/>.
- [60] Pluto, «We help humanity transcend physical location,» [En línea]. Available: <https://www.pluto.app/>.
- [61] ALEF ESTATE, «Página de inicio,» [En línea]. Available: <https://alef-estate.com/en/>.
- [62] Byung-Seok Kang, Sangheon Pack, Han-Shik Kim, Chul-Hee Kang, «Experimental Study of Voice User Interface (VUI) System Using VoiceXML,» 29-31 Octubre 2007. [En línea]. Available: https://ieeexplore.ieee.org/abstract/document/4438641?casa_token=zOVkMVBRRsoAAAAA:dsmg4A1tFu-Rj652OpV6U4N2WTpqHuVfEYMzkVubSPDezeUJnZtjZITwom8TAqpNfNIVrnPvt7Pi.
- [63] Michael H. Fischer, Giovanni Campagna, Euirim Choi, Monica S. Lam, «DIY assistant: a multi-modal end-user programmable virtual assistant,» 18 junio 2021. [En línea]. Available: <https://par.nsf.gov/biblio/10317967>.
- [64] Ondřej Dostál, «Un Mejor UX a Través de Microinteracciones,» [En línea]. Available: <https://www.toptal.com/designers/product-design/mejor-ux-a-traves-de-microinteracciones>.
- [65] Tom Page, «Skeuomorphism or flat design: future directions in mobile device User Interface (UI) design education,» 2014. [En línea]. Available: <https://flatisbad.com/publications/Page-IntjMobileLearnOrganisat14.pdf>.
- [66] Ahoi Schmerikon, «Website Portfolio for local bar / restaurant,» [En línea]. Available: <https://webflow.com/made-in-webflow/website/Ahoi-Schmerikon>.



- [67] Dribbble, «Pinterest - Skeuomorphic Buttons,» [En línea]. Available: <https://www.pinterest.com/pin/395613148498288423/>.
- [68] Chiyana Simões, «Neuromorfismo: ¿Qué es en diseño de UI?,» 18 Mayo 2021. [En línea]. Available: <https://www.itdo.com/blog/neuromorfismo-que-es-en-diseno-de-ui/>.
- [69] El equipo de Neuf TV, «El neuromorfismo no es un regreso al esqueuomorfismo - diseñador,» 28 Junio 2020. [En línea]. Available: <https://www.neuf.tv/es/le-neuromorphisme-nest-pas-un-retour-au-skeuomorphisme-designer>.
- [70] MockPlus, «An Introduction to Neumorphic Design: Basics, Examples and UI Kits,» 11 Marzo 2022. [En línea]. Available: <https://www.mockplus.com/blog/post/neumorphic-design-examples-ui-kits>.
- [71] Dribbble, «Neumorphism Web Design,» [En línea]. Available: https://dribbble.com/tags/neumorphism_web_design.
- [72] Andrea Bascetta , «'Less is More' - Minimalism in UX Design,» 22 Enero 2020. [En línea]. Available: <https://info.keylimeinteractive.com/less-is-more-minimalism-in-ux-design>.
- [73] Inês Bernardino, «A guide to minimalist design,» 4 Septiembre 2020. [En línea]. Available: <https://uxdesign.cc/a-guide-to-minimalist-design-36da72d52431>.
- [74] Angel Villanueva, «Web Design Minimalist,» [En línea]. Available: <https://dribbble.com/shots/14259885-Web-Design-Minimalist>.
- [75] ГОЛУБ ОЛЬГА ІГОРІВНА, «Glassmorphism in Web Design,» 2021. [En línea]. Available: https://er.knuid.edu.ua/bitstream/123456789/18236/1/ITPF2021_P085-087.pdf.
- [76] Ijelekhai Faith Olohijere , «What is Glassmorphism? How to Create a Glass Card in Figma,» 13 Abril 2022. [En línea]. Available: <https://www.freecodecamp.org/news/glassmorphism-how-to-create-a-glass-card-in-figma/>.
- [77] Adrian Bece, «Claymorphism: Will It Stick Around?,» 16 Marzo 2022. [En línea]. Available: <https://www.smashingmagazine.com/2022/03/claymorphism-css-ui-design-trend/>.
- [78] Diana Malewicz, «2022 UI design trends guide,» 21 12 2021. [En línea]. Available: <https://uxdesign.cc/2022-ui-design-trends-guide-22ddc386557b>.
- [79] Dropdown Design, «5 Top UI/UX Trends Of 2022 With A Bonus Tip,» 9 Mayo 2022. [En línea]. Available: <https://medium.com/@Dropdown.Design/5-top-ui-ux-trends-of-2022-with-a-bonus-tip-88ccf6326f4>.
- [80] Dixon & Moe, «RellaxJS,» [En línea]. Available: <https://dixonandmoe.com/rellax/>.



- [81] InteractJS, «JavaScript drag and drop, resizing, and multi-touch gestures for modern browsers (and also IE9+),» [En línea]. Available: <https://interactjs.io/>.
- [82] Facebook, «React Framework,» [En línea]. Available: <https://es.reactjs.org/>.
- [83] Mapbox, «Mapbox Studio,» [En línea]. Available: <https://www.mapbox.com/mapbox-studio>.
- [84] adamgiebl, «Neumorphism.io,» [En línea]. Available: <https://neumorphism.io/>.
- [85] Hype4 academy, «Claymorphism generator,» [En línea]. Available: <https://hype4.academy/tools/claymorphism-generator>.
- [86] Hype4 academy, «Glassmorphism generator,» [En línea]. Available: <https://hype4.academy/tools/glassmorphism-generator>.
- [87] Lisandra Maioli, *Fixing Bad UX Designs: Master proven approaches, tools, and techniques to make your user experience great again*, 2018.



APÉNDICE A. GLOSARIO



El objetivo de este glosario es facilitar el acceso a una definición de los principales términos que se mencionan a lo largo de este documento. Todos ellos están relacionados con la temática que enmarca la usabilidad y en concreto la experiencia de usuario inmersiva.

A

API. Acrónimo de *Application Programming Interface*, es un conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizada por otro software como una capa de abstracción.

B

Backend. Capa relacionada con el área lógica de toda página web. Arquitectura interna del sitio que asegura que todos elementos desarrollen la función correcta. No está visible a ojos del usuario y no incluye ningún tipo de elemento gráfico.

Biblioteca. Librería. Conjunto de implementaciones funcionales, codificadas en un lenguaje de programación, que ofrece una interfaz bien definida para la funcionalidad que se invoca.

C

Canvas. Lienzo. Elemento que permite la generación de gráficos. Entre otras cosas, permite la renderización interpretada dinámica de gráficos 2D y mapas de bits, así como animaciones con estos gráficos. Se trata de un modelo de procedimiento de bajo nivel, que actualiza un mapa de bits y no tiene una gráfica de escena integrada.

CDN. Conjunto de ubicaciones en el mundo que redistribuyen localmente el contenido de los servidores y guardan en caché los archivos que no necesitan actualización permanente, según unas reglas personalizables.

Click. Acción de pulsar cualquier botón o tecla del dispositivo apuntador (ratón, touchpad o trackball) de la computadora. Como resultado de esta operación, el sistema aplica algún proceso o función al objeto señalado por el cursor o puntero en el momento de realizarla.



CSS. Acrónimo de *Cascade Style Sheets*, es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado.² Es muy usado para establecer el diseño visual de los documentos web, e interfaces de usuario escritas en HTML o XHTML.

D

De iure. Que está reconocida por la legalidad vigente o por la autoridad competente en virtud de algún acuerdo o acto formal

De facto. Aun existiendo en la realidad, que no ha sido reconocida formalmente, ya sea mediante los mecanismos legales vigentes o por la autoridad competente

DOM. Interfaz de plataforma que proporciona un conjunto estándar de objetos para representar documentos HTML, XHTML y XML, un modelo estándar sobre cómo pueden combinarse dichos objetos, y una interfaz estándar para acceder a ellos y manipularlos

Drag & Drop. Expresión informática que se refiere a la acción de mover, con el cursor del ratón, los objetos de una ventana a otra o entre partes de una misma ventana. Los objetos arrastrados son habitualmente archivos, pero también pueden ser arrastrados otros tipos de elementos en función del software.

F

Fade. Tipo de transición audiovisual por la cual la imagen y/o el sonido van ganando (**fade in**) o perdiendo (**fade out**) intensidad hasta quedar únicamente el color de fondo que se haya elegido o el silencio en caso del audio.

Frame. Fotograma. Cada una de las imágenes que componen un vídeo o animación. Es la mínima imagen registrable, normalmente equivalente a 1/25 o 1/30 de segundo, aunque no tiene por qué ser así en todas ocasiones.

Framework. Conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

Frontend. Capa relacionada con el área visual de toda página web. Parte que ve el usuario y con la que interacciona para indicar qué acciones han de ser realizadas en el back.

Full-size. Referenciando a la pantalla del dispositivo, elemento o conjunto de elementos que abarcan el completo de la misma.



H

Hero. Banner grande que aparece en la parte superior de una página web. Suele situarse debajo de la barra de navegación, ocupando la anchura completa de la pantalla (*full-size*). Representa (o es relevante para) la marca y/o el producto de una manera visualmente atractiva.

Hover. Alteración del aspecto de un elemento de la interfaz gráfica cuando se sitúa el puntero sobre el mismo, pero no se ha seleccionado aún.

HTML. Acrónimo de *Hypertext Markup Language*, hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros

I

Input. En la interacción humano-computadora, la entrada es la información producida por el usuario con el propósito del control del programa. El usuario comunica y determina qué clases de entrada aceptarán los programas (por ejemplo, secuencias de control o de texto escritas a máquina a través del teclado y el ratón).

Interacción persona-ordenador. IPO. Denominado también *Interacción humano-computadora* u otro tipo de variantes utilizando sinónimos (entre otros: persona, humano, usuario / ordenador, máquina, computadora), es la disciplina dedicada a diseñar, evaluar e implementar sistemas informáticos interactivos para el uso humano, y a estudiar los fenómenos relacionados más significativos. Es el estudio sobre cómo se diseñan, implementan y usan los sistemas informáticos interactivos y la manera que influyen los ordenadores en los individuos, las organizaciones y la sociedad.

J

JavaScript. (abreviado comúnmente JS) Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos,² basado en prototipos, imperativo, débilmente tipado y dinámico.

JSON. Acrónimo de *Java Script Object Notation*, es un formato de texto sencillo para el intercambio de datos. Se trata de un subconjunto de la notación literal de objetos de JavaScript, aunque, debido a su amplia adopción como alternativa a XML, se considera un formato independiente del lenguaje.



P

Plugin. Aplicación (o programa informático) que permite extender las funciones de otra aplicación o programa sin tener que modificar el código

S

Scroll. Movimiento en 2D de los contenidos que conforman el escenario de un videojuego o la ventana que se muestra en una aplicación informática (por ejemplo, una página web visualizada en un navegador web).

SVG. Acrónimo de *Scalable Vector Graphics*, es un formato de gráficos vectoriales bidimensionales, tanto estáticos como animados, en formato de lenguaje de marcado extensible XML. Los SVG pueden ser interactivos y dinámicos y esto se debe a que no se componen por mapa de bits, sino que están compuestos por vectores, que son instrucciones matemáticas que se le dan al navegador o programas de ediciones de estos gráficos vectoriales, para escalarlos de manera infinita y sin perder resolución o calidad en el gráfico.

T

Target. En el ámbito de la publicidad, destinatario ideal de una determinada campaña, producto o servicio.

Trackpad. Touchpad. Panel táctil que permite controlar un cursor o facilitar la navegación a través de un menú o de cualquier interfaz gráfica.

Trigger. Acto de disparo de un evento justo cuando ocurre determinada situación en uno de los elementos que se están inspeccionando.

U

UI. Interacción de un sistema y un usuario de manera bilateral a través de comandos o técnicas para operar dicho sistema, introducción de datos y utilización de los contenidos que ofrece el sistema

Usabilidad. Calidad de la página web o del programa informático que son sencillos de usar porque facilitan la lectura de los textos, descargan rápidamente la información y presentan funciones y menús sencillos, por lo que el usuario encuentra satisfechas sus consultas y cómodo su uso.



UX. Conjunto de factores y elementos relativos a la interacción del usuario con un entorno o dispositivo concretos, dando como resultado una percepción positiva o negativa de dicho servicio, producto o dispositivo.

W

WebGL. Especificación estándar que define una API implementada en JavaScript para la renderización de gráficos en 3D dentro de cualquier navegador web

Web 3D. Posibilidad de desplazarnos a través del navegador por un espacio tridimensional. Para lograr esto es necesario el uso de WebGL.

Y

YAML. Acrónimo recursivo de *YAML Ain't Markup Language*, es un formato de serialización de datos legible por humanos inspirado en lenguajes como XML, C, Python, Perl, así como en el formato de los correos electrónicos.
