



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

---

**FACULTAD DE INGENIERÍA**

**Realidad aumentada como  
recurso interactivo en  
museos**

**TESIS**

Que para obtener el título de  
**Ingeniero Mecatrónico**

**P R E S E N T A**

Luis Fernando García Ramírez

**DIRECTOR DE TESIS**

M.A. Luis Yair Bautista Blanco



**Ciudad Universitaria, Cd. Mx., 2021**

A MIS DOS PERSONAS MÁS QUERIDAS: MONY Y MARTITA.

LAS EXTRAÑO MUCHO Y ATESORO CADA MOMENTO QUE TUVE CON USTEDES.

GRACIAS POR TANTO AMOR Y SACRIFICIO A LO LARGO DE LOS AÑOS.

GRACIAS POR APOYARME SIEMPRE, SIN USTEDES NO LO HABRÍA LOGRADO.

LAS AMA,

FER



# Agradecimientos

AMOI SEG' MA UNS WIEDER.

*ANDREAS GABALIER*

Primeramente a las dos personas más importantes de mi vida: Mony y Martita, por siempre haber estado conmigo y haberme dado su apoyo incondicional, por haberme motivado a seguir adelante y por haberme forjado como la persona que soy hoy.

De igual manera, agradezco a mi familia más cercana: Alejandro García Hernández, Norma Ramírez, José Kychler, Alicia García y Juan Dimas por haberme apoyado siempre y aún más en éstos tiempos difíciles.

A la UNAM, por haber sido mi segundo hogar y haberme brindado tantas oportunidades y experiencias que cambiaron mi vida: la posibilidad de aprender un tercer idioma, poder participar en concursos nacionales e internacionales de robótica, dos intercambios académicos en Alemania, mucho conocimiento y amistades, desde que ingresé por primera vez a sus aulas en el bachillerato.

A mi director de tesis, M.A. Yair Bautista, por su apoyo, paciencia, dedicación y compromiso para la realización del presente trabajo escrito.

A mis sinodales: Ing. Alejandra Garza, M.F. Gabriel Hurtado, Ing. Jorge Vera y Dra. Pilar Corona por su valiosa retroalimentación sobre el presente trabajo escrito.

A todos los docentes que en algún punto de mi formación académica me han inspirado con su pasión por la enseñanza, en especial al M.C. Edú A. Cruz Lecona.

A mis amigos: Jessica Hernández, Analí De La Cruz, Miguel Ángel García, Rodrigo González, Mariana García, Jorge González, Axel Peña, Laura Lino, Oswaldo Frausto, Karina González y Rubén Paredes, por estar en los momentos más difíciles y brindarme un refugio en el cual puedo olvidar los problemas del día a día.

LEB, ALS OB'S DEIN LETZTER TAG IST.  
LEB, ALS OB DEIN MORGEN NICHT BESTEHT.  
LEB, ALS OB ES BALD VORBEI IST.  
LEB, MACH ALLES WAS DU KANNST.  
*DJ Ötzi*

NIMM DIR DAS LEBEN  
UND LASS ES NICHT MEHR LOS  
DENN ALLES WAS DU HAST  
IST DIESES EINE BLOS  
NIMM DIR DAS LEBEN  
UND GIB'S NIE WIEDER HER  
DENN WENN MAN ES MAL BRAUCHT  
DANN FINDET MAN'S SO SCHWER.  
*UDO LINDENBERG*

ES GEHT NICHT IMMER GERADEAUS  
MANCHMAL GEHT ES AUCH NACH UNTEN  
UND DAS WONACH DU SUCHST  
HAST DU NOCH IMMER NICHT GEFUNDEN  
DIE JAHRE ZIEHEN IM FLUG AN DIR VORBEI  
DIE LAST AUF DEINEN SCHULTERN, SCHWER WIE BLEI  
JEDEN MORGEN STEHST DU AUF  
UND KIPPST DEN KAFFEE RUNTER  
DEINE TRÄUME AUFGEBRAUCHT  
UND DU GLAUBST NICHT MEHR AN WUNDER  
MIT VOLLGAS KNAPP AM GLÜCK VORBEIGERAUSCHT  
WAS DICH RUNTERZIEHT, EY ICH ZIEH DICH WIEDER RAUF  
ICH TRAG' DICH DURCH  
DIE SCHWEREN ZEITEN  
SO WIE EIN SCHATTEN  
WERD' ICH DICH BEGLEITEN  
ICH WERD' DICH BEGLEITEN  
DENN ES IST NIE ZU SPÄT  
UM NOCHMAL DURCHZUSTARTEN  
WO HINTER ALL DEN SCHWARZEN WOLKEN  
WIEDER GUTE ZEITEN WARTEN.  
*UDO LINDENBERG*

THANK YOU FOR THE MUSIC, THE SONGS I'M SINGING  
THANKS FOR ALL THE JOY THEY'RE BRINGING  
WHAT WOULD LIFE BE?  
WITHOUT A SONG OR A DANCE WHAT ARE WE?  
SO I SAY THANK YOU FOR THE MUSIC  
FOR GIVING IT TO ME.  
*ABBA*

# Índice general

<b>I. ANTECEDENTES</b>	<b>I</b>
1.1. Human - Computer Interaction. . . . .	1
1.1.1. User Experience (UX) Design. . . . .	2
1.1.2. Utilidad, Usabilidad y Afinidad. . . . .	4
1.1.3. Principios para apoyar la usabilidad. . . . .	5
1.1.4. Las ocho reglas de oro para el diseño de interfaces. . . . .	12
1.1.5. Modelos en HCI: Ley de Fitts. . . . .	14
1.2. Realidad Aumentada. . . . .	22
1.2.1. Breve historia de la Realidad Aumentada . . . . .	29
1.2.2. Métodos de detección para Realidad Aumentada . . . . .	34
1.2.3. El Metaverso y la Realidad Aumentada . . . . .	38
1.2.4. Tipos de Hardware para Realidad Aumentada . . . . .	40
1.2.5. Buenas prácticas de diseño de interfaces para Realidad Aumentada. . . . .	44
1.3. Aplicaciones de realidad aumentada en museografía. . . . .	50
1.4. El Códice Maya de México. . . . .	63
<b>2. JUSTIFICACIÓN</b>	<b>69</b>
<b>3. PRIMERA ETAPA DE DISEÑO</b>	<b>71</b>
3.1. Identificación de las funciones. . . . .	71
3.2. Interacciones del usuario. . . . .	74
3.3. Generación de modelos virtuales. . . . .	76
3.4. Desarrollo de la UI. . . . .	81
<b>4. PRUEBAS DE USABILIDAD</b>	<b>111</b>
4.1. Prueba de Fitts. . . . .	111
4.2. Prueba de tiempo de carga y percepción del usuario. . . . .	119
<b>5. VERSIÓN PRELIMINAR</b>	<b>127</b>
5.1. Modificaciones al sistema derivadas de las pruebas con usuarios. . . . .	127
5.2. Resumen del funcionamiento del sistema. . . . .	128
5.3. Aplicación de la Escala de Usabilidad del Sistema. . . . .	140
5.3.1. Resultados de la aplicación de la Escala de Usabilidad del Sistema . . . . .	143
<b>6. CONCLUSIONES</b>	<b>147</b>
<b>7. TRABAJO A FUTURO</b>	<b>151</b>
<b>REFERENCES</b>	<b>153</b>

# Índice de figuras

1.	User-centered design. . . . .	3
2.	Product Design & UX. . . . .	3
3.	Utilidad, Usabilidad y Afinidad . . . . .	5
4.	Ejemplo de predictabilidad . . . . .	6
5.	Ejemplo de capacidad de síntesis . . . . .	6
6.	Ejemplos de familiaridad . . . . .	7
7.	Ejemplo de Consistencia . . . . .	7
8.	Ejemplo de Iniciativa de Diálogo . . . . .	8
9.	Ejemplo de Multithreading . . . . .	8
10.	Ejemplo de migrabilidad de tareas . . . . .	9
11.	Ejemplo de capacidad de sustitución . . . . .	9
12.	Ejemplo de personalización . . . . .	10
13.	Ejemplo de observabilidad . . . . .	10
14.	Ejemplo de Capacidad de Recuperación . . . . .	11
15.	Ejemplo de responsiveness . . . . .	11
16.	Ejemplo de conformidad de tareas . . . . .	12
17.	Ley de Fitts como ejemplo de modelo predictivo. . . . .	15
18.	Modelo de tres estados de Buxton como ejemplo de modelo descriptivo. . . . .	15
19.	Experimentos de Fitts . . . . .	17
20.	Ejemplo de la Ley de Fitts . . . . .	17
21.	Ejemplo de las Leyes de Steering y de Fitts . . . . .	18
22.	Conclusiones de la Ley de Fitts . . . . .	20
23.	Conclusiones de la Ley de Fitts . . . . .	20
24.	Conclusiones de la Ley de Fitts . . . . .	21
25.	Conclusiones de la Ley de Fitts . . . . .	21
26.	Conclusiones de la Ley de Fitts . . . . .	21
27.	IKEA Place. . . . .	22
28.	Pokémon GO. . . . .	23
29.	BBC Civilisations AR. . . . .	24
30.	Sephora Virtual Artist. . . . .	24
31.	Visible Body. . . . .	25
32.	The Mandalorian AR Experience. . . . .	25
33.	The Mandalorian AR Experience. . . . .	26
34.	Ejemplo de Realidad Mixta (XR) . . . . .	26
35.	Ejemplo de Realidad Mixta (XR). . . . .	27
36.	Ejemplo de Realidad Virtual (VR) . . . . .	28
37.	Diferencias entre AR, VR y XR. . . . .	29
38.	Pepper's Ghost 1862. . . . .	30

39.	Ilusión Pepper's Ghost en la actualidad. . . . .	30
40.	Sensorama . . . . .	31
41.	Sword of Damocles. . . . .	32
42.	Videoplace. . . . .	32
43.	Boeing HMD. . . . .	33
44.	Virtual Fixtures. . . . .	34
45.	Audio AR. . . . .	35
46.	Marcadores AR. . . . .	36
47.	<i>Tracking</i> de objetos en AR. . . . .	36
48.	Oclusión en AR. . . . .	37
49.	Métodos de detección en AR. . . . .	37
50.	Continuo de la realidad - Virtualidad . . . . .	39
51.	Ejemplo del continuo de la realidad - Virtualidad . . . . .	39
52.	Ejemplos de AR wearable. . . . .	40
53.	Ejemplos de AR wearable. . . . .	41
54.	Ejemplos de AR wearable. . . . .	41
55.	AR Software Development Kits . . . . .	42
56.	Ejemplos de AR móvil. . . . .	42
57.	Ejemplos de AR móvil (no wearable) . . . . .	43
58.	AR en Manufactura. . . . .	44
59.	Compatibilidad de dispositivos. . . . .	44
60.	Interfaz de usuario en AR. . . . .	45
61.	Interfaz de usuario en AR. . . . .	45
62.	AR en el espacio privado y en el espacio público. . . . .	46
63.	Seguridad en la AR. . . . .	46
64.	Seguridad en la AR. . . . .	47
65.	AR Usability Lab. . . . .	47
66.	AR UI Do's and Don'ts. . . . .	48
67.	Oclusión en AR. . . . .	48
68.	AR Onboarding. . . . .	49
69.	Ejemplos de Audioguías. . . . .	51
70.	Diagrama prototipo Benderson. . . . .	52
71.	Ejemplos de audioguías mejoradas. . . . .	53
72.	Ejemplos de audioguías mejoradas. . . . .	53
73.	Ejemplos de audioguías mejoradas. . . . .	54
74.	Ejemplos de AR en museos. . . . .	55
75.	Ejemplos de AR en museos. . . . .	56
76.	Ejemplos de AR en museos. . . . .	57
77.	Ejemplos de AR en museos. . . . .	57
78.	Ejemplos de AR en museos. . . . .	58
79.	Ejemplos de AR en museos. . . . .	58
80.	Ejemplos de AR en museos. . . . .	59
81.	Ejemplos de AR en museos. . . . .	59
82.	Ejemplos de AR en museos. . . . .	60
83.	Ejemplos de AR en museos. . . . .	61
84.	Ejemplos de AR en museos. . . . .	61
85.	Ejemplos de AR en museos. . . . .	62

86.	Ejemplos de AR en museos. . . . .	62
87.	Códice Maya de México P10 . . . . .	63
88.	Códice Maya de México . . . . .	64
89.	Códice Maya de México . . . . .	65
90.	Códice Maya de México . . . . .	66
91.	Códice Maya de México . . . . .	66
92.	Códice Maya de México . . . . .	67
93.	Bosquejo 1 . . . . .	72
94.	Portrait vs. Landscape . . . . .	72
95.	Bosquejo 2 . . . . .	73
96.	Bosquejo 3 . . . . .	75
97.	Bosquejo 4 . . . . .	75
98.	Modelado 3D para el códice virtual . . . . .	76
99.	Modelos 3D del Códice Maya Mexicano . . . . .	77
100.	Pruebas de detección . . . . .	77
101.	<i>Unity's Animation Window</i> . . . . .	78
102.	<i>Unity's Animator</i> . . . . .	79
103.	Impresión 3D de los modelos del códice . . . . .	80
104.	Escaneo de Objetos Vuforia . . . . .	80
105.	Base de datos Vuforia . . . . .	80
106.	Prueba 1 raycast . . . . .	81
107.	Primera versión de la UI . . . . .	82
108.	Segunda versión incompleta de la UI . . . . .	83
109.	Bosquejos de la UI . . . . .	83
110.	Bosquejos de la UI . . . . .	84
111.	Canvas utilizados para la rotación de la UI . . . . .	86
112.	Tercera versión incompleta de la UI . . . . .	87
113.	Botón de encendido/apagado de información . . . . .	88
114.	Fotografías tomadas con la aplicación . . . . .	90
115.	Áreas interactivas del códice . . . . .	91
116.	Áreas interactivas del códice . . . . .	92
117.	Ejemplo del diccionario empleado para almacenar la información . . . . .	92
118.	Secuencia para la apertura de información . . . . .	96
119.	Botón de cerrar el <i>prefab</i> . . . . .	97
120.	Carpeta donde se almacenan las imágenes de la aplicación . . . . .	98
121.	Vista previa de la fotografía . . . . .	100
122.	Funcionamiento del botón compartir . . . . .	100
123.	Iluminación . . . . .	105
124.	Bienvenida Landscape y Portrait . . . . .	105
125.	Prototipo del <i>onboarding</i> . . . . .	106
126.	<i>Onboarding</i> . . . . .	107
127.	Orden de las escenas . . . . .	109
128.	<i>Orden de las escenas</i> . . . . .	109
129.	Prototipo para el experimento de Fitts . . . . .	112
130.	Secuencia del experimento de Fitts . . . . .	113

131.	Experimento de Fitts . . . . .	114
132.	Gráfica y Ecuación obtenida en el experimento de Fitts . . . . .	115
133.	Consideración excluida del experimento de Fitts . . . . .	115
134.	Gráfica y Ecuación de la Escala vs. Índice de Dificultad . . . . .	116
135.	Diagrama de la configuración para el experimento de Fitts . . . . .	117
136.	Fotografías de la configuración para el experimento de Fitts . . . . .	117
137.	Primer experimento para el tiempo de carga óptimo. . . . .	120
138.	Segundo experimento para el tiempo de carga óptimo. . . . .	121
139.	Segundo experimento para el tiempo de carga óptimo. . . . .	122
140.	Gráfica de los resultados del experimento del tiempo de carga óptimo . . . . .	124
141.	Gráfica de los resultados del experimento del tiempo de carga óptimo . . . . .	125
142.	Configuración utilizada para el experimento del tiempo de carga óptimo . . . . .	126
143.	Modificación a los objetivos interactivos . . . . .	128
144.	Modificación al tiempo de carga de los objetivos interactivos . . . . .	128
145.	Diagramas de Funcionamiento . . . . .	130
146.	Diagramas de funcionamiento . . . . .	131
147.	Diagramas de funcionamiento . . . . .	132
148.	Diagramas de funcionamiento . . . . .	133
149.	Diagramas de funcionamiento . . . . .	134
150.	Diagramas de funcionamiento . . . . .	135
151.	Diagramas de funcionamiento . . . . .	136
152.	Diagramas de funcionamiento . . . . .	136
153.	Diagramas de funcionamiento . . . . .	137
154.	Diagramas de funcionamiento . . . . .	138
155.	Diagramas de funcionamiento . . . . .	139
156.	<i>System Usability Scale</i> . . . . .	141
157.	<i>System Usability Scale</i> . . . . .	141
158.	<i>System Usability Scale</i> . . . . .	143
159.	<i>Aplicación de la SUS</i> . . . . .	144
160.	<i>Cálculo del tamaño de muestra</i> . . . . .	145
161.	<i>Resultados de la SUS</i> . . . . .	146

# Índice de Tablas

1.	Tabla de valores posibles para determinar la orientación del dispositivo. . . . .	84
2.	<i>System Usability Scale</i> . . . . .	142

# Índice de Abreviaturas y Acrónimos

<b>HCI</b>	Human - Computer Interaction
<b>ACM</b>	Association of Computing Machinery
<b>ISO</b>	International Organization for Standardization
<b>IxD</b>	Interaction Design
<b>UX</b>	User Experience
<b>UI</b>	User Interface
<b>KLM</b>	Keystroke-level model
<b>GOMS</b>	Goals, Operators, Methods, Selection rules
<b>AR</b>	Realidad Aumentada
<b>VR</b>	Realidad Virtual
<b>HMD</b>	Head Mounted Display
<b>SDK</b>	Software Development Kit
<b>IMU</b>	Inertial Measurement Unit
<b>AI</b>	Artificial Intelligence
<b>API</b>	Application Programming Interface
<b>SLAM</b>	Simultaneous Localization and Mapping
<b>INAH</b>	Instituto Nacional de Antropología e Historia
<b>FPS</b>	First - person shooter
<b>SUS</b>	System Usability Scale
<b>SIGCHI</b>	Special Interest Group on Computer - Human Interaction
<b>CHI</b>	Conference on Human Factors in Computing Systems

*Any sufficiently advanced technology is indistinguishable  
from magic.*

Arthur C. Clarke

# 1

## Antecedentes

### 1.1. HUMAN - COMPUTER INTERACTION.

El término "Interacción Humano - Computadora" (HCI, por sus siglas en inglés: *Human Computer Interaction*) ha sido ampliamente utilizado desde principios de 1980 [1] y se puede definir como un campo de estudio multidisciplinario dedicado al diseño de tecnología computacional y particularmente a la interacción (acciones y diálogos) entre los usuarios y las computadoras[2]. La ACM (*Association of Computing Machinery*) define la HCI como la disciplina encargada del diseño, evaluación e implementación de sistemas informáticos interactivos para uso humano y el estudio de los principales fenómenos que los rodean.[3]

Sin embargo, los orígenes de la HCI se remontan a principios del siglo pasado, con el inicio del estudio del desempeño humano en las fábricas, al realizar tareas manuales y más tarde con el impulso de la Segunda Guerra Mundial para el estudio de

la interacción entre humanos y máquinas, con la finalidad de crear armamento más efectivo. Esto dio origen a la formación de la Sociedad de Investigación de Ergonomía en 1949 y más tarde, con el uso creciente de las computadoras, hubo cada vez más investigadores especializados en estudiar las interacciones entre personas y computadoras.

Algunas de las muchas líneas de investigación y disciplinas en las cuales la HCI se basa y por las cuales es fuertemente influenciada, además de la ergonomía, son: psicología, sociología, tecnología y ciencia de la información, ciencias e ingeniería en computación y diseño de sistemas. [1]

Dentro del contexto de la HCI existen diversos conceptos, principios, reglas y buenas prácticas para el desarrollo de interfaces, algunos de los cuales serán mencionados en las siguientes páginas. Las interfaces se definen como la conexión física y funcional entre dos aparatos, sistemas o dispositivos independientes uno del otro; en este contexto, la comunicación entre un humano y una computadora se lleva a cabo mediante una interfaz.[4]

#### 1.1.1. USER EXPERIENCE (UX) DESIGN.

El término "User Experience" está definido en la ISO 9241 – 210 (*International Organization for Standardization*) como: "A person's perceptions and responses that result from the use and/or anticipated use of a product, system or service." [Las percepciones y respuestas de una persona, resultado del uso y/o uso anticipado de un producto, sistema o servicio][5].

Por otro lado, el término "Interaction Design (IxD)" hace referencia al diseño de productos y servicios interactivos, en el cual el diseñador se concentra más allá del objeto que está siendo diseñado para incluir la manera en la cual el usuario interactuará con éste. El IxD es parte del diseño de la experiencia del usuario (UX Design) y en ocasiones se utilizan ambos términos de manera similar, sin embargo el IxD se centra en el momento en el que el usuario interactúa con el diseño y el UX Design se ocupa no únicamente de ese momento en particular, sino de toda la trayectoria del usuario.

Entonces, el "Diseño de la experiencia del usuario" (*User Experience (UX) Design* en inglés) dentro del ámbito de la HCI hace referencia al proceso de diseño utilizado por equipos de diseño para crear productos que aporten experiencias significativas y relevantes a los usuarios, lo cual engloba todo el proceso de adquisición e integración del producto, incluyendo los aspectos de utilidad, usabilidad, diseño del producto y diseño de la marca. El diseño UX es un tipo de diseño centrado en el usuario y abarca toda la trayectoria del usuario, por lo cual las tareas de un diseñador UX van desde crear protopersonas (*protopersona*: representación detallada de una persona imaginaria que representa al grupo principal de usuarios. [6]), esquemas y prototipos interactivos, así como realizar pruebas de diseño, siempre priorizando las necesidades del usuario en todos los esfuerzos de diseño y desarrollo.[2]

El diseño centrado en el usuario es un proceso iterativo en el cual se tienen distintas etapas, que van desde entender el contexto en el cual un usuario podría usar un sistema, para posteriormente identificar y especificar los requerimientos del usuario, diseñar soluciones y evaluarlas contra los requerimientos y contexto del usuario, para poder iterar en alguno o varios de los pasos anteriores hasta tener una evaluación satisfactoria[7], como se muestra en la Figura 1.



Figura 1: Diagrama de las etapas del diseño centrado en el usuario.[7] (Traducido por el autor)

Por otro lado, el "Diseño de interfaces" (*User Interface / UI Design*) es el proceso utilizado para crear interfaces para desarrollar software o sistemas computarizados [8], entendiendo la interfaz como la conexión o frontera entre dos sistemas independientes, en este caso el usuario y el sistema diseñado. Dicha interfaz es la que permite que el usuario pueda controlar las acciones del sistema y que el usuario reciba retroalimentación para saber cómo seguir actuando.[9]

Tomando en cuenta lo anterior, se puede resumir que el diseño de la interacción (IXD) forma parte del diseño del producto y éste influye en tres características fundamentales para todo producto: utilidad, usabilidad y afinidad. Y por otro lado, el diseño UX es un proceso de diseño centrado en el usuario, el cual engloba todo lo relacionado con el usuario y su experiencia antes, durante y después de utilizar el sistema diseñado, englobando los diseños IxD y UI . Finalmente, los factores que influyen en la experiencia del usuario son: el contexto de uso, el diseño del producto, las habilidades y expectativas del usuario.[10]



Figura 2: A la izquierda se observa un diagrama que resume el hecho de que el IxD forma parte del Diseño del Producto y éste influye en tres características de todo producto. A la derecha se observa un diagrama que explica de manera resumida los factores que intervienen en la experiencia del usuario.[10] (Traducido por el autor)

### 1.1.2. UTILIDAD, USABILIDAD Y AFINIDAD.

Existen tres aspectos fundamentales que todo artefacto (*cualquier producto creado por humanos o cualquier objeto modificado por éstos [11]*) debe tener, los cuales son: utilidad, usabilidad y afinidad.

La utilidad hace referencia a que el artefacto realizará la función para la cual fue creado, es decir que cumplirá con las necesidades del usuario.[12]

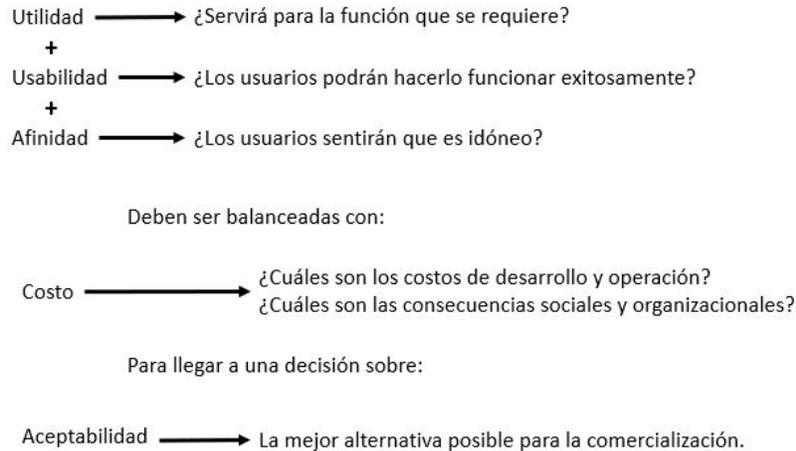
La usabilidad se entiende como el atributo de calidad que evalúa la facilidad de uso de las interfaces de usuario y también hace referencia a los métodos para mejorar la facilidad de uso durante el proceso de diseño. Y está definida por cinco componentes:

- Capacidad de aprendizaje: Hace referencia a qué tan fácil le resulta a los usuarios ejecutar tareas básicas la primera vez que se encuentran con el diseño.
- Eficiencia: Se refiere a cuando los usuarios han ganado experiencia con el diseño, qué tan rápido pueden ejecutar tareas.
- Memorabilidad: Describe cuando los usuarios dejan de usar el diseño por un tiempo y se vuelven a encontrar con éste, qué tan fácilmente pueden recuperar su habilidad para manejarlo.
- Errores: Hace alusión a cuántos errores comete el usuario, su severidad y qué tan fácilmente se puede recuperar de ellos.
- Satisfacción: Se refiere a qué tan placentero resulta usar el diseño.[13]

Y por último, la afinidad se define como un sentimiento subjetivo de idoneidad, el cual conducirá a la satisfacción. Puede tener relación con la utilidad y la usabilidad, pero no necesariamente, pues a los usuarios puede gustarles el artefacto por alguna otra razón.[10]

Los tres aspectos mencionados anteriormente deben ser balanceados con el costo del desarrollo para obtener la aceptabilidad del mismo, como se observa en la Figura 3.

Es sumamente importante la búsqueda de las tres características anteriormente mencionadas en cualquier diseño y en especial de la usabilidad, ya que al tener una interfaz de usuario con buena usabilidad se obtienen múltiples beneficios, dependiendo del artefacto del que se esté hablando, como por ejemplo: el incremento en la productividad de los usuarios, reducción de costos (menor soporte y mayor eficiencia), incremento de ganancias, aumento en la fidelidad de los usuarios y obtención de nuevos usuarios.[10]



**Figura 3:** Resumen de cómo la utilidad, usabilidad y afinidad deben ser balanceadas con el costo para obtener la mejor alternativa.[12] (Traducido por el autor)

### 1.1.3. PRINCIPIOS PARA APOYAR LA USABILIDAD.

De acuerdo con Dix et al.[14], los principios de diseño son las reglas más abstractas que pueden ser aplicadas en el diseño de cualquier sistema interactivo para promover la usabilidad de éste y se subdividen en tres principales categorías:

- Capacidad de aprendizaje: se refiere a la facilidad con la que nuevos usuarios pueden comenzar la interacción con el sistema y alcanzar un desempeño máximo.
- Flexibilidad: tiene que ver con el número de formas en las cuales el usuario y el sistema pueden intercambiar información.
- Robustez: trata acerca de cuánto soporte se le brinda al usuario para determinar si un logro ha sido exitoso y en la evaluación de los objetivos.

Dentro de cada una de las tres categorías anteriormente mencionadas se encuentran diversos principios, los cuales son:

- Capacidad de Aprendizaje.
  - Predictabilidad: Hace referencia al soporte que se le da al usuario para las acciones futuras basándose en las acciones pasadas. Un ejemplo de esto se observa cuando se abren nuevas ventanas de cualquier programa en una computadora, el usuario espera que éstas aparezcan en la barra de tareas en windows y se puede distinguir que las ventanas abiertas son aquellas con una o varias barras debajo del ícono, dependiendo de cuántas instancias se encuentren abiertas.



Figura 4: Ejemplo de predictabilidad.

- Capacidad de síntesis: Se trata del apoyo que se le da al usuario para evaluar el efecto de operaciones pasadas en el estado actual. Un ejemplo de esto se puede observar cuando se mueve un archivo dentro de una carpeta desde la interfaz gráfica de una computadora, pues el archivo inmediatamente desaparece de su ubicación antigua y aparece dentro de su nueva ubicación. Otro ejemplo de este principio se aprecia cuando se toma una fotografía con el celular y hay un efecto de pausa o parpadeo e incluso algún sonido que indica que la fotografía ha sido tomada.

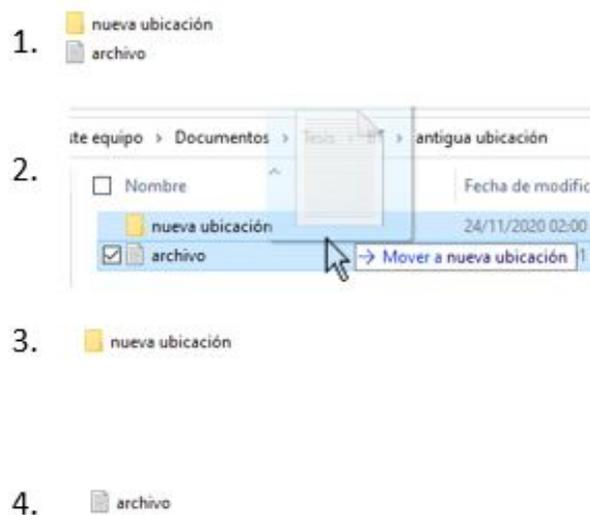


Figura 5: Ejemplo de capacidad de síntesis.

- Familiaridad: Es el grado en que el conocimiento y experiencia del usuario acerca del mundo real o en algún otro sistema computarizado pueden ser aplicados para interactuar con un nuevo sistema. Este principio tiene que ver mucho con lo que se conoce como "metáforas" utilizadas en el diseño de interfaces. Existen numerosos ejemplos de ellas y pueden ir desde un ícono hasta un modo de interacción al que los usuarios estén acostumbrados. Algunos de los ejemplos más comunes pueden ser: el ícono de la papelera de reciclaje que asemeja un bote de basura, los íconos de cámara en los smartphones, algunos lectores de documentos que simulan el pasar una página de libro como si se tratara de un libro de papel, el mismo tipo de interacciones (comandos de voz), etc.



Figura 6: Ejemplos de familiaridad.[15][16]

- Generalizabilidad: Este principio es similar al de familiaridad, pero la diferencia recae en que en éste se amplía el conocimiento que se tenga de la interacción específica dentro y entre aplicaciones a situaciones nuevas pero similares. Un ejemplo de este principio pueden ser los comandos de copiar/pegar/cortar que funcionan en la mayoría de los programas que son utilizados en la actualidad o el cómo los íconos que realizan las mismas funciones son iguales o muy similares entre diferentes aplicaciones.
- Consistencia: Este principio se refiere a la similitud que existe en la interacción entrada - salida al estar en situaciones similares o al querer realizar acciones parecidas. Dos ejemplos muy claros de este principio pueden observarse en la posición y los colores que normalmente se usan para algo positivo o para alguna afirmación (verde), mientras que para alguna negación o alguno negativo normalmente se utiliza rojo, sin embargo en algunas ocasiones es necesario romper la consistencia, de modo que se eviten errores por parte del usuario.

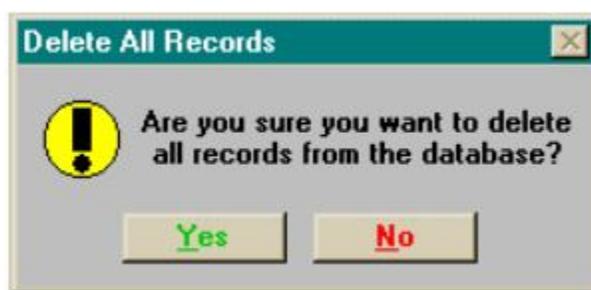


Figura 7: Ejemplo de Consistencia. Cuadro de diálogo diciendo: *¿Estás seguro de querer eliminar todos los registros de la base de datos?*. Se observa que la afirmación es de color verde y la negación en color rojo. En este caso lo ideal sería romper la consistencia para asegurarse de que el usuario realmente quiere eliminar los datos.[10]

#### ■ Flexibilidad

- Iniciativa de diálogo: Este principio hace referencia a dos situaciones cuando se está trabajando con algún sistema.
  1. Diálogo preventivo del sistema: cuando el sistema controla el flujo de diálogo, por ejemplo para prevenir algún error o confirmar la acción requerida por el usuario.

2. Dialogo prioritario para el usuario: cuando el usuario es quien tiene la libertad de iniciar el diálogo, un ejemplo de esto podría ser cuando el usuario hace clic derecho en alguna ventana y se abre un menú de opciones.

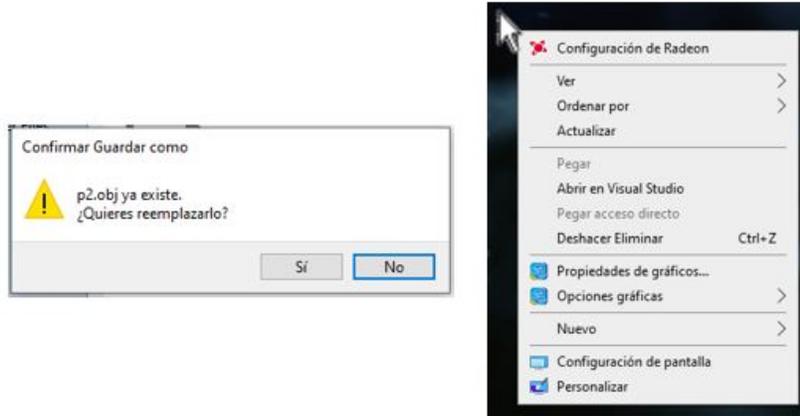


Figura 8: A la izquierda se observa un ejemplo de un diálogo preventivo del sistema y a la derecha se muestra un ejemplo de un diálogo prioritario para el usuario.

- Multithreading: Este principio trata acerca de la capacidad de algún sistema para soportar diversas tareas al mismo tiempo; esta característica es benéfica en muchos casos, por ejemplo al estar trabajando en una computadora y poder dejar tareas en segundo plano, como las notificaciones del calendario o del correo electrónico, sin embargo se puede dar el caso de que el usuario pierda el flujo de trabajo.

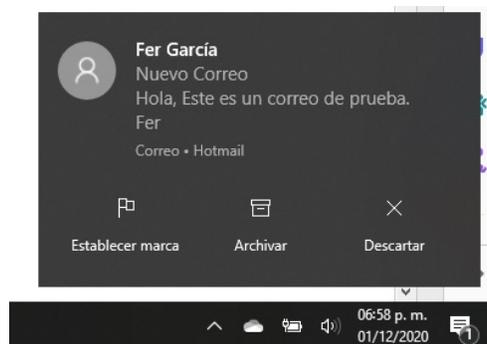


Figura 9: Notificación de correo electrónico entrante, como ejemplo de multithreading.

- Migrabilidad de Tareas: Este principio trata acerca de la posibilidad de pasar el control de alguna tarea entre el sistema y el usuario. En sistemas donde se requiere un alto grado de seguridad, esta característica reduce la posibilidad de que sucedan accidentes, como puede ser en el sistema de piloto automático de un avión. Otro ejemplo de este principio se puede observar en la función de corrector ortográfico de algún editor de texto.

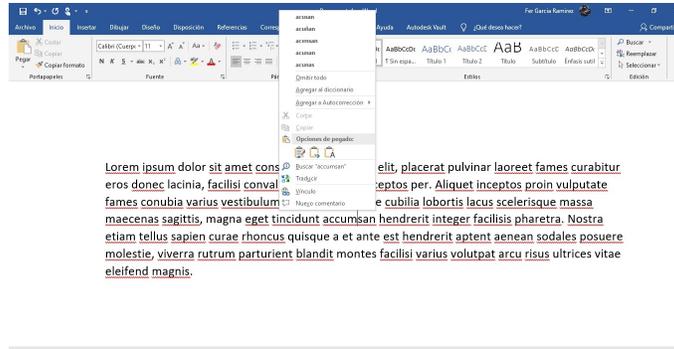


Figura 10: Función de corrector ortográfico de Microsoft Word como ejemplo del principio de migrabilidad de tareas.

- Capacidad de sustitución: Este principio habla acerca de la posibilidad que tienen los sistemas para que el usuario tenga la libertad de sustituir valores equivalentes y así poder reducir el esfuerzo cognitivo por parte del usuario. Este principio se puede ver reflejado en tres escenarios:
  1. Cuando en cualquier sistema se le permite al usuario cambiar algún valor numérico por la operación que da lugar a éste, como por ejemplo en un software de CAD cuando existe una medida, proveniente de realizar algún cálculo y se le permite al usuario ingresar dicho cálculo en vez de la cifra ya calculada.
  2. Cuando se le permite al usuario visualizar información de diferentes maneras, por ejemplo: si en algún sistema es necesario monitorear tendencias de temperatura se puede observar de manera gráfica, pero si se necesita el valor exacto se puede observar numéricamente.
  3. Cuando el sistema permite ingresar datos al sistema y éste realiza las operaciones inmediatamente, de modo que se observa inmediatamente el resultado.

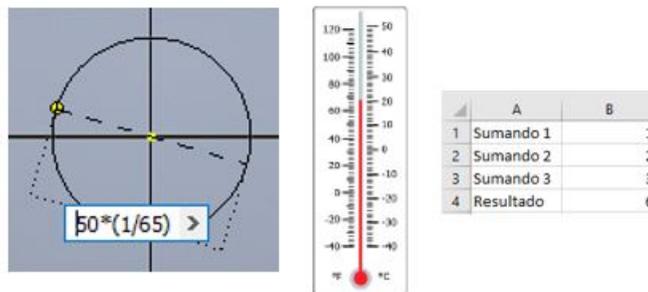


Figura 11: De izquierda a derecha: posibilidad de ingresar valores equivalentes, capacidad de visualización gráfica y capacidad de visualización del resultado inmediatamente, como ejemplos de capacidad de sustitución.

- Personalización: Este principio hace referencia a la capacidad de modificación de la interfaz de usuario por parte del usuario o del sistema. Si la modificación proviene del usuario, se está hablando de *adaptability* y si la modificación proviene del sistema, se habla de *adaptivity*; en español no existe una diferencia entre ambos términos, pues ambos significan adaptabilidad.)

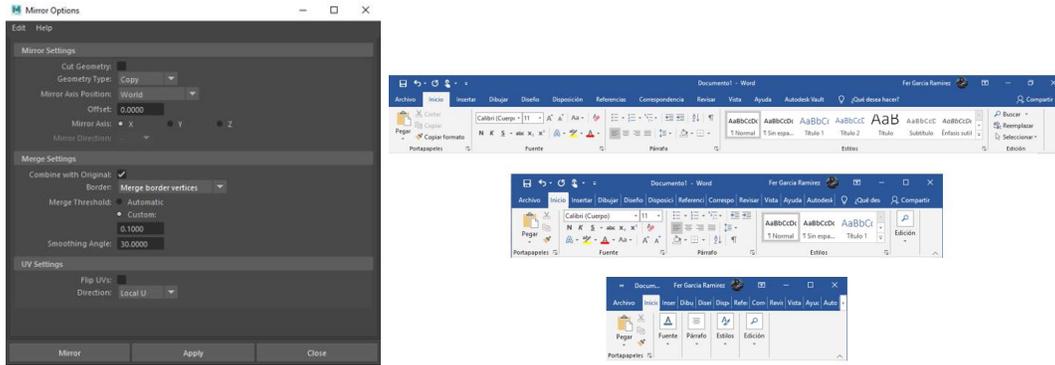


Figura 12: A la izquierda se muestra un menú que permite personalizar un comando en un software de modelado, como ejemplo de *adaptability* y a la derecha se muestra cómo la interfaz de usuario de Microsoft Word se adapta automáticamente al tamaño, como ejemplo de *adaptivity*.

## ■ Robustez

- Observabilidad: Este principio habla acerca de que el sistema debe permitir al usuario evaluar el estado interno del mismo mediante su representación en la interfaz. Un ejemplo bastante común de este principio se puede encontrar en las interfaces de smartphones y computadoras, en las barras que muestran el porcentaje de carga de la batería, estado de la conexión, alarma, etc.

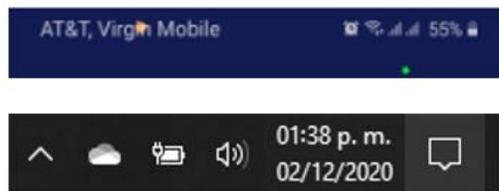


Figura 13: Ejemplo del principio de observabilidad, muy comúnmente utilizado en smartphones y computadoras.

- Capacidad de Recuperación: Este principio se refiere a la habilidad de que el usuario logre un objetivo después de haber cometido algún error en el flujo de trabajo. Dentro de este principio se encuentran dos tipos de interacciones:
  1. Recuperación de errores hacia adelante: En ésta el sistema acepta el error y ayuda al usuario a corregirlo y

a alcanzar su objetivo.

2. Recuperación de errores hacia atrás: En ésta el usuario puede volver uno o varios pasos atrás para evitar el error. Un ejemplo de esta característica se puede observar en la mayoría de las interfaces de cualquier programa, pues todos permiten hasta cierto punto la reversibilidad de acciones para regresar a algún punto en el flujo de trabajo pasado, ya sea en algún editor de texto, software de diseño, ambientes de desarrollo integrados (IDE), sistemas de control de versiones, etc.

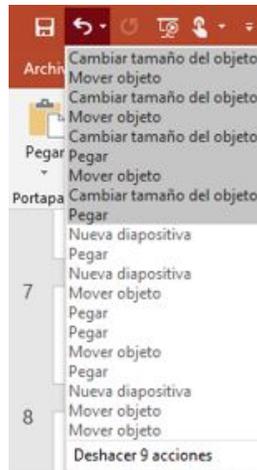


Figura 14: Botón "deshacer" como un ejemplo del principio de capacidad de recuperación.

- Responsiveness: Este principio mide la tasa de comunicación entre el sistema y el usuario. Generalmente se prefiere que el tiempo de respuesta del sistema sea inmediato o lo más breve posible. Cuando un cambio en la interfaz no es inmediato, como en el caso de algún programa de hoja de cálculo, se opta por poner barras que muestran el progreso de la operación, para que el usuario visualice que la tarea sigue en curso.

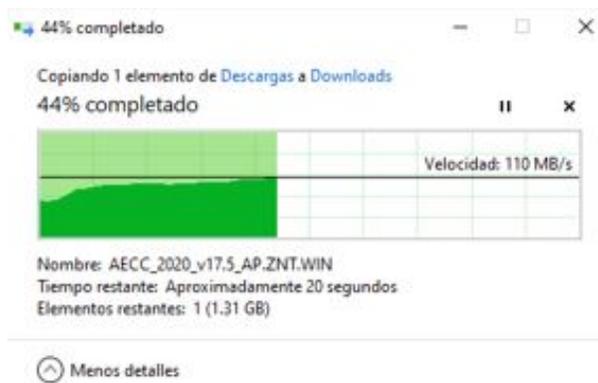
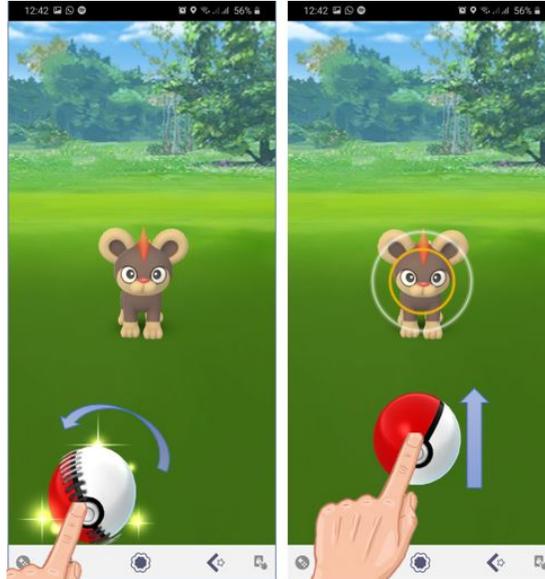


Figura 15: Se observa que el sistema está informando el porcentaje de la acción ejecutada, tiempo estimado e incluso una gráfica con la velocidad de transferencia, como ejemplo de responsiveness.

- Conformidad de Tareas: Este principio se enfoca en dos objetivos; el primero de ellos es que el sistema soporte las interacciones que el usuario quiere ejecutar, y el segundo se refiere a que dichas interacciones sean de la manera en que el usuario las quiera ejecutar.



**Figura 16:** Se muestra como una aplicación soporta dos tipos de interacciones para la misma tarea: lanzar la bola hacia el frente y lanzarla con efecto de giro, como ejemplo del principio de conformidad de tareas.[17][18]

#### 1.1.4. LAS OCHO REGLAS DE ORO PARA EL DISEÑO DE INTERFACES.

Ben Schneidermann, uno de los pioneros de la HCI e inventor del hipervínculo, propuso tres principios para el diseño de interfaces en la primera edición de su libro *”Designing the user Interface”* (1986), los cuales son[19]:

- Conocer al usuario: Este principio suena muy simple de primera instancia, pero para lograr un diseño con esto en mente se tienen que tener en cuenta los procesos del diseño de la experiencia del usuario mencionados anteriormente, como son: creación de protopersonas, definir perfiles de usuarios y crear perfiles de tareas, en los cuales se establecen las interacciones o acciones deseadas antes de comenzar el diseño, con base en lo que el usuario quiere llevar a cabo.
- Seguir las ocho reglas de oro para el diseño de interfaces: Las siguientes reglas son aplicables a prácticamente cualquier sistema interactivo y sirven de guía para mejorar la usabilidad de dichos sistemas.
  1. Esforzarse en la consistencia: Esta es la regla que más frecuentemente se debe romper para evitar alguna acción indeseada por parte del usuario, como se mencionó anteriormente. Dentro de esta regla se distinguen tres

categorías en las cuales hay que tener consistencia.

- Léxica: algunos ejemplos de consistencia en esta categoría son utilizar la misma codificación según el uso (verde = bueno o afirmación, rojo = malo o negación, izquierda = menos, derecha = más, etc.), utilizar la misma terminología en los diálogos, menús y pantallas de ayuda, usar reglas de abreviación consistentes, usar una tipografía consistente, etc.
  - Sintáctica: Esta categoría se puede ilustrar al tomar como ejemplo las dos siguientes situaciones: los objetos dentro de los menús deben estar siempre en el mismo lugar (facilita al usuario el usar los comandos rápidamente debido a la memoria muscular), los mensajes de error deben colocarse siempre en el mismo lugar.
  - Semántica: Dentro de esta categoría se encuentra el que los comandos globales siempre estén disponibles (ayuda, deshacer, cancelar, etc.), también que las operaciones sean válidas en todos los objetos razonables y que exista aplicabilidad de interacciones (gestos, menús, comandos, barras de texto, etc.)
2. Permitir a los usuarios el uso de atajos: Esta regla permite incrementar la velocidad de usuarios experimentados.
  3. Brindar retroalimentación informativa: Para cualquier acción ejecutada, el usuario debe recibir una retroalimentación apropiada e informativa.
  4. Diseñar diálogos para provocar cerradura: Las secuencias de acciones deben tener un inicio, intermedio y final.
  5. Prevenir errores: Se debe crear una UI en la que sea difícil cometer errores. Algunos ejemplos para lograr esto pueden ser: sustituir comandos por menús, utilizar opciones en vez de entradas de texto, etc.
  6. Permitir que las acciones sean fáciles de revertir: Todas las acciones deben ser reversibles, aunque por supuesto esto no siempre es posible, como por ejemplo no es posible deshacer el envío de un correo electrónico.
  7. Apoyar el locus de control interno, el cual se define como la percepción del sujeto acerca de que los eventos ocurren principalmente como efecto de sus propias acciones, es decir la percepción que él es quien controla su vida[20], pero en este caso se habla de que el usuario debe sentir que tiene el control del sistema o interfaz.
  8. Reducir la carga en la memoria de corto plazo: El sistema es el que debe guardar o recordar la información, no el usuario. Un ejemplo de esto se observa en las aplicaciones de banca electrónica, en las cuales el usuario re-

gistra una vez las cuentas a las que hace transferencias y posteriormente el sistema recuerda automáticamente todos los datos.

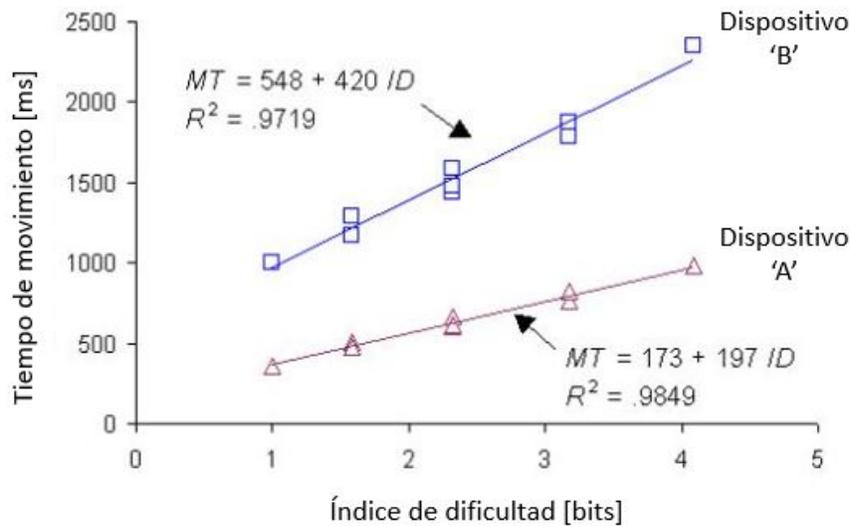
- Prevenir errores: Este punto es tan importante que se menciona dentro de las ocho reglas de oro y como principio individual. Dentro de éste se encuentran diversas sugerencias, por ejemplo: guiar a los usuarios mediante asistentes (wizards), guías paso a paso, autocompletado de comandos y mejorar los mensajes de error dentro de la interfaz para reducir la posibilidad de errores futuros.

Como se pudo apreciar en las últimas dos secciones (1.1.3 y 1.1.4), todas las reglas y principios presentados son una guía sólida para la creación de experiencias de usuario e interfaces humano - máquina; y como se pudo observar, muchas de ellas están tan interrelacionadas unas con otras, que en ocasiones se desvanece la frontera entre una y otra, pues varias de éstas reglas y principios se aplican a una sola característica del artefacto que está siendo diseñado.

#### 1.1.5. MODELOS EN HCI: LEY DE FITTS.

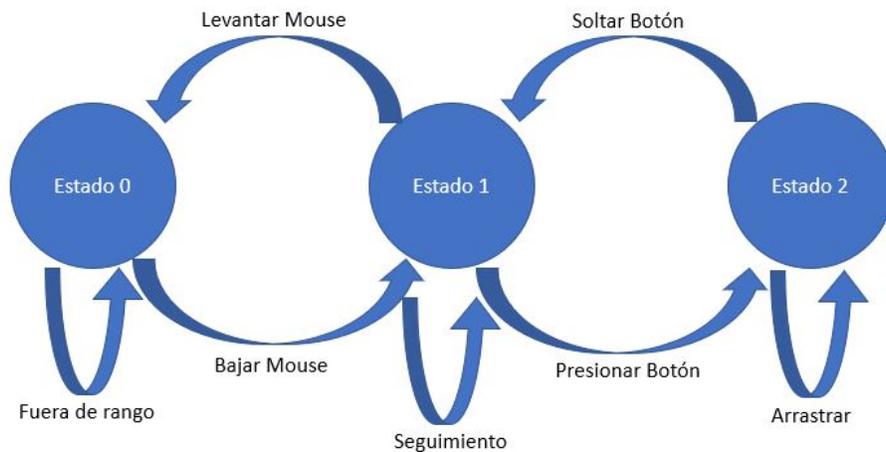
Los modelos son por definición una simplificación de la realidad; en el ámbito de la HCI son utilizados para guiar la creación de las interfaces humano - computadora y se clasifican en dos grandes grupos: modelos descriptivos y modelos predictivos.[21]

Los modelos predictivos, también llamados modelos de ingeniería o modelos de desempeño son ampliamente utilizados en muchas disciplinas. En el ámbito de la HCI, éstos permiten cuantificar analíticamente métricas del desempeño humano sin realizar experimentos que consuman mucho tiempo y recursos. Las predicciones generadas por estos modelos permiten explorar hipotéticamente escenarios de diseño sin tener que implementar sistemas reales completos y recopilar las mismas métricas de rendimiento a través de la observación directa de usuarios reales. Algunos de los modelos predictivos, son por ejemplo: Keystroke-level model (KLM), Ley de Fitts, Ley de Steering y Ley de Hick.



**Figura 17:** Gráfica de la Ley de Fitts comparando dos dispositivos de interacción como ejemplo de un modelo predictivo. En el eje Y se muestra el tiempo de movimiento y en el eje X el índice de dificultad.[21]Imagen traducida por el autor.

Los modelos descriptivos, a diferencia de los predictivos no proporcionan una medida cuantitativa del desempeño del usuario, pero no por eso resultan de menor utilidad. Los modelos descriptivos proporcionan un marco o contexto para describir un problema o situación. A menudo, el marco es poco más que una articulación verbal o gráfica de categorías o características identificables en una interfaz. Sin embargo, son una excelente herramienta para estudiar y pensar sobre la experiencia de interacción del usuario. Algunos de los modelos descriptivos, son por ejemplo: *Key-action model*, Modelo de tres estados de Buxton, Modelo GOMS (*Goals, Operators, Methods, Selection rules*), etc.



**Figura 18:** Modelo de tres estados de Buxton para la utilización de un mouse como método de interacción como ejemplo de modelo descriptivo.[21]Imagen traducida por el autor.

La Ley de Fitts fue presentada en 1954 por el psicólogo Paul Fitts, como una analogía basada en el Teorema de Capacidad del Canal de Shannon. Fitts examinó el sistema motor humano y logró demostrar que el tiempo necesario para moverse hasta un objetivo es una función de la razón entre la distancia hasta dicho objetivo y el ancho de éste y con eso se logran explicar las características de compensación de velocidad-precisión del movimiento muscular humano.[22]

El Teorema de Shannon establece que todo canal de comunicaciones, se verá afectado por el ruido electromagnético y de esta manera se limita la capacidad de un canal para transmitir información.

Resulta natural pensar que a medida que aumenta el ruido (N) de un canal, disminuye su capacidad de transmisión, ya que dicho ruido estará usando el espacio del canal que corresponde a la información.

Una forma de ponderar esta circunstancia es cuantificar la relación S/N, es decir la relación señal (S) a ruido (N). La ecuación que los relaciona, corresponde a la primera ley de Shannon:

$$C = B \log_2 \left( 1 + \frac{S}{N} \right) \quad (\text{ec. 1})$$

Dónde: C es la capacidad máxima del canal en [bps].

B es el ancho de banda del canal medido en [Hz].

S/N representa la relación señal - ruido.[23]

Actualmente, con la llegada y auge de las interfaces gráficas de usuario (GUI, por sus siglas en inglés) y el rápido aumento de nuevos métodos de interacción, la ley de Fitts ha sido utilizada para el diseño de tareas en las que el usuario debe señalar y mover o arrastrar la posición del cursor sobre un objetivo de la pantalla, como un botón o algún otro elemento. Es por ello que la Ley de Fitts ha cobrado gran importancia hasta el día de hoy.[24]

En 1954, Fitts creó un experimento con el cual comenzó a sustentar su teoría. En dicho experimento los usuarios debían tocar alternativamente dos objetivos de ancho (W), separados por la distancia (A) de manera secuencial y posteriormente, en 1964, presentó un segundo experimento, en el cual los usuarios debían tocar uno de los dos objetivos a partir de la recepción de un estímulo visual (luz), como se puede observar en la Figura 19.

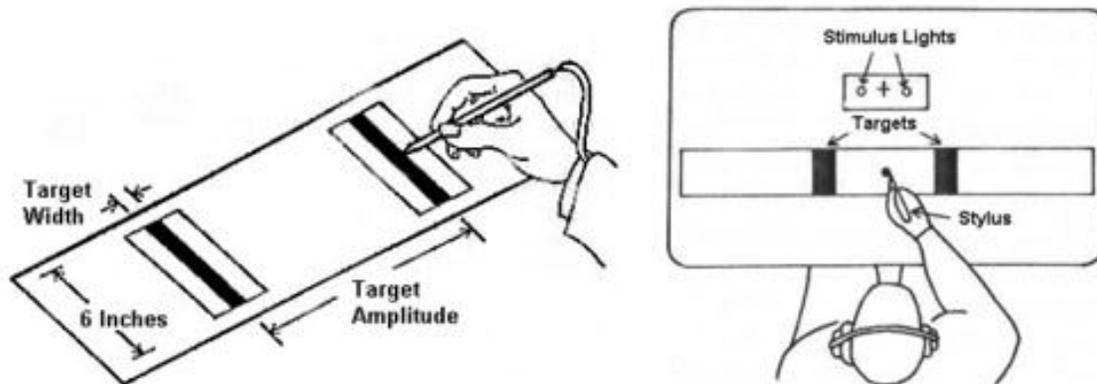


Figura 19: A la izquierda se observa el primer experimento de Fitts (1954) y a la derecha se muestra el segundo experimento de Fitts (1964).[21]

La ley de Fitts ha sido aplicada con una gran variedad de condiciones y dispositivos, utilizando miembros diferentes (manos, pies, dispositivos montados en la cabeza, ojos, etc.), dispositivos de entrada (mouse, trackpad, pantallas táctiles, pedales, etc.), entornos físicos (incluso bajo el agua) y poblaciones (jóvenes, ancianos y personas con discapacidades). Los valores que cambian ante dichos escenarios son las constantes  $a$ ,  $b$ , el índice de dificultad y por lo tanto el tiempo de movimiento, como se observa en la Figura 20. [25] [26] [27] [28]

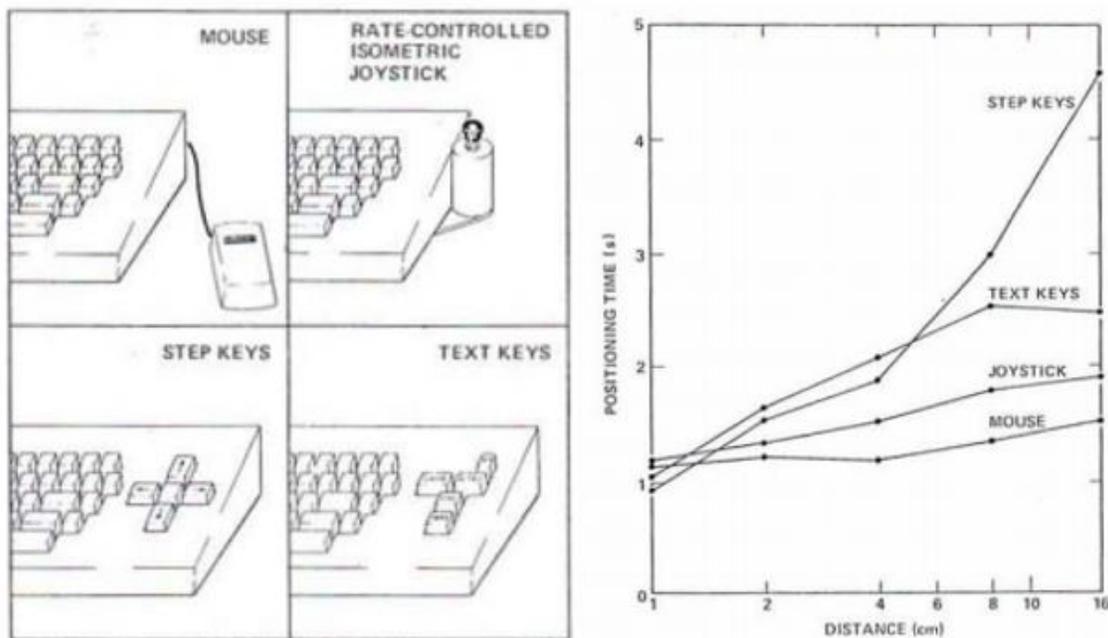


Figura 20: A la izquierda se observan cuatro dispositivos de entrada que son comparados entre sí mediante un análisis de Fitts y a la derecha el resultado de dicho análisis. Esta investigación se llevó a cabo por Xerox Palo Alto Research y fue lo que impulsó a Xerox a seguir desarrollando el mouse (inventado en 1967 por Douglas Engelbart) a finales de la década de los 70.[29]

La ley de Fitts puede modelar las acciones *point-and-click* (señalar y pinchar) y *drag-and-drop* (arrastrar y soltar), pero a pesar del atractivo del modelo, se deben tener en cuenta las siguientes consideraciones para poder utilizarla: [21]

- Es útil para describir movimientos sin entrenamiento.
- Es útil para describir respuestas motoras simples, por ejemplo el movimiento de la mano, dejando de lado la aceleración del software que suele estar aplicada al cursor.
- Aplica únicamente a movimientos en una dimensión y no de dos dimensiones, para éste último se realizó una adaptación que toma el nombre de "Ley de Steering", lo cual se ejemplifica en la Figura 21.

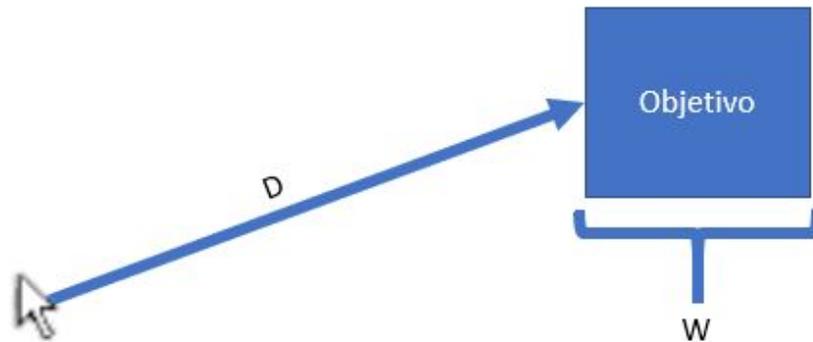
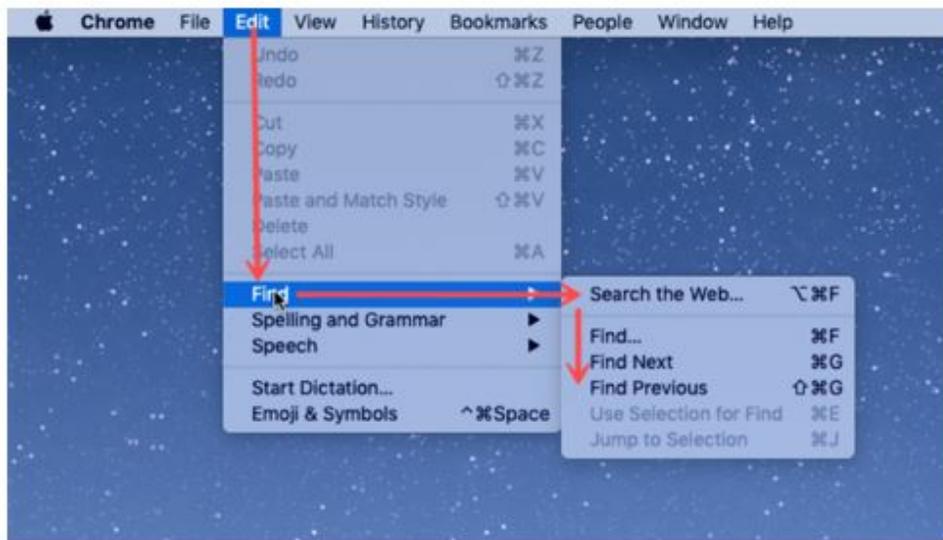


Figura 21: En la figura superior se muestra un ejemplo de los movimientos sujetos de análisis por parte de la Ley de Steering y en la figura inferior se observa un ejemplo de los movimientos sujetos de análisis por parte de la Ley de Fitts. [30]

De esta manera, la Ley de Fitts es capaz de predecir el tiempo requerido para desplazarse rápidamente hasta un objetivo utilizando algún dispositivo de entrada como medio de interacción.

Fitts definió el índice de dificultad (ID) como el logaritmo binario del cociente entre dos veces la distancia que hay entre el objetivo y el punto inicial y el ancho del objetivo.

$$ID = \log_2\left(\frac{2A}{W}\right) \quad (\text{ec. 2})$$

Donde: A es la distancia desde el punto inicial hasta el objetivo.

W es el ancho del objetivo.

Entonces la Ley de Fitts queda representada al final como:

$$MT = a + b * ID \quad (\text{ec. 3})$$

Donde: MT es el tiempo del movimiento.

a y b son constantes determinadas experimentalmente, varían en cada dispositivo.

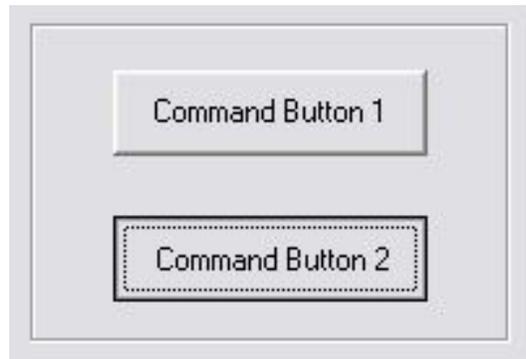
ID es el índice de dificultad.

Al construir un modelo utilizando la Ley de Fitts, los coeficientes de pendiente (a) y ordenada al origen (b) en la ecuación son determinados mediante pruebas en las que se lleva a cabo un experimento controlado utilizando un grupo de sujetos y uno o más dispositivos de entrada y una o varias tareas a realizar. En dichas pruebas son de interés los datos de tiempo, distancia y tamaño de los objetivos y posteriormente se realiza una regresión lineal para caracterizar el dispositivo de interacción o comprobar o rechazar alguna hipótesis.

El diseño de experimentos utilizando la Ley de Fitts es bastante sencillo: las tareas son diseñadas para cubrir un rango de dificultades variando A y W, de modo que para cada condición de tarea, se realizan múltiples pruebas y el tiempo de ejecución de cada una es almacenado para ser analizado posteriormente y así poder obtener las constantes a y b. De este modo queda caracterizada la interacción o el dispositivo de entrada, como se ejemplifica en la Figura 26.[31]

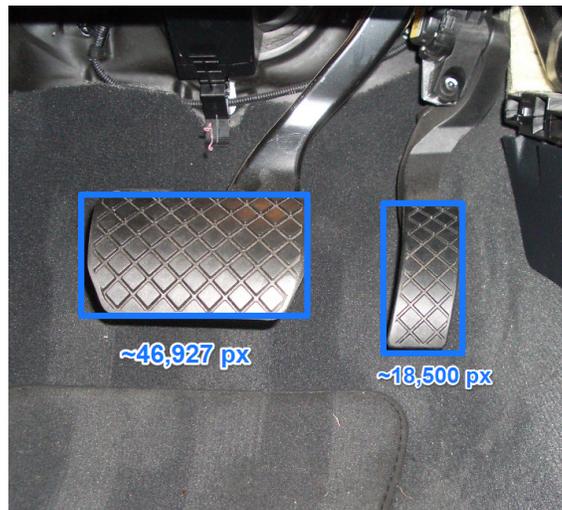
Algunas conclusiones obtenidas de la aplicación de la Ley de Fitts, que ahora se han convertido en características de muchas interfaces, las cuales ya se dan por sentadas o pasan desapercibidas son:

- Los botones o elementos interactivos deberán ser fácilmente distinguibles de aquellos con los que no se puede interactuar, como se muestra en la Figura 22.



**Figura 22:** Se muestra un ejemplo de cómo se diferencian los elementos interactivos comúnmente.[24]

- El tamaño de los botones o elementos interactivos debe ser proporcional al uso que se le dará para apoyar el principio de consistencia.



**Figura 23:** En la figura se observa como ejemplo el tamaño de los pedales de un auto; el pedal de frenado es considerablemente más grande debido a que en caso de emergencia debe ser alcanzado más rápidamente.[32]

- Las esquinas y bordes son especialmente fáciles de alcanzar, debido a que el cursor llega únicamente hasta el borde de la pantalla independientemente de cuánto más se mueva el mouse, por lo que se considera que tienen ancho infinito, como se muestra en la Figura 24.



Figura 24: En la figura se muestra un ejemplo de que los bordes son particularmente fáciles de alcanzar.[24]

- Los menús radiales son más fáciles de seleccionar para los usuarios, debido a que hay menor desplazamiento y todos los elementos suelen estar posicionados a la misma distancia del centro.

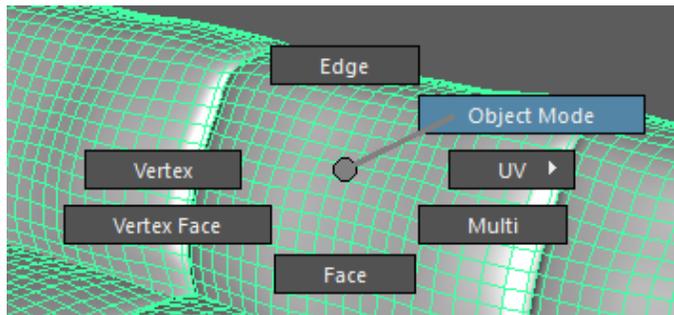


Figura 25: Menú radial que permite ahorrar tiempo y esfuerzo al usuario, como ejemplo de la aplicación de la Ley de Fitts.[33]

- Los menús de tipo pop-up son más rápidos de utilizar por los usuarios debido a que hay menor desplazamiento.

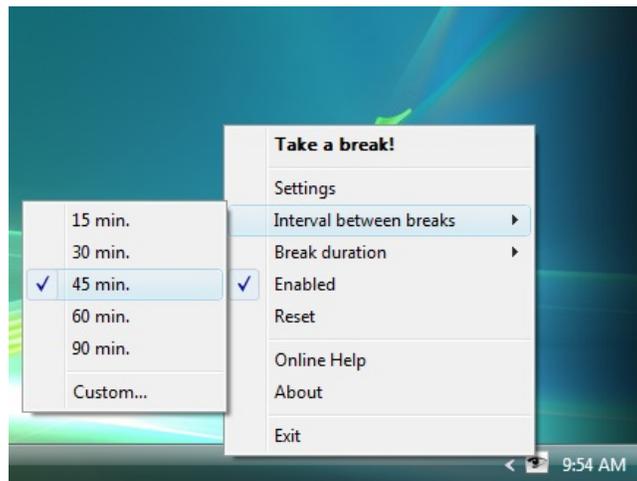


Figura 26: Menú de tipo pop-up, como ejemplo de la aplicación de la Ley de Fitts.[24]

## 1.2. REALIDAD AUMENTADA.

Actualmente, la Realidad Aumentada (AR, por sus siglas en inglés) se define como el conjunto de tecnologías que permiten la visualización directa o indirecta en tiempo real del mundo real, el cual ha sido mejorado o aumentado al añadir capas de elementos virtuales generados por computadora, esta idea de superponer capas de información al mundo real fue llamada "Artificial Reality" por Myron Krueger en 1991 [34]. La RA es entonces la combinación interactiva de elementos virtuales y reales. [35]

Por mencionar sólo algunos ejemplos de aplicaciones AR en la actualidad podemos encontrar: *IKEA Place*, *Pokémon GO*, *BBC Cultures AR*, *Sephora Virtual Artist*, *Visible Body* y *The Mandalorian AR Experience*.

*IKEA Place* es una aplicación AR lanzada en 2019 destinada al sector de venta de muebles, en la cual el usuario puede colocar los muebles virtuales en su casa o espacio donde los requiera y de esta manera se puede dar una idea mucho más real de cómo lucirá el mueble, así como si éste tiene las dimensiones adecuadas por si el usuario se decide a adquirirlo, como se muestra en la Figura 27. [36]

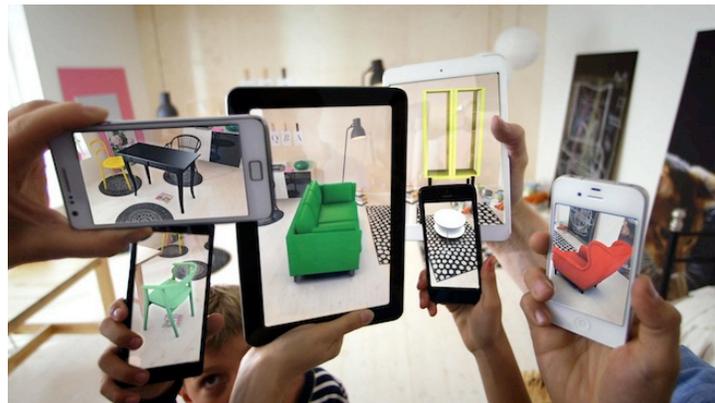


Figura 27: Varios dispositivos utilizando la aplicación *IKEA Place* en la misma habitación. [37]

Por otro lado, *Pokémon GO* es una aplicación de AR lanzada en 2016 muy popular y a partir de la cual ha habido un gran interés y crecimiento de las experiencias AR, esta aplicación utiliza el sistema de posicionamiento global (GPS, por sus siglas en inglés) del dispositivo en el que se ejecuta, de tal modo que el usuario puede ir caminando por el mundo real mientras captura personajes ficticios llamados Pokémon y al acudir a lugares populares, como un monumento por ejemplo, obtener objetos del juego al girar marcadores virtuales ubicados en dichos lugares. Con el paso del tiempo se ha ido incrementando las funciones y ahora es posible tener batallas Pokémon entre jugadores, tomar fotografías con ellos, e incluso se organizan eventos mundiales (*Community Day*) en los cuales es más fácil conseguir un Pokémon determinado

número de horas y otros en ciertas partes del mundo que duran 3 o 4 días (*GO Fest*) o (*Safari Zone*), en los cuales se reúnen jugadores de muchas partes del mundo para capturar, intercambiar, batallar e interactuar con los personajes virtuales y con otros jugadores, como se muestra en la Figura 28.

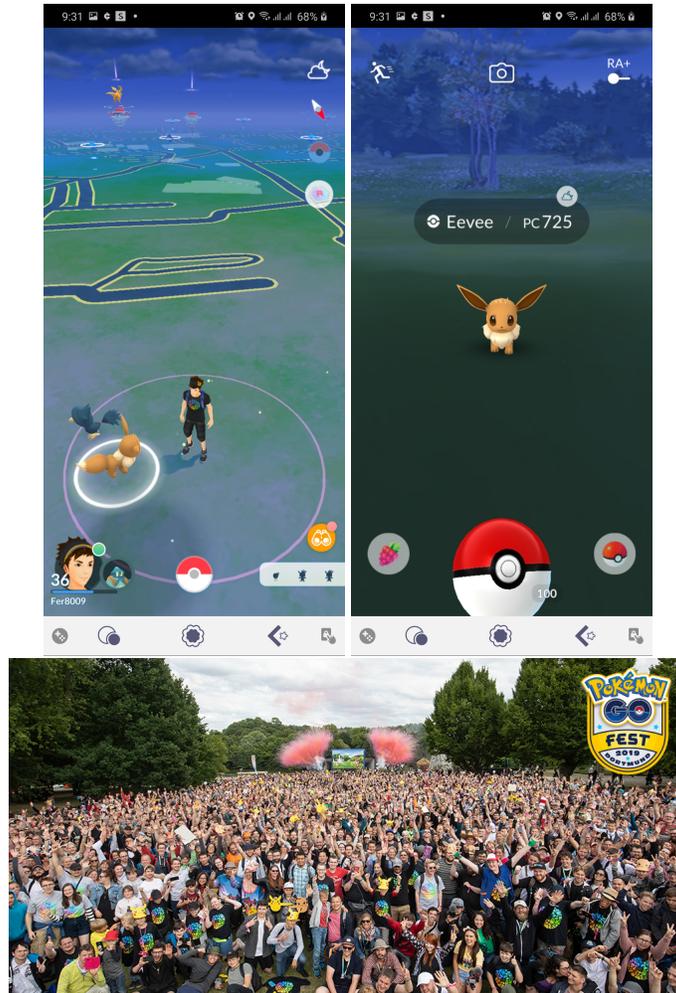


Figura 28: En la parte superior capturas de pantalla de la aplicación *Pokémon GO* y en la parte inferior la fotografía oficial del *Pokémon GO Fest Dortmund 2019*[38]

Por otro lado se tiene *BBC Civilizations AR*, la cual es una aplicación lanzada en 2018 y su principal objetivo es que los usuarios puedan interactuar con objetos históricos de diversas culturas, dentro de la aplicación es posible agrandar los objetos, rotarlos, caminar alrededor de ellos e incluso tiene una función de rayos X, para poder ver dentro del objeto. Algunos de los 41 objetos históricos que se pueden ver, son por ejemplo: una momia egipcia del Museo Torquay, El beso de Rodin del Museo Nacional de Gales y La Virgen y el Niño del Museo Nacional de Escocia, como se muestra en la Figura 29.[39]

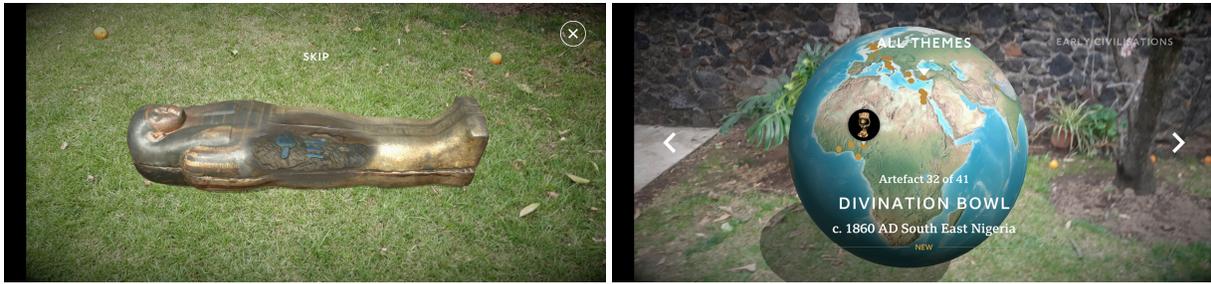


Figura 29: Capturas de pantalla de la aplicación *BBC Civilisations AR*

Por su parte, *Sephora Virtual Artist* es una aplicación lanzada por dicha compañía de maquillaje en 2017, con el propósito de que las personas puedan probarse distintos tipos de maquillajes y combinaciones de colores, también cuenta con tutoriales para aplicar el maquillaje, para que los usuarios determinen cuáles productos les agradan más y si así lo desean puedan posteriormente adquirirlos, como se muestra en la Figura 30.

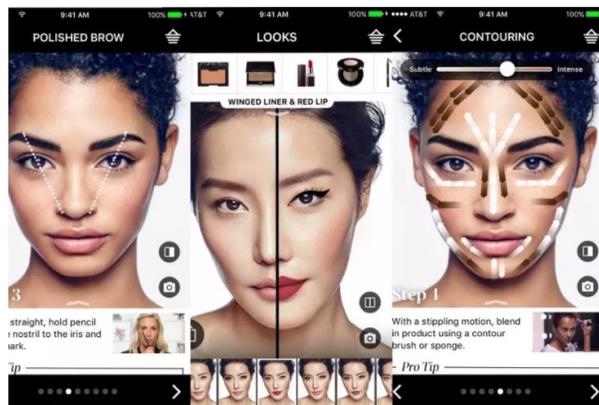


Figura 30: Capturas de pantalla de la aplicación *Sephora Virtual Artist*[40]

Del lado educativo, se encuentra *Visible Body*, la cual es una aplicación AR lanzada en 2018, útil para estudiantes y profesionales de áreas de la salud, pues es un atlas de anatomía que incluye modelos detallados de anatomía macroscópica masculina y femenina en 3D, microanatomía seleccionada de tejidos y órganos, cortes de cadáveres e imágenes diagnósticas mostradas al lado de secciones transversales en 3D, y animaciones interactivas de músculos y huesos, como se muestra en la Figura 31.[41]



Figura 31: Imagen promocional de la aplicación *Visible Body*[41]

Otro ejemplo del lado del entretenimiento es *The Mandalorian AR Experience*, la cual es una aplicación AR lanzada en 2020, basada en los personajes de la serie *The Mandalorian* del universo *Star Wars*, en la que se pueden recrear algunas escenas e interactuar con los personajes de la primera temporada de la serie de manera similar que en *BBC Civilisations AR*, ya que se puede acercar, girar y caminar alrededor de los objetos virtuales, como se muestra en la Figura 32. También Google habilitó un modelo 3D del personaje *The Child*, también conocido como *Grogu*, el cual puede ser visualizado en la mayoría de dispositivos móviles mediante AR (Figura 32 y 33).[42]

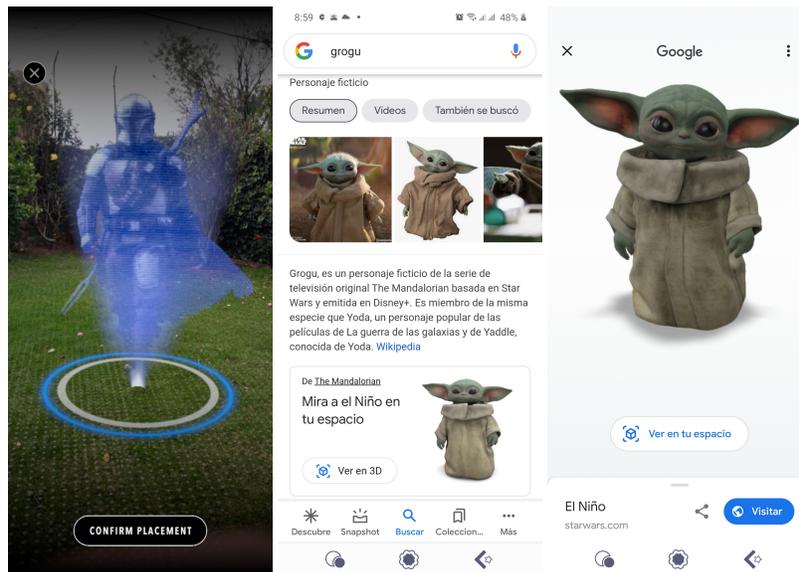


Figura 32: A la izquierda se muestra una captura de pantalla de la aplicación *The Mandalorian AR Experience*, al centro y derecha la función AR implementada por Google para visualizar a *The Child*.



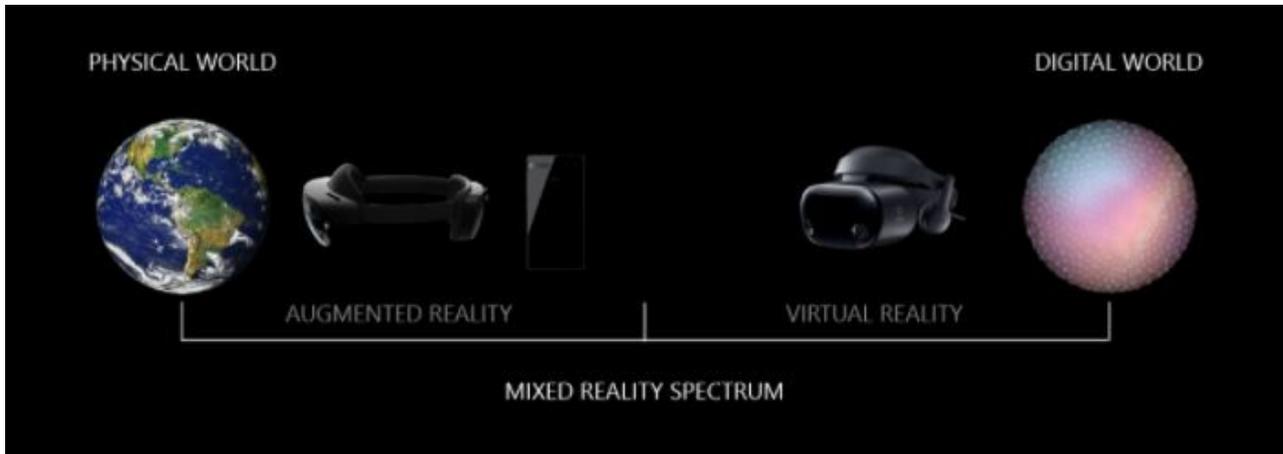
Figura 33: Imagen de la función AR implementada por Google para visualizar a *The Child*

La Realidad Mixta (XR) hace referencia a la combinación o la fusión de un entorno virtual junto con un entorno real donde ambos pueden coexistir. A veces, las personas también se refiere a ella como "realidad híbrida". En la realidad mixta, un usuario puede navegar por el mundo real y un entorno virtual simultáneamente. Los objetos virtuales están posicionados con precisión en el espacio del mundo real, si uno se mueve hacia un objeto, se hará más grande y viceversa. Y cuando uno se mueve a su alrededor, los objetos virtuales se ven desde diferentes ángulos y perspectivas, al igual que el comportamiento de los objetos reales. Algunas personas describen la experiencia a la de ver objetos holográficos. Los usuarios pueden manipular los objetos virtuales en la realidad mixta y pueden interactuar con éstos objetos virtuales como si existieran frente a ellos y ésta es la principal diferencia entre la AR y la XR: la atención en la XR está centrada en la interacción con los objetos virtuales y en la RA el mundo real es el centro de la experiencia, la cual se ve aumentada con elementos virtuales.



Figura 34: En esta figura se observa que el centro de una experiencia XR es la interacción con elementos virtuales.[43]

Al igual que la realidad aumentada, la realidad mixta emplea varias tecnologías, mediante las cuales se debe sustituir el mundo real abierto utilizando gafas de realidad aumentada o casco (HMD) con una cámara incluida, ya que la pantalla (generalmente un smartphone) superpone el ambiente o los elementos virtuales deseados, de modo que se vea lo más real posible, para lo cual es necesario ampliar el campo de visión de la cámara de los dispositivos, razón por la cual los desarrolladores agregan una lente especial al montaje de la cámara frontal del smartphone. Algunos ejemplos de la tecnología de realidad mixta son Microsoft HoloLens y Samsung HMD Odyssey+, como se observa en la Figura 35.[44]



**Figura 35:** En esta imagen se observa del lado izquierdo el Microsoft HoloLens y del lado derecho el Samsung Odyssey+ como ejemplos de realidad mixta y el hecho de que HoloLens está más cerca de la AR que Odyssey+, el cual está más cerca de la VR dentro del espectro de la realidad mixta.[45]

La Realidad Virtual (VR, por sus siglas en inglés) hace referencia al uso de tecnología informática para crear un entorno simulado, a diferencia de las interfaces de usuario tradicionales, la realidad virtual coloca al usuario dentro de una experiencia; en lugar de ver una pantalla frente a ellos, los usuarios están inmersos y pueden interactuar con mundos en tercera dimensión estimulando tantos sentidos como sea posible (vista, oído, tacto, e incluso olfato). Actualmente la principal limitante de la VR es el hardware necesario para la utilización de dichas experiencias, pues se debe usar una HMD estereoscópica para lograr que el usuario sea inmerso en el ambiente virtual, como se muestra en la Figura 36.



**Figura 36:** Sistema de VR, en el cual se utiliza una HMD estereoscópica, para situar al usuario dentro de un ambiente virtual.[46]

La principal diferencia que se puede observar entre la AR y la VR está en que la AR extiende o aumenta el mundo real, mientras que la VR reemplaza por completo el mundo real con uno virtual.[47] Otra diferencia notable recae en el hardware necesario para cada una de ellas: para experiencias en VR es esencial la visualización en algún tipo de visor o *headset* (generalmente utilizan tecnología de visión estereoscópica), para lograr inmergir al usuario en ese ambiente generado digitalmente, mientras que los contenidos de RA pueden o no utilizar algún tipo de HMD, ya que actualmente se están desarrollando muchas experiencias de RA que únicamente requieren el uso de un smartphone.[48]

En este punto ya se puede observar un conflicto entre dispositivos, categorías y etiquetas, pues todas estas tecnologías tienen funciones muy similares, por lo cual en 1994 se propuso una simplificación para la clasificación de éstas que se mencionará más adelante.

En la Figura 37 se aprecian de manera resumida las diferencias entre Realidad Aumentada, Realidad Virtual y Realidad Mixta.

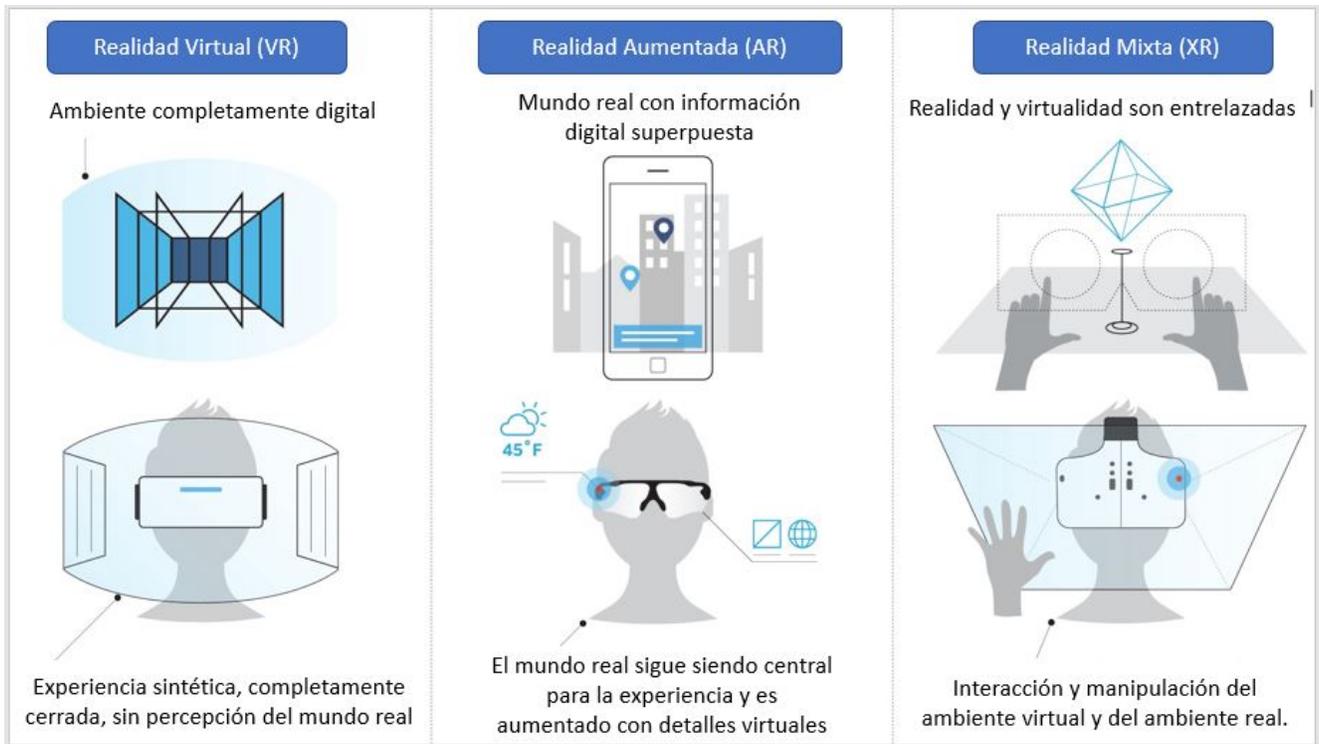


Figura 37: En esta figura se aprecian las principales diferencias entre AR, VR y XR.[49](traducido por el autor)

### 1.2.1. BREVE HISTORIA DE LA REALIDAD AUMENTADA

Actualmente, la Realidad Aumentada y la Realidad Virtual están teniendo mucha popularidad y su difusión es cada vez mayor. Estas tecnologías se están utilizando para diversas aplicaciones, como por ejemplo: entretenimiento, educación, cuidado de la salud y experiencias comerciales. Ambas son tecnologías que tienen un pasado en común, el cual se remonta a incluso antes de que existieran los términos AR y VR.

En 1862, el científico John Henry Pepper creó una ilusión óptica ("*Pepper's Ghost*"), la cual utiliza las propiedades reflectantes del vidrio y de los paneles de plástico diseñados e incorporados más tarde, mediante los cuales los objetos son proyectados virtualmente en el escenario desde un lugar invisible para el público, ya sea debajo o al lado de la superficie de presentación. y así aparecer en el campo de visión de los espectadores. Ésta ilusión óptica es todavía bastante utilizada en la actualidad. Se puede encontrar en museos, parques de diversión, algunos headsets de RA e incluso como hologramas escénicos de estrellas de música.[50] En las Figuras 38 y 39 se puede observar el principio de funcionamiento de esta ilusión óptica.



Figura 38: En esta imagen se puede observar el funcionamiento de la ilusión óptica *Pepper's Ghost* al ser utilizada en un teatro.[50]



Figura 39: A la izquierda se observa el funcionamiento de la ilusión óptica *Pepper's Ghost* al ser utilizada en el *ABBA The Museum* en Estocolmo, Suecia.[51]. A la derecha se observa un *headset* o visor de AR estereoscópico, que funciona con esta ilusión óptica, al proyectar mediante un *smartphone* una imagen distinta para cada ojo.[52]

Sin embargo no fue sino hasta 1962, que el cineasta Morton Heilig creó un prototipo, el cual describió como "el cine del futuro" (*The cinema of the future*), llamado "Sensorama" (Figura 40), el cual consistía en una cabina parecida a una máquina arcade capaz de estimular los sentidos, mediante imágenes tridimensionales a través de una pantalla estereoscópica, vibraciones, sonidos e incluso olores, con lo cual este prototipo se convirtió en el primer intento de añadir información creada por computadora a una experiencia.[35] [53][54]



Figura 40: En esta imagen se muestra una imagen publicitaria del sistema "Sensorama".[55]

Posteriormente, en 1965, el investigador Ivan Sutherland escribió un ensayo en el cual postuló la pantalla definitiva, con la famosa cita:

The ultimate display would, of course, be a room within which the computer can control the existence of matter. A chair displayed in such a room would be good enough to sit in. Handcuffs displayed in such a room would be confining, and a bullet displayed in such a room would be fatal. With appropriate programming such a display could literally be the Wonderland into which Alice walked. [La pantalla definitiva, sería por supuesto, una habitación en la cual la computadora puede controlar la existencia de materia. Una silla mostrada en tal habitación sería lo suficientemente buena para poder sentarse, unas esposas mostradas en tal habitación serían confinantes y una bala mostrada en tal habitación sería fatal. Con la programación adecuada, dicha habitación podría literalmente ser el país de las maravillas, por el cual Alicia caminó.][56]

Y tres años más tarde, en 1968 creó "La espada de Damocles", con la finalidad de crear la pantalla definitiva: una interfaz capaz de transformar el mundo físico. Este prototipo fue una de las primeras pantallas montadas en la cabeza (HMD, por sus siglas en inglés: *Head Mounted Display*), contaba con sensores que le permitían al sistema saber la posición de la cabeza y era tan pesado que debía suspenderse del techo mediante un brazo mecánico; fue uno de los primeros experimentos en agregar elementos digitales al mundo real, pues para su funcionamiento, utilizaba una pantalla transparente, con la cual se

sobreponían imágenes sencillas, hechas con líneas al mundo real. Uno de los experimentos que se realizaron con esta HMD consistió en pedirle a los usuarios alinear su cabeza con las distintas letras (N,S,E,W) que contenía el cubo mostrado en la Figura 41[57].

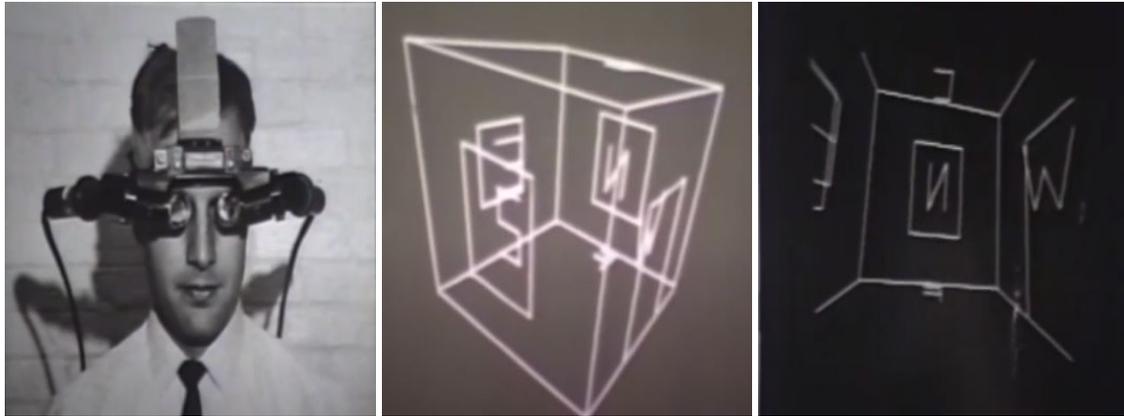


Figura 41: A la izquierda se puede observar la HMD *Sword of Damocles* siendo utilizada, al centro se muestra la imagen más compleja que se utilizó con este dispositivo y a la derecha se observa como era la vista en primera persona de la imagen central, sin el fondo del mundo real.[57]

Más tarde, en 1975, Myron Krueger, un artista computacional, creó la "Videoplace", una habitación que respondía a los movimientos y acciones de los usuarios, sin estar impedidos por el uso de gafas o lentes para la interacción de éstos con el entorno de lo que llamó "realidad artificial".[35] La cual puede verse como un experimento de interacción bastante similar y anterior al Microsoft Kinect.[58][59] Su funcionamiento se muestra en la Figura 42.

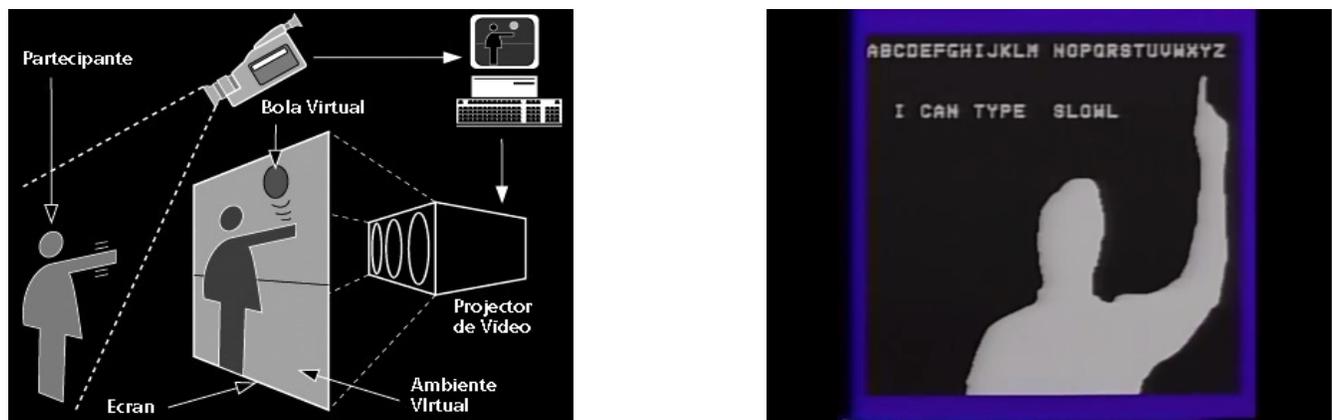


Figura 42: En la imagen derecha se muestra el funcionamiento de la habitación "Videoplace"[58] y en la imagen izquierda se observa uno de los experimentos con dicha interfaz.[59]

Formalmente, el término Realidad Aumentada fue acuñado en el año 1992 por Thomas Caudell y David Mizell, dos ingenieros que trabajaron para Boeing y utilizaron una HMD transparente para ayudar a los ingenieros aeronáuticos con

los complejos diagramas de cableado, superponiendo imágenes de éste a través de la HMD en plantillas reutilizables. A ésta tecnología la bautizaron como "Realidad Aumentada" y su objetivo era mejorar la eficiencia y permitir una reducción de costos de las operaciones humanas en la manufactura de aeronaves, logrando que en vez de tener múltiples plantillas de instrucciones de cableado para cada una de las aeronaves, ahora los trabajadores pudieran acceder a ellas al colocarse la HMD, como puede observarse en la Figura 43.[48] [53]



**Figura 43:** En esta imagen se puede observar a una persona utilizando la HMD creada en 1992 por los ingenieros de Boeing David Mizell y Thomas Caudell para facilitar la colocación del cableado en las aeronaves.[60]

También en 1992 fue creado el sistema "Virtual Fixtures", por el inventor norteamericano Louis Rosenberg en los laboratorios de la Fuerza Aérea Norteamericana *USAF Armstrong Labs* en San Antonio, Texas, este sistema es considerado el primer sistema completamente inmersivo de RA y se puede observar en la Figura 44.

Debido a que a principios de la década de los 90 los gráficos generados por computadora eran muy lentos como para poder presentar un entorno fotorealista, el sistema *Virtual Fixtures* utilizaba dos robots, controlados por un exoesqueleto controlado por el usuario y para poder crear la experiencia inmersiva, se empleó una configuración óptica, que utilizaba un par de lupas binoculares alineadas de tal modo que la vista del usuario de los brazos del robot se adelantara para que éstos aparentaran estar en la ubicación exacta de los brazos físicos reales del usuario, lo cual resultó en una experiencia inmersiva en la que el usuario movía sus brazos, mientras veía los brazos del robot en el lugar donde deberían estar sus brazos, este sistema también empleó superposiciones virtuales generadas por computadora, como por ejemplo: barreras físicas, campos y guías simuladas, diseñadas para ayudar al usuario mientras realiza tareas físicas reales.

Con *Virtual Fixtures* se llevaron a cabo experimentos con la Ley de Fitts utilizando un grupo de sujetos de prueba humanos, demostrando por primera vez que se podría lograr una mejora significativa en el desempeño humano de tareas diestras del mundo real al proporcionar superposiciones de realidad aumentada inmersivas a los usuarios.[61]



**Figura 44:** En esta imagen se muestra el sistema *Virtual Fixtures*. A la izquierda se observa a Louis Rosenberg probando el sistema y a la derecha se observa una de las tareas mostradas mediante gráficos generados por computadora.[62]

### 1.2.2. MÉTODOS DE DETECCIÓN PARA REALIDAD AUMENTADA

El seguimiento, o *tracking* en inglés, en AR es el proceso de rastrear las coordenadas de la escena virtual y los objetos en movimiento en tiempo real, dicho seguimiento en tiempo real es esencial en las aplicaciones de AR y según el tipo de aplicación (para uso en interiores o exteriores) es que se puede determinar la técnica de seguimiento que se utilizará y estas técnicas pueden ser clasificadas en tres categorías: basada en sensores, basada en visión o híbrida.

En el seguimiento basado en sensores, como su nombre indica, se utilizan dispositivos para rastrear información de posicionamiento para los modelos virtuales en 3D, los sensores pueden ser electromagnéticos o acústicos y el funcionamiento es el siguiente: se colocan transmisores en la ubicación del mundo real donde se desea colocar el modelo virtual y las señales de estos transmisores son posteriormente detectadas por sensores para calcular su posición relativa al usuario. Un aspecto a considerar al utilizar sensores electromagnéticos es que los campos electromagnéticos no siempre son precisos, debido a que pueden estar alterados por interferencia electromagnética, por ejemplo al utilizar este método con teléfonos celulares u otros dispositivos móviles no siempre se puede obtener un buen resultado únicamente utilizando este tipo de sensores, pues estos dispositivos per sé son transmisores y receptores de información electromagnética, lo cual genera interferencia. Otra técnica de seguimiento basada en sensores es el seguimiento acústico, éste puede ser utilizado para ampliar el entorno de sonido real con sonido virtual e incluso dar el efecto de sonido 3D, con el cual el usuario puede experimentar que el sonido proviene de distintas direcciones, algo así como con la AR tradicional en la cual se superponen elementos 3D al mundo real.[63]

Un ejemplo del seguimiento acústico combinado con el seguimiento electromagnético (utilizando una IMU *Inertial Measurement Unit*) se observa en la Figura 45, en la cual Yang et al. discuten un modelo de *Audio Augmented Reality* (realidad aumentada auditiva) para usar en un museo.[64]

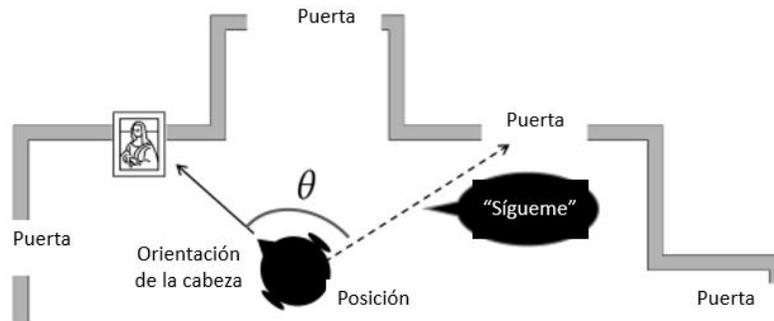


Figura 45: Ejemplo de un escenario para la realidad aumentada acústica en un museo.[64]

Las técnicas de seguimiento basadas en visión proporcionan una alternativa al seguimiento basado en sensores, en éstas se utiliza una cámara para capturar video en tiempo real de la escena, después los fotogramas del video son procesados para encontrar características artificiales o naturales en el video, con las cuales los objetos virtuales deben estar alineados. Las características artificiales son las que se colocan específicamente en la escena a propósito para que el dispositivo obtenga información de seguimiento, por otro lado, las características naturales son parte del entorno real, por lo tanto las técnicas de seguimiento basadas en visión se dividen en: seguimiento basado en marcadores y seguimiento basado en características naturales. El seguimiento basado en marcadores utiliza el seguimiento de características artificiales, los cuales suelen ser imágenes predefinidas, impresas en una hoja de papel y colocadas en la escena, posteriormente se entrena la aplicación AR usando algoritmos de AI (Inteligencia Artificial, por sus siglas en inglés) para reconocer estos marcadores, éstos son detectados en el fotograma de video y procesados por una API (*Application Programming Interface*) y se crea una matriz de transformación y dependiendo de la posición y orientación del marcador, se renderiza el objeto virtual sobre el video. El seguimiento utilizando esta técnica es relativamente rápido y suave en comparación con otras técnicas y las imágenes y formas que conforman los marcadores suelen ser personalizables, pues la mayoría de las API disponibles ofrecen la opción de crear marcadores propios, sin embargo uno de los inconvenientes al utilizar esta técnica es la modificación del entorno con marcadores que podrían ser una distracción para el usuario. Por el momento, el seguimiento basado en marcadores se utiliza para realizar muchas de las aplicaciones de AR, debido a la disponibilidad de APIs comerciales para el desarrollo de dichas aplicaciones. Un ejemplo de este tipo de marcadores se observa en la Figura 46.

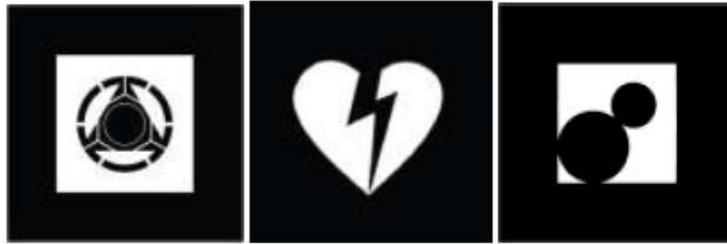


Figura 46: Ejemplo de marcadores para AR.[63]

Por otro lado, el seguimiento basado en características naturales se basa en el reconocimiento de un objeto completo o características distintivas de un objeto o imagen del mundo real y debido a que podría haber posibilidades ilimitadas de características en el mundo real, estas técnicas suelen ser más demandantes computacionalmente y los algoritmos de seguimiento requieren un entrenamiento más extenso, sin embargo ya también es bastante común encontrar APIs que ofrecen dicho tipo de *tracking*, como se muestra en la Figura 47.

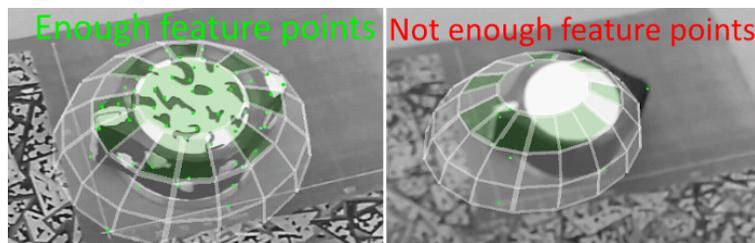


Figura 47: Ejemplo del entrenamiento de una API para reconocimiento de objetos, del lado izquierdo ya se encuentra el objeto identificado con suficientes puntos característicos y del lado derecho aún no se cuenta con suficientes puntos. [65]

Finalmente, las técnicas de seguimiento híbrido implican más de un método para calcular la información de posicionamiento, éstas se consideran útiles en entornos exteriores donde no es posible emplear marcadores u otros sistemas similares.[63] Es aquí donde entra el concepto conocido como *Spatial Computing*, en el cual la app tiene conocimiento de la posición, orientación y del ambiente que rodea al dispositivo en el que se está ejecutando, para esto se utilizan varios sensores, como la IMU, pero como las mediciones de este tipo de sensores suelen estar afectadas por ruido, el fenómeno de la deriva o *drift* en inglés, se suelen utilizar dos alternativas: La primera consiste en utilizar las lecturas de más de uno de los sensores para mejorar la precisión y estabilidad de las lecturas y la segunda consiste en utilizar filtros de Kalman, el cual es un algoritmo para actualizar, observación a observación, la proyección lineal de un sistema de variables a partir del conjunto de información disponible, según se va disponiendo de nueva información[66], lo cual lo convierte en un método muy adecuado para sistemas en tiempo real, como lo es la AR. Otra técnica importante en este punto es la llamada

”SLAM” *Simultaneous Localization and Mapping*, con la cual se mapea el ambiente y al mismo tiempo se calcula dónde se está dentro de él y se usan las características detectadas por el sistema de visión computarizada como puntos de referencia o anclas espaciales para poder hacer la corrección periódica del drifting. El uso de SLAM ha ido en aumento debido a que proporciona mayor realismo al usuario al realizar un *tracking* más preciso y permitir calcular profundidad, con lo cual se pueden ocluir objetos virtuales, lo cual proporciona una experiencia mucho más inmersiva, lo cual se observa en la Figura 48.[67] Y por último se resumen los tres métodos de detección y seguimiento en AR en el cuadro sinóptico de la Figura 49.



Figura 48: A la izquierda se observa un ejemplo de oclusión incorrecta y a la derecha se observa una correcta oclusión.[68]

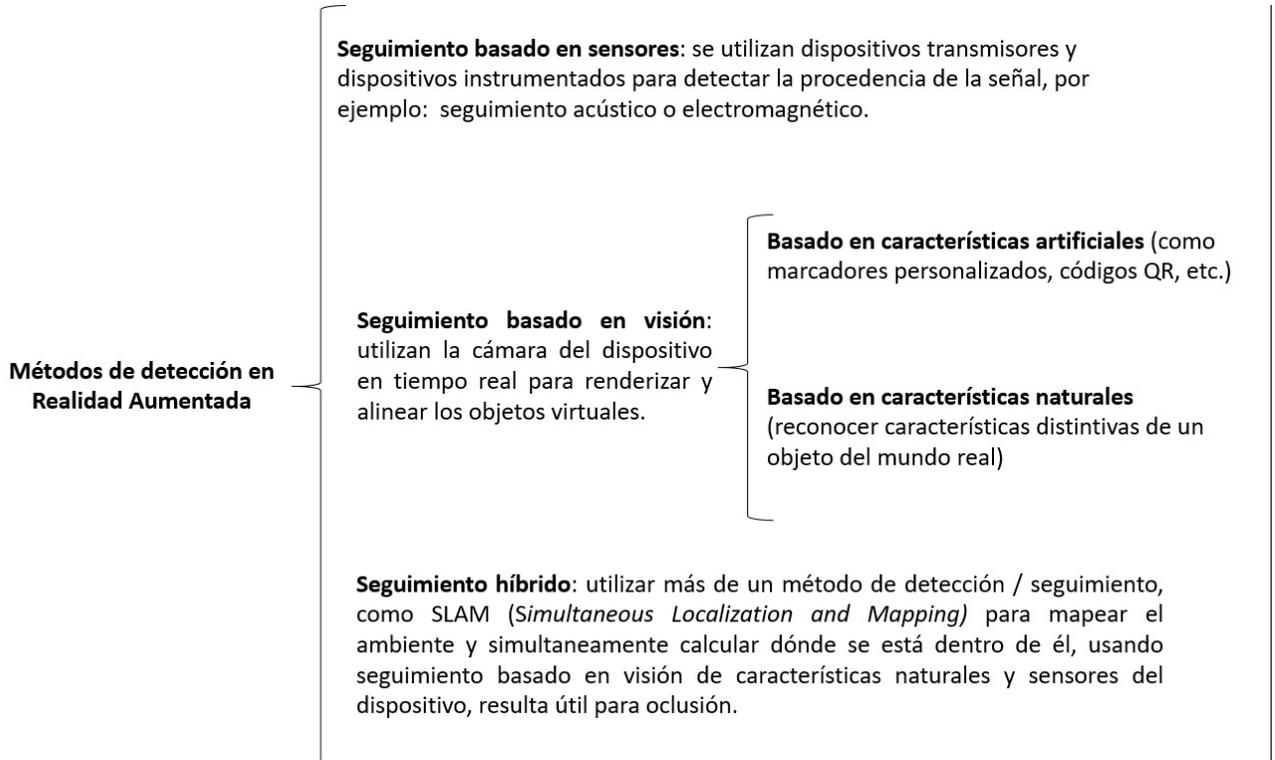


Figura 49: Cuadro sinóptico con los métodos de detección y seguimiento en AR.

### 1.2.3. EL METAVERSO Y LA REALIDAD AUMENTADA

Como se mencionó al principio del capítulo, actualmente existen tantas tecnologías y dispositivos con nombres y propósitos tan similares, que en ocasiones se entra en un dilema, por lo cual resulta útil crear una clasificación y etiquetado, para poder facilitar la conversación y comunicación, y de esta manera avanzar hacia la construcción de una taxonomía y definición.

El término "Metaverso" proviene de la novela de ciencia ficción "*Snow Crash*" de Neal Stephenson, donde los seres humanos, como avatares, interactúan entre sí junto con agentes de software en un ambiente tridimensional que utiliza una metáfora del mundo real, de modo que el Metaverso se entiende actualmente como un espacio virtual compartido, creado por la convergencia de la realidad física virtualmente mejorada y el espacio virtual físicamente persistente, y es una fusión de ambos, que permite a los usuarios experimentarlo como cualquiera de ellos.

La industria, inversionistas, gobiernos y consumidores admiten que la AR, VR y realidad mixta utilizando HMDs tiene características únicas y especiales, pero muchos cuestionan si es algo que realmente se llegaría a usar en la vida cotidiana, lo cual significa que aún no se llega al punto de integración de las HMD a la vida cotidiana, pero se están haciendo progresos para poder lograrlo.

La historia de la HCI ha demostrado que la forma en que interactuamos con nuestros dispositivos ha evolucionado y sigue haciéndolo continuamente, ya que con cada avance en informática, se ha requerido que la tecnología surja con un nuevo método de entrada: desde el teclado, el mouse, el joystick, tecnología táctil y touch, etc. Sin embargo, los dispositivos de próxima generación utilizan métodos de control heredados de las generaciones anteriores, como control con la cabeza, manos y voz. Estas interacciones también deberán evolucionar y los sistemas de realidad aumentada van a ser los cuales rompan esos paradigmas y forzarán la creación de nuevas interfaces de usuario naturales como la voz y seguimiento ocular, abandonando enfoques anteriores tomados de las pantallas táctiles y gesticulación utilizada en la realidad virtual.

La AR es tan sólo una parte del metaverso y en 1994, Paul Milgram y Fumio Kishino definieron el continuo de la realidad - virtualidad (*Reality - Virtuality Continuum*) como una escala que abarca desde el ambiente real en el extremo izquierdo de la gráfica hasta el ambiente virtual, la cual puede tomarse como una simplificación del metaverso. La AR y la VR están localizadas dentro de este espectro, estando la AR más cerca del extremo del ambiente real y la VR más cerca del extremo del ambiente virtual, como se muestra en las Figuras 50 y 51.[35]



Figura 50: Escala del continuo de la realidad - virtualidad propuesto por Milgram y Kishino en 1994.[35]Imagen traducida por el autor

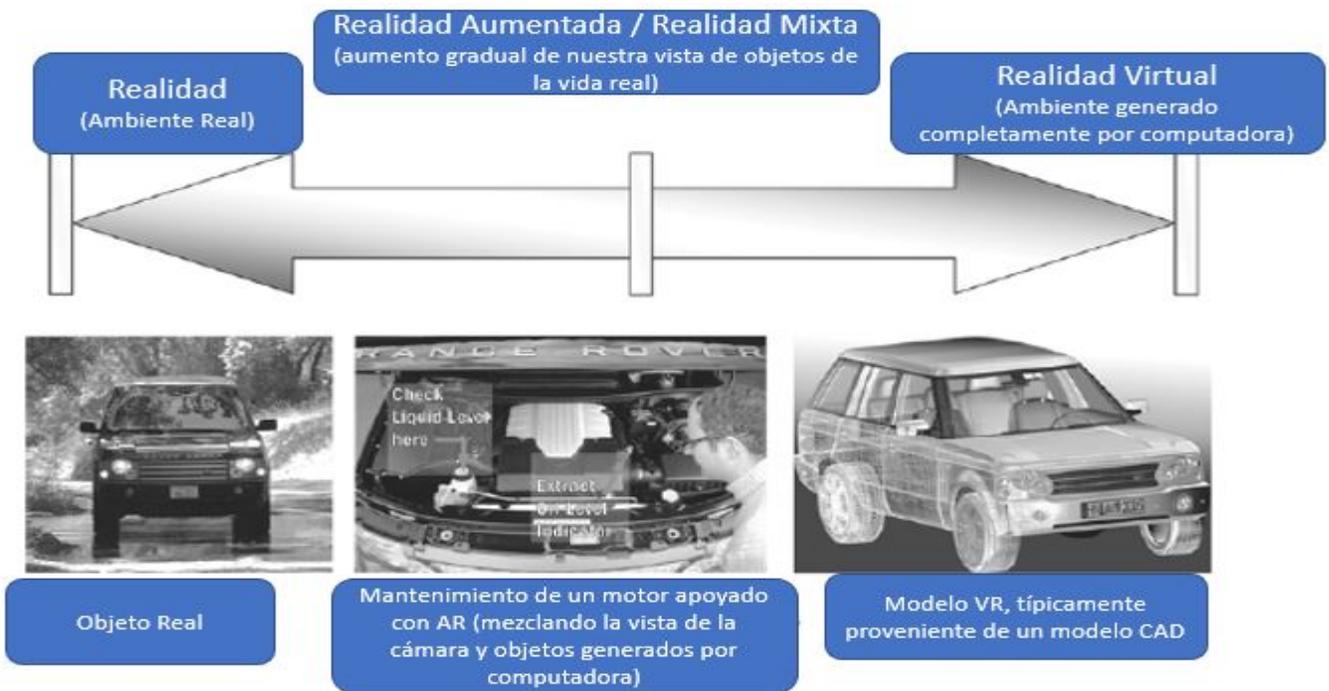


Figura 51: Ejemplo del continuo de la realidad - virtualidad.[69]Imagen traducida por el autor

#### 1.2.4. TIPOS DE HARDWARE PARA REALIDAD AUMENTADA

La realidad aumentada es un campo muy diverso, robusto y complicado, por lo cual existen varios modos en los cuales se pueden clasificar las experiencias y tecnologías de AR.

A pesar de que existen múltiples clasificaciones para los dispositivos y aplicaciones AR, de primera instancia, es posible hacer una gran división entre dos tipos de tecnologías: *wearable* y no *wearable*. [47]

La primera categoría "*wearable*" (dispositivos portátiles o estacionarios), como su nombre lo indica son aquellos que los usuarios pueden vestir e incluyen auriculares, cascos, HMDs y algún día incluirá lentes de contacto, como se muestra en la Figura 52, aunque como se mencionó anteriormente, debido a las similitudes entre las tecnologías y sus alcances, esto podría llegar a confundirse con realidad mixta.



Figura 52: A la izquierda se observa un dispositivo de RA usado para asistencia técnica y mantenimiento remoto [70] y a la derecha se muestran diferentes tipos de dispositivos de RA que han sido desarrollados [71].

Algunos de los dispositivos más famosos en esta categoría (*wearable*) son: Microsoft HoloLens, Google Glass y Lenovo Mirage AR. En cuanto a disponibilidad, Google Glass está pensado para empresas y no tanto para electrónica de consumo diario, por otro lado Microsoft HoloLens está orientado a un sector más amplio, pues se vende en distintos paquetes, algunos para desarrollo, educación y otros para empresas, sin embargo estos dos dispositivos podrían clasificarse también dentro de la categoría de realidad mixta. [72] Finalmente, el Lenovo Mirage AR está pensado únicamente para entretenimiento, pues se ofrece con dos experiencias de AR y no existe un Software Development Kit (SDK) abierto al público para crear nuevas experiencias. Además, este dispositivo depende de un smartphone, el cual ejecuta la experiencia y se utiliza visión estereoscópica utilizando una ilusión óptica "*Pepper's Ghost*", a diferencia de los dos dispositivos anteriores, que no dependen de uno. Los tres dispositivos se muestran en la Figura 53 y 54.



Figura 53: En la imagen de la izquierda se observa el dispositivo Microsoft HoloLens 2[73] y en la imagen de la derecha se muestra el dispositivo Google Glass 2.[74]



Figura 54: Dispositivo Lenovo Mirage AR siendo utilizado.[75]

La segunda categoría "no *wearable*" o móvil, al contrario de la categoría anterior, los usuarios interactúan con ellos al sostenerlos o interactuar con ellos con las manos principalmente e incluyen dispositivos móviles (smartphones, tabletas, computadoras portátiles, etc.) dispositivos estacionarios (televisores, PC, juegos, etc.) y pantallas de visualización.[47] Debido al desarrollo y apertura al público en general de los SDK ARCore de Google, ARKit de Apple, Vuforia, AR Foundation y MARS de Unity, actualmente es cada vez más común encontrar experiencias de AR que pertenezcan a la categoría de AR móvil, además de que muchas personas tienen acceso a un smartphone, haciendo que dichas experiencias tengan aún más difusión.

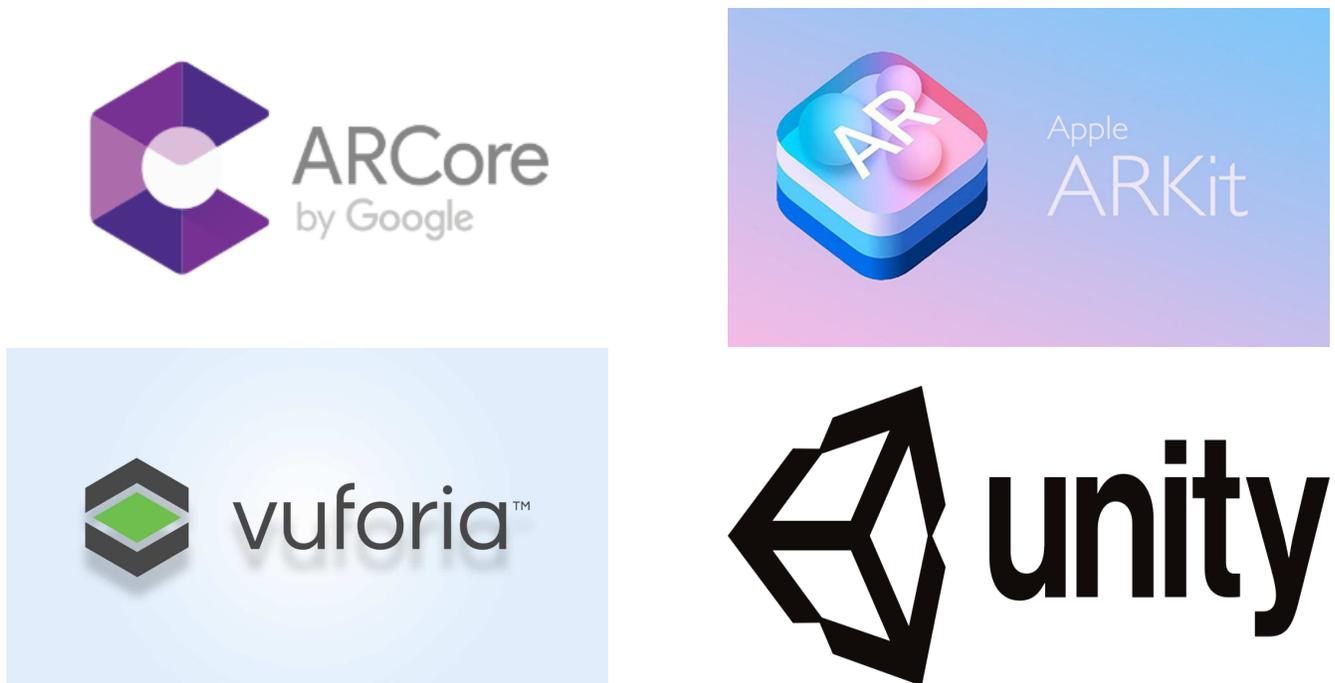


Figura 55: Logos de los SDKs más famosos para el desarrollo de aplicaciones AR.[76][77][78][79]

Por otro lado, también se ha estado experimentando con pantallas basadas en proyección (*Projection based displays*), las cuales suelen utilizar proyectores portátiles o no portátiles para proyectar información virtual en el mundo real, de modo que los usuarios no tengan que usar una HMD, por lo cual este tipo de pantallas, junto con una computadora *wearable* pueden proporcionar una solución móvil personalizada al utilizar AR proyectiva, como se muestra en la Figura 56.[63]



Figura 56: Dispositivo de AR proyectiva del proyecto Sixth Sense del MIT.[63]

El que se podría considerar el primer juego de AR lanzado para celulares fue "AR Tennis", para celulares Nokia en el año 2005 y utilizaba AR mediante el uso de marcadores que eran detectados por el celular para poder proyectar la pelota. También a partir de 2008 se volvió popular el uso de AR en las revistas, haciendo posible desplegar animaciones en ciertas

páginas e incluso con playeras, pero no fue hasta 2013 que Niantic lanzó Ingress, un juego de AR que utiliza el sistema de posicionamiento global (GPS) del smartphone donde se ejecuta y así logró aumentar la interacción con el mundo real.[80] Una de las aplicaciones más famosas de este tipo (*Location based AR*) ha sido Pokémon GO, lanzado en 2016, la cual ha ido teniendo mejoras desde su lanzamiento para lograr una interacción cada vez mayor con los personajes virtuales. Sin embargo ha habido múltiples aplicaciones con distintos propósitos que utilizan la AR, por ejemplo: venta de muebles (IKEA), filtros para el rostro, museos, etc. Todo lo anterior se ilustra en la Figura 57.

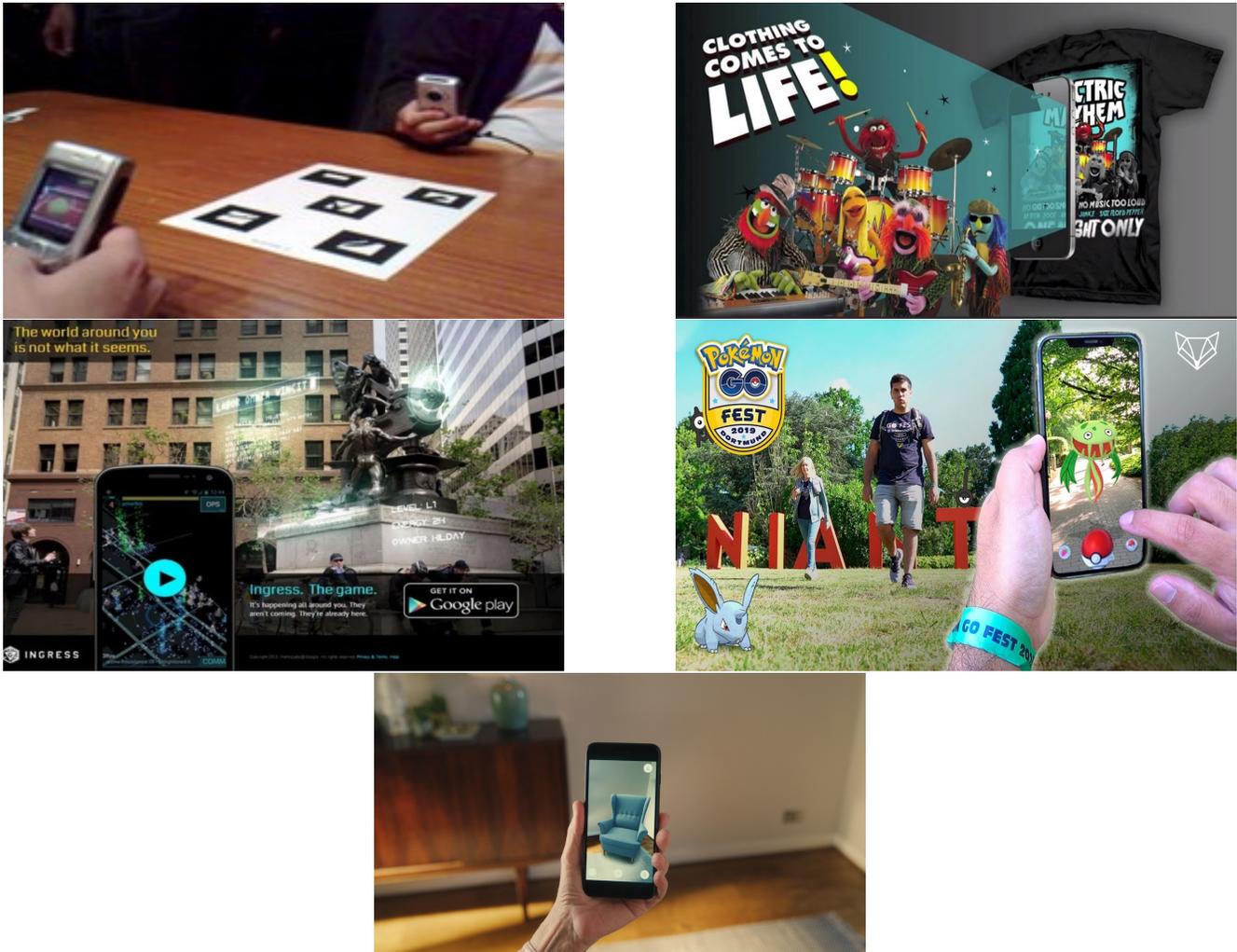


Figura 57: A la izquierda arriba se muestra el juego "AR Tennis" de Nokia[81], a la imagen derecha arriba se muestra una de las primeras marcas de ropa en utilizar AR[82], a la izquierda abajo se observa una imagen promocional del juego Ingress[83] y a la derecha abajo se observa una imagen del juego Pokémon GO[84]. Finalmente, se observa la aplicación IKEA Place[85].

### 1.2.5. BUENAS PRÁCTICAS DE DISEÑO DE INTERFACES PARA REALIDAD AUMENTADA.

Según Adobe XD, un editor de gráficos desarrollado por Adobe Inc. para diseñar y crear prototipos de experiencias de usuario tanto para aplicaciones móviles como páginas web[86], existen ciertas reglas o buenas prácticas para el desarrollo de aplicaciones de Realidad Aumentada, algunas de éstas son[87]:

- Entender el problema y asegurarse de que la RA es la tecnología adecuada para solucionarlo.
- El diseño de experiencias RA debe enfocarse en un negocio y objetivos del usuario, es decir, debe ser útil y proporcionar un valor añadido a quien la utilice, como por ejemplo ayudar a reducir el tiempo de alguna tarea.



**Figura 58:** Aplicación AR para la industria manufacturera, que puede ser utilizada para mantenimiento, ensamblaje complejo y asistencia remota.[88]

- Considerar las capacidades del hardware, ya que en ocasiones no todos los dispositivos cuentan con los sensores necesarios, por lo cual no deberá ofrecerse una aplicación a un dispositivo que será incapaz de ejecutarla.



**Figura 59:** Mensaje mostrado en una tienda de aplicaciones cuando el dispositivo no es compatible con la aplicación deseada y no es permitido que ésta sea descargada.[89]

- No limitarse a bastidores cuadrados, este punto se refiere a que los diseñadores deben considerar que en este tipo de aplicaciones, los smartphones son como una ventana por la cual el usuario puede observar una versión ampliada del mundo.



**Figura 60:** Uso de elementos circulares y una interfaz de usuario muy sencilla, para lograr mantener el mundo real como el centro de la experiencia de AR.[90]

- Diseñar interacciones adecuadas y cómodas para los usuarios.
- Darle al usuario una idea del espacio que requerirá para utilizar la aplicación.



**Figura 61:** Imagen que ilustra el principio acerca del espacio requerido para las interacciones en AR.[91]

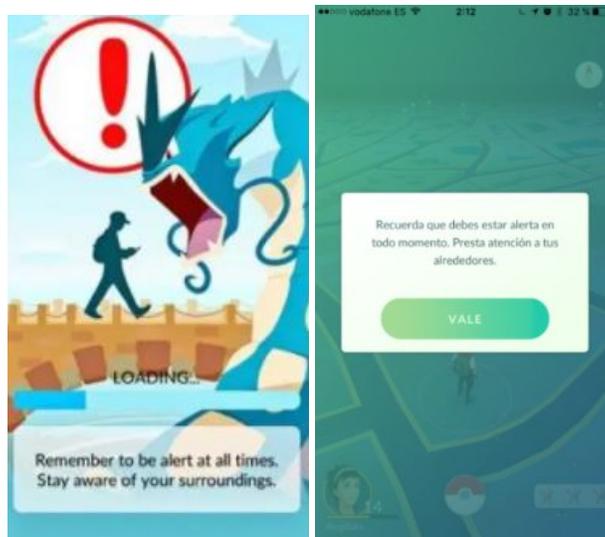
- Diseñar para el espacio privado o público:
  - Espacio privado: En un entorno privado (como una casa u oficina), los diseñadores pueden contar con que los usuarios podrán tener largas sesiones e interacciones complejas. Todo el cuerpo del usuario podría participar en la interacción.
  - Espacio público: En entornos públicos (por ejemplo al aire libre o en algún museo), es fundamental centrarse en que las sesiones del usuario serán mayormente cortas. Porque, independientemente de cuánto disfruten las personas de la experiencia de RA, no querrán caminar con las manos en alto sosteniendo un dispositivo

durante un período de tiempo prolongado, aunque el autor ha visto en persona que a algunos individuos no les incomoda el llevar hasta ocho dispositivos ejecutando la misma aplicación, pues disfrutaban mucho de ella, pero son la minoría.



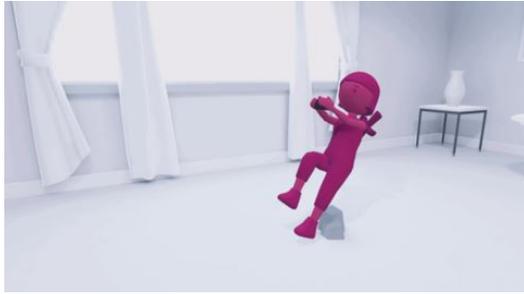
**Figura 62:** A la izquierda se observa una aplicación de AR pensada para usarse en el hogar y a la derecha se muestra una aplicación pensada para usarse en un museo.[85][92]

- Diseñar para la seguridad, en ocasiones los usuarios pueden estar demasiado inmersos en alguna experiencia AR, por lo que podrían ignorar objetos físicos que los rodean, lo cual podría dar lugar a accidentes. Para minimizar ese riesgo, se deben crear recordatorios para que los usuarios revisen su entorno.



**Figura 63:** A la izquierda se observa que en la pantalla de carga de la aplicación Pokémon GO se muestra el mensaje de "Recuerda estar alerta todo el tiempo. Permanece atento de tus alrededores" y a la derecha lo primero que se muestra cuando el juego ya ha cargado es otro mensaje para prevenir a los usuarios de su entorno.[93][94]

- Nunca hacer que los usuarios tengan que caminar hacia atrás, pues podrían tropezar.



**Figura 64:** En la imagen se ilustra el porque no se debe hacer que los usuarios caminen hacia atrás mientras interactúan con AR.[86]

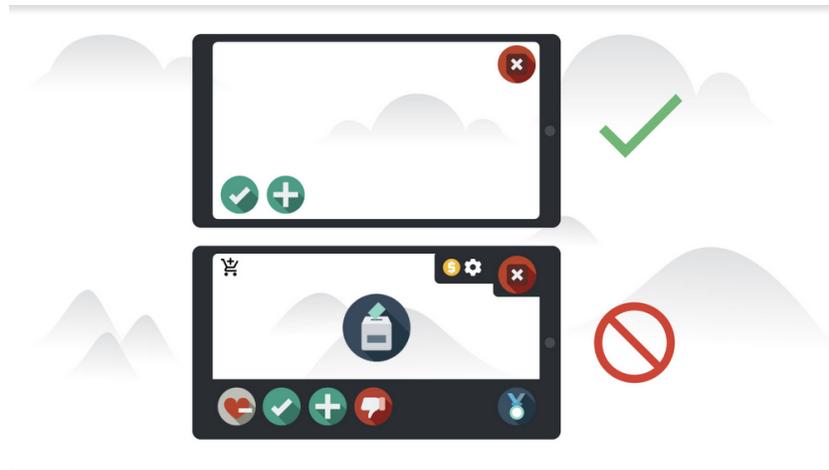
- Considerar las limitaciones físicas, como que los usuarios se pueden cansar de sostener su dispositivo durante periodos de tiempo largos.
- Permitirle a los usuarios tomar pausas, pues muchas veces éstos temen perder su progreso, es por esta razón por la cual se le debe permitir al usuario retomar lo que se encontraba haciendo en el punto donde lo dejó.
- Hacer pruebas de usabilidad, pues de este modo se descubre si el producto es cómodo para el usuario.



**Figura 65:** Laboratorio de usabilidad, idealmente utilizado para realizar las pruebas con usuarios y registrar mediante cámaras sus interacciones.[95]

- Minimizar las entradas, las experiencias de AR deben diseñarse para que requieran la menor cantidad posible de interacciones físicas de parte de los usuarios. Cuando los usuarios miran a través de la pantalla del dispositivo una imagen aumentada, será difícil para ellos ingresar datos al mismo tiempo. Se deben utilizar maneras alternativas de recopilar información. Por ejemplo, utilizando una cámara del dispositivo o sensores para esta interacción.
- No tener una interfaz de usuario que abarque la mayor cantidad de espacio, pues una buena experiencia de AR sumerge a los usuarios en las interacciones y sólo sucede cuando las personas creen que lo que ven en la pantalla es real. Es fundamental dedicar la mayor parte de la pantalla posible a mostrar el mundo físico y los objetos virtuales

de la aplicación. Se debe evitar a toda costa abarrotar la pantalla con información y controles de UI visibles porque disminuyen la experiencia de inmersión.



**Figura 66:** En la parte superior se muestra una interfaz de usuario para AR adecuada y en la parte inferior una que contiene elementos a evitar en el diseño de UI para AR.[91]

- Se debe poner esfuerzo para lograr ilusiones convincentes al colocar objetos realistas, para ayudar a los usuarios a creer que los objetos AR son reales, se debe asegurar que la aplicación actualice la escena 60 veces por segundo, para que los objetos no parezcan parpadear. También el fenómeno de la oclusión, en el cual los objetos del mundo real o virtuales obstruyen la vista de otros dependiendo de su ubicación, ayuda bastante a lograr dicha ilusión.



**Figura 67:** A la izquierda se muestra una experiencia AR sin oclusión y a la derecha se muestra con oclusión.[96]

- Usar audio, éste puede mejorar la usabilidad de una experiencia. Un ejemplo de esto podría ser agregar un efecto de sonido para confirmar alguna acción por parte del usuario. La música de fondo también puede ayudar a envolver a las personas en el mundo virtual al crear el ambiente adecuado.

- Ofrecer un *onboarding* (introducción) sencillo, evitando mostrarle al usuario todas las funciones clave de golpe, pues éstas estarán fuera de contexto.



Figura 68: En la imagen se muestra un tipo de onboarding que no es tan útil en AR.[86]

- Usar patrones de UI que sean familiares a los usuarios, pues en ocasiones, algunos diseñadores intentan inventar y utilizar nuevos modos de interacción, ya que creen que al hacer eso, harán que una aplicación sea más deseable para los usuarios potenciales, pero en realidad, aumentan la curva de aprendizaje, haciendo que los usuarios tengan que invertir más tiempo en aprender a usar la aplicación, razón por la cual, cuando se trata de diseño de interfaces de AR, siempre es mejor no "reinventar la rueda" y en su lugar, se debe aprovechar el conocimiento existente de los usuarios, ya que la mayoría de ellos están habituados al uso de dispositivos móviles y por lo tanto saben cómo tocar, arrastrar y deslizar objetos, dando lugar a que esos patrones de interacción puedan ser utilizados en la interfaz de usuario creada y de ese modo no se tendrá que enseñar al usuario una forma completamente nueva de realizar tareas simples.

### 1.3. APLICACIONES DE REALIDAD AUMENTADA EN MUSEOGRAFÍA.

Como se mencionó anteriormente, la AR es una tecnología que superpone capas de contenido virtual en el mundo real, lo cual permite a los usuarios de dispositivos móviles (tablets, smartphones, etc.) apuntarlos hacia un punto en específico de modo que una escena o exhibición inmóvil puede cobrar vida y es debido al uso tan extendido de dispositivos móviles, que se ha propiciado que el público en general se esté familiarizando cada vez más con aplicaciones de AR en distintos ámbitos.[97]

Un lugar donde una versión de realidad aumentada de baja tecnología lleva mucho tiempo en el mercado son los museos: los primeros usos de audioguías turísticas datan de 1952 en el museo *Stedlijk Museum* en Ámsterdam, mientras que en el *Natural History Museum* de Estados Unidos esta tecnología fue adoptada en 1961 y posteriormente también fue adoptada en 1970 en el *Louvre Museum*. [98] En la actualidad es bastante común que los museos ofrezcan audioguías turísticas que los visitantes llevan consigo mientras recorren las exhibiciones y si bien esta tecnología funciona razonablemente bien, muchas personas se frustraban al usar las primeras versiones de esta tecnología, pues éstas obstruían sus propósitos de socializar con amigos o familiares al asistir al museo, ya que al igual que las personas prefieren ir a un concierto por la experiencia (acústica y social, principalmente), en vez de escuchar la música en su casa, parte de la razón por la que mucha gente va a museos es socializar, estar con amigos y discutir la exhibición tal como la experimentan, lo cual entraba en conflicto con las antiguas audioguías turísticas, pues éstas eran lineales, planificadas de antemano para recorrer el museo en algún orden determinado e ir a cierto ritmo, de modo que era difícil convivir con los demás al mismo tiempo, pues si una persona apagaba la cinta temporalmente, era muy difícil que se pudiera volver a sincronizar y adicionalmente, los recorridos grabados solían describir sólo un pequeño subconjunto de las piezas expuestas y éstas podían no ser las que un visitante en particular estaba interesado en escuchar y debido a la linealidad de la cinta era imposible omitir o acceder a descripciones fuera del orden predeterminado. [99] Dos de estos primeros dispositivos utilizados como audioguía se muestran en la Figura 69.



Figura 69: Dos de los primeros dispositivos utilizados como audioguía en museos.

A pesar de que actualmente hay cada vez más tecnologías incorporándose en los museos, la mayoría del tiempo éstas se centran más en proporcionar información, que en la experiencia interactiva; como se mencionó anteriormente, con el uso de dispositivos de narración que transmiten información relevante, pero la experiencia podría resultar bastante aburrida para los visitantes, debido a que la narración podría ser lenta o la explicación con palabras no podría reemplazar un equivalente visual o interactivo, es por ello que la AR resulta una propuesta muy atractiva para aumentar el interés de las personas por los museos y sus exhibiciones.

En el artículo presentado por Benderson en 1995, se presentó un prototipo de audioguía tecnológicamente más avanzada que sus predecesoras, pues el sistema presentado reemplazaba las cintas de audio analógicas con audio digital de acceso aleatorio, mediante la incorporación de un microcontrolador y un dispositivo de localización espacial, que otorgaba mucha más libertad para el visitante, al permitir que éstos escucharan las descripciones de las piezas simplemente al caminar hacia ellas, de modo que los audios se podían escuchar en cualquier orden e interrumpir alejándose de los objetos, de esta manera las personas que visitaban juntos un museo podrían mantenerse sincronizados de forma muy natural, únicamente al caminar hacia una pieza al mismo tiempo y al alejarse de una pieza, la descripción se detendría y les daría oportunidad de discutir la exhibición o simplemente hablar entre ellos, como muestra el diagrama de la Figura 70.[99]

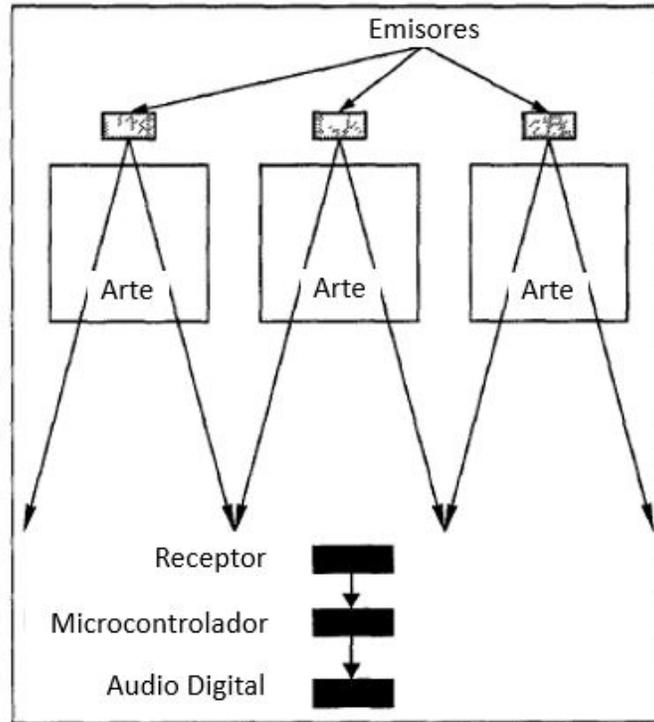


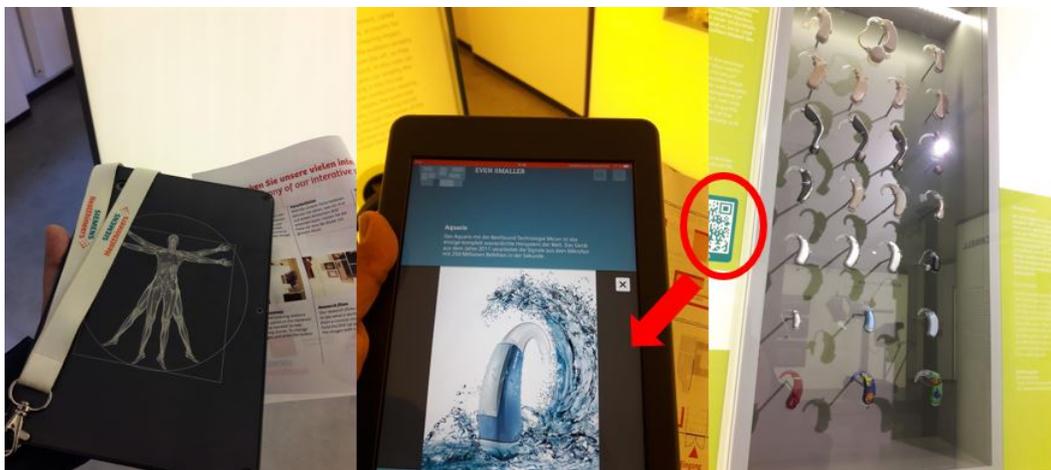
Figura 70: Diagrama que muestra el funcionamiento del prototipo propuesto por Benderson en 1995 para modernizar las audioguías en los museos.[99] Imagen traducida por el autor

Aunque en la actualidad algunos museos siguen ofreciendo audioguías antiguas, muchos se han modernizado al adoptar variantes del prototipo propuesto por Benderson en 1995, por ejemplo mediante dispositivos que hacen exactamente lo que el prototipo propuesto o al utilizar iPads u otros dispositivos para brindar experiencias de AR como las conocemos actualmente; un ejemplo del primer tipo de dispositivos mencionados, son los utilizados en los museos: *Mercedes-Benz Museum* y *Porsche Museum* en Stuttgart, Alemania, pues al iniciar se le proporciona al usuario un dispositivo y se le da la indicación de acercarlo a marcadores dentro de la exhibición para activarlo y acceder a información adicional, donde al visitante se le sugiere un orden de recorrido, pero realmente es libre de trazar su ruta, pues el dispositivo se activa según la ubicación del usuario en el museo, como se observa en la Figura 71.



**Figura 71:** A la izquierda se muestra el dispositivo de audioguía y marcadores usados en el *Mercedes-Benz Museum* y a la derecha se muestra el dispositivo de audioguía y marcadores utilizados en el *Porsche Museum*.

Otro ejemplo de guía de turistas utilizando AR es la tecnología utilizada en el *Siemens Healthineers Med Museum*, en Erlangen, Alemania, en el cual se le presta una iPad a los visitantes y éstos pueden acceder a explicaciones más detalladas, animaciones en 3D e incluso imágenes panorámicas de las invenciones que han revolucionado el mundo de la medicina utilizando marcadores y códigos QR (Figura 72), que explican la evolución desde la terapia con electricidad y el descubrimiento de los Rayos X, hasta los más avanzados tomógrafos y equipos de última generación, además de mucho equipamiento médico que se ha desarrollado con ayuda de ingenieros a lo largo de los años, tanto para diagnóstico, como para tratamiento, por ejemplo los dispositivos de ayuda auditiva y prótesis dentales.



**Figura 72:** Imágenes del dispositivo utilizado en el *Siemens Healthineers Med Museum* para la visualización de explicaciones, animaciones e imágenes panorámicas de instalaciones hospitalarias de última generación.

Otro tipo de audioguía mejorada fue la presentada en 2013 por parte de Nintendo y el Museo del Louvre de París,

Francia, en el cual mediante la videoconsola Nintendo 3DS se pudo profundizar en la cultura y la época de cada obra de manera más interactiva (Figura 73), dicho sistema incluía más de 700 comentarios y fotografías, además de comentarios breves y datos básicos para entender mejor las obras, igualmente, incluía circuitos a través del museo en función de los gustos del visitante: desde el de "Obras Maestras" hasta el de "Egipto para toda la familia", todos ellos de una duración aproximada de 45 minutos. Otra de las ventajas de utilizar una Nintendo 3DS, fue que al disponer de audio, vídeo y texto, permitía que las personas con discapacidades físicas, como la auditiva, pudieran tener una mejor experiencia en la visita gracias al lenguaje de signos en francés y debido a que El Louvre es gigantesco, no es raro encontrar turistas completamente perdidos intentando ubicarse entre los cientos de pasillos, por lo cual el software incluía también un sistema de localización que permitía saber la posición exacta del usuario en cada momento en un mapa. [100]

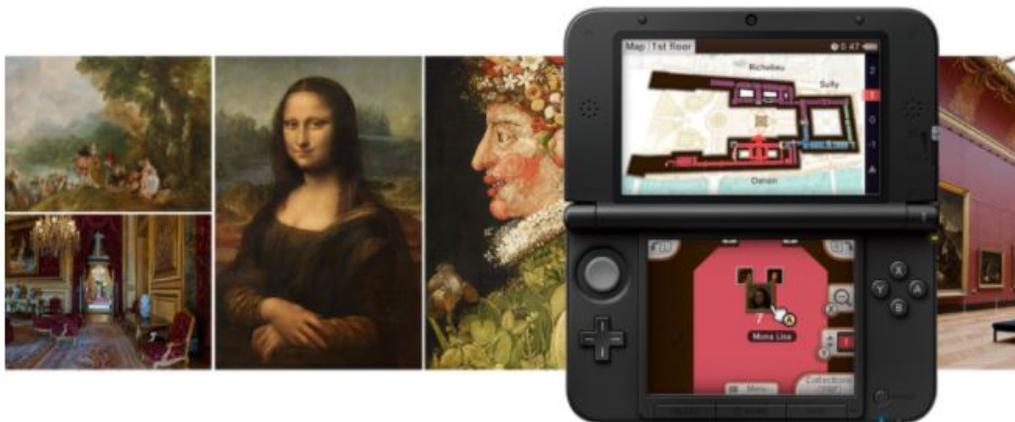


Figura 73: Imagen promocional de la audioguía oficial de Nintendo para el Museo Louvre.

El atractivo de la RA en museos es bastante claro: la tecnología permite que el usuario obtenga contenido multimedia enriquecido, como gráficos, animaciones y videos que se superponen sobre las exhibiciones reales, lo cual proporciona una forma para que los museos den vida a las exhibiciones y de esta manera haya un mayor interés en acudir a ellos y que los visitantes tengan una mejor experiencia en su visita, con más aprendizaje y más interactividad.

A continuación se mencionarán varios museos o proyectos culturales que utilizan diferentes técnicas de AR, llevando la experiencia de la visita más allá de las variantes del prototipo de Benderson presentadas anteriormente, comenzando con uno de los primeros proyectos de AR en dicho contexto fue realizado en 2010 por el *Stedelijk Museum* de Ámsterdam, un museo internacional de arte y diseño moderno y contemporáneo; el proyecto llamado "ARTours", fue financiado por el Ministerio de Cultura Holandés y dirigido en colaboración con la oficina de diseño *Fabrique*, con este proyecto se buscaba un método nuevo e innovador para exhibir la colección del museo, ya que los organizadores querían explorar algunas formas adicionales de interpretar sus elementos y mostrar sus historias, al mismo tiempo que se querían encontrar nuevas formas de involucrar a las personas con su colección, fuera del entorno de un museo tradicional, de modo que se desarrolló un evento que tuvo lugar como parte del proyecto *ARTours* para el popular festival de música holandés *Lowlands*, este concepto se llamó *ARtotheque* y tuvo como objetivo "prestar" piezas de la colección del museo al público en general, lo cual permitió a los asistentes del festival tomar prestada una versión en AR de una obra de las 175 obras de arte del *Stedelijk Museum*, entre los cuales destacan: Picasso, Karel Appel, Andy Warhol, etc. y posteriormente podían colocarlo en algún lugar del sitio del festival, donde otros también podían disfrutarlo utilizando su dispositivo, como se muestra en la Figura 74.[101][102]



**Figura 74:** Imagen del festival *Lowlands* en el cual se "prestaron" obras de arte del *Stedelijk Museum* en AR para que el público las colocara en algún sitio del festival.[102]

Otra experiencia de AR inaugurada en 2010, fue el "WeARinMoMA", llevado a cabo en el Museo de Arte Moderno de Nueva York (MoMA NY, por sus siglas en inglés), el espacio físico de este museo albergó una exhibición virtual que ocupó todos los pisos (incluido un séptimo piso completamente virtual), al mismo tiempo que la exhibición en curso. El espectáculo no fue visible para los visitantes habituales del MoMA, pero aquellos que utilizaron la aplicación "Layar Augmented Reality browser" para smartphone, fueron capaces de ver obras adicionales (2D o 3D) en cada uno de los pisos, colocados allí utilizando una técnica de AR basada en la ubicación, como se muestra en la Figura 75.[103]



Figura 75: Imagen de la exhibición "WeARinMoMa" del Museo de Arte Moderno de Nueva York.[104]

Durante 2017, en el Instituto Franklin, en Estados Unidos, tuvo lugar una exhibición en la cual los visitantes podían utilizar la aplicación de dicho instituto para ver representaciones de realidad aumentada y comprender mejor el misterio del Ejército de Terracota. La exhibición mejorada por AR incluía representaciones de cómo se predice que las esculturas, armas y artefactos aparecieron hace más de dos mil años. Una forma excelente para que los usuarios exploren el deterioro, el descubrimiento, la arqueología, la historia y la preservación del contenido que se encuentra en el complejo funerario del primer emperador de China, Qin Shihuangdi, todo esto utilizando tecnología SLAM (Figura 76).[105]



Figura 76: Imagen de la exhibición "Terracota Warriors AR".[105]

En 2018 hubo una innovadora exposición de arte de robots (*Artistes & Robots*) en el Grand Palais de París, la cual incluyó características de realidad aumentada a su aplicación de guía para visitantes para proporcionar contenido relevante de una manera inmersiva y efectiva. Al escanear una instalación de arte en sí o el catálogo de la exposición, los usuarios obtienen acceso a información detallada sobre los artistas, sus obras, la exposición y más (Figura 77), todo esto se logró utilizando detección de marcadores.

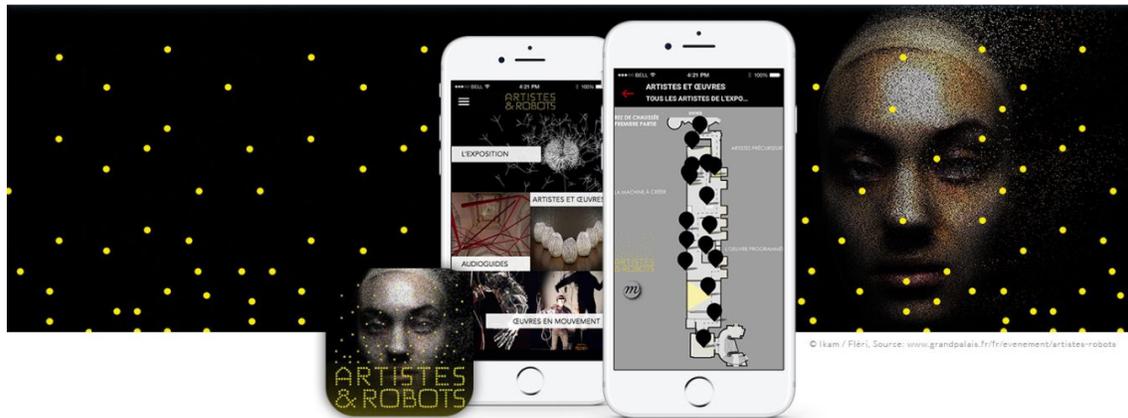


Figura 77: Imagen de la exhibición "Artistes & Robots".[105]

Otros dos proyectos nacionales de AR basados en la detección de marcadores (marcadores personalizados y códigos QR), son: "Realidad aumentada como un recurso educativo para el patrimonio cultural" e "Hilvanando cartografías" de la Universidad Autónoma de la Ciudad de México; en el primero se ofrecía un cuadernillo con las instrucciones para utilizar la app y los marcadores, con una breve explicación de los colores utilizados en el modelo de AR y en el segundo se utilizaban códigos QR para mostrar entrevistas en vídeo e iconografía acerca del tema, como se muestra en las Figuras 78 y 79.



Figura 78: A la izquierda se observa el cuadernillo con marcadores y a la derecha uno de los modelos 3D en AR del proyecto "Realidad Aumentada como un recurso educativo para el patrimonio cultural"[106]



Figura 79: A la izquierda se observa uno de los modelos en 3D AR y a la derecha el mapa del proyecto "Hilvanando cartografías"[106][107]

Por otro lado, un museo que actualmente tiene dos experiencias de AR es *ABBA The Museum*, éste es un museo dedicado a mostrar los orígenes, evolución, logros, etc. de la que es considerada una de las mejores agrupaciones musicales a nivel mundial, colocados en el número 13 de artistas con mayor número de ventas en el mundo. La primera de sus experiencias AR utiliza la ilusión de *Pepper's Ghost* y permite que los visitantes canten con el grupo en vivo, como se muestra en la Figura 80.



Figura 80: Imagen del ABBA The Museum utilizando la ilusión de "Pepper's Ghost" para hacer que los visitantes hagan karaoke con la banda.[51][108]

Y la segunda de sus experiencias AR está basada en su musical *Mamma Mía*; en esta actividad se debe ayudar a la protagonista del musical a pintar una pared de su descuidado hotel, dicha actividad está titulada "Help Donna fix the house" y se puede observar en la Figura 81. La tecnología usada en esta experiencia es similar a la de "Videoplace", mencionado anteriormente, con la diferencia que éste utiliza un sensor Kinect para saber que el usuario ha tomado la brocha e interpreta los movimientos para ir proyectando color por donde se pasa la brocha y así el usuario siente que de verdad está pintando, pero una vez que el usuario se retira, la actividad se reinicia para el siguiente visitante.[109]



Figura 81: Imagen del ABBA The Museum durante la actividad "Help Donna fix the house", para que los visitantes pinten virtualmente.[109]

Otro museo que utiliza una experiencia AR utilizando la ilusión de "Pepper's Ghost" es el *Deutsches Fußball Museum* o Museo de fútbol alemán, con su atracción *3D Kino* (Figura 81), en la cual Leroy Sané, del Bayern Múnich, hace un viaje a través de la historia del fútbol alemán, en el cual se detiene en Berna, Munich, Roma y Río de Janeiro; también aparecen Paul Breitner, Philipp Lahm, Joachim Löw y Sepp Herberger, con lo cual logran revivir los momentos clave de los cuatro éxitos mundiales de 1954, 1974, 1990 y 2014, todo esto en un tono muy motivacional.[110]



**Figura 82:** Imagen del *Deutsches Fußball Museum* en su atracción "3D Kino" en la cual utilizan la ilusión de "Pepper's Ghost" para dar vida a los más famosos jugadores de fútbol de Alemania.[110]

Por otro lado, existe una plataforma llamada Broadcast AR, la cual consiste en una o varias cámaras y una pantalla gigante, lo cual permite utilizarla como si fuera un espejo, de tal manera que los objetos virtuales se renderizan sobre la imagen proveniente de la cámara, de tal forma que los usuarios se ven interactuando con elementos virtuales en la pantalla, con lo que se obtiene un tipo de AR que ha sido utilizado en centros comerciales, para promocionar películas, pero también en museos de ciencias naturales, pues resulta muy atractivo para los visitantes el poder ver a los dinosaurios caminar frente a ellos, esta tecnología se ha utilizado en específico en el *Smithsonian National Museum of Natural History* en Estados Unidos y en el *Tekniska Museet (Swedish National Museum of Science and Technology)* en Suecia, como se muestra en las Figuras 83 y 84[111].



Figura 83: Imagen del *Smithsonian National Museum of Natural History* en la exhibición de dinosaurios utilizando Broadcast AR.[112]

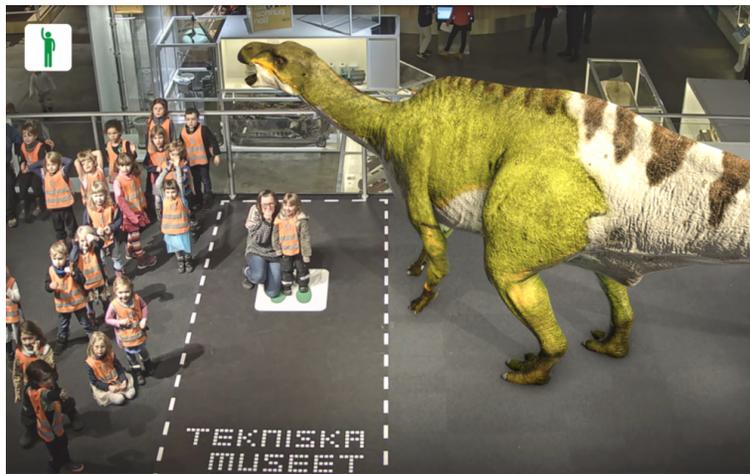


Figura 84: Imagen del *Tekniska Museet (Swedish National Museum of Science and Technology)* en la exhibición de dinosaurios utilizando Broadcast AR.[111]

A pesar de que muchos más museos han utilizado técnicas de AR, sería muy complejo citarlos a todos en el presente trabajo, por lo cual el último ejemplo de AR en museografía es de un museo dedicado a la historia de uno de los artistas más reconocidos y controversiales de Alemania: Udo Lindenberg, quien antes de la caída del muro de Berlín fue censurado en Alemania Oriental por su activismo y sus canciones críticas al gobierno comunista.[113] Este museo, "Panik City" está situado en el 20 lugar de las mejores atracciones turísticas de Hamburgo[114], debido a que es completamente innovador: se ofrece un recorrido de 90 minutos en el cual el artista, de manera virtual, mediante una enorme cantidad de pantallas va narrándole su vida a los visitantes y esa parte del diseño de la experiencia del museo se vuelve muy entretenida, convirtiéndose así en una técnica de *storytelling* bastante interactiva.[115] Durante el recorrido hay una sección para que los visitantes hagan pinturas virtuales mediante pantallas *touch*, que asemejan la técnica "Likörelle" inventada por el artista, las cuales posteriormente pueden ser descargadas y más adelante se encuentra una sala de exhibición de objetos históricos AR,

donde se prestan iPads para ver información y animaciones de dichos objetos, entre los cuales se encuentra un Trabant (el auto típico de la Alemania Oriental) bañado en oro y otros objetos que tuvieron gran impacto político, como documentos de espionaje por parte de la *Stasi*, en otra de las salas se encuentra una experiencia VR para que el visitante viva desde el escenario algunas de las canciones más famosas en un concierto y finalmente hay una sección donde se pone a que los visitantes canten con una versión en AR del artista, la cual los visitantes también pueden descargar al terminar la visita, como se muestra en las Figuras 85 y 86.



Figura 85: Imagen del museo *Panik City*, del lado izquierdo la sala AR con el Trabant y del lado derecho la sala de VR.[116][117]



Figura 86: Imagen del museo *Panik City*, del lado izquierdo la sala de pintura virtual y del lado derecho la sala de karaoke AR con el artista señalado con una flecha roja.[116]

#### 1.4. EL CÓDICE MAYA DE MÉXICO.

El códice Maya de México, antes conocido como "Códice Grolier" fue descubierto en México en la década de los años 60 por saqueadores, que lo extrajeron ilegalmente de su contexto arqueológico en la Sierra de Chiapas y posteriormente se lo ofrecieron a un coleccionista mexicano, el Dr. José Sáenz, quien más tarde, en 1971 le permitió al arqueólogo Michael Coe mostrarlo en el Club Grolier en la Ciudad de Nueva York, para la exhibición "Ancient Maya Calligraphy", es por esta razón por la cual anteriormente se le conocía como "Códice Grolier", así como a los códices Madrid, París y Dresde se les conoce por el nombre de su lugar de exhibición.[118]

Posteriormente, en abril de 1971, el periódico *The New York Times* publicó la noticia con una fotografía de las páginas 4 - 7 del códice, comenzando así la polémica de su autenticidad, y más tarde, en 1974, Sáenz le entregó el códice al entonces Director del Museo Nacional de Antropología e Historia: Ignacio Bernal, para que se le realizara un peritaje y así establecer su autenticidad, con la promesa de que si el peritaje resultaba ser positivo, el códice sería donado a la nación por parte de él (José Sáenz) y su esposa. [118]

El códice ha sido objeto tanto de apologías como de ataques a lo largo de los años, debido a la polémica de su origen [119], lo cual provocó que durante más de 50 años éste haya sido sujeto de muchos estudios para comprobar su autenticidad, sin que ninguno llegara a ser concluyente. Inicialmente se pensaba que el códice constaba de once páginas, ilustradas con el calendario de Venus, pero en 1982, el arqueoastrónomo John Carlson apoyó su autenticidad y demostró que las páginas 10 y 11 eran una misma página, como se muestra en la Figura 87 [118].

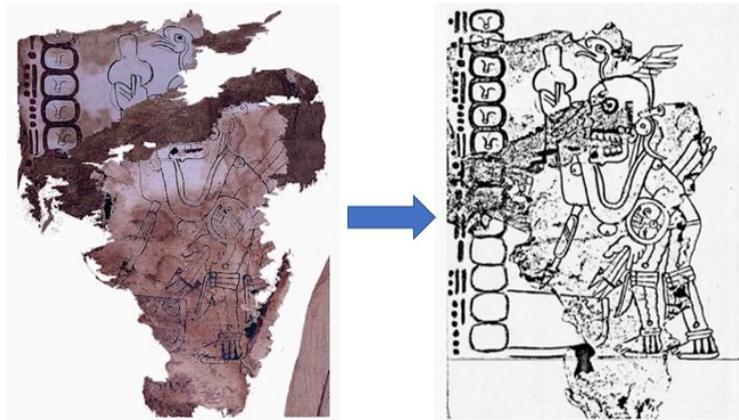


Figura 87: Ilustración de que los folios 10 y 11 del códice son uno mismo.[118]

Actualmente se sabe que el "Códice Maya de México" contaba en su origen con veinte folios con las cuatro fases de

Venus, pero en el contexto arqueológico se perdieron los ocho primeros y los dos últimos; en los folios restantes, que han sido numerados del 1 al 10 aparecen los dioses que personifican a Venus, que están asociados a las fases del planeta y al sacrificio ritual de los cautivos en el día que indica la columna de jeroglíficos, mientras que los números de distancia en la parte superior, marcan los días que Venus permanece en cada fase[120], como se muestra en la Figura 88.

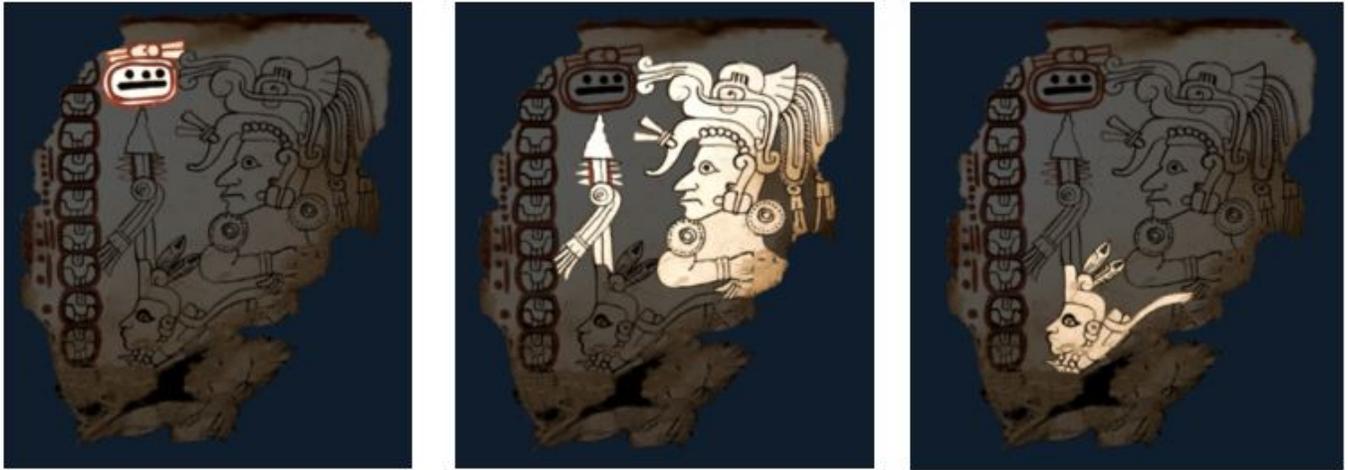


Figura 88: A la izquierda se muestra resaltada la fase de Venus, al centro la deidad y a la derecha la víctima.[120]

En 2016 Michael Coe y su equipo reafirmaron la autenticidad del código, basados en su iconografía, estilo y contenido ritual, en octubre de ese mismo año el Instituto Nacional de Antropología e Historia (INAH) conformó un equipo de especialistas de la comunidad científica y académica para examinar el código de manera exhaustiva, de todos los ángulos posibles y los resultados fueron contundentes: los diversos estudios de distintas disciplinas (exámenes de datación, materiales orgánicos e inorgánicos, técnicas de factura, entomología, iconografía, microscopía, fotografía en el espectro visible e invisible, caracterización químico-mineralógica, morfometría, cronología, astronomía, estilo y simbolismo, entre otros) concluyeron que el código era auténtico, de modo que ahora se tiene la certeza de que es el documento prehispánico legible más antiguo de América, pues únicamente en el área mesoamericana se produjeron libros o pliegos de papel con diversos fines y son pocas las obras de este tipo que han sobrevivido al paso del tiempo, ya que muchas de ellas fueron destruidas en el siglo XVI por un auto de fe encabezado por fray Diego de Landa, quien las consideraba heréticas o incluso se perdieron en épocas prehispánicas dentro de las guerras de antaño[118].

Actualmente solo se conocen tres códigos prehispánicos más, denominados Madrid, Dresde y París, por las ciudades donde se resguardan, como se mencionó anteriormente y "El Código Maya de México" es el cuarto código, pero dada su antigüedad bien podría ser reconocido como el primero[121] y en la Figura 89 se muestra una fotografía de todos los folios

de éste.

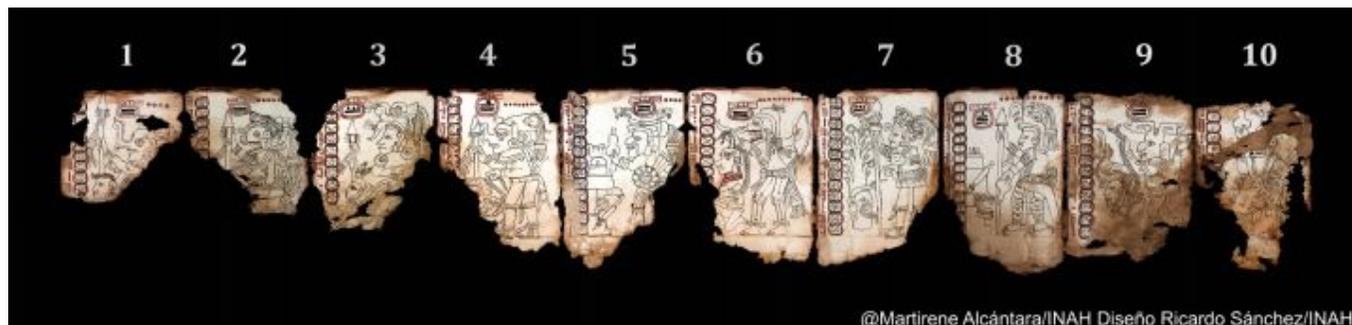


Figura 89: Folios preservados del Códice Maya de México.[122]

En el marco de la Feria Internacional del Libro de Antropología e Historia fue presentado en la exposición temporal titulada "El Códice Maya de México (antes Grolier) Eslabón, Fuente y Testigo" del 27 de septiembre al 28 de octubre de 2018, para lo cual se requirió la elaboración de una cápsula especial para fomentar su preservación, la cual fue creada por académicos de la UNAM. La cápsula que contuvo el documento es una cámara anóxica (sin oxígeno) para la preservación de documentos fue hecha de policarbonato y empaques de viton, con lo cual se aisló el códice, que es sumamente frágil y delicado, del medio ambiente, y se le brindó un ambiente que lo protege de las cambiantes condiciones externas. Para la creación de la cápsula se utilizó un sistema de doble cámara: una de vacío para generar el cierre de toda la cápsula y una de generación de presión para provocar una atmósfera adecuada en temperatura y humedad, utilizando argón en lugar de oxígeno, para evitar el desarrollo de microorganismos y bacterias[118], lo cual permite mantener las características del documento del siglo XII; para su diseño se tomaron en cuenta las dimensiones del códice, estado de conservación y la museografía, de forma que no sólo se logró su resguardo de la mejor manera, sino que también es acorde con la estética y los visitantes pudieron apreciarlo de cerca. [123] En la Figura 90 se muestra una fotografía de la cápsula en construcción y en la Figura 91 se muestra cómo lucía el códice ya expuesto en el museo.



**Figura 90:** Construcción de la cápsula anóxica del Códice Maya de México.[123]



**Figura 91:** Códice Maya de México expuesto en la cápsula anóxica durante su exhibición temporal en el Museo Nacional de Antropología.[124]

La museografía de la exhibición contó con cinco módulos; en el primer módulo se dio una explicación de qué son y la importancia de los códices, en la segunda sección se abordó la gran polémica acerca del códice, brevemente mencionada anteriormente, por otro lado el tercer módulo tocó el tema del contexto histórico del códice (Posclásico Temprano (900-1200 d. C.)), para que los visitantes se familiarizaran y de ese modo pudieran apreciar el estilo maya - tolteca del códice, mientras que la cuarta sección se centró en los aspectos cosmogónicos, rasgos estilísticos y los significados de las ilustraciones, mediante el uso de folios animados de gran formato y tabletas electrónicas para que los visitantes pudieran consultar de manera específica la identidad de las deidades, los elementos iconográficos y las acciones representadas en cada folio, fi-

nalmente, en el quinto módulo se dieron a conocer a detalle los estudios que expertos del INAH, la Universidad Nacional Autónoma de México y la Universidad de Colorado, en Boulder, practicaron al códice[125] y adicionalmente se habilitó una experiencia de AR la cual funcionaba al reconocer marcadores colocados en la exhibición y se brindaba a los visitantes información de la estática acerca del folio reconocido en el marcador.

Durante la exhibición del Códice en el Museo Nacional de Antropología e Historia, se tuvo un registro de 100,415 visitantes[125], lo cual daría un promedio de 3,138 visitantes al día, si no se considerara el hecho de que este tipo de eventos tienen una afluencia mucho mayor los fines de semana, comparado con los demás días de la semana; por otro lado, adicional a la cápsula anóxica, se colocó una cinta de tipo unifila para que la gente no se acercara demasiado al Códice, debido a que cualquier cambio en el ambiente o iluminación podría dañarlo[126], sin embargo, a pesar de los esfuerzos hechos por el INAH, las personas se aglomeraban y no respetaban la distancia de seguridad, pues querían estar más cerca del códice, como se observa en la Figura 92, dadas estas situaciones surge la oportunidad de crear una experiencia de usuario para apoyar a la exhibición y darle a las personas la oportunidad de estar tan cerca del códice como si pudieran tocarlo.



Figura 92: Exposición "El Códice Maya de México (antes Grolier) Eslabón, Fuente y Testigo"[127]



*AR is an attractive medium for use in museums because digital databases challenge existing archives with obsolescence, and the ever-growing tide of digital information can be reconciled with traditional, physical databases through the promise of AR.*

Geoffrey Alan Rhodes, Cineasta

# 2

## Justificación

Teniendo en cuenta el marco teórico anteriormente presentado, se puede llegar a la conclusión de que la realidad aumentada (AR), combinada con los principios y buenas prácticas para el diseño de interfaces, provenientes de la HCI, es un recurso de suma utilidad para incrementar la interactividad en distintos ámbitos, como lo son educación, entretenimiento, manufactura, difusión cultural, etc., y sobre todo en museos, pues resulta ser un medio muy atractivo debido a que desde la creación de bases de datos digitales, éstas han estado desafiando a los archivos físicos existentes, dejándolos obsoletos, y la creciente cantidad de información digital puede ser reconciliada con las bases de datos físicas tradicionales a través de la promesa de la AR[97]. Por otro lado, de la exposición del Códice Maya de México en el Museo Nacional de Antropología, se pudo observar que, al ser una exhibición de muy corta duración, ésta fue muy concurrida, al mismo tiempo que los numerosos visitantes se sentían bastante atraídos por el códice, razón por la cual querían estar más cerca de él, sin embargo eso no fue posible debido a su estado de conservación y para prevenir mayor deterioro del mismo, pero a pesar de ello, muchos

de los visitantes violaban la distancia de seguridad, como se mostró en la Figura 92.

Por otro lado, teniendo en cuenta que en la museografía de la exposición sí se tuvo contemplado el tener una experiencia de AR para incrementar la interactividad de la exhibición, pero ésta no cumplía con el deseo de los visitantes de tener una mayor cercanía al códice al punto de incluso poder tocarlo, pues ésta únicamente detectaba marcadores colocados a lo largo de la exhibición y mostraba información estática sobre los folios, se identifica la necesidad de crear una experiencia de AR que aumente el nivel de interacción que tienen los usuarios con el códice, de tal modo que se pueda lograr incluso un mayor interés por él y un mayor aprendizaje, satisfaciendo el deseo de los usuarios de estar más cerca del códice, para poder apreciarlo con mayor detalle o para acceder únicamente a información específica del códice, sin la necesidad de tener que buscar las partes de interés entre toda la información que el museógrafo o el curador de la exhibición pueda estar mostrando, algo similar a lo mencionado anteriormente acerca de las antiguas audioguías, que resultaban en ocasiones desesperantes para el usuario, pues no permitían saltar información y se tenía que seguir cierto orden predeterminado.

Considerando lo anterior, para la creación de una experiencia de AR para el fin descrito en los párrafos anteriores, es decir, como recurso interactivo en museos, en este caso en específico para mostrar el Códice Maya de México, se requeriría reunir conocimientos acerca de HCI, entre los cuales se encuentran los principios y buenas prácticas para el diseño de interfaces, junto con conocimientos de programación y digitalización, así como también información acerca del códice, pero principalmente se le debería dar mayor prioridad a seguir la primera de las ocho reglas de oro para el diseño de interfaces propuestas por Schneidermann: *Conocer al usuario*[19].

Y en este caso se va a partir de la premisa de que el usuario quiere tener una mayor cercanía con el códice, incluso al punto de poder tenerlo entre sus manos y de que no todos los usuarios estarán interesados en toda la información disponible, sino que se les dará la libertad de escoger qué información desean recibir.

*Weniger, aber besser.*

*Less, but better.*

Dieter Rams, Diseñador

# 3

## Primera etapa de diseño

### 3.1. IDENTIFICACIÓN DE LAS FUNCIONES.

Como se mencionó en el capítulo anterior, existen varias razones para crear una experiencia AR enfocada en incrementar la interacción entre personas y el Códice Maya de México, quizás la más importante de ellas es que las personas sienten la necesidad de estar más cerca del código de lo permitido para su óptima conservación y partiendo de ese punto y de que no necesariamente los usuarios quieren recibir toda la información o en un orden predeterminado, se empezaron a generar propuestas de diseños conceptuales, variando las funciones que la experiencia AR debería tener e incluso descubriendo nuevas, al experimentar con los conceptos de diseño.

Lo primero que se hizo para poder comenzar a experimentar con la AR fue crear versiones digitales del código, lo cual será descrito a detalle en la sección 3.3 más adelante y una vez teniendo estos modelos digitales, como algo que se tuvo claro

desde el inicio era que, en algún momento de su existencia el código se podía doblar, una de las características principales que se quiso implementar fue el crear una metáfora de que el código se desdoblaba frente al usuario, como se muestra en la Figura 93. Y al mismo tiempo que se generaban las versiones digitales del código, se estuvo leyendo la documentación y probando el funcionamiento del SDK Vuforia para el motor de desarrollo de videojuegos Unity, de modo que terminados los modelos se pudieran comenzar a realizar pruebas.

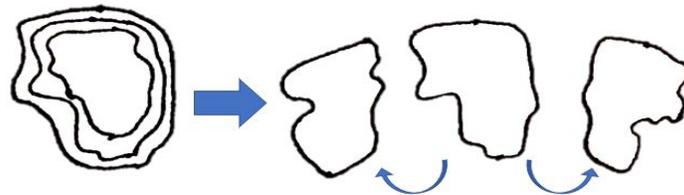


Figura 93: Bosquejo para la creación de la animación del código virtual.

Por otro lado, pensando que las personas utilizarían el diseño muy probablemente en algún museo, pero no limitándose su uso en otro espacio, se propuso utilizar ambas orientaciones de un smartphone: *landscape* (horizontal) y *portrait* (vertical), debido a que posiblemente no resulte cómodo para algunos usuarios tener que utilizar la aplicación con ambas manos en modo *landscape* y para otros podría ser mejor tener una vista amplia del código virtual al usar el modo *portrait*. En la Figura 94 se ilustra el hecho de que las imágenes en *landscape* son más anchas, permitiendo una mayor área de visualización y como se mencionó anteriormente, la ventaja de utilizar un dispositivo en modo *portrait*, recae en la posibilidad de utilizarlo con una sola mano.



Figura 94: Diferencia entre las orientaciones *Portrait* (izquierda) y *Landscape* (derecha)[128].

Como otra de las observaciones desprendidas del capítulo 2 es que no todos los usuarios desean recibir toda la infor-

mación, se pensó en un método utilizando "zonas" interactivas dentro de los modelos del código virtual, de tal modo que el usuario sea capaz de seleccionar por sí mismo las áreas que desea visualizar, como se muestra en la Figura 95. Y con esta función se entra en el terreno de la HCI de nuevo, pues habría que decidir cómo al usuario le gustaría interactuar con la aplicación, es decir, cómo le resultaría natural hacer el ingreso de información, en este caso: cómo sería la interacción con la parte del código que es de interés en ese momento, lo cual será explicado más a fondo en la siguiente sección (3.2).

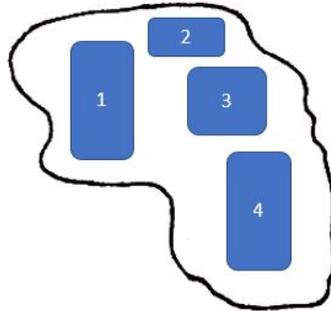


Figura 95: Bosquejo de la ubicación de zonas interactivas en el código virtual.

Por otro lado, al estar probando y experimentando con una de las primeras versiones incompletas de la app, se descubrió, gracias a una observación externa, que la interacción con el código podría verse incrementada si se implementaba la utilización de ambas cámaras: frontal y trasera, con la finalidad de que los usuarios que así lo deseen pudieran tomarse *selfies* sosteniendo el código, por lo cual desde la primera versión completa se implementó el poder cambiar de cámara y por supuesto también el poder capturar fotografías dentro de la experiencia AR y con este último punto, se desprendió la idea de crear una visualización dentro de la aplicación de las fotografías tomadas y la función de compartir dichas tomas.

Entonces, en resumen las funciones detectadas fueron:

- Detección de los marcadores y aparición de los modelos virtuales con su respectiva animación.
- Presencia de áreas interactivas para proporcionar información.
- Utilización en modo *landscape* y *portrait*.
- Toma de fotografías con ambas cámaras (frontal y trasera)
- Galería interna y función para compartir.

### 3.2. INTERACCIONES DEL USUARIO.

Una vez establecidas las funciones que debería tener la experiencia AR, surgieron de manera natural las interacciones que el usuario debería tener con la aplicación: toma de fotografía, cambio de cámara, mostrar fotografía, mostrar información, ocultar información, etc., sin embargo se tenía que definir qué método o métodos de interacción se le permitieran al usuario y tomando en cuenta los fundamentos de HCI y el hecho de que la mayoría de los potenciales visitantes de un museo utilizarían la aplicación en un dispositivo móvil, se decidió que, como el método de interacción más común en este tipo de dispositivos es la utilización de interfaces de usuario touch, éste debería ser uno de los métodos a utilizar.

Por otro lado, como la AR puede ser tomada como una ventana a través de la cual los usuarios pueden expandir su vista del entorno real, se pensó en implementar un tipo de interacción basado en *raycasting* para las interacciones directas con el código; ésta es la técnica para apuntar objetivos más comúnmente utilizada en entornos de realidad virtual[129], en la cual se dispara un rayo invisible desde un punto en una dirección específica, para detectar si hubo colisiones del rayo con algún objeto a su paso. En este caso se decidió que el rayo saldría del centro de la pantalla y este punto estaría marcado con un círculo rojo; esta técnica resulta muy útil en videojuegos como los de disparo en primera persona (FPS, por sus siglas en inglés), disparo en tercera persona (TPS, por sus siglas en inglés), e incluso juegos de aventuras, pues resulta de suma utilidad ser capaz de rastrear hacia dónde va a viajar una bala o un láser de principio a fin, ya que con ello se sabe exactamente cómo se comportará[130]. Para el presente trabajo esto se logró utilizando un cursor de ubicación fija, en este caso el centro de la pantalla, con el cual los usuarios pueden interactuar con las zonas interactivas del código, como se muestra en la Figura 96 y con el cual, una vez apuntada el área interactiva, se debe esperar un tiempo de carga para asegurar que realmente los usuarios quieran obtener la información del área a la cual están apuntando, pues resultaría muy molesto que al menor movimiento la información comenzara a desplegarse sin que los usuarios así lo deseen, para esto se utilizó un marcador gráfico, el cual aparece concéntrico al cursor del centro de la pantalla y se va llenando conforme pasa el tiempo de carga, como se muestra en la Figura 97, hasta que se llena completamente y automáticamente aparece la información solicitada.

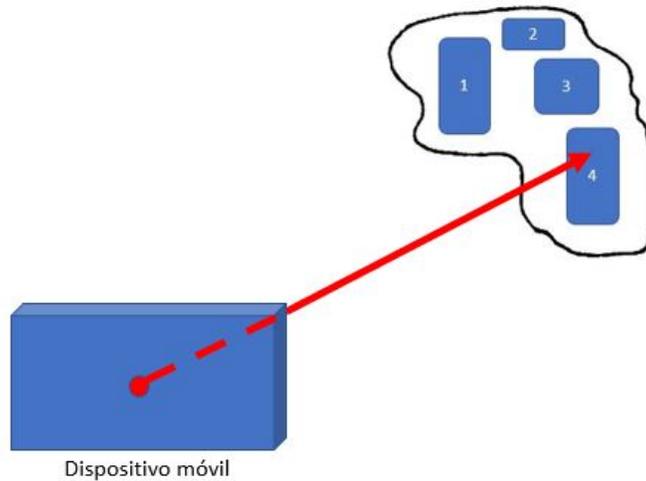


Figura 96: Bosquejo de la función para apuntar las zonas interactivas del código virtual.

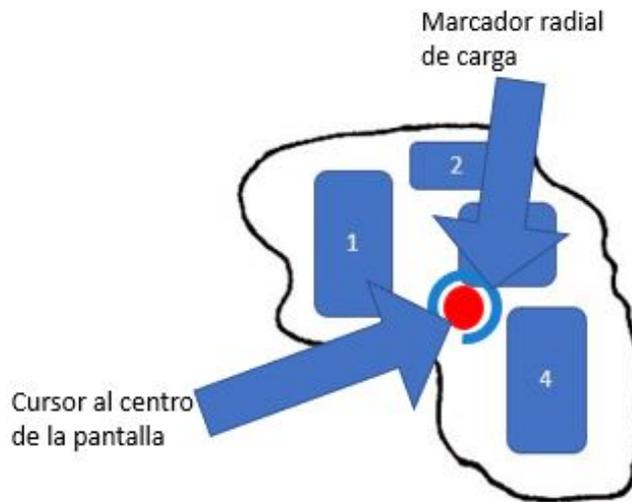


Figura 97: Bosquejo de la función tiempo de carga para después de haber apuntado las zonas interactivas del código virtual.

Por su parte, una función que se desprende de haber establecido la posibilidad de tomar fotografía y *selfie*, fue el poder "apagar" la muestra de información o las zonas interactivas, ya que no sería deseable que los usuarios estén fotografiándose con el código y que repentinamente apareciera un recuadro con información o que éstos tuvieran que estar moviendo el dispositivo o el código con tal de evitar las zonas interactivas al mismo tiempo que tratan de fotografiarse; todo lo anterior se ilustrará más a profundidad en la sección 3.4, más adelante.

### 3.3. GENERACIÓN DE MODELOS VIRTUALES.

Como se tenía claro que el códice virtual debía tener las dimensiones del códice real, se tomaron como base las imágenes contenidas dentro del facsímil del libro "El Códice Maya de México, antes Grolier"[131], debido a que éste contiene fotografías del códice en tamaño real y se utilizó el software Autodesk Inventor para generar un modelo del códice en 3D, haciendo que los trazos tuvieran cierto volumen con respecto a las partes que no contienen información a la cual también se le asignó un volumen mayor que el real, todo esto para poder experimentar más tarde creando modelos mediante manufactura aditiva; para lograr lo anterior se trazó inicialmente el volumen correspondiente al papel y posteriormente se fueron trazando las líneas y áreas correspondientes a los trazos, todo esto utilizando la herramienta *Spline Interpolation* para marcar punto por punto cada área y poderlas extruír posteriormente, como se muestra en la Figura 98, para finalmente obtener las tres piezas mostradas en la Figura 99.

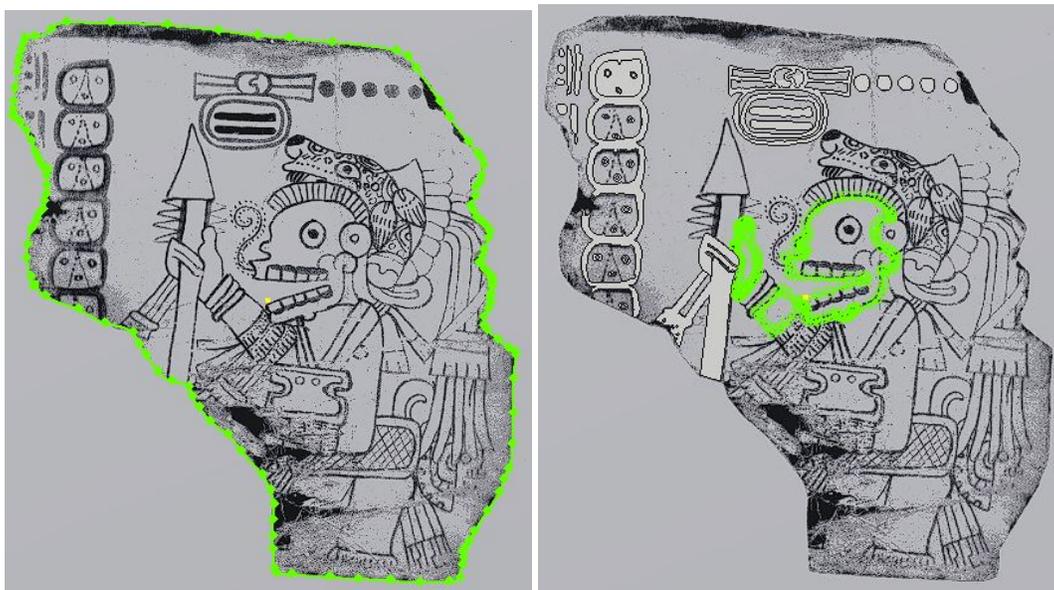


Figura 98: Trazo de las superficies para obtener un modelo 3D de los folios del Códice Maya de México.[131]



Figura 99: Modelos 3D de los folios 2, 3 y 4 del del Códice Maya de México.

Las primeras pruebas que se realizaron en Unity consistían en comprender cómo se realizaba la detección de marcadores 2D utilizando Vuforia, se llevaron a cabo utilizando fotocopias del libro "El Códice Maya de México, antes Grolier"[131] para hacer pruebas de detección de marcadores 2D y poder hacer aparecer modelos 3D dentro de la aplicación; en esas primeras pruebas se utilizaron modelos de frutas y una vez que se logró realizar la aparición de los modelos de frutas, como se muestra en la imagen 100, se procedió a hacer la animación del códice utilizando las herramientas proporcionadas por Unity, tal como el *Unity Animator* y la ventana *Animation*.

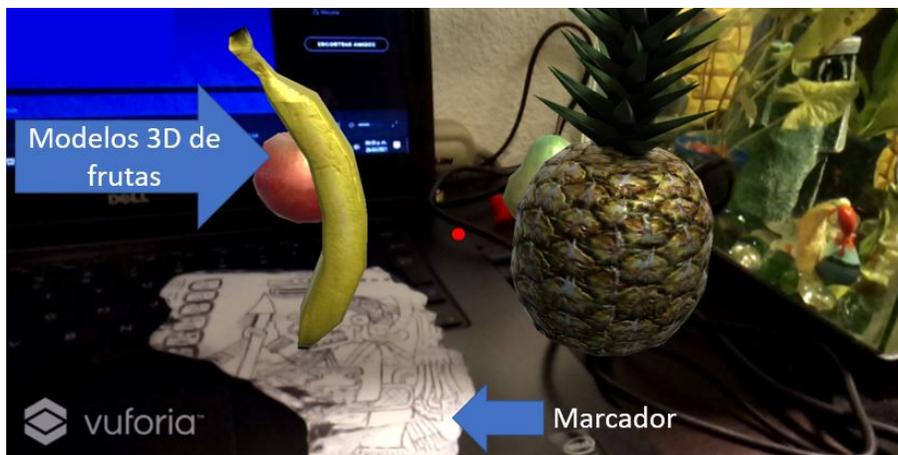


Figura 100: Detección del marcador 2D utilizando Vuforia y Unity.

Para crear las animaciones de apertura de los códigos, se utilizó primero la ventana *Animation* para almacenar las posiciones de los objetos que componen al código a lo largo del tiempo de animación; esto se hace al guardar la posición inicial al principio de la barra de tiempo y posteriormente ir moviendo los objetos hasta su posición deseada y guardarla en el tiempo de animación requerido, como se muestra en la Figura 101, para lograr esto se utilizaron objetos vacíos para contener los códigos, de tal manera que fuera más sencillo realizar la rotación desde una de las esquinas de esos objetos vacíos, para asemejar la apertura de un libro o tríptico en la animación; ésto se puede observar en la parte superior de la Figura 101, ya que en las esquinas inferiores de los códigos se pueden ver los *gizmos* en forma de esferas amarillas, los cuales se utilizaron como puntos fijos para realizar la rotación.

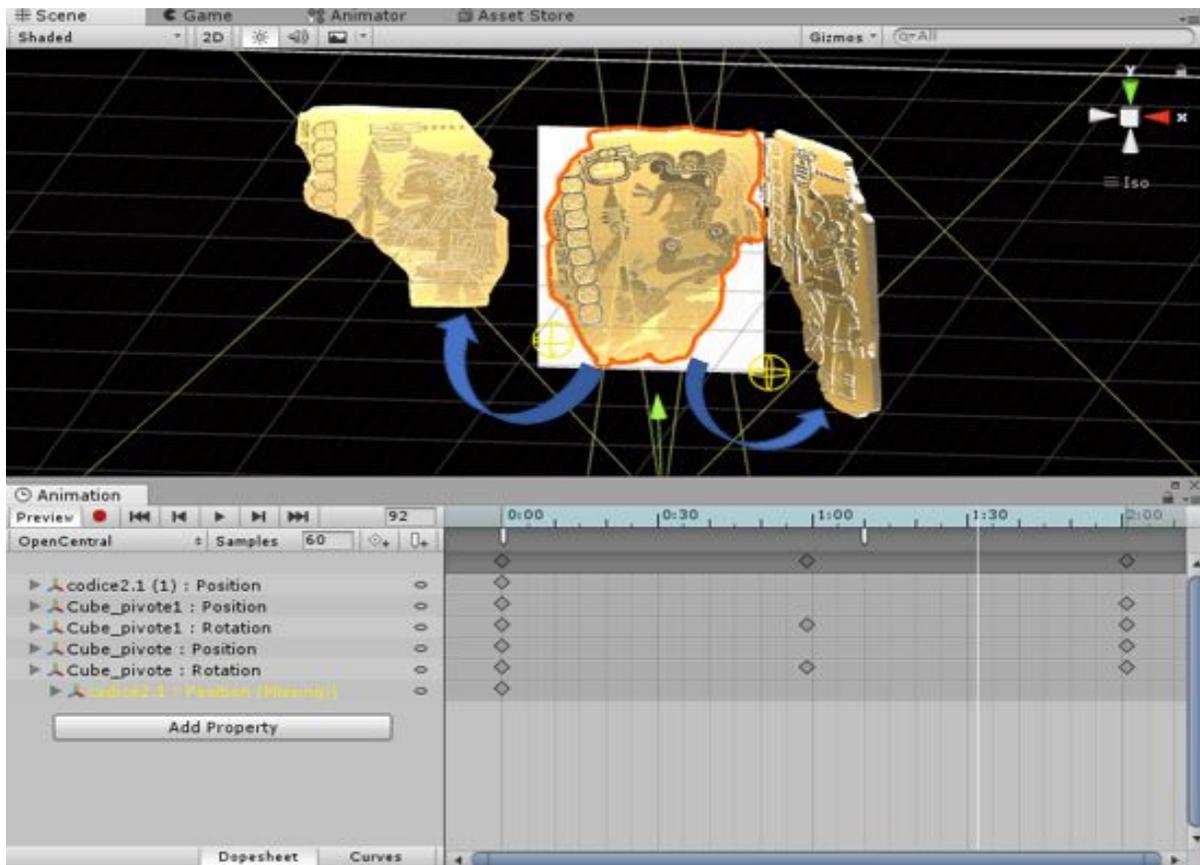


Figura 101: Ventana de animación en Unity.

Y posteriormente se creó un controlador de animación para cada código con la herramienta *Animator*, en el cual se utilizan dos variables (éstas se pueden observar en la columna de la izquierda en la Figura 102) para reproducir la animación una sola vez cuando el marcador es detectado y todo esto se realiza de manera muy gráfica, como se muestra en la Figura 102; lo único que se requiere es programar que en la detección del marcador se modifique la variable "onDetect" del código

detectado, hacer la limpieza de variables cuando ya no hay marcadores siendo detectados y modificar la variable "once" una vez que se ha reproducido una vez la animación, para prevenir que se reproduzca continuamente.

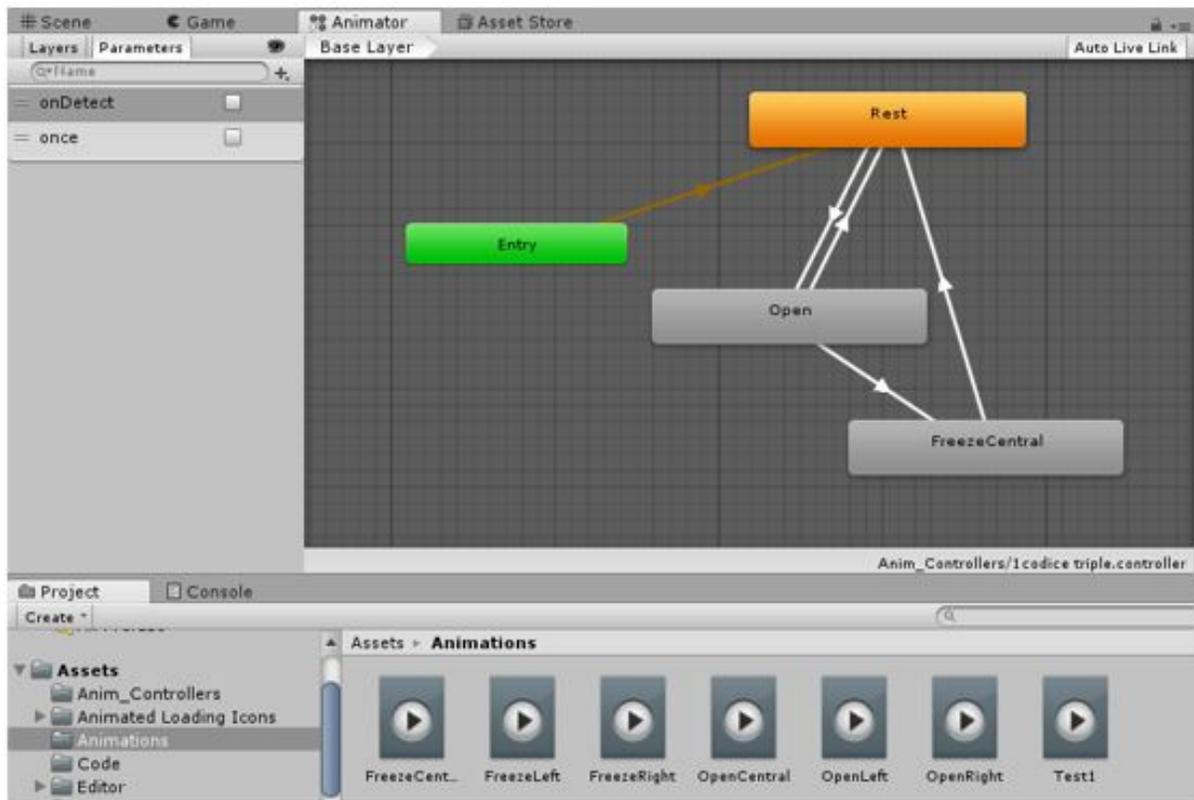
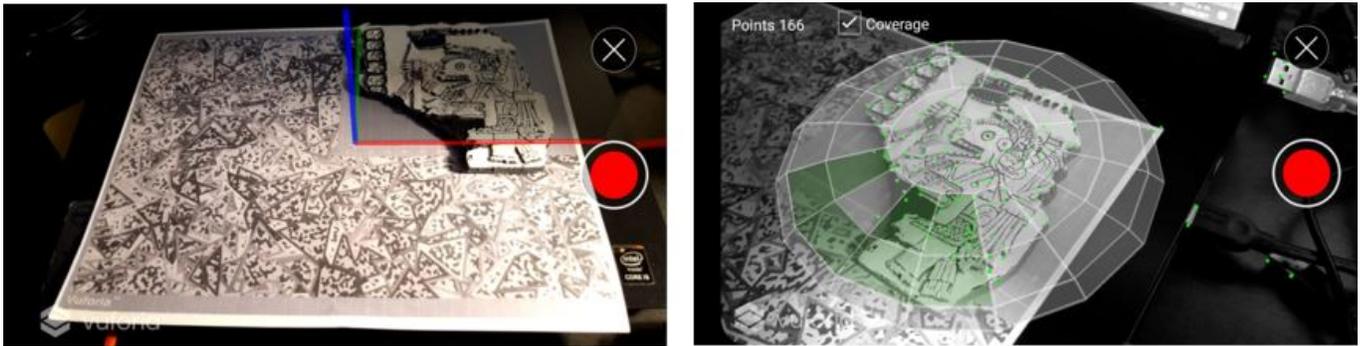


Figura 102: Controlador de animaciones en Unity.

Posteriormente, se experimentó imprimiendo uno de los modelos en 3D con PLA y posteriormente con TPU, como se muestra en la Figura 103 y al modelo en PLA se le hicieron algunas modificaciones para que resaltaran más sus características, posteriormente se utilizó una de las herramientas externas que proporciona Vuforia para hacer el reconocimiento de objetos, utilizando una plantilla imprimible sobre la cual se debe colocar el objeto que se desea escanear y la aplicación "Scanner App", de modo que se escaneó el objeto como se muestra en la Figura 104 y se subió a la base de datos, donde se cargan los marcadores 2D o 3D para posteriormente generar el archivo que se debe importar a Unity, con el fin de hacer la detección de dichos marcadores y programar los demás comportamientos, como se muestra en la Figura 105.



**Figura 103:** Impresiones 3d de uno de los modelos creados. A la izquierda se observa la impresión en TPU, la cual tiene la característica de ser flexible, al centro se muestra una impresión en PLA y a la derecha otra impresión en PLA pero con algunos de los detalles resaltados con pintura.



**Figura 104:** Escaneo de objetos en Vuforia.

**codice** [Edit Name](#)  
Type: Device

Targets (4)

[Add Target](#) [Download Database \(All\)](#)

<input type="checkbox"/>	Target Name	Type	Rating <sup>①</sup>	Status <sup>▼</sup>	Date Modified
<input type="checkbox"/>	 codice11	Object	n/a	Active	Dec 04, 2019 22:47
<input type="checkbox"/>	 codice003	Single Image	★★★★★	Active	Nov 11, 2019 17:02
<input type="checkbox"/>	 codice002	Single Image	★★★★★	Active	Nov 11, 2019 17:01
<input type="checkbox"/>	 codice001	Single Image	★★★★★	Active	Nov 11, 2019 17:00

**Figura 105:** Base de datos creada con los marcadores a detectar (2D o 3D)Impresiones 3d de uno de los modelos creados.

La siguiente prueba que se realizó fue apuntar a las frutas que se hacían aparecer al detectar un marcador, con la utilización de un punto en el centro de la pantalla, para mostrar simplemente el nombre de dicha fruta en un cuadro de texto y ésto se utilizó como prueba de concepto de que funcionaría este tipo de interacción basada en *raycasting*, como se muestra en la Figura 106.



Figura 106: Primera prueba del raycast con modelos de frutas para desplegar el nombre del seleccionado.

#### 3.4. DESARROLLO DE LA UI.

En un principio se comenzó a experimentar utilizando una fotocopia de una de las páginas del libro *“El Códice Maya de México, antes Grolier”* [131] para hacer la detección del folio contenido en la fotocopia como un marcador AR utilizando el SDK Vuforia con Unity 2018.1.9 para realizar la aparición y animación de “apertura” del códice como si se tratara de un tríptico; también se realizaron pruebas con otras versiones más recientes de Unity, pero se decidió que se trabajaría con la 2018.1.9 debido a que ésta es la última en la que Vuforia permite hacer cambio entre cámara frontal y posterior. La primera versión de la UI fue únicamente una prueba de concepto para saber si todo funcionaría del modo esperado, por lo cual no se le prestó atención a los aspectos estéticos del códice ni de la UI; ésta contaba únicamente con dos botones, correspondientes a tomar la fotografía y a la rotación de la cámara, lucía como en la Figura 107 y varios de los problemas a resolver que surgieron con esta prueba fueron: el hecho de que la interfaz también salía en la fotografía, el lugar de almacenamiento de la fotografía no era automáticamente en la galería, sino en la carpeta de datos de la aplicación y además se identificó la necesidad de mejorar la iluminación, pues era casi imposible ver la geometría 3D del códice virtual.

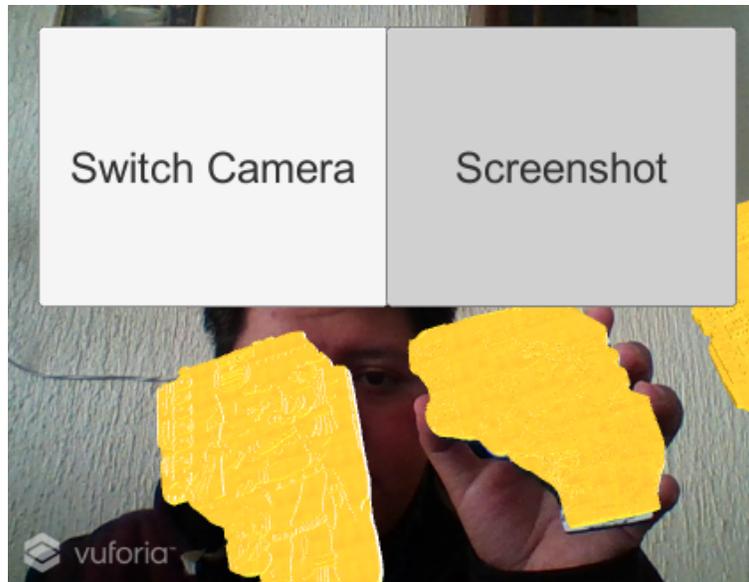


Figura 107: Prueba de concepto de la UI.

Posteriormente, como la prueba de concepto resultó ser funcional, se procedió a diseñar una UI utilizando los principios para el diseño de interfaces presentados en el capítulo 1 y a implementar las funciones creadas anteriormente, de cambio de cámara y fotografía con sus respectivos íconos. La interfaz creada utilizaba los principios de familiaridad, al usar iconografía común entre los usuarios de dispositivos móviles, como los símbolos de cámara, rotación de pantalla y galería; también se utilizó el principio de agrupación, de modo que los elementos de la UI se dispusieron de tal modo que los íconos más utilizados fueron colocados en posiciones comunes y fácilmente alcanzables, además de muy reconocibles para los usuarios de dispositivos móviles, como se observa en la Figura 108, sin embargo, como se puede ver, aún faltaban por resolver algunos problemas adicionales: el cambio de orientación de la pantalla y la toma correcta de la fotografía, pues al querer tener un efecto de "flash" o "blink" para darle retroalimentación al usuario de que la foto ha sido tomada, lo que se almacenaba era una captura en blanco junto con la UI, como se observa al centro de la Figura 110, además de que siguiendo los principios para el buen desarrollo de interfaces de AR, es aconsejable tapar la menor cantidad de área de la pantalla, por lo cual este diseño no era 100% adecuado, pues obstruye una franja de la visión total de la pantalla y adicionalmente se descubrió que si se probaba la aplicación en diferentes dispositivos, algunos con pantallas más grandes o más pequeñas que la que se estaba utilizando (2080 x 1080), la UI no se escalaba correctamente. Los primeros bosquejos para la creación de esta UI pueden ser observados en las Figuras 109 y 110.

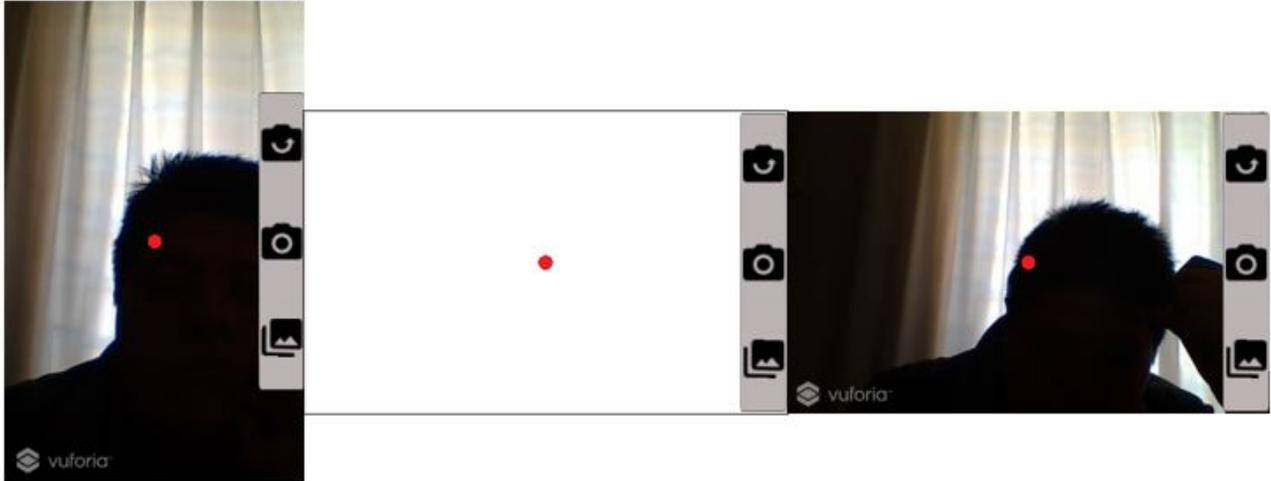


Figura 108: Segunda versión incompleta de la UI y los problemas por resolver.

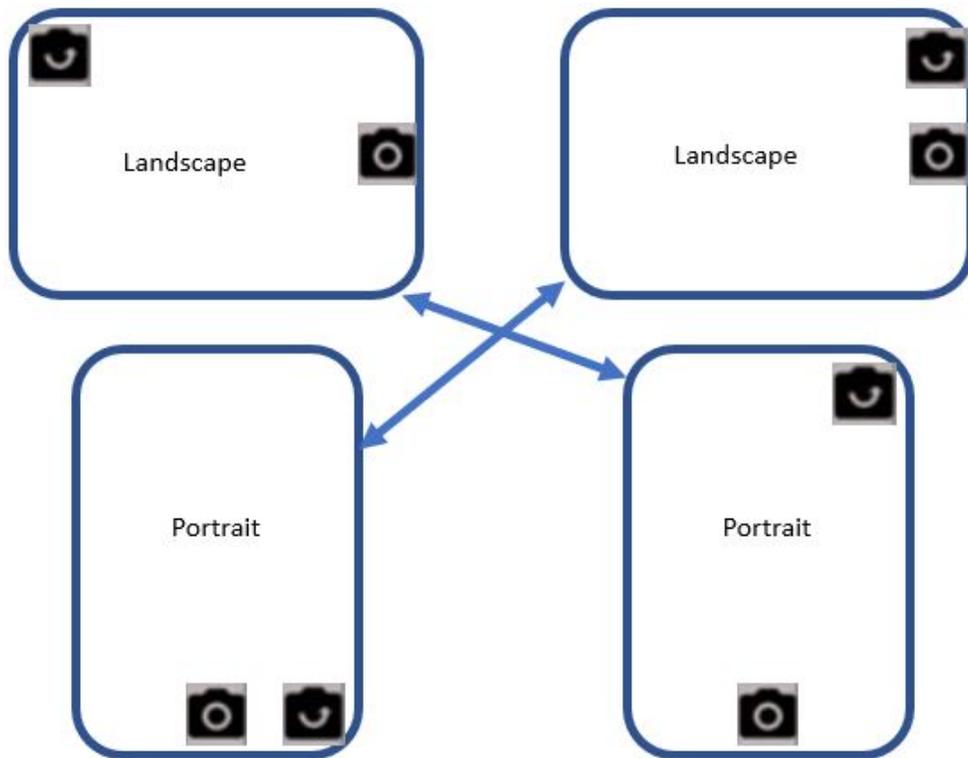


Figura 109: Primeros bosquejos de la UI

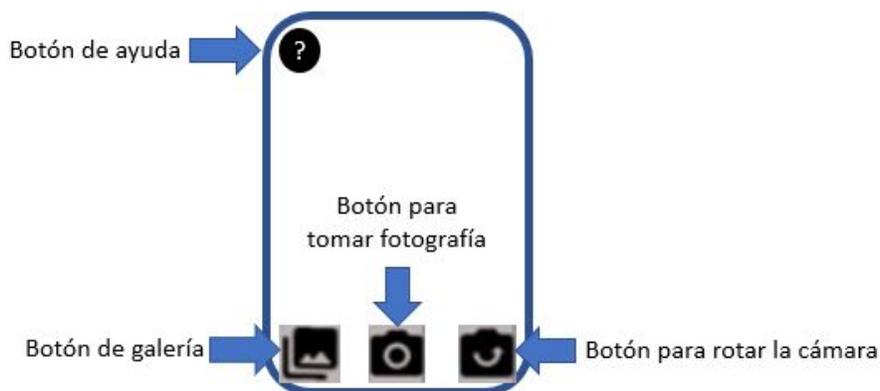


Figura 110: Primeros bosquejos de la UI

Para resolver la rotación de la pantalla lo que se hizo fue crear dos elementos *canvas*, los cuales dentro de Unity sirven como un contenedor para todos los elementos de la UI[132], de tal modo que se tenía uno de estos objetos para cada orientación y un controlador de rotación, el cual se encargaba de activar o desactivar los *canvas* dependiendo de la orientación del dispositivo, la cual se pudo obtener utilizando el comando *Input.deviceOrientation*, sentencia que arroja el valor de la orientación de la pantalla que ha sido informado por el sistema operativo del dispositivo[133], el cual puede ser fácilmente comparado con la colección de valores *DeviceOrientation*, comprendida por los valores de la Tabla 3.1.

Unknown	No se puede determinar la orientación (puede estar entre dos de las orientaciones).
Portrait	Modo portrait con el botón <i>home</i> en la parte de abajo.
PortraitUpsideDown	Modo portrait al revés, es decir con el botón <i>home</i> en la parte de arriba.
LandscapeLeft	Modo landscape con el botón <i>home</i> del lado derecho al sostener de frente el dispositivo.
LandscapeRight	Modo landscape con el botón <i>home</i> del lado izquierdo al sostener de frente el dispositivo.
FaceUp	El dispositivo es sostenido paralelo al suelo con la pantalla hacia arriba.
FaceDown	El dispositivo es sostenido paralelo al suelo con la pantalla hacia abajo.

Tabla 1: Tabla de valores posibles para determinar la orientación del dispositivo.

El código para hacer la rotación de la UI está compuesto por tres partes principales: la declaración de variables y dos

funciones; en la primera función se hace la llamada a todos los objetos de tipo *canvas* que se van a emplear posteriormente, utilizando la estructura para manejo de errores *try...catch*, debido a que para poder utilizar el mismo *script* en todas las escenas de la app se utilizaron nombres distintos para cada uno de los *canvas* y se obtienen los valores del ancho y alto de la pantalla para posteriormente poder escalar la UI según el dispositivo en que se haya instalado la aplicación, por otro lado, en la función *update* se utiliza una de las variables para almacenar el último estado de la pantalla y mediante la estructura de control *if...else* se determina en qué orientación se encuentra el dispositivo, comparando el estado actual y el anterior con los posibles valores de la Tabla 3.1, de modo que se puedan activar o desactivar los elementos de tipo *canvas* según se requiera para mostrar la pantalla en modo *landscape* o *portrait*.

Como se mencionó anteriormente, este mismo *script* fue utilizado en las tres escenas que fueron necesarias para integrar la aplicación:

- Pantalla de inicio: para mostrar el título de la tesis y los logos de la FI y la UNAM, además de un botón de inicio, con el cual se procede a mostrar el *onboarding*.
- Onboarding: un breve instructivo acerca del funcionamiento de la aplicación AR.
- Experiencia AR: la UI que se desarrolló con las funciones como apuntar para mostrar información, así como animaciones del código virtual, etc.

Se decidió reciclar este código debido a que en las tres escenas se utilizó el mismo método de tener dos elementos *canvas*, uno para cada orientación, como se ilustra en la Figura 111 y se activaban o desactivaban según lo necesario. Para obtener los valores del ancho y alto de la pantalla se utilizaron los comandos *Screen.width* y *Screen.height* y para poder escalar la UI, se accedió al componente *Canvas Scaler* que contienen todos los elementos de tipo *canvas* y se le asignaron los valores obtenidos de ancho y alto de la pantalla.

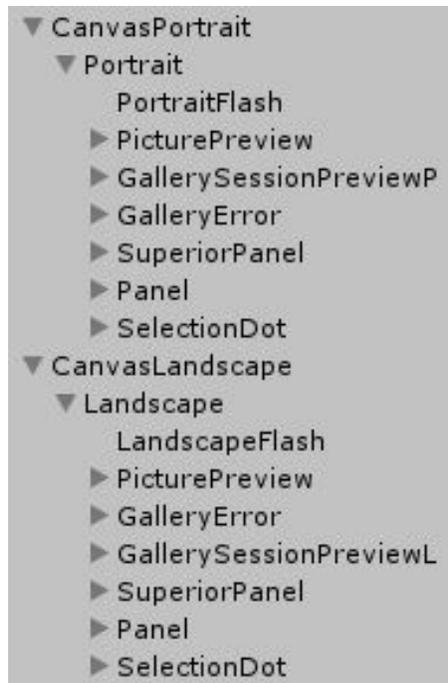


Figura 111: Utilización de dos elementos *canvas*, uno para la UI correspondiente a cada orientación.

Para determinar la orientación actual del dispositivo se utilizó la estructura de control *if...else*, como se mencionó anteriormente, de modo que las condiciones utilizadas se muestran en el siguiente pseudocódigo:

```

1  if (Orientation == Portrait ||
2     (Orientation == Unknown && LastOrientation == Portrait)||
3     (Orientation == FaceUp && LastOrientation == Portrait)||
4     Orientation == PortraitUpsideDown ||
5     (Orientation == FaceDown && LastOrientation == Portrait))
6     {
7     ACTIVAR UI PORTRAIT
8     DESACTIVAR UI LANDSCAPE
9     ESCALAR
10    }
11 else if (Orientation == LandscapeLeft ||
12          Orientation == LandscapeRight ||
13          (Orientation == Unknownk && LastOrientation == LandscapeLeft)||
14          (Orientation == Unknownk && LastOrientation == LandscapeRight))
15    {
  
```

```
16     DESACTIVAR UI PORTRAIT
17     ACTIVAR UI LANDSCAPE
18     ESCALAR
19     }
```

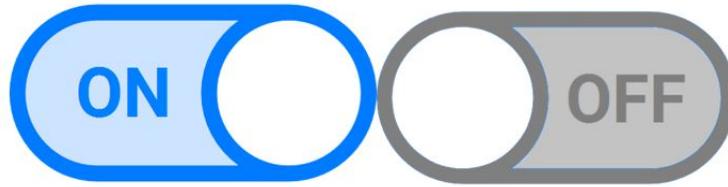
Posteriormente, se utilizaron algunas de las buenas prácticas para el diseño de interfaces de AR, como por ejemplo el obstruir la menor cantidad de espacio de la vista del usuario, por lo cual se decidió modificar la iconografía y quitar la barra que agrupaba los botones de galería, fotografía y rotar cámara, con lo cual se obtuvo la UI mostrada en la Figura 112, en la cual también se agregaron los botones de ayuda y de "apagar" y "encender" la aparición de información en las zonas interactivas, problema a resolver mencionado anteriormente en la sección 3.2, aunque estos nuevos botones aún no eran funcionales.



Figura 112: Tercera versión incompleta de la UI.

Y con el diseño elegido para el botón de activar y desactivar la aparición de información, surgió otra aplicación de las buenas prácticas o de las reglas de oro para el diseño de interfaces: brindar retroalimentación informativa al usuario, de modo que éste pueda distinguir de manera más intuitiva si se encuentra activa o inactiva la opción; para lograr esto se crearon dos imágenes, las cuales se pueden observar con más detalle en la Figura 113 y en la parte superior se les colocó dentro de la aplicación la leyenda "ARInfo", para denotar que ese era el controlador de la aparición de información; inicialmente se programó que al principio se inicializara una variable en cero y con cada toque al botón ésta tendría un incremento unitario, por lo cual si el residuo de la división de la variable entre dos es cero, se entendería que la opción de mostrar información se encuentra activa y por lo tanto se mostraría la imagen de encendido, pero por otro lado, si el residuo de la división de

la variable entre dos resultaba distinta de cero, se entendería que la información había sido inhabilitada y por lo tanto se cambiaría inmediatamente a la imagen de apagado.



**Figura 113:** Del lado izquierdo se observa el botón de encendido y del lado derecho se muestra el botón de apagado de la función de aparición de información.

A continuación se muestra en pseudocódigo el funcionamiento del botón anteriormente descrito cada vez que éste es presionado:

```
1 Variable entera contador = 0
2 Variable booleana infoController
3
4 FUNCIÓN oprimir botón:
5     contador++
6     Si contador es divisible entre 2 (contador % 2 == 0):
7         infoController = true
8         sustituir la imagen del botón por la de ''encendido''
9     En caso contrario:
10        infoController = false
11        sustituir la imagen del botón por la de ''apagado''
```

Por otro lado, para resolver el problema de que la UI salía al tomar la fotografía lo que se hizo fue programar el botón de la cámara para que primero encontrara todos los elementos de la UI y como el espacio de color usado en Unity es el definido como RGBA, por sus siglas en inglés (*red, green, blue, alpha*) se aprovechó el canal alpha de modo que durante un instante todos los elementos de la UI se volvieran transparentes, es decir con un valor alpha igual a cero, posteriormente se hacía la captura de pantalla utilizando la función `ScreenCapture.CaptureScreenshot(fileName)` de Unity, con lo cual se guardaba automáticamente en el almacenamiento del dispositivo, pero en una carpeta de difícil acceso a los usuarios, ya que la dirección en la que se guarda es: `Almacenamiento Interno/Android/data/com.dcinc.ARCodex/files`, de modo que esta captura no aparecería en la galería del usuario, por lo cual ese fue otro problema a resolver.

Para solucionar la situación de que la captura salía como una pantalla en blanco al querer implementar el efecto de "flash" al tomar la fotografía, lo que se hizo fue utilizar una corrutina (*coroutine*), debido a que cuando estas funciones son llamadas, éstas se ejecutan en su totalidad antes de retornar al punto donde se habían quedado en el código principal, con lo cual se garantiza que cualquier acción dentro de la función debe suceder en una sola actualización de cuadro (*frame*) [134], de modo que se utilizó un panel que todo el tiempo permanece sobre la UI transparente, es decir con valor alpha igual a cero y durante un instante se modifica el valor a 255 y como todos sus valores R,G y B son 255, este elemento se torna blanco, de tal manera que da el efecto de "flash" después de tomar la captura de la pantalla, ya sin los elementos de la UI, pero como todo sucede muy rápidamente, el usuario no nota que se desaparece momentáneamente la UI, únicamente se aprecia el efecto de observar que la pantalla se pone blanca un instante, para brindar una retroalimentación de que la fotografía ha sido tomada con éxito y esto también implica el principio de familiaridad dentro de la HCI, pues al tomar fotografías con *smartphones* generalmente se aprecia una pequeña pausa en el instante que ésta se toma; a continuación se muestra cómo lucía esta primera versión del código para el almacenamiento de las fotografías:

```
1 FUNCIÓN oprimir botón de cámara:
2     Comenzar corrutina (ScreenshotAndSave)
3
4 CORRUTINA ScreenshotAndSave:
5     Esperar a terminar el frame
6     Ordenar los elementos de la UI en un arreglo
7     Buscar el elemento panel flash
8     Modificar el valor alpha de cada uno de los elementos en el arreglo (alpha = 0)
9     Crear una cadena (fileName) para el nombre del archivo con la fecha y hora
10    ScreenCapture.CaptureScreenshot(fileName) -> guarda captura sin UI en almacenamiento interno, no en galería
11    Modificar el valor alpha del panel flash (alpha = 255)
12    Esperar 0.15 segundos
13    Modificar el valor alpha del panel flash (alpha = 0)
14    Modificar el valor alpha de cada uno de los elementos en el arreglo (alpha = 255)
```

Finalmente, se obtuvo como resultado el poder tomar fotografías sin que apareciera la UI ni que saliera la pantalla en blanco, como se muestra en la Figura 114.



**Figura 114:** Del lado izquierdo se observa una fotografía tomada en orientación *portrait* y del lado derecho una fotografía tomada en orientación *landscape*.

Regresando a la función de mostrar información al apuntar las zonas interactivas, como se mencionó anteriormente se utilizó *raycasting* y para resaltar las zonas interactivas se utilizó un sistema de partículas utilizando el material "Ignition Flame" proporcionado gratuitamente en el paquete "Unity Particle Pack"[135], de modo que al final se podían distinguir las zonas, pues éstas presentaban un brillo que continuamente se mueve, pero se mantiene del mismo tamaño, como se muestra en la Figura 115.

Por otro lado, para crear las zonas interactivas se utilizó un objeto vacío al cual se le asignó un *Mesh Collider*, el cual es una malla que toma la forma que se le asigne (generalmente primitivas como esferas, cilindros, etc.) con un colisionador, para poder hacer la detección de colisiones con otros objetos o rayos de manera más sencilla y precisa que con mallas más complejas[136]; en este caso se utilizó un *mesh collider* con forma de esfera, a la cual se le aplicó una transformación de escala en el eje Z (normal a la pantalla), por lo cual al final se tenía un círculo mediante el cual se podía hacer la detección de la colisión de rayo invisible proveniente del marcador del centro de la pantalla con el área interactiva, como se muestra en la Figura 116 y la razón por la cual se decidió utilizar círculos es debido a que se tiene planeado realizar un experimento de Fitts para determinar el diámetro óptimo de las áreas interactivas para incrementar la usabilidad de la aplicación.

Mientras que para hacer posible el mostrar la información después de apuntar hacia las áreas interactivas y esperar cierto tiempo, fueron necesarias diversas acciones, una de ellas por ejemplo, fue el nombrar las áreas interactivas con un número, de tal modo que al hacerse la detección de la colisión fuera posible convertir el nombre del objeto a un número entero, puesto que la sentencia *hit.transform.name*, que devuelve el nombre del objeto colisionado lo hace en un tipo de dato

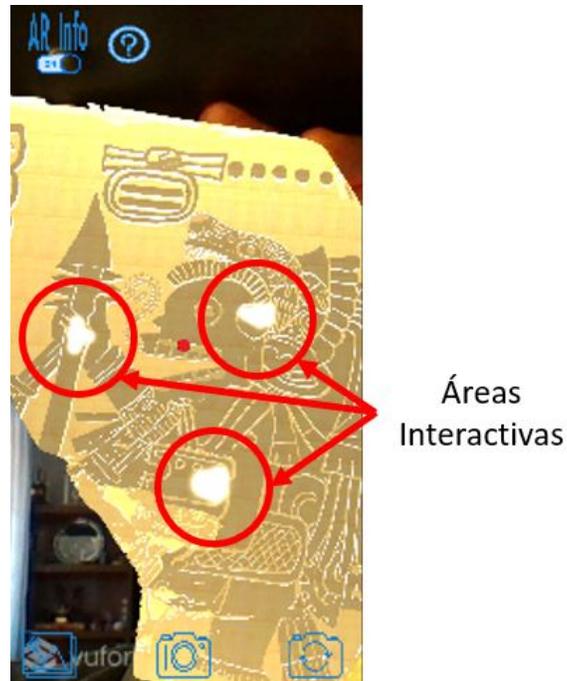


Figura 115: Sistema de partículas para identificar las áreas interactivas del código.

cadena, entonces se convirtió a un número entero mediante la función *int.Parse(bit.transform.name)*, para posteriormente utilizar un comando *switch case* con los posibles escenarios de las áreas interactivas existentes; el hecho de querer utilizar un *switch case* utilizando números enteros y no directamente la cadena devuelta fue debido a que de este modo se consumen menos recursos[137] y ésto resulta importante en este caso, puesto que este código se ejecuta en todo momento mientras la aplicación está siendo utilizada.

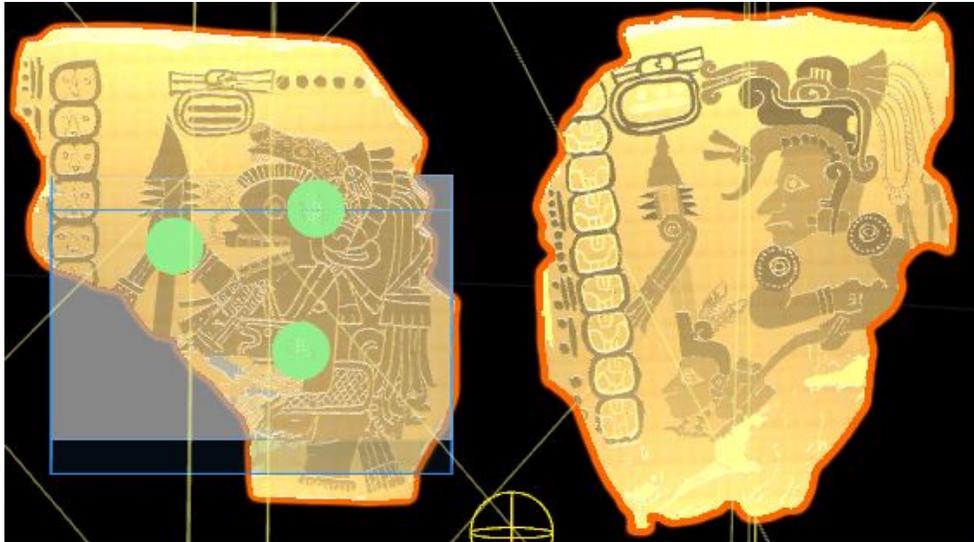


Figura 116: Del lado izquierdo se observa el código virtual que sería detectado con el marcador impreso en 3D con sus áreas interactivas circulares.

Otra de las acciones necesarias fue almacenar la información a mostrar en un "diccionario", el cual es una estructura de datos en la cual se utiliza un identificador o llave (*key*) para poder acceder a el dato o conjunto de datos que contiene el diccionario, de modo que se tenía un diccionario para cada código y dentro de cada diccionario se colocó toda la información correspondiente al área que ha sido apuntada, así como la posición y tamaño que deberá tener el recuadro con la información una vez que éste aparezca y el tamaño de letra, como se muestra en la Figura 117.

```

Dict1 = new Dictionary<int, codexDictionary>()
{
    //El primer dígito corresponde al código y el segundo a la zona
    {11, new codexDictionary{Information = "Deidad maya de la muerte conocido como...
    XPosition = -30, XSize = 50,
    YPosition = 100, YSize = 50,
    TextSize = 8,
    LinePos = true,
    }},

    {12, new codexDictionary{Information = "Lleva un tocado de cabeza de jaguar sin...
    XPosition = 100, XSize = 50,
    YPosition = 100, YSize = 60,
    TextSize = 8,
    LinePos = true,
    }},
}
  
```

Figura 117: Ejemplo del diccionario utilizado para almacenar la información de uno de los folios del código.

Como se puede observar en la Figura 117, el primer número contenido en cada elemento del diccionario, en este caso

los números 11 y 12 son los *keys* para acceder a la información que se encuentra después de la primera coma, como la cadena que es el texto a desplegar en pantalla y los otros elementos, que tienen que ver con el posicionamiento en pantalla del mismo.

Una vez detectada una área interactiva, para darle al usuario la retroalimentación de que debe esperar cierto tiempo mediante un indicador radial concéntrico a la marca del centro de pantalla mencionada anteriormente, lo que se hizo fue utilizar una corrutina que fuera activada al detectarse uno de los casos del *switch case* mencionado anteriormente y no haberse activado una variable que indica que el tiempo y el llenado del indicador han sido completados, como se muestra a continuación:

```
1 Variable booleana infoController
2 Variable flotante tiempoEspera
3 Variable string hitName
4 Variable booleana hitDetection
5 Variable booleana ready
6 Variable booleana started
7
8 FUNCIÓN Start:
9     Inicializa las variables
10
11 FUNCIÓN Update:
12     Si la variable infoController es verdadera(infoController == true, cuando la opción de mostrar la
13     información se encuentra habilitada):
14         Lanzar el rayo a partir del centro de la pantalla.
15         Si hay colisión, guardar el nombre del objeto en la variable hitName.
16         Convertir la variable hitName a tipo entero.
17
18     Switch(hitName):
19         case 0:
20             hitDetection = true
21             if(ready == true)
22                 {
23                     FUNCION ShowInfo(11)
```

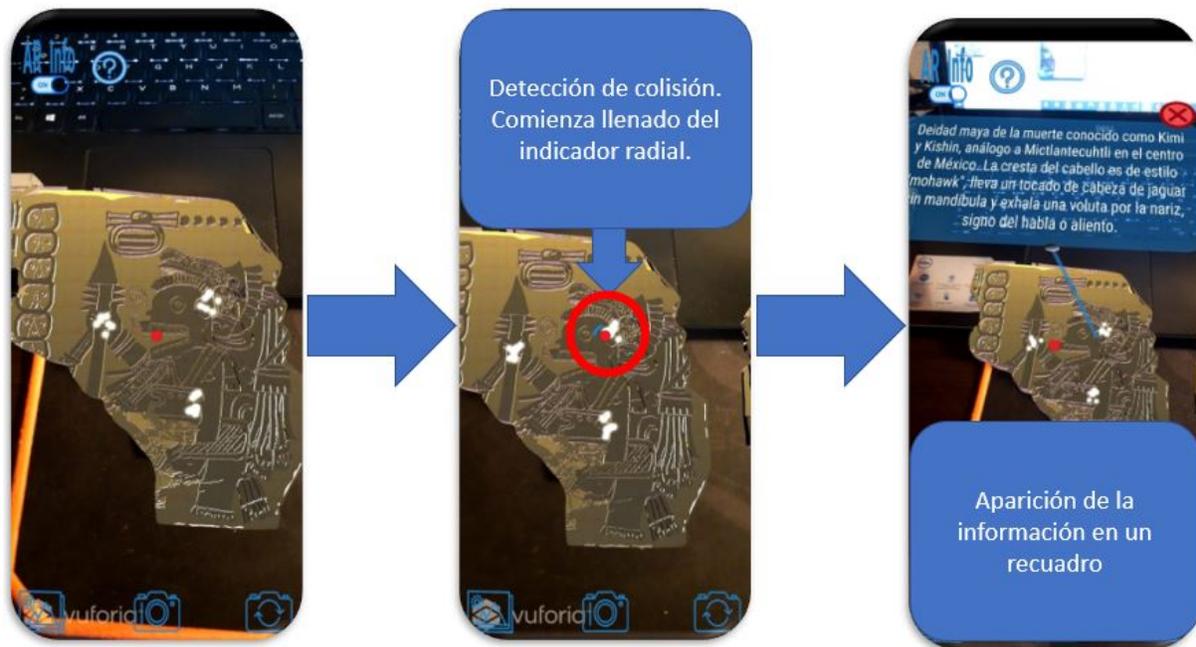
```

23     }
24     break;
25     case 1:
26         hitDetection = true
27         if(ready == true)
28         {
29             FUNCION ShowInfo(12)
30         }
31         break;
32     En caso contrario:
33         hitDetection = false
34
35     Si ya hubo una colisión, ya se inicializó la variable dentro del radialBar y aún no se inicia la corrutina(
36     hitDetection == true && ready == false && started == false):
37
38         Comenzar corrutina (LlenarTimer)
39
40     En caso contrario(hitDetection == false && ready == false && started == true):
41
42         Significa que la zona interactiva ha dejado de ser apuntada.
43         Detener todas las corrutinas.
44         Limpiar todas las variables.
45
46     FUNCIÓN ShowInfo(valor entero):
47
48         Leer la información correspondiente al valor entero utilizándolo como key del diccionario anteriormente
49         mencionado: Posición X, Posición Y, Información.
50
51         Inicializa las variables.
52
53         Crear el recuadro que va a contener la información a partir de un prefab.
54
55         Colocar el recuadro en las posiciones leídas.
56
57         Colocar la información correspondiente en el recuadro.
58
59     CORRUTINA LlenarTimer:
60
61         Buscar el objeto ''radialBar''

```

```
53 Localizar dentro del objeto el código ''radialBar''
54 Inicializar la variable del porcentaje de llenado en cero.
55 Modificar la variable trigger a verdadero.
56 Esperar en segundos (tiempoEspera)
57 ready = true
58 started = true
59 Regresar al código
```

El pseudocódigo presentado anteriormente representa el funcionamiento básico del código colocado en el objeto *AR-Camera* con la finalidad de poder detectar la colisión del rayo que sale del centro de la pantalla con alguna de las zonas interactivas, para inmediatamente comenzar a llenar un indicador radial mediante el uso de una corrutina y cuando éste se encuentre completamente lleno después de cierto tiempo preestablecido, da pie a la creación de un recuadro colocado en una ubicación específica con la información correspondiente a dicha zona a partir de un *prefab*, el cual es un tipo de objeto que puede ser almacenado con componentes y propiedades, lo cual los vuelve especialmente útiles si se requiere crear varios objetos del mismo tipo, ya que al tener todas las instancias las mismas propiedades se ahorra tiempo debido a que sólo se crea el objeto una vez y subsecuentemente sólo se crea una nueva instancia del *prefab* [138], de modo que para este caso se utilizó un recuadro azul semitransparente con un botón rojo con una cruz para poder cerrar el mismo *prefab*, todo esto para utilizar los principios de familiaridad y consistencia para apoyar la usabilidad, de modo que fuera sencillo para los usuarios poder identificar que ese botón sirve para cerrar la ventana, como se ilustra en la Figura 118.



**Figura 118:** Secuencia para la apertura de información, de izquierda a derecha: visualización de las áreas interactivas, apuntar el área interactiva con el centro de la pantalla y esperar a que se llene el indicador y aparición de la información.

En el pseudocódigo anterior únicamente se activa la variable dentro del objeto del indicador radial y se espera una cantidad de tiempo, de tal modo que la animación del llenado se activa y ejecuta al recibir el cambio en el estado de dicha variable dentro de otro objeto y la programación para crear ese funcionamiento se muestra a continuación:

```

1 Variable booleana trigger
2 Variable flotante speed
3
4 FUNCIÓN Update:
5     Si la variable trigger ha sido modificada a verdadera(trigger == true):
6         Activar el componente radialBar
7         Si el porcentaje de llenado es inferior a 100 (currentAmount < 100):           {
8             currentAmount += speed * Time.deltaTime;
9             im.fillAmount = currentAmount/100;
10            Con lo cual se va llenando progresivamente el círculo.
11            En caso contrario, si el porcentaje de llenado es mayor o igual a 100 (currentAmount >= 100):
12            currentAmount = 0;

```

Una vez creado el cuadro con la información a partir de un *prefab*, se utilizó el mismo código explicado en el pseudocódigo de la página 97 para poder hacer el cierre de la información al apuntar con el centro de la pantalla el botón rojo, para lograrlo únicamente se tuvo que agregar otro caso al *switch case* y una función en la cual se destruía el *prefab* y se limpiaban las variables del mismo modo que en los demás casos, sin embargo, para cerrar el recuadro con la información, el indicador concéntrico que indica la espera de cierto tiempo antes de poder ejecutar la acción deseada fue colocado alrededor del propio botón, como se muestra en la Figura 119 y se implementó la funcionalidad de poder cerrarlo al tocarlo, de modo que fuera más sencillo el poder cerrar el recuadro de forma rápida y sin la espera del llenado del indicador concéntrico.



**Figura 119:** A la izquierda se muestra la función de cerrar mediante apuntar al botón y a la derecha se ilustra la función de cerrar al tocar el botón.[18]

Regresando a la función de tomar fotografías, faltaban dos problemas por resolver: el primero de ellos era que el lugar de almacenamiento de las imágenes idealmente debería ser la galería del teléfono y no una ruta desconocida para el usuario y el segundo, que surgió al haber solucionado el primero, era que estando almacenada la imagen en la galería del teléfono, no se podría hacer una previsualización de la foto dentro de la aplicación debido a los permisos requeridos por Android.

Para resolver el primer problema, se utilizó el *plugin Native Gallery*, disponible en la *Asset Store* de Unity[139], con el cual se logró guardar las imágenes directamente en la galería, al guardar primero todos los píxeles como una textura y posteriormente utilizando un método incluido en el *plugin* mediante el cual se solicita permiso a Android para guardar la imagen en la galería y dentro del mismo método se incluye el nombre de la carpeta en la que se guardará (si dicha carpeta no existe, se crea), el nombre que se le dará al archivo (por conveniencia, como se explicará más adelante el nombre utilizado fue "ARCodexDD-MM-AA!No.Sesión") y un mensaje que sirve en la etapa de depuración del código, el cual devuelve en la consola la ruta en la que se almacenó la imagen, así como el resultado del permiso solicitado. Para la aplicación se creó una carpeta de nombre "ARCodex", como se muestra en la Figura 120.

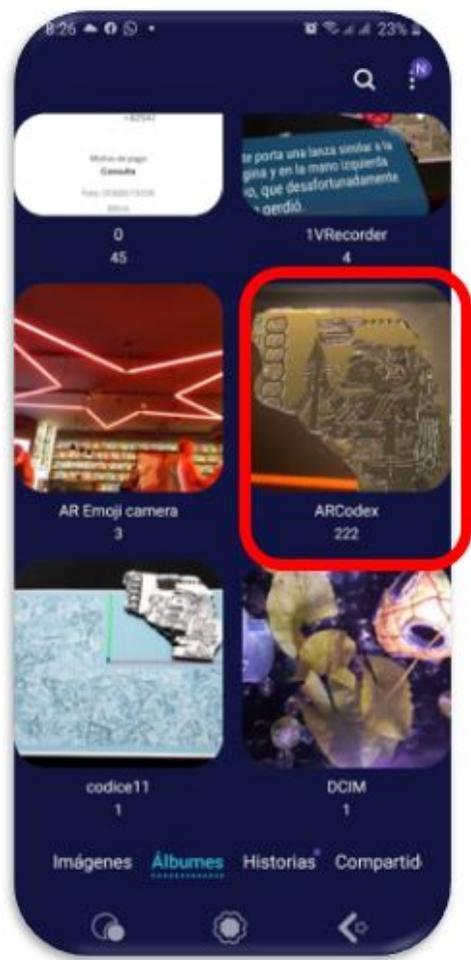


Figura 120: Carpeta donde se almacenan las imágenes de la aplicación.

Sin embargo, el almacenar la imagen en la galería provocó que ésta no fuera accesible dentro de la aplicación, a pesar de que el *plugin* cuenta con un método para obtener recursos de la galería, pues éste no funcionó adecuadamente, por lo cual

se pensó en una solución para poder mostrar una previsualización y galería de las fotos tomadas en la sesión actual dentro de la aplicación, de tal modo que fue necesario guardar la imagen en dos lugares al tomar la fotografía: en la galería y en la carpeta de datos de la aplicación, pero esto podía generar el problema de ocupar mucho espacio de almacenamiento, lo cual afectaría de manera negativa la percepción de los usuarios de la aplicación, por lo cual se generó un *script* para contabilizar las veces que ha sido abierta la aplicación y se añadió una función al código utilizado para tomar las fotografías, de modo que se compara mediante el nombre asignado a la imagen guardada en el almacenamiento interno, con el número de sesión actual, para poder eliminar las demás imágenes de la carpeta de almacenamiento interno y únicamente dejar las de la sesión en curso.

Para mostrar la previsualización, se creó un *prefab* que se mantiene oculto dentro de ambas UI: *landscape* y *portrait*. Y dentro de la misma corrutina ejecutada al llamar la función asociada al botón de tomar la fotografía, se activa el *prefab*, el cual contiene un elemento *Raw Image*; se decidió utilizar este tipo de objeto, debido a que a diferencia de los objetos de tipo *Image*, a éstos se les puede asignar cualquier textura y con los elementos *Image* únicamente se pueden utilizar *Sprites*, los cuales son otro tipo de objeto gráfico 2D utilizado en Unity [140][141], de modo que se le asigna directamente al *Raw Image* la textura creada a partir de leer los pixeles de la pantalla para poder utilizar el *plugin*, esta textura es nombrada como "ss" en el pseudocódigo. Este *prefab* cuenta de igual manera con su botón para cerrar, pero en esta ocasión, como es un elemento UI, únicamente se permite la interacción mediante un toque y adicionalmente se colocó del lado opuesto un botón con el símbolo de *share*, para que los usuarios puedan compartir la imagen recién capturada en redes sociales, correo, etc. y cabe mencionar que normalmente el botón de cerrar iría del lado derecho, pero en este caso se invirtió su posición típica, debido a que esa posición es la más alcanzable para la mayoría de usuarios (diestros) y eso la convierte en la ubicación que más naturalmente se utilizará, con lo cual los usuarios compartirán las fotografías, dándole así una mayor difusión al código o a cualquier otra exhibición en la cual se plantee utilizar una función similar, como se muestra en la Figura 121.



Figura 121: Objeto que contiene la Vista previa de la fotografía y los botones: cerrar y compartir.

Para el botón de share se utilizó el *plugin Native Share*[142] y para poder emplearlo fue necesario localizar la textura colocada en la *Raw Image* y convertirla a formato PNG, para posteriormente poder utilizar un método incluido en el *plugin*, con el cual se añaden todas las imágenes que se desea compartir, un texto que se inserta automáticamente en las aplicaciones de mensajería como WhatsApp, como se muestra en la Figura 122 y una vez añadidas todas las imágenes, se llama otro método mediante el cual se comparten los archivos.



Figura 122: Funcionamiento del botón compartir con el mensaje personalizable para las aplicaciones de mensajería.

Finalmente, para la función de galería dentro de la aplicación, fue necesario crear un *script* en el cual primero se crea un arreglo de todos los archivos contenidos en la carpeta de datos de la aplicación, luego a cada uno de los nombres contenidos en ese arreglo, se les busca el caracter "!" y éste se utiliza como separador, para identificar todos los archivos creados al tomar una fotografía y aquellos que cuenten con el separador, son guardados en otro arreglo de imágenes elegibles, en el cual se hace otra búsqueda y se guarda en un nuevo arreglo los dígitos contenidos antes del punto que indica la extensión del archivo, ya que éstos indican el número de sesión en la que fueron tomadas las imágenes, pues como se mencionó anteriormente, se asignó el nombre de las imágenes utilizando "ARCodexDD-MM-AAAA!No.Sesión.png", de modo que un ejemplo de nombre real asignado a una fotografía podría ser "ARCodex24-03-2021!246.png" y se comparan todos los elementos de este arreglo con el número de sesión actual, el cual se actualiza en el archivo de preferencias del usuario cada vez que se utiliza la aplicación y si ambos valores son iguales, se guardan en otro arreglo, el cual se utiliza para saber cuántos paneles con una *Raw Image* crear y posteriormente cargar la textura correspondiente a cada imagen en dichos objetos, pero si los valores resultan diferentes, se procede a eliminar los archivos que no coinciden con la sesión, de tal modo que únicamente se almacena por duplicado (en galería y en la carpeta de la app) el menor número de imágenes posible.

En los siguientes pseudocódigos se muestra el funcionamiento de la cámara para realizar el almacenamiento tanto en la galería como en la carpeta de datos de la aplicación dentro del almacenamiento interno del dispositivo, así como la numeración de sesiones para poder eliminar las imágenes no deseadas del almacenamiento interno, de modo que no se ocupe una gran cantidad de espacio de la memoria del dispositivo y las funciones utilizadas al oprimir los botones de compartir y galería interna.

```
1 FUNCIÓN oprimir botón de cámara:
2     Comenzar corrutina (ScreenshotAndSave):
3
4 CORRUTINA ScreenshotAndSave:
5     Esperar a terminar el frame.
6     Ordenar los elementos de la UI en un arreglo.
7     Buscar el elemento panelflash.
8     Modificar el valor alpha de cada uno de los elementos en el arreglo (alpha = 0).
9     Crear una cadena (fileName para el nombre con 'ARCodex' como inicio, fecha y hora de la captura, !,
    número de sesión ejemplo: ARCodex24-03-2021-19-47-12!246.png)
10    Crear una nueva textura con propiedades de ancho y alto de la pantalla del dispositivo.
11    Leer todos los pixeles de la pantalla y guardarlos en la textura 'ss'.
```

```
12
13 Guardar en galería utilizando el plugin ''Native Gallery'', para guardar la textura ''ss'' en la carpeta ''
    ARCodex'' con el nombre ''fileName'' y solicitar la respuesta:
14
15     NativeGallery.Permission permission = NativeGallery.SaveImageToGallery(ss, "ARCodex", fileName, (
    success, path) => Debug.Log("Media save result: " + success + " " + path));
16
17 ScreenCapture.CaptureScreenshot(fileName)
18 Modificar el valor alpha del panel flash (alpha = 255)
19 Esperar 0.15 segundos
20 Modificar el valor alpha del panel flash (alpha = 0)
21 Modificar el valor alpha de cada uno de los elementos en el arreglo (alpha =255)
22 Activar el prefab ''preview''
23 Cargar la textura ''ss'' en el componente Raw Image contenido en ''preview''.
```

```
1 Variable entera count
2 FUNCIÓN actualizar número de sesión:
3     Si existe la variable ''sesion'' en el archivo de preferencias del usuario:
4         Guardar el valor leído en la variable entera ''count''
5         Si count es igual a 999:
6             sesion = 0
7         En caso contrario:
8             sesion = sesion + 1
9     En caso contrario:
10         Crear la variable ''sesion'' en el archivo de preferencias de usuario y asignarle 0 como valor.
```

```
1 FUNCIÓN compartir:
2     Leer la textura de la Raw Image.
3     Añadir la textura a la lista de archivos a compartir.
4     Compartir mediante el plugin ''Native Share'' con el mensaje ''Imagen tomada con la aplicación 'AR como
```

recurso interactivo en museos'' y esperar el resultado de la operación

```
5
6     new NativeShare().AddFile(texture)
7     .SetSubject("Compartir")
8     .SetText("Imagen tomada con la Aplicación 'AR como recurso interactivo en museos'").
9     SetCallback((result, shareTarget) => Debug.Log("Share result: " + result + ", selected app: " +
10    shareTarget))
11    .Share();
```

1 FUNCIÓN abrir galería:

2 Guardar los archivo existentes en la carpeta de la app en el arreglo ''files''.

3 FUNCIÓN listar elegibles.

4 FUNCIÓN comparar sesión.

5 CORRUTINA cargar imágenes.

6 FUNCIÓN eliminar imágenes anteriores.

8 FUNCIÓN listar elegibles:

9 Separar cada elemento dentro de ''files'' al encontrar el separador ''!''

10 Guardar la segunda parte del nombre dentro del arreglo ''elegibles'', sólo si ésta es diferente de null.

12 Separar cada elemento dentro de '' elegibles'' al encontrar el separador ''.''

13 Guardar la primera parte del nombre dentro del arreglo ''middle'', sólo si ésta es un número entero.

15 FUNCIÓN comparar sesión:

16 Comparar cada elemento dentro de ''middle'' con el número de sesión actual.

17 Si el elemento comparado es igual al número de sesión:

18 Se guarda el elemento correspondiente en el arreglo ''finales''.

19 En caso contrario:

20 Se guarda el elemento en el arreglo ''eliminar''.

22 CORRUTINA cargar imágenes:

```

23 Activar el prefab ''galería''.
24 Clonar tantos Raw Image como elementos dentro de ''finales'' existan menos uno.
25 Espaciar los Raw Image de acuerdo al tamaño de pantalla.
26 Obtener las texturas asociadas a cada uno de los elementos del arreglo ''finales'' y guardarlas en un
arreglo de texturas ''texturas''.
27 Cargar las texturas en cada uno de los Raw Image recién creados.
28
29 FUNCIÓN eliminar imágenes anteriores:
30 Eliminar de la carpeta de archivos cada una de las imágenes asociadas a los elementos dentro del arreglo ''
eliminar''.

```

Finalmente, para tener la primera versión completa de la aplicación, faltaban por resolver dos asuntos: mejorar la iluminación de los modelos 3D, para que éstos se pudieran visualizar de mejor manera y crear las escenas correspondientes a la bienvenida y el *onboarding*.

Primero se realizó la mejora en la iluminación, de tal modo que se pudiera optimizar el aspecto de los códices virtuales, para lo cual se utilizaron dos luces por cada folio del códice, una en la parte superior y otra en la parte inferior, de tal modo que el volumen de cada una de las extrusiones que componen la pieza fuera resaltado, al reflejar la luz sobre la superficie más externa y crear sombra en las partes internas de las piezas; a estas luces se les asignó un color amarillo y la propiedad *spot*, que es utilizada en la creación de videojuegos dentro del motor Unity para iluminación proveniente de fuentes artificiales, como la originada por linternas, autos, etc. y generalmente sirven para brindar un efecto dramático y a las cuales se les pueden modificar los valores de rango y ángulo[143], además de que a cada uno de los códices virtuales se le colocó una textura amarilla que asemejara el aspecto de papel antiguo, todo esto como se muestra en la Figura 123.

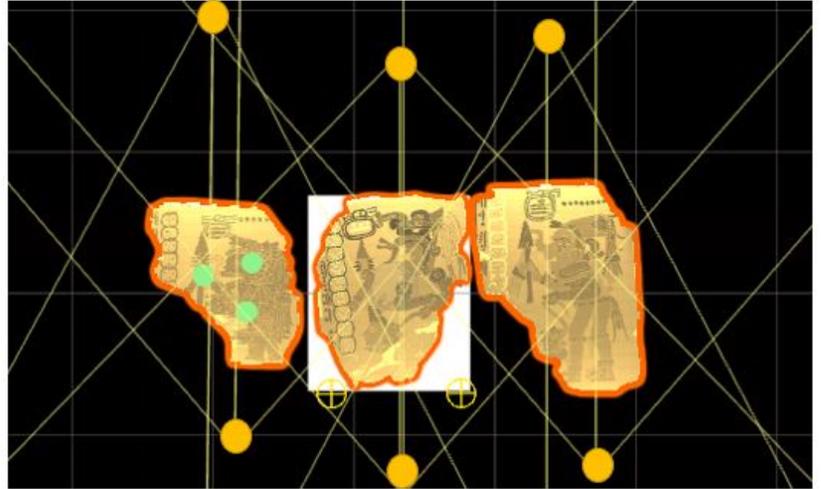
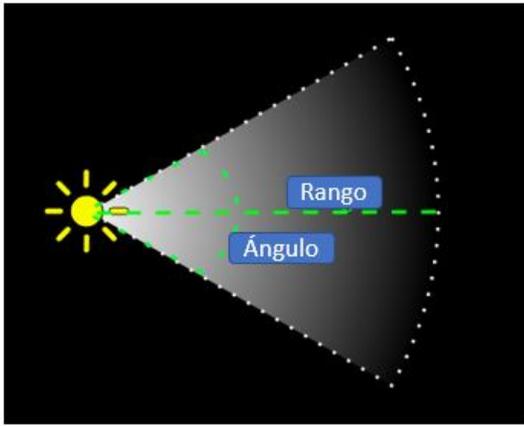


Figura 123: A la izquierda se muestra el funcionamiento de una luz "spot light"[143] (traducida por el autor) y a la derecha se muestra cómo se colocaron dos luces por pieza de código, de modo que en el ejemplo se tienen seis luces en total.

Para las escenas de bienvenida y onboarding se decidió seguir utilizando los mismos colores: azul semi transparente para los recuadros que contienen texto y blanco para las letras. Para la portada únicamente se colocó el nombre del presente trabajo, junto con los escudos de la UNAM y la Facultad de Ingeniería y el botón "Comenzar", como se muestra en la Figura 124.

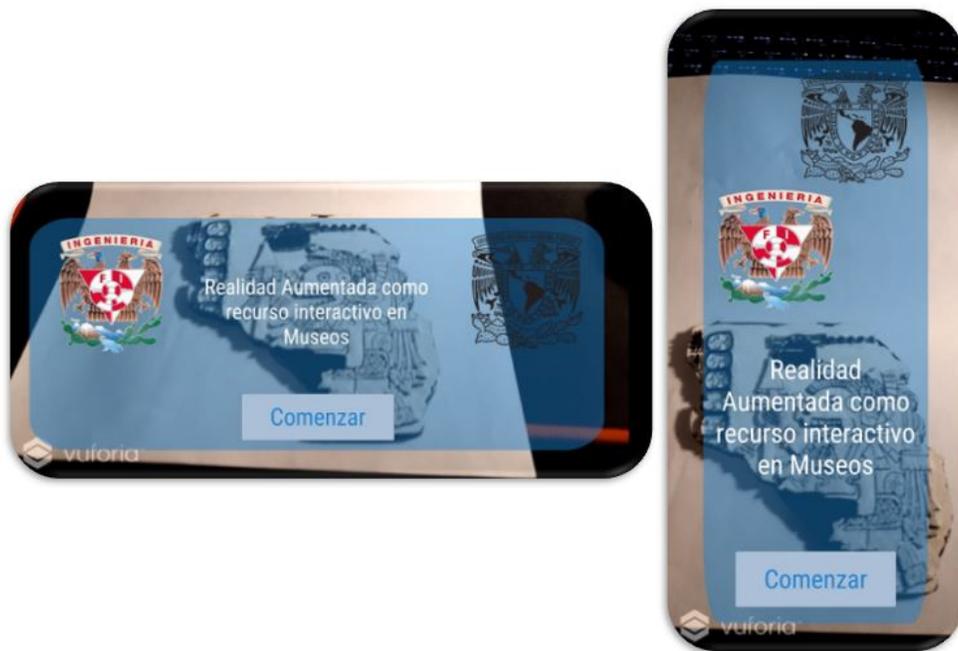


Figura 124: Escena de bienvenida a la izquierda en modo *landscape* y a la derecha en modo *portrait*.

Y por otro lado, para la escena del *onboarding* se realizó primero un prototipo utilizando diapositivas, pues ésto ayudaría a identificar toda la información necesaria para mostrarle al usuario, utilizando la secuencia más conveniente para ello y al mismo tiempo, ahorrando tiempo de desarrollo, pues al ser un prototipo de fácil creación, ayudaría a establecer las funciones necesarias para crear la escena y brindaría una idea de la cual partir para su programación, como se puede apreciar en la Figura 125.



Figura 125: Prototipo del *onboarding*.

Para la programación del *onboarding*, primero se crearon las pantallas a partir del prototipo y se colocaron todas ellas dentro de un objeto que sirvió de contenedor, para ellas y para los pequeños círculos en la parte inferior que sirven como indicador del paso que está observándose; para recrear la interacción de hacer un movimiento *swipe*, es decir, deslizar hacia los lados para mover las pantallas hacia la izquierda para avanzar a la siguiente pantalla y hacia la derecha para retroceder a la pantalla anterior se creó un *script* que permite modificar qué tanto se debe deslizar el dedo en la pantalla y que automáticamente realiza los cambios de color de los círculos que indican la pantalla actual, conforme se avanza o retrocede y para hacer más ilustrativa esta escena, se colocaron GIFs animados, de tal modo que se puede observar por ejemplo, el llenado del marcador radial concéntrico, hasta que aparece la información del código o la aparición y desaparición de los marcadores de las zonas interactivas al prender o apagar el botón de la información, etc. Las pantallas utilizadas en ésta escena se muestran

en la Figura 126 y a continuación se explica en pseudocódigo el *script* para hacer el cambio de pantallas con el movimiento *swipe*.

```
1 FUNCIÓN Start:
2     Identificar el contenedor de pantallas y su componente ''ScrollRect''.
3     Se crea un arreglo de números flotantes de tantos elementos como pantallas dentro del contenedor existan.
4     Se llena el arreglo al multiplicar por i la división de uno entre el número de elementos menos uno, para
5     obtener la cantidad de movimientos que se podrán hacer:
6         pos[i] = (1f/(pos.Length - 1f) * i)
7     Ajustar la sensibilidad del movimiento scroll.
8 FUNCIÓN Update:
9     Si se detecta un toque en la pantalla:
10        Guardar el valor inicial del ''scroll bar''.
11
12    En caso contrario: (cuando se deja de tocar la pantalla)
13        Si el valor final del toque es mayor a la distancia entre cada pantalla:
14            Activar el siguiente círculo azul y desactivar el anterior.
15            Hacer una interpolación de la posición del contenedor con el valor inicial y el siguiente valor
            guardado en el arreglo ''pos[]'', para que sea una transición suave.
```

A continuación, en la Figura 126, se muestran imágenes de la escena en modo portrait para el *onboarding*.



Figura 126: Pantallas del Onboarding.

Finalmente sólo restaba unir las tres (bienvenida, *onboarding* y experiencia AR), para que juntas crearan la experiencia completa y ésto se logró planteando el diagrama mostrado en la Figura 127, en el cual se puede observar la secuencia de aparición de las escenas, según el botón utilizado; de modo que cuando se inicia la aplicación, siempre aparece la pantalla de la portada con el nombre de la aplicación y los escudos de la UNAM y la FI, posteriormente, cuando se presiona el botón Comenzar, se pasa a la siguiente escena: el *onboarding*, en el cual se pueden seguir dos rutas: la primera es leer las seis pantallas e ir las moviendo mediante un movimiento *swipe*, hasta llegar a la última y presionar el botón: Iniciar. Y la segunda ruta es simplemente presionar el botón en forma de flecha hacia la derecha de "Saltar", con lo cual se retoma el principio de brindar atajos para usuarios más experimentados dentro de la HCI. Y el último de los movimientos entre escenas es el que se obtiene al estar ya en la escena de la experiencia AR y presionar el botón de ayuda, con el ícono de símbolo de interrogación "?", mediante el cual, se regresa a la escena del *onboarding* y se vuelven a tener las mismas opciones: leer el *onboarding* completo o saltarlo, todo ésto se logra al emplear los ajustes para crear el archivo APK de Unity (*Build Settings*), mediante el cual se selecciona el orden en el que se desea que aparezcan las escenas, como se muestra en la Figura 128 y los siguientes *scripts* empleados en los botones: Saltar, Comenzar, Iniciar y de ayuda ("?"); mediante los cuales se cambia la escena mostrada al incrementar o reducir el índice de la escena actual.

```
1 FUNCIÓN Cargar siguiente escena:  
2     Obtener el índice de la escena actual.  
3     Cargar la escena correspondiente al índice actual más 1.  
4     SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
```

```
1 FUNCIÓN Cargar escena anterior.  
2     Obtener el índice de la escena actual.  
3     Cargar la escena correspondiente al índice actual menos 1.  
4     SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex - 1);
```

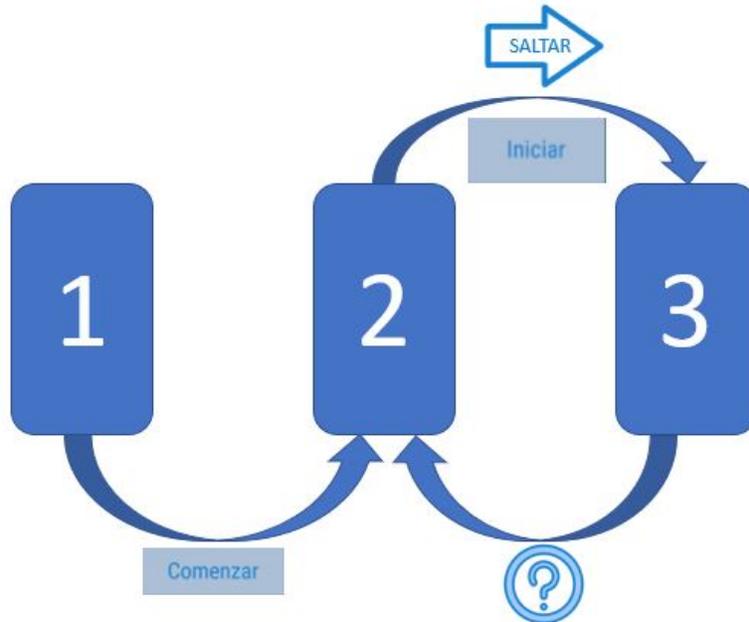


Figura 127: Diagrama del orden de las escenas.



Figura 128: Orden de las escenas en Unity.

Al finalizar todo el proceso anteriormente descrito se obtuvo la primera versión de la aplicación con todas las características mencionadas en el presente capítulo completamente funcionales, mediante las cuales se pretende crear un recurso para mejorar la experiencia de los visitantes de alguna exposición relacionada con la preservación de la cultura, como lo podría ser la exhibición del Códice Maya Mexicano, a través de utilizar una de las herramientas que ha estado cobrando cada vez más importancia dentro de la interacción humano - computadora y la cual está siendo progresivamente explorada en diversos ámbitos, como la industria manufacturera, entretenimiento, enseñanza, etc.: la Realidad Aumentada, por lo cual en el siguiente paso del desarrollo del presente trabajo corresponde realizar pruebas de usabilidad con personas ajenas a la creación de esta aplicación, para obtener información de retroalimentación que resulte relevante para poder ayudar a mejorar el diseño de la aplicación y al mismo tiempo observar de cerca las interacciones que tienen personas a las que se les presenta por primera vez esta experiencia AR y con base en dichas observaciones y en el análisis de los datos recolectados mediante las pruebas, poder mejorar la experiencia de los posibles usuarios finales.

*To design an easy-to-use interface, pay attention to what users do, not what they say. Self-reported claims are unreliable, as are user speculations about future behavior.*

Jakob Nielsen, Doctor en Diseño UI

# 4

## Pruebas de usabilidad

### 4.1. PRUEBA DE FITTS.

Como se mencionó en el capítulo 1, los experimentos de Fitts son de gran importancia en el ámbito de la HCI, ya que con estas pruebas se mide el tiempo que tardan las personas en apuntar a cierto objetivo con diámetro variable, de tal modo que, una vez recolectados y procesados los datos, se obtiene el modelo que describe la interacción, con lo cual se puede predecir el tiempo requerido para desplazarse de un lado a otro con el dispositivo de entrada, pero también puede ser utilizada para obtener el tamaño que los objetos interactivables deberían tener, para que los usuarios puedan tener una mejor usabilidad.

Y recordando lo establecido en el capítulo 1, la Ley de Fitts está representada por:

$$MT = a + b * ID \quad (\text{ec. 4})$$

Donde: MT es el tiempo del movimiento.

a y b son las constantes determinadas experimentalmente que varían en cada dispositivo.

ID es el índice de dificultad.

$$ID = \log_2\left(\frac{2A}{W}\right) \quad (\text{ec. 5})$$

Donde: A es la distancia desde el punto inicial hasta el objetivo.

W es el ancho del objetivo.

Para realizar el experimento de Fitts se realizó un prototipo meramente estético como se muestra en la Figura 129, mediante el cual se describió el funcionamiento que debería ser posteriormente programado para realizar el experimento real que fue utilizado con potenciales usuarios para poder obtener un tamaño de las áreas interactivas más cómodo para los usuarios, a continuación se describirá el proceso de creación de la aplicación para la prueba de Fitts.



Figura 129: Prototipo para el experimento de Fitts.

En la escena de Unity que constituyó el experimento de Fitts fue necesario crear un *script* que funciona de manera

simultánea y complementaria al *script* utilizado para realizar las detecciones con objetos virtuales mencionado en el capítulo anterior; la secuencia descrita a continuación se ilustra en la Figura 130, lo primero que se observa en la aplicación es un botón de inicio, el cual desaparece una vez se ha tocado y posteriormente comienzan a aparecer los objetivos dentro de un rectángulo creado para limitar el lugar de aparición de éstos, de tal modo que la amplitud del movimiento que se tendría que hacer se vea reducida, con el fin de obtener un modelo lo más certero posible, debido a que el experimento a realizar presenta algunas diferencias con el experimento original de Fitts, pues éste consiste en realizar la tarea de apuntar objetos rápidamente con algún dispositivo de entrada con una mano, como por ejemplo un *mouse* o *stylus*, mientras que en el experimento a realizar se tendrá una tarea de apuntar un objetivo dentro de la pantalla de un *smartphone* al mover éste cuando se está sujetando con una o ambas manos en modo *landscape*.

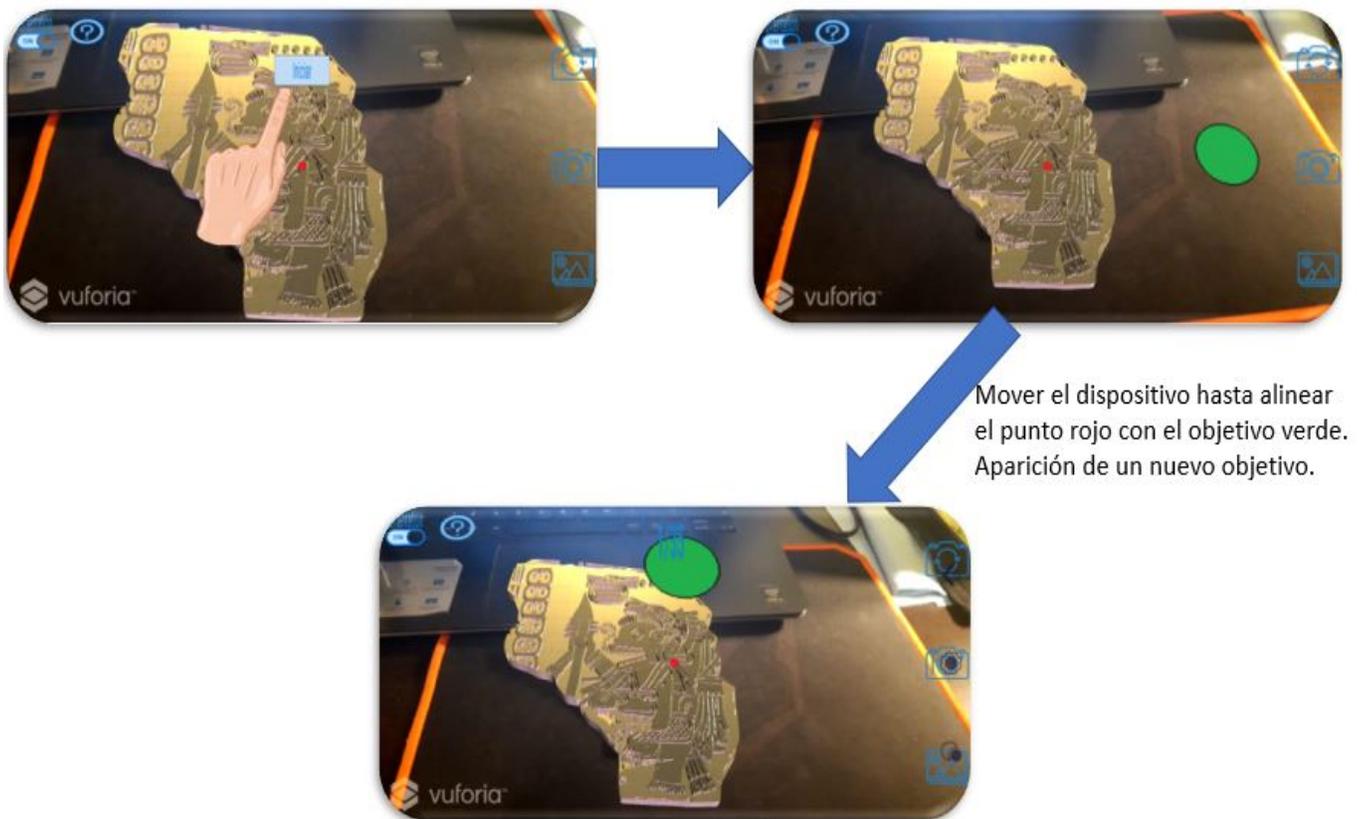


Figura 130: Secuencia del experimento de Fitts.[18]

Una vez iniciado el experimento, se genera el primer objetivo; los objetivos aparecen en ubicaciones aleatorias del rectángulo delimitado y con escalas también aleatorias, pues cada que se genera uno de éstos, se elige el tamaño que tendrá el elemento a partir de un arreglo que contiene las escalas correspondientes  $a$ : 0.5, 0.6, 0.8, 1, 1.2, 1.4, 1.6, 1.8 y 2 veces el tamaño original propuesto, el cual se generó únicamente al trazar un área circular en la cual pudiera caber la mayor parte de los elementos individuales del folio del código en el cual se implementaron las funciones completas descritas en el capítulo anterior; una vez generado el objetivo, se espera la respuesta de la colisión del marcador del centro de la pantalla (un punto rojo) con el objetivo, se procede a eliminarlo y generar el siguiente, cuando se ha acertado el primer objetivo aparece en la parte superior central de la pantalla el número de objetivos que se han acertado y los faltantes, como se observa en la Figura 131, posterior a cada colisión se almacenan el tiempo entre colisiones y la distancia calculada entre los puntos donde aparecieron los objetivos en un documento de texto para posteriormente poder obtener el modelo.

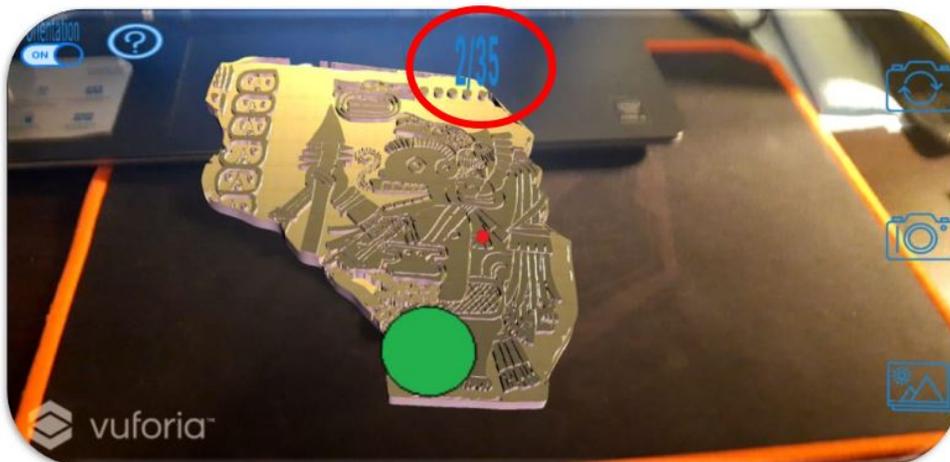


Figura 131: Número mostrado en el experimento de Fitts.

Al realizar el análisis de los datos obtenidos, se observó que al efectuar la regresión lineal de los índices de dificultad calculados y el tiempo utilizado en cada uno de los objetivos, con lo cual se obtendrían los coeficientes  $a$  y  $b$  de la ecuación de Fitts que modela la interacción, se tiene un bajo coeficiente de correlación (0.4027), como se muestra en la gráfica y ecuación presentadas en la Figura 132, lo cual podría deberse a que, a pesar de que se simplificó la tarea del experimento de Fitts al tener la misma distancia para todos los usuarios y se trató de minimizar el movimiento a efectuar, existe cierto ángulo en el que las personas pueden moverse sobre su propio eje, como se muestra en la Figura 133, el cual no está considerado en el modelo, de la misma manera, Rhos et al. [144] sugieren que para un experimento similar, el modelo debe modificarse para considerar como otra variable el tamaño de la pantalla del dispositivo para obtener mejores resultados, lo cual tampoco

fue considerado en el presente trabajo.

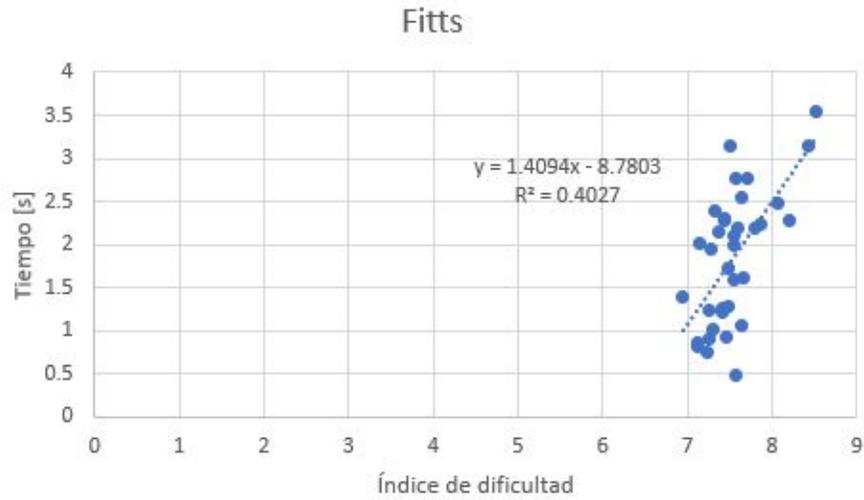


Figura 132: Gráfica y ecuación obtenida en el experimento de Fitts.

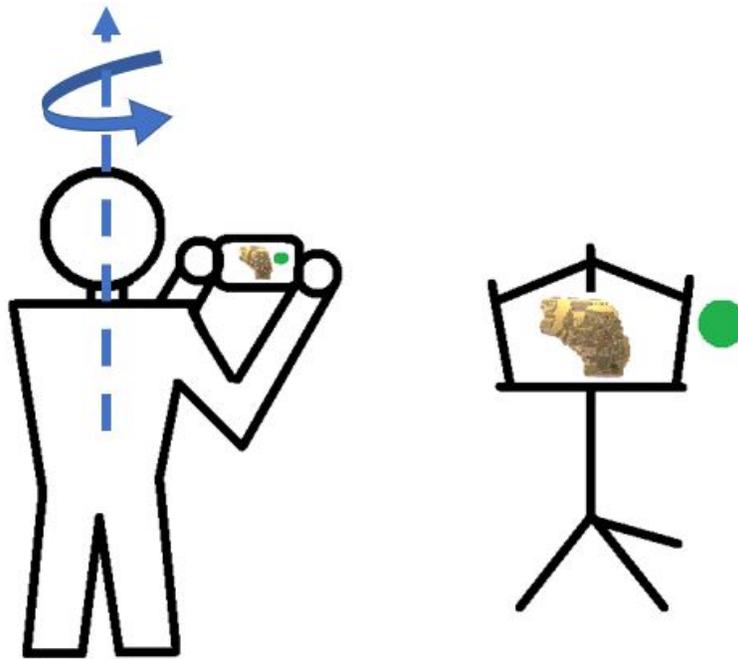


Figura 133: Consideración excluida del experimento de Fitts.

Sin embargo para los fines del experimento realizado, como se tuvieron valores fijos de tamaños de objetivo y lo que se quería determinar era cuál de los tamaños presentados era el óptimo para las áreas interactivas, se analizó la relación entre la escala y el índice de dificultad, con lo cual se obtuvo la ecuación presentada junto con la gráfica en la Figura 134 y como

era de esperarse, debido a la definición del índice de dificultad, se obtuvo un coeficiente de correlación alto (0.9455) y se corroboró lo establecido en el capítulo 1 acerca de las conclusiones de la Ley de Fitts, debido a que el tamaño de los objetos interactivos resulta ser inversamente proporcional a la dificultad, es decir que a mayor tamaño de botón, pedal, etc. menor será la dificultad con la que será alcanzado y por lo tanto la interacción se dará más rápidamente y viceversa, a menor tamaño de los objetos interactivos, mayor dificultad para el usuario y por lo tanto mayor tiempo tardará en ocurrir la interacción.

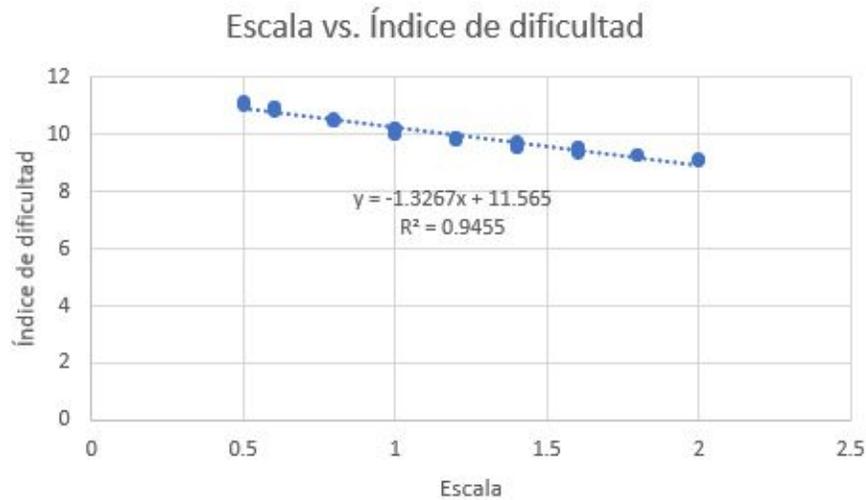


Figura 134: Gráfica y ecuación de la Escala vs. Índice de Dificultad.

Debido a la situación sanitaria causada por la COVID-19 a nivel mundial no fue posible realizar el experimento de Fitts a las afueras de algún museo para poder contar con un tamaño de muestra considerablemente más grande, pudiendo obtener así más datos de distintos tipos de posibles usuarios, sino que éste fue realizado únicamente con trece personas con las que el autor del presente trabajo tenía interacción durante el confinamiento, siguiendo las normas sanitarias. Otra de las simplificaciones realizadas para minimizar las posibles variantes con cada uno de los participantes fue colocar el código impreso en 3D con el cual se hace la detección en un atril a una altura y distancia fijas para todos los participantes, como se puede apreciar en el diagrama de la Figura 135 y en la fotografía de la Figura 136, en la cual se observa la realización de una de las pruebas de Fitts efectuadas.

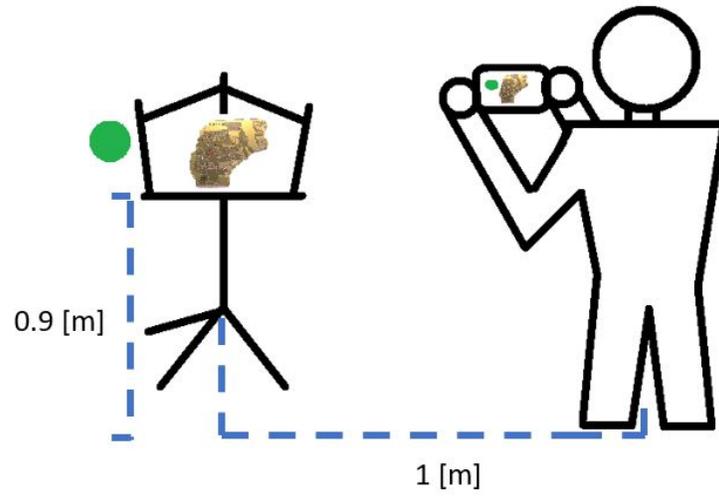


Figura 135: Diagrama de la configuración utilizada para el experimento de Fitts.



Figura 136: Fotografía de la configuración utilizada para el experimento de Fitts.

A continuación se muestra el pseudocódigo del principal *script* que permite la realización del experimento de Fitts.

```
1 FUNCIÓN Start:
2     Inicialización de variables
3
4 FUNCIÓN Update:
5     Si se detecta que el código impreso en 3D está en el área de la cámara:
6         Colocar el botón ''Inicio'' en la pantalla.
7
8     Si el botón ''Inicio'' ha sido tocado:
9
10        Crear una coordenada aleatoria en el rango (X: 70-410, Y: 80-250, Z=150)
11        Seleccionar una escala aleatoriamente del arreglo [0.5, 0.6, 0.8, 1, 1.2, 1.4, 1.6, 1.8, 2]
12        Instanciar un círculo en la coordenada anterior y modificar su escala en (x,y) de acuerdo a la escala
seleccionada.
13        Guardar el valor de tiempo actual en la variable time1
14
15    Si se detecta una colisión con el objeto ''200'' (el círculo verde) y el número de aciertos aún no es igual
al número de tests a realizar:
16
17        Desaparecer el círculo.
18        Hacer un sonido.
19        Incrementar el número que aparece en la pantalla en (número de aciertos actuales / número de tests)
20        Calcular la distancia del movimiento y el tiempo mediante:
21
22        float Dist = Vector2.Distance((Vector2)newInitialPos, (Vector2)fittsTargetPos);
23        string Distance = Dist.ToString();
24        float elapsedTime = Time.time - time1;
25
26        Escribir en un archivo de texto la distancia y la resta del tiempo de la colisión menos time1 con el
formato:
27            1; tiempo; distancia; escala
28            2; tiempo; distancia; escala
```

```

29     3; tiempo; distancia; escala
30     .
31     .
32     .
33     n; tiempo; distancia; escala
34
35     *** <- separación entre sujetos
36
37     1; tiempo; distancia; escala
38     2; tiempo; distancia; escala
39     3; tiempo; distancia; escala
40     .
41     .
42     .
43     n; tiempo; distancia; escala
44     ***
45
46     Crear una coordenada aleatoria en el rango (X: 70-410, Y: 80-250, Z=150)
47     Seleccionar una escala aleatoriamente del arreglo [0.5, 0.6, 0.8, 1, 1.2, 1.4, 1.6, 1.8, 2]
48     Instanciar un círculo en la coordenada anterior y modificar su escala en (x,y) de acuerdo a la escala
49     seleccionada.
50     Guardar el tiempo actual en time1.
51
52     Si se han hecho todos los aciertos programados:
53
54     Mostrar un mensaje de ''Gracias por participar'', con un botón de ''Aceptar'', el cual reinicia todo y
55     coloca el separador en el archivo para correr otra prueba con otra persona.

```

#### 4.2. PRUEBA DE TIEMPO DE CARGA Y PERCEPCIÓN DEL USUARIO.

Se propuso esta prueba inicialmente como se muestra en la Figura 137, es decir con un controlador de tipo *slider*, mediante el cual los usuarios ajustaban el tiempo de carga hasta sentirse cómodos con la duración de la espera del llenado

del indicador radial, para la interacción mediante *raycast*; se realizó una prueba con los mismos 13 participantes que en la prueba de Fitts, pero se observó que éstos no comprendían bien la tarea, por lo cual se decidió cambiar el experimento, al disminuir de un número infinito de posibles posiciones del slider a cinco opciones de tiempos fijos: 1, 2, 3, 4 y 5 segundos, con lo cual se obtuvo un experimento que fue más comprensible para los usuarios y también resultó más sencillo de analizar y como resultado sería más fácil extraer la conclusión que serviría para modificar el diseño según lo encontrado. De igual manera, lo anterior sirvió como comprobación indirecta del principio de la Navaja de Occam[145], debido a que resultó más sencillo para los usuarios comprender este segundo experimento, en el cual se les proporcionó un menor número de posibles opciones a los usuarios.

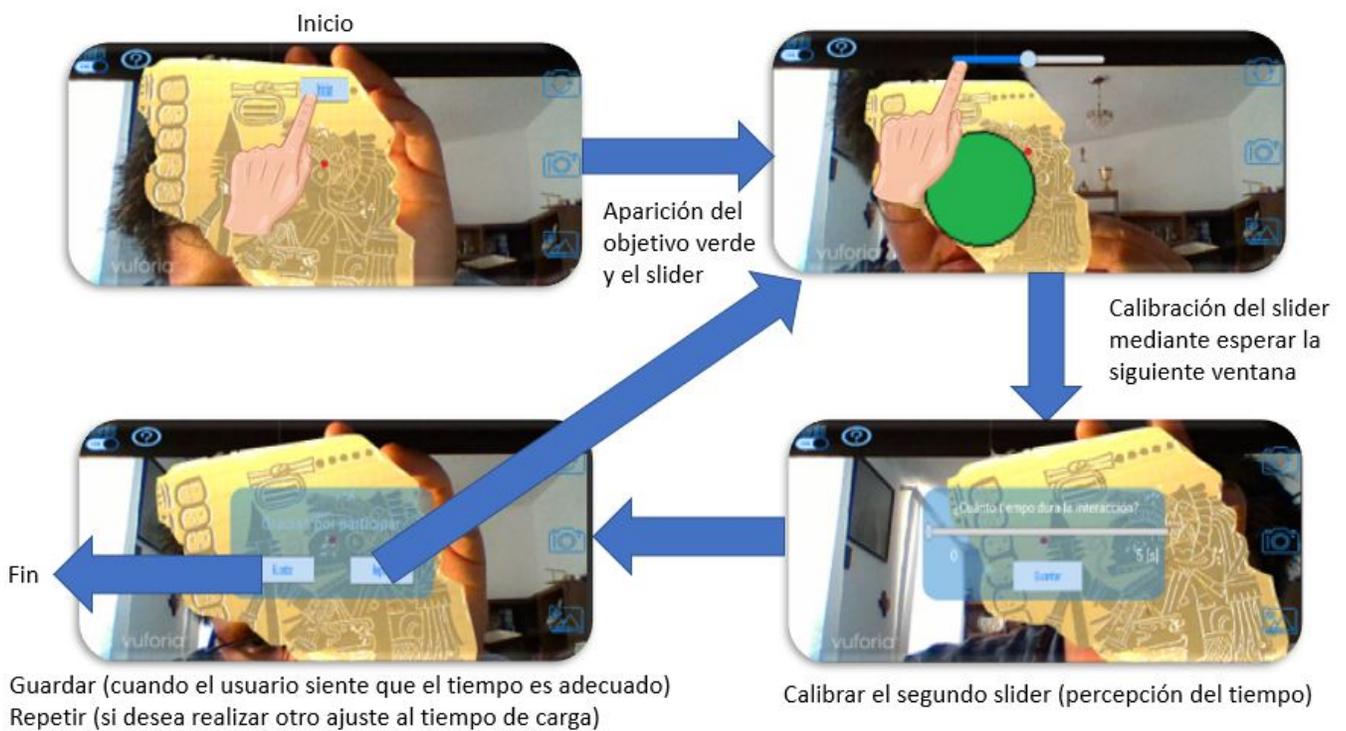


Figura 137: Primer experimento para el tiempo de carga óptimo.[18]

En la Figura 137 se aprecia la secuencia que se utilizó en una primera versión del experimento para conocer cuál sería el tiempo óptimo de carga para la interacción mediante *raycast*, la cual consistía en que al igual que con el experimento de Fitts, ésta comenzaba al hacerse la detección del código 3D con la cámara del *smartphone*, posteriormente aparecía el botón de inicio y después de que éste fuera oprimido, aparecía un círculo verde grande y un *slider*, se debía ajustar dicho controlador a un valor cualquiera inicialmente y luego apuntar al círculo verde para ver el tiempo que tardaba en aparecer la siguiente pantalla, en la cual se debía ajustar el nuevo *slider* con lo cual se pretendía saber si la percepción del tiempo resultaba ser

modificada al estar interactuando con un entorno de AR, posteriormente se debía repetir el experimento hasta que el usuario sintiera que el tiempo que había ajustado con el *slider* era adecuado para la interacción y posteriormente guardarlo y finalizar el experimento, pero se encontró que para muchos usuarios este experimento resultó confuso, pues querían información acerca de cuánto debería durar y retroalimentación para saber si lo estaban haciendo de manera correcta, a lo cual se les respondía que no había correcto o incorrecto, simplemente tenían que ajustar el tiempo a uno que ellos sintieran cómodo para la interacción de apuntar al objetivo, sin embargo se siguió observando que se tenía una pobre comprensión de la tarea, por lo cual se decidió realizar una simplificación del experimento, quitando la parte de saber si la percepción del tiempo es modificada en un entorno AR y dejando únicamente cinco opciones que se les presentaban a los usuarios y ellos debían elegir de entre cinco respuestas para describir cómo había sido el tiempo presentado ante ellos, como se muestra en la Figura 138.

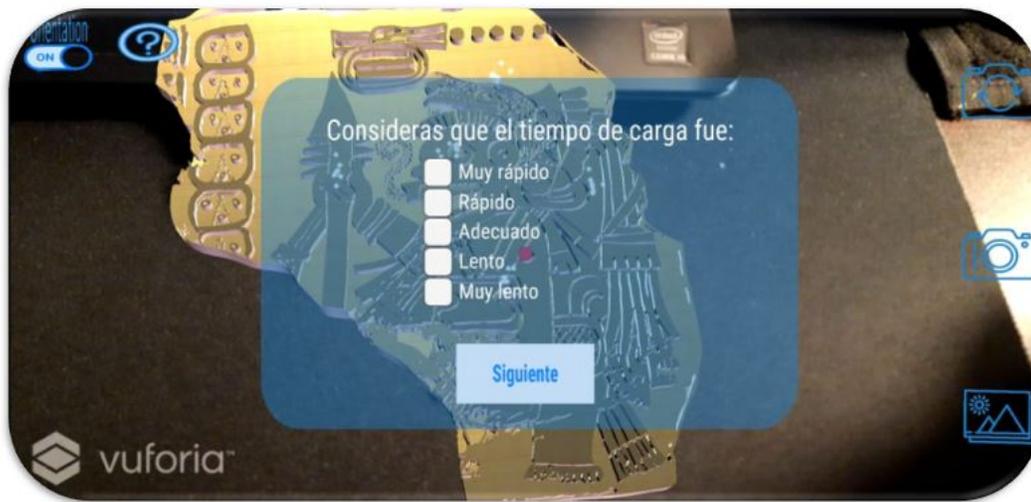


Figura 138: Segundo experimento para el tiempo de carga óptimo.

Para la aplicación de esta segunda prueba se le pedía a los usuarios responder la encuesta mostrada en la Figura 138 cinco veces, una vez después de cada una de las 5 interacciones de distintas duraciones, como se muestra en el flujo de la aplicación representado en el diagrama de la Figura 139.

De igual manera que en el experimento de Fitts, después de concluir la prueba con cada persona se guardaban los datos, se separaban para diferenciarlos de los datos recopilados de las personas anteriores y siguientes y se reiniciaban todas las variables para poder realizar el experimento con la siguiente persona, como se muestra en el pseudocódigo posterior a la Figura 139.

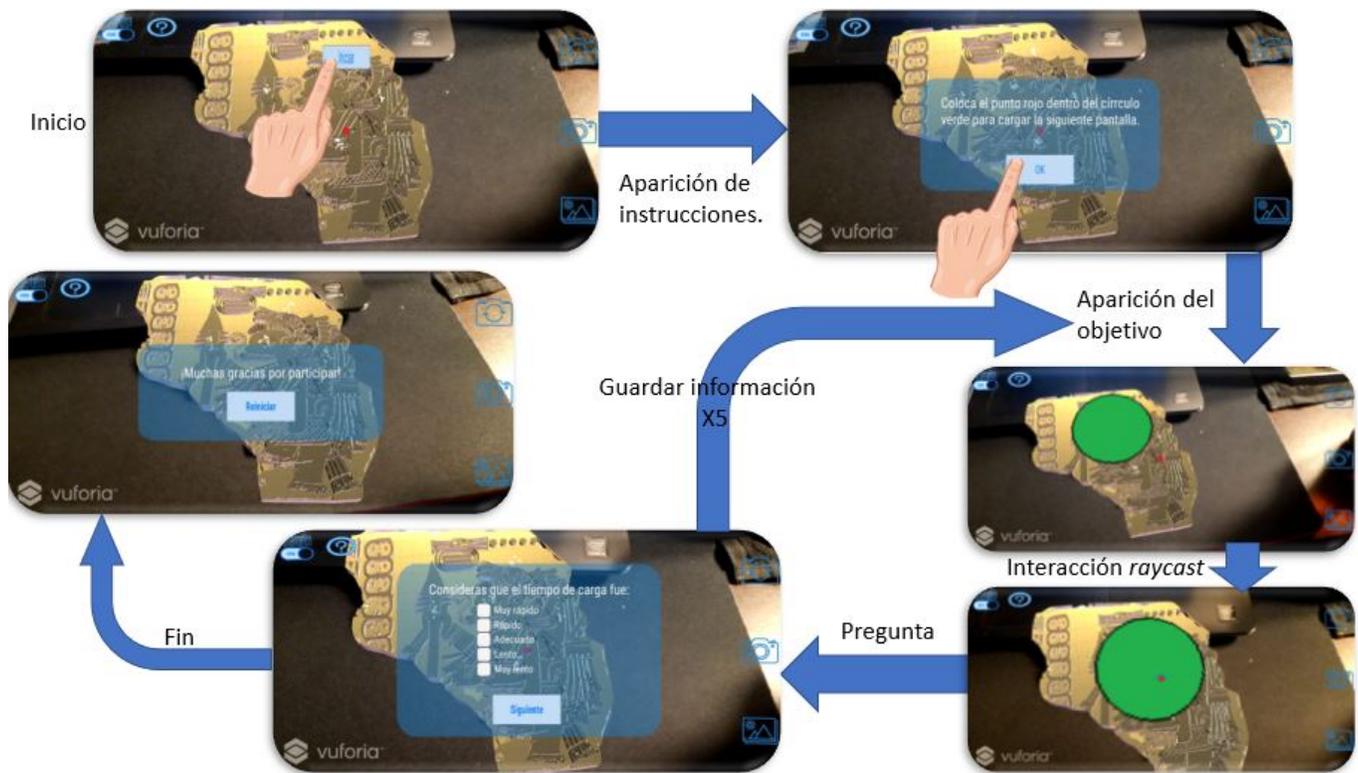


Figura 139: Secuencia del segundo experimento para el tiempo de carga óptimo.[18]

```

1 FILLING TEST
2
3 FUNCIÓN Start:
4     Inicialización de variables
5
6 FUNCIÓN Update:
7     Si se detecta que el código impreso en 3D está en el área de la cámara:
8         Colocar el botón "Inicio" en la pantalla.
9
10    Si el botón "Inicio" ha sido presionado:
11        Mostrar la pantalla con las instrucciones y el botón "OK".
12
13    Si el botón "OK" ha sido presionado:
14        Instanciar un círculo verde en una coordenada y escala fijas.
15

```

```
16 Si se detecta una colisión con el objeto ''200'' (el círculo verde):
17
18     Variable entera num +=;
19
20     Iniciar corrutina ''Interaccion''
21
22     Si num <= 5:
23         Mostrar el diálogo para el cuestionario de la percepción de la duración de la interacción.
24
25     En caso contrario, si num >5:
26         Mostrar el diálogo final.
27
28     En caso contrario:
29         Reiniciar las variables.
30
31 CORRUTINA Interaccion:
32     Activar el ícono de carga (círculo azul concéntrico al centro de la pantalla).
33     Acceder al código dentro del ícono de carga e iniciar el porcentaje de llenado en 0.
34     Acceder al código dentro del ícono de carga y activar la variable booleana ''trigger''.
35     Acceder al código dentro del ícono de carga para pasar el valor ''num'', para calcular la duración de la
    animación de llenado.
36
37     Esperar (5f * num) segundos para volver al código principal.
```

```
1 CUESTIONARIO
```

```
2
```

```
3 Función Update:
```

```
4
```

```
5 Si se presiona el botón ''Siguiete'' y se ha seleccionado alguna opción:
```

```
6     Guardar en un archivo el número de prueba (tiempo que duró la interacción) y la respuesta elegida.
```

```
1 FINAL
```

```
2
```

```

3 Función Update:
4     Si se presiona el botón ''Reiniciar'':
5         Reiniciar todas las variables dentro de los 3 códigos.
6         Colocar en el archivo un separador ''* * *'' para diferenciar entre cada prueba.

```

Posteriormente a realizar las pruebas, se analizó la información recabada y se prestó especial atención a la frecuencia con la que los participantes contestaron "adecuado" y se observó que los valores de tiempo que tuvieron mayor frecuencia fueron: 3, 4 y 5 segundos, como se muestra en las gráficas de las Figuras 139 y 140; en la primera gráfica se observan la percepción que contestaron los usuarios y el tiempo en los ejes XY y en el eje Z se muestra la frecuencia que tuvo cada una de las posibles respuestas de percepción a cada tiempo presentado y por otro lado, en la segunda gráfica se aislaron únicamente las respuestas de percepción de tiempo "adecuado", donde se observaron los tres valores de tiempo con mayor incidencia.

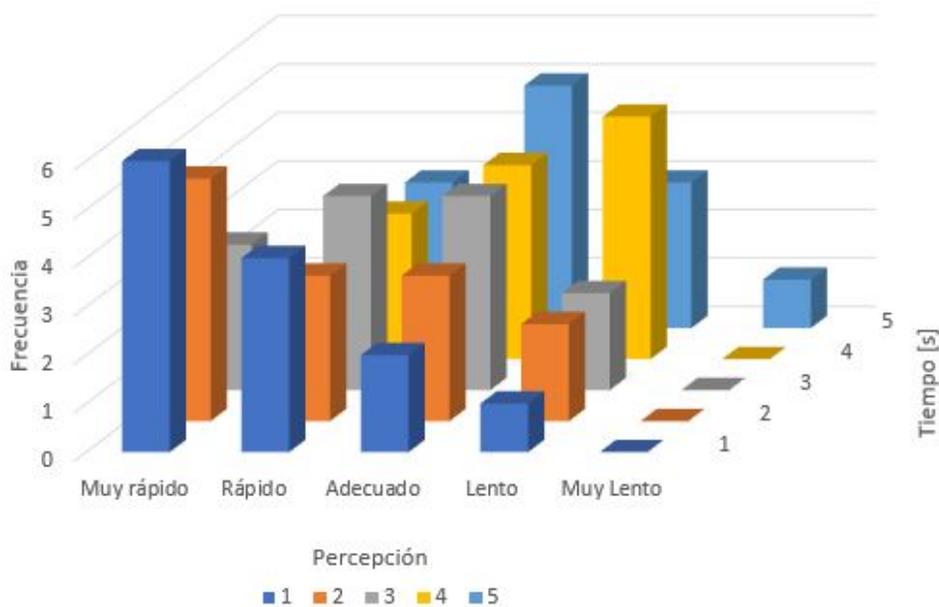
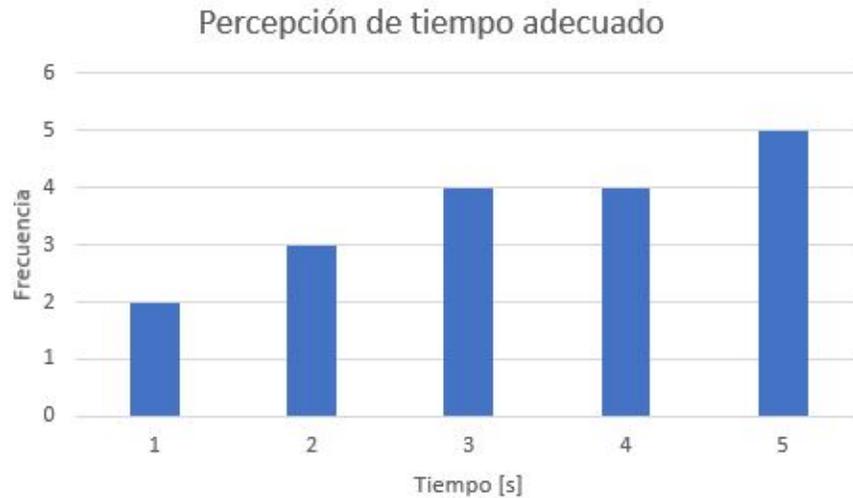


Figura 140: Gráfica de los resultados del experimento del tiempo de carga óptimo.



**Figura 141:** Gráficas de los resultados del experimento del tiempo de carga óptimo.

Finalmente, se muestra en la Figura 142 la realización de una de las pruebas, teniendo en consideración las mismas medidas sanitarias que en el experimento de Fitts, con la diferencia de que para emular de mejor manera la interacción real del usuario con el código, esta vez no se colocó el modelo 3D en un atril, sino que las personas lo tenían en una mano y con la otra sostenían el dispositivo.

Durante el desarrollo de las pruebas fue posible utilizar uno de los modelos más importantes y que ha tenido mucha relevancia históricamente dentro de la interacción humano - máquina: la Ley de Fitts y, a pesar de que no se obtuvo un modelo preciso de la interacción debido a diversas causas, como el no haber considerado el tamaño de pantalla o los ángulos en los que las personas giran para completar la tarea de apuntar con el dispositivo, se pudo comprobar lo establecido en el capítulo 1: el tamaño de los botones o elementos interactivos debe ser proporcional al uso que se les dará, lo cual implica que un elemento de tamaño más grande será más fácil y más rápidamente alcanzado que uno pequeño. Y por otro lado, al realizar la prueba para conocer los tiempos de carga óptimos, se logró obtener resultados que fueron consistentes con lo esperado, es decir que 1 segundo resultó mayormente rápido o muy rápido y que entre 3, 4 y 5 segundos pareció más adecuado para los usuarios.

Finalmente, la siguiente etapa en el desarrollo del presente trabajo, será analizar y discutir los resultados obtenidos para modificar el sistema de AR para incrementar su usabilidad.



Figura 142: Configuración utilizada para el experimento del tiempo de carga óptimo.

*When the right technology is set up properly, the impact is substantial.*[146]

Niklas Stratmann, Gerente de Innovación Empresarial  
en DB Schenker.

# 5

## Versión Preliminar

### 5.1. MODIFICACIONES AL SISTEMA DERIVADAS DE LAS PRUEBAS CON USUARIOS.

Como se mencionó en el capítulo anterior, se utilizarán los resultados obtenidos en las pruebas realizadas con usuarios para poder mejorar la usabilidad del sistema AR descrito en el capítulo 3.

Para comenzar, en el experimento de Fitts se obtuvo un modelo que no logró describir adecuadamente la interacción de tipo *raycast* debido a algunos factores, como son: el tamaño de pantalla utilizado, la utilización de ambas manos y los movimientos corporales involucrados (no únicamente los de las manos), sin embargo, se comprobó que la dificultad es inversamente proporcional al tamaño de los objetivos interactivos, razón por la cual se decidió emplear la escala más grande utilizada en el experimento, es decir el doble del tamaño original, debido a que al usar este tamaño de objetivo se conserva una distancia de separación entre cada una de las tres zonas interactivas, que le permite al usuario/ programa la diferenciación de las áreas y que el usuario reciba una info incorrecta, como se muestra con círculos azules en la Figura 143.

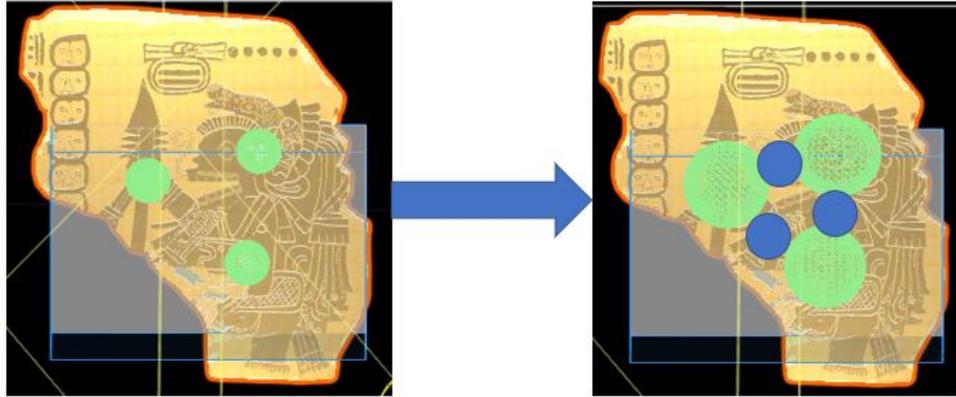


Figura 143: Modificación a los objetivos interactivos: Los círculos azules representan la separación entre objetivos.

De la misma manera, al observar los resultados de las dos pruebas realizadas para conocer el tiempo de carga óptimo en las interacciones *raycast*, en específico la gráfica de la Figura 141, en la cual se muestra el número de personas que respondieron que el tiempo presentado en la interacción les pareció adecuado; se observa que tres, cuatro y cinco segundos son los valores de tiempo que presentan una mayor frecuencia, por lo cual se decidió promediar los tres valores y se dejó fijo un valor de cuatro segundos para todas las interacciones de tipo *raycast*, como se ilustra en la Figura 144.



Figura 144: Modificación al tiempo de carga de los objetivos interactivos.

## 5.2. RESUMEN DEL FUNCIONAMIENTO DEL SISTEMA.

En el capítulo 3 se describieron todas las funciones del sistema AR y el proceso para crearlas a profundidad, sin embargo en esta sección se brindará un resumen con diagramas que describen gráficamente el funcionamiento de la aplicación.

Inicialmente, cuando un usuario abre la aplicación en un smartphone o dispositivo Android, se muestra una pantalla con el título del presente trabajo, los escudos de la UNAM y de la Facultad de Ingeniería, así como el botón "Comenzar"; no importa si el usuario está sosteniendo el dispositivo en modo *landscape* o *portrait*, pues ésta se ajusta automáticamente, como se ilustra en la Figura 145; una vez que el usuario presiona el botón para iniciar desaparece esa portada y aparece el *onboarding*, en el cual el usuario deberá ir pasando cada una de las seis pantallas que lo componen mediante un movimiento de tipo *slide* de derecha a izquierda para pasar a la siguiente pantalla o de izquierda a derecha para retroceder a la pantalla anterior.

En la última pantalla del *onboarding* se colocó un botón "Iniciar", mediante el cual se inicia la aplicación de AR, la cual también puede ser accesada en cualquier otra pantalla del *onboarding* al presionar el botón "Saltar" en forma de flecha en la esquina superior derecha para el modo *portrait* o en la esquina inferior derecha para el modo *landscape*, lo cual se ilustra en la Figura 146.

Una vez terminado o saltado el *onboarding* se tiene inmediatamente acceso a una pantalla con la imagen proveniente de la cámara posterior del dispositivo en el que se esté ejecutando la aplicación, esta pantalla cuenta con un punto rojo que marca el centro de la pantalla y botones con los que el usuario puede tener las seis interacciones siguientes:

- Apuntar una zona interactiva al mover su dispositivo.
- Apagar o encender la función de aparición de información interactiva.
- Volver al *onboarding*.
- Tomar una fotografía.
- Cambiar de cámara.
- Abrir la galería de la aplicación

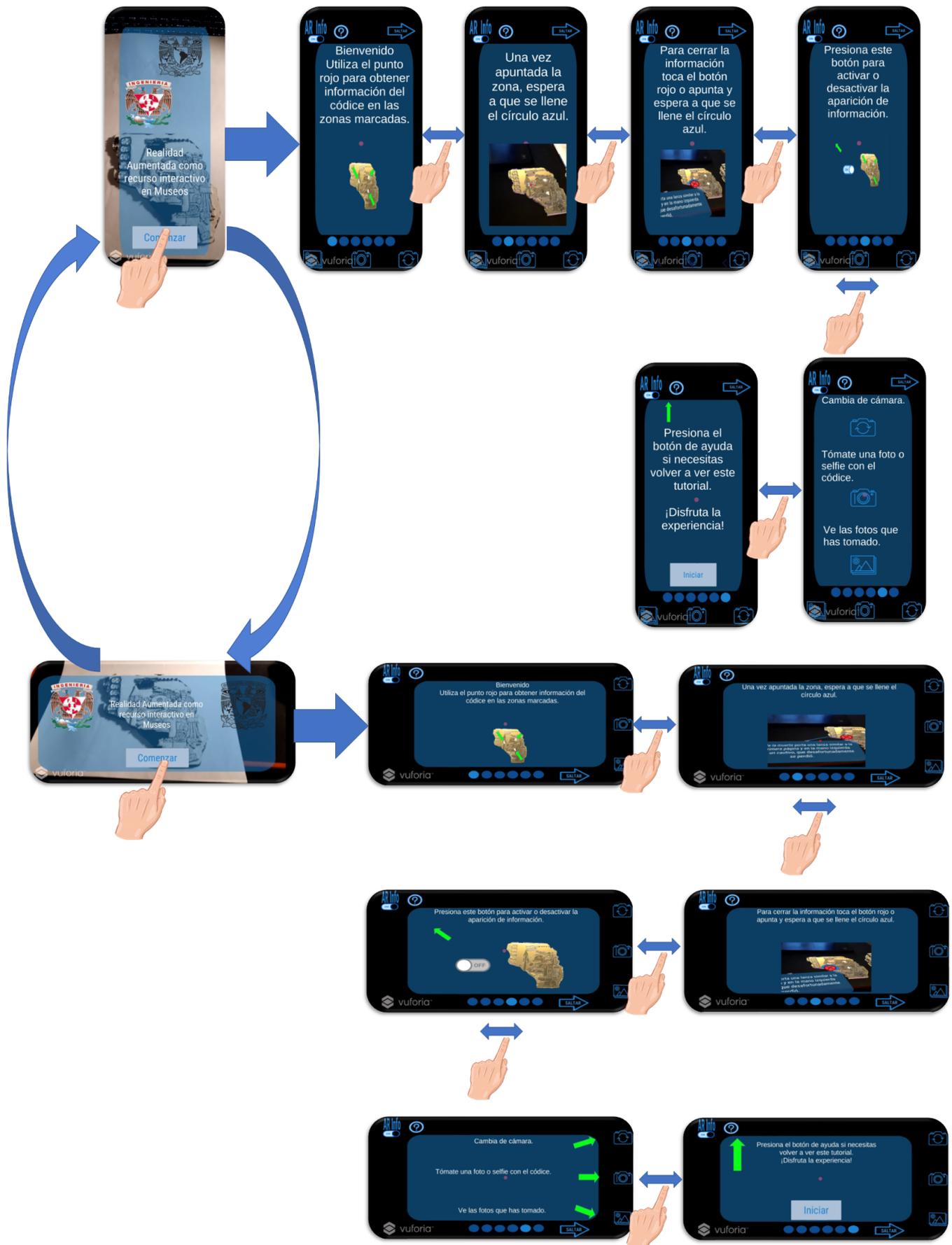
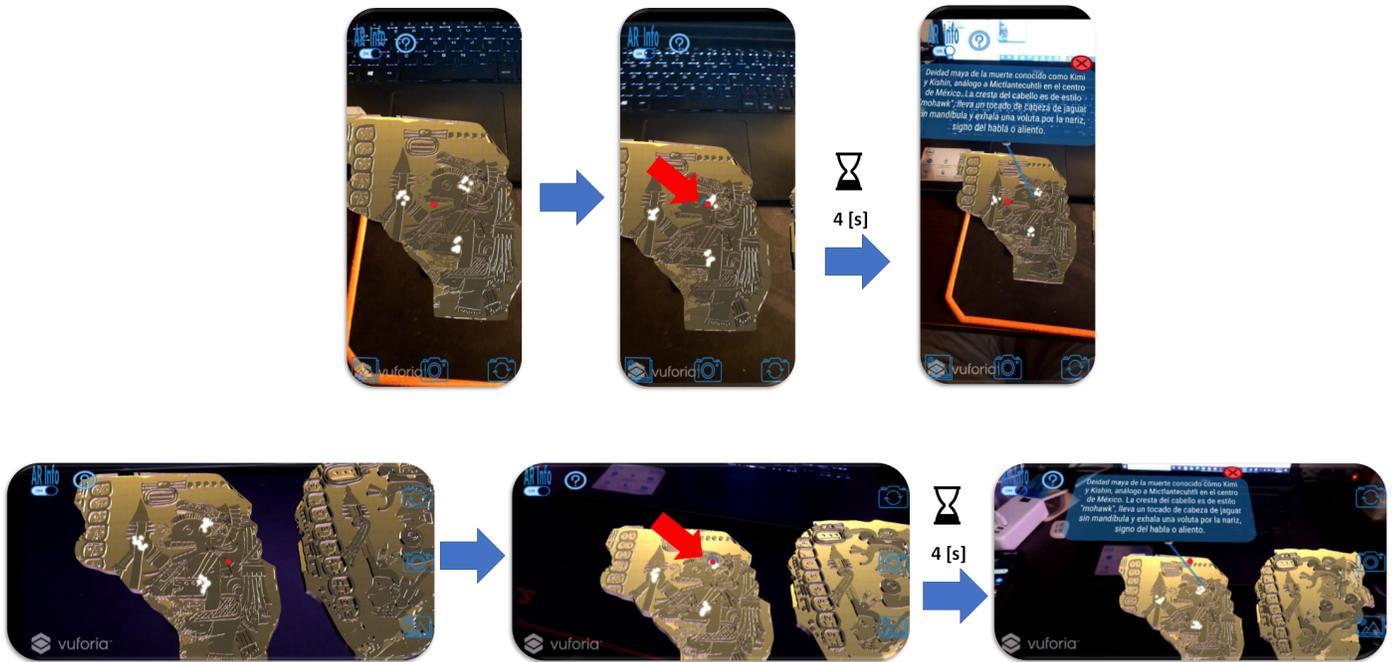


Figura 145: Primer diagrama de funcionamiento: Transición de la portada al onboarding.[18]



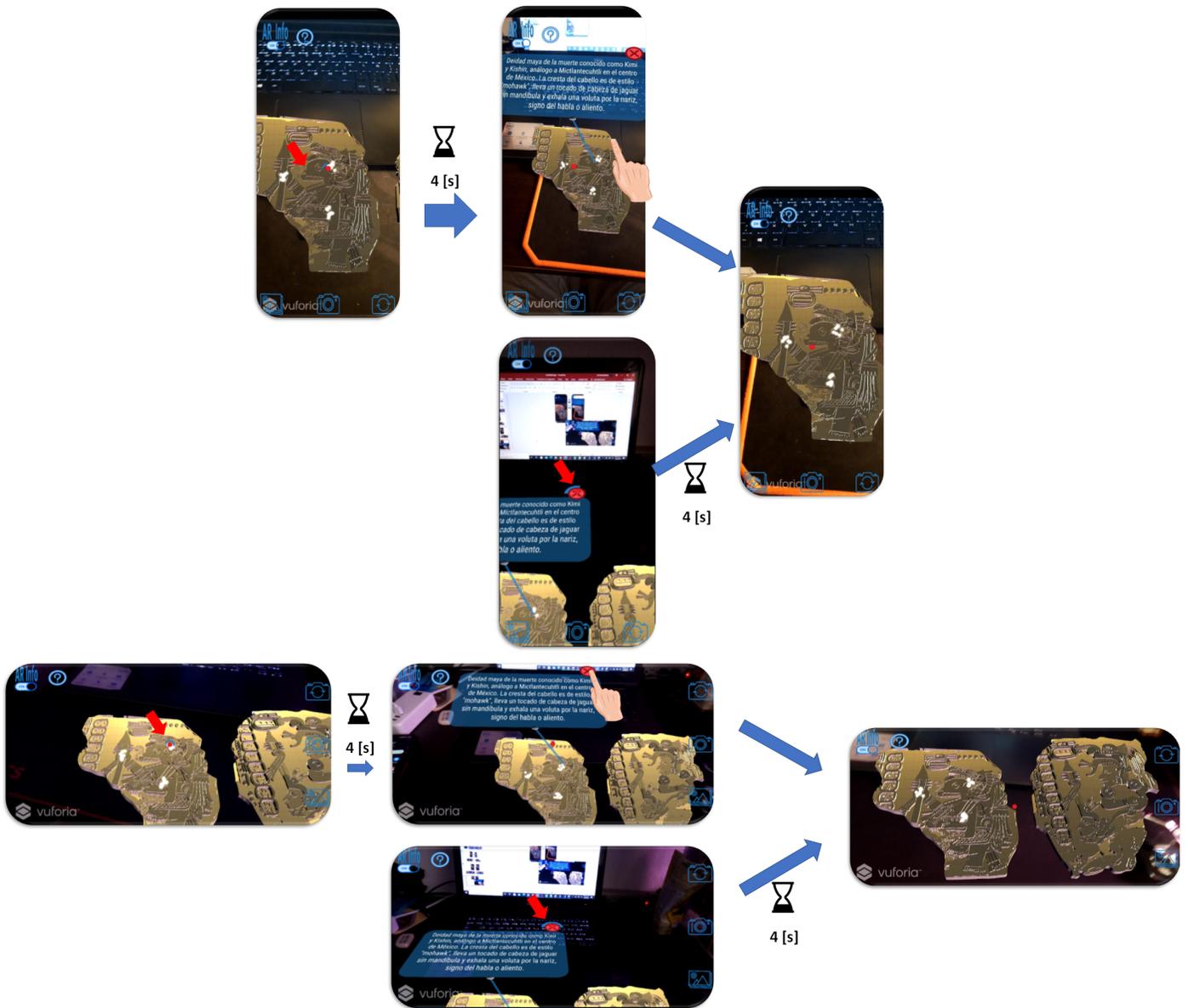
Figura 146: Segundo diagrama de funcionamiento: Transición del *onboarding* a la aplicación AR.[18]

La primera de las interacciones dentro de la aplicación AR consiste en apuntar con el dispositivo, ya sea para únicamente observar los folios virtuales del código o para obtener información de las zonas interactivas con las que éstos cuentan, utilizando el dispositivo como si fuera un visor mediante el cual aparecen los folios virtuales del código al apuntar con dicho dispositivo los marcadores reales (en papel o impresos en 3D) asociados a cada uno de los folios virtuales; cuando el usuario se encuentra visualizando alguno de los folios virtuales por primera vez observará que predeterminadamente se encuentra habilitada la opción de "AR Info", por lo que encima del código virtual podrá observar zonas interactivas, marcadas con partículas brillantes, a las cuales puede apuntar al alinear el centro de la pantalla con dicha área, con lo cual aparecerá un indicador radial de carga que se va llenando gradualmente mientras se siga apuntando a la misma área interactiva, hasta que transcurran cuatro segundos y aparezca un recuadro con información relevante de la parte a la cual se apuntó, como se muestra en la Figura 147.



**Figura 147:** Tercer diagrama de funcionamiento: Función apuntar zonas interactivas con el marcador del centro de la pantalla.

Una vez abierto el recuadro con información, éste se puede cerrar de dos maneras: apuntado con el centro de la pantalla el botón rojo con una cruz situado en la esquina superior derecha del recuadro y esperando cuatro segundos a que se llene por completo el indicador radial que aparece alrededor del botón, o bien tocando el mismo, como se muestra en la Figura 148.



**Figura 148:** Cuarto diagrama de funcionamiento: Cerrar el recuadro de información al apuntar con el centro de la pantalla o al tocar el botón rojo.[18]

La segunda de las interacciones posibles dentro de la aplicación AR consiste en apagar y encender las zonas interactivas de los folios virtuales, de modo que si un usuario desea únicamente tomar una fotografía del código sin las marcas de las zonas interactivas, lo puede hacer al presionar el botón de tipo *on/off* situado en la esquina superior izquierda de la pantalla etiquetado con la leyenda: *AR Info*, el cual cambia de color a gris cuando la opción se encuentra desactivada y vuelve a color azul cuando se activa, como se ilustra en la Figura 149.

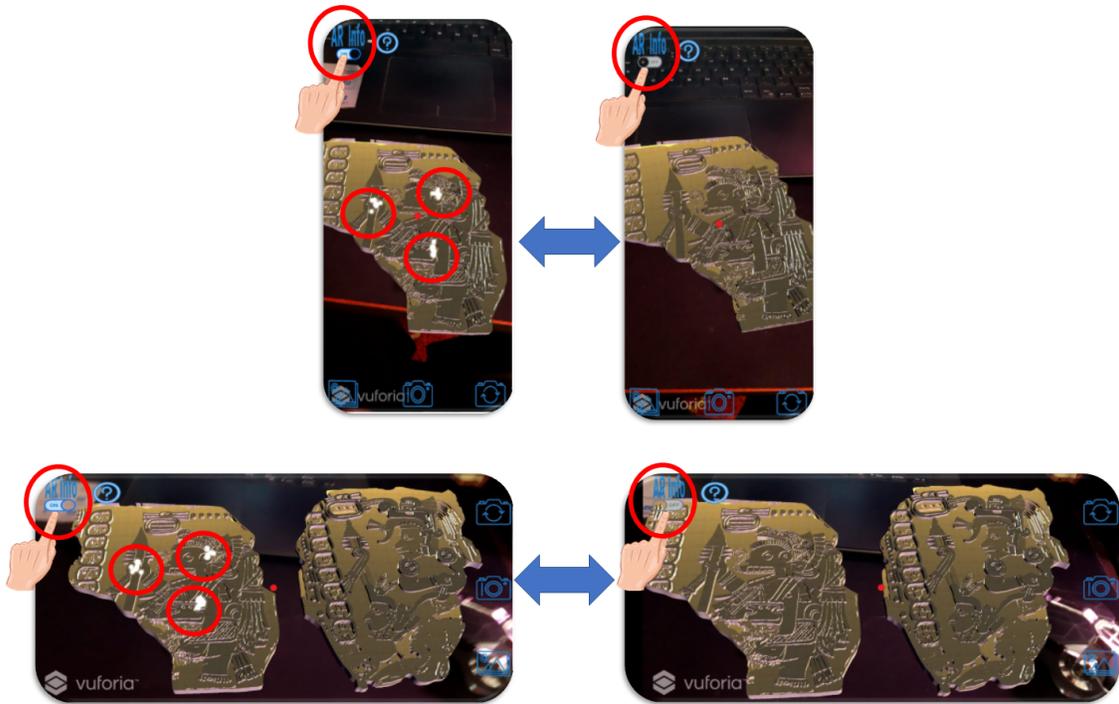


Figura 149: Quinto diagrama de funcionamiento: Habilitación o deshabilitación de las zonas interactivas.[18]

La tercera interacción posible estando en la pantalla principal de la aplicación AR consiste en volver al *onboarding* mediante el botón de ayuda identificado con un signo de interrogación, como se muestra en la Figura 150.



Figura 150: Sexto diagrama de funcionamiento: Transición de la aplicación AR al *onboarding*. [18]

La cuarta posible interacción con la pantalla principal de la aplicación AR es la de tomar una fotografía, mediante el botón con forma de cámara colocado en la parte central inferior de la pantalla en modo *portrait* o en la parte central derecha en modo *landscape*, esta función es independiente de si se ha detectado o no el código, por lo cual es posible capturar fotografías como en cualquier aplicación de cámara, aunque resulta claro que fue creada con el fin de que los usuarios puedan guardar en sus dispositivos fotografías del código o eventualmente compartirlas en alguna red social, aplicación de mensajería instantánea, etc., razón por la cual inmediatamente después de haberse tomado la fotografía se observa una completamente en blanco durante 150 milisegundos, para darle retroalimentación al usuario de que la fotografía está siendo tomada y posteriormente aparece la imagen capturada en un recuadro emergente, en el cual se muestra una vista previa de la fotografía que se acaba de tomar y también cuenta con dos botones: cerrar y compartir, lo cual se observa en la Figura 151, mientras que en la Figura 152 se observan las dos posibles interacciones del recuadro que contiene la vista previa de la fotografía recién tomada, las cuales son: al presionar el botón "cerrar" la ventana emergente se cierra, lo cual devuelve al usuario a la pantalla principal de la aplicación AR y al presionar el botón compartir se hace el llamado para que el dispositivo muestre su menú nativo mediante el cual se puede compartir la fotografía en cualquier otra aplicación que lo permita.

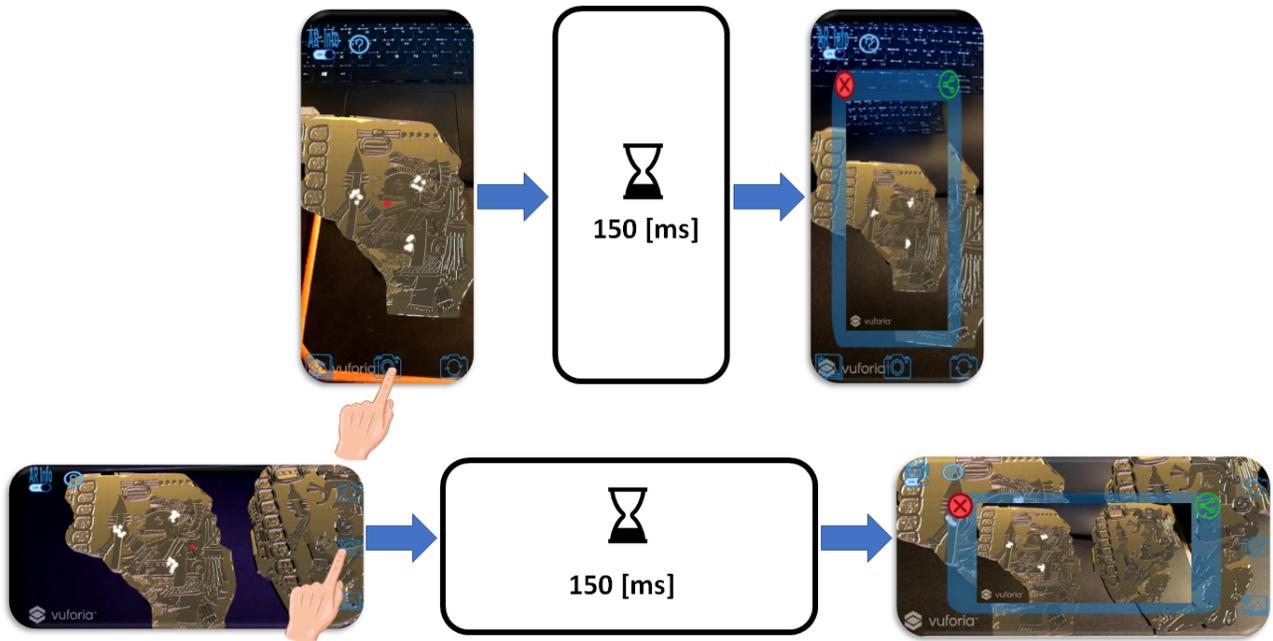


Figura 151: Séptimo diagrama de funcionamiento: Toma de fotografía.[18]

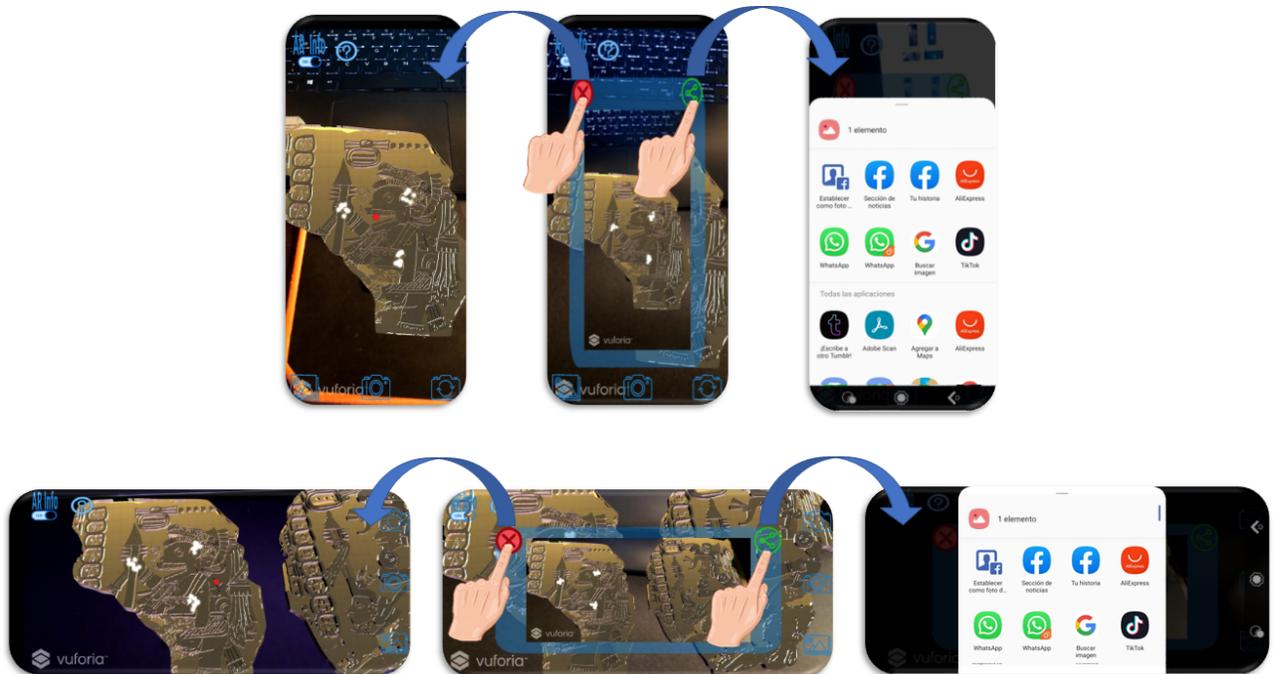


Figura 152: Octavo diagrama de funcionamiento: Interacciones con la ventana emergente después de la toma de fotografía.[18]

La quinta interacción posible dentro de la pantalla principal consiste en poder cambiar de la cámara posterior a la frontal y viceversa, utilizando el botón con un ícono de cámara y dos flechas colocadas en círculo en la esquina inferior derecha en modo *portrait* o en la esquina superior derecha en modo *landscape*, de tal modo que los usuarios puedan capturar tanto fotografías como *selfies* con los folios virtuales, como se ilustra en la Figura 153.

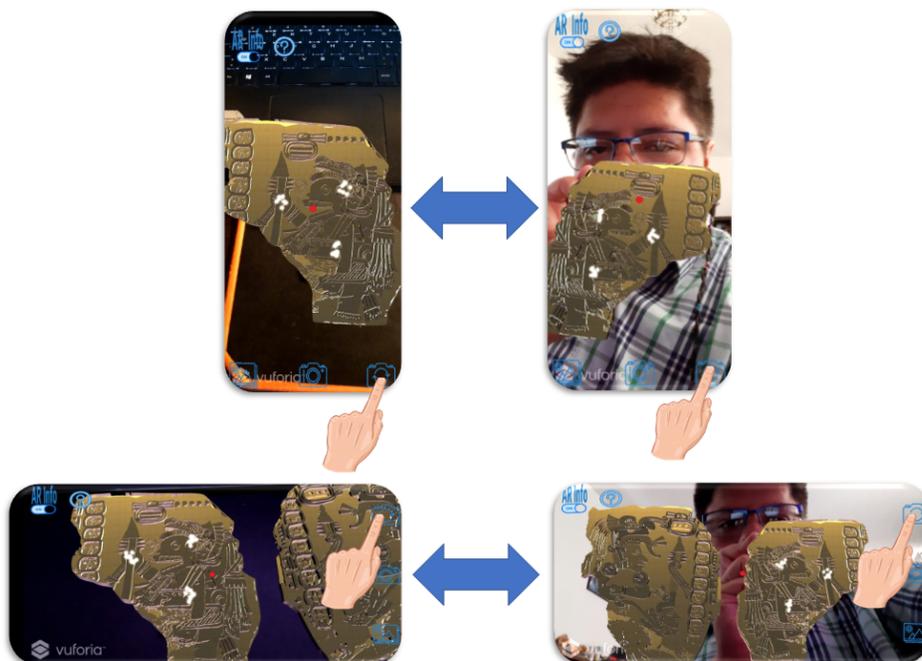


Figura 153: Noveno diagrama de funcionamiento: Cambio de cámara.[18]

Finalmente, la sexta posible interacción consiste en presionar el botón que representa un ícono de galería situado en el modo *portrait* en la esquina inferior izquierda y en modo *landscape* en la esquina inferior derecha; al presionar este botón se mostrarán las fotografías tomadas en la sesión actual, las cuales será posible visualizar una por una al tocarlas y hacer un movimiento de tipo *slide* hacia los lados para pasar a la siguiente o anterior fotografía; si el usuario llegase a cerrar la aplicación y quiere volver a ver las fotografías de la sesión pasada se tendrá que ir directamente a la galería del dispositivo y no a la función de galería dentro de la aplicación, pues ésta almacena todas las imágenes en una carpeta especial de nombre "ARCodex" dentro de la galería de Android, pero únicamente muestra las fotografías de la sesión en curso dentro de la galería interna de la aplicación, cabe mencionar que si no se han tomado fotografías en la sesión actual y el usuario presiona el botón de galería aparecerá un mensaje que dice: "No has tomado fotos en esta sesión. Las fotos tomadas en sesiones pasadas se encuentran en la carpeta "ARCodex" de tu galería.", como se ilustra en la Figura 154.

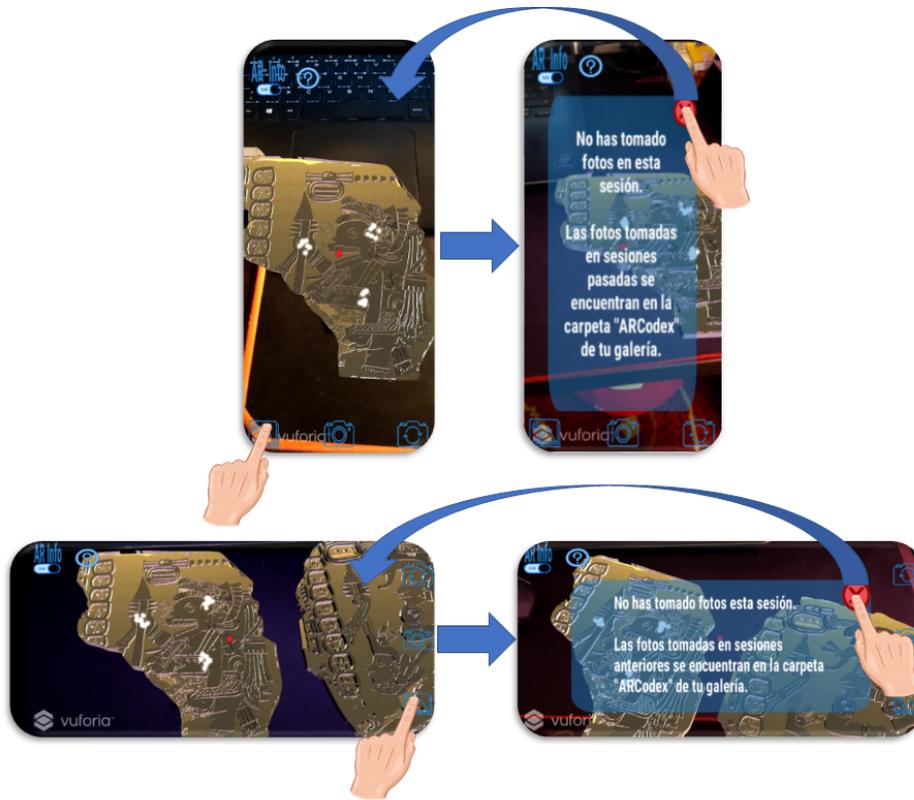
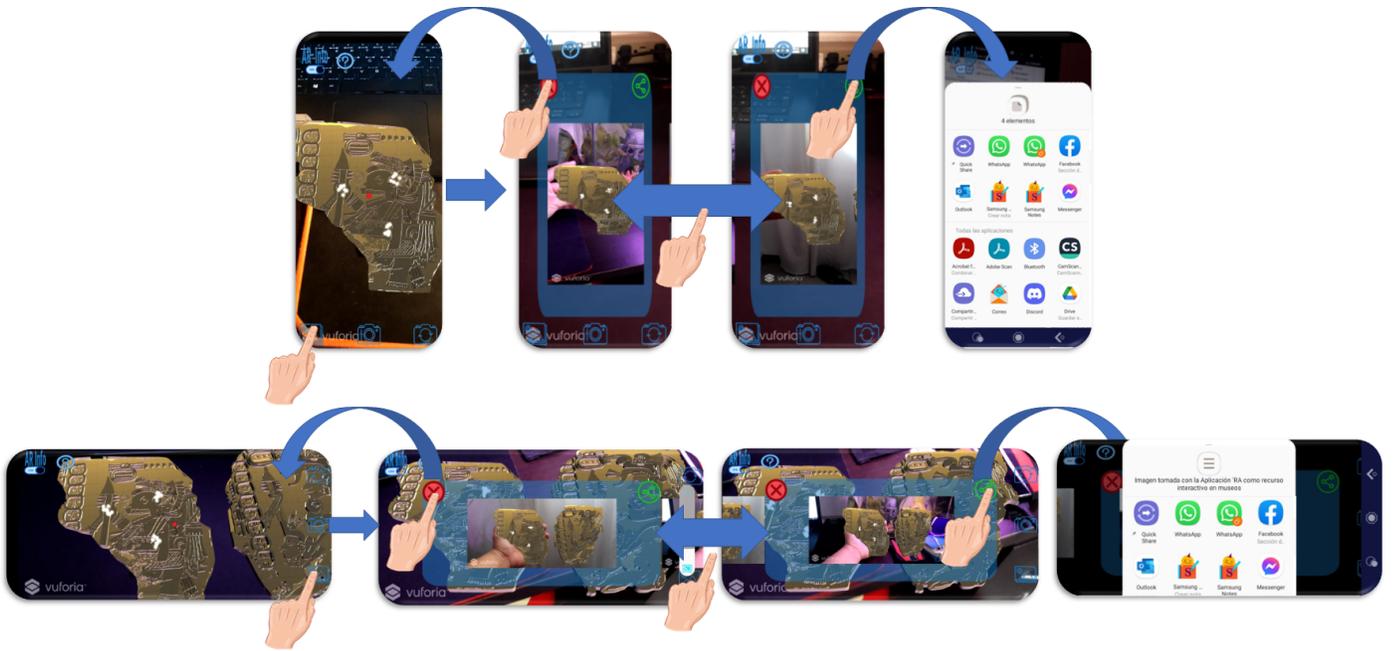


Figura 154: Décimo diagrama de funcionamiento: Galería interna.[18]

Por otro lado, el recuadro que contiene las fotografías anteriormente tomadas contiene dos botones: cerrar y compartir, con los cuales se cierra el recuadro y se comparten todas las fotografías tomadas respectivamente, como se ilustra en la Figura 155.



**Figura 155:** Décimo diagrama de funcionamiento: Interacciones con el recuadro de la galería.[18]

Como se puede observar hasta este punto, la navegación a través de la aplicación en ambos modos fue elaborada meticulosamente, con la finalidad de eliminar cualquier posible zona en la que el usuario pudiera quedarse sin salida, mejorando así la usabilidad de la aplicación; de igual manera, al tener habilitada la opción de "AR Info" desde que se abre por primera vez la aplicación, se disminuye el número de interacciones que el usuario debe tener en los primeros momentos al ejecutar la aplicación, con lo cual también se pretende apoyar la usabilidad, razones por las cuales restaría evaluar con usuarios qué tan adecuada resultó la usabilidad del sistema mediante alguna escala validada para medirla, como lo es la *System Usability Scale*, lo cual se analizará en el siguiente apartado.

### 5.3. APLICACIÓN DE LA ESCALA DE USABILIDAD DEL SISTEMA.

La Escala de Usabilidad del Sistema o *System Usability Scale (SUS)* en inglés, es una eficiente herramienta de investigación que consiste en un cuestionario estandarizado de diez preguntas, cada una de las cuales cuenta con cinco posibles respuestas que van de totalmente de acuerdo a totalmente en desacuerdo.

La SUS fue desarrollada por John Brook en 1986 para evaluar la usabilidad de aplicaciones para las terminales VT100, pero debido a que ésta es independiente de la tecnología, ha sido utilizada para probar una amplia gama de productos y servicios que incluyen hardware, software, dispositivos móviles, sitios web, e incluso en guías telefónicas (*yellow pages*) etc.[147][148]

La SUS es una de las herramientas más confiables para medir y cuantificar la percepción de usabilidad por parte de los usuarios y ha sido ampliamente usada en el ámbito de la HCI, debido a que su aplicación resulta sencilla y rápida y los resultados que se obtienen son bastante fiables y fáciles de interpretar.[149]

Dentro de la ACM (*Association for Computing Machinery*) existe el SIGCHI (*Special Interest Group on Computer - Human Interaction*), el cual ha presentado desde 1982 una conferencia anual (*Conference on Human Factors in Computing Systems*), abreviada como CHI, la cual es una de las más prestigiosas conferencias en el área de ciencias de la computación[150], en la cual se presentan algunos de los trabajos más innovadores dentro de la HCI y dentro de una buena cantidad de los trabajos presentados en la CHI, la SUS ha sido ampliamente utilizada por investigadores de todo el mundo en conjunto con algunos otros métodos de evaluación para valorar la usabilidad de nuevos conceptos y validar los resultados de sus investigaciones, además de que este método de evaluación se ha convertido en un estándar de la industria, teniendo referencias en más de 600 publicaciones.[148]

Debido a que la SUS es un cuestionario estandarizado, ha sido traducida cuidadosamente a diferentes idiomas y al emplearla no se debe modificar el orden de aparición de las preguntas, ni ninguna palabra, pues cada una de las preguntas están intencionalmente diseñadas de modo que la puntuación sea correctamente calculada.[151]

El cuestionario de la SUS original en inglés está compuesta por las siguientes preguntas:

The System Usability Scale Standard Version		Strongly Disagree					Strongly Agree				
		1	2	3	4	5	1	2	3	4	5
1	I think that I would like to use this system frequently.	<input type="radio"/>									
2	I found the system unnecessarily complex.	<input type="radio"/>									
3	I thought the system was easy to use.	<input type="radio"/>									
4	I think that I would need the support of a technical person to be able to use this system.	<input type="radio"/>									
5	I found the various functions in this system were well integrated.	<input type="radio"/>									
6	I thought there was too much inconsistency in this system.	<input type="radio"/>									
7	I would imagine that most people would learn to use this system very quickly.	<input type="radio"/>									
8	I found the system very awkward to use.	<input type="radio"/>									
9	I felt very confident using the system.	<input type="radio"/>									
10	I needed to learn a lot of things before I could get going with this system.	<input type="radio"/>									

Figura 156: System Usability Scale en su idioma original.[152]

Por otro lado, el cuestionario correspondiente a la SUS traducida al español está compuesta por las siguientes preguntas:

		Muy en desacuerdo					Totalmente de acuerdo				
		1	2	3	4	5	1	2	3	4	5
1.	Creo que usaría este sistema frecuentemente.	<input type="checkbox"/>									
2.	Encuentro este sistema innecesariamente complejo.	<input type="checkbox"/>									
3.	Creo que el sistema fue fácil de usar.	<input type="checkbox"/>									
4.	Creo que necesitaría ayuda de una persona con conocimientos técnicos para usar este sistema.	<input type="checkbox"/>									
5.	Las funciones de este sistema están bien integradas.	<input type="checkbox"/>									
6.	Creo que el sistema es muy inconsistente.	<input type="checkbox"/>									
7.	Imagino que la mayoría de la gente aprendería a usar este sistema en forma muy rápida.	<input type="checkbox"/>									
8.	Encuentro que el sistema es muy difícil de usar.	<input type="checkbox"/>									
9.	Me siento confiado al usar este sistema.	<input type="checkbox"/>									
10.	Necesité aprender muchas cosas antes de ser capaz de usar este sistema.	<input type="checkbox"/>									

Figura 157: System Usability Scale traducida al español.[153]

Como se puede observar, las preguntas pares expresan actitudes negativas, mientras que las nones expresan actitudes positivas, por lo cual una vez obtenidas las respuestas es necesario convertir la escala mediante un sencillo algoritmo que

consta de restar un punto a las respuestas nones y con las respuestas pares se resta el valor de la respuesta a cinco, con lo cual se tendrá una nueva escala que va de cero a cuatro, posteriormente se suman todos los valores y se multiplica el resultado por dos punto cinco, para así tener una escala que irá de cero a cien, como se ejemplifica a continuación; los resultados obtenidos pueden ser fácilmente clasificados utilizando la siguiente tabla, con la cual se pueden comunicar los resultados fácilmente a posibles *stakeholders*. [151][152]

Para ejemplificar el uso del algoritmo de la SUS se supondrá que se obtuvieron los siguientes valores [149]:

3, 4, 3, 5, 3, 2, 1, 2, 4, 5

Y se aplicará el algoritmo de la siguiente manera:

$$((3 - 1) + (5 - 4) + (3 - 1) + (5 - 5) + (3 - 1) + (5 - 2) + (1 - 1) + (5 - 2) + (4 - 1) + (5 - 5)) * 2.5 \text{ (ec. 6)}$$

$$(2 + 1 + 2 + 0 + 2 + 3 + 0 + 3 + 3 + 0) * 2.5 = 40 \text{ (ec. 7)}$$

Por lo cual, utilizando la tabla siguiente, se entendería que el sistema del cual se obtuvieron los resultados tiene una muy mala usabilidad.

Puntaje SUS	Calificación Letra	Adjetivo clasificatorio
Mayor a 80.3	A	Excelente
Entre 68 y 80.3	B	Buena
68	C	OK
Entre 51 y 67	D	Pobre
Menor a 51	F	Horrible

Tabla 2: Evaluación de los resultados de la *System Usability Scale*. (Traducida por el autor)[152]

Adicionalmente, existen otras maneras muy similares de evaluar los resultados de la aplicación de una SUS, las cuales también consisten en asignarle un adjetivo calificativo a rangos comprendidos dentro de los valores posibles como resultado de la SUS (0 - 100); en la mayoría de las tablas para evaluar los resultados de la SUS se considera que por encima de 80 puntos es un muy buen resultado, aunque existen pequeñas variaciones en los rangos, como por ejemplo en la siguiente imagen de

Adobe Xd, el cual es un servicio de Adobe para crear prototipos de interfaces para aplicaciones o servicios de todo tipo de manera sencilla y rápida, en la cual se considera como una excelente usabilidad a partir de 85.5 y no a partir de 80.3.

## Escala de Usabilidad del Sistema

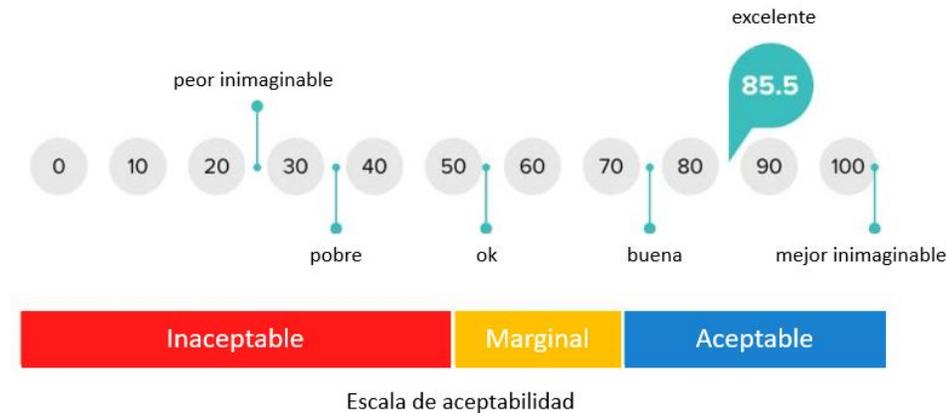


Figura 158: Evaluación de los resultados de la *System Usability Scale*. (Traducida por el autor)[154]

Es por todo lo anteriormente mencionado que se decidió utilizar la SUS para evaluar la usabilidad de la experiencia AR creada en el presente trabajo y de ese modo valorar si se logró el objetivo inicial de crear una experiencia AR con buena usabilidad, dado que su creación estuvo guiada en todo momento por los principios que apoyan la usabilidad y algunos otros dentro del ámbito de la HCI.

### 5.3.1. RESULTADOS DE LA APLICACIÓN DE LA ESCALA DE USABILIDAD DEL SISTEMA

Debido a la pandemia ocasionada por la COVID-19, la aplicación de la SUS fue realizada exclusivamente con personas que tenían contacto cercano con el autor y no a las afueras de algún museo, como hubiera sido deseable, desinfectando el smartphone entre cada uno de los usuarios y respetando el uso del cubrebocas, así como el distanciamiento social; al iniciar, se le explicaba al usuario que primero aparecería una portada, seguida de un tutorial que debía leer y una vez finalizado éste, ya se podrían realizar las interacciones con la aplicación; se les solicitó a los usuarios realizar todas las interacciones que desearan hasta sentir que habían explorado completamente el funcionamiento de la aplicación y posteriormente se les pidió contestar el cuestionario de la SUS mostrado en la Figura 157.

En la Figura 159 se muestran algunas fotografías de las personas encuestadas.

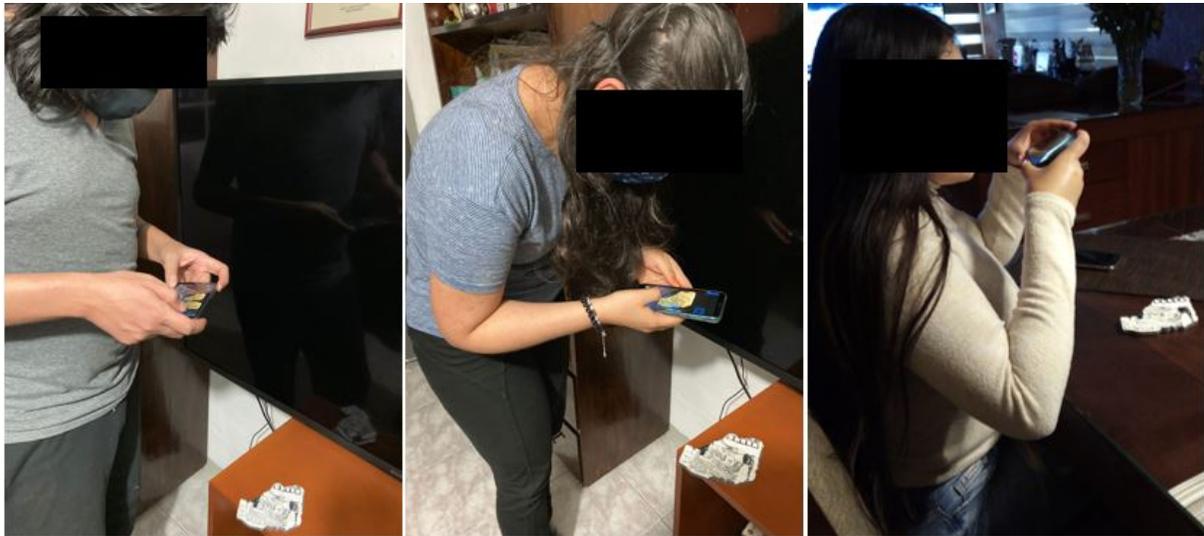


Figura 159: Aplicación de la SUS.

Para determinar a cuántas personas se les debería aplicar la SUS, se debería haber tomado como referencia el número de visitantes que tuvo la exposición temporal "El Códice Maya de México" en el Museo Nacional de Antropología, siendo éstos la cantidad de 100, 415 [155] y realizando el cálculo del tamaño de muestra mediante la ecuación 8, con un 95 % de nivel de confianza y un margen de error del 5 %, se necesitarían 383 personas y entrevistar a ese número de personas teniendo en cuenta la situación actual de contagios ocasionados por el SARS-CoV-2 podría resultar peligroso para todos los entrevistados y su círculo social, por lo cual se redujo el nivel de confianza a 80 % y partiendo de que se les podía aplicar la encuesta a quince personas, debido a que éstas eran el círculo social cercano del autor, se realizó el cálculo del error que se tendría, lo cual se muestra en la ecuación 9, obteniendo como resultado un margen de error de 16.5 %.

Para realizar los cálculos anteriormente descritos se utilizó la siguiente ecuación, mediante la cual se puede obtener el tamaño de muestra de una población finita.

$$n = \frac{N * Z_{\alpha}^2 * p * q}{e^2 * (N - 1) + Z_{\alpha}^2 * p * q} \quad (\text{ec. 8})$$

Donde: n es el tamaño de muestra.

N es el tamaño de la población.

$Z_{\alpha}$  es el parámetro estadístico que depende del nivel de confianza, obtenido mediante una tabla de distribución normal.

p es la probabilidad de que el evento estudiado suceda.

q es la probabilidad de que el evento estudiado no suceda. (1-p)  
 e es el error de estimación máximo aceptado.

$$e = \sqrt{\frac{(N * Z_{\alpha} - n)(Z_{\alpha} * p * q)}{n(N - 1)}} = \sqrt{\frac{(100415 * 1.28 - 15)(1.28 * 0.5 * 0.5)}{15(100415 - 1)}} = 0.1652 = 16.52 \% \quad (\text{ec. 9})$$

Adicionalmente, utilizando la herramienta para el cálculo de tamaño de muestra de SurveyMonkey (software para creación de encuestas)[156], se ingresaron los valores conocidos de tamaño de población, nivel de confianza y margen de error, logrando corroborar el resultado obtenido anteriormente, como se puede observar en la Figura 160, con la excepción de que dicha herramienta únicamente toma en cuenta valores enteros, por lo cual el margen de error se redondearía a 17 %.

The image shows a web-based calculator for determining sample size. The title is "Calcula el tamaño de la muestra". There are three input fields: "Tamaño de la población" (Population size) with the value 100415, "Nivel de confianza (%)" (Confidence level) with a dropdown menu set to 80, and "Margen de error (%)" (Margin of error) with the value 17. Below these fields, the result is displayed as "Tamaño de la muestra" (Sample size) with a large green number "15".

Figura 160: Cálculo del tamaño de muestra.[156]

Tomando en cuenta todo lo anterior, se le realizó la prueba a quince personas y posteriormente se vació el puntaje de cada pregunta a una hoja de cálculo, con la finalidad de poder aplicar fácilmente el algoritmo mencionado anteriormente para hacer el cálculo del puntaje de cada una de las encuestas, como se muestra en la Figura 161, en la cual se puede observar la primera parte del algoritmo en la barra de fórmulas; por otro lado, en cada columna de izquierda a derecha se encuentran los puntajes obtenidos por persona a cada una de las diez preguntas, mientras que en las columnas K y L, se muestra el total después de la primera parte del algoritmo y el total después de haber realizado la multiplicación por 2.5 del algoritmo, y finalmente en la columna M se observa el resultado de promediar los totales correspondientes a una escala de cero a cien.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	1	2	3	4	5	6	7	8	9	10	Total	Total escala 100	Promedio
2	3	2	4	5	4	3	5	1	4	1	28	70	85.833333
3	5	1	4	1	4	1	5	1	5	1	38	95	
4	3	1	5	1	2	2	4	1	2	1	30	75	
5	4	2	5	2	4	1	5	4	5	1	33	82.5	
6	4	1	4	2	4	2	3	1	4	1	32	80	
7	5	4	5	1	3	4	5	1	4	1	31	77.5	
8	5	1	5	1	5	1	5	1	5	1	40	100	
9	3	1	5	3	5	1	5	1	2	2	32	80	
10	4	1	5	1	5	1	5	1	5	1	39	97.5	
11	5	1	5	2	5	1	5	1	3	1	37	92.5	
12	5	3	4	2	5	2	4	3	4	2	30	75	
13	5	1	5	1	5	1	5	1	5	1	40	100	
14	4	1	5	5	5	1	4	1	5	5	30	75	
15	4	3	5	1	5	1	5	1	5	1	37	92.5	
16	5	1	3	1	5	1	5	1	5	1	38	95	

Figura 161: Resultados de la SUS.

Como es posible observar en la última columna, se obtuvo un promedio de 85.8, lo cual indica que la usabilidad de la aplicación es excelente, de acuerdo a las escalas de la Tabla 2 y la Figura 158, lo cual permite afirmar que el haber utilizado los principios, reglas y buenas prácticas para apoyar la usabilidad y la experiencia del usuario cumplió su cometido.

# 6

## Conclusiones

Durante el desarrollo de la aplicación creada, la fase de pruebas y posterior análisis, se pudieron obtener las siguientes conclusiones:

- En la creación de cualquier herramienta que será utilizada por un usuario, resulta indispensable el uso de principios para apoyar en todo momento la usabilidad, en este caso en específico acerca del desarrollo de una aplicación de AR para museografía, el enfoque tomado hacia tener una óptima usabilidad del sistema, siguiendo todos los principios, reglas, buenas prácticas, guías, etc. encaminados a apoyar y mejorar la usabilidad y la experiencia del usuario provocó que, utilizando como herramienta la *System Usability Scale* (SUS) se pudiera obtener un resultado que apunta a que la usabilidad de la aplicación es excelente, pues la media de los resultados fue de 85.8 de 100 puntos, valor que en dos de las fuentes consultadas sobrepasa el valor mínimo para que se considere que el sistema analizado cuenta con una excelente usabilidad.

- Del experimento de Fitts realizado se obtuvieron dos observaciones principales, la primera de ellas es que como era

esperado, el tamaño de los objetivos que se deben alcanzar, tocar o apuntar, dependiendo del tipo de interacción que se esté analizando, es inversamente proporcional al tiempo que tardarán los usuarios en alcanzarlos, es decir a la dificultad; y la segunda de las observaciones fue que al querer utilizar directamente el modelo de Fitts original, concebido para interacciones 2D como por ejemplo: apuntar con un *mouse* o un *trackpad*, hubo variables que se dejaron de lado y en un futuro se deberían considerar acerca de la interacción de tipo AR, ya que en ésta el usuario puede girar sobre su propio eje, cambiar la posición de sus brazos, las variantes en los tamaños de pantalla, etc., lo cual resultó en que se obtuviera una baja correlación (0.4027) al graficar los datos y querer obtener la ecuación que caracteriza el dispositivo y la interacción AR. Sin embargo, se puede concluir que en el desarrollo de herramientas para usuarios es de vital importancia tomar en cuenta la existencia de la Ley de Fitts, pues ésta forma parte importante de los elementos que apoyan la usabilidad de los sistemas desarrollados.

- La intención del experimento propuesto inicialmente para saber cuál sería el tiempo de carga óptimo para la aparición de información era brindarle todas las opciones de respuesta posibles para que los usuarios, al mover un *slider*, pudieran hacer la interacción de carga de información varias veces, modificando el tiempo hasta que se sintieran cómodos con la duración del tiempo de carga, pero en la práctica se observó que al tener tantas posibilidades para colocar el slider entre cero y cinco segundos, los usuarios se confundían y preguntaban por retroalimentación, por lo cual se tuvo que cambiar el experimento, fijando el número de opciones de tiempos posibles y preguntándole a los usuarios un calificativo al tiempo que se les acababa de mostrar, con lo cual ya no se presentaron confusiones entre los participantes, lo cual permite comprobar que a pesar de que varios de los principios para apoyar la usabilidad apuntan a que hay que brindarle opciones a los usuarios, se debe alcanzar un balance óptimo para no sobrecargar al usuario y limitar el número de opciones que se les brinda, lo cual se menciona en una de las ocho reglas de oro para el diseño de interfaces, la cual dice que se debe reducir la carga en la memoria de corto plazo de los usuarios y por otro lado la Ley de Hick es similar, pues dice que el tiempo necesario para tomar una decisión es directamente proporcional al número de opciones y la complejidad disponibles[157].

- Uno de los aspectos que más relevancia tuvo en el desarrollo de la aplicación AR y en el que se puso especial interés y tiempo de desarrollo fue la adaptabilidad, que es algo fundamental en todo servicio digital, ya que los usuarios deben ser capaces de disfrutar el contenido de manera adecuada en cualquier dispositivo en el que éste se encuentre disponible, no importando la resolución de la pantalla o la velocidad de procesamiento del dispositivo, con la finalidad de alcanzar el mayor número de usuarios posible, sin embargo, como todo servicio digital actual, existirán siempre elementos de mejora a considerar dentro de la aplicación con la finalidad de ampliar el rango de usuarios a los que se desearía alcanzar, así como para mejorar aún más la usabilidad del sistema.

- Como se observó en el marco teórico, existe un gran número de ejemplos de AR en cada vez más ámbitos, pero especialmente en museos hace que sea cada vez más atractivo este tipo de atracciones para complementar la experiencia de los visitantes, sin embargo no únicamente la AR puede mejorar la experiencia en dichos lugares sino también otros tipos de tecnologías inmersivas, como la VR, hologramas y tecnología háptica podrían ser aprovechadas para crear nuevas formas de interactuar en los museos y revolucionar la experiencia en dichos recintos, con la finalidad de que muchas más personas se interesen en visitarlos y en aprender en su visita de una manera mucho más sencilla; si bien existen numerosos ejemplos de AR aplicada a visualización en museos, esto aún no se convierte en un estándar a nivel nacional ni mundial, por lo que se da una base para promover el uso de la AR para complementar la visualización de información de manera tradicional, así como una alternativa dirigida a museógrafos para curar futuras exhibiciones. Por otro lado, al realizar los experimentos y la aplicación de la SUS se observó que a los usuarios les pareció agradable e interesante la interacción creada al poder ellos tocar una versión impresa en 3D del código y al mismo tiempo poder visualizarlo en su versión digital mediante el uso de un dispositivo móvil.

- Y por último, la aplicación creada demuestra que es posible incorporar este tipo de tecnología como un elemento más en museografía, ya que el funcionamiento de la aplicación creada podría ser extrapolado a cualquier otro tipo de exposiciones o eventos culturales, principalmente en los cuales no es posible exponer las piezas históricas a ningún tipo de deterioro, con la finalidad de incrementar el interés de los visitantes y mejorar su experiencia durante la visita.



# 7

## Trabajo a futuro

Durante el desarrollo del presente trabajo se identificaron varios aspectos a mejorar en alguna versión posterior, como por ejemplo:

- Mejorar el tutorial u *onboarding* de tal modo que los usuarios puedan realizar las interacciones al mismo tiempo que van utilizando el tutorial por primera vez, ya que durante la aplicación de la SUS se identificó que los usuarios estaban buscando esa funcionalidad.
- Crear un logo y colocarlo como ícono de la aplicación, mediante el cual se identificaría más fácilmente que el logo genérico que proporciona el entorno de programación de manera predeterminada.
- Mejorar la pantalla de la galería interna de la aplicación, debido a que si se está en la primera foto y se hace el movimiento para ver la fotografía anterior, se crea un espacio vacío.

- Mejorar la detección de la posición del smartphone, debido a que utilizando las funciones por *default* del entorno de programación para realizar la detección, en ocasiones hay que girar un poco más el dispositivo para que se detecte correctamente la orientación: *landscape* o *portrait*.
- Durante la última modificación al código de la aplicación se detectó que debido a la actualización de seguridad de Android en abril de 2021, la aplicación ya no se podía utilizar, debido a que no se había firmado, por lo cual algo a mejorar sería tener en cuenta este punto para su posible futura publicación en alguna tienda de aplicaciones.
- Durante el desarrollo del trabajo se detectó que a pesar de que Vuforia ofrece una rápida integración de la detección de marcadores, con la cual se puede acelerar el proceso de creación de aplicaciones de AR, en ocasiones la detección no es óptima, dependiendo de la iluminación, movimiento, etc., lo cual puede afectar el nivel de inmersión en la AR que tengan los usuarios, por lo cual se debería cambiar a otra de las plataformas que incluso son capaces de hacer un mapeo del lugar, con lo cual se podrían agregar elementos que contribuyan a la inmersión, como podría ser la oclusión, sin embargo esta tecnología únicamente es compatible actualmente con dispositivos de gama alta, por lo cual se tendría que evaluar esta posibilidad conforme al desarrollo de la tecnología de los dispositivos móviles.
- Si se deseara utilizar la aplicación como parte de la exposición "El Códice Maya de México" se debería realizar el modelado de los otros siete folios restantes del códice, así como crear las zonas interactivas para cada uno de ellos, colocar la información requerida en un diccionario para que ésta pueda ser visualizada y habilitar la interacción de tipo *raycast* para todas las piezas faltantes.
- Otra función que se podría implementar sería la de añadir marcadores de tipo código QR o imágenes, de tal modo que la aplicación desarrollada pudiera utilizarse más fácilmente en cualquier otro lugar, sin tener la necesidad de tener un modelo impreso en 3D para poder acceder a las funciones interactivas.

# Referencias

- [1] A. Dix, J. Finlay, G. D. Abowd, and R. Beale, *Human-Computer Interaction*. Essex, Inglaterra: Pearson Prentice Hall, 2004, pp. 3–5.
- [2] Interaction Design Foundation. (2016) Human computer interaction. Recuperado el 13/10/2020. [Online]. Available: <https://www.interaction-design.org/literature/topics/human-computer-interaction>
- [3] T. T. Hewett, R. Baecker, S. Card, T. Carey, J. Gasen, M. Mantei, G. Perlman, G. Strong, W. Verplank, and B. Hefley, *ACM SIGCHI Curricula for Human Computer Interaction*. Nueva York, Estados Unidos: The Association for Computing Machinery, 1992, pp. 5–6.
- [4] (2019) Significado de Interfaz. Recuperado el 20/05/2022. [Online]. Available: <https://www.significados.com/interfaz/>
- [5] usability.de GmbH Co. KG. (2013) Usability & user experience. Recuperado el 17/11/2020. [Online]. Available: <https://www.usability.de/en/usability-user-experience.html>
- [6] A. Dix, J. Finlay, G. D. Abowd, and R. Beale, *Human-Computer Interaction*. Essex, Inglaterra: Pearson Prentice Hall, 2004, pp. 201–202.
- [7] Interaction Design Foundation. (2017) User centered design. Recuperado el 17/11/2020. [Online]. Available: <https://www.interaction-design.org/literature/topics/user-centered-design>
- [8] usability.de GmbH Co. KG. (2013) User interface design. Recuperado el 17/11/2020. [Online]. Available: <https://www.interaction-design.org/literature/topics/ui-design>
- [9] Universitat Oberta de Catalunya. (2015) Diseño de interfaces. Recuperado el 17/11/2020. [Online]. Available: <http://multimedia.uoc.edu/blogs/dii/es/que-es-una-interficie/>
- [10] B. Eskofier, “Notas de la clase: Human computer interaction.” Friedrich - Alexander University, Erlangen, Alemania, 2019.
- [11] Interaction Design Foundation. (2012) The glossary of human computer interaction. Recuperado el 23/11/2020. [Online]. Available: <https://www.interaction-design.org/literature/book/the-glossary-of-human-computer-interaction/artifact#:~:text=An%20artifact%20simply%20means%20any,activities%20in%20a%20design%20process.>
- [12] B. Shackel and S. Richardson, *Human Factors for Informatics Usability*. Cambridge, Gran Bretaña: Cambridge University Press, 1991, pp. 20–23.
- [13] J. Nielsen. (2012) Usability 101: Introduction to usability. Recuperado el 23/11/2020. [Online]. Available: <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>

- [14] A. Dix, J. Finlay, G. D. Abowd, and R. Beale, *Human-Computer Interaction*. Essex, Inglaterra: Pearson Prentice Hall, 2004, pp. 260–273.
- [15] F. S. Co. (2012) FlashCatalogMaker PDF Reader - Freeware. Recuperado el 25/11/2020. [Online]. Available: <http://www.flashcatalogmaker.com/free-ebook-tools/free-flashcatalogmaker-pdf-reader.html>
- [16] Google. (2020) File:google camera icon.svg. Recuperado el 25/11/2020. [Online]. Available: [https://commons.wikimedia.org/wiki/File:Google\\_Camera\\_Icon.svg](https://commons.wikimedia.org/wiki/File:Google_Camera_Icon.svg)
- [17] Niantic Inc. (2016) Pokémon go. Captura de pantalla tomada el 08/01/2021. [Online]. Available: [https://play.google.com/store/apps/details?id=com.nianticlabs.pokemongo&hl=es\\_MX&gl=US](https://play.google.com/store/apps/details?id=com.nianticlabs.pokemongo&hl=es_MX&gl=US)
- [18] 588ku. (2020) File: finger clipart.png. Recuperado el 02/12/2020. [Online]. Available: [https://pngtree.com/freepng/outstretched-index-finger-pointing-to-something-illustration\\_4719934.html](https://pngtree.com/freepng/outstretched-index-finger-pointing-to-something-illustration_4719934.html)
- [19] B. Schneidermann and C. Plaisant, *Designing the User Interface*. Texas, Estados Unidos de América: Pearson Education Inc., 2005, pp. 66–78.
- [20] R. Domingo. (2014) Locus de control. Recuperado el 08/02/2021. [Online]. Available: <https://www.revistainsight.es/locus-de-control/>
- [21] I. S. Mackenzie. (2003) Motor behaviour models for human-computer interaction. Recuperado el 10/01/2021. [Online]. Available: [https://www.yorku.ca/mack/mackenzie\\_chapter.html](https://www.yorku.ca/mack/mackenzie_chapter.html)
- [22] L. Y. B. Blanco, “Notas de la clase: Instrumentación.” Facultad de Ingeniería, UNAM, Ciudad de México, México, 2019.
- [23] M. Semeria. (2015) Los tres teoremas: Fourier - Nyquist - Shannon, Serie de Documentos de Trabajo No. 582. Recuperado el 12/01/2021. [Online]. Available: <https://www.econstor.eu/bitstream/10419/130833/1/844215546.pdf>
- [24] Interaction Design Foundation. (2008) Fitts’ law. Recuperado el 11/01/2021. [Online]. Available: <https://www.interaction-design.org/literature/book/the-glossary-of-human-computer-interaction/fitts-s-law>
- [25] I. Schuetz, T. S. Murdison, K. J. MacKenzie, and M. Zannoli. (2019) An Explanation of Fitts’ Law-like Performance in Gaze-Based Selection Tasks Using a Psychophysics Approach. Recuperado el 13/01/2021. [Online]. Available: <https://research.fb.com/publications/an-explanation-of-fitts-law-like-performance-in-gaze-based-selection-tasks-using-a-psychophysics-approach/#:~:text=Eye%20gaze%20as%20an%20input,nature%20of%20saccadic%20eye%20movements.>
- [26] C. G. Drury, “Application of fitts’ law to foot-pedal design,” *HUMAN FACTORS*, vol. 17, no. 4, pp. 368–373, 1975.
- [27] R. Kerr, “Movement time in an underwater environment,” *Journal of Motor Behavior*, vol. 5, no. 3, pp. 175–178, 1973.
- [28] R. Sleimen-Malkoun, J.-J. Temprado, and E. Berton, “Age-related changes of movement patterns in discrete fitts’ task,” *BMC Neuroscience*, vol. 145, no. 14, 2013.
- [29] S. K. Card, W. K. English, and B. J. Burr, “Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys for text selection on a crt.” *Ergonomics*, vol. 21, no. 8, 1978.

- [30] P. Laubheimer. (2019) Accot-Zhai Steering Law: Implications for UI Design. Recuperado el 12/01/2021. [Online]. Available: [https://www.nngroup.com/articles/steering-law/#:~:text=Definition%3A%20The%20Steering%20Law%20predicts,scroll%20bar%2C%20or%20slider\).&text=On%20YouTube%2C%20the%20playback%20scrubber,that%20obeys%20the%20Steering%20Law](https://www.nngroup.com/articles/steering-law/#:~:text=Definition%3A%20The%20Steering%20Law%20predicts,scroll%20bar%2C%20or%20slider).&text=On%20YouTube%2C%20the%20playback%20scrubber,that%20obeys%20the%20Steering%20Law).
- [31] I. S. MacKenzie. (2018) Fitts' Law. Recuperado el 12/01/2021. [Online]. Available: <https://www.yorku.ca/mack/hhci2018.html>
- [32] M. Torbert. (2015) Fitts Law: The size and distance of a target affects the time required to move to it. Recuperado el 13/01/2021. [Online]. Available: <https://baalteshuvamedia.com/fitts-law/>
- [33] Autodesk. (2018) Maya Keyboard Shortcuts. Recuperado el 13/01/2021. [Online]. Available: <https://help.autodesk.com/cloudhelp/2018/ENU/Maya-KeyboardShortcuts/images/GUID-411DFA9A-FC6F-4485-8083-5E6A6B831EA6.png>
- [34] *Artificial Reality II*. Estados Unidos: Addison-Wesley, 1991.
- [35] J. Carmigniani and B. Furth, *Handbook of Augmented Reality, Chapter 1: Augmented Reality: An Overview*. Nueva York, Estados Unidos: Springer, 2011, pp. 3–4.
- [36] IKEA. (2019) Say hej to IKEA Place. Recuperado el 27/01/2021. [Online]. Available: <https://www.ikea.com/au/en/customer-service/mobile-apps/say-hej-to-ikea-place-pub1f8af050>
- [37] K. Melnick. (2017) Ikea Place ARKit App Brings Virtual Furniture Into Your Home. Recuperado el 27/01/2021. [Online]. Available: <https://vrscout.com/news/ikea-place-arkit-app-brings-virtual-furniture/#>
- [38] Pokémon GO Team. (2019) Las aventuras veraniegas continúan en el Pokémon GO Fest 2019 de Dortmund! Recuperado el 27/01/2021. [Online]. Available: [https://pokemongolive.com/post/gofest-dortmund-2019/?hl=es\\_mx](https://pokemongolive.com/post/gofest-dortmund-2019/?hl=es_mx)
- [39] B. News. (2018) BBC launches augmented reality app for Civilisations. Recuperado el 27/01/2021. [Online]. Available: <https://www.bbc.com/news/technology-42966371>
- [40] A. Carman. (2017) Sephora's latest app update lets you try virtual makeup on at home with AR. Recuperado el 27/01/2021. [Online]. Available: <https://www.theverge.com/2017/3/16/14946086/sephora-virtual-assistant-ios-app-update-ar-makeup>
- [41] Visible Body. (2018) Sephora's latest app update lets you try virtual makeup on at home with AR. Recuperado el 27/01/2021. [Online]. Available: <https://www.visiblebody.com/es/ar>
- [42] C. Otero. (2020) Cómo activar el 'Baby Yoda' 3D en Realidad Aumentada que esconde Google. Recuperado el 27/01/2021. [Online]. Available: [https://as.com/meristation/2020/12/17/betech/1608203224\\_632911.html](https://as.com/meristation/2020/12/17/betech/1608203224_632911.html)
- [43] E. Kucera. (2018) Fig 4- uploaded by Erik Kucera. Recuperado el 25/01/2021. [Online]. Available: [https://www.researchgate.net/figure/Example-of-mixed-reality-Microsoft-HoloLens\\_fig2\\_32454865](https://www.researchgate.net/figure/Example-of-mixed-reality-Microsoft-HoloLens_fig2_32454865)
- [44] J. Peddie, *Augmented Reality: Where we all live*. Suiza: Springer, 2017, pp. 10–27.
- [45] B. Bray and D. Coulter. (2020) What is Mixed Reality? Recuperado el 25/01/2021. [Online]. Available: <https://docs.microsoft.com/en-us/windows/mixed-reality/discover/mixed-reality>

- [46] H. Engels. (2017) Education and VR: 5 examples of bending reality to enhance learning. Recuperado el 25/01/2021. [Online]. Available: <https://visualise.com/2017/11/education-vr-5-examples-bending-reality-enhance-learning>
- [47] J. Peddie, *Augmented Reality Where We All Live*. California, Estados Unidos: Springer, 2017, pp. 20–21.
- [48] Google AR VR, “Introduction to augmented reality and arcore,” Coursera, 2018. [Online]. Available: <https://www.coursera.org/learn/ar>
- [49] A. K. Soni. (2019) The difference between Augmented Reality, Virtual Reality and Mixed Reality. Recuperado el 25/01/2021. [Online]. Available: <https://medium.com/@shivsoni377/the-difference-between-augmented-reality-virtual-reality-and-mixed-reality-a028bdd81f9d>
- [50] M. Mangold. (2019) Pepper’s ghost – the innovation from the 19th century. Recuperado el 27/10/2020. [Online]. Available: <https://magic-holo.com/en/peppers-ghost-the-innovation-from-the-19th-century/>
- [51] F. Oldenburg. (2010) Dreams come true at abbaworld – perform live with abba on stage! Recuperado el 27/10/2020. [Online]. Available: <https://news.cision.com/touring-exhibitions-sweden/r/dreams-come-true-at-abbaworld--perform-live-with-abba-on-stage-,c468583>
- [52] C. Davies. (2017) Star wars: Jedi challenges hands-on: Lightsaber meets ar. Recuperado el 27/10/2020. [Online]. Available: <https://www.slashgear.com/star-wars-jedi-challenges-hands-on-lightsaber-meets-ar-31497173/>
- [53] P. Mealy. (2018) The history of virtual and augmented reality. Dummies: A Willey Brand. [Online]. Available: <https://www.dummies.com/software/the-history-of-virtual-and-augmented-reality/>
- [54] Interaction Design Foundation. (2020) Augmented reality – the past, the present and the future. Recuperado el 26/10/2020. [Online]. Available: <https://www.interaction-design.org/literature/article/augmented-reality-the-past-the-present-and-the-future>
- [55] C. Trilnick. (2015) Sensorama. Recuperado el 26/10/2020. [Online]. Available: <https://proyectoidis.org/sensorama/>
- [56] I. E. Sutherland, “The ultimate display,” *Proceedings of IFIP Congress*, pp. 506–508, 1965.
- [57] Virtual reality before it had that name. Computer History Museum. [Online]. Available: <https://www.youtube.com/watch?v=Y2AIDHjylMI>
- [58] C. Velázquez. (2015) Videoplace. Recuperado el 26/10/2020. [Online]. Available: <https://proyectoidis.org/videoplace/>
- [59] M. Krueger. (2019) Videoplace (1985).
- [60] T. Hollerer and D. Schmalstieg. (2016) Introduction to augmented reality. Recuperado el 26/10/2020. [Online]. Available: <https://www.informit.com/articles/article.aspx?p=2516729&seqNum=2>
- [61] J. Norman. (2014) Louis Rosenberg Develops Virtual Fixtures, the First Fully Immersive Augmented Reality System. Recuperado el 25/01/2021. [Online]. Available: <https://www.historyofinformation.com/detail.php?id=4231>
- [62] ——. File:Virtual-Fixtures-USAF-AR.jpg.

- [63] T. Asghar and A. A. Nauman, *Augmented Reality in Museum Environments: A Case of Study at Lund University Historical Museum*. Sweden: Lund University, 2010, pp. 4–7.
- [64] Z. Yang, S. Shen, Y.-L. Wei, and R. R. Choudhury, “Ear-ar: Indoor acoustic augmented reality on earphones,” *MobiCom*, 2020.
- [65] A. Bobeshko. (2017) 7 GREAT EXAMPLES OF AUGMENTED REALITY IN MUSEUMS. Recuperado el 03/02/2021. [Online]. Available: <https://virtualrealitypop.com/object-recognition-in-augmented-reality-8f7f17127a7a>
- [66] A. Novales, *Filtro de Kalman: teoria y aplicaciones*. Madrid: Universidad Complutense, 2017, pp. 1–2.
- [67] Unity, “Handheld ar app development with unity,” Coursera, 2018. [Online]. Available: <https://www.coursera.org/learn/handheld-ar>
- [68] J. KASPERI, “Occlusion in outdoor augmented reality using geospatial building data,” KTH ROYAL INSTITUTE OF TECHNOLOGY SCHOOL OF COMPUTER SCIENCE AND COMMUNICATION, 2017. [Online]. Available: <http://kth.diva-portal.org/smash/get/diva2:1084547/FULLTEXT01>
- [69] D. Ma, J. Gausemeier, X. Fan, and M. Grafe, *Virtual Reality Augmented Reality in Industry*. Shanghai: Springer, 2011, pp. 4–7.
- [70] M. P. Bailey. (2020) SIBUR ADOPTS BROCHESIA WEARABLE, AR TECHNOLOGIES FOR DIGITAL TRANSFORMATION. Recuperado el 26/01/2021. [Online]. Available: <https://www.chemengonline.com/sibur-adopts-brochesia-wearable-ar-technologies-for-digital-transformation/>
- [71] S. Veenhof. (2019) No, the Augmented Reality wearable is not going to go away. Recuperado el 26/01/2021. [Online]. Available: <https://medium.com/@beyourownrobot/no-the-augmented-reality-wearable-is-not-going-to-go-away-dad92e948300>
- [72] A. Martí. (2020) Glass Enterprise Edition 2: A hands-free device for smarter and faster hands-on work. Recuperado el 26/01/2021. [Online]. Available: [https://www.xataka.com/realidad-virtual-aumentada/google-abre-venta-sus-glass-enterprise-edition-2-manteniendo-subiendo-precio-ahora-no-se-venden-solo-a-empresas#:~:text=As%C3%AD%2C%20a%20diferencia%20de%20las,1.299%20d%C3%B3lares%20\(sin%20rebaja\)](https://www.xataka.com/realidad-virtual-aumentada/google-abre-venta-sus-glass-enterprise-edition-2-manteniendo-subiendo-precio-ahora-no-se-venden-solo-a-empresas#:~:text=As%C3%AD%2C%20a%20diferencia%20de%20las,1.299%20d%C3%B3lares%20(sin%20rebaja)).
- [73] Microsoft. (2021) Hololens 2. Recuperado el 26/01/2021. [Online]. Available: <https://www.microsoft.com/es-es/hololens#OneGDCWeb-Feature-wpfce8d>
- [74] Google AR VR. (2020) Glass Enterprise Edition 2: A hands-free device for smarter and faster hands-on work. Recuperado el 26/01/2021. [Online]. Available: <https://www.youtube.com/watch?v=5IK-zU51MU4&feature=youtu.be>
- [75] C. Allison. (2017) Becoming a Jedi with the Lenovo Mirage. Recuperado el 26/01/2021. [Online]. Available: <https://www.wearable.com/ar/lenovo-mirage-jedi-challenges-review>
- [76] K. Levi. (2018) Creación de experiencias de RA para I/O: Nuestro proceso. Recuperado el 26/01/2021. [Online]. Available: <https://developers-latam.googleblog.com/2018/06/creacion-de-experiencias-de-ra-para-io.html>
- [77] I. de la Horra. (2018) Creación de experiencias de RA para I/O: Nuestro proceso. Recuperado el 26/01/2021. [Online]. Available: <https://www.alfabetizaciondigital.redem.org/arkit/>
- [78] E. Vegas. (2019) Licencia de Vuforia – mayo 2019 ¿Cómo nos afecta? Recuperado el 26/01/2021. [Online]. Available: <https://emiliusvgs.com/licencia-vuforia-mayo-2019-afecta/>

- [79] U. Technologies. (2011) File:Unity Technologies logo.svg. Recuperado el 26/01/2021. [Online]. Available: [https://commons.wikimedia.org/wiki/File:Unity\\_Technologies\\_logo.svg](https://commons.wikimedia.org/wiki/File:Unity_Technologies_logo.svg)
- [80] blippar. (2018) INFOGRAPHIC: HISTORY OF AUGMENTED REALITY. Recuperado el 26/01/2021. [Online]. Available: <https://www.blippar.com/blog/2018/06/08/history-augmented-reality#:~:text=2000%3A%20AR%20Quake%20launched%20%2D%20the,game%20developed%20for%20Nokia%20phones.>
- [81] Dylski. (2008) AR Tennis. Recuperado el 26/01/2021. [Online]. Available: <https://vimeo.com/1766231>
- [82] Zappar. (2012) Augmented Reality for Clothing - The Muppets | Electric Mayhem - for one night only! Recuperado el 26/01/2021. [Online]. Available: <https://www.youtube.com/watch?v=HEXkU94Pcwo>
- [83] C. Collado. (2017) Ingress Prime, la gran actualización del juego precursor de Pokémon GO que llegará en 2018. Recuperado el 26/01/2021. [Online]. Available: <https://andro4all.com/2017/12/ingress-prime-actualizacion-niantic>
- [84] Neludia. (2019) LA MEJOR EXPERIENCIA DE NUESTRA VIDA - POKEMON GO FEST 2019 DORTMUND. Recuperado el 26/01/2021. [Online]. Available: [https://www.youtube.com/watch?v=3Zm\\_DGOmWPs](https://www.youtube.com/watch?v=3Zm_DGOmWPs)
- [85] Space10. (2017) Assembling IKEA's new AR app, without a manual. Recuperado el 26/01/2021. [Online]. Available: <https://medium.com/space10/assembling-ikeas-new-ar-app-without-a-manual-c74c09d0488d>
- [86] L. M. Molera. (2020) ¿Qué es Adobe XD y para qué sirve? Recuperado el 20/01/2021. [Online]. Available: <https://blog.hubspot.es/marketing/para-que-sirve-adobe-xd>
- [87] N. Babich. (2020) UX Design Principles for Augmented Reality. Recuperado el 26/01/2021. [Online]. Available: <https://xd.adobe.com/ideas/principles/emerging-technology/ux-design-principles-for-augmented-reality/>
- [88] M. R. Nichols. (2019) How Augmented Reality Will Disrupt The Manufacturing Industry. Recuperado el 26/01/2021. [Online]. Available: <https://blog.thomasnet.com/augmented-reality-manufacturing>
- [89] videos tutos. (2019) How To Fix Your device isn't compatible with this version in Android on Google Play Store. Recuperado el 26/01/2021. [Online]. Available: <https://www.youtube.com/watch?v=9dOkxF5Egzw>
- [90] R. Agarwal. (2018) Google Motion Stills Update Brings New UI and AR Mode. Recuperado el 26/01/2021. [Online]. Available: <https://beebom.com/motion-stills-update-new-ui-ar/>
- [91] A. Unpingco. (2017) Best practices for mobile AR design. Recuperado el 26/01/2021. [Online]. Available: <https://blog.google/products/google-ar-vr/best-practices-mobile-ar-design/>
- [92] S. Veenhof and M. Skwarek. (2010) Augmented Reality Art Invasion! Recuperado el 26/01/2021. [Online]. Available: <http://www.sndrv.nl/moma/>
- [93] Desconocido. (2016) Pokemon GO - Consejos de seguridad para entrenadores y mitos. Recuperado el 26/01/2021. [Online]. Available: <https://www.mundoplayers.com/pokemon-go-consejos-de-seguridad-para-entrenadores-y-mitos/>
- [94] Ángel Carrero. (2016) Pokémon GO se toma en serio la seguridad de sus usuarios. Recuperado el 26/01/2021. [Online]. Available: [https://messenger.es/pokemon-go-se-toma-serio-la-seguridad-usuarios\\_37353/](https://messenger.es/pokemon-go-se-toma-serio-la-seguridad-usuarios_37353/)
- [95] A. J. Hunsucker. (2017) Usability testing in Augmented Reality. Recuperado el 26/01/2021. [Online]. Available: <https://uxdesign.cc/usability-testing-in-augmented-reality-df8f6c8d0d71>

- [96] K. Ahir. (2020) Occlusion in Augmented Reality. Recuperado el 26/01/2021. [Online]. Available: <https://kumar-ahir.medium.com/occlusion-in-augmented-reality-2090911b3da1>
- [97] M. Ding, "Augmented reality in museums." 2017.
- [98] Monserrat Narváz Naranjo. (2019) History of Audioguides. Recuperado el 28/01/2021. [Online]. Available: <https://www.museummate.com/en/audioguide-history/>
- [99] B. B. Benderson, "Audio augmented reality: A prototype automated tour guide," *CHI 95 Mosaic of Creativity*, 1995.
- [100] J. Flores. (2012) La Nintendo 3DS se convierte en la audioguía del Louvre. Recuperado el 28/01/2021. [Online]. Available: <https://www.muyinteresante.es/innovacion/articulo/la-nintendo-3ds-se-convierte-en-la-audioguia-del-louvre>
- [101] C. Coates. (2020) Museums bring history and culture to life with AR technology. Recuperado el 28/01/2021. [Online]. Available: <https://blooloop.com/features/museum-ar/>
- [102] jlapoutre and @remcovrOOm. (2010) Premier for Lowlands: the Stedelijk Museum art collection in augmented reality. Recuperado el 28/01/2021. [Online]. Available: <https://wttfuture.wordpress.com/2010/08/19/premier-for-lowlands-the-stedelijk-museum-art-collection-in-augmented-reality/>
- [103] Unknown. (2010) MoMA DIY day: Augmented Reality. Recuperado el 30/01/2021. [Online]. Available: <https://www2.stetson.edu/creative-arts/2010/10/moma-diy-day-augmented-reality/>
- [104] S. Veenhof and M. Skwarek. (2010) October 9th 2010 Augmented Reality ART invasion. Recuperado el 30/01/2021. [Online]. Available: <http://www.sndrv.nl/moma/?page=photos>
- [105] L. Dubinsky. (2017) Augmented Reality In Museums: 7 Success Stories. Recuperado el 30/01/2021. [Online]. Available: <https://www.wiktude.com/blog-augmented-reality-museums/>
- [106] D. J. F. Vilchis, R. B. B. Roldan, E. P. Grajales, and L. A. M. Laurencio, "Realidad aumentada como un recurso educativo para el patrimonio cultural," Universidad Autónoma de la Ciudad de México, 2014.
- [107] D. J. F. Vilchis, R. B. B. Roldan, and E. P. Grajales, "Hilvanando cartografías," Universidad Autónoma de la Ciudad de México, 2016.
- [108] J. Nackstrand. (2014) Musion: Meet the holograms. Recuperado el 28/01/2021. [Online]. Available: <https://edition.cnn.com/2014/01/08/tech/gallery/musion-hologram-eyeliner-3d/index.html>
- [109] Spree. (2010) Help Donna fix the house. Recuperado el 28/01/2021. [Online]. Available: <https://www.spree.se/en/portfolio/abba-help-donna-fix-the-house-eng/>
- [110] Deutsche Fussball Museum. (2019) 3D-Kino: Perfekte Illusion dank magischer Projektion. Recuperado el 28/01/2021. [Online]. Available: <https://www.fussballmuseum.de/museum/ausstellung/3d-kino>
- [111] Unknown. (2019) 7 GREAT EXAMPLES OF AUGMENTED REALITY IN MUSEUMS. Recuperado el 30/01/2021. [Online]. Available: <https://old.industry.com/blog/2018/8/21/augmented-reality-museum-examples>
- [112] K. Fawcett. (2014) Please DO Touch the Dinosaurs at the Natural History Museum. Recuperado el 30/01/2021. [Online]. Available: <https://www.smithsonianmag.com/smithsonian-institution/please-do-touch-the-dinosaurs-natural-history-museum-180951911/>

- [113] R. Merril. (2019) Cultura.21 - Udo Lindenberg - Pionero del rock alemán. Recuperado el 30/01/2021. [Online]. Available: <https://www.dw.com/es/cultura21-udo-lindenberg-pionero-del-rock-alem%C3%A1n/av-51195644>
- [114] TripAdvisor. (2018) Panik City. Recuperado el 30/01/2021. [Online]. Available: [https://www.tripadvisor.com.mx/Attraction\\_Review-g187331-d13875674-Reviews-Panik\\_City-Hamburg.html](https://www.tripadvisor.com.mx/Attraction_Review-g187331-d13875674-Reviews-Panik_City-Hamburg.html)
- [115] D. Rodgett. (2018) HOW MUSIC, ART AND VIRTUAL REALITY IS CAPTURING IMAGINATIONS AT PANIK CITY. Recuperado el 30/01/2021. [Online]. Available: <https://digitalsignagesummit.org/europe/how-music-art-and-virtual-reality-is-capturing-imaginations-at-panik-city/>
- [116] Unknown. (2019) PANIK CITY: WIE EIN MULTIMEDIALES MARKENERLEBNIS ENTSTEHT. Recuperado el 30/01/2021. [Online]. Available: <https://pilot-screentime.de/inside-the-panik-city/>
- [117] T. Kletschke. Panik City unter Hamburgs Top 3 der Museen.
- [118] INAH. (2018) El Códice Maya de México (Antes Grolier) Eslabón, Fuente y Testigo. Recuperado el 01/02/2021. [Online]. Available: <https://codicemayademexico.inah.gob.mx/#section-polemic>
- [119] M. Coe, S. Houston, M. Miller, and K. Taube, “El cuarto código maya,” *Maya Archaeology*, 2015.
- [120] INAH. (2018) El Códice Maya de México (Antes Grolier) Eslabón, Fuente y Testigo. Recuperado el 01/02/2021. [Online]. Available: <https://codicemayademexico.inah.gob.mx/interpretacion/folio-detail.html>
- [121] ——. (2018) El Códice Maya de México (Antes Grolier) Eslabón, Fuente y Testigo. Recuperado el 01/02/2021. [Online]. Available: <https://www.inah.gob.mx/boletines/7497-inah-ratifica-al-codice-maya-de-mexico-antes-llamado-grolier-como-el-manuscrito-autentico-mas-antiguo-de-america#:~:text=%E2%80%9CEl%20C%C3%B3dice%20Maya%20de%20M%C3%A9xico,m%C3%A1s%20antiguo%20del%20continente%20americano%E2%80%9D.&text=Todo%20ello%20lo%20convierte%20en%20el%20c%C3%B3dice%20prehispanico%20m%C3%A1s%20antiguo%20conocido.>
- [122] A. A. B. Ruiz and R. S. Hernández. (2020) Desarrollan en la Facultad de Ingeniería cápsula del código maya. Recuperado el 01/02/2021. [Online]. Available: [http://bsqm.org.mx/pdf-boletines/V14/V14N2/BSQM201402\\_Completo.pdf](http://bsqm.org.mx/pdf-boletines/V14/V14N2/BSQM201402_Completo.pdf)
- [123] L. Romero. (2018) Desarrollan en la Facultad de Ingeniería cápsula del código maya. Recuperado el 01/02/2021. [Online]. Available: <https://www.gaceta.unam.mx/desarrollan-en-la-facultad-de-ingenieria-capsula-del-codice-maya/>
- [124] D. de Yucatán. (2018) El Códice Maya más antiguo de América, es expuesto al público . Recuperado el 05/02/2021. [Online]. Available: <https://www.inah.gob.mx/boletines/7497-inah-ratifica-al-codice-maya-de-mexico-antes-llamado-grolier-como-el-manuscrito-autentico-mas-antiguo-de-america>
- [125] M. N. de Antropología. (2018) EL CÓDICE MAYA DE MÉXICO. Recuperado el 05/02/2021. [Online]. Available: [https://mna.inah.gob.mx/exposiciones\\_temporales\\_detalle.php?pl=El\\_Codice\\_Maya\\_de\\_Mexico](https://mna.inah.gob.mx/exposiciones_temporales_detalle.php?pl=El_Codice_Maya_de_Mexico)
- [126] INAH. (2018) INAH ratifica al Códice Maya de México, antes llamado Grolier, como el manuscrito auténtico más antiguo de América. Recuperado el 05/02/2021. [Online]. Available: <https://www.inah.gob.mx/boletines/7497-inah-ratifica-al-codice-maya-de-mexico-antes-llamado-grolier-como-el-manuscrito-autentico-mas-antiguo-de-america>
- [127] M. A. Mercado. (2018) Exposición “El código maya de México”, MNA. Recuperado el 05/02/2021. [Online]. Available: <https://youtu.be/Ab71ZZBMFQc?t=128>

- [128] G. VIJAYAKUMAR. (2021) Portrait Vs Landscape – Which Orientation to Shoot? Recuperado el 13/02/2021. [Online]. Available: <https://www.photographyaxis.com/photography-articles/portrait-vs-landscape-which-mode-to-shoot/>
- [129] M. Baloup, T. Pietrzak, and G. Casiez. (2019) RayCursor: A 3D Pointing Facilitation Technique based on Raycasting. Recuperado el 15/02/2021. [Online]. Available: <https://dl.acm.org/doi/10.1145/3290605.3300331>
- [130] J. Glover. (2017) Learn and Understand Raycasting in Unity3D. Recuperado el 15/02/2021. [Online]. Available: <https://gameacademy.org/learn-and-understand-raycasting-in-unity3d/>
- [131] S. M. del Campo Lanz, *El Códice Maya de México, antes Grolier*. Ciudad de México, México: Instituto Nacional de Antropología e Historia, 2018.
- [132] Unity. (2019) Canvas. Recuperado el 12/03/2021. [Online]. Available: <https://docs.unity3d.com/es/2019.4/Manual/UICanvas.html>
- [133] ——. (2019) Input.deviceOrientation. Recuperado el 12/03/2021. [Online]. Available: <https://docs.unity3d.com/es/2019.4/ScriptReference/Input-deviceOrientation.html>
- [134] ——. (2018) Coroutines (corrutinas). Recuperado el 08/03/2021. [Online]. Available: <https://docs.unity3d.com/es/2018.4/Manual/Coroutines.html>
- [135] ——. (2018) Unity Particle Pack. Recuperado el 10/03/2021. [Online]. Available: <https://assetstore.unity.com/packages/essentials/tutorial-projects/unity-particle-pack-127325>
- [136] ——. (2016) Mesh Collider. Recuperado el 18/03/2021. [Online]. Available: <https://docs.unity3d.com/es/530/Manual/class-MeshCollider.html#:~:text=El%20Mesh%20Collider%20toma%20un,colisionar%20con%20otros%20Mesh%20Colliders.>
- [137] GeeksforGeeks. (2019) C | How to use strings in switch statement . Recuperado el 10/03/2021. [Online]. Available: <https://www.geeksforgeeks.org/c-sharp-how-to-use-strings-in-switch-statement/>
- [138] Unity. (2016) Prefabs. Recuperado el 22/03/2021. [Online]. Available: <https://docs.unity3d.com/es/530/Manual/Prefabs.html>
- [139] yasirkula. (2018) Native Gallery for Android iOS. Recuperado el 22/03/2021. [Online]. Available: <https://assetstore.unity.com/packages/tools/integration/native-gallery-for-android-ios-112630>
- [140] Unity. (2018) RawImage. Recuperado el 24/03/2021. [Online]. Available: <https://docs.unity3d.com/es/2018.4/Manual/script-RawImage.html>
- [141] ——. (2018) Sprites. Recuperado el 24/03/2021. [Online]. Available: <https://docs.unity3d.com/es/2018.4/Manual/Sprites.html>
- [142] yasirkula. (2018) Native Share for Android iOS. Recuperado el 24/03/2021. [Online]. Available: <https://assetstore.unity.com/packages/tools/integration/native-share-for-android-ios-112731>
- [143] Unity. (2018) Types of Light. Recuperado el 25/03/2021. [Online]. Available: <https://docs.unity3d.com/Manual/Lighting.html>
- [144] M. Rohs, A. Oulasvirta, and T. Suomalainen, “Interaction with magic lenses: Real-world validation of a fitts’ law model,” *CHI 2011 Session: Interaction on Mobile Devices*, 2011.

- [145] K. Vargas. (2019) Decimosegunda ley de UX en español: La navaja de Occam. Recuperado el 05/05/2021. [Online]. Available: <https://starteq.net/decimosegunda-ley-de-ux-en-espanol/>
- [146] now that's logistics. (2020) Putting Augmented Reality to Work in the Warehouse. Recuperado el 21/05/2021. [Online]. Available: [https://nowthatslogistics.com/putting-augmented-reality-to-work-in-the-warehouse/?doing\\_wp\\_cron=1621541772.5314869880676269531250](https://nowthatslogistics.com/putting-augmented-reality-to-work-in-the-warehouse/?doing_wp_cron=1621541772.5314869880676269531250)
- [147] usability.gov. (2013) System Usability Scale (SUS). Recuperado el 02/06/2021. [Online]. Available: <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>
- [148] J. Sauro. (2011) Measuring Usability with the System Usability Scale (SUS). Recuperado el 02/06/2021. [Online]. Available: <https://measuringu.com/sus/>
- [149] F. Devin. (2017) Sistema de Escalas de Usabilidad: ¿qué es y para qué sirve? Recuperado el 04/06/2021. [Online]. Available: <https://uxpanol.com/teoria/sistema-de-escalas-de-usabilidad-que-es-y-para-que-sirve/>
- [150] SIGCHI. (2017) CHI. Recuperado el 02/06/2021. [Online]. Available: <https://sigchi.org/conferences/conference-history/chi/>
- [151] Usabilitest. (2011) System Usability Scale (SUS) Plus. Recuperado el 04/06/2021. [Online]. Available: <https://www.usabilitest.com/system-usability-scale>
- [152] J. R. Lewis and J. Sauro. (2018) Item Benchmarks for the System Usability Scale. Recuperado el 04/06/2021. [Online]. Available: <https://uxpajournal.org/item-benchmarks-system-usability-scale-sus/>
- [153] Usabilitest. (2011) SUS PDF Generator. Recuperado el 04/06/2021. [Online]. Available: <https://www.usabilitest.com/sus-pdf-generator>
- [154] A. Smyk. (2020) The System Usability Scale How It's Used in UX. Recuperado el 12/07/2021. [Online]. Available: <https://xd.adobe.com/ideas/process/user-testing/sus-system-usability-scale-ux/>
- [155] M. N. de Antropología. (2018) EL CÓDICE MAYA DE MÉXICO. Recuperado el 21/07/2021. [Online]. Available: [https://mna.inah.gob.mx/exposiciones\\_temporales\\_detalle.php?pl=El\\_Codice\\_Maya\\_de\\_Mexico](https://mna.inah.gob.mx/exposiciones_temporales_detalle.php?pl=El_Codice_Maya_de_Mexico)
- [156] S. Monkey. (2019) Calculadora del tamaño de muestra. Recuperado el 21/07/2021. [Online]. Available: <https://es.surveymonkey.com/mp/sample-size-calculator/>
- [157] C. Busquets. (2019) Principios de UX: Ley de Hick y Ley de Fitts. Recuperado el 11/08/2021. [Online]. Available: <https://www.uifrommars.com/principios-ux-ley-hick-y-fitts/>