

UNIVERSIDAD POLITÉCNICA DE VALENCIA

MASTER THESIS

---

**Posibilidades de la Realidad Aumentada  
en entornos laborables. Estudio  
comparativo entre dispositivos video  
see-through y optical see-through**

---

*Autor:*

David MILLÁN ESCRIVIVÁ

*Director de tesina:*

Dra M. Carmen Juan Lizandra

Dr. Ramón MOLLÁ

*A thesis submitted in fulfilment of the requirements  
for the degree of Master IARFID*

*in the*

Departamento de Sistemas Informáticos y Computación

15 de septiembre de 2014

# Declaration of Authorship

I, David MILLÁN ESCRIVIVÁ, declara que la siguiente tesina titulada, 'Posibilidades de la Realidad Aumentada en entornos laborables. Estudio comparativo entre dispositivos video see-through y optical see-through' y el trabajo presentado son de su propiedad.

Declaro:

- Este trabajo ha sido realizado para el Máster IARFID de la Universidad Politécnica de Valencia.
- Que ninguna parte de esta tesina ha sido previamente usada en otros trabajos o universidades.
- Cualquier consulta o referencia a otros trabajos publicados son claramente mencionados y atribuidos.
- Las referencias o trabajos mencionados son siempre de sus autores. Con excepción de dichas referencias el trabajo es completamente del autor de la tesina.
- Se han reconocido todas las principales fuentes de ayuda.

Signed:

---

Date:

---

*“I started connecting things to my body during my childhood. I approached the computer as a mediating element, as a form of visual art.”*

Steve Mann

UNIVERSIDAD POLITECNICA DE VALENCIA

## *Abstract*

Facultad de informática

Departamento de Sistemas Informáticos y Computación

Master IARFID

**Posibilidades de la Realidad Aumentada en entornos laborables. Estudio comparativo entre dispositivos video see-through y optical see-through**

by David MILLÁN ESCRIVÁ

Después de más de 30 años de desarrollo en la investigación de gafas de realidad aumentada y virtual, en estos últimos años se ha venido produciendo un aumento y mejora sustancial de dichos dispositivos, siendo éstos más livianos, ergonómicos y con mayor durabilidad de sus baterías. De tal forma, estos dispositivos empiezan a ser un gran centro de atención de grandes empresas para darles un uso comercial real.

Todos estos avances sobre los nuevos dispositivos no existentes en el mercado han dejado de lado los estudios de usabilidad y experiencia de usuario en cuanto al desarrollo de software aplicado al uso comercial, y que este trabajo intenta explorar y para ello se presenta un estudio de UI/UX (usabilidad y experiencia de usuario) para un caso de uso concreto y extensible a otros casos, como es el de selección y recogida de objetos o pick-up. Un ejemplo concreto de este caso de uso es por ejemplo unos grandes almacenes de ventas en las que se tiene que coger una serie de productos para ser enviados a un destinatario. Este caso implica múltiples aspectos como localización en interiores, control por voz, visión por computador, gráficos por computador para la gestión de interfaces 2D y 3D, interacción con hardware de dispositivo, etc., y que son tratados en esta tesina.

## *Agradecimientos*

Gracias principalmente a las 20 personas que colaboraron en dicho proyecto para realizar las pruebas y rellenar los formularios necesarios para poder realizar todos los estudios necesarios. También agradecer el esfuerzo titánico en soportar mis inconvenientes tanto de M. Carmen Juan como de Ramón Mollá por la colaboración estrecha en este proyecto. Y como no, a Izakun por todo su apoyo infinito durante todos los años del Máster y a mi pequeña Eider y recién nacido Pau.

# Índice general

<b>Declaration of Authorship</b>	<b>I</b>
<b>Abstract</b>	<b>III</b>
<b>Acknowledgements</b>	<b>IV</b>
<b>List of Figures</b>	<b>VII</b>
<b>List of Tables</b>	<b>VIII</b>
<b>Abreviaturas</b>	<b>IX</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	1
1.2. Hipótesis y objetivos . . . . .	3
1.3. Estructura de la tesina . . . . .	4
<b>2. Estado del arte y análisis del proyecto</b>	<b>6</b>
2.1. Introducción . . . . .	6
2.2. Dispositivos de visión, presente y futuro . . . . .	8
2.3. Software de desarrollo . . . . .	12
<b>3. Desarrollo</b>	<b>16</b>
3.1. Introducción . . . . .	16
3.2. Desarrollo del framework . . . . .	17
3.2.1. Módulos Core y Eventos . . . . .	19
3.2.2. Módulo interfaz de Usuario y componentes . . . . .	23
3.2.3. Módulo Dispositivos . . . . .	24
3.2.4. Módulo AR . . . . .	25
3.2.5. Módulo Utilidades . . . . .	26
3.3. Diseño de la prueba para el estudio de UX . . . . .	26
3.4. Desarrollo del PoC . . . . .	29
<b>4. Validación</b>	<b>33</b>
4.1. Participantes . . . . .	33
4.2. Medidas . . . . .	33
4.3. Procedimiento . . . . .	33

---

4.4. Resultados . . . . .	34
<b>5. Conclusión y trabajos futuros</b>	<b>39</b>
5.1. Conclusión . . . . .	39
5.2. Trabajos Futuros . . . . .	40
5.3. Repercusión profesional . . . . .	40
<b>A. Apéndice</b>	<b>42</b>
A.1. Cuestionario 1 . . . . .	42
A.2. Cuestionario 2 . . . . .	43
<b>Bibliografía</b>	<b>44</b>

# Índice de figuras

1.1. Evolución de los dispositivos de realidad aumentada en la vida de Steve Mann, reconocido como el padre de la realidad aumentada y de las gafas de realidad aumentada. . . . .	2
2.1. Evolución de los dispositivos de realidad aumentada desde 1980 a 1990. . . . .	8
2.2. Google Glass, Vuzix M100, Meta, Epson Moverio . . . . .	9
2.3. Vuzix Star 1200 XL. . . . .	9
2.4. Vuzix Wrap 920 AR. . . . .	9
3.1. Diagrama de clases. . . . .	20
3.2. Diagrama de clases. . . . .	21
3.3. Flujo de eventos. . . . .	22
3.4. Detección de códigos QR. . . . .	25
3.5. Reconocimiento y detección de marcadores. . . . .	26
3.6. Esquema de dispositivos. . . . .	27
3.7. Información del producto a recoger. . . . .	28
3.8. Información del producto a recoger en un lateral. . . . .	28
3.9. Información de la localización del producto a seleccionar. . . . .	28
3.10. Diagrama sistema PoC . . . . .	29
3.11. Página principal de bienvenida y gestión de stocks. . . . .	30
3.12. Página de descripción de producto . . . . .	31
3.13. PoC en funcionamiento . . . . .	31
4.1. BoxPlot T1. . . . .	35
4.2. BoxPlot T2. . . . .	35
5.1. Ejemplo aplicación QR . . . . .	40

# Índice de tablas

2.1. Comparativa de dispositivos de visión. VSTm Video See Trough Monocular, OSTm Optical See Trough Monocular . . . . .	10
2.2. Comparativa de dispositivos de visión. VSTm Video See Trough Monocular, OSTm Optical See Trough Monocular . . . . .	11
4.1. one-way Anova. T1 tiempo primer contacto con un dispositivo y T2 tiempo segundo contacto con el otro dispositivo . . . . .	36
4.2. T-Test. Preferencia de sistemas . . . . .	37
4.3. T-Pair Test. Preferencia de sistemas . . . . .	37
4.4. ChiSquare. Preferencia de sistema . . . . .	38
A.1. Tabla A1. Cuestionario 1 . . . . .	42
A.2. Tabla A2. Cuestionario 2 . . . . .	43
A.3. Tabla A3. Cuestionario 2 comparativa . . . . .	43

# Abreviaturas

<b>UI/UX</b>	<b>U</b> ser <b>I</b> nterfaces / <b>U</b> ser <b>eX</b> perience
<b>AR</b>	<b>A</b> ugmented <b>R</b> eality
<b>RA</b>	<b>R</b> ealidad <b>A</b> umentada
<b>MIT</b>	<b>M</b> assachusetts <b>I</b> nstitute of <b>T</b> echnology
<b>OST</b>	<b>O</b> ptical <b>S</b> ee <b>T</b> hrough
<b>VST</b>	<b>V</b> ideo <b>S</b> ee <b>T</b> hrough
<b>IPS</b>	<b>I</b> ndoor <b>P</b> osition <b>S</b> ystem
<b>GPS</b>	<b>G</b> lobal <b>P</b> osition <b>S</b> ystem
<b>HMD</b>	<b>H</b> ead <b>M</b> ounted <b>D</b> isplay

# Capítulo 1

## Introducción

### 1.1. Motivación

Después de más de 30 años de desarrollo en la investigación de gafas de realidad aumentada y virtual, en estos últimos años se ha venido produciendo un aumento y mejora sustancial de dichos dispositivos, siendo éstos más livianos, ergonómicos y con mayor durabilidad de sus baterías. De tal forma, estos dispositivos empiezan a ser un gran centro de atención de grandes empresas para darles un uso comercial real.

Desde sus inicios en el MIT de este tipo de dispositivos sobre los años 65 con Ivan Sutherland y posteriormente con Steve Mann en los años 80 (fig. 1.1) hasta los actuales dispositivos todavía no comercializados en su totalidad como las Google Glass o el M100 de Vuzix, la mayoría de recursos y estudios se han centrado en mejorar el hardware y buscando casos de uso de dichos dispositivos de forma académica. Aunque no son explotados claramente de forma comercial ni se han realizado extensos estudios de UI/UX sobre ellos.

Así pues este trabajo pretende realizar un estudio principalmente de experiencia de usuarios (UX en sus siglas en inglés), es decir, el conjunto de factores y elementos relativos a la interacción del usuario con un entorno y/o dispositivo concreto. En este caso de dos tipos de gafas de realidad aumentada, junto a un software asociado, generando una respuesta de percepción positiva o negativa del conjunto de objeto de estudio, como aspectos de utilidad, facilidad, eficiencia, etc.

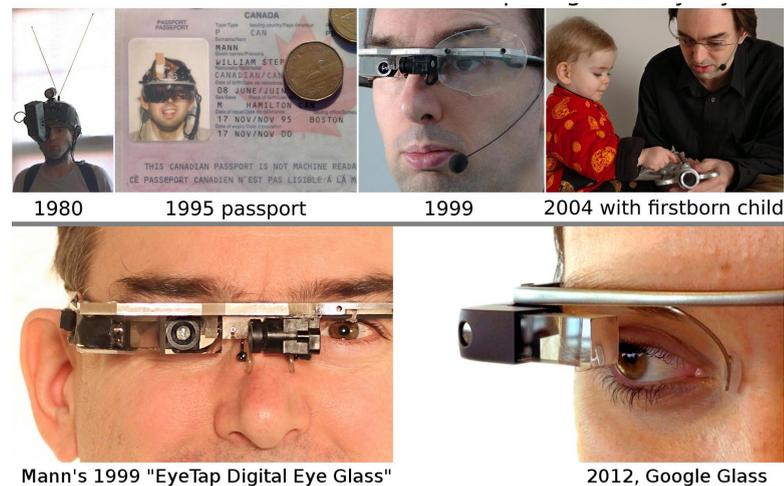


FIGURA 1.1: Evolución de los dispositivos de realidad aumentada en la vida de Steve Mann, reconocido como el padre de la realidad aumentada y de las gafas de realidad aumentada.

Este estudio se realiza sobre un caso de uso concreto, aplicable comercialmente como se verá en la conclusión y extensible a cualquier otro caso, con el uso de gafas de realidad aumentada. El caso de uso seleccionado se basa en las empresas de almacenaje o también comúnmente llamadas de Pickup y se basa en un gran almacén en la que los empleados deben recoger una serie de productos de un pedido para un cliente particular y ser enviados, como ejemplo de este tipo de industria puede ser mensajerías como MRW o venta de productos como Amazon. Este tipo de empresas tienen una serie de requisitos y limitaciones en las que dispositivos de visión de realidad aumentada pueden ayudar a mejorar la eficacia y eficiencia de los trabajadores, ayudando a éstos en diferentes aspectos como se explica con más detalle en el capítulo 3.

Además del estudio principal de UX, se van a analizar en profundidad diferentes aspectos necesarios y relacionados para llevar a cabo el correcto desarrollo del proyecto como son, un análisis de las características de las gafas de realidad aumentada del mercado, qué dispositivos de visión son recomendados para este caso de uso en concreto, una comparativa de los dos tipos de dispositivos de visión de RA dependiendo de si usan tecnología de visión a través de vídeo o de visión a través de lente, mas comúnmente mencionados en inglés como Video See Through (VST) u Optical See Through (OST). También se va a explorar el software y librerías existentes de desarrollo de realidad aumentada y cuáles implementan accesos o facilidades al hardware y especificaciones de gafas de realidad aumentada.

Así pues este estudio pretende desarrollar un prototipo completo para el estudio de la experiencia de usuario y posteriormente aplicarlo a la industria de almacenaje o Pickup mediante el desarrollando de un producto completo explicado con más detalle en el capítulo 3. A su vez se ha aplicado este proyecto directamente en el mercado como prototipo de una startup norteamericana y usada para la búsqueda de inversores. Gracias a este estudio, desarrollo del prototipo y diferentes vídeos de dicho prototipo varios inversores han invertido 2M de dólares en 2013/14 financiando dicha empresa para el desarrollo comercial de este tipo de aplicaciones.

## 1.2. Hipótesis y objetivos

Este estudio de experiencia de usuario empleando los dos diferentes dispositivos de visión de realidad aumentada VST y OST aplicado al caso de uso de Pickup pretende demostrar que el uso de dispositivos Optical See through en contra de los Video See through son una selección preferente para el desarrollo de trabajos con interacción y manipulación directa con el mundo real, dejando los dispositivos VST para otro tipo de tareas. También deja entrever la falta de un framework de desarrollo completo para dispositivos de visión que faciliten el acceso e integren todas las herramientas necesarias para el desarrollo rápido, fácil e intuitivo de aplicaciones de realidad aumentada con el uso de este tipo de dispositivos.

Los principales objetivos de esta tesina son:

- Estudio de los dispositivos de visión actuales y futuros
- Análisis de los frameworks y librerías del mercado para realidad aumentada
- Desarrollo de una librería basada en eventos usando las librerías de OpenCV y OpenGL para el desarrollo ágil de aplicaciones de realidad aumentada y para dispositivos de visión como por ejemplo la Marca Vuzix
- Estudio del caso de uso de pickup, requisitos, necesidades y estado de las tecnologías involucradas
- Estudio de UI/UX

- El diseño de diferentes cuestionarios para la recogida de datos para su posterior análisis
- Medir la satisfacción y experiencia de los usuarios a partir de la encuesta realizada
- Extraer conclusiones del estudio estadístico y analítico de los diferentes cuestionarios y la población estudiada
- Análisis de la repercusión en el mercado del prototipo y desarrollos elaborados

### **1.3. Estructura de la tesina**

La tesina se ha estructurado de la siguiente forma, intentando mantener una lógica secuencial en el desarrollo del trabajo para una fácil comprensión:

- El capítulo 1 introduce el estudio, la motivación y objetivos de esta tesina, así como su estructura, dejando entrever las peculiaridades y dificultades que entraña dicho proyecto.
- El capítulo 2 muestra el estado del arte, mostrando la literatura más relevante relativa a la experiencia de usuario en entornos de realidad aumentada con el uso de dispositivos de visión, a su vez realizaremos una revisión y comparativa de los diferentes dispositivos del mercado y software de desarrollo, mostrando todas las herramientas que disponemos para poder desarrollar nuestro proyecto.
- El capítulo 3 describe el desarrollo realizado para llevar a cabo el proyecto, tanto en cuanto a la arquitectura desarrollada, las dependencias directas, y cómo desarrollar nuevo software a partir de nuestro nuevo framework. También describiremos el proceso realizado para el prototipo y captación de datos de los usuarios que se utilizarán en el capítulo 4.
- El capítulo 4 muestra el estudio principal de la tesina, donde se evalúa el prototipo con un grupo numeroso de personas para realizar la comparativa entre los dos diferentes tipos de dispositivos de visión usados, VST y OST.
- El capítulo 5 muestra las conclusiones, repercusiones del trabajo y prototipo realizado y el trabajo futuro a realizar.

- Además se añade el Apéndice A que muestra los diferentes cuestionarios usados en este estudio.

## Capítulo 2

# Estado del arte y análisis del proyecto

### 2.1. Introducción

En la actualidad y después de más de 30 años de desarrollo e investigación en los dispositivos de visión para aplicaciones de realidad aumentada, estos dispositivos empiezan a ser objetivo del mercado tanto industrial como del consumidor [35]. Éstos dejan de ser dispositivos pesados y conectados directamente a un ordenador, a ser dispositivos inalámbricos, livianos y con la suficiente autonomía como para ser usados en entornos laborales.

Existen diferentes estudios relacionados con el uso de dispositivos de visión desde hace muchos años en diferentes campos como por ejemplo el médico [30] en la que se compara el uso de los dos dispositivos de visión OST y VST para la superposición de imágenes de rayos X. Con un estudio completo tanto desde el desarrollo pasando por la sensación del usuario al usar los dos tipos de dispositivos y concluyendo que los dos dispositivos en dicho momento mostraban importantes problemas en la latencia, oclusión y fidelidad entre la superposición de la realidad aumentada junto al mundo real. Además apuntan que el uso de OST ofrece esencialmente una percepción del mundo real sin latencia y sincronizada con el usuario, cosa totalmente contraria con el VST que el mundo real es capturado por las cámaras con un tiempo de latencia que el usuario percibe, pero permitiendo realizar otras como la posibilidad de renderizar oclusiones en objetos reales.

Más recientemente [29] para la visualización de objetos tridimensionales superpuestos a objetos reales, trabajo bastante similar que el anterior y que concluyen de la misma forma, necesitando mejoras tanto en los dispositivos de visión, como en el software, así notando la necesidad de la oclusión de objetos reales para el campo médico. De los que deriva este estudio en que se hace un análisis de las aplicaciones de realidad aumentada y sus dispositivos en el caso de uso de diseño e industria [7], analizando todos los posibles casos de uso como trabajos colaborativos, los factores humanos y su interacción, dificultades técnicas y presentando algunas aplicaciones de realidad aumentada para la industria. En dicho estudio se concluye con la importancia de diseñar y proveer interfaces de usuario intuitivas y eficientes a su vez de proveer de las herramientas necesarias para poder llevar a cabo dichos desarrollos. Punto de partida de este proyecto, en el que se va a estudiar desde el desarrollo de dichas herramientas hasta la experiencia de un grupo de usuarios sobre un caso concreto.

Así pues el trabajo que presentamos se centra en el estudio del uso de dispositivos de visión OST y VST para el caso de uso de recogida de productos, haciendo un estudio de qué dispositivo se ajusta mejor a este tipo de mercado mediante la visualización de imágenes 3D superpuestas, imágenes y textos bidimensionales de información, y mapas de indicación. Creando un producto y framework de desarrollo que se ajuste a la mayoría de necesidades de la industria de realidad aumentada para los dispositivos de visión.

La industria del almacenaje, nuestro caso de uso, utiliza múltiples tecnologías que actualmente se están estudiando y mejorando continuamente como es por ejemplo el posicionamiento en interiores o en sus siglas en inglés IPS (Indoor Position Systems). El uso de GPS en interiores como almacenes o grandes superficies comerciales no recibe la señal de los satélites. Por ello un sistema IPS es necesario para indicar al usuario la posición de los productos dentro del almacén con una cierta exactitud. Estos sistemas trabajan mediante la triangularización de señales inalámbricas procedente de redes Wifi [3] u otro tipo de redes o inclusive mediante luz modulada mediante leds [15].

También haciendo uso de algoritmos como SLAM para la reconstrucción tridimensional del entorno [14] [43] [33] [19] que rodea al usuario para poder así realizar la superposición de imágenes bidimensionales o tridimensionales en el entorno real del usuario.

De tal forma que el desarrollo del framework se convierte en un concepto modular y complejo que une diferentes librerías y piezas que permitan desarrollar cualquier tipo de

producto, no solo nuestro caso de uso, sino cualquier que este relacionado con la realidad aumentada y el uso de los dispositivos de visión. Aunque estos últimos no son necesarios para desarrollar una aplicación con el framework desarrollado bajo esta tesina.

## 2.2. Dispositivos de visión, presente y futuro

Los dispositivos de realidad aumentada, en concreto los dispositivos de visión o HMD (Head-mounted display), han evolucionado mucho desde su aparición. Éstos eran pesados y voluminosos, a su vez que dependientes de un ordenador al que estaban conectados (Fig. 2.1). Actualmente han pasado a ser dispositivos ligeros, ergonómicos, eficientes, aunque dependientes la mayoría de una computadora como son los dispositivos que existen actualmente en el mercado comercial a excepción de las Google Glass [13] y Vuzix M100 [41].



FIGURA 2.1: Evolución de los dispositivos de realidad aumentada desde 1980 a 1990.

En la actualidad existen diferentes tipos de HMD, tanto OST como VST, y existe un mercado creciente en la innovación de dichos dispositivos intentando desvincularlos de un ordenador mediante el uso de los nuevos procesadores de menor consumo y tamaño y mayor potencia, empleados normalmente en móviles.

Actualmente se están desarrollando nuevos dispositivos en un afán de hacer realidad el uso cotidiano de los dispositivos de visión o HMD. Empresas como Google con sus Google Glass [13], o Vuzix [41] con su nuevo dispositivo M100, dispositivos monoculares con un gran mercado potencial, o empresas como Epson con su dispositivo Moverio [12], o el nuevo dispositivo Meta [23] asesorado por el profesor Steve Mann (fig. 2.2) están revolucionando el mercado de los dispositivos HMD.



FIGURA 2.2: Google Glass, Vuzix M100, Meta, Epson Moverio

Todavía no todos han salido a la venta ya que algunos no lo harán hasta 2014/15 y que solo unos pocos privilegiados tienen acceso a ellos. Todo este incremento en tecnología pronostica un mercado en auge con máximos de mercado entre el 2017 y 2018 tanto en hardware como en software.

Una vez introducido el estado actual vamos a realizar mediante la tabla 2.1 una comparativa de los dispositivos actuales y futuros.

Una de las empresas pioneras y con más andadura en el mercado de los HMD es Vuzix. Vuzix cuenta con diferentes tipos de dispositivos de visión tanto OST como VST como son las Vuzix Star 1200 XL (fig. 2.3) o las Vuzix Wrap 920 AR (fig. 2.4) respectivamente y con las cuales hemos desarrollado el proyecto.



FIGURA 2.3: Vuzix Star 1200 XL.



FIGURA 2.4: Vuzix Wrap 920 AR.

La selección de estos dos dispositivos para el desarrollo de la tesina se debe principalmente a la necesidad de tener dos dispositivos de similares o exactas especificaciones

Dispositivo	OST/VST	FOV	Resolución	Conectividad	Peso	Tecnología	Procesador
<b>Meta Pro</b> 3650\$ pre-orden Agosto 2015	OST	40°	1280x720	Wifi 802.11b/g/n Bluetooth 4.0 Usb 3.0	180g	sensor de profundidad 3D dos cámaras RGB 9-ejes (acelerómetro, giroscopio, y compás)	Computador de bolsillo: Intel i5 CPU 4Gb ram 128Gb SSD 32Whr batería
<b>Moverio™ BT-200 Smart Glasses</b> 699.9\$	OST	23°	960x540	Wifi 802.11b/g/n Bluetooth 3.0 Usb 2.0	88g	RGB Cámara GPS Compas Giroscopio Acelerómetro	Computador de bolsillo: CPU: TI OMAP 4460 1.2Ghz Dual Core RAM: 1 GB Internal Memory: 8 GB External Memory: microSD (max.2GB) / microSDHC(max.32GB)
<b>Vuzix Star 1200XLD</b> 4999\$	OST	35°	852x480	No	-	3 grados de libertad de track de las gafas (yaw, pitch y roll) giroscopio acelerómetro Cámara 1080p con autofocus y frame rates 1Hz a 60Hz	Sin computador asociado, necesita de un pc conectado o dispositivo con salida HDMI
<b>Vuzix Warp 1200DXAR</b> 1499\$	VST	35°	852x480	No	-	3 grados de libertad de track de las gafas (yaw, pitch y roll) acelerómetros giróscopo Cámara estero (2 cámaras) VGA 640x480 a 30fps ajustables en ángulo	Sin computador asociado, necesita de un pc conectado o dispositivo con salida HDMI

CUADRO 2.1: Comparativa de dispositivos de visión. VSTm Video See Trough Monocular, OSTm Optical See Trough Monocular

(FOV, resolución, tecnología...) pero diferente tipo de visión (OST o VST) además de la calidad y el extenso y conocido uso de la marca Vuzix.

Estos dos dispositivos de características excelentes tienen una gran carencia que es la

Dispositivo	OST/VST	FOV	Resolución	Conectividad	Peso	Tecnología	Procesador
<b>Vuzix M100</b> 999.99\$	VSTm	15°	WQVGA	WiFi 802.11b/g/n Bluetooth Micro USB	-	3 grados de libertad de track de gafas acelerómetro giroscopo mag GPS Proximidad Ambient Light 4 botones de control Cámara 5Mp 1080p	Computador integrado en las gafas: OMAP4460 at 1.2GHz 1 GB RAM Android ICS 4.04, API 15 4GB flash
<b>Vuzix M2000AR</b> 5999\$	OSTm	30°	1280x720	No	-	3 grados de libertad de seguimiento de las gafas (yaw, pitch y roll) giroscopo acelerómetro Cámara 5Mp 83fps VGA	Sin computador asociado, necesita de un pc conectado o dispositivo con salida HDMI
<b>Google Glass</b> 1500\$ Pre-order only USA	OSTm	14°	640x360	Wi-Fi 802.11b/g, Bluetooth micro USB Cámara 5Mp TouchPad	50g	Comandos por voz integrado acelerómetro giroscopio magnetómetro Detección de luz ambiental Sensor de proximidad	Computador integrado en las gafas: OMAP 4430 SoC dual core 1GB Ram 16GB Flash storage Android 4.4.2

CUADRO 2.2: Comparativa de dispositivos de visión. VSTm Video See Trough Monocular, OSTm Optical See Trough Monocular

dependencia de una conexión directa a un dispositivo externo para enviar las imágenes que se deben superponer o enviar a la pantalla del dispositivo. Además de recoger las señales de las diferentes cámaras que incorporan y micrófono para ser procesadas y enviar una respuesta correcta al usuario.

Otra buena selección hubiera sido Google Glass y Vuzix M100, ya que son una opción muy interesante para nuestro caso de uso. Pero en el momento de desarrollo de la tesina no estaban en el mercado las Google Glass, y existían pocas unidades de las Vuzix M100.

Además de las diferencias tanto de software como de hardware entre dichos dispositivos, haciendo la comparativa de experiencia de usuario también dependiente de marca, ergonomía y otros factores que no solo el de nuestro interés sobre Optical See Through y Video See Through.

### 2.3. Software de desarrollo

Otro aspecto interesante a estudiar para el correcto desarrollo de nuestro proyecto es el software existente en el mercado para el desarrollo de una aplicación para nuestro caso de uso. Actualmente existen multitud de librerías para el desarrollo de aplicaciones de realidad aumentada. Pero cada una de ellas implementa una pequeña parte o pieza de lo que realmente se necesita en un proyecto completo de realidad aumentada y que a continuación se detallan la mayoría de requisitos que debería tener una librería de realidad aumentada y de nuestro framework:

- librerías gráficas 2D y 3D, que permitirán la visualización en tiempo real y superposición de los elementos virtuales en nuestro campo de visión.
- reconocimiento de objetos, que permitirá asociar a un contexto o reconocer imágenes [44] o formas [26] capturadas por la cámara, y poder realizar un seguimiento [34] de dichos objetos o superponer información sobre ellos [26].
- reconocimiento de voz, extremadamente útil en casos en que el usuario no puede interactuar directamente sobre los controles o botones del dispositivo por tener las manos ocupadas o con dispositivos externos y necesita realizar o mandar una acción a ejecutar en el dispositivo [22].
- reconocimiento de gestos, que permite realizar acciones sobre el dispositivo mediante gestos. Este tipo de interacción es usado en entornos donde el reconocimiento de voz no es factible debido al ruido [22].
- reconstrucción de entornos en 3D, mediante algoritmos como SLAM que permite reconstruir nuestro entorno en 3D mediante una nube de puntos y poder tener una mayor comprensión del entorno e interacción con él.

- superposición de elementos virtuales 3D, que mediante ayuda de las librerías gráficas y diferentes técnicas algorítmicas para el cálculo correcto del espacio 3D son capaces de simular que un objeto virtual esté en nuestro espacio real con sistemas de marcas [26] o sin el [18].
- acceso a controles hardware de vuzix u otras marcas de dispositivos de visión
- acceso o gestión de dispositivos hápticos, que permitan de una forma fácil el uso de dispositivos externos como guantes, etc.

Una vez definidos los requisitos necesarios para nuestro software o framework de desarrollo vamos a analizar el software actual del mercado tanto de Realidad Aumentada, como aquellas librerías que nos sean útiles para el desarrollo de nuestro framework.

Librerías o software específicamente de realidad aumentada:

- **ARToolKit** [6], librería open source de realidad aumentada basada en marcadores
- **ArUco** [11], librería open source de realidad aumentada basada en marcadores y múltiples marcadores con dependencia de OpenCV
- **DART** [27], toolkit para el soporte rápido al diseño y desarrollo de aplicaciones de realidad aumentada.
- **ATOMIC Authoring Tool** [36] es una herramienta que permite la creación de aplicaciones de realidad aumentada, desarrollada especialmente para no-programadores.
- **GRATF** [2], librería open source en C# para la realidad aumentada basada en marcadores
- **droidar** [9], librería open source en Java para dispositivos android.
- **arpa sdk** [5], librería propietaria para el desarrollo de aplicaciones de realidad aumentada para escritorio, web y dispositivos móviles
- **vuforia sdk** [32], librería propietaria libre para el desarrollo de aplicaciones de realidad aumentada y algunas tareas de visión optimizado para dispositivos móviles ios y android

- **metaio sdk** [25], solución completa de realidad aumentada, con soporte multi-plataforma y soporte a diferentes dispositivos visuales como epon moverio, Vuzix M100 o Google Glass, con acceso a múltiples tecnologías como SLAM, face tracking, escaneado de barras y QR, ...
- **wikitude** [42], librería de desarrollo para dispositivos móviles, incluyendo reconocimiento de imágenes, posicionamiento GPS y modelado 3D.
- **junaio** [24], es un navegador diseñado para la realidad aumentada para los dispositivos móviles.
- **layar** [20], es una aplicación y solución completa para el marketing y publicidad, que permite la integración de realidad aumentada en publicidad y libros impresos.
- **arlab** [4], es un conjunto de librerías con diferentes soluciones para navegador, reconocimiento de imágenes, tracking de objetos e imágenes, inclusión de 3D...

Debido a los requisitos de nuestra aplicación, el software o librería que más se ajusta es metaio sdk pro, pero debido al excesivo coste de dicha librería se propone el desarrollo de un toolkit o framework que cumpla las necesidades del proyecto y sea flexible y extensible, para ello se estudian y usan las siguientes librerías de las que se hacen uso:

- **CMU Sphinx** [39], librería open source para el reconocimiento del habla multi-idioma en C/C++.
- **OpenGL** [16], librería estándar de gráficos por computador principalmente 3D y con soporte 2D en diferentes lenguajes de programación.
- **OpenCV** [28], librería de visión por computador general optimizada en C/C++ y portable a dispositivos móviles.
- **aruco** [11], librería compacta de realidad aumentada basada en marcadores y OpenCV, esta librería extrae de un marcador o múltiples los parámetros para poder realizar integración 3D.
- **vuzix sdk** [41], librería en C/C++ para la integración de los dispositivos de visión de la compañía Vuzix.

- **assimp** [1], librería en C/C++ open source para la importación y carga de objetos 3D, animaciones, texturas, etc. en diferentes formatos.
- **gsoap** [40], librería en C/C++ open source para servicios web REST y SOAP y aplicaciones basadas en XML
- **libxml2** [21], librería open source en C/C++ de interpretación, lectura y escritura de formatos XML
- **libdecodeqr** [37], librería open source en C/C++ de decodificación de códigos QR, librería basada en OpenCV
- **pthread** [8], librería open source en C/C++ para arquitecturas multiprocesador con memoria compartida para el desarrollo de aplicaciones con procesos paralelos.
- **curl** [10], aplicación de línea de comandos y librería open source para la transferencia de datos mediante sintaxis URL soportando protocolos como HTTP, HTTPS, FILE, FTP, FTPS, RTMP, SMTP, ...

## Capítulo 3

# Desarrollo

### 3.1. Introducción

El objetivo de la tesina es el desarrollo de un prototipo o aplicación para la evaluación de la experiencia de usuario en un entorno industrial como es el de Pickup. De tal forma se ha planteado desarrollar dos aplicaciones, una para el experimento de UX con usuarios no experimentados en los tipos de dispositivos que vamos a usar, y otra como prototipo completo del caso de uso para probar en el entorno real laboral y también como PoC (Prueba de concepto) para la captación de inversores para la empresa norteamericana Augmate.

Así pues para llevar a cabo ambos desarrollos y poder a su vez abordar nuevos desarrollos, a partir del análisis y estado del arte estudiado en el capítulo 2 se ha creado una librería especializada para dar soporte a los dispositivos de visión o HMD seleccionados para el proyecto.

Este desarrollo permitirá a su vez extenderlo a otros dispositivos futuros y abstraer el programador de las complejidades, especificaciones y librerías de cada dispositivo, si es un dispositivo monocular, binocular, estereoscópico, con 1 cámara ó 2 y pudiendo centrarse en el desarrollo de la aplicación y su lógica de negocio.

A su vez, la librería proporciona una serie de utilidades y herramientas para facilitar el desarrollo de aplicaciones de realidad aumentada, como puede ser la importación de objetos tridimensionales, el reconocimiento y detección de marcadores, seguimiento y

posición de cámara respecto a los marcadores, detección y decodificación de códigos QR, utilidades de calibración, gestión de eventos, interfaces de usuario (botones, ventanas, menús, ...), etc.

De tal forma este capítulo está dividido en las siguientes secciones para analizar con detalle cada uno de los desarrollos mencionados:

- Desarrollo del framework
- Desarrollo del prototipo para el estudio de UX
- Desarrollo de la prueba de concepto PoC

### **3.2. Desarrollo del framework**

El objetivo del desarrollo del framework es proveer un nuevo conjunto de herramientas que faciliten el desarrollo rápido de aplicaciones de realidad aumentada, sea multiplataforma y soporte múltiples dispositivos, desde dispositivos de visión hasta dispositivos hápticos. Es por ello que el desarrollo del framework se ha realizado en C/C++, compatible con todos los sistemas operativos y con facilidad de usar en dispositivos android e iOS, pudiendo compilar el proyecto tanto en Windows con VC++ y otros entornos, en Linux o en OSX.

Para poder gestionar de una forma fácil y efectiva todas las configuraciones diferentes de cada sistema operativo se ha hecho uso de CMake [17]. CMake es un sistema que permite mediante una serie de scripts generar los procesos nativos de compilación para cada sistema operativo generando de tal forma proyectos para VC++, Eclipse, XCode, makefiles, etc.

Debido a la complejidad del desarrollo el framework hace uso de múltiples librerías que facilitan el desarrollo e integración de diferentes tecnologías en un mismo marco. Pero la decisión de qué librerías van a ser usadas es crítica y dependiente de los requisitos iniciales de multi-plataforma y soporte a múltiples dispositivos. Además, otro requisito añadido es el de no exceso de dependencias de librerías para no aumentar la complejidad de los script y compilación de dependencias y el tamaño final de la aplicación generada, ya que una aplicación de gran tamaño no puede ser instalada en dispositivos con poca

memoria. Esto significa que si se hace uso de una librería de reconocimiento facial y otra de SLAM que dependen de una librería de visión por computador, ésta ha de ser la misma librería para los dos.

Previa definición de la librerías y dependencias de nuestro framework debemos definir claramente las especificaciones para el desarrollo de nuestro framework, ya mencionados en el capítulo 2. Especificaciones que den soporte a los desarrollos que vamos a realizar en los capítulos 3.3 y 3.4, así como posibles futuros proyectos. Estas especificaciones son parte de las especificaciones definidas en la sección 2.3.

- Soporte a gráficos 2D y 3D para mostrar la información al usuario
- Componentes básicos de interacción, botones, alertas, menús, multimedia ...
- Importación de modelos 3D, animaciones y texturas, para mostrar al usuario información del producto seleccionado
- Lectura de códigos QR para el escaneo de productos
- Localización en interiores IPS, para poder situar en un plano la localización del usuario y del producto a recoger.
- Acceso al hardware de los dispositivos como cámara, acelerómetros, giroscópios, ...  
Se ha integrado la marca Vuzix con soporte de los dispositivos 920AR y 1200XL. Pero se ha dejado abierta la posibilidad de añadir más de forma simple.
- Reconocimiento de voz, para que el usuario pueda interactuar con el sistema y al mismo tiempo poder portar objetos entre sus manos o realizar otras tareas
- Superposición e integración de objetos virtuales 3D en entornos reales

Para dar soporte a estas especificaciones se han integrado las siguientes librerías en nuestro framework:

- OpenCV, que proporciona diferentes aspectos de visión por computador, haciendo fácil la creación de algoritmos como reconocimiento de imágenes, reconocimiento de rostros, ...
- OpenGL, encargado de todo lo relacionado con la interfaz del usuario 2D y 3D

- Aruco, que se encarga de detectar marcadores y posicionar una cámara virtual devolviendo la matriz de perspectiva en formato OpenGL para su uso.
- libqrdecode encargado de decodificar códigos QR
- assimp, librería encargada de importar objetos 3D en diferentes formatos, junto a sus texturas y animaciones
- gsoap, librería para comunicaciones en red mediante protocolos SOAP
- libxml, librería para gestionar archivos o cadenas en formato XML
- Freetype, librería para gestionar tipografías
- CMU Sphinx, librería de reconocimiento de Voz
- Vuzix, librerías de soporte a dispositivos Vuzix
- pthread, librería para el desarrollo de aplicaciones con procesos paralelos, usada para el procesamiento en paralelo de aquellas tareas que necesiten de un tiempo de cálculo o transferencia grande, y no deban bloquear la interfaz del usuario u otras tareas.
- curl, librería para la transferencia de datos mediante sintaxis URL soportando protocolos como HTTP, HTTPS, FILE, FTP, FTPS, RTMP, SMTP, ... necesaria para conectarse con los servidores de la empresa para descargar la información necesaria

El desarrollo se ha realizado completamente bajo el paradigma de programación orientado a objetos en C++, y bajo un modelo dirigido por eventos. Es decir, tanto la estructura como la programación van dirigidos por los sucesos que ocurren en el sistema, ya sean interacciones del propio usuario o del hardware. Para llevar a cabo un esquema organizado y limpio la librería se divide en 6 módulos básicos que son Core, Devices, Events, UI, AR y Utils como se observa en el diagrama de clases (Fig. 3.2) y arquitectura (Fig. 3.1), y que albergan toda la lógica de la librería creada y que vamos a detallar a continuación.

### 3.2.1. Módulos Core y Eventos

El módulo Core contiene los objetos básicos del framework, de los que prácticamente todo el sistema depende. Existe una clase llamada 'Object' que es la clase básica de la

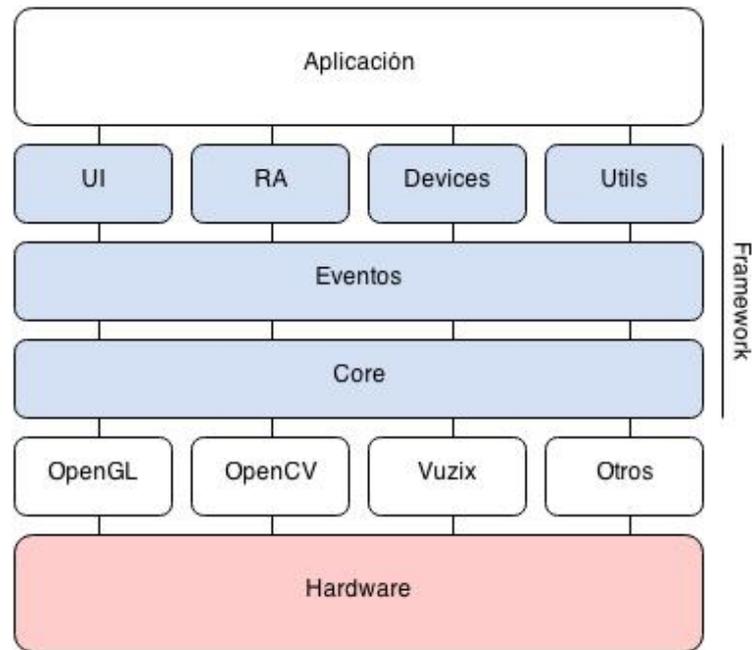


FIGURA 3.1: Diagrama de clases.

que el resto de clases debe heredar. Esta clase implementa básicamente la definición de un identificador para cualquier nuevo objeto creado en el sistema. Este identificador es un identificador numérico creado automáticamente en la construcción del objeto y único entre el resto de objetos. Además también se recoge el tipo del objeto como una cadena de caracteres y que es definido por la clase.

Como hemos comentado, la librería se centra en un modelo orientado a eventos (Fig. 3.3), y que viene definida en este módulo en las clases 'EventHandler' y 'EventDispatcher'. Para ello se hace uso de una pila FIFO, en la que se van almacenando todos los eventos disparados para ser ejecutados. Esta funcionalidad la implementa la clase 'EventHandler'. A su vez cada evento tiene asociado una serie de llamadas de respuesta o Callbacks que el desarrollador ha definido según sus necesidades, de tal forma que cuando un evento es disparado, todas las funciones (Callbacks) asociadas a dicho evento son llamadas y gestionadas por 'EventHandler'. Este tipo de modelo orientado a eventos proporciona una gran flexibilidad al desarrollador a la hora de decidir qué desea hacer cuando un evento ocurre, sin tener que definir un bucle general donde recoger todos los eventos, e incluso definir múltiples funciones para un mismo evento. Pero para que 'EventHandler' llame a los callbacks se deben disparar los eventos. Esto se hace a partir de la clases que hereden 'EventDispatcher', por ejemplo, si una nueva clase como por ejemplo un botón, necesita disparar un evento de ratón, esta clase botón debe heredar de

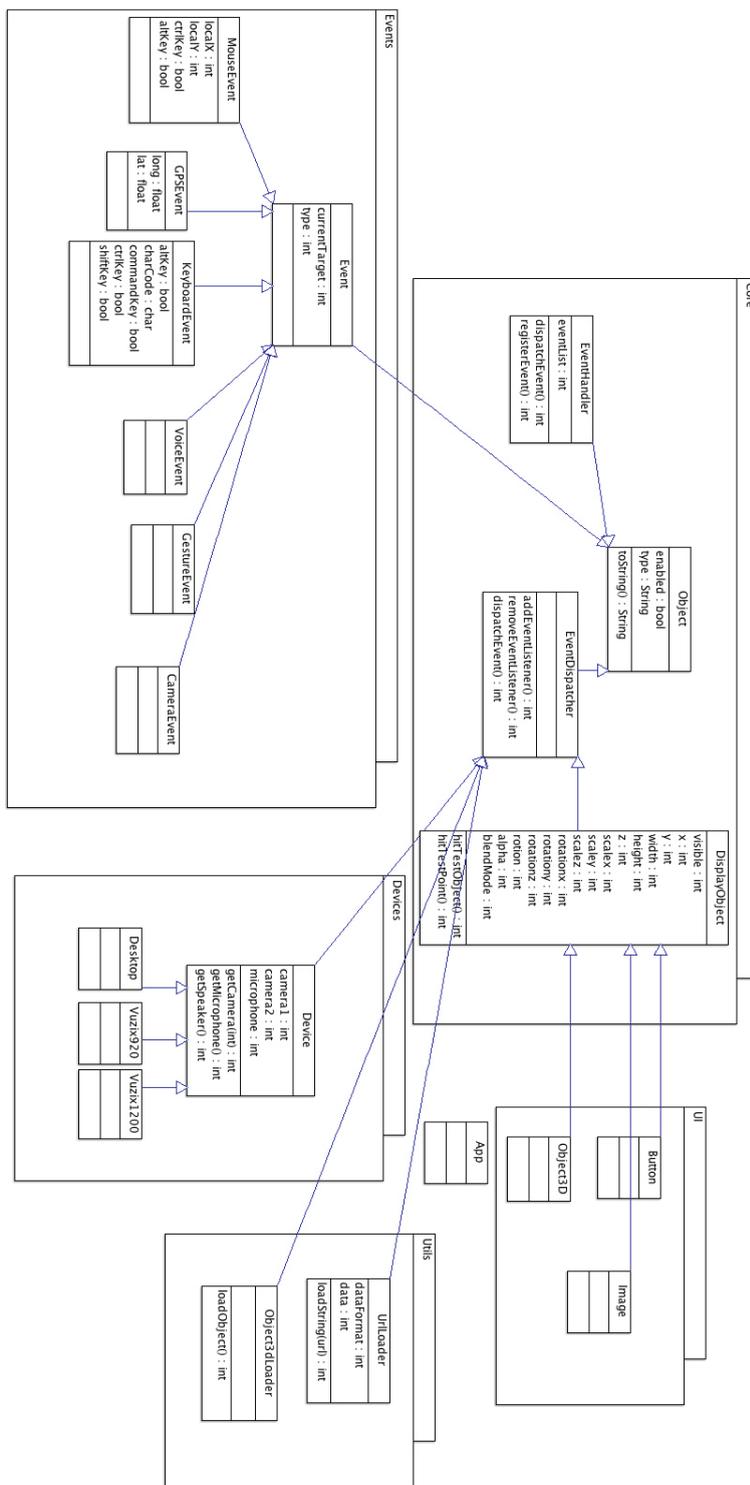


FIGURA 3.2: Diagrama de clases.

'EventDispatcher', de forma que al llamar a la función 'dispatchEvent' junto al evento ratón asociado.

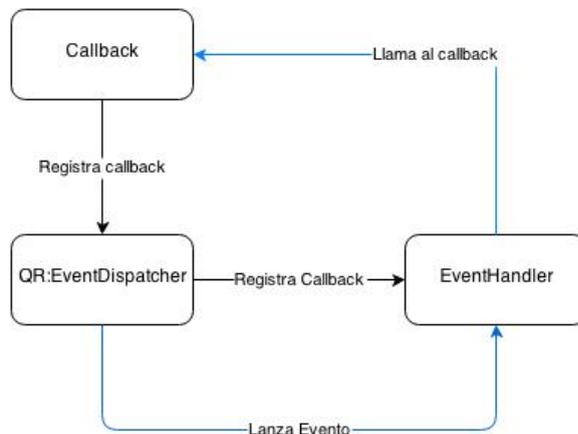


FIGURA 3.3: Flujo de eventos.

'EventHandler' se encargará de llamar a las funciones CallBack y gestionar la pila. Cabe destacar que 'EventHandler' es un objeto de solo una instancia y que la clase 'EventManager' es la única encargada de llamarla y ejecutar, de tal forma que un programador se debe abstraer de dicha clase.

Cada evento disparado, viene asociado con un objeto que almacena toda la información asociado al evento disparado. Por ejemplo, si es un evento de ratón, viene la posición del ratón, el botón pulsado etc. Si es un evento de QR, viene el texto decodificado si ha sido posible la de-codificación, etc. Los eventos definidos en el framework son:

- Cámara
- Gestos
- Teclado
- Menú
- Ratón
- QR
- Voz

De tal forma si en una aplicación queremos definir un callback asociado al reconocimiento de voz, definiremos primero un Callback, con la función que queremos ejecutar y asociaremos el callback con el objeto y el tipo de evento que se encargará de lanzar, como podemos ver en el siguiente ejemplo:

```
//Creamos el callback
Core::TCallback<MyApp> voiceCallBack;
//asociamos el callback con una función
voiceCallBack.SetCallback(this, &MyApp::voiceCB);
//Creamos el objeto de reconocimiento de voz
voiceRecognition= AR::VoiceRecognition::Instance();
//Asociamos el objeto con el callback y el evento de voz reconocido
voiceRecognition->addEventListener(
    Events::VoiceEvent::VOICE_RECOGNIZED,
    &voiceCallBack);
...
...
void voiceCB(Event *ev)
{
    VoiceEvent *ve=(VoiceEvent*)ev;
    textCommand->setText("Command: "+ ve->data);
    ...
}
```

### 3.2.2. Módulo interfaz de Usuario y componentes

El módulo interfaz de usuario y componentes se encuentra bajo la carpeta de UI, junto a la clase 'DisplayObject' del módulo 'Core' que se encarga de aplicar las bases de todos los objetos que puedan ser dibujados en pantalla.

Cada objeto de interfaz, se puede configurar para que se muestre en todas las pantallas del dispositivo de visión o en solo una de ellas. La opción de mostrar solo en una de ellas es cuando un objeto se encuentra en un lateral del campo de visión. No es necesario duplicarlo en las dos pantallas, sino con la pantalla del ojo correspondiente es suficiente para la correcta visión del texto y otros elementos gráficos, optimizando de dicha forma el renderizado de objeto.

Para el desarrollo de este proyecto se han creado los siguientes componentes:

- Botón, este componente permite crear botones con textos e iconos, junto a eventos de ratón (click, over, out).
- Diálogo Aceptar/Cancelar, este componente crea una ventana en la que se permite introducir un texto, junto a eventos de aceptar o cancelar.
- Dibujos básicos 2D, esta clase permite dibujar elementos básicos 2D como rectángulos, rectángulos con bordes redondeados y líneas.
- Texto, que permite de una forma optimizada, la posibilidad de seleccionar la tipografía y renderizar texto 2D en cualquier parte de nuestra interfaz.
- Aplicación base, esta clase es la base de cualquier aplicación a crear, y que configura los bucles y llamadas necesarias para cualquier proyecto.
- Imágenes 2D, esta clase permite crear, cargar y añadir imágenes en cualquier parte de nuestro dispositivo en formato y coordenadas pixels.
- Imágenes 3D, esta clase permite crear, cargar y añadir imágenes en un espacio 3D.
- Listas, componente que permite añadir listado de textos.
- Menú, permite la construcción de menús a partir de lectura XML o directamente en código, asociando a cada elemento del menú el evento correspondiente.
- Modelos 3D, este componente permite la carga y visualización de objetos 3D en nuestro campo de visión.

### 3.2.3. Módulo Dispositivos

Este módulo contiene los dispositivos implementados, en nuestro caso solamente se da soporte a Vuzix y escritorio. Todos los dispositivos son heredados de la clase Platform, que implementa las configuraciones básicas de cualquier entorno de realidad aumentada, con acceso a las cámaras y opciones de salida a pantalla mono o estereoscópica en modos progresivos o side by side. Posteriormente para cada dispositivo se añaden sus especificaciones, como la resolución en la que se debe renderizar, número de cámaras, acceso a ellas, si posee otro tipo de hardware para acceder a él...

De tal forma en la clase Vuzix se ha implementado el acceso a los dispositivos de dicha marca, implementando los dispositivos VR920 y 1200XL con cada peculiaridad.

Para crear una aplicación basada en la plataforma Vuzix solo debemos crear dicho objeto y pasarlo a nuestra aplicación base como mostramos en el siguiente trozo de código:

```
//Creamos el objeto
Devices::Vuzix *platform;
//Asociamos el objeto a la instancia
platform=Devices::Vuzix::Instance();
//Se inicializa con el tipo de dispositivo deseado
platform->Init(Devices::VUZIX_DEVICE_1200XL);
//Asociamos la plataforma a nuestra aplicación
app= new MyApp(platform);
```

### 3.2.4. Módulo AR

Este módulo implementa funciones directas de realidad aumentada, como puede ser detección de marcadores, detección y reconocimiento de QR, reconocimiento facial y de voz. Estas clases a menudo son heredadas de la clase 'EventDispatcher' que implementa las funciones básicas de registro y disparo de eventos como se ha mencionado en la sección 3.2.1.

Se han desarrollado completamente las siguientes funcionalidades:

- Reconocimiento de QR mediante la librería libdecodeqr (Fig. 3.4)
- Reconocimiento de voz mediante la librería CMU SPhinx
- Reconocimiento y detección de marcadores mediante la librería ArUco (Fig. 3.5)



FIGURA 3.4: Detección de códigos QR.



FIGURA 3.5: Reconocimiento y detección de marcadores.

### 3.2.5. Módulo Utilidades

Este módulo proporciona diferentes accesos y facilidades para completar un proyecto de realidad aumentada, permitiendo y dando facilidades al programador a utilizar herramientas comunes como acceso a contenidos online, lectura de archivos en formato XML, carga de objetos 3D, conexión con servicios web, acceso al sistema de ficheros, etc.

Las clases implementadas finalmente en este módulo para el proyecto son:

- Carga de objetos 3D mediante librería assimp
- Lectura de formatos XML mediante librería libxml2
- Carga de contenido online a través de direcciones URL, mediante librería cURL

## 3.3. Diseño de la prueba para el estudio de UX

El conjunto de interfaces y dispositivos del que se va a hacer uso para el experimento de experiencia de usuario se forma de:

- Computadora portátil en una mochila.
- Gafas Vuzix 1200XL o 920AR
- Ratón
- Micrófono

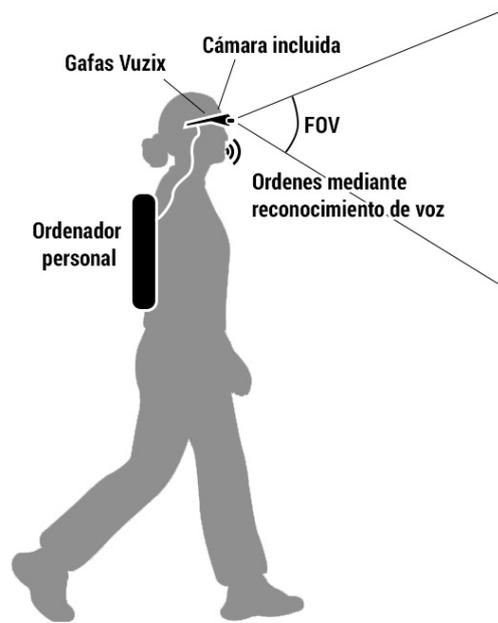


FIGURA 3.6: Esquema de dispositivos.

Como se muestra en la siguiente figura (Fig. 3.6):

El procedimiento a seguir para el estudio de experiencia de usuario, va a ser el siguiente:

- Una vez el dispositivo bien colocado y el sistema funcionando, el usuario recibe la información e instrucciones que debe realizar, recoger un elemento mostrado en una ventana flotante e indicado en un plano.
- El usuario con todos los dispositivos debe seguir las instrucciones del plano e ir al lugar indicado.
- El usuario recoge el objeto indicado en la ventana flotante de entre diferentes objetos similares.
- El usuario vuelve al punto de origen con todos los dispositivos.

Para el desarrollo de la aplicación de test se ha generado una aplicación simple basada en la librería mencionada anteriormente y creada para tal uso.

La aplicación consta de 3 pantallas, una pantalla inicial en la que el usuario ve una descripción del producto a recoger, descripción, título y precio del producto, y un elemento tridimensional de dicho objeto en movimiento rotatorio todo centrado en el campo de visión (Fig. 3.7).

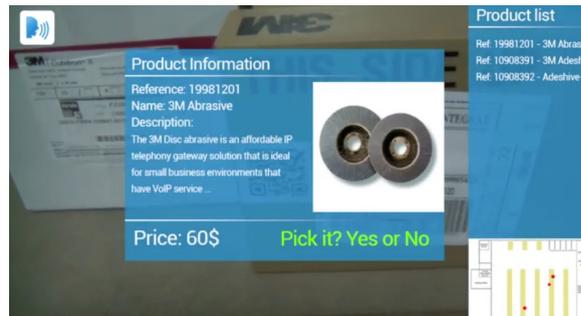


FIGURA 3.7: Información del producto a recoger.

Una segunda pantalla en la que dicha información se muestra en un lateral para que el usuario tenga referencia del producto a seleccionar (Fig. 3.8).



FIGURA 3.8: Información del producto a recoger en un lateral.

Una tercera pantalla que el usuario puede mostrar a necesidad y que es un plano de situación del producto dentro de un espacio cerrado (Fig. 3.9).



FIGURA 3.9: Información de la localización del producto a seleccionar.

Esta estructura se repite para los dos dispositivos usados en el proyecto, Vuzix 920 AR y Vuzix Star 1200 XL con la diferencia de que los productos son diferentes en cada dispositivo para que la persona deba centrar su visión en el objeto a recoger y poder evaluar correctamente aspectos como visibilidad de textos, imágenes, etc.

### 3.4. Desarrollo del PoC

El desarrollo de la prueba de concepto es un poco más complejo que el ejemplo anterior. Este sistema no solo depende directamente de sistema offline sino que se ha desarrollado un sistema online dónde toda la información está almacenada.

Este desarrollo consta de diferentes partes debido a su naturaleza. El objetivo de esta prueba de concepto es mostrar cómo un sistema de realidad aumentada con dispositivos de visión pueden conectarse a un sistema web en la nube donde se almacena toda la información, imágenes, objetos 3D, etc. Estos desarrollos darán soporte a la industria de almacenaje o warehouse como pueden ser empresas tipo Amazon, MRW, etc. Mejorando en cuanto a eficiencia y eficacia en el trabajo de recogida de productos de cada pedido realizado, de tal forma los pedidos son almacenados en la nube, mediante la propia web, o un tercer software. También se almacena la información de clientes, planos de la fábrica, productos, su stock y dónde están localizados...

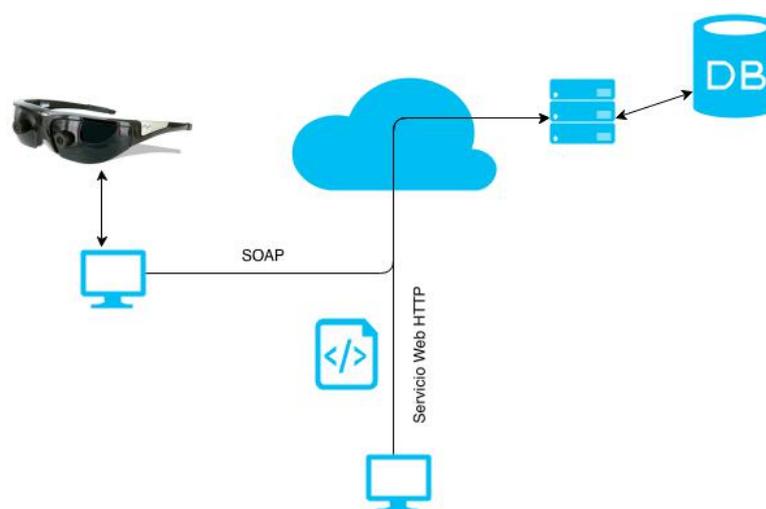


FIGURA 3.10: Diagrama sistema PoC

Para ello se ha desarrollado todo un sistema web (Fig. 3.11) en php con los frameworks yii [31], que es un framework para el desarrollo rápido de aplicaciones en php y bootstrap [38], que ofrece un conjunto de interfaces de usuario y componentes para el desarrollo web, todo esto junto al desarrollo del servicio soap y mediante la integración de la aplicación de realidad aumentada. Esto crea un ecosistema intercomunicado y multi-usuario como se muestra en la Figura 3.10, integrando cartera de clientes, stock de productos junto su localización (Fig. 3.12), planos del edificio...

En dicho ecosistema existen:

- Cliente de realidad aumentada: aplicación de realidad aumentada en ordenador personal, conectado a las gafas de realidad aumentada Vuzix 1200XL, y con conexión a internet para poder conectarse con los servicios en la nube.
- Cliente web: donde los usuarios registrados pueden dar de alta productos, clientes, ...
- Servidor web: en la que se almacena los elementos multimedia (imágenes, objetos 3D...) necesarios para describir productos, planos, ...



FIGURA 3.11: Página principal de bienvenida y gestión de stocks.

Este sistema tiene un proceso y flujo simple que se resume en los siguientes pasos:

- El dispositivo de realidad aumentada y pc al conectarse el usuario se identifica mediante un código QR, enviando los datos al servicio web mediante soap para autenticar el usuario (Fig. 3.13)
- Una vez identificado correctamente el usuario desde el servidor, el servidor le responde con el listado de órdenes a realizar
- El usuario da la acción de empezar con la primera orden mediante voz.
- El sistema muestra al usuario el listado de los productos a recoger y el primer producto a recoger mostrado en el plano de guía (Fig. 3.9)
- El usuario se desplaza hasta la localización indicada y recoge el producto, escaneándolo mediante el código QR (Fig. 3.4)



FIGURA 3.12: Página de descripción de producto

- Si el producto escaneado es incorrecto avisa al usuario. En caso contrario le indica el número de productos a coger y espera la confirmación del usuario
- Este proceso se repite hasta que se finaliza con la recogida de todos los productos de la orden. Al finalizar el sistema le indica al usuario que lleve los productos al punto de empaquetado y envío y continua con la siguiente orden.



FIGURA 3.13: PoC en funcionamiento

Se puede ver una demostración del PoC en funcionamiento en un entorno real en youtube en el siguiente enlace: <https://www.youtube.com/watch?v=oDUPL1rhYQ>

# Capítulo 4

## Validación

### 4.1. Participantes

Para el desarrollo de la evaluación y test se ha escogido 20 participantes de forma aleatoria entre amistades y se han citado todos en un mismo lugar desconocido a priori por los participantes y sin ningún contacto previo relacionado con este tipo de tecnologías. La muestra consta de un 60 % de hombres y un 40 % de mujeres con una edad media de  $33.35 \pm 4.52$ .

### 4.2. Medidas

Para la toma de medidas se han desarrollado dos cuestionarios. El primero es para en el primer momento en el que el usuario toma contacto con uno de los dos dispositivos, tabla [A.1](#), y evalúa la satisfacción con dicho dispositivo. El segundo dividido en dos partes evalúa la satisfacción con el segundo dispositivo, tabla [A.2](#), y un cuestionario de comparativa entre los dos tipos de dispositivos, tabla [A.3](#), después que el usuario ha tomado contacto con los dos dispositivos.

### 4.3. Procedimiento

Para llevar a cabo el test cada participante del test, por turnos, se colocaba una mochila que contenía todo el sistema hardware necesario, un portátil, ratón, salidas VGA para

conexión de los dispositivos visuales, etc. Cada usuario realizaba el test con el dispositivo que se le había asignado aleatoriamente entre OST o VST (Vuzix Star 1200XL o Vuzix 920AR respectivamente). Una vez debidamente equipado, se cronometraba el tiempo que tardaba desde que visualizaba el objeto que tenía que recoger, el lugar en que se encontraba, dirigirse hacia él, recogerlo y traerlo de vuelta. Una vez realizado el recorrido procedía a rellenar el cuestionario correspondiente. Finalizados los dos cuestionarios (A.1 y A.2) procedía a rellenar el cuestionario de comparativa (A.3).

Todos los usuarios y dispositivos hacían siempre el mismo recorrido y de esta forma mantener las mismas distancias y problemáticas de entorno como espacios angostos, puertas, giros... Los productos a recoger eran diferentes en cada caso, y se diferenciaban únicamente en su forma, manteniendo el mismo color y disposición, haciendo que el participante se centrara en qué tipo de producto debía recoger y de la misma forma poder centrar al usuario en preguntas que posteriormente debía rellenar en el cuestionario.

#### 4.4. Resultados

Para el estudio de los datos se ha hecho uso del paquete estadístico OpenSource R. Realizando diferentes análisis de los datos obtenidos. El fin del estudio es poder seleccionar cuál es el dispositivo que mejor se ajusta a tareas relacionadas con la recolección o selección de productos en entornos de trabajo. Por ello debemos hacer hincapié en los resultados que diferencien los dos sistemas VST y OST.

El primer análisis que vamos a realizar es el tiempo empleado en recoger un elemento/producto. Para ello realizamos una comparativa de los tiempos empleados por los usuarios la primera vez que toman contacto con un dispositivo de visión, comparando los tiempos empleados con OST y VST. Podemos observar en la figura 4.1 como las medianas y desviaciones son muy similares (48.5 +-14.8 OST, 50.5 +- 9.4 VST) no habiendo prácticamente diferencia entre los dos tipos de dispositivos.

Si nos centramos en solo este tiempo como referencia, podríamos indicar que no existe ningún tipo de diferencia ni mejora en la experiencia de usuario en cuanto a tiempo y por consiguiente productividad entre los dispositivos.

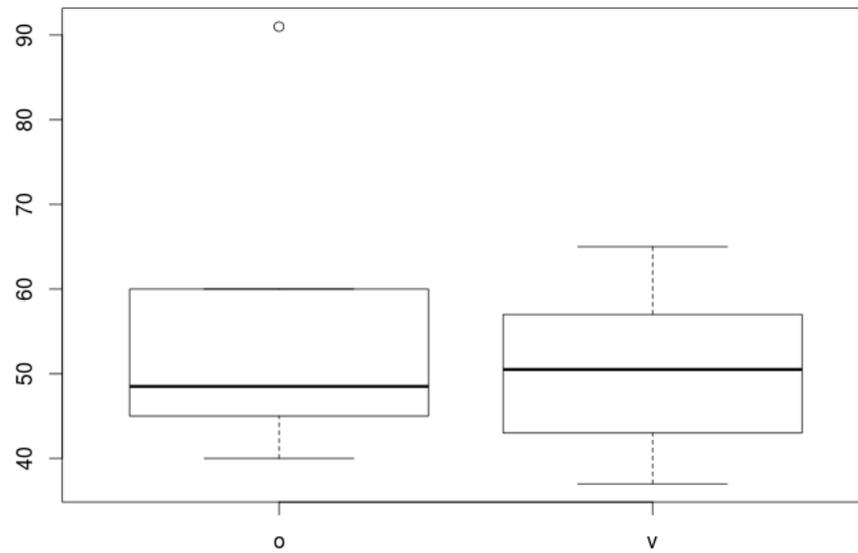


FIGURA 4.1: BoxPlot T1.

Pero si analizamos los tiempos empleados cuando los usuarios utilizan el segundo sistema, sí que observamos en la figura 4.2 cómo las medianas y desviaciones poseen una diferencia sustancial ( $34 \pm 3.3$  OST,  $46.5 \pm 9.1$  VST).

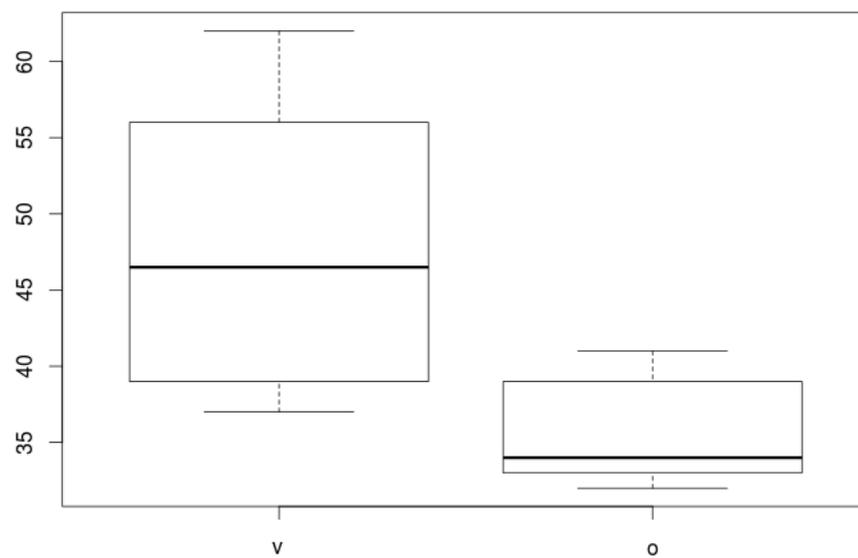


FIGURA 4.2: BoxPlot T2.

Para demostrar la clara diferencia entre los dos sistemas presentamos el análisis one-way ANOVA, tabla 4.1, tanto del tiempo empleado por los participantes en el primer contacto con esta tecnología (T1), y el tiempo empleado una vez ha probado uno de los dos sistemas (T2).

	Df	Sum Sq	Mean Sq	F value	Pr
T1	1	51.2	51.2	0.332	0.572
T2	1	696.2	696.2	14.6	0.00125 **

CUADRO 4.1: one-way Anova. T1 tiempo primer contacto con un dispositivo y T2 tiempo segundo contacto con el otro dispositivo

Podemos observar como el valor del T2 es muy inferior a 0.01 y que representa una diferencia estadísticamente significativa entre los dos sistemas.

Esta diferencia es debida a que en un primer momento, cualquier usuario que nunca ha probado un sistema de visión, no está acostumbrado, y pierde un cierto tiempo explorando mínimamente el sistema y hardware, haciendo que el tiempo del dispositivo OST sea equivalente al de VST. La segunda vez que se ponen un dispositivo de visión, aun siendo el dispositivo diferente del primero, el usuario no explora tanto el sistema realizando la tarea de forma más directa y tal vez podríamos decir, más productiva, centrándose en la tarea a realizar y no en el dispositivo.

En esta segunda recogida de tiempos en la que se ve una diferencia de tiempos entre un dispositivo VST y un dispositivo OST (Fig. 4.2). Debido a que la percepción del entorno con un dispositivo VST viene determinado por el FOV y las lentes por sus características, captura y calibración de la cámara, realizando una distorsión en la visión del usuario y dificultando sus movimientos, tanto al andar como al recoger un producto. Mientras que mediante el dispositivo OST, al ser transparente y únicamente superponer en la visión del usuario los elementos gráficos, no distorsiona el entorno y percepción del usuario.

Para el análisis comparativo entre los dos sistemas a partir de la encuesta realizada a cada usuario vamos a usar la función t-test que muestra si existen diferencias significativas entre los diferentes dispositivos, anotando con \*\* aquellas cuestiones que poseen diferencias estadísticas significativas con un  $p < 0.05$ .

En el primer cuestionario, realizado a los participantes una vez han realizado el test sobre el primer dispositivo asignado aleatoriamente observamos en la tabla 4.2 cómo existen diferencias significativas en la cuestión Q1.3 y Q1.7 del cuestionario 1 (A.1),

	OST	VST	t	df	p
Q1.1	4.1	4.1	0	18	1
Q1.2	4.3	5.0	0.7579	17.679	0.4585
Q1.3	5.0	4.1	3.8571	9	0.003864 **
Q1.4	2.3	2.8	-0.9358	17.939	0.3619
Q1.5	4.3	3.3	2,1108	13.305	0.05426
Q1.6	3.9	3.3	0.9909	17.832	0.335
Q1.7	4.8	4.1	2.6047	14.311	0.0205 **
Q1.8	4.0	3.9	0.2873	16.81	0.7774
Q1.9	4.2	4.2	0	14.918	1
Q1.10	4.2	3.9	0.7006	16.317	0.4934

CUADRO 4.2: T-Test. Preferencia de sistemas

en concreto en las preguntas: 'Me ha parecido fácil' y 'He manipulado sin problemas los objetos reales' en la que los usuarios que han probado inicialmente el sistema OST consideran dicho sistema más fácil y con mayor facilidad de manipulación de objetos.

	VST	OST	t	df	p
Q2.1	3.44444	4	-1.6925	13.883	0.1128
Q2.2	4	4.4	-0.8729	14.833	0.3966
Q2.3	4.11111	4.6	-1.0241	12.16	0.3257
Q2.4	2.66666	2.2	0.6991	16.459	0.4942
Q2.5	3.77778	4.2	-0.6537	16.159	0.5225
Q2.6	3.33333	4.3	-1.6314	10.359	0.1328
Q2.7	4	4.5	-1.0742	12.519	0.303
Q2.8	3.77778	4.6	-2.0964	14.41	0.05415
Q2.9	3.44444	4.5	-2.2807	14.804	0.03781 **
Q2.10	3.66667	4.4	-1.866	16.681	0.07972

CUADRO 4.3: T-Pair Test. Preferencia de sistemas

En el segundo cuestionario (A.2) el usuario ya ha probado anteriormente uno de los sistemas, y acaba de probar el otro, de forma que tiene el concepto de los dos sistema y su contestación tiene una mínima dependencia de su experiencia. En este segundo test se observa que la única diferencia significativa se encuentra en la pregunta Q2.9 y que muestra cómo los usuarios recomiendan para el caso de estudio que estamos trabajando el dispositivo OST.

Para finalizar se analiza un tercer cuestionario (A.3) en que se compara directamente los dos sistemas en busca de la selección del usuario por una u otra preferencia de dispositivo para el caso de uso estudiado.

	X-squared	df	p
QC1	28.9	2	<0.0001 **
QC2	22	3	<0.0001 **
QC3	7.3	2	0.02599
QC4	16.3	2	0.0002887 **
QC5	24.1	2	<0.0001 **
QC6	10.9	2	0.004296 **
QC7	10.9	2	0.004296 **

CUADRO 4.4: ChiSquare. Preferencia de sistema

Como podemos observar en la tabla 4.4 existe gran diferencia entre los sistemas seleccionados por el usuario ( $p < 0.001$ ) siendo el sistema seleccionado por los participantes de la encuesta el dispositivo OST Vuzix STAR 1200 XL. La única pregunta en la que no existe diferencia estadística significativa es la QC3 '¿En cuál se veían mejor las imágenes/texto?' en la que los participantes no tenían preferencia por ninguno de los dispositivos, ya que las mejoras en los dispositivos OST han conseguido unas visualizaciones de los elementos gráficos correctos. Aunque no tienen la legibilidad y claridad que en un dispositivo VST en la que la imagen proyectada al usuario es una composición de la señal capturada por la cámara del dispositivo y los elementos gráficos, pudiendo realizar una mejor visualización de los mismos.

## Capítulo 5

# Conclusión y trabajos futuros

### 5.1. Conclusión

En esta tesina se ha presentado un estudio comparativo de dos dispositivos de visión de realidad aumentada de diferente tipología como son los dispositivos de visión Optical See Through y los dispositivos de visión Video See Through para un caso de uso concreto que es el de manipulación y recogida de objetos, y extensible a otros marcos como grandes almacenes en que se deben gestionar órdenes de ventas, o empresas de paquetería, etc.

El objetivo de esta tesina es analizar los diferentes aspectos relacionados con estos dispositivos para poder seleccionar el dispositivo más adecuado a partir de los estudios realizados.

Después del análisis estadístico de los datos recogidos en los estudios (Capítulo 4) podemos concluir que el uso de dispositivos de visión Optical See Through son más adecuados y aptos para la manipulación y recogida de objetos, haciendo la manipulación de objetos, movimientos y desplazamientos del usuario por su entorno de trabajo mas cómodo, sencillo y productivo que el uso de dispositivos Video See Through.

A partir de este trabajo también se ha creado el framework, empleado en la empresa Augmate para el desarrollo de prototipos que aglomera todas las tecnologías necesarias para desarrollar aplicaciones de realidad aumentada de forma rápida y sencilla.

A partir de este framework se han desarrollado diversas aplicaciones basadas en marcos con descarga de contenidos online, lectores de códigos QR, aplicaciones de escritorio en 3D, etc.

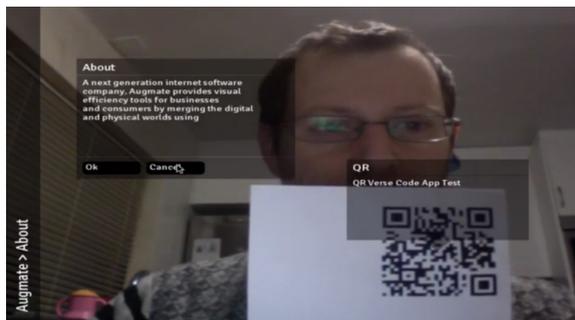


FIGURA 5.1: Ejemplo aplicación QR

## 5.2. Trabajos Futuros

Debido a la próxima aparición en el mercado de nuevos dispositivos con mejores especificaciones, y de diferente tipología (monoculares como Google Glass y Vuzix M100), este estudio puede ser explorado en estos nuevos dispositivos. Pudiendo comparar estos nuevos dispositivos monoculares OST (google glass) o VST (Vuzix M100), dejando futuras investigaciones de comparativas entre dispositivos binoculares OST, monoculares OST y monoculares VST.

También se derivan de este estudio trabajos sobre el uso de sistemas de posicionamiento en interiores IPS y las diferentes comparativas entre estos sistemas para su uso en sistemas de picking en grandes superficies de almacenes. También el estudio de interfaces de usuario, en los que se debe estudiar el uso de tipografías y tamaños óptimos para el uso en dispositivos tanto OST como VST, monoculares o binoculares.

## 5.3. Repercusión profesional

Augmate, es una startup norteamericana fundada en 2010. En 2011 me solicitó el desarrollo del framework Verse, framework opensource, para situarse en el mercado del desarrollo de software de realidad aumentada especializado en dispositivos visuales.

Durante el desarrollo y lanzamiento de la startup se trabajaron en diferentes frameworks, posibilidades y productos. A partir de una propuesta mía al desarrollo final por mi parte del ecosistema comentado en esta tesina, el desarrollo de un framework flexible y el prototipo y prueba de concepto de un almacén o empresa de pickup.

El desarrollo de la primera versión del framework se trabajó desde finales del 2011 hasta principios de 2013, siendo descartado por mí, debido a problemas de base derivados de desarrollos anteriores a mi acceso. A partir de ahí se tomó la decisión de crear el nuevo framework que se desarrolló desde principios de 2013 hasta mediados del mismo. En agosto de 2013 todo el equipo de Augmate se concentró en Nueva York para desarrollar el producto de concepto (PoC) desarrollando yo mismo la plataforma web y dando las bases de la integración del framework en los dispositivos Vuzix M100, además del ya desarrollado Vuzix 1200XL.

A finales de Agosto de 2013 se mantuvieron diversas reuniones con inversores. Todos ellos muy interesados en el proyecto y el PoC. También se llegaron, gracias a los PoC desarrollados, a contratos con diversas multinacionales como Toyota y otras recibiendo ingresos necesarios para continuar los desarrollos.

Un inversor derivado de las reuniones mantenidas conmigo y el equipo de Augmate en el verano de 2013, mantuvo contacto con la empresa Augmate, decidiendo finalmente en el verano de 2014 invertir 2.000.000 de dólares, lanzando de esta forma finalmente la empresa Augmate. Para más información sobre la empresa Augmate y su desarrollo <http://www.augmate.com> y mi perfil linkedin <http://es.linkedin.com/pub/david-mill%C3%A1n-escriv%C3%A1/1b/263/801>.

# Apéndice A

## Apéndice

### A.1. Cuestionario 1

Hombre/Mujer

Edad:

OST/VST

Código:

Te vamos a hacer unas preguntas para conocer tu opinión sobre el sistema utilizado.

Puntúa del 1 al 5 (1= totalmente en desacuerdo, 5= Totalmente de acuerdo)

Nº	Pregunta	Respuesta
1.	Me gustaría utilizar este sistema más veces	-
2.	Me ha parecido útil	-
3.	Me ha parecido fácil	-
4.	Necesitaría una persona de apoyo para usar el sistema	-
5.	Me ha resultado cómodo	-
6.	Las imágenes/texto las he visto bien (nítidas)	-
7.	He manipulado sin problemas los objetos reales	-
8.	No he tenido problemas para ver el mundo real y las imágenes/texto	-
9.	Recomendaría este sistema para su uso	-
10.	Puntúa el sistema de 1 a 5	-

CUADRO A.1: Tabla A1. Cuestionario 1

## A.2. Cuestionario 2

Datos generales:

OST/VST

Código:

Te vamos a hacer unas preguntas para conocer tu opinión sobre el sistema utilizado.

Puntúa del 1 al 5 (1= totalmente en desacuerdo, 5= Totalmente de acuerdo)

Nº	Pregunta	Respuesta
1.	Me gustaría utilizar este sistema más veces	-
2.	Me ha parecido útil	-
3.	Me ha parecido fácil	-
4.	Necesitaría una persona de apoyo para usar el sistema	-
5.	Me ha resultado cómodo	-
6.	Las imágenes/texto las he visto bien (nítidas)	-
7.	He manipulado sin problemas los objetos reales	-
8.	No he tenido problemas para ver el mundo real y las imágenes/texto	-
9.	Recomendaría este sistema para su uso	-
10.	Puntúa el sistema de 1 a 5	-

CUADRO A.2: Tabla A2. Cuestionario 2

Compara los dos sistemas

Nº	Pregunta	Optical	Video
1.	¿Cuál te ha resultado más fácil?	-	-
2.	¿Cuál es el más cómodo?	-	-
3.	¿En cuál se veían mejor las imágenes/texto?	-	-
4.	¿En cuál se veía mejor la combinación del mundo real y las imágenes/texto?	-	-
5.	¿En cuál has manipulado mejor los objetos reales?	-	-
6.	¿Recomendarías alguno?. Si es que sí, ¿cuál?	-	-
7.	¿Cuál te ha gustado más? ¿Por qué?	-	-
8.	¿Qué cambiarías? Añade los comentarios que consideres.	-	-

CUADRO A.3: Tabla A3. Cuestionario 2 comparativa

# Bibliografía

- [1] Gessler A., Schulze T., Kulling K., and Nadlinger D. Assimp (<http://assimp.sourceforge.net/>), 14-9-2014.
- [2] Kirillov A. Gratf (<http://www.aforgenet.com/projects/gratf/>), 14-9-2014.
- [3] Medina A., Gómez I., Romera M., Gómez J., and Dorrozero E. Indoor position system based on bitcloud stack for ambient living and smart buildings. *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, 82:127–136, 2012.
- [4] arlab. arlab (<http://www.arlab.com/>), 14-9-2014.
- [5] arpa solutions. Arpa sdk (<http://www.arpa-solutions.net/>), 14-9-2014.
- [6] ARToolkit. Artoolkit (<http://www.hitl.washington.edu/artoolkit/>), 14-9-2014.
- [7] Neea A.Y.C., Onga S.K., Chryssolourisb G., and Mourtzisb D. Augmented reality applications in design and manufacturing. *CIRP Annals - Manufacturing Technology*, 61(2):657–679, 2012.
- [8] Blaise B. Pthreads (<https://computing.llnl.gov/tutorials/pthreads/>), 14-9-2014.
- [9] bitstars. Droidar (<https://github.com/bitstars/droidar>), 14-9-2014.
- [10] cURL. curl (<http://curl.haxx.se/>), 14-9-2014.
- [11] Universidad de Cordova. Aruco (<http://www.uco.es/investiga/grupos/ava/node/26>), 14-9-2014.
- [12] Epson. Moverio bt-200 (<http://www.epson.com/cgi-bin/store/jsp/landing/moverio-bt-200-smart-glasses.do>), 14-9-2014.
- [13] Google. Google glass (<https://www.google.com/glass/start/>), 14-9-2014.

- 
- [14] Weingarten J. and Siegwart R. EKF-based 3d slam for structured environment reconstruction. *Intelligent Robots and Systems*, pages 3834 – 3839, 2005.
- [15] Panta K. and Armstrong J. Indoor localisation using white leds. *Electronic Letters*, 48(4):228–230, 2012.
- [16] khronos. Opengl (<http://www.opengl.org>), 14-9-2014.
- [17] KitWare. Cmake (<http://www.cmake.org/>), 14-9-2014.
- [18] Carozza L., Tingdahl D., Bosché F., , and Gool L. Markerless visio. based augmented reality for urban planning. *Computer-Aided Civil and Infrastructure Engineering*, pages 2–17, 2012.
- [19] Teixeira L., Raposo A. B., and Gattass M. Indoor localization using slam in parallel with a natural marker detector. *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, pages 18 – 22, 2013.
- [20] layar. layar (<https://www.layar.com/>), 14-9-2014.
- [21] LibXML2. libxml2 (<http://xmlsoft.org/>), 14-9-2014.
- [22] Billinghurst M. Hands and speech in space: multimodal interaction with augmented reality interfaces. *Proceedings of the 15th ACM on International conference on multimodal interaction*, pages 379–380, 2013.
- [23] Meta. Space glasses. meta pro (<https://www.spaceglasses.com/>), 14-9-2014.
- [24] metaio. junaio (<http://www.junaio.com/>), 14-9-2014.
- [25] metaio. metaio sdk (<http://www.metaio.com/sdk/>), 14-9-2014.
- [26] Hagbi N., Bergig O., El-Sana J., and Billinghurst M. Shape recognition and pose estimation for mobile augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, 17(10):1369 – 1379, 2011.
- [27] Georgia Institute of Techonology. Dart (<http://ael.gatech.edu/dart/>), 14-9-2014.
- [28] OpenCV. Opencv (<http://www.opencv.org>), 14-9-2014.
- [29] Rolland J. P. and Fuchs H. Optical versus video see-through head-mounted displays in medical visualization. *Presence: Teleoperators and Virtual Environments*, 9(3):287–309, 2000.

- 
- [30] Rolland J. P., Holloway R., and Fuchs H. A comparison of optical and video see-through head-mounted displays. *SPIE*, 2351:293–307, 1994.
- [31] Xue Q. Yii framework (<http://www.yiiframework.com/>), 14-9-2014.
- [32] Qualcomm. Vuforia sdk (<https://developer.vuforia.com/>), 14-9-2014.
- [33] Hyun C. R., Chang H. S., and Myung J. C. Rapid slam using simple map representation in indoor environment. *Frontiers of Computer Vision*, pages 225 – 229, 2013.
- [34] Gammeter S., Gassmann A., Bossard L., Quack T., and Van Gool L. Server-side object recognition and client-side object tracking for mobile augmented reality. *Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1 – 8, 2010.
- [35] Mann S. "glass eyes": the theory of eyetap digital eye glass. *IEEE Technology and Society*, 31(3):10–14, 2012.
- [36] sologicolibre. Atomic authoring tool (<http://atomic-project.sourceforge.net/>), 14-9-2014.
- [37] Nishi T. libdecodeqr (<https://github.com/josephholsten/libdecodeqr>), 14-9-2014.
- [38] Twitter. Bootstrap (<http://getbootstrap.com/>), 14-9-2014.
- [39] Carnegie Mellon University. Cmu sphinx (<http://cmusphinx.sourceforge.net/>), 14-9-2014.
- [40] Florida State University. Gsoap (<http://www.cs.fsu.edu/engelen/soap.html>), 14-9-2014.
- [41] Vuzix. Vuzix, view the future (<http://www.vuzix.com>), 14-9-2014.
- [42] wiktitude. wiktitude (<http://www.wiktitude.com/>), 14-9-2014.
- [43] Zou Y., Chen W., Wu X., and Liu Z. Indoor localization and 3d scene reconstruction for mobile robots using the microsoft kinect sensor. *Industrial Informatics (INDIN)*, pages 1182 – 1187, 2012.
- [44] Balint Z., Kiss B., Magyari B., and Simon K. Augmented reality and image recognition based framework for treasure hunt games. *IEEE 10th Jubilee International Symposium on Intelligent Systems and Informatics (SISY)*, pages 147 – 152, 2012.