

INTERFAZ DE REALIDAD AUMENTADA PARA LA INTERNET DE LAS COSAS



Omar Eduardo Guzmán Álvarez

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Sistemas
Popayán
2016

INTERFAZ DE REALIDAD AUMENTADA PARA LA INTERNET DE LAS COSAS



Omar Eduardo Guzmán Álvarez

TRABAJO DE GRADO

Presentado como requisito para optar al título de Ingeniero en Electrónica y
Telecomunicaciones

Director

PhD. Miguel Ángel Niño Zambrano

Co-Director: PhD. Gustavo Adolfo Ramírez González

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones

Departamento de Sistemas

Popayán, Noviembre 2016

AGRADECIMIENTOS

Gracias a Dios, a él todo el poder, el honor y la gloria por siempre. A mi madre Fabiola, a mi Director y Amigo Miguel Ángel Niño Zambrano, cuya valiosa guía y conocimiento me hicieron mejor profesional y sobre todo mejor persona, a su señora esposa Elizabeth Granados, y de manera especial a los ingenieros Gustavo Ramírez y Carlos Cobos los cuales fueron un gran apoyo junto con mis familiares para llevar a feliz término este proyecto.



TABLA DE CONTENIDO

1. INTRODUCCIÓN	1
1.1. CONTEXTO GENERAL.....	1
1.2. DECLARACION DEL PROBLEMA.	1
1.3. ESCENARIO DE MOTIVACIÓN.	2
1.4. OBJETIVOS.....	3
1.4.1. <i>Objetivo general.</i>	3
1.4.2. <i>Objetivos específicos.</i>	3
1.5. CONTRIBUCIONES Y ALCANCE DEL PROYECTO.....	3
1.6. ESTRUCTURA DE LA MONOGRAFIA.	4
2. MARCO DE REFERENCIA PARA LA CONSTRUCCION DE INTERFACES DE REALIDAD AUMENTADA PARA LA INTERNET DE LAS COSAS.....	7
2.1. INTRODUCCION.....	7
2.2. LINEAMIENTOS DEL MARCO DE REFERENCIA.....	7
2.3. CONCEPTOS FUNDAMENTALES.....	8
2.3.1. <i>Datos, metadatos e información en la IoT.</i>	8
2.3.3. <i>Servicio inteligente.</i>	9
2.3.4. <i>Interfaces multimodales.</i>	9
2.3.5. <i>Realidad aumentada AR (Augmented Reality).</i>	10
2.3.6. <i>Contexto.</i>	10
2.3.7. <i>Objetos inteligentes - OI.</i>	11
2.3.8. <i>Localización y mapeo simultaneo SLAM.</i>	11
2.3.9. <i>Interacción semántica.</i>	12
2.4. CARACTERISTICAS DEL DESARROLLO DE APLICACIONES AR EN LA IOT.....	12
2.5. METODOLOGIA PARA LA CREACION DEL MARCO DE REFERENCIA.....	14
2.6. ANALISIS DE MODELOS Y METODOLOGIAS.	14
2.6.1. <i>Modelos de desarrollo de software vs características de Interfaces en AR.</i>	15
2.6.2. <i>Metodologías de requerimientos.</i>	16
2.6.3. <i>Metodologías de desarrollo.</i>	16
2.6.4. <i>Conclusiones finales acerca de los modelos y metodologías analizadas.</i>	19
2.7. TRABAJOS RELACIONADOS.	20
2.7.1. <i>Exploración de tecnologías y metodologías en los trabajos relacionados.</i>	20
2.7.2. <i>Trabajos relevantes.</i>	22
2.7.3. <i>Criterios acerca de la escogencia del software de desarrollo para aplicaciones en AR para la IoT.</i>	29
2.7.4. <i>Hitos y consideraciones en la creación de interfaces de AR para la IoT.</i>	29
2.7.5. <i>Brechas encontradas en el análisis de trabajos relacionados.</i>	31
2.8. DISEÑO DE INTERFACES EN AR PARA LA IoT.	32
2.8.1. <i>Naturaleza del software.</i>	32
2.8.2. <i>Interacción del usuario en AR.</i>	33
2.8.2.1. <i>Uso de la metáfora.</i>	34
2.8.2.2. <i>Elementos de Asistencia.</i>	36
2.8.3. <i>Hardware y dispositivos para la captura de la escena.</i>	39
2.8.4. <i>Métodos de Visualización.</i>	43
2.8.5. <i>Técnicas de Detección y Seguimiento.</i>	43
2.8.6. <i>Consideraciones finales acerca del diseño de interfaces de AR para la IoT.</i>	45
2.9. USABILIDAD Y EXPERIENCIA DE USUARIO ORIENTADAS A INTERFACES DE AR PARA LA IoT.	47
2.9.1. <i>Concepto de Usabilidad.</i>	47



2.10.	CONCLUSIONES.....	53
3.	GUIA PARA EL DESARROLLO DE INTERFACES DE REALIDAD AUMENTADA PARA LA INTERNET DE LAS COSAS.....	55
3.1.	COMPONENTES.....	55
3.2.	FILOSOFIA Y VALORES.	55
3.3.	CUESTIONAMIENTOS A RESOLVER.....	56
3.4.	GUIA DE DESARROLLO.....	57
3.4.1.	<i>Fase 1. Analisis</i>	57
3.4.2.	<i>Fase 2. Diseño</i>	58
3.4.3.	<i>Fase 3. Construcción</i>	60
3.4.4.	<i>Fase 4. Validación</i>	61
3.5.	CONSIDERACIONES FINALES.	62
4.	INTERFAZ DE REALIDAD AUMENTADA PARA LA INTERNET DE LAS COSAS	63
4.1.	INTRODUCCION Y APORTES.	63
4.3.	ANALISIS.....	63
4.3.1.	<i>Contextualización</i>	63
4.3.2.	<i>Requerimientos</i>	66
4.4.	DISEÑO.....	68
4.4.1.	<i>Establecimiento de tecnologías</i>	68
4.4.2.	<i>Plan de prototipos</i>	71
4.5.	CONSTRUCCION.....	72
4.5.1.	<i>Solución general</i>	72
4.6.	PROTOTIPO1.....	73
4.6.1.	<i>Solución para la historia de usuario 1</i>	73
	• Escena 1. Contexto.....	74
	• Escena 2. Consulta.....	76
4.6.2.	<i>Solución para la historia de usuario 2</i>	79
4.6.3.	<i>Solución para la historia de usuario 3</i>	81
	• Escena 3: Modo AR.....	82
4.6.4.	<i>Solución Historia de usuario 4</i>	83
	• Escena 4 Gestor de ECA.....	85
	• Escena 5 Crear ECA.....	86
4.7.	PROTOTIPO 2.....	90
4.7.1.	<i>Observaciones generales para la creación del prototipo 2</i>	90
4.7.2.	<i>Solución General</i>	91
4.7.3.	<i>Solución para la escena 1</i>	93
4.7.4.	<i>Solución para la escena2</i>	93
4.7.5.	<i>Solución para la escena 3</i>	94
4.7.6.	<i>Solución para la escena 4</i>	95
4.7.7.	<i>Solución para la escena 5</i>	96
4.8.	PROBLEMAS, SOLUCIONES Y CONCLUSIONES.....	97
5.	VALIDACION Y EVALUACION DE USABILIDAD	99
5.1.	CONSIDERACIONES.....	99
5.2.	PROPUESTA DE EVALUACION DE USABILIDAD.....	99
5.2.1.	<i>Fase 1. Validación del prototipo 1 (Funcionalidad)</i>	99
5.2.2.	<i>Fase 2. Evaluación de usabilidad del prototipo 2 (Usabilidad)</i>	101
6.	CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO	107
6.1.	CONCLUSIONES.....	107



6.3. TRABAJO FUTURO 108



LISTA DE FIGURAS

FIGURA 1. CONCEPTO GAFAS <i>GOOGLE GLASS</i>	9
FIGURA 2. CONTINUO DE LA VIRTUALIDAD (<i>VIRTUALITY CONTINUUM</i>).....	10
FIGURA 3. TEMÁTICAS QUE DEFINEN LOS LINEAMIENTOS DEL MARCO DE REFERENCIA.....	14
FIGURA 4. METODOLOGÍAS DE DESARROLLO UTILIZADAS POR LOS TRABAJOS RELACIONADOS.....	20
FIGURA 5. IDENTIFICACIÓN DE ELEMENTOS DE DISEÑO EN LOS TRABAJOS RELEVANTES.....	22
FIGURA 6. WIDGET MINI MAPA (DERECHA), FLECHAS INDICADORAS ANCLADAS AL ENTORNO (IZQUIERDA).....	24
FIGURA 7. MANUAL DE INSTRUCCIONES AR PARA CAFETERA.....	25
FIGURA 8. ARQUITECTURA IoT PARA PERSONAS CON DISCAPACIDAD.....	26
FIGURA 9. INTERFAZ REALITY EDITOR.....	27
FIGURA 10. MODELO DE COMPONENTES DEL SISTEMA GALLAG STRIP.....	28
FIGURA 11. DIFERENTES TECNOLOGÍAS DE REALIDAD AUMENTADA.....	33
FIGURA 12. EVOLUCIÓN DE LA INTERACCIÓN EN AR.....	34
FIGURA 13. RETÍCULA EN DISPOSITIVO ESTEREOSCÓPICO.....	35
FIGURA 14. INDICADORES AUXILIARES Y RADAR EN UNA APLICACIÓN DE AR.....	36
FIGURA 15. HUD SOBRECARGADO.....	37
FIGURA 16. SUPERPOSICIONES VISUALES.....	38
FIGURA 17. ZOOM SEMÁNTICO.....	39
FIGURA 18. ESPECTRO DE DISPOSITIVOS DE REALIDAD AUMENTADA Y VIRTUAL (2015).....	41
FIGURA 19. COMPARATIVA DE DISPOSITIVOS AR VS VR.....	42
FIGURA 20. PROYECCIÓN REALISTA.....	43
FIGURA 21. REALIDAD AUMENTADA POR MARCADORES.....	44
FIGURA 22. REALIDAD AUMENTADA BASADA EN LOCALIZACIÓN.....	44
FIGURA 23. USABILIDAD DENTRO DE LA EXPERIENCIA DE USUARIO.....	48
FIGURA 24. DESAVENENCIAS ENTRE UTILIDAD Y USABILIDAD.....	50
FIGURA 25. COMPONENTES DE LA GUÍA DE DESARROLLO.....	55
FIGURA 26. GUIA DE DESARROLLO.....	57
FIGURA 27. VISTA ESTÁTICA DE LA ARQUITECTURA DE INTERACCIÓN SEMÁNTICA.....	65
FIGURA 28. ESCENARIO DE INTERACCIÓN SEMÁNTICA.....	66
FIGURA 29. METADATOS DEL MIDDLEWARE XIVELY.....	69
FIGURA 30. CÓDIGO QR “SALA”.....	75
FIGURA 31. IDENTIFICACIÓN DEL ESCENARIO – ESCENA 1. FUENTE PROPIA.....	76
FIGURA 32. SERVICIO WEB ÍNDICE DE DOMÓTICA. FUENTE PROPIA.....	77
FIGURA 33. MATRIZ DE DATOS OBTENIDA DEL SERVICIO WEB. FUENTE PROPIA.....	78
FIGURA 34. UBICACIÓN DE LOS SCRIPTS PARA LA LECTURA DE LAS RESPUESTAS XML.....	80
FIGURA 35. DATOS Y METADATOS EN LA ESCENA 2.....	81
FIGURA 36. CÓDIGOS QR DE CADA OI.....	82
FIGURA 37. ESCENA 3 DATOS Y METADATOS.....	83
FIGURA 38. ESCENA 3 ESTADO DEL SERVICIO BÁSICO ENCENDIDO.....	84
FIGURA 39. CARPETA ECA DEL SISTEMA DE ARCHIVOS DEL OI.....	85
FIGURA 40. ESCENA 4 SERVICIOS DE INTERACCIÓN. FUENTE PROPIA.....	86
FIGURA 41. VARIABLES DEL SCRIPT <i>ECAEVENTO.CS</i>	87
FIGURA 42. ESCENA 5 NOMBRE DEL SERVICIO DE INTERACCIÓN ECA. FUENTE PROPIA.....	88
FIGURA 43. ESCENA 5 – SELECCIÓN DEL RECURSO EVENTO Y ESTABLECIMIENTO DE LA CONDICIÓN DEL SERVICIO DE INTERACCIÓN ECA.....	89
FIGURA 44. ESCENA 5 SELECCIÓN DEL RECURSO ACCIÓN, SU ESTADO Y LA CREACIÓN DEL SERVICIO DE INTERACCIÓN.....	89
FIGURA 45. IMÁGENES <i>FADES</i> INICIALES.....	91
FIGURA 46. ESCENA DE TRANSICIÓN <i>LOADER</i> ENTRE ESCENAS.....	92
FIGURA 47. ELEMENTOS MODIFICADOS DEL HUD.....	92
FIGURA 48. ESCENA 1.....	93
FIGURA 49. ESCENA 2.....	94



Interfaz de Realidad Aumentada Para la Internet de las Cosas.

FIGURA 50. ESCENA 3.....	95
FIGURA 51. ESCENA 4, FUNCIÓN DESCRIPCIÓN.....	96
FIGURA 52. ESCENA 5.....	97
FIGURA 53. PROCESO DE VALIDACION DEL PROTOTIPO 1.....	100
FIGURA 54. PLAN DE PRUEBAS PROTOTIPO 2.....	101
FIGURA 55. ESCALA DE PROMEDIOS SUS.....	104
FIGURA 56. RESULTADOS DE LA ENCUESTA EN LAS PREGUNTAS 11 Y 12.....	105



LISTA DE TABLAS

TABLA 1: COMPARATIVA ENTRE MODELOS DE DESARROLLO CLÁSICOS.	15
TABLA 2: ELEMENTOS DE LAS METODOLOGÍAS SCRUM Y XP.....	17 ¹⁸
TABLA 3. RELACIÓN ENTRE APLICACIONES EN AR, LAS METODOLOGÍAS DE DESARROLLO Y LAS TECNOLOGÍAS UTILIZADAS.....	21
TABLA 4. EVALUACIÓN DE USABILIDAD EN ALGUNOS TRABAJOS RELACIONADOS.....	52
TABLA 5. CARACTERÍSTICAS DE TECNOLOGÍAS DE IDENTIFICACIÓN.....	74
TABLA 6. UMBRALES DE LOS SERVICIOS BÁSICOS DE LOS OI.....	83
TABLA 7. PUNTAJE DE USABILIDAD SUS ENTRE LOS SUJETOS DE PRUEBA.....	104



1. INTRODUCCIÓN

1.1. CONTEXTO GENERAL.

La Internet de las Cosas IoT (*Internet of Things*) permite conectar las “cosas” existentes de nuestro mundo real a la internet, dándole la posibilidad a los usuarios de identificarlas e interactuar con ellas, sin embargo, esta tecnología aún es emergente y con varios vacíos por resolver [1]–[4] . Al integrar sensores en las cosas, se pueden obtener mediciones de propiedades y fenómenos del entorno con la precisión que logran las máquinas, ofreciendo la posibilidad de crear mejores y nuevos servicios en la Web para las personas y otras aplicaciones, a esto último se le ha llamado la Web de las Cosas WoT (*Web of Things*)[5].

Los teléfonos inteligentes (*Smartphones*) [6]–[8] literalmente se comunican unos a otros, estos vienen equipados con gran cantidad de sensores que permiten al usuario interactuar de manera inteligente con el entorno, a través de un conjunto de tecnologías de comunicación, procesamiento y almacenamiento. Este concepto, aplicado a diferentes cosas de la vida real, abre un mundo de posibilidades para generar interacciones entre objetos y personas, a estos se les denomina objetos inteligentes OI (*Smart Things*) [5].

Actualmente, en el desarrollo de la IoT se busca no solo la interacción de las personas con los objetos inteligentes OI, sino también la interacción entre los mismos objetos [9]–[11] para generar nuevos servicios y más complejos al usuario, sin embargo, la gran variedad de tecnologías hardware y software, así como la poca estandarización [4] generan problemas de heterogeneidad [12] impidiendo que se puedan conectar los objetos de la IoT de manera transparente. Esto en parte se ha solucionado con la creación de servidores middleware [5], [13], incorporación de técnicas semánticas en la WoT [14] y las estandarizaciones como ODF y OMI, aportadas por The Open Group [15], [16].

Por otra parte, los desarrollos tecnológicos a nivel de percepción de la realidad e interfaces que interactúan con el mundo real han ido avanzando de manera paralela a lo anteriormente mencionado, desde la aparición de la internet, la evolución de los *Smartphones*, pasando por los videojuegos y la realidad virtual, poco a poco se observa que estas tecnologías tienden a converger para generar más y mejores servicios a través de interfaces intuitivas y fáciles de usar, donde la ampliación de los sentidos, la necesidad de asistencia para lograr ciertas tareas, el entretenimiento entre otros ha creado la necesidad de crear conceptos como el de realidad aumentada AR (*Augmented Reality*) [17].

1.2. DECLARACION DEL PROBLEMA.

Uno de los principios de la computación ubicua es la interacción fácil, intuitiva y transparente, de los usuarios con su entorno [18], sin embargo, no se han explorado suficientemente los mecanismos para que el usuario interactúe de manera natural con los objetos de su entorno y con los servicios generados por la interacción entre los recursos de la IoT. Actualmente las interfaces existentes son de interacción compleja, en una sola vía, con muchos pasos para realizar acciones, lo cual crea frustración al usuario, dificultando el manejo los objetos, y por ende impide que la IoT



se evolucione más rápidamente, además si el usuario tiene alguna limitación física o cognitiva estos problemas se acentúan [3].

Haciendo una analogía con el internet como se conoce hoy en día, los navegadores web son la forma más natural de poder interactuar con la información de la Web, permitiendo ocultar la heterogeneidad y las interacciones a las que hay lugar en ella, sin la necesidad de intervención por parte de usuarios especializados. De igual manera, para la WoT es necesario construir soluciones que permitan una adecuada forma de interacción por parte del usuario con los objetos de la IoT y los servicios de información que se pueden aprovechar en un entorno dado. Así surge la pregunta de investigación: *¿Cómo crear una interfaz que permita al usuario interactuar de manera natural y transparente con los objetos de la IoT y sus servicios en la WoT?* Teniendo en cuenta que los servicios son los ofrecidos por los OI o también los generados por la interacción entre ellos mismos.

El enfoque de solución de la presente propuesta es explorar la creación de una interfaz que utilice técnicas de AR, como mecanismo para lograr la interacción M2H (*Machine to Human*) de una forma intuitiva y fácil de usar, tal como se ha podido evidenciar su aporte en otros campos como el entrenamiento y publicidad [17], [19], [20].

1.3. ESCENARIO DE MOTIVACIÓN.

El diario vivir, viene marcado por una serie de características que permite observar profundos cambios en el comportamiento humano y en la manera de realizar sus actividades cotidianas a lo largo de la historia; la tecnología es uno de esos factores que cambia de manera drástica los comportamientos entre las personas, culturas e incluso la naturaleza misma del ser humano; se pueden observar cambios en los hogares y sitios de trabajo antes y después de aparecer los primeros televisores y radios, o con los primeros ordenadores personales, los comportamientos humanos antes y después de aparecer el teléfono móvil y posteriormente los teléfonos inteligentes (*Smartphones*).

La AR es una tecnología que permite expandir los sentidos y la percepción del entorno, sin embargo, aún se encuentra en evolución, es una tecnología emergente ya que constantemente se está integrando a otras tecnologías y a dispositivos de fácil acceso como los Smartphones, esto hace que se abran muchas posibilidades de integración con la IoT, ya que esta tecnología también se encuentra en evolución y se puede acceder a ella con mayor facilidad día con día.

Actualmente existe una gran competencia entre distintos desarrollos que involucran AR tanto a nivel Hardware como Software, y es previsible que, a través de la aceptación del público, la evolución del software y su campo o campos de dominio, muchas de estas tecnologías desaparezcan y otras se masifiquen en su consumo.

El reto de crear interfaces de AR para la IoT puede brindar una solución capaz de permitir la interacción entre objetos inteligentes y la creación de nuevos servicios, en este caso particular haciendo uso de la arquitectura propuesta por Niño-Zambrano [11] y el escenario de interacción semántica definido por Guerrero-Riobamba [21], se realiza una exploración documental que permite determinar los elementos metodológicos y de diseño más adecuados para la creación de



interfaces AR en la IoT, generando una aplicación en AR para dicho escenario, utilizando tecnología accesible y con referencia a lo utilizado por la comunidad científica.

1.4. OBJETIVOS.

1.4.1. Objetivo general.

Crear una interfaz de realidad aumentada intuitiva y fácil de usar, con el fin de soportar la interacción M2H (*Machine to Human*) en la Internet de las Cosas, ofreciendo servicios de información para el usuario, apoyados en el modelo de interacción semántica definido por Niño-Zambrano [11].

1.4.2. Objetivos específicos.

- Definir un marco de referencia que permita desarrollar una interfaz de realidad aumentada intuitiva y fácil de usar para la IoT.
- Crear una interfaz de realidad aumentada basándose en el marco de referencia propuesto que permita la gestión de interacciones semánticas y la visualización de datos enfocándose en:
 - i. Detectar Objetos Inteligentes en un espacio inteligente.
 - ii. Visualizar la información básica y servicios de los objetos inteligentes.
 - iii. Proveer la interfaz que permite definir nuevos servicios, producto de la interacción entre los objetos inteligentes, basados en el trabajo de Niño-Zambrano [11].

Evaluar la interfaz obtenida respecto a la usabilidad centrada en que tan fácil e intuitiva es la experiencia del usuario, utilizando indicadores definidos para usabilidad.

1.5. CONTRIBUCIONES Y ALCANCE DEL PROYECTO.

- **Marco de referencia para la creación de interfaces de AR para la IoT.**

A partir de la documentación, estudio y análisis de diferentes referencias tanto bibliográficas como tipo web, se construyó un marco de referencia que permite:

- i. Evidenciar que modelos de software o que características de los mismos brindan una base científica a la hora de crear interfaces en AR.
- ii. Analizar diferentes metodologías y recoger los factores que permiten crear una guía de desarrollo en interfaces de AR para la IoT.
- iii. Identificar tecnologías y elementos de diseño utilizados por la comunidad científica en los trabajos relacionados que permiten una construcción de aplicaciones en AR orientadas a la gestión de recursos IoT.

- **Guía de desarrollo de interfaces de AR para la IoT.**

A partir del marco de referencia propuesto se indican una serie de pasos y elementos que permiten el desarrollo de aplicaciones en AR para la IoT, con una base teórica sobre distintos modelos y metodologías junto con los elementos de diseño adecuados para este dominio.



- **Interfaz de AR para la interacción semántica de OI en la IoT.**

Mediante el uso de las tecnologías para el desarrollo de AR más adecuadas, sobre la base del escenario de interacción semántica de Guerrero-Riobamba [21], y utilizando la guía de desarrollo propuesta en esta monografía se crea una aplicación sencilla e intuitiva que permite la gestión de objetos inteligentes OI en el escenario definido.

1.6. ESTRUCTURA DE LA MONOGRAFIA.

De acuerdo al contexto planteado, se han definido las siguientes secciones:

Capítulo 2.

Se construye un marco teórico con los conceptos fundamentales que abren paso a la temática y hacen más sencilla su comprensión, además de identificar las características básicas para un desarrollo de AR orientado a gestionar recursos de la IoT, luego se hace un recorrido por los modelos de software y metodologías más adecuados para este tipo de desarrollos, los trabajos relacionados con los cuales se extraen los elementos de diseño, las tecnologías y finalmente un apartado que orienta la usabilidad a este tipo de desarrollos.

Capítulo 3.

Se propone una serie de pasos como guía de desarrollo de interfaces de AR para la IoT, basados en el marco de referencia construido en el capítulo anterior, haciendo ciertas consideraciones y delimitando la propuesta a este tipo de aplicaciones.

Capítulo 4.

Siguiendo la guía de desarrollo propuesta, se desarrollan 2 prototipos funcionales de la interfaz de AR para la IoT, sobre el escenario de interacción semántica de Guerrero-Riobamba [21], abordando su desarrollo mediante escenas.

Capítulo 5.

Con base en el capítulo 2 de este documento se propone una evaluación de la usabilidad orientada a validar el prototipo 1 según su utilidad de acuerdo al marco de referencia propuesto, seguidamente la facilidad de uso y la intuición del prototipo 2.

Capítulo 6.

Se establecen los resultados de la evaluación del prototipo 2, las conclusiones finales del desarrollo y el trabajo futuro.

Adicionalmente, en los anexos se presenta información complementaria sobre los temas tratados en los capítulos, así:

Anexo A.

Se muestra el modelo y las metodologías de requerimientos analizadas, junto con las historias de usuario creadas.

Anexo B

Se describe la implementación de la herramienta de AR escogida sobre el entorno de desarrollo seleccionado.



Anexo C.

Se describe la implementación del servicio web sobre el entorno de desarrollo escogido.

Anexo D.

Se describen los códigos y algoritmos creados para las funcionalidades de la interfaz de AR para la IoT.

Anexo E.

Se describe la implementación del intermediario (*bróker*) IoT utilizado en el escenario de interacción semántica de Guerrero-Riobamba [21] sobre el entorno de desarrollo utilizado.

Anexo F.

Se describe el formato de los mensajes utilizados por el escenario de interacción semántica de Guerrero-Riobamba[21] para la comunicación de los recursos IoT involucrados en dicho escenario.

Anexo G.

Se muestra el procedimiento para acceder y modificar las carpetas y archivos del directorio raíz de los recursos físicos IoT desplegados sobre el escenario de interacción semántica de Guerrero-Riobamba[21].

Anexo H.

Se presentan dos algoritmos desarrollados para el escenario de interacción semántica de Guerrero-Riobamba [21], para solucionar la necesidad de implementar ciertas funcionalidades para la interfaz propuesta.

Anexo I.

Describe los artefactos software prefabricado para facilitar la construcción de ciertos elementos necesarios para el despliegue de algunas escenas en el entorno de desarrollo escogido.

Anexo J.

Describe las observaciones encontradas en el proceso de validación del prototipo 1 de este desarrollo.

Anexo K.

Muestra el despliegue de la evaluación de la usabilidad del prototipo 2 de esta propuesta.





2. MARCO DE REFERENCIA PARA LA CONSTRUCCION DE INTERFACES DE REALIDAD AUMENTADA PARA LA INTERNET DE LAS COSAS

2.1. INTRODUCCION.

Actualmente la AR se ha convertido en un campo de desarrollo de interfaces de información que están más cercanas a los usuarios y que se adaptan de manera ideal a las nuevas tecnologías, pudiendo brindar a estos la posibilidad de interactuar de forma natural y transparente con la IoT, y poder generar nuevos servicios en la WoT.

Existen proyectos en la IoT que están incursionando en desarrollos tipo AR con el fin de solucionar los problemas de interacción existentes de los recursos IoT con los usuarios. Sin embargo, éstos desarrollos son creados a la medida de sus propios objetivos, utilizando metodologías tácitas o poco formalizadas, lo cual obliga a los desarrollos posteriores tener que hacer un doble esfuerzo para superar los retos de crear interfaces AR y más aún en el dominio de la IoT.

El presente estudio pretende formalizar o proponer un marco de referencia para interfaces AR, con el fin de que se convierta en una herramienta para los grupos de desarrollo que tengan por objetivo crear interfaces de AR que impliquen un alto grado de interacción con los usuarios, es decir, no solo de información, sino de ejecución de servicios, ya que es esa interacción la que potencializa el concepto de ubicuidad[18].

2.2. LINEAMIENTOS DEL MARCO DE REFERENCIA.

Un punto de partida para estructurar el marco de referencia propuesto es estudiar las metodologías y modelos existentes para el desarrollo de software. La ingeniería de software brinda maneras de afrontar un problema de modo sistemático, disciplinado y cuantificable; esto permite que las soluciones que se pretenden dar a un problema tengan una fuerte fundamentación científica, sean eficaces y eficientes[22].

Para tal fin se establecen modelos y metodologías, los modelos o paradigmas de desarrollo, a través de una abstracción de la realidad, establecen un ciclo de vida del software con el cual se puede diseñar una solución de forma efectiva, brindándole escalabilidad y buena calidad [23]; las metodologías implementan patrones de diseño específicos para obtener soluciones adaptadas a las necesidades del cliente y a los recursos disponibles.

Para generar conocimiento sobre una temática de manera eficaz y con bases fuertes se debe poder contar con una referencia histórica sobre estudios e investigaciones previas, la experimentación y los resultados obtenidos por otros investigadores junto con las herramientas teóricas que permiten describir su investigación [22], a esto se le conoce como un marco de referencia, inclusive la inventiva o la innovación no se dan si el investigador no tiene herramientas que le permitirán llevar a cabo el planteamiento o teoría para la solución a algún problema.



La AR es un campo del conocimiento encaminado a enriquecer nuestro entorno real a través de nuestros sentidos, las soluciones en AR son orientadas principalmente hacia dominios como la educación, entretenimiento, cuidado de la salud, investigación civil o militar, asistencia y entrenamiento principalmente [24], [25], la IoT es una tecnología que combinada con AR puede brindar soluciones para mejorar la calidad de vida de las personas [11]; los artículos científicos, libros y repositorios de conocimiento junto con tutoriales, páginas web y la experiencia personal, se convierten en una buena fuente para crear un marco de referencia o proponer una guía de desarrollo para generar interfaces de AR de mediana o alta complejidad alrededor de la IoT.

A continuación se establecen los conceptos básicos, se definen las características que debería abordar un modelo de desarrollo de software para aplicarlo al proceso de creación de una aplicación en AR, se analizan los modelos de desarrollo existentes con base a estas características, se analizan las metodologías más utilizadas por la comunidad científica a la hora de proponer soluciones en este campo del conocimiento, junto con los trabajos relacionados, para así crear un marco de referencia que se pueda tomar como base para el desarrollo de interfaces en AR para la IoT.

2.3. CONCEPTOS FUNDAMENTALES.

2.3.1. Datos, metadatos e información en la IoT.

Un dato es una representación simbólica de alguna variable cuantitativa o cualitativa, es una representación de la realidad en cualquier nivel [26], un metadato es un dato acerca de otro dato, en particular para la IoT un metadato describe los datos acerca de un recurso, es análogo a una ficha bibliográfica donde se describen datos acerca de un libro, cabe notar que estos datos se encuentran en su estado más puro sin procesar, y su utilidad es validada por la entidad que tiene interés sobre esos datos, en particular para la IoT los metadatos son la base para la interacción semántica entre OI.

Por ejemplo: un dispositivo de localización mediante sistema de posicionamiento global GPS (*Global Positioning System*) donde se muestran datos acerca de la posición de un individuo sobre el planeta tierra, aquí la entidad de interés es el punto donde está ubicado el usuario, que puede ser su casa, el colegio o su lugar de trabajo, los datos los muestra el dispositivo GPS, en GPS las dos maneras más comunes de formato de coordenadas son los grados decimales y los grados sexagesimales (grados, minutos y segundos), estos formatos son datos acerca de los datos GPS, es decir sus metadatos.

El procesamiento de los datos da origen al concepto de información, este proceso surge de un interés por parte de una entidad que puede darse o no dependiendo de los hechos que condicionan el entorno, esto se conoce como contexto. La información lleva ciertas características como: semántica o significado, importancia o relevancia, vigencia, validez o fiabilidad y valor [27][11], sin embargo para la IoT ciertas de estas características se pueden perder cuando no se presenta la información al usuario de manera adecuada, cambiando la utilidad de la información [26], de aquí la necesidad de tener interfaces que brinden este servicio.

2.3.2. Metáfora.

En diseño de interfaces de usuario es un tipo de analogía con el mundo real donde se utiliza un concepto que le es familiar al usuario, permitiéndole transferir sus experiencias del mundo real al uso de algún dispositivo como un computador [28], las más conocidas son la metáfora del escritorio en los sistemas operativos y la metáfora del papel en blanco para los editores de texto.

2.3.3. Servicio inteligente.

Un servicio inteligente se presenta cuando los agentes que prestan el servicio son capaces de tomar decisiones efectivas en un entorno bien definido, así los servicios inteligentes pueden ir desde un servicio de control básico de temperatura, hasta un escenario de interacción semántica entre OI de la IoT [11], [21].

2.3.4. Interfaces multimodales.

Son interfaces que pueden interpretar diferentes capacidades sensoriales a la entrada de un sistema y crear comunicación de la información obtenida a través de diferentes salidas para obtener mecanismos naturales de interacción hombre-máquina[29].

El ejemplo más sencillo y aunque no fue una tecnología pionera si fue la primera en popularizarse y generar y generar expectativa en el público fueron las gafas inteligentes de google (*google glass*)¹ las cuales consistían en un prototipo de gafas transparentes, con múltiples entradas como control táctil, micrófono, táctil, micrófono, cámara, acelerómetro, brújula, y salidas como inducción ósea para escuchar a través de los huesos, través de los huesos, permitía grabar video y aceptaba comandos de voz [24], como se observa en la

Figura 1 ~~Figura 1~~, sin embargo se suspendió su soporte y venta a partir de 2014.

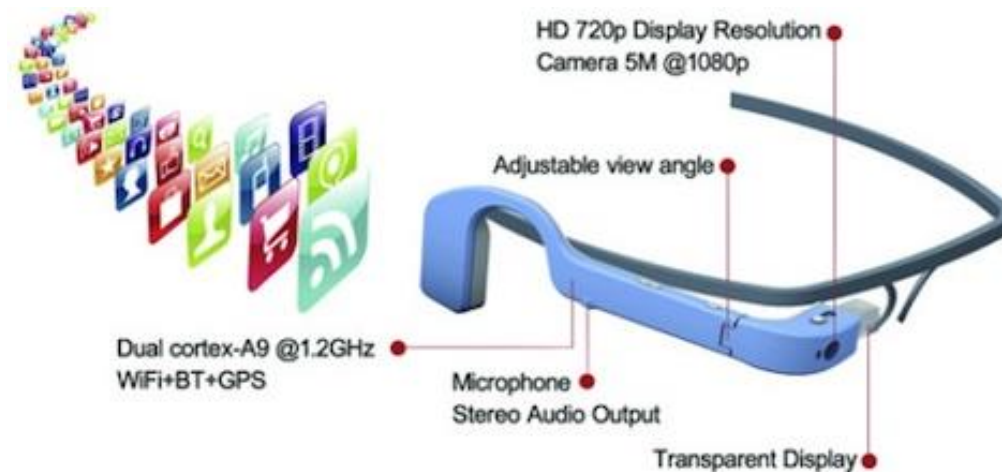


Figura 1. Concepto Gafas Google Glass.

Fuente: <http://www.cookingideas.es/imagenes/2014/01/def1.jpg?6f1304>

¹ <http://gglassday.com/google-glass-toda-la-informacion-que-es-google-glass-para-que-sirven/>



2.3.5. Realidad aumentada AR (Augmented Reality).

La interacción humano computador HCI (*Human Computer Interaction*) es la rama del conocimiento que indica las formas más usables y naturales para desplegar la información, actualmente se pueden establecer los paradigmas de interacción como: la computación de escritorio, la realidad virtual VR (*Virtual Reality*), la computación ubicua, y la AR [30], esta última consiste en poder tener una presentación de nuestro entorno de forma enriquecida, vinculando objetos del mundo virtual hacia el mundo real [17], [31], logrando así la expansión de nuestros sentidos, la forma más común de hacerlo es enriqueciendo nuestro campo de visión a través de una pantalla, pero cabe aclarar que no es la única manera, también se pueden aumentar o enriquecer otros sentidos o crear nuevas sensaciones, inclusive más útiles cuando el usuario tiene una limitación ya sea sensorial o de otra índole [25], apoyándose en otros campos tecnológicos como la IoT [3].

La AR es una opción que combinada con otras tecnologías puede ser un factor de enriquecimiento de la experiencia de usuario en un escenario definido o contexto [3], permitiendo gestionar entornos sin la necesidad de un usuario experto, en particular facilitaría la visualización de la información que brindan los recursos IoT, sus interacciones si las hay, y la generación de nuevos servicios al usuario, cumpliendo con una de las características de la ubicuidad que es la transparencia[18].

Azuma [17] Define la AR como una variación de la realidad virtual - VR, donde al combinar objetos virtuales con el mundo real, se realiza la percepción y la interacción del usuario con este. Los tres aspectos que define el autor para la realidad aumentada son: Combinación de elementos reales y virtuales, interacción en tiempo real, y registro en tres dimensiones.

P. Milgram y F. Kishino [32] ubican la AR en una posición especial en una recta continua, en la cual en un extremo se encuentra el entorno real y en el otro el entorno virtual, para el caso de la AR se tiene un entorno real enriquecido con elementos virtuales como lo muestra la Figura 2. La realidad mixta MR (*Mixed Reality*) también conocida como realidad híbrida HR (*Hybrid Reality*) combina el mundo virtual y el mundo físico en tiempo real, permitiendo interacción de objetos y personas reales y virtuales a lo largo y en cualquier espacio entre los extremos del continuo de la virtualidad.



Figura 2. Continuo de la virtualidad (*Virtuality Continuum*).

2.3.6. Contexto.

Es importante definir en que consiste el contexto, las máquinas deben saber dónde están [18], aspectos como localización y la escala son las bases de esta dimensión de realidad, sin embargo el



concepto de contexto es un poco más complejo, así se comparte la definición de contexto de Dey y Abowd, et al. [33] en la que dicen que el contexto² es “cualquier información que puede ser usada para caracterizar la situación de una entidad. Una entidad es una persona, lugar, u objeto que es considerado relevante para la interacción entre el usuario y la aplicación en sí misma”. Así el contexto se puede ver como un conjunto de atributos que se pueden recopilar no sólo de los sensores, sino también de las personas y las entidades de interés. En un escenario de interacción dado. También Dey y Abowd, et al. [33] definen el contexto - consciencia³ (*context-aware*) como: “Un sistema es de contexto – consciencia si usa el contexto para proveer información relevante y/o servicios para el usuario, dónde la relevancia depende de las tareas del usuario”. Así, los sistemas que se enfoquen a la colaboración, necesariamente deben gestionar el contexto, por lo cual en su arquitectura se debería incorporar la orientación basada en el contexto – consciencia.

Con respecto al tipo de contexto, Perera, et al. [13] hace un estudio muy detallado del mismo, enfocándose principalmente en el contexto de los sensores, es decir información de metadatos y datos medidos. Distingue dos tipos de contexto (primario y secundario) agrupados en las dimensiones de: localización, identidad, tiempo y actividad. El contexto primario maneja datos sin procesar y el contexto secundario realiza algún tipo de procesamiento a esos datos para generar información. Sin embargo, el contexto para el presente trabajo es más amplio que la información técnica de los sensores y de los datos que generan sus mediciones, así el contexto incluye también la conciencia del escenario de interacción y de los usuarios que pueden ser personas, aplicaciones u otros objetos, los cuales se pueden gestionar con ontologías de dominio.

2.3.7. Objetos inteligentes - OI.

Según Estrada-Martinez y Garcia-Macias [4] los OI son aquellos objetos que pueden detectar y procesar información, proporcionar servicios y comunicarse con otros objetos, un objeto se considera inteligente si posee memoria, capacidad de comunicación y procesamiento[5]. La propuesta de Niño-Zambrano [21] amplía el concepto de OI a recursos de la IoT que incorporan en su infraestructura sensores, actuadores y controladores al mismo tiempo con capacidad de razonamiento basado en una ontología llamada Objeto Semántico - OOS, llamándolos objetos semánticos. Este concepto es utilizado con el fin de establecer las propiedades básicas de los objetos de la IoT con los que se van a trabajar.

2.3.8. Localización y mapeo simultáneo SLAM.

SLAM (Simultaneous Localization and Mapping) es un conjunto de técnicas y cuestionamientos acerca de cómo un recurso u objeto al encontrarse en un entorno aleatorio, puede obtener información acerca de su ubicación y crear una representación dinámica de dicho entorno [34][35][36] . Un ejemplo claro de este conjunto de técnicas es la tecnología Tango Project⁴ de Google.

² Traducción libre de la definición original de contexto.

³ Traducción libre de la definición original de contexto – consciencia.

⁴ <https://www.google.com/atap/project-tango/>



2.3.9. Interacción semántica.

La interacción semántica de objetos de la WoT se puede entender como la capacidad de dos o más objetos inteligentes de la WoT para comunicarse entre sí en un contexto bien definido, exhibiendo comportamientos cooperativos, con el fin de resolver requerimientos de los usuarios, aplicaciones y otros objetos de la WoT [21]. Bajo esta definición la solución de este proyecto debe contemplar la forma de presentar la información útil como producto de la comunicación entre objetos.

2.3.10. Enactividad o enacción

Derivado del verbo en ingles “*to enact*”, considera obtener conocimiento a partir de “aprender haciendo” donde un objeto o persona al asumir un papel o rol, adquiere el conocimiento sobre este, por ejemplo: el hecho de aprender a montar bicicleta, cuando alguien se sube en la misma⁵. Se puede describir como el conocimiento que se adquiere a través de la acción de un organismo en el mundo, su propia experiencia. Este concepto es muy aplicado en biología, pero bien podría aplicarse a la robótica y otros campos como la IoT para obtener una verdadera autonomía⁶, pero para esto los recursos deberán tener capacidades como el poder tener una experiencia y a partir de esta inferir información que les permita asumir un comportamiento o rol, actualmente la tecnología presenta aproximaciones a esta característica como por ejemplo las interfaces multimodales.

2.4. CARACTERÍSTICAS DEL DESARROLLO DE APLICACIONES AR EN LA IOT.

Las soluciones en interfaces de AR, aún se encuentran limitadas a escenarios simples con visualización simple de objetos generados por computadora [37], esto no representa un reto metodológico. Cuando las personas con un nivel bajo de conocimiento desean generar aplicaciones AR sencillas (que dibujan objetos virtuales en el mundo real), existen diferentes tipos de software similares a herramientas de autoría, los cuales son amigables con el usuario, inclusive existen algunas para dispositivos móviles intuitivas y fáciles de utilizar [34] [38]. Entre las características identificadas son:

- **Tecnologías AR e IoT combinadas:** Cuando se desea llegar a un nivel de complejidad un poco más elevado como el que exige la IoT, se involucran aspectos como cierta inteligencia, *context aware*, protocolos, formatos, gestión de datos (ontologías, bases de datos, sistemas de información o de recomendación) e interacción con el usuario a un nivel más alto e intuitivo de HCI, existen opciones como la VR, sin embargo estas cortan la comunicación entre el mundo real, los usuarios y lo virtual o sus representaciones [31] y tampoco existe una metodología adecuada a la generación de soluciones en AR con esta complejidad, esto en parte conduce a que el proceso de desarrollo sea tedioso y requiera de un experto [34], [37], [38]; en general los desarrolladores toman las metodologías existentes y las adaptan, o establecen fases y lógica guiados por el entorno de desarrollo de forma enactiva.

⁵ <http://web.archive.org/web/20150920120650/http://www.enactivenetwork.org/index.php?9/enactive-interfaces>

⁶ http://www.scielo.cl/scielo.php?pid=S0716-97602001000200002&script=sci_arttext



- **Recursos:** a medida que suben las expectativas del usuario en el campo de la IoT combinada con AR, surgen necesidades a nivel de comunicaciones, procesamiento, memoria, gráficos de mayor calidad, fusión de datos de sensores, contextualización, personalización, ergonomía, calidad en el seguimiento *Tracking* y otras características que aumentan la exigencia [34], [37][31], traducándose en gestión adecuada de los recursos disponibles.
- **Hardware:** La compatibilidad entre los dispositivos de nueva generación como son los visores de VR, visores de AR indirecta, las gafas inteligentes (*Smart Glasses*), los dispositivos montados en la cabeza HMD (*Head Mounted Display*), y en general dispositivos vestibles (*Wearables*) orientados al sentido de la vista (*Eyewear*) crean una alta competitividad y a su vez una contienda en el mercado, lo que traduce en la necesidad de tener métodos de desarrollo muy flexibles, tanto en tiempo como recursos y la adecuada escogencia de la tecnología a utilizar o tener la posibilidad de tener desarrollos genéricos que se adapten a cualquier tipo de tecnología [22].
- **Metodología:** Desde un punto de vista práctico, el modelo a utilizar debe ser ligero, flexible y de rápido desarrollo, con tolerancia a riesgos, gestión del cambio, que maneje la incertidumbre que puedan generar los requerimientos del cliente, involucrándolo a lo largo de todo el proceso de desarrollo, debe adaptarse al hecho de que en la mayoría de tecnologías utilizadas para AR no existen roles en los usuarios, ya que la experiencia del mundo real está expuesta hacia una sola persona, y es la que hace uso del dispositivo personal, el cual le permite la sensación del entorno enriquecido.

También se debe tener en cuenta que ciertos eventos que se dan entre el usuario y el entorno son intrínsecos, por ejemplo la acción del desplazamiento y aunque existen unos aspectos facilitan el desarrollo, otros deben manejarse con mayor cuidado y control. La gestión de recursos es esencial, sobre todo el recurso del tiempo, la rapidez de desarrollo es necesaria ya que para realizar contenidos funcionales que sean competitivos a nivel de negocio, estos deben crearse rápidamente debido a la naturaleza del mercado actual [22].

- **Calidad:** aunque el desarrollador asume los roles que tienen que ver con el diseño gráfico y el desempeño de la aplicación en la medida de sus posibilidades y las herramientas que domine o posea, estos aspectos no son su fortaleza y el resultado obtenido en AR puede tener un nivel de calidad aceptable, pero no optimizado, además que los tiempos de desempeño se pueden ver afectados por la alta demanda de recursos que exige un despliegue gráfico de este tipo, o por algún otro aspecto tecnológico que implique un riesgo a lo largo del desarrollo.

La creación de entretenimiento junto con la gamificación⁷ del entorno o el desarrollo de juegos serios es un aspecto que está tratado bajo enfoques orientados a la creación de videojuegos, así mismo el diseño de interfaces tangibles de usuario TUI (*Tangible User Interfaces*) tiene otro enfoque diferente al que se aborda en este estudio, sin embargo, algunos aspectos aquí tratados pueden ser comunes y útiles para su diseño.

⁷ También conocido como ludificación, es el uso de mecánicas de juego en entornos ajenos al juego [66].



2.5. METODOLOGIA PARA LA CREACION DEL MARCO DE REFERENCIA.

El modelo de investigación documental MID propuesto por Hoyos-Serrano [39] [40] propone unas fases de las cuales se tomaron algunos elementos que hacen referencia a la documentación de las unidades de análisis como son libros, artículos entre otros, obteniendo una construcción teórica global que da paso al marco de referencia obtenido.

Así se exploraron las unidades de análisis bajo 3 temáticas fundamentales: Realidad aumentada, Internet de las cosas y diseño de interfaces, lo cual permitió identificar los lineamientos del marco de referencia acerca de la exploración de metodologías de desarrollo, las tecnologías involucradas y la interacción con el usuario como lo indica la

Figura 3. Temáticas que definen los lineamientos del marco de referencia.

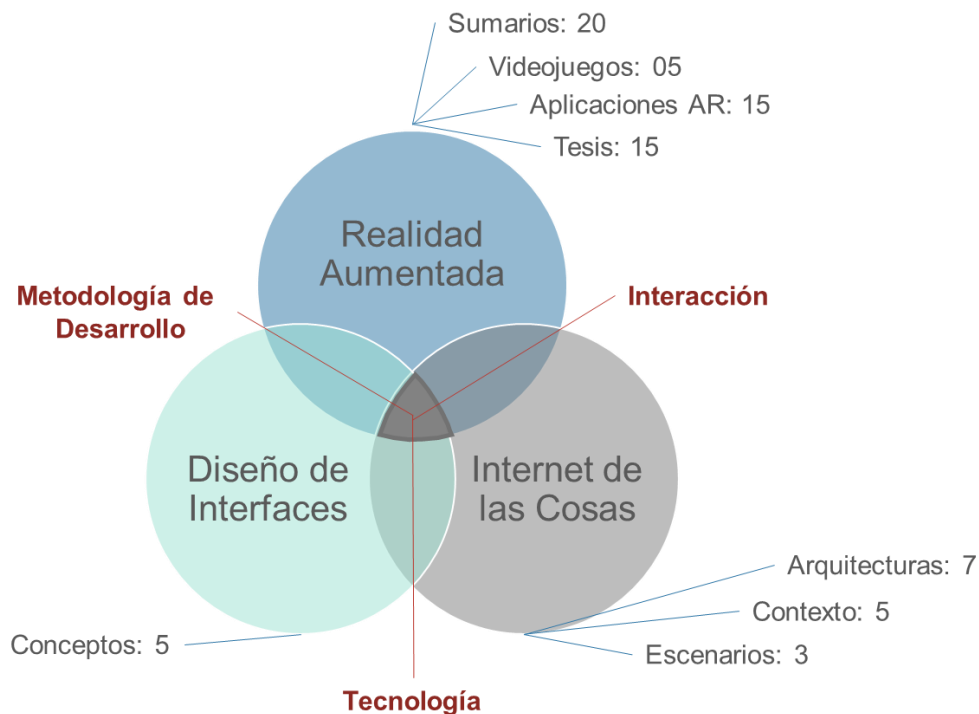


Figura 3. Temáticas que definen los lineamientos del marco de referencia.

2.6. ANALISIS DE MODELOS Y METODOLOGIAS.

Dependiendo del tipo de desarrollo y el equipo de trabajo disponible, se deben orientar los modelos y metodologías a analizar sobre unas características específicas. El presente trabajo recorre y muestra estas características, enfocándose en menor medida sobre las referencias orientadas al desarrollo de juegos en AR y haciendo más énfasis en la experiencia de usuario para cumplir con los objetivos específicos sobre una interfaz natural y eficiente, al enriquecer el escenario donde se despliegan recursos IoT, con alta participación del cliente en las etapas de desarrollo. También es importante tener en cuenta la escogencia del material bibliográfico, ya que habrá propuestas metodológicas obsoletas o que su desarrollo actualmente no se puede replicar,



debido a la poca documentación, o por utilizar herramientas que antes fueron de acceso público, pero que en la actualidad son de acceso parcial o nulo.

2.6.1. Modelos de desarrollo de software vs características de Interfaces en AR.

Las características de desarrollo de este tipo de interfaces generan necesidades en todo aspecto, en particular el aspecto metodológico sobresalen la gestión de recursos y la rapidez de desarrollo. Con el fin de establecer que modelos de desarrollo de software son los más adecuados para crear una aplicación en AR orientada hacia la IoT, se realizó un estudio comparativo, en el cual se observan las características del modelo analizado que están relacionadas al tipo de solución que se desea obtener, estas son:

- **Tolerancia a riesgos:** Es la capacidad del gestionar los recursos adecuadamente cuando suceden eventos que se salen del control del desarrollador o equipo de desarrollo.
- **Manejo de la incertidumbre:** Característica que permite que un desarrollador o grupo de desarrollo aborde el planteamiento de la solución sin tener certeza de su funcionamiento y de la interacción del usuario con la máquina.
- **Priorización de requerimientos y gestión del cambio:** Característica por la cual se le da mayor relevancia y se aborda primeramente las funcionalidades más importantes de un requerimiento, y se crean estrategias para abordar los cambios que se puedan dar en los requerimientos a lo largo del proceso de desarrollo.
- **Tipo de proceso:** Lineal (L), Evolutivo (E) o Paralelo (P).

La Tabla 1 muestra los resultados de la comparativa realizada. Cabe notar que este análisis es adecuado solamente al enfoque AR, y otros análisis orientados a otro tipo de software puede que no sean similares.

Modelo	Tolerancia a riesgos	Manejo de la incertidumbre	Priorización de requerimientos	Tipo de proceso
Cascada				L
Incremental			✓	L,E,P
Prototipos	✓	✓	✓	E
Espiral				L,E
Concurrente		✓		P
Etapas		✓	✓	E,P
RAD				E,P

Tabla 1: Comparativa entre modelos de desarrollo clásicos.

Se observa que la mayoría de modelos son poco flexibles, pues desde un principio definen rigurosamente su estructura, y son basados en el modelo de cascada, pero con ciertas modificaciones, lo cual en esencia los hace similares.

Existen modelos más disciplinados como el proceso unificado, o el modelo basado en componentes los cuales se orientan a un enfoque especializado o muy específico, y aunque



garantizan alta calidad, son demasiado rigurosos y poco flexibles con un cronograma y un presupuesto asignado previamente, lo cual va en contra de la filosofía que se desea seguir.

Dentro de los modelos clásicos sobresale el modelo incremental, ya que es muy utilizado en el desarrollo de videojuegos, aquí se priorizan las interfaces y en cada incremento se van desarrollando otras funcionalidades, sin embargo, para AR en la IoT seguir este modelo no es conveniente al no priorizar los requerimientos.

Se puede concluir que el modelo más cercano al desarrollo de interfaces en AR en la IoT es el modelo de prototipos, ya que cumple con las características deseables para un modelo de software, que gestione el riesgo y sea flexible, este modelo se describe en el Anexo A del presente documento.

2.6.2. Metodologías de requerimientos.

El proceso de definir los requerimientos es una base fundamental al momento de desarrollar una solución software [41] y es la inestabilidad o el cambio en estos, un aspecto que dificulta el desarrollo de dicha solución[42], en AR es básico la gestión de estos cambios, para esto es fundamental escoger una adecuada metodología de requerimientos.

Existen 4 modelos de requerimientos definidos, estos son: modelos basados en escenario, modelos de clase, modelos de comportamiento, y modelos de flujo [22], debido a la necesidad de interactuar con el cliente a lo largo del proceso de desarrollo, los modelos basados en escenarios serían la manera más adecuada de abordar requerimientos para el desarrollo de aplicaciones de AR en la IoT.

Para un escenario definido, los métodos que más se ajustan a lo requerido son: los casos de uso, las historias de usuario y algunos elementos del método de descripción de tareas, estas metodologías se encuentran descritas en el Anexo A de este documento, aunque existen otras metodologías de requerimientos que están orientadas a grandes grupos de trabajo, con estándares empresariales previamente definidos [42], para este caso, no son muy útiles por su costo en tiempo y complejidad. También están las metodologías de desarrollo que incluyen la especificación de requerimientos como un proceso interno [42].

2.6.3. Metodologías de desarrollo.

- **Metodologías Ágiles.**

Las aplicaciones en AR se encuentran en permanente evolución y volatilidad; la generación de contenidos en esta área no cuenta con metodologías orientadas específicamente a su desarrollo, sino que se adaptan las existentes para brindar soluciones de este tipo.

La filosofía de las metodologías ágiles es de proceso iterativo e incremental y sus valores se resumen en el manifiesto ágil donde se establecen unos de valores que a su vez derivan en una serie de prácticas [22], [23], [42]. Para un desarrollo en AR estos valores brindan un marco de referencia importante para un buen desarrollo y se alinean las necesidades del cliente con los objetivos del negocio o empresa, tanto en eficacia como eficiencia, incluso gestionando el



cambio y manejando riesgos, esta es precisamente la característica que define el aspecto de agilidad [22][42].

Se observa que, aunque cada metodología ágil tiene sus propios parámetros y características, tienen en común un proceso de 4 pasos (planeamiento, obtención de requerimientos priorizados, iteración y puesta en marcha) además de definir pocos roles en el grupo de trabajo, pero con tareas concretas.

La metodología ágil más utilizada se llama Programación Extrema (XP), y las más representativas son [22]:

- Desarrollo adaptativo de software (DAS).
- Scrum.
- Método de desarrollo de sistemas dinámicos (MDS).
- Crystal.
- Desarrollo impulsado por las características (DIC).
- Desarrollo esbelto de software (DES).
- Modelado ágil (MA).
- Proceso unificado ágil (PUA).

Analizando las metodologías ágiles más representativas que pueden ser utilizadas para la generación de interfaces en AR para la IoT, se puede obtener elementos altamente valiosos para obtener buenos resultados en el diseño de un método de desarrollo para proyectos de AR, sin embargo, las que en su estructura y modo de trabajo podrían aportar más elementos a este proyecto son Scrum y XP [42], las cuales proporcionan procesos y prácticas específicas para la gestión de un proyecto, cubriendo las características que debe tener uno en AR para la IoT como se observa en la Tabla 2.

Si estas metodologías se observan como marcos de trabajo, se pueden escalar a métodos de gestión de proyectos más formales, combinándose con otras metodologías para cubrir el ciclo de vida del software en alguno de los modelos especificados anteriormente [42]. A continuación, se extraen aspectos de estas dos metodologías como referencia para la creación de interfaces en AR para la IoT.

	Scrum	XTREME PROGRAMING (XP)
Valores	Disciplina y organización	Comunicación, valentía y respeto por el proceso.
Filosofía	Revisar y Adaptar	Orientado a objetos, uso de metáforas.
Fases	Scrum's y Sprint's o Bloques cortos de tareas en tiempo fijo (Descubrir, Diseñar, Desarrollar y probar)	Planeación, diseño codificación y pruebas
Requerimientos	Historias de usuario, requerimientos priorizados.	Historias de usuario, casos de uso.
Cliente	Presente a lo largo del desarrollo.	El cliente debe estar siempre disponible.
Roles	Scrum master, propietario, equipo, cliente, gestor y usuario	programadores, consultores, clientes, consejeros, rastreadores, jefes
Entregas	Reunión diaria, entrega mensual.	Iteraciones de 1 a 3 semanas por fase, e iteraciones de cada fase máximo 4.

Tabla 2: Elementos de las metodologías Scrum y XP.



Respecto a Scrum los bloques cortos de tareas permiten manejar la incertidumbre, el trabajo de la mano con el cliente permite gestionar el cambio, Scrum está indicado para entornos complejos con requerimientos cambiantes o poco definidos, brinda resultados en corto tiempo[42]. Este modo de trabajar exige un equipo homogéneo y disciplinado, además que los tiempos que maneja el cliente deben estar a la par con los del equipo de desarrollo, es un método que pone al equipo bajo presión y puede pasar por alto detalles que si bien no son relevantes para cierto tipo de aplicaciones, en AR si pueden ser críticos [22], [23].

Respecto a XP [22], [23], para un solo desarrollador o un grupo pequeño de desarrolladores, propone roles que aunque son pocos, aún siguen siendo difíciles de asumir con rigurosidad, y como alternativa los desarrolladores se turnan o asumen más de un rol. Existe una variante llamada XP industrial pero está orientada a grandes grupos de trabajo y no se entrara a analizarla en este documento.

- **Metodologías para un solo desarrollador.**

Es un fenómeno creciente en el cual las aplicaciones para dispositivos móviles o de escritorio, son desarrolladas por un único programador, una pareja o un grupo pequeño de diseño, incluso con personas que no son expertas, esto debido a la gran competencia que existe en el mercado creciente tanto en software como hardware [23] y la fácil accesibilidad que presentan las herramientas de desarrollo junto con el acceso a la información a través internet. Cuando no existe un equipo de tipo empresarial o de gran escala, existen métodos sugeridos para abordar un problema de desarrollo de software en solitario, ya que es diferente a la típica asignación de roles en un grupo de trabajo.

El método *Cowboy* [43] integra las características del personaje típico del oeste de Estados Unidos de América, infiriéndole al programador características como el ser imprudente, agresivo e independiente, refiere un estilo llamado *Cowboy Coding* el cual sirve para trabajo individual pero en equipo genera incertidumbre lo cual es indeseable, la filosofía de este método se resume en: “*Se hace en cada momento lo que se cree, se debe hacer*”, así se convierte en 100% adaptativo, sin una metodología definida, y muy diferente de una metodología ágil. Puede representar una buena alternativa para proyectos en AR, pero a medida que la complejidad del desarrollo aumenta o haya cambios en los requerimientos, los tiempos de desarrollo pueden extenderse o llegar a un punto donde el proyecto sea insostenible.

El proceso personal del software (PPS) [22], [23]hace énfasis en registrar y analizar prontamente los errores propios, y desarrollar estrategias para eliminarlos, es un enfoque disciplinado para no caer en un desarrollo caprichoso, partiendo de 4 fases personales: Planeación, Diseño y revisión de alto nivel, Desarrollo y Post Mortem lo cual genera eficiencia a la hora de crear soluciones. Desafortunadamente es un choque cultural que muy pocos desarrolladores están dispuestos a adoptar.

El modelo de desarrollo Ágil Solo [23] tiene como valor el respeto de los valores de las metodologías ágiles, se fundamenta en priorizar requerimientos, realizar presentaciones semanales y entregas para pruebas con el cliente mes a mes para obtener realimentación. Las tareas se les asignan un tiempo específico durante la planeación de las iteraciones semanales, además de tener una visión global de los requerimientos y probar funcionalidades con el



cliente en cada entrega. Toma muchos elementos de cada metodología ágil, pero es sobre todo para fomentar la disciplina y la motivación en el desarrollador en solitario aplicando técnicas como Pomodoro⁸ y Montessori⁹.

Scrum en solitario¹⁰ plantea la filosofía de Equipo de un solo hombre (*One Man Team*), es un descubrimiento personal donde se retoman aspectos de Scrum y se adaptan a las condiciones dadas de acuerdo a los elementos que brinda la metodología, bajo los valores de disciplina y auto organización.

Shelley Doll¹¹ propone un estilo de programación ágil para el desarrollador solitario, basándose en un estilo iterativo fundamentado en la metodología XP y sus 4 actividades, con entregas funcionales, a tiempo, priorizando requerimientos sin perder de vista el objetivo general. Sugiere utilizar historias de usuario y fichas de clases para mantener la metáfora del programador, y “probar antes de pasar” para comprobar que la solución en verdad cumple la funcionalidad requerida.

2.6.4. Conclusiones finales acerca de los modelos y metodologías analizadas.

- Como elementos fundamentales para la creación de interfaces en AR se observa como elemento importante a incorporar lo siguiente: la gestión de recursos incluido el tiempo, la tolerancia a riesgos, la gestión del cambio, el uso de artefactos como las historias de usuario, la correcta definición y delimitación de los alcances del proyecto y la necesidad de documentar.
- La metodología XP propone pocos roles, pero difíciles de asumir en un grupo pequeño de desarrolladores, lo cual se convierte en una desventaja para el desarrollo de aplicaciones en AR, pues implica poca flexibilidad.
- Los elementos de XP como entregas cortas, requerimientos priorizados, y contacto con el cliente en todo momento son ventajas útiles a la hora de desarrollar interfaces en AR.
- Se sugiere utilizar los artefactos y técnicas de Scrum respecto a la gestión de riesgos para la implementación de aplicaciones en AR.
- Los valores fundamentales de Scrum como la disciplina, la auto organización, la documentación y el contacto con el cliente son elementos útiles para la creación de interfaces en AR.
- Aunque en las metodologías ágiles la documentación no es prioritaria, para el caso de los desarrollos en AR si se aconseja que lo sea.
- La mayoría de elementos útiles encontrados en el análisis de estas metodologías coinciden con la filosofía del modelo de prototipos, es así que este modelo se reafirma como base para crear una guía de desarrollo de interfaces en AR orientadas a un dominio específico.
- Las iteraciones pueden tener cierta flexibilidad, pero como tiempo promedio es aceptable una semana, y la generación de prototipos cada mes. Por lo tanto se aconseja que se manejen tiempos similares para desarrollos en AR.

⁸ <http://pomodorotechnique.com/>

⁹ <http://ami-global.org/>

¹⁰ <http://tracks.roojoom.com/r/7349#/end>

¹¹ <http://www.techrepublic.com/article/agile-programming-works-for-the-solo-developer/1046511/>



- Los requerimientos bajo descripción de tareas, brinda elementos como son los eventos detonantes, útiles a la hora de abordar desarrollos en AR para la IoT.

2.7. TRABAJOS RELACIONADOS.

Para realizar el análisis, se encontraron características que permitieron agrupar los trabajos a nivel de aplicaciones, metodologías y elementos de diseño, también se exploraron tecnologías en el mercado que permiten describir con mayor facilidad los elementos de diseño. A continuación, se analizan los trabajos encontrados que definen aplicaciones en AR y los métodos utilizados para su desarrollo. Luego se determinan elementos de importancia hallados en los artículos relacionados para el diseño de interfaces en AR para la IoT.

2.7.1. Exploración de tecnologías y metodologías en los trabajos relacionados.

Se realizó una exploración de aproximadamente 50 unidades de análisis, donde 18 trabajos relacionados mostraban de forma explícita la metodología usada, y a su vez las herramientas software utilizadas obteniendo los siguientes resultados (ver Figura 4).

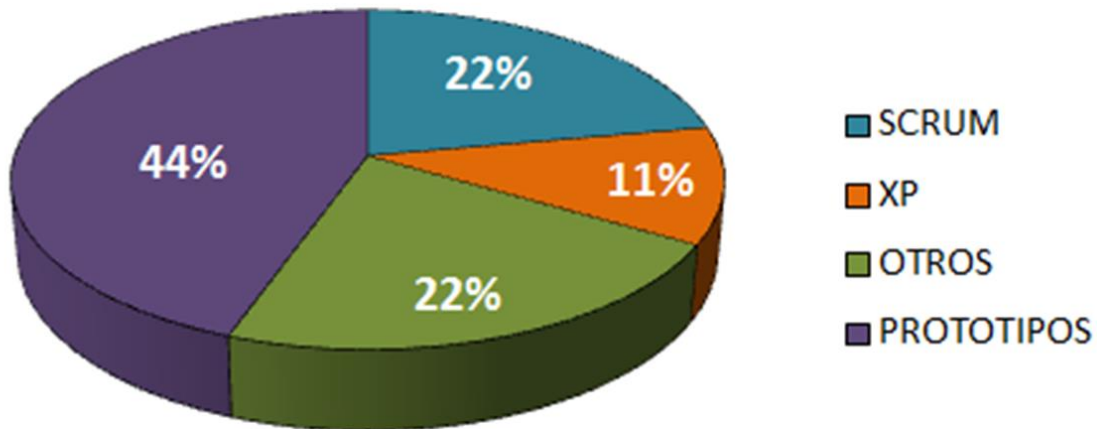


Figura 4. Metodologías de desarrollo utilizadas por los trabajos relacionados.

La tecnología utilizada por el 60% de los trabajos fue un conjunto híbrido de herramientas compatibles entre sí, las cuales fueron: El motor de desarrollo de videojuegos Unity 3D, junto con la librería de desarrollo de Realidad aumentada de PTC Vuforia y como sistema operativo objetivo Android.



La Tabla 3 agrupa los trabajos relacionados respecto los enfoques metodológicos de desarrollo utilizados, junto con las tecnologías en AR que fueron implementadas.

Trabajos relacionados.	Scrum	XP	Prototipos	Escenas	Otros	Tecnología de AR Utilizada
G. Llanos [38], "Guía de AR para publicidad"	X					Flartoolkit
D. Rosero [44], "AR para personas con deficiencia mental leve"	X					Android, Beyond AR
V. Caballero [45], "Aplicación para Turismo en Lima"	X					Android, Artoolworks, Open CV.
T. Marshall [46], "Moving museum outside its walls"	X					Android, Unity 3D, Vuforia SDK
C. Lodts [24], "Tecnic for AR inside OWG"			X	X		Unity 3D, OpenCV, Oculus SDK
J. Lopez [47], "Videojuego multijugador basado en AR"			X	X		Android, Unity 3D, Vuforia SDK
O. Sedano [48], "Estudio y desarrollo de una aplicación AR"			X	X		Android, Unity 3D, Vuforia SDK
H. Reuterdaahl [49], "Mobile AR Intuitive Instruction Manual,"			X	X		Android, iOS, Unity 3D, Vuforia SDK
A. Palma [50], "Aplicación en AR para enseñanza"			X	X		Unity 3D, Vuforia SDK
D. Seo [51], "Hybrid Reality"			X	X		Varios dispositivos, Leap Motion, Oculus rift DK2
L. Garduno [52], "Gallag Strip"			X	X		C# Multiplataforma
M. Tatzgern [36], "Transitional AR navigation"				X		KinectFusion, Pointcloud libray PCL
T. Chou [31], "AR Smartphone environment orientation App"					X	Layar
J. Moreno [53], "Maqueta arquitectónica interactiva"				X		Android, Unity 3D, Vuforia SDK
S. Uceda [54], "AR para lectura e interpretación de planos"				X		Vuforia, Metaio, ARmedia
D. Molina [55], "AR sistema de ayuda a alumnos"				X		Vuforia para iOS
P. Ochoa [56], "Sistema de información para el ejército en AR"		X				iOS, AR basada en Geolocalización
C. Alfaro [57], "Interfaz y módulo de entrenamiento ARpool"		X				ARPool, Open CV, Open GL

Tabla 3. Relación entre aplicaciones en AR, las metodologías de desarrollo y las tecnologías utilizadas.

Se observa bastante variedad en cómo se abordan soluciones de este tipo predominando el uso de prototipos y escenas, esto se debe a que el entorno de desarrollo escogido no es un entorno especializado, sino que se requiere adaptarlo y jugar con sus funcionalidades, por ejemplo el uso de un motor de videojuegos condiciona el desarrollo al uso de escenas, en este caso se puede concluir que el motor de videojuegos Unity3D¹² junto con el kit de desarrollo SDK (*Software Development Kit*) PTC Vuforia¹³ son los más utilizados para el desarrollo de aplicaciones en AR. También cabe destacar la asignación de escenas para el desarrollo de cada funcionalidad o grupo de funcionalidades comunes a través de Unity 3D.

¹² <https://unity3d.com/es>

¹³ <https://www.vuforia.com/>



Otros desarrolladores prefieren apearse al uso de metodologías ágiles, como un patrón de organización y formalidad, por esto XP y Scrum se posicionan como metodologías adaptables y flexibles de las cuales se pueden extraer elementos para el proceso de desarrollo de interfaces de AR en la IoT.

2.7.2. Trabajos relevantes.

Para abordar un desarrollo en AR, orientado hacia algún dominio en específico como lo es la IoT, de forma general se pueden establecer ciertos elementos que permiten agrupar los trabajos más relevantes y que hacen notables aportes hacia este tipo de soluciones. A continuación se agrupan y se nombran de manera general dichos trabajos, más adelante a lo largo de este trabajo y en especial en el apartado de diseño se establecen como tal los aportes de cada cual, ya que es complejo definir de manera precisa sus aportes en una sola área.

Figura 5 permite identificar los núcleos temáticos encontrados en los trabajos más relevantes, que definen los elementos de diseño para interfaces AR en la IoT, donde se agrupan los trabajos en estos núcleos de manera cronológica.

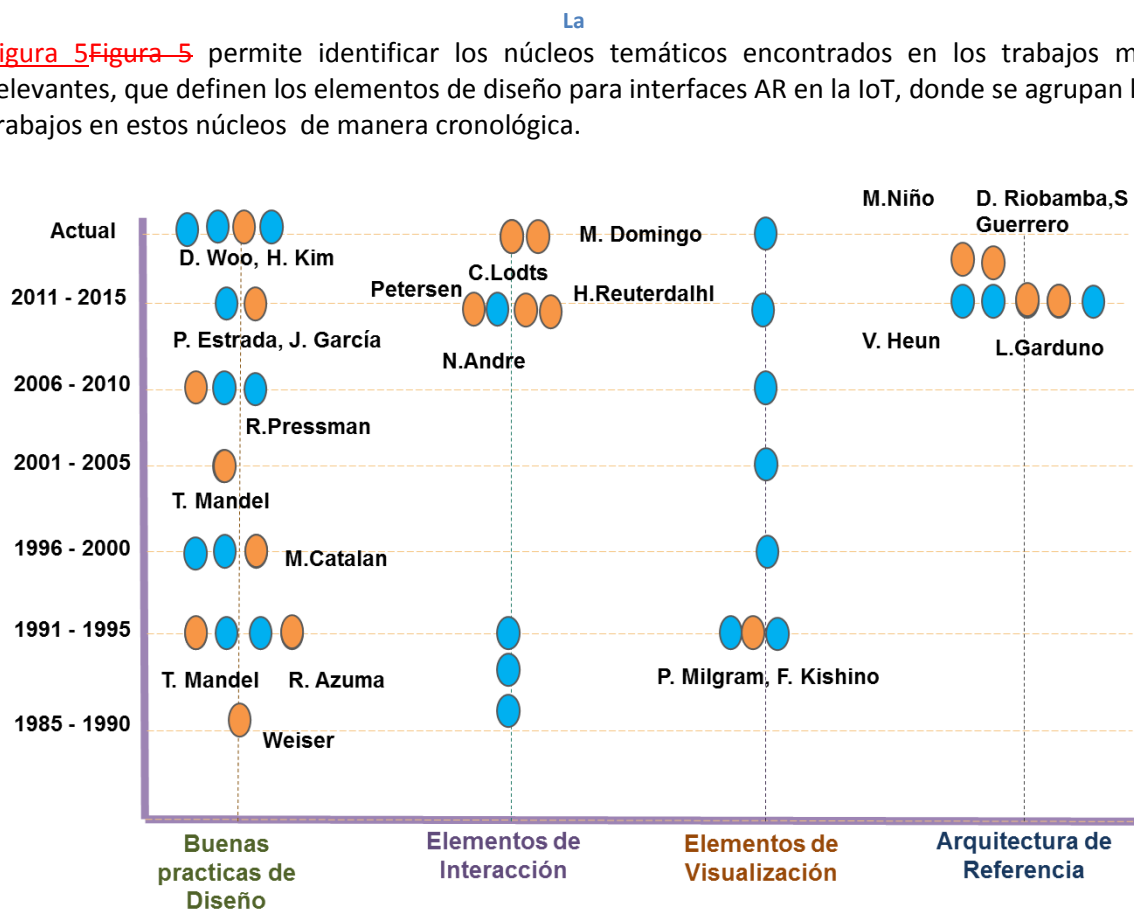


Figura 5. Identificación de elementos de diseño en los trabajos relevantes.

- **Buenas prácticas de diseño.**

Como trabajos que hacen importantes aportes para generar unas buenas prácticas en la creación de interfaces se tiene el trabajo de Theo Mandel [28] con su trabajo titulado *User / System Interface Design* hace una descripción detallada acerca del proceso necesario para



desarrollar una interfaz de usuario acorde a las necesidades ergonómicas y de HCI, que se requieren para una buena aceptación y una alta calidad, basándose en tres principios básicos: Dar el control de la interfaz al usuario, no llenarlo de asuntos por memorizar y crear una interfaz que siga un flujo natural, es decir que sea consecuente.

Dong Woo [51] define un sistema de evaluación de un entorno inteligente utilizando AR y VR de forma conjunta, su análisis se basa en que las aplicaciones de este tipo sirven para evaluar un entorno inteligente, en este caso una casa, de forma más económica y flexible que hacerlo de forma física. La sincronización entre componentes virtuales y físicos se hace a través de una ontología, generando contextos nuevos a partir de los existentes, este análisis del contexto como elemento de diseño es un aporte que influye en varios aspectos para una solución en AR.

Estrada y García [4] en su trabajo *Semantic interactions in the Internet of Things* presentan varios escenarios inteligentes, haciendo énfasis en que la presencia de OI y sus interacciones representan un nuevo nivel de conectividad donde el concepto de IoT se puede extender a cualquier objeto que se pueda imaginar y es necesario tener modos de interacción adecuados en cada caso. Proponen 3 paradigmas de la IoT: orientados a las cosas, al internet y a la semántica, se destaca el paradigma orientado hacia la semántica como el proceso de investigación sobre la representación, el almacenamiento, la búsqueda y la organización de la información generada por los recursos IoT en espacios o escenarios inteligentes.

Los autores brindan dos aplicaciones: la primera es ubiSOA (Arquitectura ubicua orientada a servicios), que es una plataforma para crear escenarios inteligentes en la IoT con servicios web tipo RESTful, y la segunda aplicación presenta un visor sensible (*sentient visor*) para interactuar en espacios inteligentes, los autores introducen el concepto de navegador para la IoT, donde un solo dispositivo puede obtener información de diferentes proveedores de servicio, semejante a una aplicación en AR. Combinando las dos aplicaciones se obtiene la aplicación llamada UbiVisor para la adquisición contextual de datos desde sensores como RFID y servicios como la localización en interiores *indoor* entre otros. Estos aspectos cobran relevancia al diseñar aplicaciones en AR para la IoT.

- **Elementos Hardware y Software para la visualización de la interfaz en AR.**

Milgram y Kishino [32] hacen un estudio taxonómico para diferenciar los aspectos reales y virtuales de un ambiente enriquecido, moviéndose sobre el centro de la recta definida como continuo de la virtualidad *virtuality continuum*, teniendo en cuenta si se manipulan objetos o imágenes reales o virtuales, si se visualizan de manera directa o indirecta logrando así una taxonomía bajo tres dimensiones: extensión del mundo conocido, fidelidad en la reproducción, y metáfora de la presencia. Desde estas interfaces parten los desarrollos para entornos aumentados, de ahí que el término de MR no sea tan utilizado pero muy imparcial, este se extiende hacia el término de AR a la izquierda del *virtuality continuum* y hacia virtualidad aumentada a su derecha y entre esos términos se traslapan los tipos de interfaces ya definidas, por esta razón también se puede utilizar el término de HR indistintamente. La relevancia del artículo se focaliza en la definición del tipo de interfaces visuales en MR.

- **Elementos de Interacción en AR.**

El Trabajo de Nelson J. Andre [34] propone una forma de crear aplicaciones en AR de fácil desarrollo, intuitivas y de calidad , estableciendo que las técnicas de seguimiento híbridas brindan mayor robustez a la aplicación, Andre propone un marco de trabajo (*framework*) de navegación en realidad Aumentada/Virtual - NAVR, el cual permite el desarrollo de amplia gama de aplicaciones en AR, apoyándose en las librerías y herramientas software existentes, de manera modular y organizada, sirviendo como interfaz entre las plataformas de desarrollo abiertas (*open source*) y las de acceso restringido. Este trabajo se enfoca más en las tecnologías para la creación de AR, que en el proceso de desarrollo en sí mismo, sin embargo, aporta elementos importantes a nivel técnico.

Cedric Lodts [24] propone extraer elementos de los videojuegos que se ambientan en espacios abiertos realistas como Watchdogs¹⁴ de Ubisoft, análogamente para la creación de elementos útiles para la visualización con HMD transparentes o como los llama el autor OHMD- HMD ópticos, de entornos reales enriquecidos con AR; estos elementos se llaman *AR-Helpers* y *Widget-Helpers* como se observa en la Figura 6.



Figura 6. Widget Mini mapa (Derecha), Flechas indicadoras ancladas al entorno (Izquierda).

Fuente: C. Lodts, [24] Exploring Different Visualization Techniques For Augmented Reality Navigation Helpers Inspired By Open World Games.

Los elementos encontrados en los trabajos explorados y los juegos OWG fueron la brújula dinámica y mini mapas para navegación o a modo de radar, mapas grandes sobrepuestos para exploración, visión térmica, alertas visuales y sonoras para orientación, etiquetas para anotación, resaltar en cierto color una entidad de interés El o punto de interés PI, resaltar la ruta o camino hacia un objetivo seleccionado. Mediante un estudio estadístico se concluyó que para los desarrolladores como para los usuarios priman los aspectos de navegación y anotación, ya que los aspectos de exploración y orientación requieren más procesamiento de información para el usuario y requieren más esfuerzo por parte del programador.

El trabajo de Petersen y Stricker [37] tiene como consideraciones para diseñar una interfaz tipo asistente en AR, el *context aware* como base fundamental, el manejo de imágenes en

¹⁴<http://watchdogs.ubisoft.com/watchdogs/es-es/home/>

primer plano como escenas, considerando la oclusión de marcadores o imágenes, propone el uso de planos de relevancia y secuencias de imágenes para entrenamiento a modo de enacción comparativa a través de 4 tipos de superposiciones visuales: procedimiento, enactivas, anotación y evaluación. La manera de validar la tesis propuesta es a través de una herramienta de autoría en AR para generar y realizar procedimientos, es decir la misma interfaz permite guardar una secuencia de imágenes de una tarea, ejecutar la tarea y compararla de forma enactiva, así la funcionalidad de la herramienta proporciona al usuario 3 aspectos: autoría, asistencia y enseñanza, estos aspectos enmarcan el uso de los elementos de asistencia en la interacción de interfaces AR con el usuario.

Hannah Reuterdaahl [49] desarrolla una interfaz eficiente e intuitiva para el uso dispositivos médicos por parte de profesionales de la salud, ya que el uso de manuales de instrucciones es una experiencia incomoda y los médicos y enfermeras prefieren la experiencia práctica. El problema radica en que los elementos instructivos como documentos, animaciones y video se encuentran fuera del contexto del dispositivo, esto hace que el usuario tenga que hacer mayor esfuerzo cognitivo y desvíe su atención tratando de comprender las siglas, numeraciones y haciendo mapeo con las partes del sistema en el dispositivo real. Los prototipos de la interfaz de instrucciones en AR basados en marcadores fueron una máquina de café y un contador Geiger Muller como se observa en la Figura 7.



Figura 7. Manual de instrucciones AR para cafetera.

Fuente: H.Reuterdaahl [49], Mobile Marker-based Augmented Reality as an Intuitive Instruction Manual

Las implicaciones, observaciones y recomendaciones halladas en este trabajo pueden extenderse a desarrollos de interfaces en AR para cualquier campo de las ciencias, adaptando

Interfaz de Realidad Aumentada Para la Internet de las Cosas.

estos elementos a cada necesidad, destacando los conceptos de usabilidad y contextualización.

El trabajo de M. Domingo [3] titulado *An overview of the Internet of Things for people with disabilities* presenta una serie de aspectos referentes a arquitectura y diseño de soluciones IoT desde el punto de vista técnico para personas con limitaciones. Propone una arquitectura para la IoT (ver Figura 8) de 3 capas:

- Percepción: su función es identificar objetos y reunir información tipo *context aware* mediante sensores y actuadores.
- Red: comunicación
- Aplicación: solución IoT para la necesidad específica.

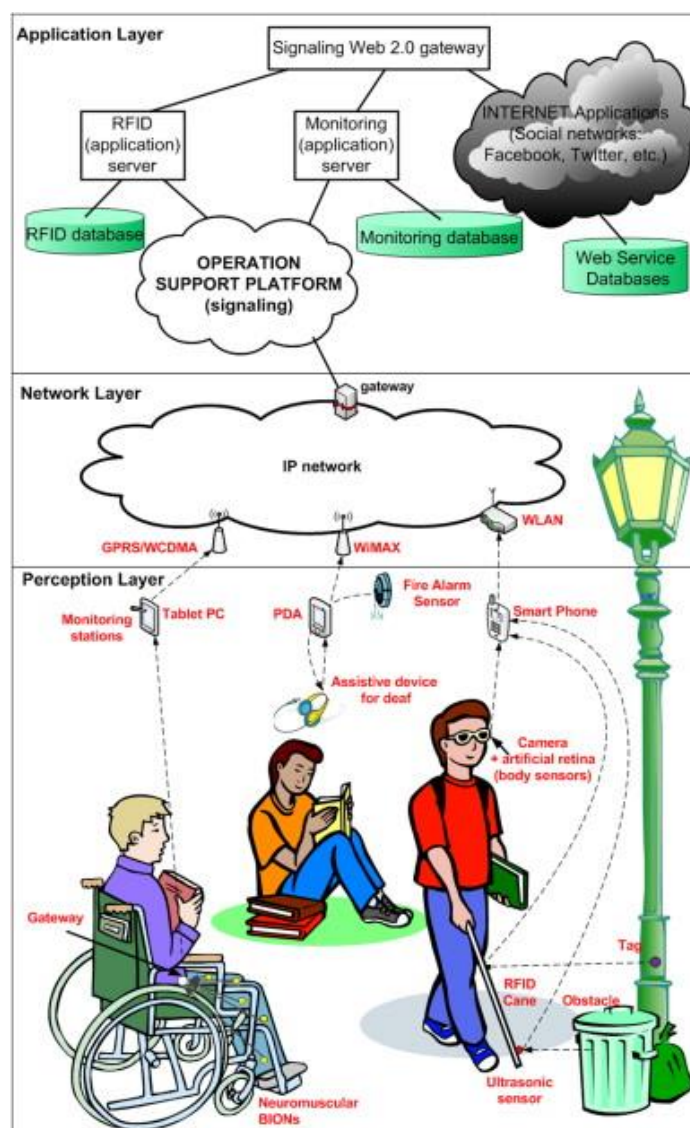


Figura 8. Arquitectura IoT para personas con discapacidad.

Fuente: M. Domingo [3], *An overview of the Internet of Things for people with disabilities*.

También se hacen consideraciones a nivel del tipo de discapacidad, los protocolos de red y su interoperabilidad, y los servicios que se deben desplegar utilizando una arquitectura adecuada para resolver la heterogeneidad.

Se puede observar que las tecnologías mencionadas permiten en conjunto aumentar la percepción de la realidad, en general casi todas van orientadas a convertirse en elementos tecnológicos para generar AR y cualquier persona sin ningún tipo de discapacidad también es un consumidor en potencia de esta integración tecnológica, de ahí la necesidad de explorar este trabajo a profundidad. Las comunicaciones juegan un papel importante a la hora de fijar un margen de usabilidad respecto al tiempo y desempeño, por eso la importancia de escoger, sincronizar y adaptar protocolos de comunicaciones y aplicaciones adecuados desde la IoT hacia la aplicación de AR en cuestión.

- **Referentes para la creación de interfaces AR en un escenario IoT definido.**

Reality Editor [58], define interacciones entre recursos IoT, para esto, realiza un mapeo directo de funcionalidades de objetos reales por medio de AR, utilizando comunicación par a par P2P (*peer to peer*) entre los objetos del escenario propuesto y un servidor que coordina las interacciones básicas que ahí tienen lugar. En la Figura 9 se puede observar como el mapeo directo permite interconectar funcionalidades como las de prender y apagar objetos similares, e incluso de diferente naturaleza, pero con funciones en común, además de permitir guardar las interacciones para un uso posterior o personalizado, este trabajo permite observar elementos para crear interacción AR entre recursos de un escenario IoT.



Figura 9. Interfaz Reality Editor.

Fuente: Reality editor <http://fluid.media.mit.edu/projects/reality-editor-programming-smarter-objects>.

Gallag Strip es una aplicación desarrollada por Luis Garduno Massieu [52], basada en sensores tipo *context - aware* para dispositivo móvil dentro de un entorno demostrativo, para experimentar la realidad como un juego, empoderando a los usuarios a crear su propio contexto, generando un entorno motivador. Este desarrollo se apoya en un API llamada GALLAG, la cual facilita el

Interfaz de Realidad Aumentada Para la Internet de las Cosas.

desarrollo de aplicaciones para un conjunto de sensores y sus servicios debido a su interacción, utilizando programación basada en ejemplos EBP (*Example Based Programming*) para simplificar la captura de acciones que realiza el usuario junto con su respectiva secuencia y la respuesta del sistema, permitiendo que el usuario vea lo que está programando y lo pueda editar fácilmente.

Se establecen servicios como emails, recordatorios, alarmas, notificaciones y mensajes de texto entre otros. La interacción con la aplicación se realiza grabando una actividad y asociándole una respuesta. La interfaz de usuario es programable, personalizable y fluida, se pueden colocar dispositivos en secuencia así como personalizar iconos y acciones a realizar como respuesta al evento o eventos asociados (actividad).

El diseño de la aplicación tiene en cuenta la experiencia del usuario, mediante un enfoque iterativo y participativo hace uso de prototipos, donde se destaca del diseño es la programación interactiva, así como el buen control de flujo de la información que se le presenta al usuario, pudiendo grabar sus actividades, y guardar las secuencias que se programan, además que este pueda disfrutar al hacerlo. La aplicación se diseñó para un solo usuario de tal manera de evitar conflictos, además de fijar horarios y tiempos de uso para liberar los sensores que no se estén utilizando siempre y evitar la concurrencia entre sensores y también entre usuarios.

A continuación, en la Figura 10 se observa la arquitectura de componentes de Gallag Strip.

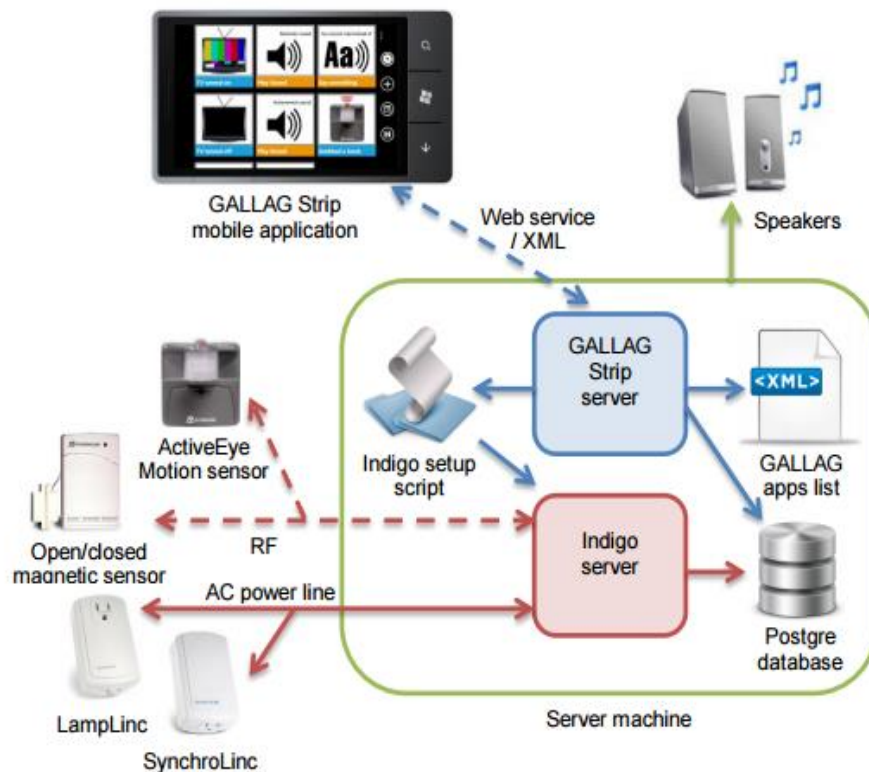


Figura 10. Modelo de componentes del sistema GALLAG Strip.

Fuente: L. Garduno[52], GALLAG Strip: A Mobile, Programming With Demonstration Environment for Sensor-Based Context-Aware Application Programming.



2.7.3. Criterios acerca de la escogencia del software de desarrollo para aplicaciones en AR para la IoT.

Debido que la tecnología software en AR es emergente[5], buena parte de los desarrolladores optan por integrar diversas librerías propias y/o adaptadas a motores gráficos de videojuegos, y de manera personalizada crean kits de desarrollo SDK (*software development kit*) para AR o algunas de sus características según sea el caso [22]; desarrolladores más especializados, por ejemplo a nivel científico generan sus propias herramientas de desarrollo apoyándose en dichas librerías o kits, por estas razones existe una alta heterogeneidad entre los entornos de desarrollo, los diversos lenguajes de programación utilizados y las múltiples plataformas de usuario final [22].

El software disponible se encuentra a nivel de SDK's, conjuntos de herramientas y librerías de tipo de código abierto *open source* como OpenCV¹⁵ y software propietario con acceso libre de modo parcial como Vuforia¹⁶ [34], que son las más utilizadas actualmente, sin embargo, el espectro de recursos disponibles es amplio¹⁷, se concluye que la mayoría de recursos no se popularizan por cerrarse a un solo entorno de desarrollo, un solo sistema operativo o por su limitada capacidad a la hora de integrar elementos de interacción hardware y software, lo cual es crítico en un desarrollo en IoT.

Se ha popularizado el uso de motores de videojuegos para la creación de aplicaciones en AR debido a su eficiencia en tiempo de desarrollo, costos y uso de recursos; el motor más destacado y para el cual las demás herramientas crean interfaces de programación de aplicaciones - API's se llama Unity 3D¹⁸ [34], este motor es versátil por su gran comunidad de soporte, los lenguajes de programación y librerías que soporta, además de un esquema de desarrollo muy cómodo y fácil de utilizar, prácticamente se ha convertido en la herramienta de autoría estándar para aplicaciones de AR tanto de baja como alta complejidad, de uso comercial y científico hasta el momento.

El uso de estos motores brinda la posibilidad de asignar escenarios respectivos a cada conjunto de funcionalidades relacionadas, esto acelera el proceso de desarrollo y permite tener una visión global de la aplicación [24][47][49] además de permitir asignar y manejar cada contexto de manera particular.

Respecto a la captura de eventos a través de imágenes del mundo real se requieren ciertas herramientas matemáticas, procesamiento de imágenes y *frameworks* avanzados como visión por computadora y SLAM entre otros, además de requerir una plataforma hardware adecuada [36], [50].

2.7.4. Hitos y consideraciones en la creación de interfaces de AR para la IoT.

- En la IoT a nivel de sensores, desarrollo y despliegue de servicios la diversidad del hardware hace difícil probar entornos inteligentes reales por costos y flexibilidad, y a nivel de red por lo complejo de las interconexiones entre diferentes recursos IoT. La AR y la VR representan soluciones de bajo costo y flexibles para probar entornos inteligentes en un contexto donde

¹⁵ <http://opencv.org/>

¹⁶ <http://www.vuforia.com/>

¹⁷ <http://socialcompare.com/en/comparison/augmented-reality-sdks>

¹⁸ <https://unity3d.com/es>



coexisten e interactúan OI que generan alta heterogeneidad en aspectos de aplicación, comunicación, servicios y la variedad de datos generados por sí mismos y por las interacciones entre ellos en escenarios inteligentes [4], [51].

- La AR sirve para evaluar la percepción visual del usuario, pero no para simular experiencias totalmente inmersivas [51], sin embargo, combinada con entornos en VR generan la MR, permitiendo brindar otro tipo de interacción más enriquecida a nivel de exploración del entorno, aprendizaje y asistencia [24][51].
- Los widgets y asistentes en AR brindan información en los aspectos de navegación, orientación, exploración y anotación (alerta, recordatorio, comentario, registro), sin embargo algunos factores combinados, son relevantes al momento de escoger estos asistentes, por ejemplo: son diferentes las necesidades y las condiciones de un usuario que camina, a uno que se desplaza en automóvil, o uno que se encuentra buceando [24], pues aquí se enfrentan factores como la velocidad de desplazamiento vs. la atención que se requiere del usuario, generando situaciones en las que se resta o se anula la utilidad de la información brindada por la interfaz.
- El *context-aware* es un factor importante en la creación de entornos inteligentes [51], ya que los OI generan diferentes informaciones dependiendo del contexto en que se encuentran, es importante que la interfaz en AR sea una aplicación con capacidad de contextualizar o ser sensible al contexto, para generar información y servicios en un nivel superior por ejemplo a nivel de situación y habilidad se desarrollaría una AR cognitiva [37]. La interacción con objetos virtuales a través de AR con *context-aware*, debería producirse en tiempo real, la información desplegada debe dar la sensación de interacción natural [51].
- Para manejar el contexto y establecer necesidades a nivel de comunicación e interacción con los elementos del mundo real (medidas, fenómenos, eventos, cosas) que se desean representar u obtener algún tipo de información, es importante tener una arquitectura previa que facilite el desarrollo de aplicaciones de AR en la IoT [3][43], considerando la interoperabilidad, los servicios a prestar, la población objetivo y los roles de cada OI dentro de la aplicación.
- Dentro de dicha arquitectura se pueden incluir maneras de manejar el conocimiento; necesidades como la sincronización entre objetos virtuales y físicos en AR, pueden resolverse utilizando ontologías [51], las interacciones entre OI se pueden ver enriquecidas mediante estas herramientas y utilizando tecnologías web semánticas como índices, modelos semánticos y sistemas de aumentación semántica, así como el uso de estándares que se adecuan a estas tecnologías como por ejemplo RDF (*Resource Description Framework*), OWL (*Ontology Web Language*), SKOS (*Simple Knowledge Organization System*), SQWRL (*Semantic Query-Enhanced Web Rule Language*), SPARQL (*SPARQL Protocol and RDF Query Language*) y JENA¹⁹ entre otros [4][51].
- El aspecto de la heterogeneidad en todas las áreas estudiadas[3], crea cuellos de botella en el flujo de información, además que obstaculiza el correcto desarrollo de la tecnología, la normalización y la convergencia se logra utilizando estándares, más aun cuando la tendencia

¹⁹ <https://jena.apache.org/>



general apunta a formar una red social de objetos y personas, esta tendencia trae consigo retos a nivel de seguridad, privacidad, trafico, desempeño, interacción en tiempo real, calidad, escalabilidad, cooperación entre objetos entre otros.

- Los problemas más destacados en dispositivos médicos fueron encontrados en el trabajo de Hannah Reuterdaahl [36], éstos son escalables a la mayoría de objetos, y el tratamiento de estos problemas puede ser extendidos a los recursos relacionados con la IoT como son los OI. Estos problemas son:
 - Dispositivos difíciles de preparar o configurar.
 - Problemas relacionados con la instalación.
 - Aspectos de programación.
 - Instrucciones de uso.

Estos problemas aumentan la probabilidad de accidentes, lesiones, averías y todo aquello que se deriva de una mala manipulación de los objetos involucrados, materia prima, muestras, artefactos y demás, también genera la necesidad de tener un experto para resolverlos, así las tecnologías en AR pueden disminuir estos inconvenientes mediante la implementación de una interfaz intuitiva y de fácil uso con las siguientes recomendaciones:

- Adaptar la información de acuerdo con el usuario, su experiencia previa y conocimiento.
- Permitirle al usuario realizar las tareas prácticas con el dispositivo mientras recibe instrucciones.
- Eliminar la necesidad de traducir la información (etiquetas poco intuitivas, siglas, numeraciones) entre el dispositivo real y la información que brinda el asistente.

2.7.5. Brechas encontradas en el análisis de trabajos relacionados.

- Muy pocos trabajos tienen en cuenta diferentes perfiles de usuario, y no permiten personalización. Tampoco la creación de nuevos servicios basados en el procesamiento del contexto o servicios inteligentes [51].
- La mayoría de desarrollos no tienen en cuenta aspectos mecánicos como la velocidad de desplazamiento, la aceleración, ni el nivel de luminosidad, o el estado del tiempo, ni las combinaciones de estos factores en el contexto del usuario.
- El posicionamiento en interiores *indoor* y las técnicas de SLAM aún no se tratan frecuentemente, pues en un nivel de baja complejidad no se necesitan, o su uso es transparente al usuario y en aplicaciones más complejas debido a lo impreciso de la tecnología existente poco soporte y no son muy utilizadas.
- Poco se mencionan interfaces hápticas para AR, ni el uso de interfaces que utilizan señales electroencefalograficas - EEG como Emotive²⁰, ni electromiograficas - EMG como MYO²¹ para el desarrollo de aplicaciones en AR.
- Aunque el uso de marcadores poco a poco es reemplazado por otras técnicas híbridas y combinadas con visión por computadora, casi no se mencionan debido a que las plataformas

²⁰ <https://emotiv.com/>

²¹ <https://www.myo.com/>



más populares restringen el uso a un pago, o la tecnología solo está disponible para ciertos dispositivos como Smartphones de alta gama o un hardware especial.

- Solo hasta 2016 los SDK's y los motores de videojuegos más utilizados integraron elementos para el desarrollo de interfaces AR/VR de forma explícita, lo que implica una nueva curva de aprendizaje y la adaptación al nuevo paradigma de desarrollo para aplicaciones de alto nivel en MR.

La necesidad de la interdisciplinariedad o asesoría de un experto en cada dominio del conocimiento donde se desea crear la aumentación es mencionada, pero no se realiza en la mayoría de desarrollos.

2.8. DISEÑO DE INTERFACES EN AR PARA LA IoT.

2.8.1. Naturaleza del software.

Debido a la naturaleza del software es necesario entender para que se desea la solución en AR, es decir cuál es el dominio en el que se basa o en el que actúa, Roger Pressman [22] define los siguientes dominios:

- Software de sistemas
- Software de aplicación
- Software de ingeniería y ciencias
- Software incrustado
- Software de línea de productos
- Aplicaciones web
- Software de inteligencia artificial
- Computación en mundo abierto
- Construcción de redes
- Fuente abierta

El concepto de computación ubicua como dominio de software, actualmente es tan amplio que se diluye fácilmente entre varias categorías, y precisamente esa es su característica, la naturalidad y la transparencia al usuario a través de la colaboración entre diferentes tecnologías. Luego de escoger la naturaleza del software, se tiene el grupo de tecnologías en AR que permiten determinar el aspecto de diseño a nivel de requerimientos y recursos, como se observa en la Figura 11.

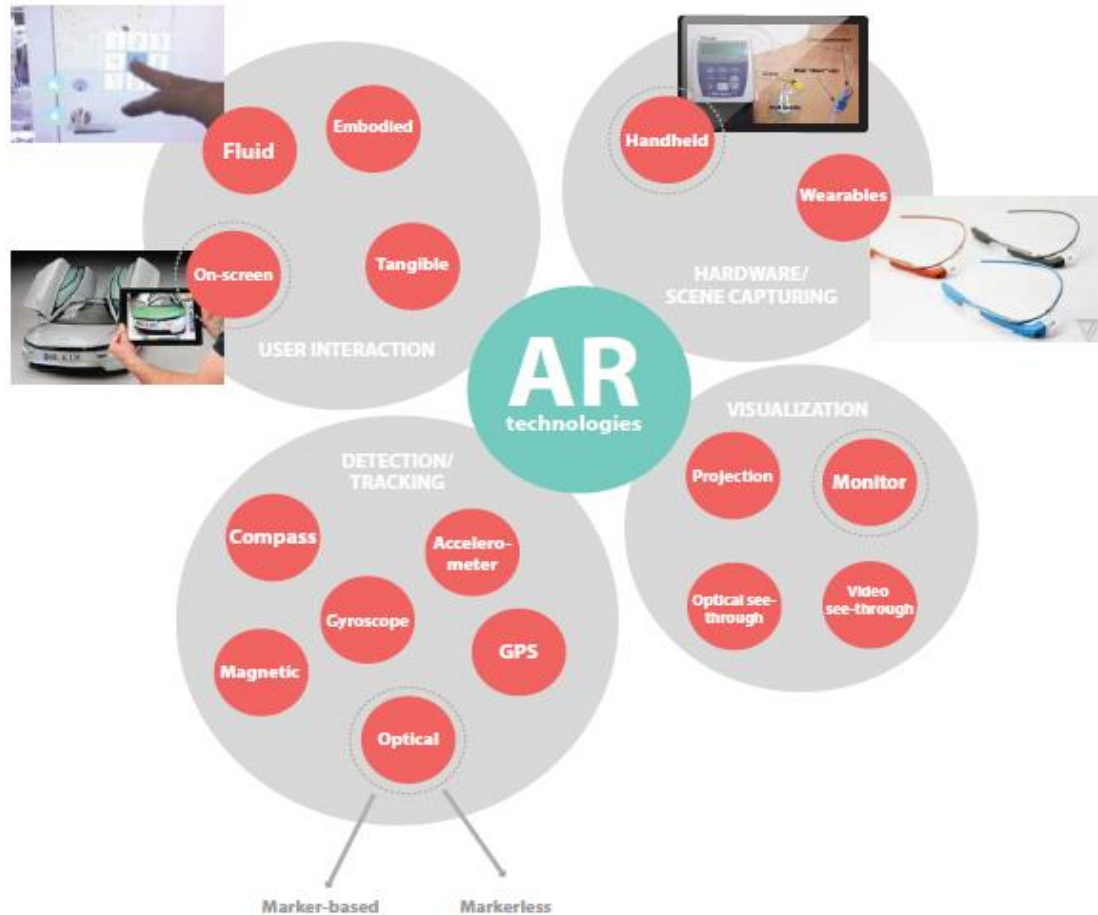


Figura 11. Diferentes tecnologías de realidad aumentada.

Fuente: H.Reuterdaahl [49], Mobile Marker-based Augmented Reality as an Intuitive Instruction Manual

Estos elementos son: La interacción con el usuario, el tipo de dispositivo hardware utilizado, el método de visualización y las técnicas de detección y seguimiento. La escogencia adecuada de las tecnologías a utilizar en cada uno de estos aspectos es determinante en la gestión de recursos, el tiempo de desarrollo y los costos de un desarrollo en AR, más aún si está orientado hacia algún otro tipo de tecnología como la IoT. A continuación, se analizan cada una de estas tecnologías y se relacionan de acuerdo a los elementos encontrados en los trabajos relacionados de este capítulo.

2.8.2. Interacción del usuario en AR.

Se ha delimitado este estudio a la interacción en pantallas (*Interaction On Screen*), por aspectos de facilidad en el desarrollo sobre una arquitectura de IoT, ya que es la forma más expandida de interacción. Las interfaces tangibles y las embebidas dentro de algún artefacto tienen otro método de interacción que es complementario y no es el núcleo de esta investigación.

La HCI brinda de diversas maneras de interacción, desde el clásico uso de comandos escritos, menús, botones, menús, botones, manipulación directa del dispositivo, lenguaje natural, entre otros métodos [30]; actualmente se ha evolucionado a la interacción mediante análisis del estado de ánimo o humor (*mood*), los gestos y (*mood*), los gestos y la mirada con la implementación de retículas o apuntadores, esto se observa en la en la

Figura 12

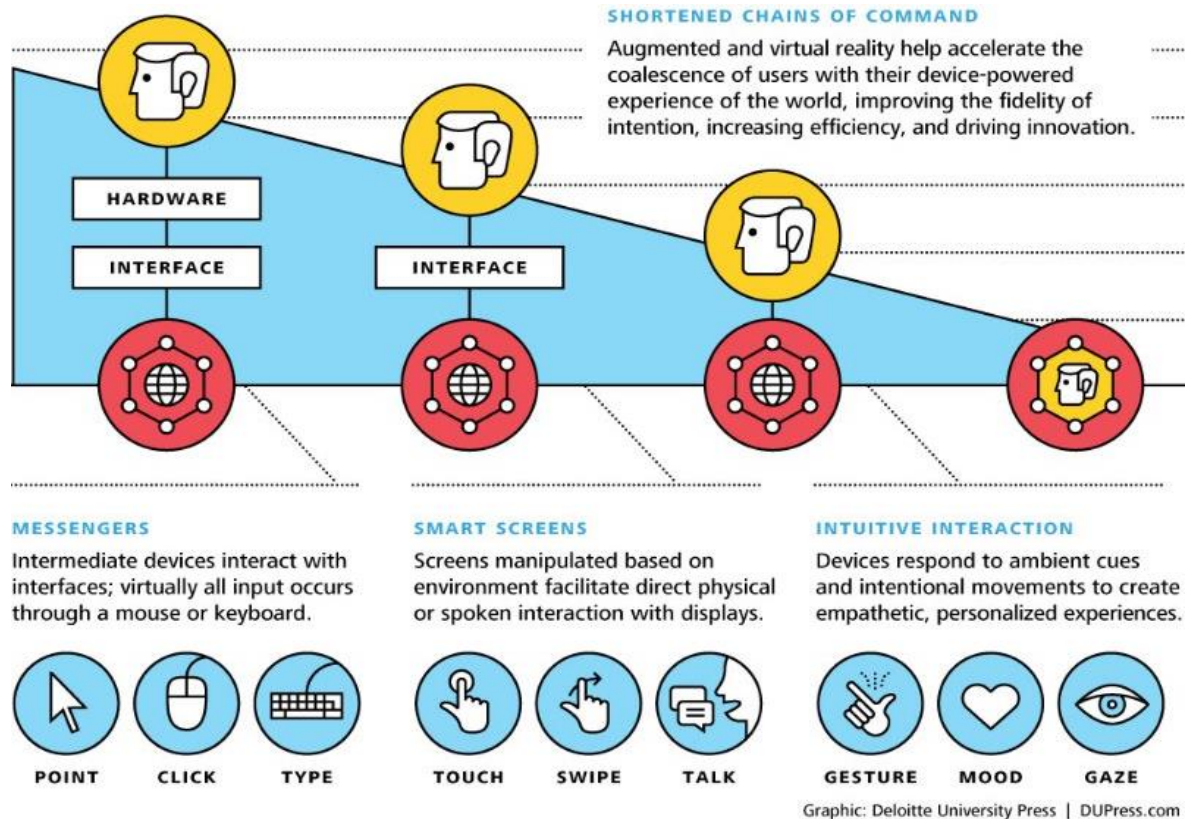


Figura 12. Evolución de la interacción en AR.

Fuente: DUPress, <http://dupress.com/articles/augmented-and-virtual-reality/?id=gx:2el:3dc:dup3036:awa:cons:tt16>.

Cabe observar que en AR orientada a la IoT, algunos de estos métodos de interacción al orientarlos al uso de dispositivos móviles y *eyewear*, carecen de sentido y en otros se hace necesario adaptarlos de tal manera que no vayan en contravía del concepto de la interfaz en AR, ya que si una interfaz es muy cargada de instrucciones o información no relevante según el contexto, ésta deja de ser útil o puede dejar de ser AR, además que existen funcionalidades que obligatoriamente deben crearse utilizando un modelo clásico de interacción.

2.8.2.1. Uso de la metáfora.

Para AR es ampliamente utilizado el concepto de la metáfora del mundo real, como método de interacción. La interacción natural con el mundo real se basa en los sentidos, así mismo una metáfora de estos es útil para crear interacciones similares a las del usuario con el mundo físico, o para crear nuevas maneras de interactuar tanto en las entradas como salidas de un sistema, las metáforas más destacadas en relación con los sentidos involucrados AR son:

- **Sentido del tacto:** Están relacionados con entradas táctiles (*touchpad*), o la pantalla táctil de los Smartphones, guantes hapticos para traducir señas [3], el uso de sensores de presión, botones virtuales entre otros.

Sentido de la vista: Dependen del medio de propagación de la luz, la iluminación y el movimiento de la cabeza y los movimientos de la cabeza y los ojos. Por ejemplo si se desea tener como referencia el centro de una pantalla a través una pantalla a través de una retícula visual, se utiliza un cursor o indicador (*pointer reticle*) fijo en este punto, el en este punto, el usuario lo percibe como su propio centro de visión dinámico y puede apuntar a un objeto de interés a un objeto de interés en el escenario definido [49], disparando la funcionalidad que se desea como se observa en la como se observa en la

- **Figura 13** También existen sensores que pueden capturar el movimiento de los ojos como la tecnología de SMI Tracking²², o las tecnologías para detección de rostros y movimientos gestuales. Si el medio de propagación tuviera diferentes condiciones de turbiedad o baja luminosidad, se pueden utilizar dispositivos infrarrojos para facilitar la visualización.

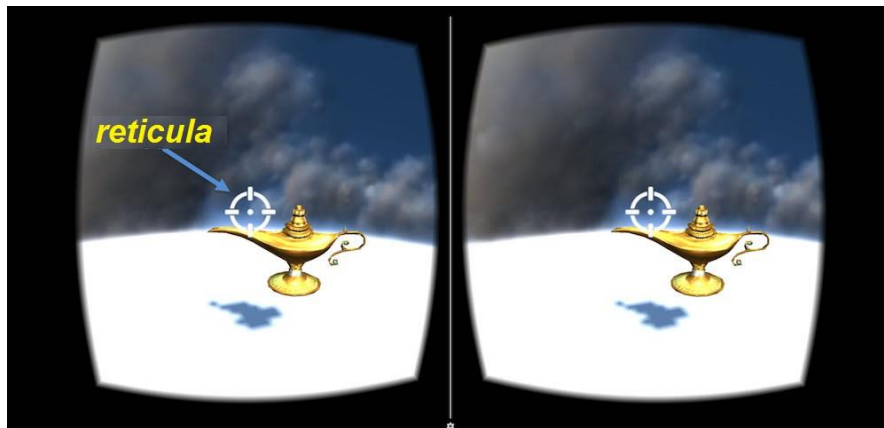


Figura 13. Retícula en dispositivo estereoscópico.

Fuente: Sitepoint, <https://www.sitepoint.com/what-in-the-world-is-a-reticle-cardboard-unity-sdk/>

- **Sentido del oído:** Los comandos por voz, texto a voz, alertas y guías sonoras, además de los avances científicos iniciados con los implantes cocleares abren un nuevo camino a la percepción de la realidad a través de este sentido.
- **Sentidos del Olfato y gusto:** Desarrollos como los de Meta Cookie de la Universidad de Tokio, intentan emular las sensaciones olfativas a través de la dosificación y combinación de estímulos olfativos químicos la visualización gustativa, demostrando que los seres humanos perciben diferentes sabores sin cambiar los estímulos químicos, solamente modificando su percepción visual y olfativa de lo que degustan.

También se pueden crear metáforas de los sensores comúnmente utilizados para tener la sensación de poseer nuevos sentidos, por ejemplo una brújula que se comporte de forma dinámica en el campo de visión del usuario, o un termómetro que cambia de color, evitando en lo posible que el usuario tenga que leer, procesar demasiada información o tener que mapearla entre convenciones simbólicas como numeraciones [49], o una visión del mundo real en los espectros infrarrojos, ultravioleta, o poder visualizar niveles de campos electromagnéticos entre otros.

Estas metáforas también son útiles para las personas con algún tipo de discapacidad [3], en combinación con las interfaces físicas, pues recientemente se ha demostrado que el uso de HMD's

²² <http://www.smivision.com/augmented-reality-eyetracking-glasses/>

de la mano con estímulos externos y metáforas de la realidad permiten recuperar en algún grado las funcionalidades perdidas por personas en situación de discapacidad, o tratar ciertas fobias.

2.8.2.2. Elementos de Asistencia.

En el campo de interacción con el usuario, otro concepto que presenta una gran importancia es la capacidad de asistencia al usuario por parte de la interfaz, este concepto cobra mayor relevancia cuando se habla de interfaces tipo AR.

- **Widgets.**

En AR se pueden crear elementos auxiliares y artefactos software (*widgets*), que permiten al usuario ubicarse en su entorno, seguir instrucciones o aprender, y en general cumplir un objetivo deseado. Elementos como alertas sonoras y visuales para los eventos que suceden dentro y fuera del campo de visión del usuario [3], [37], los indicadores de navegación como flechas estáticas que indican dirección o ubicación de una EI o PI y los widgets como mini mapas y radares son los más aceptados y bien percibidos por los usuarios a nivel de anotación y navegación [24].

Como se puede observar en la

Figura 14 ~~Figura 14~~, los mini mapas, radares e indicadores son útiles en AR siempre y cuando no vayan en contravía de este concepto, pues cuando los elementos de mundo real son ocluidos casi en su totalidad se pierde la noción de realidad aumentada y puede llegar a ser incluso contraproducente para el usuario el uso de estos widgets.



Figura 14. Indicadores auxiliares y radar en una aplicación de AR.

Fuente: Francis Storr, <https://www.flickr.com/photos/fstorr/4029336046/>

Los elementos asistenciales referentes a exploración y orientación son útiles, pero requieren más concentración por parte del usuario, teniendo que enfocarse en la herramienta y procesar la información que esta le brinda, así se obtienen datos útiles para la navegación en el entorno de AR [24].

Los widgets asistenciales encontrados, se pueden clasificar de la siguiente manera:

- **Navegadores AR:** Camino de niebla, camino de objetos, camino resaltado, flecha móvil, flecha anclada al usuario, flechas estáticas a lo largo del camino, widget mini-mapa con camino demarcado, flecha anclada a la cabecera de la pantalla.
 - **Orientación AR:** Visión térmica, fondo contrastado, widget brújula, widget radar.
 - **Anotación AR:** Etiquetas con nombres, etiquetas con logos o iconos, objetos resaltados, anotaciones e iconos en el mini-mapa.
 - **Exploración AR:** Mega-mapa, widget mini-mapa.
- **Cabecera de Pantalla HUD (Head Up Display).**

El elemento de asistencia más utilizado en los videojuegos se llama HUD y se refiere a la información que permanece sobre el primer plano de la pantalla en todo momento en forma de iconos o etiquetas, de iconos o etiquetas, como se observa en la

Figura 15 para AR también representa utilidad siempre y cuando no sature o genere oclusión en la experiencia de aumentación.



Figura 15. HUD Sobrecargado

Fuente: Hyper Reality, http://usa1.usastreams.com:81/storage/551/pictures10/Keiichi-Matsuda-Hyper-Reality_dezeen_01_644.jpg

El uso de HUD para anotar un objeto en el entorno es un aspecto útil para la navegación [24], además de ofrecerle otra alternativa de información al usuario o hacer seguimiento de un procedimiento o tarea [37][58], sin embargo se debe contrastar la distancia del objeto respecto al HUD evita tener elementos muy pequeños o muy grandes, esto previene que los objetos pierdan información o utilidad en la interfaz AR.

En la IoT los puntos y las EI son bases para la interacción, cuando el usuario pierde de vista el objeto de interés, la asistencia se puede realizar a través de superponer un túnel o embudo grafico que regrese al usuario al PI o EI [37].

- **Asistencia por superposiciones.**

Los recursos IoT, no solo brindan información relevante, también brindan servicios, en este punto el nivel de asistencia en una interfaz AR para la IoT debe poder ayudar al usuario a consumir dichos servicios llevando a cabo tareas o procedimientos, la asistencia por superposiciones brinda una alternativa útil para este efecto.

Las superposiciones consisten en colocar imágenes del mundo real o virtual para guiar al usuario en una tarea o usuario en una tarea o consecución de un objetivo. El trabajo de Petersen – Stricker [37] define cuatro técnicas de define cuatro técnicas de superposición visual como se observa en la

[Figura 16](#) ~~Figura 16~~.

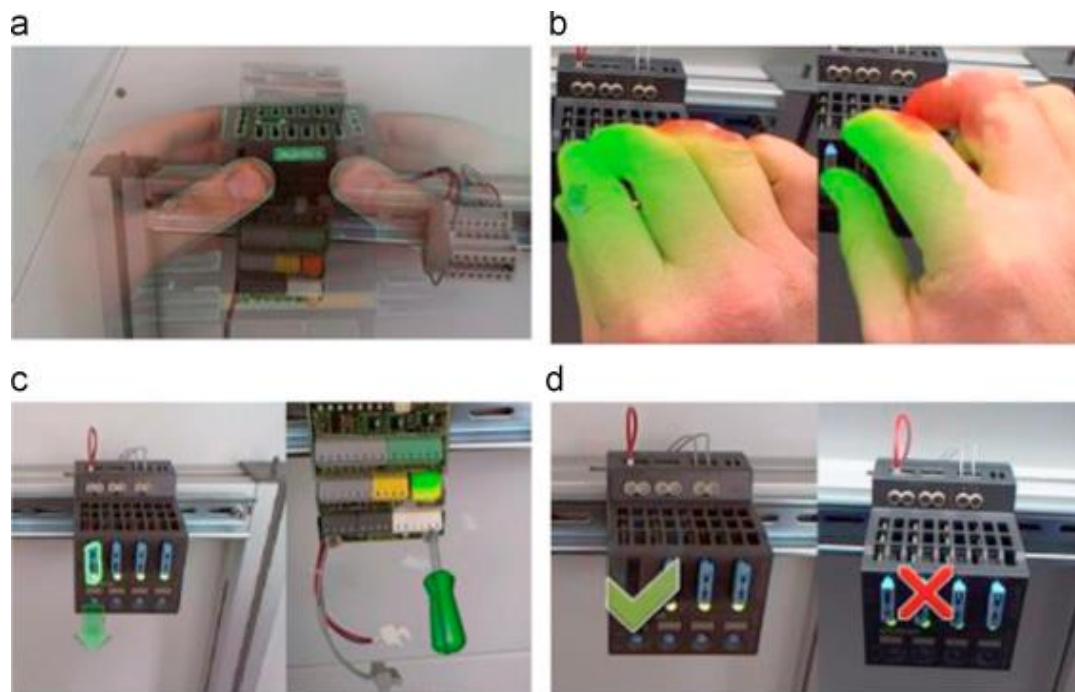


Figura 16. Superposiciones visuales.

Fuente: N. Petersen-D. Stricker [37], Cognitive Augmented Reality.

a) Superposiciones de procedimiento: Es una animación o imagen que brinda información sobre la tarea a realizar, se puede utilizar para brindar asistencia a través del contexto.

b) Retroalimentación enactiva: Se ejecuta en tiempo real durante la ejecución de la tarea, brindando una retroalimentación positiva cuando el procedimiento realizado es correcto y negativa en caso contrario, por ejemplo, utilizando los colores rojo y verde.

c) Superposiciones por anotación: Destacan o resaltan ciertas áreas de la escena dando instrucción o información adicional del procedimiento sea por etiquetas de texto o gráficos.

d) Evaluación de Superposiciones: Comparativa entre la tarea realizada y la imagen con la tarea hecha de modo correcto.

- **Zoom semántico.**

Es una jerarquía para manejar diferentes niveles de información que brinda un objeto o entidad a través de una interfaz, en donde cada nivel brinda más detalle en dicha información y los niveles se activan a petición o demanda del usuario como se observa en la Figura 17, es una metáfora que dosifica la cantidad de información que se le entrega al usuario sin saturarlo, según su perfil y sus necesidades. Para la IoT el zoom se puede realizar de manera automática apoyándose en servicios web semánticos [4].

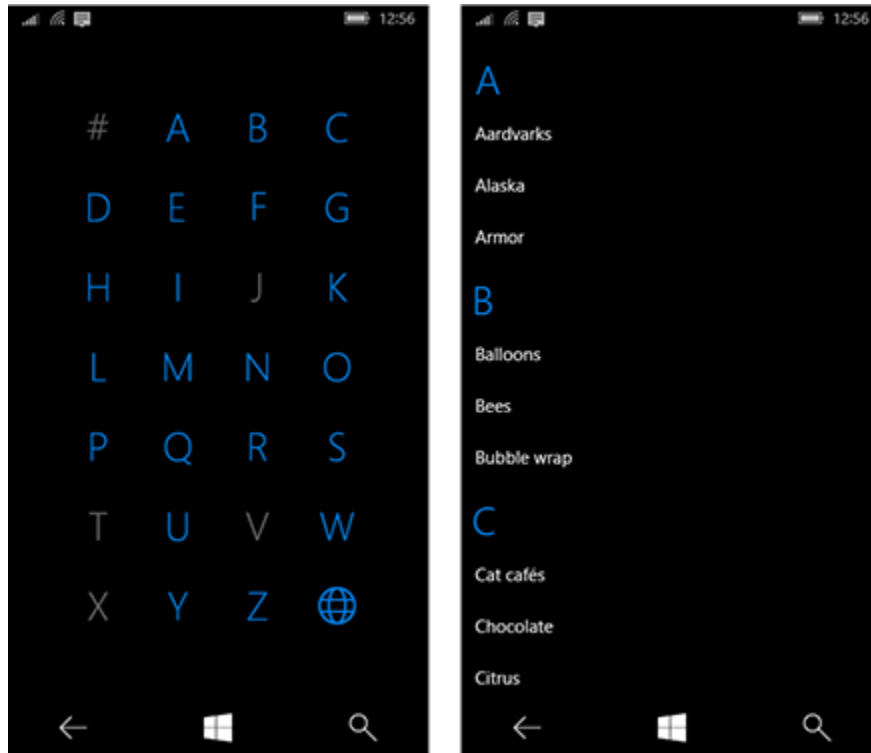


Figura 17. Zoom semántico.

Fuente: Microsoft, <https://msdn.microsoft.com/es-es/windows/uwp/controls-and-patterns/semantic-zoom>

Para la creación de interfaces en AR para la IoT el uso de un zoom semántico puede brindar un nivel valioso de interacción básica con los recursos IoT del escenario definido, un ejemplo es la aplicación de Hannah Reuterdahl [49], donde de manera intuitiva sin realizar un mapeo de los botones reales de los dispositivos involucrados, hace una descripción de su funcionalidad y su ubicación utilizando un cursor virtual que brinda un zoom semántico del punto de interés apuntado en el objeto virtual asociado al real.

2.8.3. Hardware y dispositivos para la captura de la escena.

La taxonomía propuesta por Milgram y Kishino [32], permite tener una visión amplia sobre las dimensiones que tienen los dispositivos visuales para realidad mixta, junto con las técnicas de visualización directa o transparente e indirecta o de video, y con las proyecciones realistas creando otra dimensión de la tecnología de dispositivos en AR, donde las técnicas de visualización y la tecnología hardware utilizada van ligadas, por esa razón estos dos aspectos tecnológicos se traslapan y dependen uno del otro.



- **Pantallas Montadas en la Cabeza *Head Mounted Display* HMD.**

Los dispositivos HMD brindan una vista transparente y/o estereoscópica de la realidad enriquecida con alta fidelidad, permitiendo mayor libertad de desplazamiento y un campo de visión mucho más natural que el que ofrece un dispositivo de mano [24], [34], [51], brindando orientación y navegación con mejor ergonomía [24], [37]. Actualmente existe gran diversidad de estos dispositivos y no existe una tendencia claramente definida sin embargo los dispositivos que proyectan imágenes directamente en la retina como Glyph²³ de Avegant, gafas inteligentes y dispositivos híbridos como el Microsoft Hololens²⁴ son tecnologías prometedoras, pero de alto costo, estos son los llamados vertibles para los ojos (*wearable eyewear*).

- **Dispositivos de proyección envolvente.**

Nelson Andre [34] muestra que los dispositivos utilizados para proyectar en video el mundo, en tiempo real también son una alternativa a la hora de implementar AR con HMD's. Los más populares son los desarrollos de la empresa Oculus²⁵ como el Oculus Rift y Oculus Gear, sin embargo para la proyección de la realidad hace necesario adaptar accesorios de video a este tipo de dispositivos; también existe la opción más accesible y económica del mercado, el proyecto de Google Cardboard²⁶, Google Daydream²⁷ y sus diversas variantes hardware.

En la

~~Figura 18~~ **Figura 18** se observa el espectro de los diversos dispositivos para la cabeza (*Headsets*), y aunque aparentemente la tendencia es hacia la virtualidad por la gran oferta de hardware en el mercado, los problemas a resolver respecto a portabilidad y ergonomía entre ambas tecnologías traduce en un mercado muy heterogéneo e inestable donde no se puede predecir su comportamiento, incluso la AR gana campo en los dispositivos de realidad virtual, con ciertas modificaciones, u otros fabricantes han optado por ofrecer soluciones a mitad de camino entre ambas tecnologías, lo que se conoce como realidad mixta.

²³ <https://www.avegant.com/>

²⁴ <https://www.microsoft.com/microsoft-hololens/en-us>

²⁵ <https://www.oculus.com/en-us/>

²⁶ <https://www.google.com/get/cardboard/>

²⁷ <https://vr.google.com/daydream/>

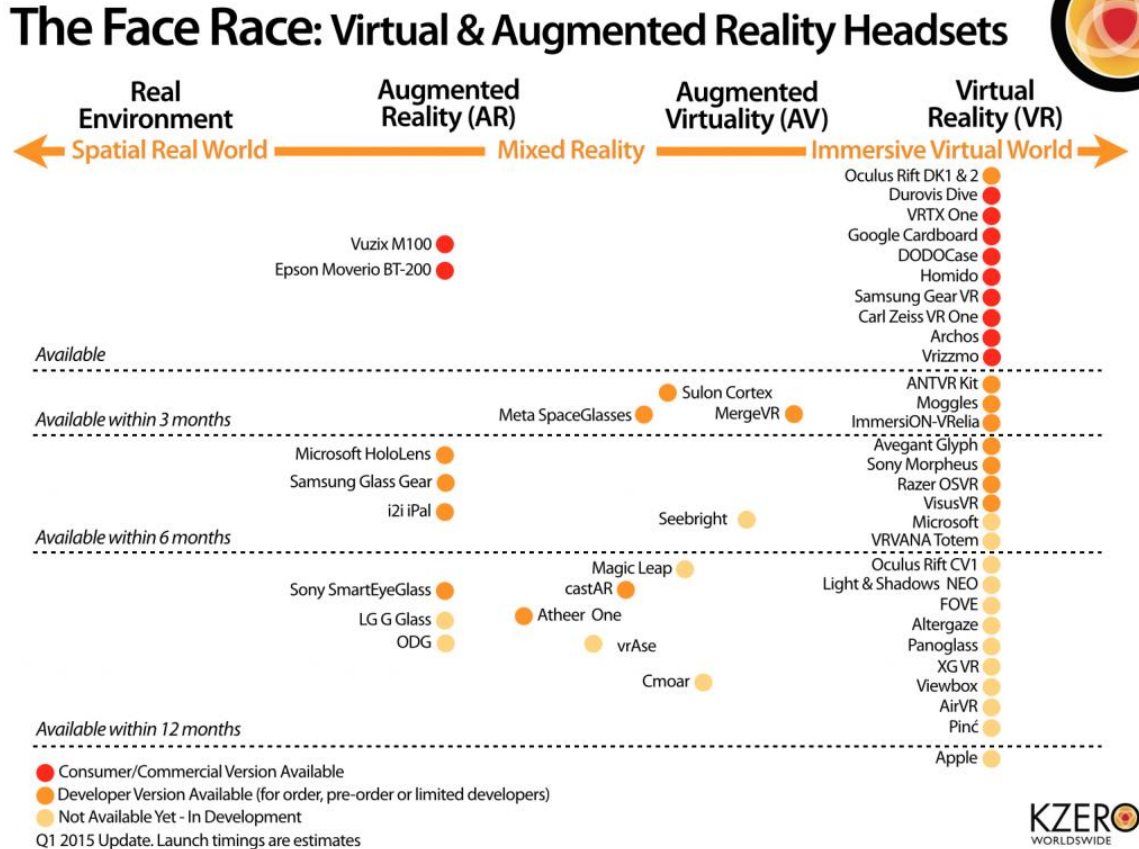


Figura 18. Espectro de dispositivos de realidad aumentada y virtual (2015).

Fuente: Kzero Blog, <http://www.kzero.co.uk/blog/category/virtual-reality/>

- **Interfaces de mano.**

Las tabletas y los teléfonos inteligentes brindan la libertad de desplazarse y explorar el entorno enriquecido por la AR entorno enriquecido por la AR de forma confortable [4], pero con menor ergonomía al tener que sostener el que sostener el dispositivo [17], sin embargo esto es debatible, como se observa en la

Figura 19Figura 19, pues las mismas razones de heterogeneidad y adaptabilidad de la tecnología de AR, hacen que los conceptos de ergonomía traduzcan en los conceptos de portabilidad y accesibilidad del dispositivo donde claramente el Smartphone como hardware de uso masivo gana en todos estos aspectos hasta el momento.

También de forma general se pueden incluir las interfaces de interacción natural como Microsoft Kinect²⁸, Leap Motion²⁹ [34], y Real Sense³⁰, las cuales brindan la posibilidad de interactuar en AR mediante el propio cuerpo, gestos y señas [51], aunque su estudio se acerca más al estudio de interfaces tangibles el cual no se aborda en este estudio.

²⁸ <https://developer.microsoft.com/en-us/windows/kinect>

²⁹ <https://www.leapmotion.com/>

³⁰ <http://www.intel.com/content/www/us/en/architecture-and-technology/realsense-overview.html>

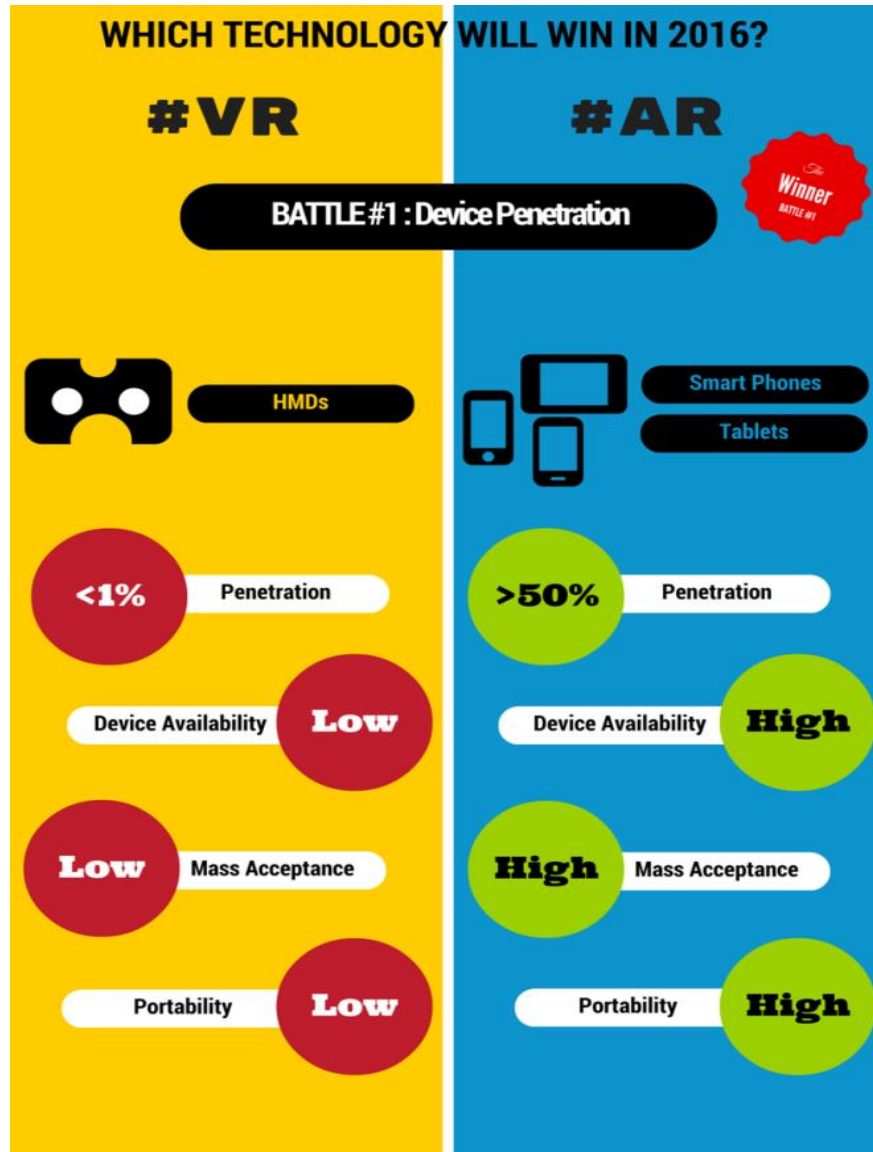


Figura 19. Comparativa de dispositivos AR vs VR.

Fuente: Iwanthistech Blog, <https://iwanthistech.wordpress.com/2016/01/01/battle-1-between-vr-ar-device-penetration/>

A futuro, algunas de estas interfaces se fusionarán o desaparecerán, dando paso a visiones futuristas como los hologramas o proyecciones en 3D, la tele presencia, entre otros. Además, debido a los avances científicos, los materiales como el grafeno y el fosfeno, permitirán tener pantallas flexibles ultra delgadas, pantallas polarizadas más sofisticadas y artefactos funcionales como bio-sensores, a un bajo costo y con menor tamaño, lo cual hará de la AR un concepto transparente y de uso cotidiano.

La tecnología orientada al desarrollo de interfaces de interacción en AR pueden incluso reemplazar los sentidos, lo cual es de vital utilidad para personas con algún tipo de discapacidad, por ejemplo una retina artificial, un implante auditivo combinados con alguna técnica de localización mejorará la calidad de vida de esta población [3], y en un escenario futurista si estas mejoras son útiles o presentan algún atractivo para las personas sin discapacidad, su uso también será masivo.

2.8.4. Métodos de Visualización.

Visualizar indica poder ver lo que no se puede ver a simple vista, en interfaces físicas se acude al uso de dispositivos para visualizar la AR utilizando HUD's como pantallas transparentes o de proyección de video, los dispositivos móviles como tabletas y Smartphones o una proyección envolvente de la realidad a través de video en vivo [32], también se utilizan proyecciones directamente en la retina, así se tiene aumentación a diferentes niveles de proximidad con la vista. Un ejemplo singular es el uso del fenómeno de polarización de la luz como el que utiliza castAR³¹ para crear HR. Esto evidencia que la técnica de visualización va de acuerdo al tipo de hardware utilizado.

Las proyecciones realistas o realidad aumentada espacial SAR (*Spatial Augmented Reality*) tiene como objetivo brindarle al usuario la sensación de estar interactuando con objetos reales bien sea proyectados en el mundo físico o a través de una pantalla, debiendo hacer un uso adecuado de las sensaciones de luces y sombras [33], son utilizadas para publicidad, arte o experimentos sociales como por ejemplo la proyección observada en un edificio (ver Figura 20) sin tener que cargar con el dispositivo que realiza la proyección. También se utilizan técnicas que brindan la posibilidad de eliminar la percepción de un objeto real, o modificarlo interactivamente a partir de una abstracción del objeto físico y el fondo que ocupa.



Figura 20. Proyección realista.

Fuente: Altavisión Producciones, <http://www.avproducciones.com/blog/mapeo-de-proyeccion-o-projection-mapping/>

2.8.5. Técnicas de Detección y Seguimiento.

La detección se refiere a la capacidad de ubicar y mantener la posición de las representaciones virtuales en el entorno real. La técnica más popular en AR es el uso de marcadores, generalmente de tipo QR (*Quick Response*) colocados en el entorno real o en maquetas como se observa en la

³¹ <http://www.castar.com/>

Interfaz de Realidad Aumentada Para la Internet de las Cosas.

Figura 21, también se pueden utilizar imágenes e incluso representaciones 3D de los objetos del mundo real, para este proceso de detección se hace uso de cámaras simples o dobles como la cámara web o la/las cámaras de un Smartphone [51].



Figura 21. Realidad aumentada por Marcadores.

Fuente: ALS, <http://www.alife-studios.com/portfolio>

Otras técnicas de detección y seguimiento son las técnicas de localización, siendo las más comunes el uso de la brújula o magnetómetro, junto con GPS y los acelerómetros como se observa en la Figura 22, estas técnicas están siendo explotadas con gran éxito por juegos como Pokemon Go³² de Niantic Inc., sin embargo, existe discusión si en realidad son juegos de AR o simplemente gráficos superpuestos en el entorno. La localización se ha clasificado en dos grandes campos, la localización en exteriores (*Outdoor*) con GPS como la tecnología dominante [3], [34], esta tecnología se fusiona con otras como la telefonía celular para lograr mayor precisión.



Figura 22. Realidad aumentada basada en localización.

Fuente: ALS, <http://www.alife-studios.com/portfolio>

El otro campo es la localización en interiores (*indoor*) aún no es muy precisa, pues tiende a ser una tecnología que indica proximidad, mas no una localización con exactitud, para esto algunas tecnologías se valen de las características de las ondas de radiofrecuencia – RF de las cuales se pueden obtener datos que se procesan y brindan información acerca de la localización aproximada de un objeto de interés en un sistema de referencia (coordenadas), mediante técnicas de procesamiento como la triangulación. Las fuentes de ondas RF más comunes para aplicar estas

³² <http://www.pokemongo.com/es-es/>



técnicas son los puntos de acceso WIFI, los dispositivos Bluetooth, y las antenas de telefonía celular [59].

También se pueden usar técnicas de huella (*fingerprint*), donde se hace posicionamiento del objeto de interés cuando se activa un marcador, por ejemplo, tipo QR o de RF como los RFID desplegados en una red [3], y comparando los datos asociados con una referencia previa [60]. Las técnicas que utilizan RF para comunicar estos datos de localización deben tener en cuenta los retardos por diversos fenómenos como la trayectoria o la propagación y demás fenómenos físicos que afectan a este tipo de ondas, pues son técnicas aun emergentes y no muy precisas [3].

Las técnicas híbridas combinan cámaras y otros sensores brindando mayor robustez y transparencia, eliminando el uso de marcadores en el proceso de detección [34], a esto se le denomina AR sin marcadores (*Markerless AR*). Un ejemplo de este tipo de técnicas es el llamado *SLAM*, un conjunto híbrido como algoritmos y datos de sensores que al procesarse en conjunto brindan la localización, orientación y otras características que le permiten a un recurso IoT reconocer su entorno, pudiendo extender sus funcionalidades y brindando más información [34]. Estas técnicas permiten reconocer cambios en el escenario en tiempo real [36], lo cual representa un alto valor en interacción de las interfaces de AR. Los proyectos actuales de esta tecnología son Tango Project de Google³³ y el SDK de Kudan³⁴.

2.8.6. Consideraciones finales acerca del diseño de interfaces de AR para la IoT.

Una interfaz de usuario debe permitir la comunicación entre humano y computadora, en su proceso de diseño se identifican objetos y acciones, la metodología de desarrollo que se escoja debe fundamentar sus bases en la eficacia de estas interacciones, y es determinante a la hora de probar la funcionalidad de la solución brindada, pues es la tecnología la que debe acoplarse al usuario, ya que el fin último es que el usuario controle la computadora y no al contrario.

El proceso de desarrollo de la interfaz, se debe definir de manera genérica sin importar el hardware, el software utilizado, el sistema operativo, ni las herramientas para el diseño del producto, las fases de creación de la interfaz pueden definirse como: análisis, diseño, construcción y validación, cada una con su iteración interna, siendo muy similar a los ciclos de las metodologías ágiles. Estos elementos aplican de una manera natural a las interfaces en AR sin embargo, es importante adaptar los elementos de diseño y tener ciertas consideraciones, ya que el mundo real intrínsecamente es parte de la interfaz y por ende de la solución software buscada.

Para este propósito se puede hacer referencia a Theo Mandel [28] quien define 3 simples reglas, las cuales se pueden adaptar al diseño de interfaces de la AR para la IoT:

1. Dar el control al usuario.

Al superponer contenidos virtuales en la realidad, se debe tener en cuenta que el objetivo es ayudar a obtener mayor información del entorno que la que se obtiene a simple vista y poder interactuar de alguna manera con las representaciones gráficas, cuando se desea gamificar el

³³ <https://get.google.com/tango/>

³⁴ <https://www.kudan.eu/>



entorno, la interfaz del juego le define las condiciones al usuario, pero en AR, por reflejo e instintivamente el usuario desea y debe tener el control.

Lo anterior se evidencia en la clásica acción refleja de intentar tocar la representación virtual en el entorno real y la frustración que se genera al no poder interactuar con dicha representación es alta. En AR se pueden distinguir modos de interacción, de manera directa o indirecta, con la simple mirada o el enfocarse en un área u objeto determinado, es ahí donde el control juega un papel importante a la hora de diseñar la interfaz, fuera de las clásicas funciones secuenciales de avanzar, guardar y salir.

2. Reducir la carga de memoria del usuario.

Observe que la interfaz debe facilitar las interacciones, más aún en una aplicación en AR cuyo objetivo es enriquecer la experiencia de la realidad, e incluso dar la sensación de aumentar los sentidos del usuario, por lo tanto, no tiene caso que se le invite a recordar procedimientos o detalles innecesarios o inútiles, pues entre más tiene por recordar el usuario, más proclive es a cometer errores y a frustrarse.

El equilibrio entre la necesidad de memorizar procedimientos por parte del usuario, la asistencia brindada por la interfaz, y la fluidez o intuición de la misma depende del campo de acción de la aplicación requerida, es decir de la asesoría obtenida en cada área del conocimiento[37], por ejemplo si el diseño está orientado a personas con algún tipo de limitación sensorial, decremento mental o de alguna otra índole, es clave la asesoría de un profesional de la salud para lograr el efecto o la funcionalidad deseada con un buen nivel de calidad y usabilidad [3].

Las consideraciones para una interfaz de AR en el dominio de la IoT, deberán tener en cuenta llevar un equilibrio adecuado, pues una interfaz demasiado intuitiva puede dejar de brindar información útil, más aún bajo los aspectos de tiempo real, y las necesidades de asistencia para lograr una tarea, pero por otra parte una interfaz sobrecargada de información irá en contravía del concepto de AR.

El etiquetado de objetos, las flechas, cursores, diálogos y notificaciones se adaptan muy bien a un entorno de AR, sin embargo, no se debe saturar al usuario con demasiadas etiquetas o representaciones. La virtualización de los objetos con características y funciones que le son familiares al usuario complementan el uso de la “metáfora” del mundo real [28] y junto con el adecuado uso del entorno objetivo, ayuda bastante a evitar que el usuario tenga que leer demasiadas cosas, o aprender muchas instrucciones, existen opciones como utilizar colores, animaciones, sonidos entre otros para reemplazar el texto de forma natural, así pues se precisa tener en cuenta llevar un equilibrio entre lo real, el despliegue de información, y la virtualización.

3. Construir una interfaz consecuente.

Se debe abordar la solución dividiéndola en posibles escenarios organizados según las necesidades del cliente, esto con el fin de tener una interfaz consecuente y homogénea, se deberán seguir las reglas de diseño generales y aplicarlas en cada escena AR presentada al usuario, implementando mecanismos de entrada consistentes y bien definidos.



Los objetos de interés de un escenario pueden tener niveles de complejidad, desde el más simple un objeto inanimado, un sensor, un actuador, un *Gateway* de objetos, un OI [3], e incluso un objeto semántico, cada uno con características definidas, el diseño de la interfaz debe tener en cuenta estas características para no romper con la continuidad de la escena, por ejemplo a la hora de contextualizar a un usuario mediante su ubicación. Los seres vivos también tendrían ciertos roles, bien sea de usuario o de objeto animado y en este caso las ayudas biométricas como la detección de rostros, la personalización y la autenticación también deben ser tenidos en cuenta a la hora del diseño consecuente a través de la contextualización.

El uso del zoom semántico en AR puede brindar más información acerca de un objeto o un punto de interés, esta es una característica de interacción valiosa, que brinda más herramientas al usuario para obtener el efecto deseado. En AR el zoom semántico debe permitirle al usuario saltar entre la información sin perder de vista el entorno, porque de lo contrario podría confundirse y generar consecuencias indeseables en el mundo real. No se debe usar el Zoom semántico para cambiar el ámbito del contenido, pues genera desorientación en el usuario y rompe la consistencia de la aplicación.

2.9. USABILIDAD Y EXPERIENCIA DE USUARIO ORIENTADAS A INTERFACES DE AR PARA LA IoT.

2.9.1. Concepto de Usabilidad.

Cuando se pone en marcha el proceso creativo para diseñar una interfaz de usuario, el concepto a tener en mente es la usabilidad, definida como la medida cualitativa de la facilidad y eficiencia con la que un humano emplea las funciones y características que ofrece un producto de alta tecnología [37], [46], sin embargo, es importante explorar este concepto, junto con algunos otros relacionados, en mayor profundidad, orientándolos al diseño de interfaces AR para la IoT.

Aunque este término no hace parte de la lengua española ni posee una definición formal, es muy utilizado para describir la experiencia del usuario al usar un producto, también es visto como la característica funcional de la HCI, viene del término inglés “*Usability*”, cuya traducción lo define como facilidad de uso, para este caso la facilidad de uso de una interfaz de usuario.

Se reconoce a Jakob Nielsen³⁵ como el padre de la usabilidad, pero muchos otros autores han desarrollado la idea desde diferentes puntos de vista, sin embargo, para este trabajo fue necesario orientar los conceptos hacia el uso de interfaces en AR para la IoT.

Formalmente y de manera general el estándar internacional de calidad del software ISO9126³⁶ define usabilidad de la siguiente manera: “*Es el conjunto de atributos relacionados con el esfuerzo necesario para el uso de productos software, y en la valoración individual de tal uso, por un establecido o implicado conjunto de usuarios.*”

³⁵ <https://www.nngroup.com/people/jakob-nielsen/> .

³⁶ <http://www.issco.unige.ch/en/research/projects/ewg96/node14.html#SECTION00311000000000000000>

Esta definición hace parte de un conjunto de atributos específicos que difieren con la definición de usabilidad como la facilidad de uso ergonómico (eficacia y eficiencia). Esta norma presenta una complejidad elevada [30], tanto para el uso de sus métricas como para implementar la evaluación sugerida, pues abarca todo tipo de software y muchos otros conceptos aparte del de usabilidad.

La ISO 9241-11³⁷ define usabilidad como: “Grado en que un producto puede ser usado por determinados usuarios para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso.” Esta norma analiza la calidad del producto software desde el punto de vista ergonómico en medida del rendimiento y satisfacción al usuario, definiendo los factores para darle calidad a un producto o ajustarlo para que cumpla cierta norma.

Bajo estas definiciones se entiende que la usabilidad puede ser medida según el tipo de producto, si está orientado a un producto software, son métricas orientadas al esfuerzo cognitivo para apropiarse dicho software, sin embargo, el uso de interfaces en AR combina elementos hardware y software, por lo tanto, la ergonomía es un factor que cobra importancia a la hora de evaluar usabilidad de manera integral. La usabilidad respecto a rendimiento (efectividad y eficiencia) puede ser medida, pero la satisfacción del usuario es un concepto más subjetivo y mucho más difícil de entender o medir.

Figura 23

La **Figura 23** es una adaptación que clarifica estos conceptos y los ubica adecuadamente dentro de un concepto mucho más amplio como lo es la experiencia de usuario, este concepto trae consigo otros factores relacionados con la sensación que percibe el usuario al usar la aplicación en un contexto determinado³⁸.

EXPERIENCIA DE USUARIO

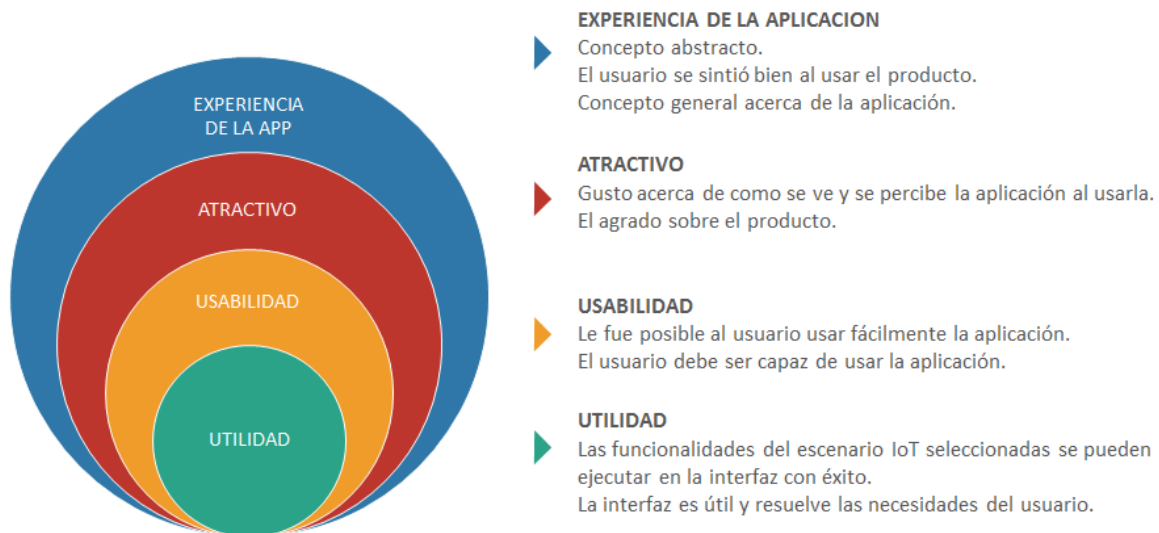


Figura 23. Usabilidad dentro de la experiencia de usuario.

Adaptado de: <http://www.neospot.se/usability-vs-user-experience/>

³⁷ http://webdiis.unizar.es/assignaturas/IPO/wp-content/uploads/2013/02/UNE-EN_ISO_9241-111998.pdf

³⁸ <http://usabilitygeek.com/the-difference-between-usability-and-user-experience/>



Aquí se involucra el concepto de utilidad por encima del de usabilidad, orientando el diseño a partir de las necesidades del usuario, si estas no se tienen en cuenta el producto final se convierte un producto en inútil, lo cual implicaría que sería irrelevante evaluar la usabilidad de un producto si este no le permite al usuario alcanzar la meta u objetivo deseado, también se deduce que la actitud del usuario no es relevante si no existe utilidad del producto. La satisfacción del usuario abarca un nivel más abstracto y exterior, bajo los conceptos de que tan deseable es el producto, como luce y como se siente, y los sentimientos que le genera dicha experiencia.

En conclusión el concepto de usabilidad de un producto redundante con su facilidad de uso y la valoración de este nivel de facilidad por parte de los usuarios finales, pero teniendo en cuenta que la prioridad para el usuario es que dicho producto le permita alcanzar sus objetivos, es decir le sea útil.

Enmarcar la valoración de la facilidad de uso o usabilidad de una interfaz de AR para la IoT, depende en gran medida del enfoque de la solución y su naturaleza, es decir si es un producto terminado o un prototipo, además que la utilidad de la interfaz la determina el escenario IoT donde se despliega, así para efectos de evaluación de usabilidad, es necesario verificar en primer lugar que la interfaz logre ejecutar las funcionalidades del escenario que se desean mediante realidad aumentada.

2.9.2. .Conceptos relacionados con la usabilidad

- **Intuición.**

Se puede definir la intuición en términos de la interacción del usuario con la aplicación, en el punto en el que el hardware y las interfaces se comienzan a ocultar a la vista del usuario y la informática se va fusionando en el entorno, esto hace que las funcionalidades y servicios que brinda una interfaz, vayan siendo más evidentes sin tener que ser deducidos o explicados.

Respecto a Usabilidad se pueden considerar 4 dimensiones para la medida de la intuición de la interfaz: la facilidad de uso (simplicidad), la asistencia, la rapidez y la ergonomía.

Desde el punto de vista de la AR para la IoT, se tiene análogamente como se intuye el mundo a través de los sentidos, la experiencia aumentada también debe ser intuitiva, es decir que no requiera esfuerzo, pero siempre existirá la necesidad de una asistencia, el ingreso de datos y el procesamiento de los mismos, aquí se requiere la habilidad del desarrollador para equilibrar ambos aspectos, el funcional y el intuitivo.

- **Métricas de usabilidad.**

Teniendo en cuenta que el factor de efectividad de la aplicación radica en la consecución de objetivos por parte del usuario y el factor de eficiencia radica en el mejor uso de los recursos por parte de la interfaz (incluido el tiempo), estas métricas brindan la medida sobre estos aspectos:

- **Exactitud:** Número de errores cometidos por los sujetos de prueba y si estos fueron recuperables o no al usar los datos o procedimientos adecuados.
- **Tiempo de ejecución:** Tiempo requerido para concluir la actividad.



Interfaz de Realidad Aumentada Para la Internet de las Cosas.

- **Recuerdo:** Qué tanto recuerda el usuario después de un periodo sin usar la aplicación.
- **Respuesta emocional:** Cómo se siente el usuario al terminar la tarea (bajo tensión, satisfecho, molesto, etcétera).

Las medidas cuantitativas tienen sentido si parten de un punto de comparación o referencia, es decir un desarrollo previo, y además si se trata de un producto terminado o un prototipo, las medidas subjetivas son más difíciles de medir y dependerán mucho del estado de ánimo o emocional del sujeto de prueba y de su opinión o percepción personal, la cual puede ser diferente en cada individuo.

- **Uso de la Metáfora.**

Como se observó anteriormente, el uso del concepto de metáforas es ampliamente utilizado en la creación de interfaces, este concepto soluciona los problemas de comunicación entre el usuario y la máquina, permitiendo así la facilidad de uso o la característica de ser una interfaz usable, sin embargo la utilidad de una metáfora va ligada a que tan representativa o comprensible sea para el usuario [61].

- **Interfaz.**

Es el hardware y software que le brindan comunicación y control al usuario, lo que el usuario ve o la puesta en escena, ve o la puesta en escena, la interfaz se encarga de hacer usable una aplicación, su objetivo es ayudar al usuario, y si no ayudar al usuario, y si no lo cumple se considera una interfaz inútil. La

Figura 24 permite clarificar estos conceptos y entender su relación con el diseño de interfaces de usuario [28], [61], [62].

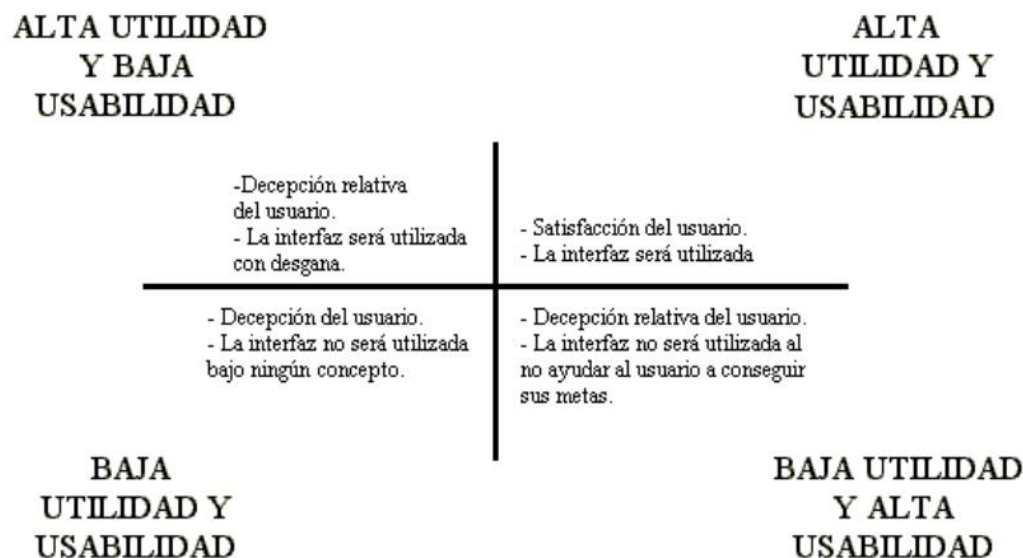


Figura 24. Desavenencias entre utilidad y usabilidad.

Fuente: Catalán Vega [61], Metodologías de evaluación de interfaces gráficas de usuario.

Bajo las premisas de los anteriores elementos, un producto final de AR en IoT, deberá considerar primeramente la utilidad de la aplicación mediante un proceso de validación simple, es decir que la aplicación ejecute adecuadamente las tareas o funcionalidades del escenario IoT, luego



dependiendo de los objetivos de la misma, hacia quien va dirigida y los recursos disponibles, se deberán considerar los aspectos de usabilidad y la experiencia de usuario (Apego) en el contexto definido [51], en este punto la evaluación pasa a ser subjetiva respecto a la satisfacción y experiencia del usuario.

Sin embargo al desplegar una interfaz AR sobre un escenario IoT, no siempre se tiene un producto final, ya sea porque el escenario es una prueba de concepto o un prototipo, o la misma interfaz no es un producto terminado, al evaluar estos casos, la utilidad se redefine como validación de las funcionalidades del escenario a través de la interfaz de AR a evaluar o validar.

2.9.3. Modelo de Evaluación de Usabilidad en aplicaciones de AR para la IoT.

Debido a que la tecnología de AR es un a tecnología emergente y más aún combinada con la IoT, pocos son los trabajos acerca de la usabilidad de interfaces en estos campos combinados, por lo tanto, se hace necesario analizar cómo se evalúan trabajos similares y crear una perspectiva de evaluación adaptado a las necesidades de este desarrollo.

Catalán Vega [61] hace un recorrido por las metodologías de evaluación de interfaces de usuario, específicamente para los test de usabilidad, define que su objetivo es evaluar la usabilidad y la utilidad de la interfaz propuesta, observando el comportamiento real de los usuarios al utilizarla, utilizando una muestra de los mismos, para obtener un marco de referencia para futuras versiones reduciendo riesgos y minimizando los servicios de atención al usuario tras la versión final.

Así, a partir del estudio anterior, se pueden definir unos pasos simples para realizar una evaluación de usabilidad que pueden ser aplicados para los objetivos de este trabajo, consisten en tener una muestra de la población objetivo, capacitarla, tener un entorno realista para realizar la evaluación, donde pueda actuar con naturalidad minimizando la presión de sentirse observado, para esto se le puede sugerir al usuario pensar en voz alta, lo cual podría brindar cierta realimentación al evaluador.

Respecto al enfoque de la evaluación puede ser orientado a evaluar el comportamiento del usuario, el como hizo las tareas asignadas y como percibe la aplicación, donde encontró dificultades, tratando de cuantificar su opinión a través de encuestas o formularios.

El otro enfoque es respecto al análisis de las tareas realizadas, a través de una medida de los logros o tasa de éxito, el tiempo que toma cada tarea en su ejecución, o el simple gusto de la interfaz.

A continuación, la Tabla 4 muestra algunos trabajos que guardan cierta similitud con interfaces de AR para la IoT, analizando su forma de evaluar dicha aplicación.

Aplicación	Sujetos de prueba	Fase1	Fase2
Gallag Strip [52].	Fase1. Usuarios expertos. Fase2. Inicialmente 2 expertos, 2 sin	Usuarios Framework de dimensiones cognitivas (CDN) – 13 cuestionamientos	de • Asignación de Tareas – 20 preguntas basadas en la



experiencia, luego 13 voluntarios.	dimensiones.	escala de usabilidad (System Usability Scale) John Brooke[63].
Aplicación Arquitectónica en AR [64].	Escala Likert.	
Aplicaciones Layar y Wikitude en Android [30].	Validación con las Heurísticas de Nielsen ³⁹ .	

Tabla 4. Evaluación de usabilidad en algunos trabajos relacionados.

Contexto

Se debe establecer el contexto de la prueba, bajo los aspectos de las condiciones iniciales como pueden ser el humor de la persona, su disponibilidad de tiempo, sus conocimientos acerca de la temática, su disposición a colaborar con la prueba, entre otros, esto se conoce como el “Background”. Luego se definen las tareas a realizar y una introducción al tema como capacitación previa, se prepara el entorno, y se realizan las mediciones a lugar a través de cuestionarios, o evaluación de las tareas según el tipo de medida que se requiera.⁴⁰

Sujetos de prueba:

Se procura en la medida de lo posible tener usuarios descansados, con la disponibilidad de tiempo y el estado de ánimo adecuado para realizar las pruebas, con motivación por conocer acerca de la tecnología de AR y su uso para gestionar recursos en la IoT. También se tiene en cuenta que el usuario que dispone de un dispositivo móvil tipo Smartphone, condiciona su perfil al tener previo conocimiento sobre el uso de este tipo de dispositivos, lo cual es deseable.

Capacitación:

En los últimos años el término “Realidad Aumentada” aunque puede sonar complejo e incluso intimidante para gran parte de la población, cuando un desarrollo se hace popular, las personas muestran un mayor interés o curiosidad por saber cuál es la tecnología que se involucra o en la cual se hace énfasis o es la novedad, incluso se introducen nuevos términos al imaginario colectivo, por ejemplo para mediados de 2016 se lanzó el juego Pokemon Go de Niantic, con un éxito sin precedentes, este juego introduce el término de “Realidad Aumentada” y “AR” en la cultura del momento, convirtiendo el termino en algo muy popular, sin embargo las personas perciben dicho termino pero no lo comprenden aun en su totalidad.

³⁹ <https://www.nngroup.com/articles/ten-usability-heuristics/>

⁴⁰ http://es.slideshare.net/DCU_MPIUA/usabilidad-ux-cmo-lo-medimos



Se hace necesario capacitar a los usuarios acerca de lo que es la tecnología de AR y el concepto de IoT pues aunque tengan interés en conocer sobre la misma o acerca de sus funcionalidades, deben tener claro los alcances de la misma y su correcta definición.

Los usuarios tienen diversos niveles de conocimiento acerca de la tecnología AR, se busca un grupo heterogéneo que pueda brindar luces acerca de los niveles respecto a facilidad de uso e intuición de la aplicación como si fueran los usuarios finales, para esto se les ofrece una breve introducción de que es lo que hace la aplicación y sus funcionalidades, sin influenciar la naturalidad con la que se debe ejecutar la aplicación, ni la transparencia con la que los OI prestan sus servicios e interactúan.

También se indica que el supervisor es un simple espectador cuyo deseo es solamente evaluar la interfaz y no al usuario en sí; el supervisor debe sugerir las tareas para la evaluación, mas no imponerlas e informar el tiempo de duración de la prueba, además de ayudar al usuario si este no puede completar alguna de las tareas.

Para evaluar usabilidad de manera cualitativa en interfaces de AR en la IoT, se puede utilizar la escala SUS, esta permite tener una medida cuantitativa de aspectos cualitativos como los de la experiencia de usuario, sin perder los aspectos de facilidad de uso e intuición de la interfaz.

Si la evaluación de este tipo de interfaces es cuantitativa, se debe tener en cuenta que la interfaz se diseña con un equilibrio entre el dominio de AR y las funcionalidades del escenario IoT asignado, esto implica que los tiempos de respuesta están condicionados por ambas tecnologías, además que todas las funcionalidades del escenario no siempre se van a poder desplegar en AR.

También se hace necesario para la toma de datos cualitativos de la interfaz de AR en la IoT, tener claro que no se trata de evaluar el escenario sino la aplicación que lo utiliza, y se requieren datos de otra aplicación AR sobre el mismo escenario o alguno similar para poder comparar los datos obtenidos en la evaluación.

2.10. CONCLUSIONES.

- Los elementos de diseño de interfaces son muy similares a los que aborda una metodología ágil, y los pasos de desarrollo expuestos garantizan en gran medida la usabilidad de la interfaz.
- Los problemas de heterogeneidad se extienden sobre el IoT en varios aspectos, a nivel de red, flujo de datos generados por las interacciones, tecnologías Hardware y software utilizados, siendo una necesidad el uso de estándares, y arquitecturas que permitan la convergencia y la normalización.
- La contextualización es fundamental para el diseño de una interfaz en AR ya que su funcionalidad es precisamente brindar guía, instrucción, y asistencia al usuario como una experiencia práctica en el entorno, y no como datos o metadatos aislados. Esto en últimas es lo que permite generar información útil y enriquecida al usuario además de nuevos servicios.
- La localización en interiores añade complejidad a un desarrollo de interfaces en AR, haciendo un balance de las necesidades del cliente y las posibilidades del desarrollador. Se pueden encontrar soluciones adecuadas con algún esfuerzo, debido a que es una tecnología en



Interfaz de Realidad Aumentada Para la Internet de las Cosas.

desarrollo que presenta alta heterogeneidad, pero también variadas opciones al momento de su implementación tanto en costos, recursos y precisión.

- La característica de los objetos de ser *context aware* es la que permite la interacción y la creación de nuevos servicios, estas dos características enriquecen y dan valor agregado a las interfaces de realidad aumentada en la IoT.
- La identificación de escenas con las tareas específicas de cada requerimiento, por ejemplo, en cada historia de usuario o cada funcionalidad, es una buena manera de abordar el desarrollo de interfaces en AR.
- El uso de servicios web de tipo semántico como ontologías y modelos brinda mayor intuición y riqueza a las aplicaciones para la IoT en entornos inteligentes, permitiendo mejor interacción y mejores servicios.
- El zoom semántico utilizado correctamente es una herramienta que brinda un valor agregado a la interacción en interfaces de AR para la IoT.
- En AR las técnicas de seguimiento son la primera fuente de información sobre la orientación de un objeto en un entorno.
- El Smartphone se comporta a la vez como OI y dispositivo de visualización, e incluso como HMD con el accesorio VR adecuado.
- La evaluación de la interfaz deberá primeramente validar la utilidad de la misma, si la interfaz no es un producto final o está basada en un escenario que no está terminado, esta utilidad se redefine como la validación de las funcionalidades de dicho escenario y se hace de manera informal con el cliente asumiendo el rol de usuario.
- La evaluación de la usabilidad de la aplicación se ha de basar en la experiencia de usuario para obtener refinamiento y equilibrio entre lo funcional del escenario y lo intuitivo de la aplicación.

A continuación en el capítulo 3, a partir de este marco de referencia se propone una guía de desarrollo para aplicaciones de AR orientadas a la IoT teniendo en cuenta las principales características, técnicas, y elementos detectados en el marco de referencia.



3. GUIA PARA EL DESARROLLO DE INTERFACES DE REALIDAD AUMENTADA PARA LA INTERNET DE LAS COSAS

A partir del marco de referencia propuesto en el capítulo 2 de este documento, se pueden obtener una serie de elementos y pasos a seguir para el desarrollo de interfaces de AR para la IoT, teniendo en cuenta los modelos y metodologías estudiadas junto con el análisis de los trabajos relacionados y la extracción de elementos que permitan crear la aplicación requerida. A continuación se describen ciertas consideraciones y las actividades de cada fase.

3.1. COMPONENTES.

El proceso de creación de esta guía de desarrollo ha tomado varios elementos a partir del modelo de prototipos, las metodologías ágiles Scrum y XP, y una serie de buenas prácticas que recogen elementos de AR tanto a nivel tecnológico como a través del análisis de trabajos relacionados, todo esto orientándolo a la IoT, en conjunción se obtiene una guía de desarrollo fácil de utilizar, enactiva, y rápida (ver Figura 25).



Figura 25. Componentes de la guía de desarrollo.

3.2. FILOSOFIA Y VALORES.

Es importante antes de definir el enfoque a seguir, establecer ciertos valores mediante los cuales se desea agilizar el proceso de desarrollo sin perder flexibilidad, enmarcados en una filosofía de creación de prototipos útiles y usables con una buena experiencia de usuario, sin abordar los aspectos de calidad como lo son el diseño gráfico, la personalización y la seguridad, ya que son roles especializados que no debería asumir el desarrollador primario. Estos valores son:



- **Enactividad o enacción.**

Esta característica permite al desarrollador corregir, insertar modificaciones y resolver imprevistos durante el proceso de desarrollo de la aplicación de la mano con el cliente, es una experiencia personal durante la cual al tiempo que se buscan las soluciones que resuelven los requerimientos del cliente, se apropia y adapta conocimiento orientado a otras tecnologías; para este caso las necesidades de un desarrollo para la IoT, construir nuevo conocimiento al adaptarlas hacia una aplicación en AR.

No se debe confundir este concepto con la enactividad aplicada a los recursos IoT, esta representa un gran reto a nivel de inteligencia y la capacidad de experimentar y aprender de los OI, y no se considera en esta propuesta, sin embargo, la acción de contextualizar junto con el nivel de interacción que posean los objetos en un escenario definido podrían mostrar el camino para implementar esta característica en dichos recursos.

- **Auto organización.**

La disciplina es clave a la hora de asumir un proyecto de desarrollo, este aspecto es el más difícil de asumir ya que implica un cambio en la mentalidad del desarrollador, para esto las diferentes metodologías a través de unos valores establecidos, exigen apropiarse ciertas reglas. Sin embargo, dada la flexibilidad de la guía propuesta se sugiere como medida básica establecer un horario de trabajo, y un espacio para la discusión del desarrollo del proyecto con el cliente para que exista retroalimentación.

- **Documentación.**

Se ha de documentar el desarrollo de una forma clara y sintetizada con el objetivo de poder ser replicado sin importar la heterogeneidad que pueda surgir a lo largo de su desarrollo, ya que algunas herramientas software con el tiempo pasan a ser de propiedad privada, se estanca su desarrollo, dejan de estar soportadas o desaparecen.

3.3. CUESTIONAMIENTOS A RESOLVER.

- 1 ¿Cuáles son los objetivos de construir una interfaz de AR para la IoT?
- 2 ¿Cuáles son los requerimientos para la creación de interfaces AR para la IoT?
- 3 ¿Cuáles son los pasos a seguir para crear una interfaz de AR para la IoT?
- 4 ¿Qué roles participan en la construcción y cuáles son sus habilidades?
- 5 ¿Cuáles son los elementos necesarios para la creación de interfaces AR en la IoT?
- 6 ¿Qué tecnologías están asociadas a la creación de desarrollos en AR para la IoT?
- 7 ¿Qué consideraciones se deben tener en cuenta a la hora de crear interfaces AR para la IoT?

3.4. GUIA DE DESARROLLO.

De manera general se definen 4 fases generales que se pueden utilizar para un desarrollo de software en particular (ver Figura 26), así para un desarrollo de AR en la IoT, se adaptan las siguientes: análisis, diseño, construcción y validación, este enfoque va orientado hacia un solo desarrollador o un grupo pequeño de desarrolladores.



Figura 26. Guia de desarrollo.

3.4.1. Fase 1. Analisis.

Dentro de esta fase se describen el contexto de la interfaz junto con el enfoque y los requerimientos para la construcción de la misma.

Actividad 1. Contextualización: Se indaga al cliente para definir claramente el contexto de la aplicación en AR, donde será desplegada y para quien. Esto con el fin de seleccionar la tecnología y los elementos más adecuados para establecer una solución.

Tarea 1. Establecimiento del Contexto. En esta parte se delimita el dominio de trabajo de la aplicación, si hace parte de algún proyecto más grande o con algún objetivo en particular y un enfoque que indique el objetivo de la interfaz y para quien va dirigida, también se establece si la interfaz va a ser montada sobre un escenario terminado, y si se requiere un producto final o un prototipo según los recursos que se vayan a asignar.

Tarea2. Establecimiento de la Arquitectura IoT: Partiendo del hecho que la interfaz de AR se va a desplegar sobre un escenario IoT definido, se requiere comprender como fluye la información dentro de dicho escenario, para esto se requiere la arquitectura base para su creación, con lo cual se puede crear una estrategia orientada a resolver los requerimientos HCI que nacen de implementar la interfaz AR sobre dicho escenario.

Tarea 3. Definición del Escenario IoT: Aquí se describe explícitamente el escenario IoT donde será desplegada la interfaz, sus recursos IoT, como se denominan y cuál es su nivel de inteligencia si la tuvieran.

Actividad 2. Requerimientos: Dentro de esta actividad se definen lineamientos referentes a funcionalidades y riesgos, delimitando el alcance de la aplicación.



Tarea1 Definición de roles: En la definición de los actores o partes, el rol del cliente es indispensable, pues al incluirlo en el proceso de desarrollo será una persona con la cual se pueden intercambiar opiniones, puntos de vista y sobretodo establecer los requerimientos, de ahí que este rol es vital para el desarrollo.

El rol de usuario depende del enfoque del software a desarrollar y el equipo de desarrollo, si el producto no es un producto terminado como lo es el primer prototipo desarrollado, el cliente o incluso el mismo desarrollador pueden hacer las veces de usuario para la validación de las funcionalidades del escenario IoT, brindando la retroalimentación necesaria.

Para los prototipos siguientes, una vez comprobada la funcionalidad del escenario, se pueden desarrollar pruebas de usabilidad con usuarios de prueba expertos o no, y el desarrollador asumirá el rol de supervisor de la misma, obteniendo nuevamente retroalimentación para refinar el siguiente prototipo.

Tarea 2. Recolección de requerimientos: De común acuerdo con el cliente, de manera informal se le pide que narre sus necesidades acerca de la aplicación, de forma realista, estableciendo funcionalidades básicas, luego se delimita el proyecto estableciendo los alcances del mismo, para luego generar como producto unas historias de usuario, también se establecen los recursos con los que se cuenta y un tiempo estimado de desarrollo, teniendo en cuenta apropiación de tecnologías, etapas de diseño y el número prototipos a construir, dejando claro que cada prototipo se ha de refinar de la mano con el cliente, para que no se frustre haciéndose altas expectativas.

Tarea 3. Asignación de Tiempos: Bajo el tiempo asignado a la realización del proyecto es conveniente administrarlo de tal manera que permita apropiar las tecnologías usadas en el escenario IoT para la interacción con la interfaz, y el tiempo requerido para el desarrollo, teniendo en cuenta la necesidad de documentar y apropiar las tecnologías en AR necesarias, pero sin perder flexibilidad, es decir un equilibrio entre tiempo suficiente y necesario.

3.4.2. Fase 2. Diseño.

Durante el diseño se debe tener un flujo de desarrollo consecuente, siguiendo la filosofía del modelo de prototipos, se establecerán tareas y requerimientos referentes a los servicios IoT y las interacciones que se tengan a lugar en el escenario objetivo, agrupados mediante escenas, y ese conjunto de escenas generarán un prototipo funcional. Es conveniente respetar el espacio de tiempo que requiere el desarrollo de cada escena, junto con la retroalimentación respectiva con el cliente, las modificaciones encontradas en las pruebas de cada escena tienen lugar en la siguiente iteración, es decir en el desarrollo del siguiente prototipo, mas no durante el tiempo asignado al desarrollo de otra escena para no romper dicha secuencialidad.

Actividad 1. Establecimiento de tecnologías. A partir de la arquitectura, el escenario físico IoT, y las necesidades del cliente captadas como funcionalidades se establece que elementos tecnológicos se encuentran definidos, y cuales están por definirse.



Tarea1. Tecnologías establecidas por la arquitectura y el escenario IoT: En esta tarea se extraen las tecnologías que serán la base necesaria para la interacción entre la interfaz y el escenario IoT, con esto se puede si el tiempo asignado previamente para su apropiación es suficiente o demasiado, además de poder brindar el espacio suficiente para la enacción.

Tarea 2. Tecnologías a utilizar para el despliegue de la interfaz AR: Aquí se establecen las herramientas necesarias para el despliegue de la interfaz, a partir de la población objetivo y con los recursos asignados, se establece el sistema operativo para el desarrollo y el despliegue, los entornos de desarrollo integrados IDE (*Integrated Development Environment*) o los motores de videojuegos a usar, y las librerías o SDK's necesarios para el despliegue AR en dichos entornos.

Cabe notar que por la naturaleza de la aplicación existe la posibilidad que se incorporen nuevas tecnologías a lo largo del proceso de desarrollo según las necesidades que vayan surgiendo, pero siempre manteniendo un equilibrio entre planeación y desarrollo.

Actividad 2. Planeación de los prototipos. En esta actividad se definen el número de prototipos a utilizar, que mínimo serán 2, junto con sus tiempos estimados en lo posible según el número de escenas generadas que se sugieren no sean más de 5.

Tarea 1. Definición de los prototipos. El prototipo 1 está encaminado a resolver la utilidad de la interfaz, en este caso los servicios que el cliente requiere en base al escenario IoT definido, para esto se hace necesario priorizar los requerimientos encontrados en las historias de usuario y haber establecido el enfoque de la solución definiendo claramente quien tomara el rol de cliente y el rol de usuario de prueba ya que el prototipo no es un producto final. El prototipo 2 y los siguientes se crean a partir de las observaciones encontradas en el prototipo 1.

Tarea 2. Asignación de las escenas: una vez se tiene las funcionalidades básicas prioritarias de cada historia de usuario, se sugiere asignarle a cada historia una escena, más adelante si se hace necesario por practicidad o usabilidad fusionar algunas escenas, se puede hacer rápidamente, por eso la definición de los requerimientos prioritarios se debe hacer primero y debe verificarse que su distribución a lo largo de las escenas sea uniforme y consecuente con el flujo de información del escenario para crear el primer prototipo, los requerimientos no prioritarios se resuelven en los siguientes prototipos según su relevancia.

Tarea 3. Planeación. Cuando se comprenda la forma como se encuentra desplegado el escenario IoT, el flujo de información y la tecnología involucrada, el tiempo asignado al desarrollo debe mantener cierta flexibilidad, pero como mínimo se sugiere tener 1 semana por escena, donde al final de esta, se haga una retroalimentación con el cliente de común acuerdo pero como mínimo una vez por semana, luego pasar a la siguiente escena sin implementar las observaciones de la anterior escena, pues es contraproducente y le quita tiempo de desarrollo a la escena en desarrollo.

Se estima entonces que la creación de un primer prototipo sea alrededor de un mes, pero se debe tener en cuenta un margen de tiempo para la enacción y la apropiación de las tecnologías AR requeridas. Al finalizar el primer prototipo se hace necesaria la validación



de las funcionalidades por parte del cliente, para lo cual también hay que separar un tiempo, esto permite decidir si el primer prototipo se destruye o si sirve como base para los siguientes. Seguidamente los demás prototipos entraran a resolver los demás requerimientos bajo el mismo procedimiento, pero al final de estos no se validan las funcionalidades básicas, sino se entra a evaluar la usabilidad de la solución.

3.4.3. Fase 3. Construcción.

En esta fase se definen los elementos técnicos de la interfaz referente a la comunicación, la interacción con el escenario, y las tecnologías de interacción con el usuario, los dispositivos de visualización en AR, técnica de visualización y seguimiento.

Actividad 1. Vision general de la interfaz AR. En esta actividad se definen aspectos generales de la interfaz condicionados por los recursos disponibles, dispositivos y técnicas a utilizar en el despliegue, los servicios que se pueden ofrecer utilizando AR y los que requieren otro tipo de interacción.

Tarea 1. Definición de las Interfaces físicas: Según los recursos asignados y el contexto de la aplicación se selecciona la tecnología más adecuada según las opciones del marco de referencia propuesto, como pueden ser dispositivos móviles o HMD's, esto permite determinar la técnica de visualización a utilizar.

Tarea 2. Comunicación con el escenario: Según la tecnología definida por el escenario IoT y la arquitectura básica, se tiene una serie de elementos como protocolos y formatos, los cuales se hace necesario incluir en el despliegue de la interfaz a través de librerías y desarrollos externos, evitando en lo posible tener que implementar nuevas funcionalidades a dicho escenario para resolver necesidades de la interfaz, pues esto se sale de los objetivos de la misma.

Tarea 3. Clasificación de los servicios: En esta tarea se observan los servicios que requiere el cliente, y se clasifican los servicios que requieren una interacción clásica parcial o total, es decir una secuencia de opciones, menús y texto, decidiendo si es posible o necesario implementarle un despliegue en AR, o alguna opción de desarrollo.

Actividad 2. Visión técnica de la interfaz AR. En esta actividad se definen los elementos de interacción con el usuario, elementos asistenciales, las representaciones virtuales de los recursos IoT, el HUD, los menús y botones necesarios según se requiera en cada escena, en cada prototipo según lo visto en el marco de referencia creado.

Tarea 1. Establecimiento del HUD. Esta pantalla deberá mostrar las funcionalidades básicas como la navegación entre escenas, el contexto, salir, cancelar, y otras opciones según se requiera y en este orden de prioridad. Según cada escena se requerirá entradas por parte del usuario, estas deberán ser consecuentes con las salidas del HUD como son alertas y convenciones según sean los requerimientos y su prioridad.

Tarea 2. Representaciones de los recursos IoT: Cada recurso IoT tiene sus características propias y su nivel de inteligencia, algunas representaciones se obviarán dependiendo del contexto y las necesidades establecidas, pero en general cada recurso IoT según su nivel



de inteligencia podrá prestar ciertos servicios, referentes a medidas, datos, interacciones, cada aspecto puede ser representado de manera diferente y utilizando metáforas del mundo real como se observó en el marco de referencia propuesto.

Tarea 3. Despliegue de servicios: Los servicios que presta el escenario IoT, según su clasificación, serán puestos en escena de manera consecutiva, es decir siguiendo la lógica de interacción del usuario, algunos necesariamente harán uso de elementos de interfaz de usuario como botones, o campos de texto entre otros, sin embargo en AR, la opción más primitiva es la selección del objeto, de ahí en adelante el desarrollador según su conocimiento y recursos disponibles puede implementar otro tipo de interacciones para desplegar los servicios.

3.4.4. Fase 4. Validación.

En la fase de diseño se han designado espacio para la validación y evaluación de los prototipos creados con los roles previamente establecidos, de esta manera la validación de la interfaz en el primer prototipo es un proceso que permite el refinamiento de la misma a través de la continua retroalimentación con el cliente y los usuarios de prueba; cabe hacer la observación que la validación es sobre la interfaz de AR, mas no sobre el escenario IoT, por esto la necesidad de tener claro el enfoque de la solución respecto al escenario IoT y como este condiciona el funcionamiento de la interfaz en AR.

Actividad 1. Validación del prototipo 1. Se debe validar la funcionalidad del escenario IoT en el prototipo 1 a lo largo del proceso de desarrollo, con alta participación del cliente, pues este es en últimas quien define y clarifica las necesidades que debe resolver la solución, además de permitir delimitar que servicios es posible desplegarlos en AR con los recursos disponibles, pues de antemano se tiene una alta complejidad al ser una tecnología emergente, heterogénea y de constante evolución. La prueba por parte del cliente asumiendo el rol de usuario en cada escena permite tener observaciones puntuales y mejora la secuencialidad del prototipo 1.

Las funcionalidades básicas del escenario IoT y las interacciones entre los recursos del mismo pueden permitir la generación de nuevos servicios, pero no necesariamente la AR puede ser la mejor solución para todos ellos.

Tarea 1. Validación por escenas: Cada escena tiene requerimientos diferentes, e incluso su desarrollo puede durar más o menos del tiempo asignado, se debe confirmar que la funcionalidad básica del escenario IoT funciona correctamente, el cliente evalúa y realiza sus observaciones de manera informal, el desarrollador toma nota para la optimización en el siguiente prototipo debiendo eliminar, mover o integrar elementos en la escena objetivo.

Tarea 2. Validación general: Una vez terminado el prototipo 1 se hace una prueba final con el cliente para hacer observaciones generales de la aplicación, y verificar que los servicios y funcionalidades relacionadas con el escenario IoT y los requerimientos prioritarios sean cumplidos en su totalidad.



Actividad 2. Validación de los prototipos siguientes: Con los datos obtenidos se procede a mejorar la interfaz de AR para la IoT, pues al tener las funcionalidades y servicios básicos del escenario IoT funcionando correctamente, falta mejorar la experiencia de usuario, en aspectos como la presentación, la facilidad de uso, la asistencia brindada y la intuición de la interfaz.

Tarea 1. Evaluación de usabilidad: Mediante el uso de test y métricas de usabilidad se puede realizar la evaluación de la interfaz de manera cualitativa o cuantitativa, cualitativamente se hace esta evaluación si no existen antecedentes de la interfaz desarrollada o si no es un producto final, pues no existe un punto de comparación con la misma, cualitativamente se evalúan tiempos y tareas como lo indique la técnica de evaluación escogida con los usuarios finales si el producto es una versión final.

Tarea 2. Revalidación de la interfaz. Si se requieren más prototipos y los recursos asignados lo permiten, se procede a evaluar el prototipo de la misma manera que el prototipo 1, anotando observaciones por escena para su mejora posterior y a diferencia de dicho prototipo una evaluación general de usabilidad nuevamente como en el prototipo 2.

3.5. CONSIDERACIONES FINALES.

- En AR para la IoT, las representaciones virtuales correspondientes a los objetos del mundo real se diseñaran de acuerdo a las necesidades requeridas, si existieran objetos que no son de interés, se pueden obviar, las interfaces que realizan las transiciones entre tareas se diseñan en principio para probar su funcionalidad, luego en iteraciones subsecuentes se entraría a mejorar su usabilidad a partir de los elementos disponibles como los colores, las animaciones, los modelos en 3D y demás.
- El uso de AR con marcadores (*Marker AR*) o sin ellos (*Markerless AR*) depende del grado de conocimiento de las herramientas a implementar por parte del desarrollador, los recursos disponibles y el contexto.
- La ubicación en interiores (*indoor*) es un problema que, aunque tiene algunas soluciones propuestas a nivel de proximidad, aún no tiene un desarrollo deseable respecto a la localización, ni las herramientas a nivel software son lo suficientemente robustas o accesibles.
- Deberán observarse las condiciones del entorno, como iluminación, ruido, y demás para que de acuerdo a estas se pueda seleccionar las herramientas a utilizar de acuerdo a los recursos disponibles.
- Los niveles de ergonomía de la aplicación varían según los dispositivos que se utilicen para la implementación de la aplicación, y el sistema operativo utilizado.
- La personalización, seguridad y posicionamiento en el mercado de la solución no se incluirán como parte del proceso de desarrollo de la interfaz gráfica.
- La enactividad como característica del proceso de desarrollo prima sobre la metodología o modelo escogido.



4. INTERFAZ DE REALIDAD AUMENTADA PARA LA INTERNET DE LAS COSAS

4.1. INTRODUCCION Y APORTES.

La interfaz que aquí se propone, viene soportada por el marco de referencia propuesto en el capítulo 2 del presente documento, donde se exploraron una serie de referencias y trabajos que se toman como estado del arte, donde sobresalen el trabajo de Valentin Heun [58] llamado Reality Editor, y el trabajo de Luis Garduno Massieu [52] llamado Gallag Strip.

Con respecto a Reality Editor, el aporte diferenciador de esta propuesta radica en la naturaleza del macroproyecto de Niño Zambrano [11] llamado Interacción semántica de objetos en la web de las cosas, donde la arquitectura propuesta resuelve problemas de heterogeneidad de los objetos inteligentes, y conectividad, mientras que Reality Editor se limita a un servidor propio y una serie de escenarios predefinidos, además que Reality Editor hace un mapeo de funcionalidades mientras que la interfaz propuesta permite definir interacciones entre objetos inteligentes.

El desarrollo de Garduno Massieu permite grabar interacciones entre objetos, pero al igual que Reality editor lo hacen sobre escenarios predefinidos con objetos rigurosamente definidos, mientras que con el uso de ontologías e índices semánticos la aplicación de AR es escalable permitiendo ubicar objetos del propio escenario y de escenarios distantes alrededor del mundo siempre y cuando estén indexados, esto brinda la posibilidad de asignar comportamientos de evento o acción a los recursos de cada objeto inteligente, permitiéndole a la interfaz de AR aprovechar esta funcionalidad para definir interacciones semánticas y contextualizadas de manera más transparente e intuitiva.

4.2. METODOLOGIA

A partir de la guía de desarrollo propuesta en el capítulo 3 de este documento, se siguen las fases propuestas y se adaptan los elementos sugeridos para la creación de prototipos funcionales cada vez más refinados.

4.3. ANALISIS.

4.3.1. Contextualización.

- **Contexto.**

Este proyecto está enmarcado como subproyecto de la tesis de doctorado del PhD Miguel Ángel Niño Zambrano [11], y soportado a nivel hardware sobre el escenario de interacción semántica para la IoT propuesto por Santiago Guerrero y Dalila Riobamba [21], que a su vez



Interfaz de Realidad Aumentada Para la Internet de las Cosas.

referencian los grupos de investigación de Tecnología en Telemática⁴¹ e I+D en Tecnologías de la Información⁴² de la Universidad del Cauca⁴³, el usuario objetivo es quien hace uso de la oficina donde se despliega el escenario, pero puede ser cualquier persona que posea un dispositivo móvil que permita desplegar la aplicación AR.

El enfoque que se tiene para este proyecto es el de crear una aplicación que le permita al usuario acceder a los servicios que se generan debido a la interacción entre OI, en un escenario definido, junto con su gestión. Este enfoque debe tener como base los objetivos del proyecto propuesto establecidos en la sección 1.4 del capítulo 1 del presente documento.

Se debe considerar que el escenario de interacción semántica es una prueba de concepto y no está destinada a usuarios finales, así la aplicación propuesta tampoco está diseñada para un usuario final.

- **Arquitectura.**

El trabajo de Niño – Zambrano [11], establece una arquitectura en 6 capas (ver Figura 27), cuyo objetivo es resolver la heterogeneidad y la conectividad entre las cosas de la IoT. La capa de objetos es donde los recursos IoT suben sus datos a internet, en la capa de middleware se encuentran las distintas plataformas que reciben estos datos.

La capa de objeto semántico centraliza toda esta información de distintas fuentes hacia un solo servicio, a partir de un análisis ontológico en un dominio o dominios definidos, esto con el fin de indexar la información respecto a dicho dominio, así se logran datos homogéneos y contextualizados de distintas fuentes en un servicio común. Esta capa también crea una representación digital de cada recurso IoT indexado, llamada objeto semántico, a través de la ontología de objeto semántico con el fin de almacenar datos, metadatos y servicios de los objetos inteligentes para que puedan ser utilizados sin importar su fuente.

Las capas de red social, servicios y aplicación resumen las funcionalidades de un escenario IoT de tipo semántico, donde interactúan objetos entre sí, objetos con personas y los servicios que consumen las aplicaciones.

⁴¹ http://web.unicauca.edu.co/telematica/?q=telematica/ver_info_grup_inv&grupo_inv=1

⁴² <http://gti.unicauca.edu.co:81/gti/>

⁴³ <http://www.unicauca.edu.co/>

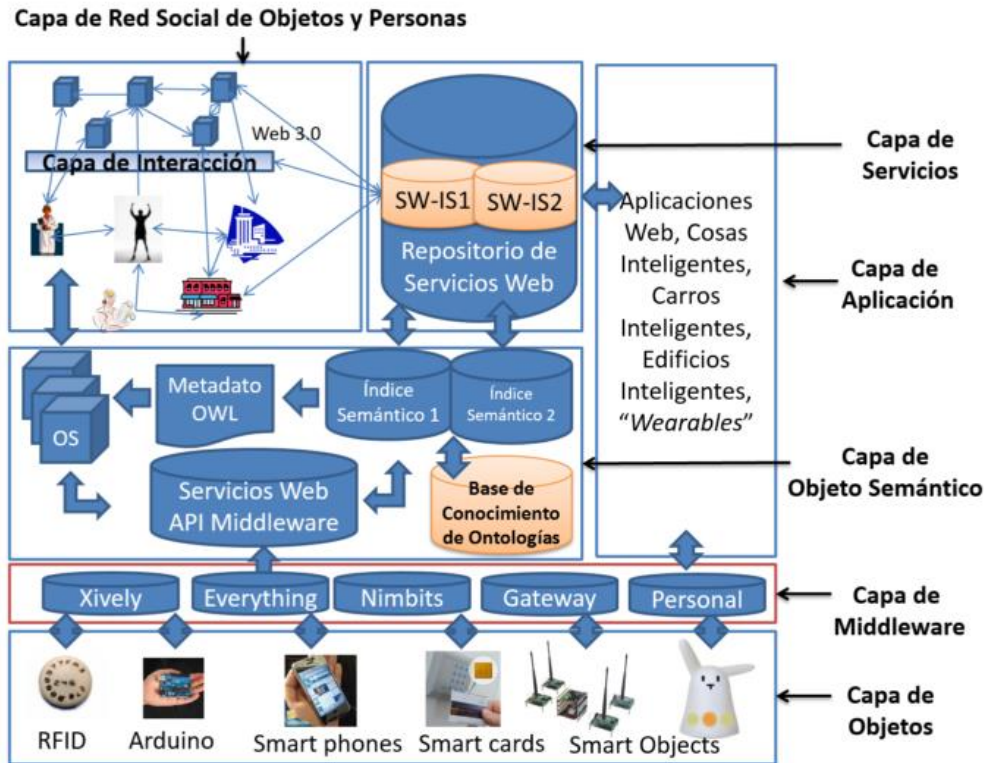


Figura 27. Vista estática de la arquitectura de interacción semántica.

Fuente: N. Zambrano [11], Interacción semántica de objetos en la web de las cosas.

De manera específica las aplicaciones consumen un servicio web de acceso simple SOAP (*Simple Object Access Protocol*) donde pueden consultar los recursos asociados al contexto, una gran cantidad de metadatos y datos provenientes de su respectivo middleware, realizar consultas expandidas y otras funcionalidades. La consulta directa a la representación digital llamada objeto semántico se hace a través del protocolo de mensajería que define el escenario de interacción dependiendo de las necesidades del mismo.

La interfaz de AR propuesta además de visualizar la información de dichos objetos, gestiona las interacciones entre ellos bajo un contexto definido y los nuevos servicios como producto de su interacción.

La interfaz de AR para la IoT se ubica en tanto en la capa de interacción como en la de aplicación, consultando tanto al índice semántico para establecer dicha contextualización, como al objeto semántico para obtener datos y metadatos y gestionar las interacciones entre ellos.

- **Escenario IoT.**

La interfaz de AR para la IoT utiliza el escenario generado en el trabajo de Guerrero Riobamba [21], este desarrollo es una prueba de concepto y como tal la interfaz propuesta está montada sobre dicha prueba, mas no para un producto final.

Interfaz de Realidad Aumentada Para la Internet de las Cosas.

Se tiene un escenario de interacción semántica entre OI de la IoT, este escenario consta de 3 recursos IoT como se observa en la Figura 28, son OI definidos como:

- **Regulador de Humedad en planta:** 2 recursos, el sensor de humedad y el dispositivo de riego, emulado por un Diodo emisor de luz LED (*light emitting diode*).
- **Regulador de Luz:** 2 recursos, el sensor de luminosidad y la lámpara o bombillo.
- **Regulador de Temperatura:** 3 recursos, el sensor de temperatura, y los actuadores, calefactor emulado por una fuente de aire caliente y un ventilador.



Figura 28. Escenario de interacción semántica.

Estos recursos se encuentran implementados sobre una tarjeta de desarrollo diseñada para la IoT, su nombre es Intel Galileo Gen 1⁴⁴, cuya característica principal es contar con un microprocesador que soporta sistemas operativos orientados a la IoT y un micro controlador tipo Arduino⁴⁵, ambos integrados al mismo hardware, además de soportar comunicación vía Ethernet entre otras características que en conjunto definen los OI respectivos.

Para este caso en particular se utilizó el sistema operativo Yocto Linux⁴⁶ y el lenguaje de programación Python⁴⁷ para desplegar las funcionalidades del escenario, además del Grove starter kit V3 de Seeed Studio⁴⁸ para desplegar los recursos físicos.

4.3.2. Requerimientos.

⁴⁴ <http://www.intel.com/content/www/us/en/embedded/products/galileo/galileo-g1-product-brief.html>

⁴⁵ <https://www.arduino.cc/>

⁴⁶ <https://www.yoctoproject.org/>

⁴⁷ <https://www.python.org/>

⁴⁸ http://www.seeedstudio.com/wiki/Grove_-_Starter_Kit_v3



- **Definición del Roles.**

En este contexto y con el escenario IoT definido, se define el rol de cliente al PhD Miguel Ángel Niño Zambrano, cuyas funciones son generar los requerimientos de la aplicación, discutir los avances, y generar la retroalimentación a lo largo del proceso de desarrollo para validarlo.

Como desarrollador se tiene al estudiante Omar Eduardo Guzmán Álvarez, del programa de Ingeniería Electrónica y Telecomunicaciones de la Universidad del Cauca, cuyas funciones son llevar a cabo la captura de requerimientos, el desarrollo de los prototipos funcionales y usables para una interfaz intuitiva y fácil de usar, siguiendo las consideraciones que se tienen a lugar en los objetivos de este trabajo.

El rol de usuario de prueba para el prototipo 1 será asumido por el mismo cliente y para la evaluación de usabilidad del siguiente prototipo se tendrán algunos sujetos de prueba.

- **Captura de Requerimientos.**

Para la captura de requerimientos se escoge utilizar historias de usuario, estas buscan identificar el flujo de información a través de la interacción del usuario con el escenario definido, para encontrar un equilibrio entre usabilidad de la aplicación y simplicidad de desarrollo, ya que se desea desarrollar una interfaz de usuario en AR para la IoT que tenga en cuenta el cliente y le brinde una buena experiencia de usuario. Cada historia es relevante en su escena asignada, las fichas de historia de usuario se encuentran en el anexo A del presente documento, a continuación se lista cada historia en general, los requerimientos obtenidos se muestran a lo largo del desarrollo de las escenas asignadas.

1. **Historia de usuario 1. Contextualización:** El usuario desea poder reconocer el escenario o contexto asociado a través de la aplicación mediante su dispositivo móvil, además de los recursos IoT disponibles.
2. **Historia de usuario 2. Consultar Datos:** El usuario desea visualizar los metadatos del objeto inteligente cuando este se encuentre en su campo de visión lejano.
3. **Historia de usuario 3. Consultar OI:** El usuario debe poder obtener información acerca de las propiedades y características del OI cuando este se encuentre en su campo de visión cercano.
4. **Historia de usuario 4. Servicio básico:** El usuario debe poder activar o desactivar los servicios básicos de cada OI al tenerlo en mi campo de visión cercano.
5. **Historia de usuario 5. Servicios de interacción entre OI:** El usuario debe poder gestionar la interacción entre OI. Para esto define los eventos, las condiciones y las acciones que se deben ejecutar en el escenario. En el modelo definido los eventos corresponden a los sensores de un OI, las acciones corresponden a los actuadores de un OI y la condición la define el usuario a criterio propio, a esto se le ha llamado servicio de interacción o ECA (Evento-Condición-Acción).



Los recursos disponibles son: El escenario de interacción semántica, un pc de escritorio, dos teléfonos móviles. Como recursos humanos esta la asesoría del director de este proyecto y el codirector, más los recursos informáticos como acceso a repositorios de conocimiento facilitados por la Universidad del Cauca.

- **Asignación de tiempos.**

Este proyecto al ser implementado sobre un escenario IoT con una tecnología definida, requiere apropiación de dicho escenario, además de requerir tiempo para realizar la curva de aprendizaje de las tecnologías de AR y la exploración documental que dio origen al marco de referencia propuesto, lo cual consume recursos fuera del tiempo de desarrollo, diferentes a las que requiere la metodología y la implementación de la interfaz, esto se encuentra por fuera del contexto de un desarrollador de AR como tal, ya que este actor se presume tiene las capacidades y los conocimientos necesarios.

Se definen entonces 6 meses para la apropiación del escenario y las tecnologías a utilizar, además de la exploración documental, y 6 meses asignados al proyecto, donde el tiempo de apropiación y curva de aprendizaje es aproximadamente 2 meses, como tiempo de desarrollo se tienen 2 meses más, y es probable que se requiera algún tiempo extra para la realización de las pruebas y la documentación durante el mes sobrante.

4.4. DISEÑO.

4.4.1. Establecimiento de tecnologías.

- **Tecnologías establecidas por la arquitectura y el escenario IoT.**

El escenario de interacción semántica de Guerrero Riobamba [21] se fundamenta en la arquitectura propuesta por Niño-Zambrano [11], por lo tanto este desarrollo hereda ciertos elementos de ambos desarrollos para poder crear la interfaz de AR para la IoT.

En primer lugar, el middleware establecido se llama Xively⁴⁹, es el servicio donde se alojan los datos y metadatos de los recursos IoT, es la base del índice semántico asociado, para este caso se trata de un índice semántico en el contexto de domótica, pero cabe notar que puede ser cualquier contexto, la referencia del índice se hace inicialmente mediante la palabra asociada en los metadatos del recurso IoT en el middleware (ver Figura 29).

⁴⁹ <https://www.xively.com/>



Figura 29. Metadatos del middleware Xively.

El servicio web del índice de domótica se ubica en la siguiente dirección web:

<http://facfiet.unicauca.edu.co/SemanticSearchIoT/WSSemanticSearch/WSSemanticSearch.aspx>

Y la aplicación web de dicho índice se encuentra en:

<http://facfiet.unicauca.edu.co/SemanticSearchIoT/AppWeb/>

Es necesario establecer una convención acerca de las jerarquías de los nombres genéricos de cada objeto para no confundirlos durante la elaboración de los prototipos, así:

- **Recursos IoT:** Son Cualquier objeto que encuentra el índice al realizarle una consulta.
- **Objetos Inteligentes - OI:** hacen referencia a los 3 reguladores presentes en el escenario definido, contienen los recursos propios de cada regulador.
- **Recursos:** Son las entidades mínimas que componen cada regulador u OI, pueden ser actuadores o sensores.



Por otra parte, para los mensajes entre los OI se utiliza el protocolo MQTT⁵⁰, con los mensajes codificados bajo el estándar ODF-OMI. El intermediario *Broker* que soporta los mensajes es el de eclipse⁵¹.

Se puede concluir que bajo un escenario 100% funcional, tomaría alrededor de un mes para la apropiación de dicho escenario, su arquitectura base y demás tecnologías que implementa.

- **Tecnologías a utilizar para el despliegue de la interfaz AR.**

Se establece Android como el sistema operativo objetivo, ya que los dispositivos asignados vienen en ese sistema. Android es un sistema operativo basado en Linux⁵² de la empresa Google Inc., principal subsidiaria de Alphabet inc.⁵³, posee alta flexibilidad y licencia abierta, además de ser el referente para las librerías de desarrollo de la mayoría de tecnologías en AR junto con los sistemas operativos iOS de Apple⁵⁴, sus ventajas radican en el costo y la libertad de uso del software, a diferencia de iOS cuyos costos son altos a nivel de hardware y el software es cerrado solo a estos dispositivos.

Para el desarrollo se utiliza el sistema operativo más difundido y utilizado, Windows⁵⁵ soporta gran variedad de hardware y software, más aun para aplicaciones en AR, compatible con las herramientas de desarrollo como motores de videojuegos y las librerías especializadas, esto es lo que le brinda ventaja al desarrollador al poder crear y probar sus aplicaciones rápidamente, a través de dispositivos hardware como cámaras, interfaces de red, periféricos y demás elementos necesarios como los que exige una aplicación en AR para la IoT.

Como primera opción, como se observó en el capítulo anterior, Unity 3D y PTC Vuforia son los recursos software más utilizados para aplicaciones en AR.

Unity 3D es la herramienta por excelencia para este tipo de soluciones, el flujo de trabajo de este motor de videojuegos es muy intuitivo, basado en elementos llamados *GameObjects*, junto con componentes como los scripts y funcionalidades propias⁵⁶, este conjunto de herramientas permiten crear aplicaciones de una manera simple y rápida, además de brindar múltiples alternativas en cuanto a plataformas objetivo, los desarrollos no se limitan a una sola tecnología, y el motor permite probarlos directamente en los dispositivos objetivos como en este caso, uno con sistema operativo Android, sin la necesidad de utilizar entornos de desarrollo nativos.

La plataforma PTC Vuforia brinda soporte para iOS y Android a través de sus SDK's, además de ejemplos, funcionalidades exclusivas como terreno inteligente (*Smart Terrain*), Seguimiento extendido y una plataforma de gestión de marcadores y licencias, aunque sus lanzamientos de versiones no son frecuentes, si se evidencian cambios sustanciales de versión en versión, por

⁵⁰ <http://mqtt.org/>

⁵¹ lot.eclipse.org

⁵² <https://www.debian.org/releases/stable/mips/ch01s02.html.es>

⁵³ <https://abc.xyz/>

⁵⁴ <https://www.apple.com/>

⁵⁵ <https://www.microsoft.com/es-co/windows/>

⁵⁶ <https://unity3d.com/es/learn/tutorials/topics/interface-essentials/game-objects-and-components>



esto es conveniente tener respaldos de los desarrollos según la versión específica y tener cuidado especial al actualizar el SDK.

Como consideración final, a medida que se desarrollan las historias de usuario, el SDK de AR puede que requiera cambiarse en una o varias escenas, todo depende de las necesidades que puedan surgir durante el proceso, por esto se han considerado y probado algunas otras opciones, y la justificación del porque no se utilizaron:

- **Wikitude.**

Este SDK posee un ecosistema en crecimiento, lo cual le da un respaldo confiable, está basado en localización y visión de computador, diferenciándolo de otros motores para AR, además de implementar para enero de 2016 la tecnología de Visual SLAM⁵⁷. También es muy interesante la funcionalidad de navegación en interiores la cual puede ser relevante para el proyecto que se está desarrollando. Sin embargo, la complejidad para ejecutarlo en Windows es alta ya que no se puede hacer prueba en el entorno sino que se debe mudar el proyecto hacia Android Studio o Xcode en IOS.

- **Kudan.**

Es el SDK más prometedor a la fecha, ofreciendo SDK para IOS, Android y un paquete para Unity3D, a enero de 2016 ha lanzado su tecnología de AR sin marcadores tipo SLAM⁵⁸. Los tutoriales para sus características más interesantes como el Motion Tracking y la AR sin marcadores están solamente disponibles para IOS y en Android no funciona ni se da correcto soporte a los bugs encontrados, lo cual es una gran limitante a la hora de desarrollar, en conclusión, es un SDK que aún se encuentra muy joven y con muchos aspectos por corregir en soporte, lo cual no permite evaluar completamente sus capacidades como herramienta para la creación de AR.

- **Google Tango Project.**

Esta tecnología, aunque tiene aspectos funcionales muy útiles para una aplicación en AR, que se pueden ejecutar en un motor de desarrollo como Unity 3D, para dispositivos móviles requiere un hardware especial con cierto número de cámaras y sensores que aún no son muy accesibles en el mercado, los demás dispositivos aparecen como no compatibles y no permiten ejecutar las aplicaciones ni instalar las asociadas al desarrollo.

4.4.2. Plan de prototipos.

Según el estudio del escenario de interacción semántica, se definen dos prototipos, el prototipo 1 resuelve las funcionalidades básicas del escenario definido implementando AR donde es posible y el prototipo 2 refina la interfaz de AR obtenida. Los servicios que brinda el escenario de interacción semántica son:

- Consultar los datos y metadatos de un recurso en los OI del escenario.

⁵⁷ <http://www.wikitude.com/>

⁵⁸ <https://www.kudan.eu/features/>



Interfaz de Realidad Aumentada Para la Internet de las Cosas.

- Cambiar el estado de un servicio básico asociado al OI.
- Cambiar el estado de un servicio de interacción ECA.
- Borrar un servicio de interacción ECA.
- Crear un servicio de interacción ECA.
- Reiniciar el escenario.
- Responder a las peticiones.

Se asignan en principio 5 escenas, una por historia de usuario, las escenas se van asignando a medida que se avanza en el desarrollo, procurando un equilibrio que le brinde intuición a la interfaz.

4.5. CONSTRUCCION.

4.5.1. Solución general.

Como visión general se requiere una interfaz que preste asistencia para realizar consultas sea al índice objetivo o al escenario de interacción, muestre en tiempo real los datos de los OI, y permita la gestión y creación de servicios de interacción.

- **Interfaces físicas.**

Se tienen como recursos para el desarrollo un pc de escritorio con una webcam, esto permite probar rápidamente la interfaz, sin embargo se debe instalar la versión de Unity3D de 32 bits, ya que esta es compatible con la cámara web asignada.

Para el despliegue se deben utilizar los dispositivos móviles en el modo de desarrollador, esto se consigue en Android presionando siete veces sobre el número de compilación en la configuración del sistema, luego seleccionando la opción de depuración USB y permitiendo aplicaciones de origen desconocido. En Unity 3D se puede seleccionar el dispositivo una vez instalado, siguiendo los tutoriales para desarrolladores⁵⁹.

El límite colocado al compilar la aplicación para el S.O Android, es la versión Android 4.0 API Nivel 14, este límite es el mínimo que deberá tener el dispositivo para poder ejecutar la aplicación, sin embargo son varios los problemas conocidos que se halla en los foros de desarrolladores acerca de este tema.

Así la técnica de visualización será video indirecto, donde el usuario sostendrá el dispositivo móvil con una mano y con la otra hará la interacción mediante el panel táctil (*touch panel*).

- **Comunicación con el escenario.**

El escenario tiene como sistema de comunicación, mensajes MQTT, en un formato definido que es ODF-OMI del Open Group, por lo tanto se hará necesario instalar en Unity 3D una

⁵⁹ <https://docs.unity3d.com/es/current/Manual/android-GettingStarted.html>



librería o alguna herramienta que permita dicha comunicación, además de alguna otra u desarrollo que permita leer los mensajes , ya que estos están basados en el formato XML.

Se encontró que el escenario no brinda acceso al número de servicios de interacción ni a sus archivos específicos de cada uno, llamados ECA, y si se quiere una funcionalidad dinámica en este sentido, se hace necesario crear una solución a este requerimiento aunque no haga parte de los objetivos de esta propuesta.

- **Clasificación de los servicios**

El establecimiento general del contexto y los recursos IoT hallados en el mismo, requieren que se consulte el índice asociado, se hace necesario apropiarse del funcionamiento del índice y su implementación en Unity 3D.

El servicio de creación de servicios de interacción ECA, tiene una alta complejidad, y será desarrollado como una combinación entre una interfaz clásica con unos pasos a seguir, y algunas funcionalidades en AR.

4.6. PROTOTIPO1.

El HUD implementado consta de un asistente como aviso de texto en la parte superior de la pantalla, y un menú de botones para navegar entre las escenas y salir, anclado al extremo derecho de la misma. Según como avance el desarrollo en cada escena el HUD ira cambiando.

Las representaciones de los recursos y el despliegue de los servicios se va dando a medida que se asignan las escenas a cada historia de usuario, en algunos casos, se asignaran dos historias de usuario a una sola escena o dos escenas a una sola historia de usuario según como fluya la construcción de las mismas.

4.6.1. Solución para la historia de usuario 1.

Como primera medida se establecieron los requerimientos prioritarios con base en el análisis de la historia de usuario 1 los cuales fueron:

- Identificar el escenario.
- Contextualizar.
- Detectar la cantidad de recursos IoT disponibles.

La identificación del escenario se realiza utilizando un dispositivo móvil y capturando la consulta al índice, las modalidades de consulta serán dos, el modo clásico de consulta mediante texto y botones, y aprovechando los sensores del dispositivo móvil, se hace la consulta de otra manera.

Para la contextualización se utiliza el índice semántico asociado a una ontología específica, de tal manera que al realizar una consulta al índice con unos parámetros de contexto definidos, la ontología retorne los objetos asociados a la consulta.

A continuación, se presenta un análisis de las tecnologías actuales que se pueden utilizar para identificar elementos de contexto como la EI; para esto se realizaron pruebas en el motor de



desarrollo Unity 3D, en conjunto con la librería PTC Vuforia, teniendo en cuenta las características de cada tecnología se obtuvo la Tabla 5.

TECNOLOGIA DE IDENTIFICACION	VENTAJAS	DESVENTAJAS
Tags NFC Y RFID	-Simplicidad -Economía -Estandarizado	-Integración compleja con el motor de desarrollo -NFC requiere cercanía con el dispositivo móvil. -Necesidad de un experto para programarlo o una interfaz especializada. -Depende de la antena del dispositivo y el alcance.
Códigos QR	-Economía -Fácil creación -Desechable -Integración media con el motor de desarrollo. -Fácil detección en el dispositivo móvil. -Personalizable.	-Depende de las condiciones de iluminación y posición del dispositivo de captura. -Poco natural para el usuario. -Inseguro -Depende de un marcador
Identificación de entorno por visión de computadora.	-Naturalidad para el usuario. -Sin marcadores	-Hardware complejo tanto en detección como en uso. -Alto costo -Necesidad de un experto para vincular el entorno a un identificador.
GPS	-Proporciona localización -Fácil integración con el motor de desarrollo. -Hardware integrado en la mayoría de Smartphones.	-Resolución en metros. -No funciona bien en interiores. -Alto consumo de energía
BLE - Bluetooth 4.0 Low energy	-Permite calcular proximidad. -La señal tiene un alcance aceptable. -Es económico	-Poco estandarizado -Integración compleja con el motor de desarrollo. -Tecnología emergente

Tabla 5. Características de tecnologías de identificación.

Se puede concluir que para el primer prototipo la solución más accesible y rápida para la identificación del escenario es la utilización de un código QR que asocie la palabra con la que se hace la consulta al índice semántico por economía e integración con el software de desarrollo. A continuación se describe el desarrollo de la escena 1 y 2, las cuales se asignan a la historia de usuario 1:

- **Escena 1. Contexto.**



El escenario de interacción tiene asociada la palabra “Sala” de una casa para su identificación, así mismo se asocia este término al código QR respectivo en Vuforia, ya que es una de las palabras definidas en los metadatos del middleware Xively para cada OI.

El proceso de creación de las escenas en Unity 3D tiene su base fundamental en la creación e implementación de marcadores con Vuforia, descrito en el Anexo B del presente documento, la implementación del servicio web se encuentra en el Anexo C y los códigos de los scripts en C#, utilizados en este desarrollo se encuentran en el Anexo D.

El marcador utilizado para la palabra “Sala” fue generado en la web:

<http://www.qrcode.es/es/generador-qr-code/>

y el código o marcador QR obtenido se observa en la Figura 30.



Figura 30. Código QR “Sala”.

Esta palabra se envía a una base de datos interna de Unity 3D junto con los datos del GPS utilizando el script *Tracking.cs*, esto con el fin de pasarle estos datos a la siguiente escena, para la acción de contextualizar; este mismo procedimiento se sigue si el usuario realiza una consulta mediante el método clásico de ingresar texto, asistido por el HUD implementado. (ver Figura 31).

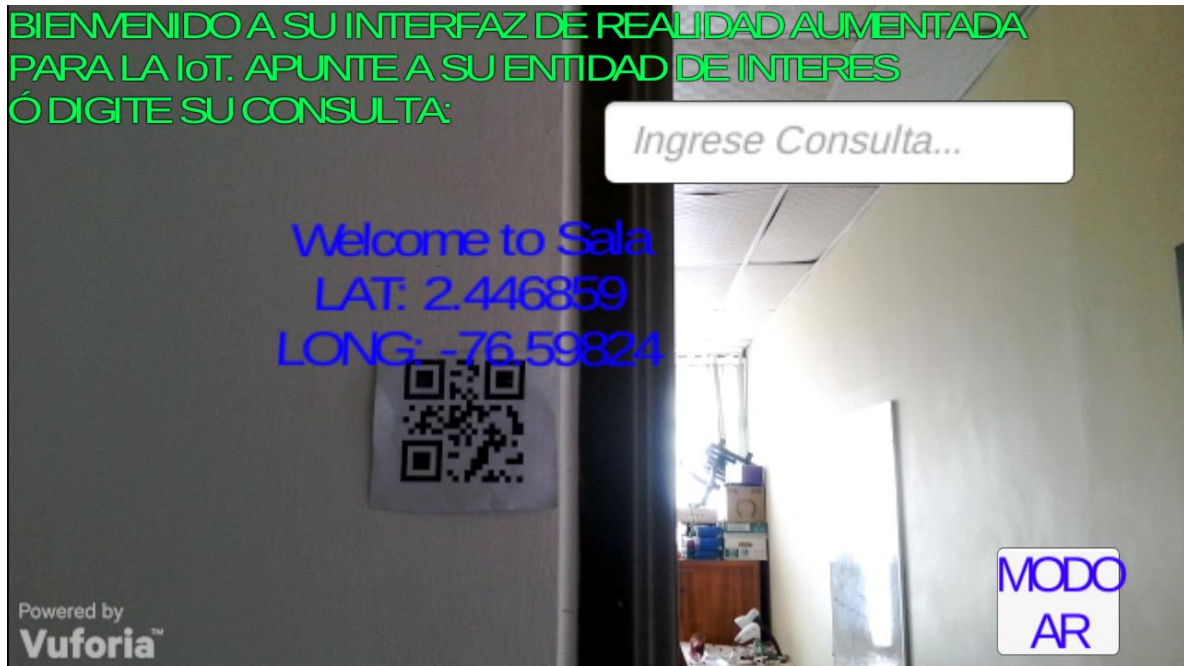


Figura 31. Identificación del escenario – escena 1. Fuente Propia.

Estas acciones dependen del tipo de índice con que se haga la consulta y de la ubicación del usuario al momento de consultar los datos GPS; aunque en este prototipo se utilizó un índice para domótica, se pueden cargar servicios web de diferentes dominios mediante el procedimiento descrito en el Anexo C de presente documento.

La tarea de identificación sucede cuando el usuario abandona el código QR de su campo de visión, la aplicación cambia automáticamente a la siguiente escena para realizar la consulta respectiva al servicio web asociado al índice semántico, en este caso mediante la palabra “Sala” del código QR.

La localización está vinculada con la ubicación del marcador QR de la EI, que se ha establecido como una ubicación fija, sin embargo, si se cambia de sitio el marcador asociado, simplemente cambiaran las coordenadas GPS, pero el contexto sigue siendo el mismo a excepción de que se realice una consulta escrita y se cambie la palabra a consultar. Cabe notar que el hecho de poder mover el marcador de la EI o de los PI va en contravía con el concepto de AR, por lo tanto, se hace necesario vincular la identificación del escenario y de los OI de manera fija.

- **Escena 2. Consulta.**

La aplicación instancia el servicio del índice a través de la referencia web creada en el entorno de desarrollo, utilizando el script *Consulta.cs*, una vez el índice obtiene la palabra almacenada (por ejemplo “sala”) se utilizan las funciones del servicio web para consulta simple y expandida (ver Figura 32.)



WSSemanticSearch

Las operaciones siguientes son compatibles. Para una definición formal, revise la [descripción de servicios](#).

- [AsignarConfiguracion](#)
- ⇒ [Buscar](#)
- [BuscarFeeds](#)
- [BuscarFeedsDataPoints](#)
- [CarqarFeedXivelyBDD](#)
- [CarqarOntologia](#)
- [CrearIndiceSemantico](#)
- ⇒ [ExpandirConsultaConceptosOntologia](#)
- [GeoNames_Hierarchy](#)
- [ObtenerCiudadesConsulta](#)
- [ObtenerConfiguracion](#)
- [ObtenerListaLugarCoordenadas](#)
- [ObtenerListaLugarCoordenadasJerarquica](#)
- [ObtenerListaLugarCoordenadas_AM1](#)
- [ObtenerLocalizacion](#)
- [ObtenerLugardeCoordenadas](#)
- [ObtenerLugardeCoordenadasJerarquia](#)
- [RetornarConceptos](#)
- [RetornarDatapointsFeed](#)
- [RetornarDatosSensor](#)
- [RetornarJsonSensor](#)
- ⇒ [RetornarMapaLugar](#)
- [RetornarMapaLugarDatapoints](#)

Figura 32. Servicio web índice de domótica. Fuente Propia.

La consulta al índice a través de la EI “Sala”, brinda unos resultados generales y globales, así nace la necesidad de establecer el contexto jerárquicamente, pues aunque en este prototipo no se consideran los aspectos de seguridad y personalización, la jerarquía de contexto define y delimita el dominio de trabajo, junto con la EI y el PI a través de la consulta al índice semántico utilizado y la localización, marca o huella del usuario al desplazarse (*fingerprint*) la cual es intrínseca para este caso.

La ontología y el índice de domótica utilizados brindan la posibilidad de realizar consultas combinadas respecto a la EI, y la localización enriqueciendo el contexto de manera transparente, haciendo uso de las coordenadas GPS, y estableciendo un radio de búsqueda en kilómetros, además del idioma de consulta, así pues la funcionalidad de contextualización en la interfaz se establece en la consulta de manera expandida.

También se pueden asociar más palabras y definirlas en cada marcador mediante la plataforma de desarrolladores de Vuforia, puede implementarse la funcionalidad para que una interfaz multimodal pueda redefinir el contexto con algún otro tipo de búsqueda o se personalice el contexto en relación al usuario final. En este sentido Vuforia brinda la posibilidad de reconocer texto, imágenes, Objetos 3D, y el S.O Android permite entradas como comandos de voz, texto y datos GPS entre otros, pudiendo aumentar o disminuir la complejidad de la búsqueda a gusto del cliente en el proceso de desarrollo.

Se hace necesario brindar todas estas alternativas al usuario a través de elementos tipo interfaz de usuario UI, para este caso desde la anterior escena se obtienen los datos referentes a identificación del escenario y localización, ambos idiomas de consulta: inglés y español, para el



radio de acción se implementa un deslizador o *slider*, para representar los recursos encontrados a la vista del usuario, siguiendo con la filosofía de la AR, las representaciones o modelos 3D deberán representar los elementos encontrados con algún dato o metadato relacionado y algún tipo de interacción con el usuario.

Los datos para instanciar las representaciones graficas necesarias se obtienen de la respuesta del servicio web, la cual es una matriz de datos que son estáticos y solo se actualizan mediante una función del mismo servicio. Unity 3D le permite al desarrollador tener una vista completa de dicha información y tomar decisiones de diseño en tiempo de ejecución como se observa en la Figura 33. Para el prototipo 1 se tiene en amarillo el número de objetos encontrados y el nombre de cada uno, en rojo el numero identificador *Feed* de cada recurso IoT encontrado, y en azul el número de recursos que tiene dicho objeto con su respectivo nombre si los hay.

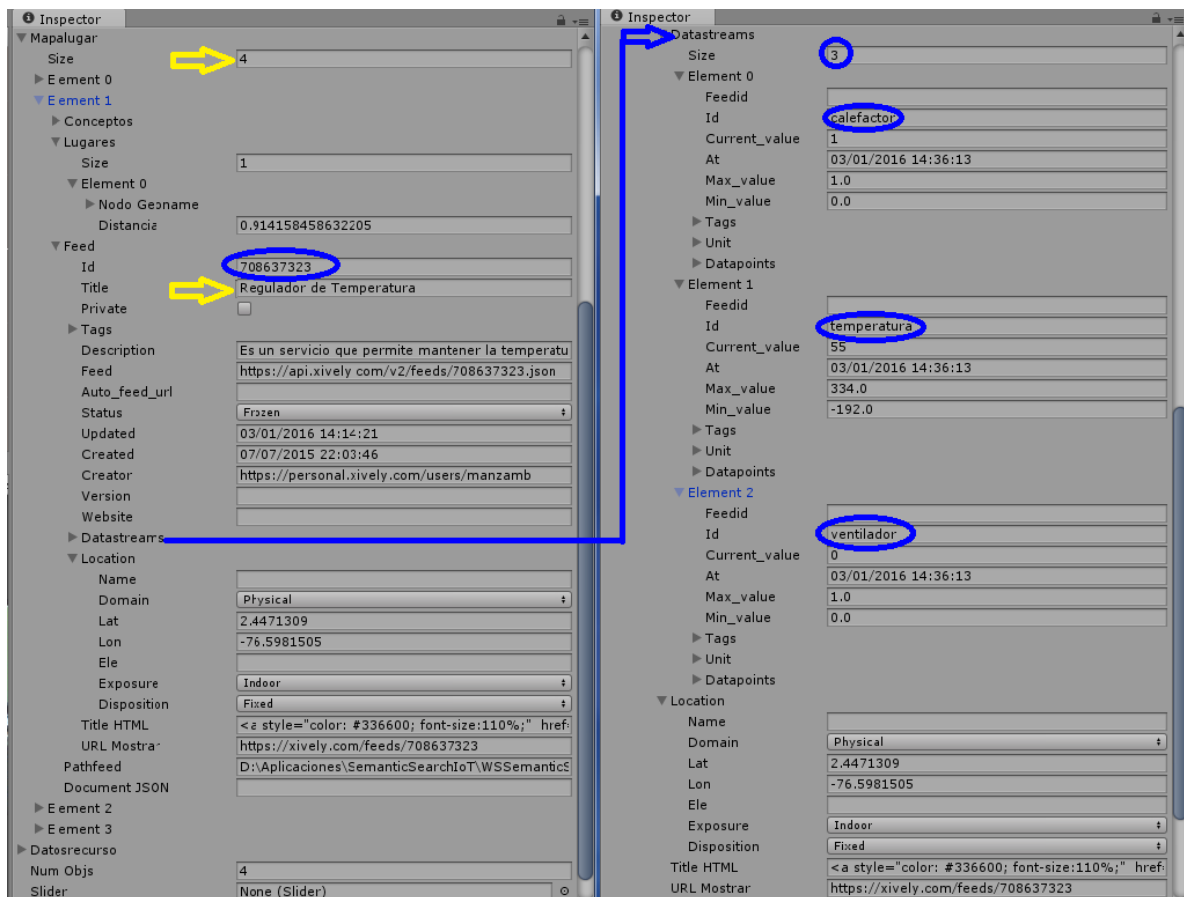


Figura 33. Matriz de datos obtenida del servicio web. Fuente Propia.

En este punto se tiene una lista de recursos IoT ordenados por relevancia a la consulta y algunos de ellos están relacionados al contexto definido, estos son los tres OI definidos en el trabajo de Guerrero Riobamba [21], dando cumplimiento al tercer requerimiento de la historia de usuario número uno.



4.6.2. Solución para la historia de usuario 2.

Los requerimientos que se establecieron como prioritarios a partir del análisis de la historia de usuario numero 2 fueron:

- Obtener los metadatos y datos básicos de cada objeto.
- Establecer comunicación con los OI.
- Desplegar los metadatos básicos con un toque al cursor del objeto.

El servicio web entrega gran cantidad de datos, entre ellos los metadatos del objeto consultado, su localización, datos históricos y sus recursos disponibles, esto lo hace en formatos como JSON o XML si es requerido, por lo tanto se puede desplegar tanta información como se requiera, sin embargo para el prototipo actual se debe respetar la filosofía de no saturar al usuario con demasiada información, por esta razón solo se despliegan ciertos metadatos básicos.

Para un despliegue dinámico de la información, no basta con los datos que brinda el servicio web ya que estos son estáticos⁶⁰, para obtener datos en tiempo real se hace necesario establecer comunicación con el OI el cual combina los datos obtenidos del índice, junto con sus propios datos. Dentro de la misma escena 2 y para las escenas siguientes, la comunicación con los OI se realiza a través del bróker MQTT definido para el escenario de interacción semántica, la implementación de las librerías MQTT para Unity 3D se puede observar en el Anexo E del presente documento. El formato de los mensajes MQTT utilizados en este escenario, su dinámica y funcionamiento se encuentran en el Anexo F del presente documento.

- **Consulta de los datos y metadatos.**

El funcionamiento de las peticiones es transparente al usuario, los OI se encuentran siempre suscritos a un canal MQTT llamado coordinador, asociado a su respectivo código identificador o *Feed* para escuchar las peticiones tipo XML. Este identificador se obtiene de la consulta al índice, junto con los demás metadatos a través de la matriz de datos *mapalugar[x]* del script *Consulta.cs*, el metadato *Feed.id* es el más importante para el funcionamiento de la interfaz.

Cabe notar que para el escenario de interacción semántica, los identificadores *Feed* de los OI se conocen con antelación al ser un escenario previamente definido, pero la consulta en el servicio web permite obtener el identificador de cualquier Recurso IoT indexado en el rango de consulta, por esta razón se pueden diferenciar los Recursos IoT y los OI.

Para obtener los demás metadatos de los recursos IoT también se puede utilizar la matriz de datos obtenida en el servicio web, o se puede hacer la solicitud de los mismos al Recurso IoT a través de los mensajes MQTT si pertenece al escenario, esto también permite discriminar si el Recurso IoT es un OI del escenario de interacción semántica definido dependiendo si hay respuesta o no a las peticiones realizadas al mismo.

En un OI la consulta de los metadatos se hace a través de la petición *MetaDataQuery.xml* en el canal coordinador con el identificador *Feed* obtenido en el índice, la respuesta *MetaData.xml* se recibe suscribiéndose al canal asociado, todo mediante el script *Metadato.cs*.

⁶⁰ El índice semántico se actualiza cada vez que el administrador del mismo realice su re indexación.

La lectura de las respuesta XML, se realiza mediante los scripts creados por Fraser McCormick⁶¹, los cuales se copian en la carpeta *Assets* del directorio raíz del proyecto en Unity 3D como se observa en la Figura 34, luego se implementan sus clases en cada respuesta XML obtenida, los archivos para la lectura de los archivos tipo XML son: *XMLNode.cs*, *XMLNodeList.cs* y *XMLParser.cs*.

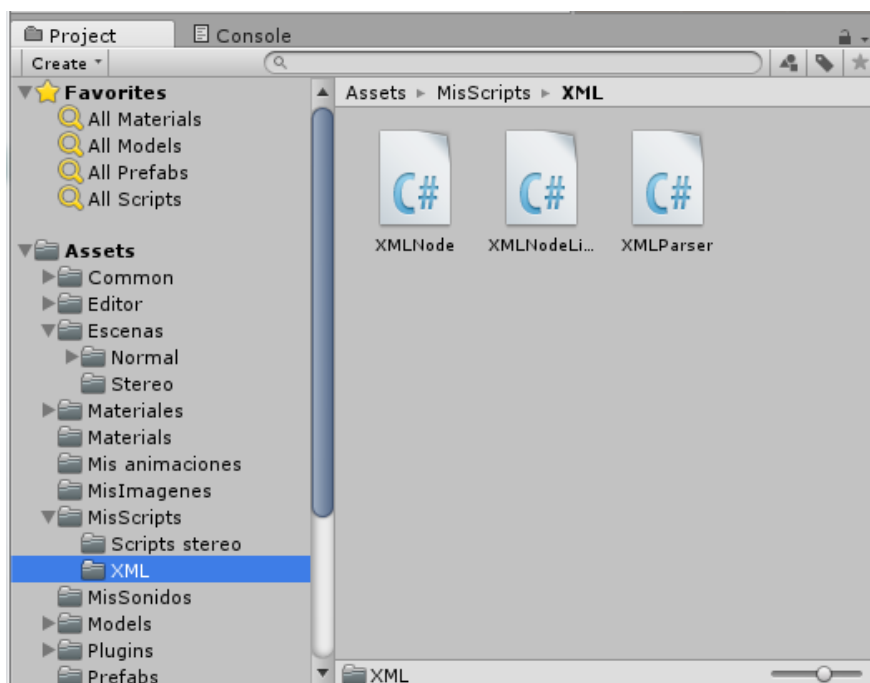


Figura 34. Ubicación de los scripts para la lectura de las respuestas XML.

Los OI siempre se encuentran publicando sus datos en el canal MQTT *Feed_Objeto/Nombre_recurso*, estos datos son los valores actuales de los sensores y el estado actual de los actuadores; para obtenerlos, la interfaz se suscribe a dicho canal utilizando el identificador *Feed* del OI y el nombre del recurso asociado, los cuales se pueden obtener tanto desde el índice como del OI como se vio anteriormente. La suscripción a dicho canal y la respuesta obtenida del OI se realiza a través del script *Feed.cs*, la respuesta viene dada en el formato *SimpleValue.xml*.

- **Representación Gráfica.**

Las representaciones graficas de los OI de esta escena se generan instanciando un prefabricado, esta es una funcionalidad de Unity 3D para crear o agrupar objetos personalizados con componentes y lógica propia en un solo objeto llamado *prefab*⁶², con el fin de reutilizar, ahorrar en recursos y código. El prefabricado utilizado tiene como nombre OIX y su creación y funcionalidad se describe en el Anexo I de este documento.

⁶¹ <http://dev.grumpyferret.com/unity/>

⁶² <https://docs.unity3d.com/es/current/Manual/Prefabs.html>

OIX es instanciado mediante el script *Consulta.cs*, posee etiquetas de texto que a través de la matriz *mapalugar[x]* o mediante los scripts *Feed.cs* y *Metadato.cs* brindan la información requerida y una implementación de un zoom semántico simple a través del script *Zoom.cs* permitiendo observar más metadatos del recurso IoT, al hacer touch o click en su representación gráfica, los objetos que no contienen datos o no pertenecen al escenario, se muestran en rojo.

Así se tiene una representación gráfica de los recursos IoT encontrados en la consulta enriqueciendo el entorno real como se observa en la Figura 35, en el HUD se indica el contexto, el número de recursos encontrados, la barra tipo *Slider*, para establecer el radio de la consulta y los botones para navegar en la aplicación.

Los datos han sido representados mediante el *GameObject* llamado *Text 3D* de Unity 3D, el dato estático que se refiere al nombre del primer recurso encontrado proviene de la consulta al índice, y el dato dinámico proviene del Recurso IoT, el cual deberá ser un OI del escenario de interacción semántica.



Figura 35. Datos y metadatos en la escena 2.

4.6.3. Solución para la historia de usuario 3.

Como requerimientos prioritarios se tienen:

- Obtener y permitir la visualización de los metadatos y datos básicos de cada objeto en el campo cercano de visión del usuario.
- Establecer comunicación con los OI.

Al tener un escenario definido, se toma como base el desarrollo de la escena 2 acudiendo a la comunicación con el OI, con la diferencia que para agilizar la respuesta entre el OI y el objeto

coordinador o dispositivo móvil se establecen fijamente los metadatos básicos como son nombre del recurso e identificador *Feed*.

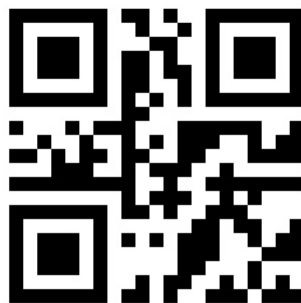
Algunos servicios son intrínsecos y transparentes al usuario tal como están definidos en el trabajo de Guerrero Riobamba [21], pero se pueden implementar algunos otros, como por ejemplo un historial de los datos de los recursos o definir ciertas acciones ante un evento utilizando los mismos recursos del OI sin intervención del usuario, esto se conoce como servicio básico.

- **Escena 3: Modo AR.**

En este punto es posible distinguir entre OI y otros objetos no asociados al escenario definido, los OI han sido discriminados mediante un único código QR (ver Figura 36), así mismo se les ha asociado un *Feed* en la arquitectura utilizada, lo cual es suficiente para consultar los datos y metadatos de cada uno como se observó en la escena anterior.



Regulador de temperatura.
Feed: 1931642039



Regulador de Humedad.
Feed: 78091938



Regulador de Temperatura.
Feed: 708637323

Figura 36. Códigos QR de cada OI.

- **Detección del marcador.**

Mediante el SDK de Vuforia para Unity 3D se crean los marcadores necesarios para que a través de la cámara del dispositivo móvil, la aplicación los detecte y relacione con los OI asociados. Este procedimiento se describe en el Anexo D del presente documento.

- **Representación gráfica.**

Los datos y metadatos necesarios se solicitan al OI a través de los scripts *Feed.cs* y *Metadato.cs* de la misma manera que en la escena 2, al ser detectado el código QR asociado, para este prototipo se muestran los actuadores mediante un *GameObject* tipo esfera, el nombre y valor actual de los recursos tipo sensor se muestra mediante *GameObjects* tipo Texto 3D como se observa en la Figura 37, y los estados de los recursos actuadores de manera binaria con 1 y 0.



Figura 37. Escena 3 Datos y metadatos.

4.6.4. Solución Historia de usuario 4.

En el escenario de interacción semántica, un servicio básico de cada OI define un valor umbral fijo como evento del recurso sensor, donde se activan o se desactivan los recursos tipo actuador ante este evento como se observa en la [Tabla 6. Umbrales de los servicios básicos de los OI.](#)

Sensor	Umbral	Condición	Estado Actuador
Humedad	10%	mayor	Riego apagado
		menor	Riego encendido
Luz	10%	mayor	Bombillo apagado
		menor	Bombillo encendido
Temperatura	26 °C	mayor	Calefactor apagado
		menor	Calefactor encendido
	29 °C	mayor	Ventilador encendido
		menor	Ventilador apagado

Tabla 6. Umbrales de los servicios básicos de los OI.

El estado de estos servicios básicos se encuentra en los metadatos del OI, y debe cambiar a petición del usuario; para gestionar un servicio básico mediante al interfaz de AR, dentro de la misma escena 3, como primera medida se requiere conocer el estado de dicho servicio, esto se logra a través del script *Metacons.cs*, luego se hace la solicitud de cambiar dicho estado (encenderlo o apagarlo según sea el caso) al OI.

La solicitud de cambio de estado se realiza publicando el archivo *SetBasicState.xml* a través del script *ServicioBasico.cs* con sus respectivos parámetros en el canal coordinador, entre los cuales se encuentra el estado deseado del servicio básico, el OI publica la respuesta *SimpleResponse.xml*, obteniendo una cadena “200” si el estado del servicio básico fue cambiado con éxito, y “404” en caso contrario. En la Figura 38 se observa el cambio de color de un *GameObject* tipo cubo, el cual se puede seleccionar al hacer click o touch sobre el mismo para cambiar el estado del servicio básico.



Figura 38. Escena 3 Estado del servicio básico encendido.

4.6.4.1. Solución para Historia de usuario 5.

Analizando esta historia de usuario se encontraron como prioridad los siguientes requerimientos:

- Gestionar los servicios de interacción o ECA's de cada OI.
- Diseñar el flujo de información que permita solicitar la creación del servicio de interacción al OI.
- Crear el servicio de interacción ECA en el escenario de interacción semántica.

Para este prototipo, el usuario desea poder crear y gestionar un servicio de interacción definido como evento, condición y acción en el escenario de interacción semántica, esto se logra construyendo un archivo XML al cual se le ha llamado ECA, luego se envía este archivo al OI como solicitud para la creación del servicio de interacción requerido.

Cabe notar que no existe un método o petición que permita obtener una lista dinámica de los ECA creados ni de sus estados actuales, lo cual genera conflictos con otras aplicaciones como Clipio [21], esta aplicación implementada previamente en el escenario que se utiliza como base para este desarrollo, hace uso de una base de datos interna para realizar la gestión requerida.

La característica mencionada anteriormente crea la necesidad de implementar una solución que permita la gestión de servicios de interacción, bien sea a través de una base de datos interna en la interfaz de AR para la IoT, o acceder al sistema operativo en la tarjeta Galileo, donde está



Interfaz de Realidad Aumentada Para la Internet de las Cosas.

desplegado el OI respectivo, y así obtener la lista de los archivos ECA creados y su estado respectivo, accediendo al sistema de archivos de Yocto Linux, en el cual se implementó el escenario de interacción semántica con cada OI.

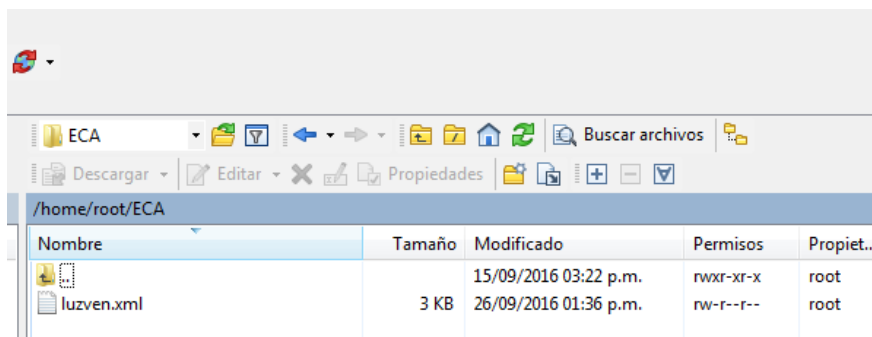
La solución propuesta crea la necesidad de desarrollar dos escenas, una para gestionar los servicios de interacción de manera dinámica y otra para poder crear dichos servicios.

- **Escena 4 Gestor de ECA.**

El primer aspecto a tener en cuenta para diseñar esta escena es poder conocer cuántos y cuáles son los ECA's existentes en cada OI y sus estados para poder cambiarlos, o poder borrar dichos servicios; el usuario se posiciona en el objeto o punto de interés PI, a continuación se despliega la lista de ECA's si los hay, con un color de fondo verde si esta encendido y rojo si se encuentra apagado, este estado se puede cambiar mediante la selección de la etiqueta por medio del *touch* o *click* en la pantalla. La funcionalidad de borrar el servicio de interacción funciona de la misma manera mediante una etiqueta adyacente a la etiqueta del servicio de interacción ECA.

- **Acceso al directorio ECA del OI.**

Cada OI funciona con un sistema de archivos en Linux Yocto, el proceso de acceso y gestión de estos archivos se indica en el Anexo G del presente documento. Como se observa en la Figura 39, la carpeta ECA almacena las interacciones programadas por el usuario, los demás archivos son los propios del funcionamiento interno del OI.



Nombre	Tamaño	Modificado	Permisos	Propiet...
luzven.xml	3 KB	15/09/2016 03:22 p.m.	rwxr-xr-x	root
		26/09/2016 01:36 p.m.	rw-r--r--	root

Figura 39. Carpeta ECA del sistema de archivos del OI.

Para acceder a la carpeta ECA se ha creado un proceso en el lenguaje Python con el nombre *Ecadirinfo.py*, este algoritmo se ejecuta mediante un hilo en el directorio *ObjetoInteligente.py*. Toda la construcción, el código y la lógica para la creación de esta funcionalidad y similares se encuentran descritos en los anexos G y H de este documento.

La comunicación con el OI se hace a través de la palabra petición "ecadir_ok" utilizando el script *ConsultaDirECA.cs*, este script también recibe la lista de servicios de interacción ECA's obtenidos como respuesta, esto permite obtener cuantos elementos tipo Etiqueta se deben instanciar o crear, según el número de servicios existentes en el OI.

- **Acceso al estado del servicio de interacción ECA.**

Interfaz de Realidad Aumentada Para la Internet de las Cosas.

Para acceder al estado actual de cada ECA se crea un programa en Python de forma similar al anterior, con el nombre *Stateeca.py*, el cual responde con el estado del servicio de interacción ECA en una cadena.

La comunicación con el OI, se realiza mediante el envío de la cadena que contiene el nombre del servicio de interacción ECA a consultar, obteniendo como respuesta una cadena “on” u “off” según el estado del servicio, todo utilizando el script *ConsultaStateECA.cs*

Para la creación de las etiquetas que indican el nombre, el estado y la acción de borrar el servicio, junto con sus funcionalidades respectivas se utiliza el script *EtiquetasECA.cs*, el cual instancia el prefabricado llamado *ecacuads* cuya estructura y funcionalidades se describen en el anexo I del presente documento.

La Figura 40 muestra una etiqueta con el nombre de un servicio de interacción activo en ambos OI relacionados, evento y acción. Al hacer click o touch en dicha etiqueta se cambia el estado del servicio en cada uno de los OI, otra etiqueta marcada con una X contigua al tocarla permite borrar el servicio de interacción en cada OI.



Figura 40. Escena 4 Servicios de interacción. Fuente propia.

En este prototipo, la gestión de los servicios de interacción se hace en cada OI por separado, es decir que si el usuario se ubica en el regulador de humedad, los eventos que puede gestionar son los relacionados con los recursos sensor de humedad y actuador de riego. Debido a esto, este prototipo requiere que para completar un cambio de estado o borrar un servicio, se realice la actividad en cada uno de los OI involucrados en el ECA, es decir que el usuario deberá realizar las actividades tanto en el objeto evento como en el objeto acción haciendo uso de su dispositivo móvil.

- **Escena 5 Crear ECA.**



Como se mencionó anteriormente la lógica establecida en el escenario de interacción semántica, es la lógica de ejecutar una acción a partir de un evento dado mediante una condición específica, en este escenario los eventos son los recursos sensores y actuadores de los OI, mientras que las acciones solamente las ejecutan los actuadores.

La condición en un servicio de interacción ECA se establece mediante comparadores matemáticos como son: mayor, menor e igual, y los valores de la condición son: un valor entero para la medida de los sensores, y un valor lógico de estado para los actuadores, es decir un valor binario o booleano indicando "0" para apagado y "1" para encendido. Estos valores se ingresan mediante texto a través de la interfaz ya que la prioridad es crear la petición ECA.

Esta petición se va creando a medida que el usuario recorre la interfaz, para esto se tiene un *GameObject* genérico o *Empty* de nombre *ECAObj*, donde se van llenado las variables de las peticiones *ECAEvento.cs* y *ECAAccion.cs* como se observa en la Figura 41, las cuales son idénticas y se envían al objeto evento y acción respectivamente para crear el servicio de interacción ECA.

Variable	Valor
Script	ECAEvento
Broker Hostname	iot.eclipse.org
Broker Port	1883
User Name	
Password	
Sustopic	
Pubtopic	
Respuesta E	
Id_objeto Destino	
Nombre ECA	
Estado ECA	off
Entidad_interes	
Id_objeto_evento	
Nombre_objeto_eve	
Id_recurso_evento	
Nombre_recurso_ev	
Comparador_condici	
Tipo_variable_condic	
Variable_condicion	none
Unidad_condicion	
Significado_condicion	cambio recurso evento
Id_objeto_accion	
Nombre_objeto_acci	
Id_recurso_accion	
Nombre_recurso_aci	
Estado_accion	
Significado_accion	accion cambiada

Figura 41. Variables del script *ECAEvento.cs*.

- **Creación del servicio.**

El servicio de interacción se crea mediante la petición *ECA.xml*, la cual se le asigna un nombre en específico y deberá ser publicada tanto en el objeto que genera el evento, como en el que genera la acción, con los parámetros específicos de cada uno. Los servicios de interacción ECA por defecto se crean en estado apagado.

Interfaz de Realidad Aumentada Para la Internet de las Cosas.

La interfaz primeramente requiere el nombre del ECA como se observa en la Figura 42, la captura de este dato se da mediante el elemento UI campo de texto *Input Field*, utilizando el script *UIInputField.cs*, el cual espera que el usuario ingrese la cadena con el nombre, a través de una función de escucha tipo *Listener* dispara una función interna generando una serie de acciones en el HUD de la interfaz para permitirle al usuario seleccionar el recurso evento.

Esta misma dinámica aplica con cada uno de los elementos UI usados para la captura de datos en la interfaz como son listas desplegables o *UI Dropdown*, botones *UI Button*, y demás con sus respectivos scripts.



Figura 42. Escena 5 Nombre del servicio de interacción ECA. Fuente Propia.

Seguidamente el usuario se posiciona sobre el marcador asociado al OI de su interés para seleccionar el recurso evento, haciendo click o touch sobre su representación gráfica, luego la interfaz usuario debe colocar los elementos UI para capturar la condición y el valor del evento como se observa en la Figura 43, esto con el fin de que la interfaz le permita al usuario escoger el recurso para realizar la acción en el mismo OI o en cualquier otro recurso actuador de otro OI del escenario.



Figura 43. Escena 5 – Selección del recurso evento y establecimiento de la condición del servicio de interacción ECA.

Una vez seleccionado el recurso acción, aparece la lista desplegable con los dos estados posibles del actuador, “on” para encendido y “off” para apagado como se observa en la Figura 44, el usuario selecciona uno de ellos y presiona el botón de OK para así crear el servicio de interacción publicando los archivos ECA correspondientes en los OI que contienen los recursos evento y acción.



Figura 44. Escena 5 Selección del recurso acción, su estado y la creación del servicio de interacción.

Cabe notar que si se activa un ECA, se deberán desactivar previamente los servicios básicos asociados a los OI involucrados para evitar conflictos, sin embargo para este prototipo se



desactiva el servicio básico asociado a cada OI por separado y solo en el momento que se escoge un recurso evento.

4.7. PROTOTIPO 2.

Para este prototipo se utiliza la versión 6 del SDK Vuforia el cual permite ser utilizado en arquitecturas de 64 bits, y la versión 5.4 de Unity 3D, con lo cual se tienen las versiones más actuales de las herramientas software a la fecha.

Teniendo en cuenta las pruebas y consideraciones realizadas al prototipo 1 por el cliente en cada iteración o escena y en la evaluación general, se obtuvieron nuevas funcionalidades a adicionar, y la necesidad de eliminar de otras que requieren esfuerzo cognitivo y de memoria por parte del usuario, la validación general del prototipo 1 y las observaciones de cada iteración se encuentra en el anexo J de este documento.

Como primera medida se pueden implementar dos modos de interacción, uno normal cuya ergonomía se base en sostener el dispositivo móvil con una o ambas manos, y realizar las tareas de la aplicación a través de la pantalla táctil o *touch* del dispositivo, el otro modo de interacción sería estereoscópico, donde a través de un artefacto puesto en la cabeza, se monta el dispositivo móvil, brindando un tipo de experiencia AR tipo *eyewear* donde se requiere el uso de otros tipos de interacción como el uso de retículas o *gaze*, y el seguimiento de los movimientos de la cabeza haciendo más intuitiva la interfaz, cabe notar que este último aspecto no es prioritario.

La división por escenas facilita realizar algunos de estos requerimientos, así pues se aborda cada escena para realizarle las modificaciones que se tengan a lugar.

4.7.1. Observaciones generales para la creación del prototipo 2.

Mediante la prueba general del prototipo 1 se logró la validación de las funcionalidades de la interfaz de AR para la IoT, comprobando su funcionalidad como utilidad del prototipo al no ser un producto final. Las observaciones y sugerencias brindadas por el cliente en su rol de usuario sirvieron para mejorar el anterior prototipo sin desecharlo, y con el fin de brindarle mayor usabilidad e intuición a la interfaz, así se establecieron los requerimientos generales para el prototipo 2.

Algunos de estos requerimientos no se desarrollaran debido a varios factores, como el hecho de que la interfaz está desplegada sobre un escenario que es una prueba de concepto, el cronograma asignado para este proyecto, la carencia de algunos recursos y el tiempo de desarrollo, por esto irán señalados como no prioritarios o indicados como trabajo en este documento. A continuación la lista de requerimientos:

- Mejorar los tiempos de respuesta o utilizar transiciones graficas en los servicios
- Mantener en todo momento informado al usuario sobre el contexto en el que se encuentra.
- Fijar un marcador o textura al recurso IoT respectivo.
- Botones de navegación entre escenas.
- Establecer convenciones de color.

- Uniformidad con los elementos UI y el texto.
- Establecer HUD para varias resoluciones.
- Optimización, encendido y apagado de sensores, y disponibilidad de los mismos en el dispositivo móvil. (no prioritario)
- Implementar modo estereoscópico.(no prioritario)
- Utilizar botones virtuales. (no prioritario)
- Ayudas Auditivas y visuales como el resaltado(no prioritario)

4.7.2. Solución General.

Vuforia ofrece a los desarrolladores un paquete de Digital Eyewear llamado AR/VR Sample⁶³ el cual brinda la interacción en realidad mixta, de este se pueden aprovechar ciertos elementos para realizar las funcionalidades requeridas en este prototipo.

Las imágenes iniciales *splash screen* fueron modificadas para lograr el efecto deseado de las transiciones, brindando información acerca de la aplicación y unas breves instrucciones, las escenas que hacen parte de estas transiciones o Fades se muestran en la Figura 45. La primera imagen es una presentación de los grupos de desarrollo que engloban el proyecto y las tecnologías involucradas, tiene una duración de tres segundos, esto se logró modificando el script *AsyncSceneLoader.cs*, la siguiente imagen presenta una serie de instrucciones y las dos opciones de la interfaz, el modo, móvil o normal y el modo estereoscópico.



Figura 45. Imágenes *Fades* iniciales.

Las transiciones entre escenas se logran a través de insertar entre estas, una escena nueva llamada *Loader*, la cual llamada *Loader*, la cual carga la escena siguiente mediante el script *LoadingScreen.cs*, esta escena es una funcionalidad del ejemplo de AR/VR de Vuforia, trae consigo una pequeña animación tipo *spinner* para la transición como se observa en la Figura 46

Figura 46, estas transiciones se crean para cada cambio de escena.

⁶³ <https://developer.vuforia.com/downloads/samples>

Interfaz de Realidad Aumentada Para la Internet de las Cosas.

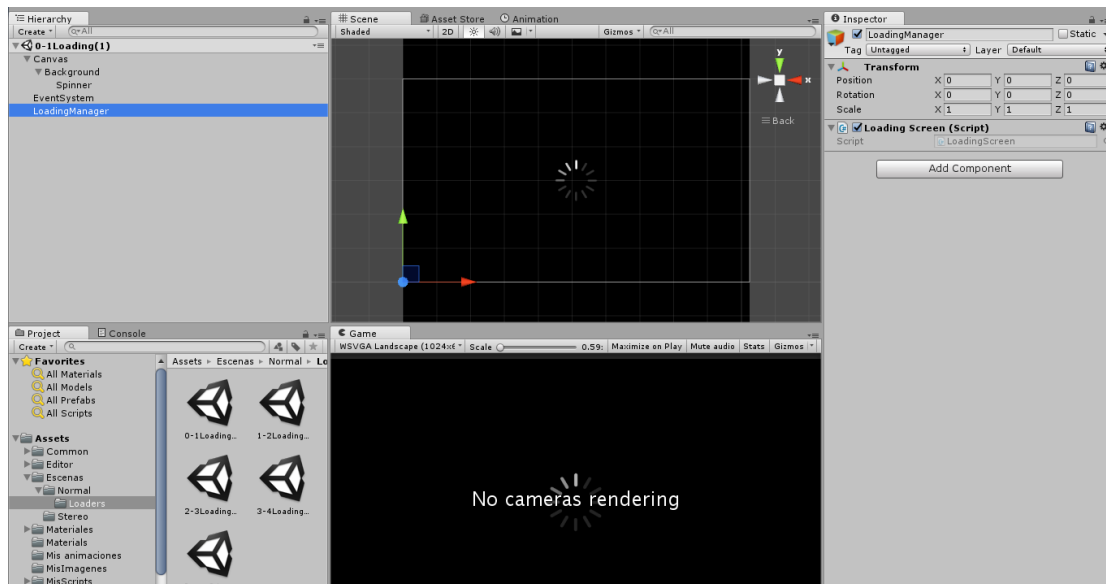


Figura 46. Escena de transición Loader entre escenas.

El HUD se modificó de tal manera que permite resolver los requerimientos del prototipo 2, respecto al contexto, la EI se muestra en todo momento, el usuario puede verla en un texto en la parte inferior, centrada en el HUD, en todas las escenas. Los botones de navegación se personalizaron utilizando algunos iconos y sonidos del juego Pokemon Go de Niantic, como son los botones de Salir de la aplicación y navegar entre escenas o funcionalidades haciendo la interfaz más intuitiva, así mismo se estableció el color blanco para las fuentes de asistencia y se estandarizaron todas a un mismo tamaño y uniformidad.

Respecto a las múltiples resoluciones de pantalla, la aplicación AR/VR fija el modo horizontal o *Landscape* como predefinido, y un modo de renderizado de los elementos *Canvas* llamado *Screen Space-Overlay* el cual se ajusta a diversos tipos de resolución de pantalla para un mejor ajuste. La Figura 47 muestra todos estos cambios realizados a la aplicación.

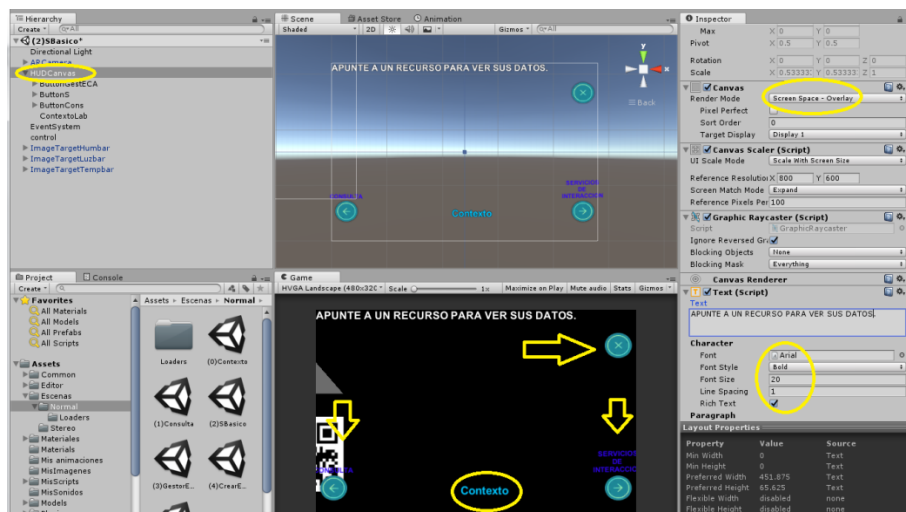


Figura 47. Elementos modificados del HUD.

La optimización de la aplicación se va dando a medida que lo permiten los tiempos de desarrollo priorizando o dejando de lado algunas funcionalidades requeridas, haciendo uso de co-rutinas para la creación de funciones asíncronas, es decir que se ejecuten en segundo plano en la aplicación, y la gestión de recursos como GPS y las comunicaciones MQTT.

4.7.3. Solución para la escena 1.

A continuación se muestran los requerimientos obtenidos a partir de la evaluación de la iteración de la escena 1 del prototipo 1.

- Menú para escoger entre modo móvil o estereoscópico - *eyewear*.
- Al utilizar el código QR Indicar al usuario que debe dirigir su mirada fuera del marcador para proceder a contextualizar.
- Escoger índice para la consulta (no prioritario)
- Escoger tipo de consulta(no prioritario)

En la solución general del prototipo 2 se implementaron las pantallas que permiten escoger el modo de interacción entre móvil y estereoscópico.

Se ha movido la barra deslizadora o *slider* desde la escena 2 a esta escena para una mejor funcionalidad, pudiendo seleccionar el radio independientemente del tipo de consulta, escrita o por marcador. Cuando se realiza la consulta apuntando al marcador, la interfaz muestra el contexto, indicándole al usuario que retire la vista de la imagen marcador como se observa en la Figura 48, cuando esto sucede, se dispara la escena 2 donde se observan los recursos disponibles, con el radio seleccionado.

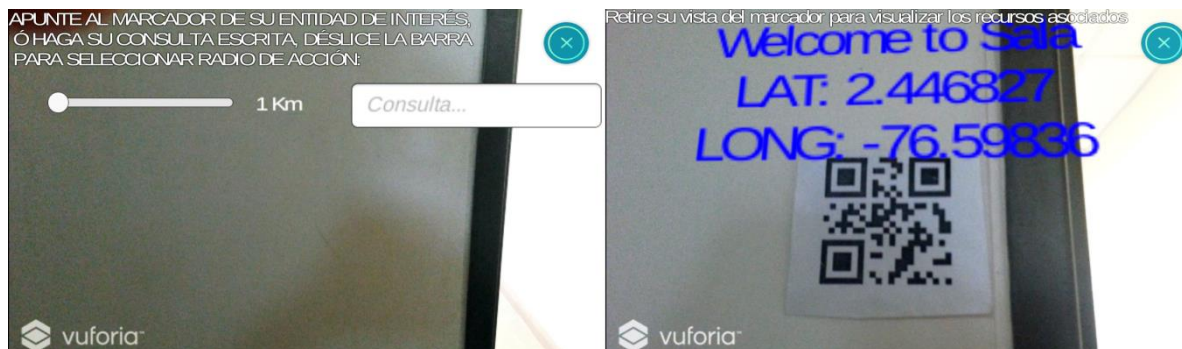


Figura 48. Escena 1.

4.7.4. Solución para la escena2.

Estos son los requerimientos encontrados en la iteración de la escena 2 del prototipo 1:

- Vincular un link o URL web para acceder a los metadatos de cada recurso IoT en su middleware asociado.
- Resolver la funcionalidad del slider o eliminarlo.
- Detener el GPS para optimizar el código.
- Implementar asistentes mini mapa y radar(no prioritario)

Interfaz de Realidad Aumentada Para la Internet de las Cosas.

- Localizar los recursos IoT en el campo de visión del usuario. (no prioritario)
- Mejorar las representaciones graficas de los OI (no prioritario)

Se definió un botón para el link web de los metadatos del OI, a través del servicio del middleware de Xively, cuyo formato es: http://www.personal.xively.com/feeds/Feed_objeto con el fin de no saturar la pantalla del HUD de la aplicación como se observa en la Figura 49.



Figura 49. Escena 2.

Este link se crea a través del script *Zoom.cs* y se implementa a través de un botón que aparece al seleccionar la representación gráfica de los recursos IoT desplegados en la pantalla.

Se tomó la decisión de reubicar el slider del radio de acción de la consulta hacia al escena 1, para evitar hacer consultas cada vez que el usuario movía el slider, pues esto hacia que la pantalla se congelara debido a la consulta en el índice, la instanciación de los objetos, y sumado al proceso de consulta de los datos en el objeto respectivo, la aplicación consume bastantes recursos.

Se tiene que aunque es ideal poder tener representaciones virtuales de los OI que indiquen su localización en tiempo real en el escenario definido, la dificultad de poder tener una localización en interiores o *indoor* dinámica radica principalmente en su nivel de precisión. Por tanto las representaciones se siguen dejando estáticas en la pantalla.

Las etiquetas del HUD, presentan más información al usuario, y la optimización del código de la escena *Consulta.cs*, se mejoró implementando co-rutinas y deteniendo el sensor GPS cuando ya no sea necesario utilizarlo.

4.7.5. Solución para la escena 3.

Para esta escena se encontraron los siguientes requerimientos:

- Mejorar las representaciones virtuales o modelos 3D.

Interfaz de Realidad Aumentada Para la Internet de las Cosas.

- Resaltar las representaciones virtuales de los OI, indicarle al usuario la funcionalidad del servicio básico y sus umbrales.
- Implementar botones virtuales para el servicio básico(no prioritario)
- Implementar el servicio de apagar los actuadores a voluntad(no prioritario)

Para mejorar las representaciones graficas correspondientes a los sensores de los OI, se implementaron barras de nivel verticales las cuales de forma visual y dinámica fluctúan a razón del valor del sensor y el fenómeno a medir como se observa en la Figura 50, también se implementó el cambio de color de los actuadores, rojo si está apagado y verde si esta encendido.

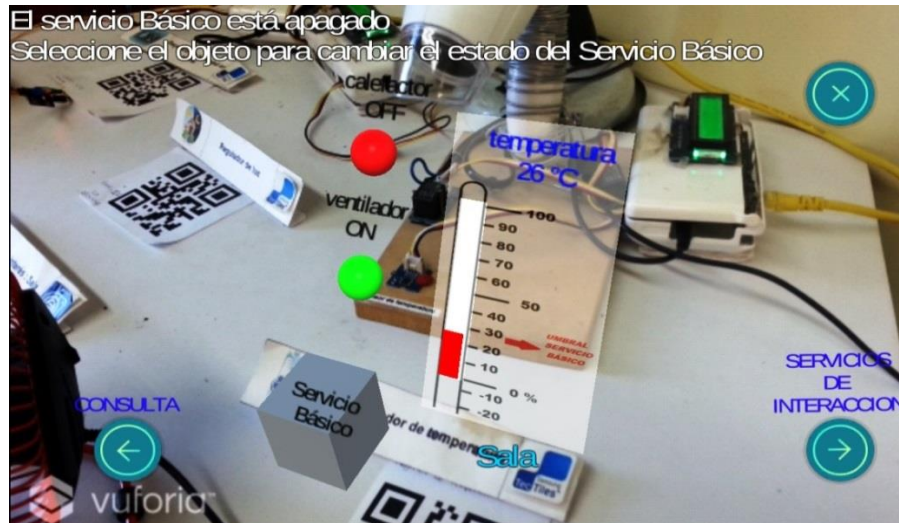


Figura 50. Escena 3.

4.7.6. Solución para la escena 4.

A partir de las observaciones y pruebas realizadas al prototipo 1, se obtuvieron los siguientes requerimientos.

- Cambiarle el nombre a la escena para que sea comprensible al usuario.
- Prevenir las peticiones de estado vacías o nulas al OI.
- Brindar información relevante del ECA al usuario.
- Apagar el servicio básico cuando se enciende el ECA.
- Modificar el estado o apagar el ECA tanto en el objeto evento como en el objeto acción sin importar en cual OI haga la gestión el usuario.
- Organizar los objetos tipo Etiqueta o *Label* como instancia hija del marcador (no prioritario).

Se asignó como nombre a la escena, “Gestión de servicios de interacción”, ya que es el nombre más adecuado para la funcionalidad de la misma y no cae en ambigüedades.

Se creó la segunda versión del código *Stateeca.py* para obtener el archivo ECA consultado, esto permite mostrar más información al usuario sobre el servicio de interacción.

Se creó un nuevo botón llamado “Descripción” a la izquierda de la etiqueta que lleva el nombre y el estado del servicio, en esta nueva etiqueta se brinda información acerca de la función del servicio de interacción como se observa en la Figura 51, funciona mediante el script *ToqueInfoECA.cs*.

Se creó la lógica para que la función de apagar el servicio básico suceda en esta escena cuando se activa un servicio de interacción, para esto simplemente se agregó al objeto prefabricado *Ecaquads* el script *SBEca.cs* y al script *ToqueSet.cs* asociado se le agregó la funcionalidad de apagar el servicio básico al activar el servicio de interacción ECA.



Figura 51. Escena 4, función Descripción.

Para que las activaciones o el borrado de los servicios de interacción se hagan efectivos tanto en el recurso evento como en el recurso acción, se crearon los scripts *SetEventoState.cs*, *SetAccionState.cs*, *BorrarEvento.cs*, *BorrarAccion.cs* y se asociaron al objeto prefabricado *Ecaquads*.

Respecto a las peticiones nulas se capturaron dichas excepciones en cada script o petición creada para modificar el estado del servicio de interacción o ECA.

4.7.7. Solución para la escena 5.

Como requerimientos en la prueba de la escena 5 del prototipo 1 se encontró:

- Ajustar los valores de selección de evento para booleano y flotante según sea el caso un recurso tipo actuador o un sensor.
- Crear un botón para cancelar.
- Seguir la secuencia de la creación del servicio de interacción y cambiar la palabra “ok” por “crear servicio” al finalizar.
- Restringir o deshabilitar las selección de recursos al seleccionar uno de ellos.
- Generar el nombre del servicio de interacción ECA de manera automática(no prioritario)
- Crear o solicitar al usuario la descripción del evento y de la acción.(no prioritario)

Para una mejor interacción, se bloquearon los marcadores mientras se escribe el nombre del ECA, también se ajustaron los valores para discriminar entre tipo flotante y booleano los recursos de los eventos, también se creó un botón de cancelar que devuelve al usuario al inicio de la escena y se modificó el botón de finalizar colocando una etiqueta de crear servicio, la interfaz también evita poder seleccionar como acción un sensor como se observa en la Figura 52.



Figura 52. Escena 5.

En un principio La acción de crear el ECA requiere reiniciar el sistema, lo cual toma alrededor de 1 minuto, el reinicio se hace a través de la petición “reboot_ok” mediante el script *Reboot.cs*, indicándole al usuario que deberá esperar para volver a utilizar la interfaz.

4.8. PROBLEMAS, SOLUCIONES Y CONCLUSIONES.

- Debido a que la aplicación está fundamentada en el escenario de interacción semántica de Guerrero-Riobamba, hubo que dedicar un tiempo extra a la comprensión y adaptación de dicho escenario a la interfaz requerida, pues ambos proyectos se realizaron casi paralelamente.
- La versión de 64 bits de Unity 3D no fue compatible con el SDK Vuforia respecto a los permisos para utilizar la cámara web, sino hasta agosto del año 2016, esto impidió la evaluación de las escenas desarrolladas sobre esta arquitectura, por esta razón se instaló la versión de Unity 3D de 32 bits.
- Los lanzamientos continuos de versiones del motor y SDK utilizado, son hechos en tiempos muy cortos, dificultando la curva de aprendizaje y frustrando al desarrollador, debido a que se incluyen nuevas funcionalidades que no son compatibles con versiones anteriores. Se destacaron entre estos lanzamientos la inclusión formal de una plataforma exclusiva para el desarrollo de aplicaciones en realidad virtual VR en el segundo trimestre de 2016.
- La aplicación Clipio implementada por Guerrero Riobamba para el escenario de interacción semántica, no es comparable con la aplicación propuesta en este proyecto, ya que Clipio hace



Interfaz de Realidad Aumentada Para la Internet de las Cosas.

uso de una base de datos interna, mientras la aplicación consulta datos en forma dinámica aplicando los principios de computación en la niebla (*fog computing*) donde las tareas de procesamiento se asignan esperando que tengan una baja latencia.

- El escenario de interacción semántica no tiene un servicio que permita apagar o prender un actuador, funcionalidad que es echada de menos por los usuarios y por consiguiente la interfaz AR no tiene dicha funcionalidad.
- Se encontró que no todas las funcionalidades del escenario se pudieron representar en AR de una manera directa, ya que algunas no estaban bien desarrolladas o requerían desarrollos adicionales, por ello se concluye que se requiere validar primero los servicios que ofrece el escenario IoT y su capacidad para ser representados en una interfaz AR, con el fin de establecer las funcionalidades a desarrollar en la aplicación AR.
- Los tiempos de respuesta de una aplicación de AR están condicionados a los tiempos de respuesta de la funcionalidad del escenario IoT asignado.

La fase de validación y evaluación de usabilidad de los prototipos se realiza a continuación en el capítulo 5 de este documento.



5. VALIDACION Y EVALUACION DE USABILIDAD

5.1. CONSIDERACIONES.

Para evaluar la interfaz obtenida con respecto a la usabilidad, se enfocó la evaluación en que tan fácil e intuitiva fue la experiencia para el usuario de prueba, basándose en el apartado de usabilidad del capítulo 2 del presente documento.

La evaluación de la experiencia de usuario en interfaces de AR casi siempre es tratada desde el punto de vista del desarrollador y no desde el cliente o consumidor [51], sin embargo los roles sugeridos en la guía de desarrollo permitieron probar la interfaz sin asumir la formalidad de un usuario final.

Como la interfaz está desplegada sobre un escenario definido, esto condiciona la usabilidad, es decir partiendo de la afirmación que la interfaz es usable si es apropiada en su contexto, esta característica hace que la interfaz deba ser apropiada respecto a las tareas a realizar, el entorno y la experiencia del usuario como tal.

Se tuvo en cuenta que la interfaz está desplegada sobre el escenario de interacción semántica, el cual es una prueba de concepto, por lo tanto se esperaban conflictos de usabilidad y por ende en la experiencia de usuario, algunos debidos a la manera de interactuar entre los objetos y las tecnologías involucradas, además de los tiempos de la lógica de la interfaz en AR desarrollada.

A partir del objetivo propuesto en este documento sobre la evaluación de la aplicación, se tuvo en cuenta que el escenario de pruebas ya tiene unas restricciones inherentes como el hecho de ser una prueba de concepto, y partiendo de la referencia del capítulo 2, los indicadores de usabilidad cuantificables no se aplican en esta evaluación, pues la experiencia de usuario es una medida subjetiva y además al no existir un referente previo de la aplicación, no existe un punto de comparación respecto a datos cuantificables como uso de recursos y tiempos.

La evaluación de la aplicación se dividió en dos fases, la primera fue un proceso de validación del primer prototipo respecto a la consecución de los objetivos funcionales de la interfaz a través de las tareas definidas y las observaciones pertinentes para mejorar el siguiente prototipo, esta fase de pruebas fue completada y documentada en el Anexo J de este documento.

La segunda fase es la que se expone en este capítulo como la facilidad de uso de la aplicación en su segundo prototipo, pues es esto último lo que mejora la experiencia del usuario [30], el formato de la prueba de usabilidad se basa en la propuesta que se formula a continuación.

5.2. PROPUESTA DE EVALUACION DE USABILIDAD.

5.2.1. Fase 1. Validación del prototipo 1 (Funcionalidad).

La prueba del primer prototipo es un proceso de validación de la funcionalidades del escenario IoT desplegadas en la interfaz de AR, donde la herramienta de desarrollo es el motor de videojuegos



Unity 3D, este permite visualizar e interactuar con el software en tiempo real, además se integra y soporta en un IDE de programación, el cual hace las veces de editor, esto permite probar la aplicación por etapas y a lo largo del proceso de desarrollo. Estas características posibilitan realizar pruebas casi sin costo en recursos o logística y con mucha retroalimentación para quien hace el rol de desarrollador y el rol de cliente.

Como se observó en el capítulo 2 de este documento, antes de evaluar usabilidad primero se debe validar su utilidad, así el prototipo 1 al no ser un producto final se considera útil si cumple con las funcionalidades requeridas en las historias de usuario y capturadas en los requerimientos priorizados, es decir una validación previa por parte del desarrollador y el cliente para establecer si la funcionalidad desarrollada es importante y resuelve un problema de interacción de manera adecuada, de acuerdo al marco de referencia propuesto.

Estas pruebas de validación son informales, se realizaron durante todo el proceso de desarrollo en cada etapa o escena de acuerdo a la guía de desarrollo, también se evalúa la entrega final del prototipo 1 con la aplicación funcional completa como se observa en la Figura 53. Estas pruebas se consideran como la aproximación a una prueba alfa, o a una evaluación heurística realizada por un experto que para el caso sería el mismo cliente y quien hizo el rol de desarrollador.



Figura 53. Proceso de validación del prototipo 1

En este prototipo la capacidad de la aplicación de realizar las funcionalidades básicas del escenario IoT es la prioridad, estas funcionalidades son:

- *Contextualizar o realizar una consulta al índice semántico.*
- *Consultar los datos y metadatos de un recurso IoT tanto en el índice semántico asociado, como en el OI respectivo.*
- *Cambiar el estado de un servicio básico asociado al OI.*
- *Cambiar el estado de un ECA.*
- *Borrar un ECA.*
- *Crear un ECA.*

La metodología utilizada para el desarrollo del prototipo 1 sugiere hacer las pruebas a cada escena a lo largo del proceso de desarrollo para la retroalimentación entre cliente y desarrollador y una



pueba final de cada prototipo, en el anexo J del presente documento se detallan las observaciones encontradas por el cliente.

Estas observaciones permitieron evitar descartar el prototipo 1, y entrar a mejorarlo en el desarrollo del prototipo 2 a partir de las observaciones halladas en cada escena y en la validación general del prototipo 1.

Se realizaron unas pruebas fundamentadas, informales y rápidas teniendo siempre en cuenta que la interfaz se encuentra desplegada sobre un escenario de interacción semántica, que no es un producto final, sino una prueba de concepto, por lo tanto ya de por si se tienen consideraciones y características que pueden generar conflictos con el desarrollo de la interfaz de AR para la IoT.

5.2.2. Fase 2. Evaluación de usabilidad del prototipo 2 (Usabilidad).

Una vez validadas las funcionalidades del escenario en la interfaz de AR, se procede a evaluar la usabilidad a través de la experiencia de usuario como lo sugiere el apartado de usabilidad del capítulo 2 del presente documento. Las pruebas del segundo prototipo son una aproximación a una prueba beta, cuyo objetivo no es validar la aplicación, sino evaluar su facilidad de uso e intuición.

Con la información recolectada en el capítulo dos se formuló un plan de pruebas orientado sobre las siguientes consideraciones para evaluar la usabilidad en el prototipo dos (2) como se observa en Figura 54.

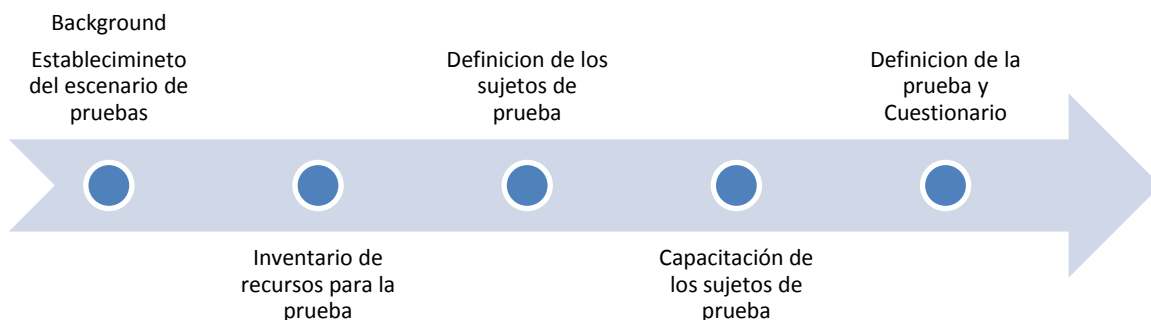


Figura 54. Plan de pruebas prototipo 2.

- **Escenario de pruebas y contexto.**

La oficina 422 del edificio de ingenierías de la Universidad del Cauca en la ciudad de Popayán-Cauca, es la entidad de interés definida para el despliegue del escenario de interacción semántica, este escenario esta descrito en el capítulo 4 de este documento, y presenta los reguladores básicos para una oficina, con esto el usuario percibe una interacción natural y real con su entorno al poder manipular factores cotidianos en una oficina.



El contexto o *Background*, define las condiciones ambientales necesarias y/o deseables para la prueba, en este caso que se tenga un clima ambiental agradable y sobretodo un lugar con buena iluminación, condición necesaria para poder desplegar la aplicación en AR.

Las pruebas se realizaron individualmente en cada sujeto de prueba, sin necesidad de agruparlos en un momento en específico, esto no influirá en los resultados ya que la experiencia es individual y al contrario puede ser beneficioso para los resultados de la evaluación al eliminar la presión por parte de un grupo de usuarios

- **Recursos.**

Los dispositivos son dos Smartphones de gama alta marca Sony y Samsung, donde previamente se les ha instalado la aplicación de nombre Prototype2.apk, estos dispositivos poseen las características necesarias para el despliegue de la aplicación como son cámara, GPS entre otros, de acuerdo al límite colocado al compilar la aplicación para el S.O Android, el cual es la versión Android 4.0 API Nivel 14.

El *Smartphone* hace las veces de coordinador y los OI desplegados en el escenario son los recursos IoT. El servicio básico para la aplicación es la conexión a internet, y no necesariamente el dispositivo coordinador debería encontrarse en la misma intranet o LAN para que la aplicación funcione.

Los cuestionarios para la prueba están disponibles para el usuario así como la aplicación se encuentra instalada previamente en los dispositivos móviles.

Los recursos humanos para realizar prueba son los supervisores de la misma, en este caso el desarrollador y el cliente que para este caso no asumen el rol de usuario, sino de supervisores, además de los mismos sujetos de prueba.

- **Sujetos de prueba.**

Se tuvo un solo grupo de usuarios cuya motivación dependió de que tan cercanos eran al área de conocimiento abarcada, y en la funcionalidad que le encontraron a la interfaz, en este caso al utilizar un índice de domótica en el escenario de interacción, las personas se encontraron interesadas en controlar los recursos IoT de una casa u oficina, evaluando el “como” lo hacen a través de una interfaz en AR, su nivel de satisfacción y la usabilidad de la interfaz.

Los usuarios fueron 8 personas entre mujeres y hombres, estudiantes de ingeniería de sistemas y electrónica con una buena actitud anímica e interés, de una edad promedio de 25 años, con un manejo promedio de aplicaciones móviles y buena disposición de tiempo para la prueba.

- **Capacitación de los sujetos de prueba.**

Antes de realizar la prueba, el usuario y/o usuarios se les pregunto acerca de su disposición para realizar la prueba, luego recibieron una pequeña introducción acerca del concepto de AR, junto con el contexto y las funcionalidades del escenario de interacción semántica, con el fin



Interfaz de Realidad Aumentada Para la Internet de las Cosas.

de que pudieran comprender cada escena de la aplicación y no generasen expectativas inesperadas ya que no era el fin de la aplicación a evaluar.

Esta capacitación previa está orientada a enfocar la evaluación en la aplicación móvil, mas no en el escenario IoT, pues este último es el que genera mayores expectativas al usuario, acerca de sus funcionalidades y el despliegue Hardware del mismo.

- **Prueba**

El usuario sostuvo su dispositivo móvil con una mano y con la otra interactuó con el dispositivo, para realizar las funcionalidades básicas de cada escena por sí solo, o si se requería, entonces a pedido del supervisor. El supervisor tomo atenta nota acerca de las preguntas y/o dificultades encontradas a lo largo del desarrollo de las tareas y brindo ayuda si alguna de estas no puede realizarse con éxito.

- **Cuestionarios.**

La facilidad de uso, ergonomía e intuición de la interfaz junto con la satisfacción del usuario se pueden evaluar a través de un cuestionario simple basado en el test de usabilidad SUS de John Brooke [63], adaptando las preguntas o añadiendo unas nuevas respecto a las necesidades de evaluación de usabilidad que se tienen como objetivo. Este test se utiliza como base por ser ampliamente referenciado y permite que el usuario piense y se cuestione las preguntas antes de responderlas, al ser intercaladas entre positivas y negativas, siendo un método fiable, claro y preciso de una manera razonable, además de ser más barato de implementar y con resultados rápidos.

El cuestionario implementado en esta prueba se encuentra descrito en el Anexo K de este documento junto con los datos obtenidos de la misma.

5.3. RESULTADOS DE LA EVALUACION

La ponderación SUS es un puntaje que varía entre 0 a 100, es el resultado de sumar las contribuciones de cada ítem de los cuestionarios así:

- En el cuestionario al usuario, por cada ítem, se le presentan cinco posibles opciones de calificación (1 a 5) que van desde muy en desacuerdo a muy en acuerdo. Sin embargo, cada ítem hace una contribución de puntaje a la ponderación SUS entre 0 a 4, calculándose de la siguiente manera:
 - Para los ítems 1,3,5,7 y 9 que corresponden a las afirmaciones positivas, se le resta 1 a la calificación dada por el usuario de cada ítem.
 - Para los ítems 2,4,6,8 que corresponden a las afirmaciones negativas, la contribución se calcula como 5 menos la calificación dada por el usuario,
 - Finalmente, se normaliza el resultado obtenido para pasar de una escala de 0 a 40, hacia una escala de 0 a 100, multiplicando toda la contribución por 2,5.



Interfaz de Realidad Aumentada Para la Internet de las Cosas.

Así se obtiene la Tabla 7, cabe anotar que este resultado no es un porcentaje, sino una medida de que tan usable es la interfaz respecto a su facilidad de uso.

SUJETO DE PRUEBA	PUNTAJE SUS
Sujeto 1	66
Sujeto 2	95
Sujeto 3	27.5
Sujeto 4	67.5
Sujeto 5	70
Sujeto 6	85
Sujeto 7	77.5
Sujeto 8	60
Promedio	75.2

Tabla 7. Puntaje de usabilidad SUS entre los sujetos de prueba.

Para describir cualitativamente este resultado, existe un referente general de proyectos que han sido evaluados con la escala SUS a lo largo del tiempo [65], y aunque no necesariamente son aplicaciones en AR, permiten tener un promedio de referencia, respecto a usabilidad, como se observa en la Figura 55.

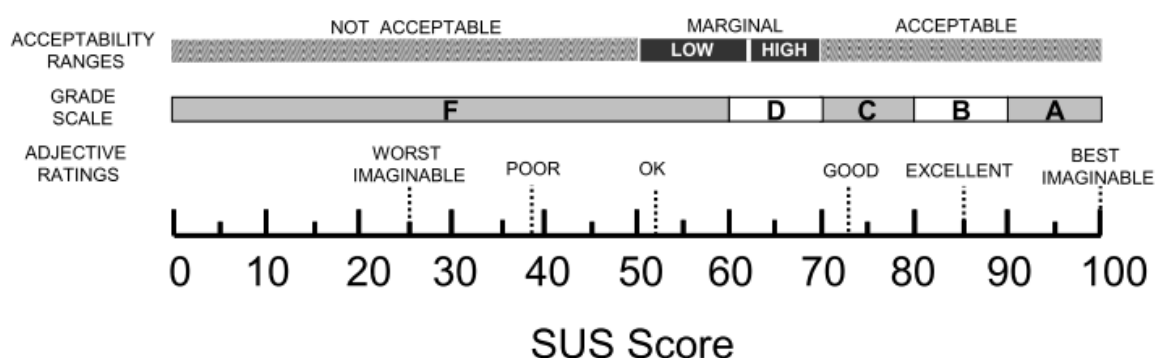


Figura 55. Escala de promedios SUS.

Fuente: A. Bangor [65], Determining what individual SUS scores mean: Adding an adjective rating scale.

Según estos estudios, la escala de promedios SUS indica que después de 68 puntos se tiene una aplicación usable. Para la interfaz de AR desarrollada se obtuvo un puntaje SUS de 75,2, lo cual la ubica como una solución con buena usabilidad.

Se observó que la capacitación previa ponía en contexto a los sujetos de prueba acerca del escenario y por tanto este no les generó tanta expectativa, pues conocían de antemano los servicios que prestaba, el efecto positivo de esto fue que tuvieron mayor interés en el comportamiento de la interfaz que en el despliegue del escenario físico.

La mayor motivación de los sujetos de prueba, estuvo en poder interactuar con las abstracciones de los OI, poder “tocar” sus funcionalidades y observar el cambio de los datos en tiempo real.



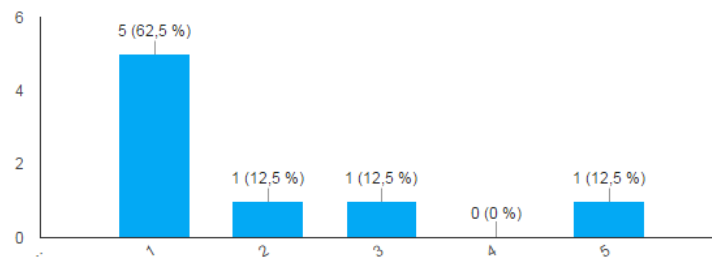
Interfaz de Realidad Aumentada Para la Internet de las Cosas.

Respecto a los servicios de interacción, siempre surgían preguntas para saber cómo funcionaba este servicio, ya que no se comprendía su mecanismo, por ello, se tuvo que asistir a algunos usuarios para que crearán un servicio de interacción hasta hacerlo funcionar, lo cual le generó satisfacción al usuario. Posteriormente a esto, los usuarios ya podían hacer sus propios servicios.

Las preguntas 11 y 12 pretenden complementar la evaluación de la intuición en la interfaz, a continuación, las gráficas relacionadas a estas preguntas en la Figura 56.

11. Creo que las representaciones virtuales de los objetos no tienen nada que ver con la realidad.

(8 respuestas)



12. La interfaz parece saber lo que necesito para poder realizar las funciones de la aplicación.

(8 respuestas)

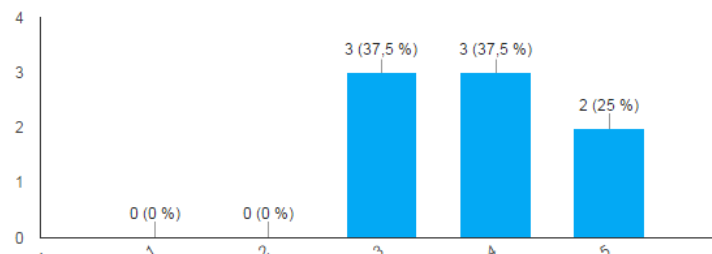


Figura 56. Resultados de la encuesta en las preguntas 11 y 12.

Se observa que las abstracciones de los OI usadas en AR como son barras de nivel, resaltado por colores y funcionalidades como la de poder tocarlos en la pantalla, brindan satisfacción al usuario y hacen que la interfaz tenga cierta naturalidad respecto a cómo se perciben las cosas.

Respecto a intuir las necesidades de los usuarios, la interfaz cumple de manera aceptable, esto puede deberse a los tiempos de latencia del sistema, o lo rígido de las funcionalidades del escenario donde se despliega la interfaz, con estas observaciones se puede concluir que la interfaz es medianamente intuitiva.



Interfaz de Realidad Aumentada Para la Internet de las Cosas.

Como lecciones aprendidas se tiene que los parámetros importantes para el desarrollo de esta aplicación, como el tiempo de respuesta se pueden ver afectados por los mismos tiempos de respuesta de escenario definido, sus funciones internas y el procesamiento de servicios, por lo tanto la creación de software debería incluir una evaluación previa del hardware del escenario o tener en cuenta estos parámetros para que a solución software tome la decisión de utilizar o no alguno de los recursos o servicios del escenario y no vea comprometida su usabilidad.

Se observó que aunque el escenario IoT donde se despliega la interfaz AR no es un producto final, la interfaz como tal se puede evaluar desde el punto de vista funcional, teniendo claro que no toda funcionalidad de dicho escenario se puede llevar a realidad aumentada, sino que es necesario complementar la interfaz con métodos de interacción clásicos sin ocluir la experiencia aumentada, es ahí donde está la habilidad del desarrollador para llevar un equilibrio entre funcionalidad e intuición de la interfaz.

Dependiendo la experiencia del desarrollador en soluciones AR, y el nivel de comprensión de la arquitectura del escenario, el tiempo como recurso puede ser un factor determinante a la hora de implementar desarrollos de este tipo, si el desarrollador posee estas aptitudes, puede que se requiera menor tiempo para el desarrollo, caso contrario las curvas de aprendizaje crecerían y puede estar en riesgo la viabilidad del proyecto.



6. CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO

6.1. CONCLUSIONES.

Este trabajo allana el camino hacia el desarrollo de nuevas y mejores formas de interactuar en un escenario IoT definido a través del uso de realidad aumentada en interfaces intuitivas y fáciles de usar, bajo los limitantes que se derivan de trabajar con tecnologías emergentes, es decir donde aún no hay estándares definidos, ni un escenario futuro claro además de la limitante tecnológica actual. Bajo este contexto es difícil determinar si la aplicación de AR en la IoT presta utilidad para los usuarios, pero es un avance respecto a lo que existe actualmente y lo que se puede lograr con recursos limitados en un escenario de prueba.

- Se creó un marco de referencia, que exploró de manera conjunta modelos y metodologías de desarrollo de software, seleccionando las más adecuadas para la creación de interfaces de AR en el dominio de la IoT, extrayendo elementos útiles en cuanto a gestión de recursos, gestión de riesgos y flexibilidad, así como la identificación de requisitos necesarios para este tipo de desarrollos.
- A partir del marco de referencia creado, se proponen unas buenas prácticas de diseño de aplicaciones AR en la IoT, con criterios de usabilidad orientada a la experiencia de usuario.
- Se creó una guía de desarrollo basada en: la filosofía ágil y prototipos de desarrollo de software, buenas prácticas de desarrollo en AR, las tecnologías AR en la IoT, enacción y auto organización, con el fin de orientar al desarrollador en la creación de este tipo de interfaces.
- Se creó una interfaz de AR para un escenario de interacción semántica de objetos inteligentes para la IoT, donde el usuario puede gestionar y crear servicios de una manera clara e intuitiva, bajo las condiciones y las limitantes que establece dicho escenario.
- La validación de la utilidad de la interfaz se realizó de manera informal con alta participación del cliente, evaluando cada escena por semana, y una validación general del primer prototipo funcional como una aproximación a una prueba alfa.
- La evaluación de usabilidad se realizó mediante una escala de usabilidad llamada *System Usability Scale (SUS)*, esta permitió evaluar la facilidad de uso y la intuición de la interfaz desde la experiencia del usuario, obteniendo un ponderado de 75,2 sobre 100, con lo cual se concluye una buena usabilidad de la interfaz. Se complementó dicha evaluación con dos preguntas extra, las cuales indicaron que se tenía una buena metáfora de interacción respecto a las representaciones gráficas de los recursos, y una mediana intuición de la interfaz respecto a la intuición de la misma.



6.2. RECOMENDACIONES.

- Unity 3D es un motor muy versátil que, combinado con las diferentes soluciones para el desarrollo de AR orientadas a esta herramienta, permite crear proyectos a base de conexiones, mensajes y jerarquía de componentes, lo cual facilita el desarrollo, y permite probar la aplicación en cualquier momento a muy bajo costo tanto en recursos como en tiempo. Además su característica multiplataforma permite probar los desarrollos en diferentes dispositivos y sistemas.
- Al abordar el escenario IoT en el cual se desplegara la interfaz de AR se debe tener claro y hacerle entender al cliente que no todas sus funcionalidades del mismo son adaptables a AR, y algunas es necesario hibridar entre las interfaces clásicas y este tipo de interacción.
- Se debe evaluar el escenario IoT respecto a sus funcionalidades independientes de cualquier otro tipo de aplicación que haya sido implementada previamente, aunque estas sirvan de referencia, algunas pueden estar apoyadas sobre sí mismas y no sobre el escenario lo cual implicaría problemas respecto al manejo y flujo de datos en etapas posteriores en el desarrollo de la interfaz.
- Siempre es deseable poseer más fuentes de datos aparte de la que brinda el escenario, ya que pueden existir tecnologías que hagan dependiente a la interfaz, por ejemplo una conexión a internet, si por alguna razón los servicios del escenario se caen, conviene tener en la interfaz una captura de estos hechos, u otra fuente de datos.
- Respecto a los mensajes utilizados para crear servicios de interacción, existen datos que son innecesarios o que la misma aplicación los puede crear internamente, ya que pedirselos al usuario es innecesario, estos son los significados de la condición y la acción, y el nombre del servicio, además otras variables pueden ser controladas mediante la ontología de cada recurso, restándole trabajo de procesamiento al dispositivo que implementa la interfaz, más aun cuando está ya tiene una alta carga de procesamiento grafico al desplegar una aplicación en AR.

El acceso al identificador o Feed del OI es uno de los factores que se debe tener en cuenta para la seguridad de la aplicación, es conveniente revisar el uso de códigos QR, etiquetas NFC u otros medios dependiendo de los recursos disponibles se deben crear más servicios de contexto de segundo nivel teniendo en cuenta filosofías como la computación en la niebla (*fog computing*) y la computación en la nube (*cloud computing*), equilibrando calidad, seguridad y tiempo real.

6.3. TRABAJO FUTURO.

- La aplicación propuesta, abarca las funcionalidades del escenario IoT definido más la gestión de los servicios de interacción entre OI, se hace necesario brindarle aspectos de calidad, tanto en las representaciones graficas de los objetos, como en aspectos básicos de seguridad, así como la personalización de la misma, de tal manera que permita su utilización en más dominios aparte del de domótica por ejemplo utilizando más índices y permitiéndole ser útil a un rango más amplio de personas.



Interfaz de Realidad Aumentada Para la Internet de las Cosas.

- Las representaciones gráficas de los objetos y servicios que presta la interfaz requieren elementos que permitan definir y establecer cuáles son los modelos, gráficos y metáforas más adecuados para cada escenario y recurso IoT en su contexto, así poder brindar una solución que permita utilizar de manera ordenada estos recursos AR, resolviendo problemas como la heterogeneidad en los desarrollos y la poca estandarización de representaciones para la IoT.
- Se ha creado un guía de desarrollo la cual puede brindar rapidez en la construcción, flexibilidad y gestión adecuada de recursos y riesgos a un desarrollo de AR orientado al dominio de la IoT para el cuál es necesario realizar más desarrollos de AR que lo utilicen y así aportar a su mejor definición y capacidad para soportar los desarrollos de este tipo.
- Se debe desarrollar un mecanismo para informar al usuario acerca de la disponibilidad de ciertos recursos, por ejemplo, cuando el servicio web del índice esta caído, cuando el middleware utilizado no está recibiendo datos de los recursos, los permisos para acceder a esos datos, también cuando la red LAN que aloja los recursos se cae, cuando el servidor bróker deja de funcionar por alguna razón, o cuando los recursos no están subiendo datos bien sea a la misma aplicación o al middleware.
- Se necesitan realizarle más pruebas a la interfaz de AR, respecto a la usabilidad de manera cuantitativa, para crear un precedente y un punto de inicio para desarrollos venideros, así la evaluación subjetiva que se refiere a la experiencia de usuario e intuición de la aplicación mejoraría sustancialmente, brindándole mayor calidad.
- Respecto a otras tecnologías de desarrollo, la gran competencia en el mercado genera tendencias y nuevas alternativas, una de las más prometedoras es el motor de videojuegos de Unreal Engine, que se ha dispuesto para su uso libre, y el SDK de AR llamado Kudan, que en sus más recientes actualizaciones brinda funcionalidades muy atractivas a la hora del diseño de AR sin marcadores, con la funcionalidad de SLAM, se hace necesario probar dichas tecnologías para el dominio de la IoT.



REFERENCIAS:

- [1] International Telecommunication Union (ITU), "ITU Internert Report: The Internet of Things," pp. 1–28, 2005.
- [2] N. Gershenfeld, R. Krikorian, and D. Cohen, "The Internet of Things," *Sci. Am.*, vol. 291, pp. 76–81, 2013.
- [3] M. C. Domingo, "An overview of the Internet of Things for people with disabilities," *J. Netw. Comput. Appl.*, vol. 35, no. 2, pp. 584–596, 2012.
- [4] P. E. E. Martinez and J. A. G. Macias, "Semantic interactions in the Internet of Things," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 13, no. 3/4, p. 167, 2013.
- [5] S. S. Mathew, Y. Atif, Q. Z. Sheng, and Z. Maamar, "Web of things: Description, discovery and integration," *Proc. - 2011 IEEE Int. Conf. Internet Things Cyber, Phys. Soc. Comput. iThings/CPSCOM 2011*, no. Contribution 3, pp. 9–15, 2011.
- [6] C. V. Ramdas, N. Parimal, M. Utkarsh, S. Sumit, K. Ramya, and B. P. Smitha, "Application of sensors in Augmented Reality based interactive learning environments," *2012 Sixth Int. Conf. Sens. Technol.*, pp. 173–178, 2012.
- [7] P. Meda, M. Kumar, and R. Parupalli, "Mobile augmented reality application for Telugu language learning," in *2014 IEEE International Conference on MOOC, Innovation and Technology in Education (MITE)*, 2014, pp. 183–186.
- [8] G. Papagiannakis, "A survey of mobile and wireless technologies for augmented reality systems," ... *Animat. Virtual ...*, pp. 3–22, 2008.
- [9] J. Wan, D. Li, C. Zou, and K. Zhou, "M2M communications for smart city: An event-based architecture," in *Proceedings - 2012 IEEE 12th International Conference on Computer and Information Technology, CIT 2012*, 2012, pp. 895–900.
- [10] G. Fortino, A. Guerrieri, and W. Russo, "Agent-oriented smart objects development," in *Proceedings of the 2012 IEEE 16th International Conference on Computer Supported Cooperative Work in Design, CSCWD 2012*, 2012, pp. 907–912.
- [11] M. A. Niño-Zambrano, "Interacción Semántica de Objetos en la Web de las Cosas," *XI Coloq. Dr. CLADEA*, no. May, pp. 1–15, 2013.
- [12] T. Leppanen, A. Heikkinen, A. Karhu, E. Harjula, J. Rieki, and T. Koskela, "Augmented Reality Web Applications with Mobile Agents in the Internet of Things," in *2014 Eighth International Conference on Next Generation Mobile Apps, Services and Technologies*, 2014, pp. 54–59.
- [13] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the internet of things: A survey," *IEEE Commun. Surv. Tutorials*, vol. 16, no. 1, pp. 414–454, 2014.
- [14] H. Son, S. Han, and D. Lee, "Contextual information provision on augmented reality with IoT-based semantic communication," in *Proceedings - 2012 International Symposium on Ubiquitous Virtual Reality, ISUVR 2012*, 2012, pp. 46–49.
- [15] The Open Group, "Open Group Standard Open Data Format (O-DF), an Open Group Internet of Things (IoT) Standard." The Open Group, 2014.
- [16] The Open Group, "Open Group Standard Open Messaging Interface (O-MI), an Open Group Internet of Things (IoT) Standard." The Open Group, 2014.
- [17] R. Azuma, "A survey of augmented reality," *Presence Teleoperators Virtual Environ.*, vol. 6, pp. 355–385, 1997.
- [18] M. Weiser, "The Computer for the 21st Century," *Sci. Am.*, vol. 265, no. 3, pp. 94–104, Sep. 1991.
- [19] J. Carmigniani, B. Furht, M. Anisetti, P. Ceravolo, E. Damiani, and M. Ivkovic, "Augmented reality technologies, systems and applications," *Multimed. Tools Appl.*, vol. 51, no. 1, pp. 341–377, 2011.



- [20] D. W. F. van Krevelen and R. Poelman, "A Survey of Augmented Reality Technologies, Applications and Limitations," *Int. J. Virtual Real.*, vol. 9, no. 2, pp. 1–20, 2010.
- [21] D. Riobamba and S. Guerrero, "ESCENARIO DE INTERACCION SEMÁNTICA DE OBJETOS INTELIGENTES EN LA WoT," 2016.
- [22] R. S. Pressman, *Ingeniería Del Software.Un enfoque Practico*, 7th ed. 2010.
- [23] N. ANNA, "Agile Solo. Defining and Evaluating an Agile Software Development Process for a Single Software Developer.," 2011.
- [24] C. Lodts, "Exploring Different Visualization Techniques For Augmented Reality Navigation Helpers Inspired By Open World Games," 2015.
- [25] C. Arth, R. Grasset, L. Gruber, T. Langlotz, A. Mulloni, and D. Wagner, "The History of Mobile Augmented Reality," *arXiv Prepr. arXiv1505.01319*, 2015.
- [26] G. A. García and A. D. R. Reyes, "PROPUESTA DE UN MODELO DE REFERENCIA BASADO EN INTERNET DE LAS COSAS PARA DISEÑAR SOLUCIONES UTILIZANDO TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES," *XX Congr. Int. contaduria , Adm. e Inform.*, 2015.
- [27] A.G. Gurrola, "Unidad 1. Términos básicos para la gestión de la informacion." 2008.
- [28] T. Mandel, "User / System Interface Design," vol. 4, pp. 1–15, 2003.
- [29] B. Dumas, D. Lalanne, and S. Oviatt, "Multimodal interfaces: A survey of principles, models and frameworks," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5440 LNCS, pp. 3–26, 2009.
- [30] I. Isabel, P. Ruiz, G. Víctor, and W. Ortecho, "Realidad Aumentada Para Dispositivos Móviles Con Sistema Operativo Android," 2013.
- [31] T.-L. Chou and L.-J. ChanLin, "Augmented Reality Smartphone Environment Orientation Application: A Case Study of the Fu-Jen University Mobile Campus Touring System," *Procedia - Soc. Behav. Sci.*, vol. 46, pp. 410–416, 2012.
- [32] P. Milgram and F. Kishino, "A TAXONOMY OF MIXED REALITY VISUAL DISPLAYS," *IEICE Trans. Inf. Syst. No.12*, vol. E77–D, no. 12, 1994.
- [33] A. K. Dey and G. D. Abowd, "Towards a Better Understanding of Context and Context-Awareness," *Comput. Syst.*, vol. 40, no. 3, pp. 304–307, 1999.
- [34] N. J. Andre, "A Modular Approach to the Development of Interactive Augmented Reality Applications .," no. December, 2013.
- [35] S. Thrun, M. Montemerlo, D. Koller, B. Wegbreit, J. Nieto, and E. Nebot, "Fastslam: An efficient solution to the simultaneous localization and mapping problem with unknown data association," *J. Mach. Learn. Res.*, vol. 4, no. 3, pp. 380–407, 2004.
- [36] M. Tatzgern, R. Grasset, D. Kalkofen, and D. Schmalstieg, "Transitional Augmented Reality navigation for live captured scenes," *Proc. - IEEE Virtual Real.*, pp. 21–26, 2014.
- [37] N. Petersen and D. Stricker, "Cognitive Augmented Reality," *Comput. Graph.*, vol. 53, pp. 82–91, 2015.
- [38] G. A. LLANOS VILLARREAL, RIVERA GARCIA CARLOS MAURICIO, "Guia Para La Creacion De Realidad Aumentada Orientada a La Publicidad," 2012.
- [39] C. E. Serrano Castaño, "Modelo Integral para el profesional en ingenieria.Capítulo 2 modelo para la investigación documental," 2008.
- [40] C. Hoyos, "Un modelo de Investigacion Documental," 2000.
- [41] S. Lauesen, "Task descriptions as functional requirements," *IEEE Softw.*, vol. 20, no. 2, pp. 58–65, 2003.
- [42] S. Rivadeneira, G. Vilanova, M. Miranda, and D. Cruz, "El modelado de requerimientos en las metodologías ágiles," *XV Work. Investig. en Ciencias la Comput.*, pp. 383–387, 2013.
- [43] A. B. Hollar, "Cowboy: An Agile Programming Methodology for a Solo Programmer,"



- no. December, pp. 1–73, 2006.
- [44] D. Rosero and A. Vela, “Sistema de realidad aumentada para ayuda a personas con deficiencia mental leve.” 2016.
- [45] V. Caballero and A. Villacorta, “Aplicación móvil basada en realidad aumentada para promocionar los principales atractivos turísticos y restaurantes calificados del centro histórico de lima,” 2014.
- [46] T. Marshall, “Moving the Museum Outside Its Walls : An Augmented Reality Mobile Experience,” 2011.
- [47] J. Lopez, “Videojuego multijugador basado en realidad aumentada,” pp. 1–67, 2014.
- [48] O. J. Sedano Fernández, “Estudio y desarrollo de una aplicación móvil de Realidad Aumentada,” pp. 1–91, 2014.
- [49] H. Reuterdaahl, “Mobile Marker-based Augmented Reality as an Intuitive Instruction Manual,” no. May, 2014.
- [50] A. Palma and N. Sanchez, “Desarrollo de una aplicación móvil con realidad aumentada para apoyar la enseñanza de las matemáticas en educación secundaria .,” 2013.
- [51] D. W. Seo, H. Kim, J. S. Kim, and J. Y. Lee, “Hybrid reality-based user experience and evaluation of a context-aware smart home,” *Comput. Ind.*, vol. 76, pp. 11–23, 2016.
- [52] L. Garduno Massieu, “GALLAG Strip: A Mobile, Programming With Demonstration Environment for Sensor-Based Context-Aware Application Programming.” 2012.
- [53] J. A. Moreno Parra, “Desarrollo de un prototipo de maqueta arquitectonica interactiva utilizando realidad aumentada.” 2014.
- [54] S. Uceda, “Uso de la Realidad Aumentada para facilitar la lectura e interpretación de planos,” pp. 1–117, 2014.
- [55] D. Molina, “Realidad Aumentada en dispositivos iOS como sistema de ayuda a los alumnos,” p. 83, 2012.
- [56] P. Ochoa, “Desarrollo de un sistema de informacion para el ejercito en AR_VR,” 2012.
- [57] C. F. Alfaro and C. C. Morales, “Desarrollo e implementación de una interface de usuario interactiva y un módulo de entrenamiento para el sistema de simulación de juego de billar ARPool,” vol. 1, pp. 97–117, 2012.
- [58] V. Heun, J. Hobin, and P. Maes, “Reality Editor: Programming Smarter Objects,” 2013.
- [59] A. Martínez, “Realidad Aumentada en Interiores : posicionamiento del usuario en dispositivos móviles y aplicaciones en rehabilitación y guiado (GuIAR),” p. 146, 2013.
- [60] F. Barbara, “Estudio de algoritmos de localización en interiores, para tecnologías móviles de última generación,” 2012.
- [61] M. A. Catalán Vega, “Metodologías de evaluación de interfaces gráficas de usuario,” *EPrints Libr. Inf. Sci.*, p. 138, 2000.
- [62] J. G. CORDOBA DEL CASTILLO and D. F. SÁNCHEZ VILLAQUIRÁN, “PROPUESTA DE SITIO WEB INSTITUCIONAL ESPECÍFICO PARA LA CARRERA DE DISEÑO DE LA COMUNICACIÓN GRÁFICA,” 2012.
- [63] J. Brooke, P. W. Jordan, B. Thomas, B. A. Weerdmeester, and I. L. McClelland, “SUS: A quick and dirty usability scale.” pp. 189–194, 1996.
- [64] D. Fonseca, N. Martí, I. Navarro, E. Redondo, and A. Sánchez, “Using Augmented Reality and Education Platform in Architectural Visualization Evaluation of Usability and Student’s level of Sastisfaction,” *2012 Int. Symp. Comput. Educ. SIIE 2012*, pp. 1–6, 2012.



- [65] A. Bangor, P. Kortum, and J. Miller, "Determining what individual SUS scores mean: Adding an adjective rating scale," *J. usability Stud.*, vol. 4, no. 3, pp. 114–123, 2009.
- [66] S. Deterding, R. Khaled, L. Nacke, and D. Dixon, "Gamification: toward a definition," *Chi 2011*, pp. 12–15, 2011.