

**APLICACIÓN DE REALIDAD VIRTUAL EN
ARQUITECTURA – ESTUDIO DE CASO ESPACIOS
DINÁMICOS**

**JULIÁN ANDRÉS RAMÍREZ ROJAS
JHON EDWARD RODRÍGUEZ VÁSQUEZ**

**UNIVERSIDAD PILOTO DE COLOMBIA
FACULTAD DE INGENIERÍA
PROGRAMA INGENIERÍA MECATRÓNICA
BOGOTÁ D.C.
2015**

**APLICACIÓN DE REALIDAD VIRTUAL EN
ARQUITECTURA – ESTUDIO DE CASO ESPACIOS
DINÁMICOS**

**JULIÁN ANDRÉS RAMÍREZ ROJAS
JHON EDWARD RODRÍGUEZ VÁSQUEZ**

Proyecto de grado

Directores

**Prof. Doc. Oscar Fernando Avilés Sánchez
MSc. Rubén Darío Hernández Beleño**

**UNIVERSIDAD PILOTO DE COLOMBIA
FACULTAD DE INGENIERÍA
PROGRAMA INGENIERÍA MECATRÓNICA
BOGOTÁ D.C.
2015**

AGRADECIMIENTOS

Agradezco a cada una de las personas que de alguna manera contribuyeron para que este trabajo fuera presentado, especialmente a mi familia quienes siempre me apoyaron y alentaron para seguir adelante.

Julián Andrés Ramírez Rojas

Agradezco a aquellas personas que participaron ya sea de forma directa o indirecta durante el transcurso del proyecto, Principalmente a las arquitectas Laura Camila Rodríguez y Yeraldine Artunduaga quienes con su conocimiento y colaboración permitieron el desarrollo de este proyecto, de igual forma a nuestros tutores Prof. Doc. Oscar Fernando Avilés y MSc. Rubén Darío Hernández quienes guiaron este trabajo de grado por buen camino a mi padre quien contribuyó en mi educación profesional como ingeniero permitiendo culminar este proceso.

Jhon Edward Rodríguez Vásquez

Índice general

1. INTRODUCCIÓN	9
1.1. ANTECEDENTES	9
1.1.1. Evolución urbana de la ciudad de Bogotá	10
1.1.2. Conflicto Colombiano: Desplazamiento forzado	10
1.2. PLANTEAMIENTO DEL PROBLEMA	12
1.3. JUSTIFICACIÓN	14
1.4. ESTADO DEL ARTE	15
1.4.1. Living roof	15
1.4.2. ComMuniQue Vivienda adaptable	15
1.4.3. 3D Print canal house	16
1.4.4. Primer workshop de realidad virtual en arquitectura	17
1.4.5. Micro apartamentos en Nueva York	18
1.5. OBJETIVOS	19
1.5.1. OBJETIVO GENERAL	19
1.5.2. OBJETIVOS ESPECÍFICOS	19
2. MARCO TEÓRICO	20
2.1. INTRODUCCIÓN	20
2.1.1. Smart cities	20
2.1.2. Domótica	20
2.1.3. Flexibilidad en la vivienda	21
2.1.4. Realidad virtual	21
2.2. MARCO HISTÓRICO	22
2.2.1. Casas Japonesas estilo <i>Sukiya</i>	22
2.2.2. Muebles multiplicación: <i>Murphy bed</i>	23
2.2.3. Cubismo transformación del espacio y la arquitectura	24

2.2.4.	Unidad de vivienda	27
2.2.5.	Rotor house	27
2.2.6.	Viviendas crecientes	28
2.3.	ESTUDIO CASO ESPACIOS DINÁMICOS	29
2.3.1.	Wall beds	35
2.3.2.	Paredes plegables	35
2.3.3.	Sala de juntas	36
2.3.4.	Blackout	37
3.	INGENIERÍA DEL PROYECTO	38
3.1.	METODOLOGÍA	38
3.1.1.	Análisis	38
3.1.2.	Factibilidad	38
3.1.3.	Modelado	39
3.1.4.	Conectividad e Interacción	39
3.1.5.	Visualización y Recorrido	39
3.2.	PARÁMETROS DE ELECCIÓN DE SOFTWARE	40
3.2.1.	Modelado	40
3.2.2.	Texturas	41
3.2.3.	Motor gráfico	41
3.2.4.	Herramientas de configuración	41
3.3.	PARÁMETROS DE ELECCIÓN DE HARDWARE	42
3.3.1.	Gafas de realidad virtual	42
3.3.2.	Controlador de movimiento	42
3.3.3.	Comunicación inalámbrica	42
3.4.	ARQUITECTURA PROPUESTA	43
3.4.1.	Arquitectura de software	45
3.4.2.	Arquitectura de hardware	48
3.5.	DESARROLLO DEL SOFTWARE	51
3.5.1.	Diseño del mundo virtual	52
3.5.2.	Unity	61
3.6.	DESARROLLO DEL HARDWARE	75
3.6.1.	Caracterización Oculus rift DK2	76
3.7.	INTEGRACIÓN	78
3.7.1.	Comportamiento de hardware en la realidad virtual	79

4. VALIDACIÓN EXPERIMENTAL	81
4.1. RECORRIDO VIRTUAL	82
4.2. INTERACCIÓN	85
4.3. PERCEPCIÓN	85
5. RESULTADOS Y CONCLUSIONES	87

Índice de figuras

1.1. Crecimiento urbano de Bogotá.	10
1.2. Cambio de población urbana en Colombia	11
1.3. Espacio necesario según la posición del cuerpo	12
1.4. Estructura interna de un <i>Living room</i>	15
1.5. Viviendas adaptables del proyecto ComMuniQue	16
1.6. Módulos de la primer 3D print canal house	17
1.7. Ganador proyecto adAPT Edificio My micro NY	18
2.1. Casa Japonesa estilo <i>sukiya era edo</i>	23
2.2. Muebles Muntifunción del siglo XVIII	24
2.3. Georges Braque, El clarinete-collage 1913	25
2.4. Theo Van Doesburg, relación de planos horizontales 1920.	26
2.5. Kazimir Malevich Arquitectón con fuerte influencia cubista1920.	26
2.6. Boceto de una <i>Rotor House</i> hecho por Luigi Colani	28
2.7. Casa living roof, Gelnhausen 2005.	29
2.8. Vivienda equipada con espacios dinámicos	30
2.9. Plano entregado por el diseñador del apartamento	31
2.10. Sección modular con espacios flexibles	32
2.11. Planos obtenidos luego de las modificaciones	34
2.12. Corte del plano, funcionamiento de las Wall beds.	35
2.13. Corte del plano, funcionamiento de las paredes plegables.	36
3.1. Metodología usada en el proyecto	40
3.2. Arquitectura del proyecto	44
3.3. Arquitectura de software, etapa de modelamiento del espacio virtual	46
3.4. Arquitectura de software, etapa de conexión y configuración de dispositivos	47
3.5. Arquitectura de software, etapa de integración y compilación	48

3.6. Arquitectura de hardware, eje visualización gafas VR.	49
3.7. Arquitectura de hardware, eje movimiento joystick.	50
3.8. Arquitectura de hardware, eje comunicación con dispositivo móvil.	51
3.9. Vista isométrica de las paredes del apartamento en <i>Autocad</i>	54
3.10. Modelo realizado 3D max para culminar la base del apartamento	55
3.11. Modelos hechos en <i>Inventor</i> de los actuadores propuestos	57
3.12. Dona glaseada con chispas de dulce	58
3.13. Etapas animación de Wall bed multifunción en <i>3D max design</i>	60
3.14. Crear Nuevo proyecto en <i>Unity</i>	61
3.15. Interfaz grafica de Unity.	62
3.16. Exportar archivos en formato <i>FBX</i>	63
3.17. Programa CrazyBump para crear texturas.	64
3.18. Habitación principal con modelos 3D en Unity.	65
3.19. Animaciones en Unity.	66
3.20. Tipos de luces presentes en Unity.	67
3.21. Sistema de partículas en Unity.	67
3.22. Manejo de vídeo y sonido en Unity.	68
3.23. Manejo de personaje	69
3.24. Plantilla de modificacion del personaje	69
3.25. Extensiones OVR en Unity	70
3.26. Como crear un <i>script</i>	71
3.27. Diagrama de flujo, enlace animaciones-click del mouse	72
3.28. Diagrama de flujo, Controlador de animaciones	73
3.29. Diagrama de flujo, comunicación proyecto-tablet	74
3.30. Compilar en formato Oculus	75
3.31. Primer contacto con las gafas VR y el universo virtual	77
3.32. Demo, vectores limites entre sensor y gafas VR	78
3.33. Integración de hardware y software, conexión fisica	79
4.1. Punto de partida en apartamento virtual	81
4.2. Cocina apartamento virtual	82
4.3. Sala social apartamento virtual	83
4.4. Habitación principal apartamento virtual	83
4.5. Área de trabajo apartamento virtual	84
4.6. Cielo visible en el apartamento	86

Índice de cuadros

- 1.1. Área total aprobada para vivienda en m^2 para 88 municipios y Bogotá. 13

RESUMEN

En esta monografía de proyecto de grado se presenta un modelo arquitectónico de vivienda provisto con espacios modulares flexibles, los cuales permiten en un mismo espacio tener múltiples ambientes aumentando la versatilidad y utilidad de los mismos. Este modelo arquitectónico se sustenta bajo la necesidad de reducir el crecimiento desmedido de las ciudades usando los diferentes recursos tecnológicos que se encuentran a disposición en el mundo de hoy, de igual manera se exponen los actuadores que pueden brindar el dinamismo necesario a los diferentes espacios. Con el fin de mostrar el potencial de este tipo de vivienda se realiza una integración tecnológica entre varios componentes de software y hardware para finalmente desarrollar una herramienta de realidad virtual la cual permite desplazarse e interactuar con la vivienda flexible.

PALABRAS CLAVE

Crecimiento urbano, Espacios flexibles, Integración tecnológica, Realidad virtual.

ABSTRACT

On the current monograph's grade present an architectural housing model, it's provide with dynamic spaces, flexibly and modular rooms, they allow on the same space, have numerous environments, increasing their versatility and utility. This model is support under the necessity's cities per reduce the urban growth using different technological gadgets on the current world, also shows the distinctive actuators that could give dynamism to different spaces. In order that the project can be present in the proper way, it involve with various kinds of technologies that need components from software and hardware with the target of develop a tool of virtual reality that would permit travel and interact with the home designed.

KEYWORDS

Urban growth, Flexibly spaces, Technological integration, Virtual reality.

Capítulo 1

INTRODUCCIÓN

1.1. ANTECEDENTES

Las ciudades y su crecimiento dependen evidentemente de la cantidad de habitantes que en ellas se encuentren, creciendo año tras año y agregando nuevas periferias a ella debido a la necesidad de mano de obra, la tasa de natalidad, personas que se desplazan por necesidad, entre otros; Esto conduce a un crecimiento desmedido y sin control, como lo plantea Le Corbusier en 1962 con su libro *la ciudad del futuro*, donde critica fuertemente las ciudades por su crecimiento en extensión y no en altura, tomando en múltiples casos como ejemplo el *Wall Trade Center* de Manhattan como referente de crecimiento en altura, llegando a debatir las propuestas urbanísticas de la época como la casa de familia de planta baja dispersa en la naturaleza, afirmando que estas son:

«la experiencia de este ultimo medio siglo que ha llevado a las ciudades a un callejón sin salida y a las sociedades que las habitan al desorden mas amenazador.»[7]

De igual manera propone cambiar el enfoque planteado por occidente de utilizar construcciones principalmente para aspectos laborales y llevarlo a la vivienda o el comercio también. posteriormente se adoptarían estas ideas haciendo múltiples ejemplos alrededor del mundo transformando las ciudades y dando una solución parcial al crecimiento de estas.

1.1.1. Evolución urbana de la ciudad de Bogotá

El crecimiento urbano en Bogotá se intensificó hasta poco antes de la mitad del siglo XX cuando la ciudad empezó a tener características de las ciudades modernas como un centro altamente transitado, edificios y partes dedicadas únicamente a la industria, el crecimiento residencial en altura en Bogotá inicio con los edificios hechos por Leopoldo Rother al rededor de la ciudad universitaria para acoger a profesores y alumnos[11]. En los siguientes años Bogotá se extendería incorporando en promedio 309 hectáreas por año [?] (ver Figura 1.1).

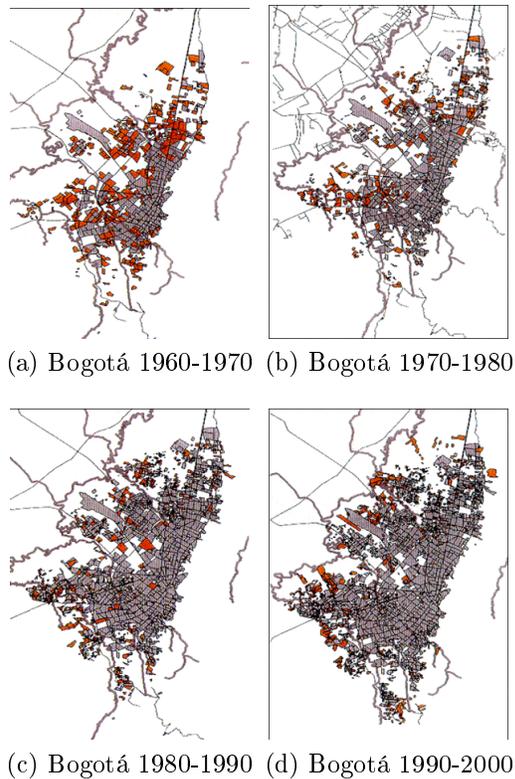


Figura 1.1: Crecimiento urbano de Bogotá.

1.1.2. Conflicto Colombiano: Desplazamiento forzado

La duración del conflicto en Colombiano lo ha convertido en el país con la guerra interna sin negociar mas antigua del mundo, produciendo que ambas partes de este

se vean afectadas a gran manera. En el ámbito social los grupos al margen de la ley han provocado el desplazamiento forzado de familias de campesinos, diferentes grupos étnicos e incluso de poblaciones pequeñas (4500 habitantes), solo hasta el año 1997 fue posible tener un estimado de la magnitud de este problema en Colombia, arrojando una cifra de 330,000 personas por año desplazadas entre 1996 – 2003[9] posteriormente la cifra decaería debido a las acciones tomadas por el estado en una ofensiva contra los diferentes grupos armados por el territorio nacional.

La movilización forzada del pueblo colombiano se vio reflejada en las grandes ciudades departamentales del país como Bogotá, Medellín, Cali y Barranquilla. Esto creo una necesidad de vivienda económica y accesible de fácil implementación, La base de datos GapMinder permite apreciar la evolución de la población en las áreas urbanas, de ella se destaca que, en los 60's esta relación en Colombia estaba alrededor del 45 % (ver Figura 1.2)[?].

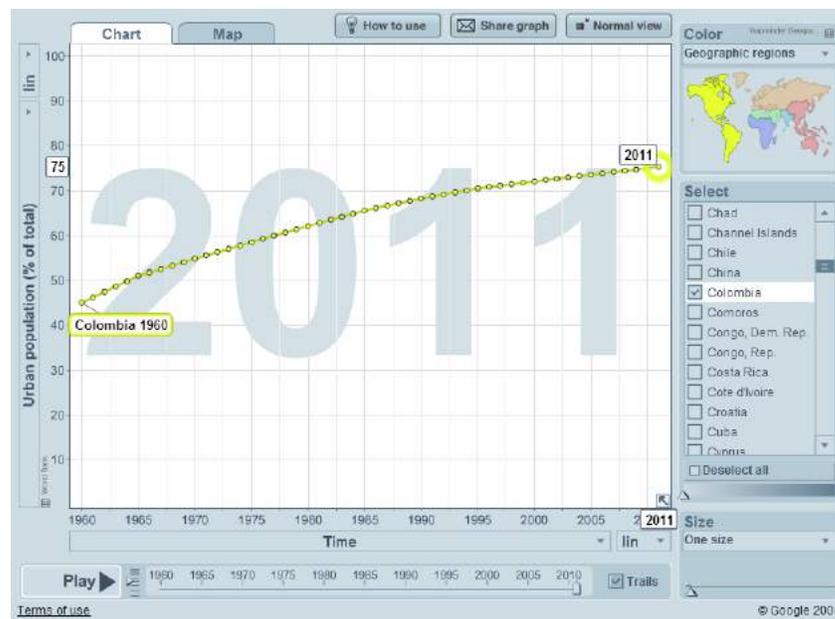


Figura 1.2: Cambio de población urbana en Colombia

1.2. PLANTEAMIENTO DEL PROBLEMA

Actualmente los espacios de las viviendas se diseñan cumpliendo el mínimo establecido, quedando al borde de lo necesario tal como se muestra en el manual *Arte de proyectar en arquitectura* hecho por Peter Neufert donde el establece todas las medidas mínimas necesarias, por ejemplo, la altura mínima de un apartamento debe ser 2,2 metros ya que una persona promedio al levantar las manos alcanza esa medida (Ver figura 1.3) [17]. Al limitarse a cumplir las medidas mínimas se deja por fuera las necesidades particulares que tenga el habitante, si el propietario de la vivienda tuviera un gusto peculiar o extraño como el arte o la culinaria no podría desarrollarlo en esta vivienda estrecha y rígida, que lo obliga a adaptarse a ella.

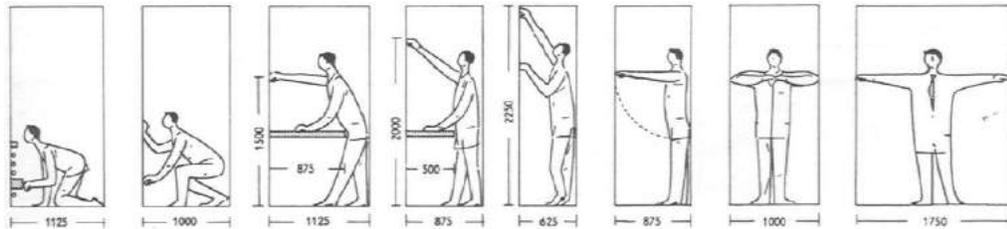


Figura 1.3: Espacio necesario según la posición del cuerpo

En este momento lo que se busca el país para sus viviendas es lograr un mayor crecimiento en altura, esto se ve demostrado por los datos brindados por el Departamento Administrativo Nacional de Estadística (DANE) para el mes de Enero de 2015 donde podemos apreciar que en Colombia del total de m^2 aprobados para construcción el 74,64% son apartamentos, para el caso de Bogotá este porcentaje asciende a 91,97% [8]. Sin embargo esto obliga a los habitantes a la conformidad de espacios reducidos en área, siendo poco cómodos e inconfortables para los mismos. La adaptabilidad y la flexibilidad es una necesidad de la sociedad, lograr resolver un problema de crecimiento urbano por medio de una vivienda flexible cómoda y confortable para que el usuario salga de la rigidez y acoplamiento al que está siendo sometido actualmente.

2015 Departamentos y Bogotá	Metros cuadrados					
	Vivienda de interés social			Vivienda diferente de VIS		
	Total	Casas	Aptos.	Total	Casas	Aptos.
Antioquia	286.710	9.566	277.144	2.175.442	318.065	1.857.377
Arauca	32.162	10.489	21.673	12.439	11.260	1.179
Atlántico	472.809	61.291	411.518	1.264.293	126.808	1.137.485
Bogotá	1.062.616	138.984	923.632	2.541.556	150.527	2.391.029
Bolívar	334.800	252.982	81.818	290.067	57.465	232.602
Boyacá	101.253	6.544	94.739	508.345	175.587	332.758
Caldas	82.584	6.427	76.157	169.718	85.066	84.652
Caquetá	1.819	1.819	-	41.392	27.421	13.971
Casanare	50.947	45.558	5.389	138.037	85.244	52.793
Cauca	27.536	27.168	368	157.261	100.984	56.277
Cesar	20.530	20.530	-	139.538	106.605	32.933
Córdoba	13.938	12.791	1.147	182.536	87.403	95.133
Cundinamarca	1.001.850	78.244	923.606	1.032.851	631.614	401.237
Chocó	2.048	1.726	322	11.704	7.240	4.464
Huila	66.099	8.076	58.023	247.900	142.305	105.595
La Guajira	1.033	1.033	-	33.746	21.624	12.122
Magdalena	32.025	2.610	29.415	155.533	55.088	100.445
Meta	155.898	91.595	64.303	213.798	147.272	66.526
Nariño	143.982	18.054	125.928	298.253	87.880	210.373
Norte de Santander	171.713	41.602	130.111	174.056	96.473	77.583
Quindío	250.675	10.475	240.200	91.103	47.899	43.204
Risaralda	165.538	36.720	128.818	409.423	222.750	186.673
Santander	246.574	25.240	221.334	1.052.639	142.133	910.506
Sucre	3.373	3.373	-	102.884	28.969	73.915
Tolima	224.544	54.033	170.511	414.757	147.540	267.217
Valle del Cauca	418.701	131.401	287.300	987.869	410.774	577.095
Total	5.371.757	1.098.301	4.273.456	12.847.140	3.521.996	9.325.144

FUENTE: DANE

- Sin Movimiento

Cuadro 1.1: Área total aprobada para vivienda en m^2 para 88 municipios y Bogotá.

Por lo anterior la pregunta de investigación es:

¿Qué posibilidad hay de incluir espacios flexibles en viviendas adaptables para el uso de múltiples áreas habitables en un apartamento?

1.3. JUSTIFICACIÓN

El crecimiento de las ciudades no afecta solo a aquellas personas que en ella habitan, también se ven perjudicados los animales y la vegetación que antes crecían libremente en esos lugares que se les fueron arrebatados y en los cuales ahora solo hay concreto, acero y vidrio; Lo cual claramente es un problema preocupante así lo demuestran los programas de previsión de la Organización de las Naciones Unidas (ONU) los cuales aseguran que para 2050 el 70% de la población mundial vivirá en entornos urbanos y de no encontrar un equilibrio entre los ámbitos sociales, ambientales e incluso entre los propios ciudadanos pueden haber graves consecuencias[12]. Es importante destacar que en Colombia para el año 2011 alrededor del 75% de la población ya vivía en las ciudades (ver figura 1.2).

La tasa de licenciamiento de construcción en Bogotá se había mantenido desde 2006 sobre el promedio anual de 5'400.000 m^2 , esto cambio en 2014 cuando se habilitaron 6'225.119 m^2 (promedio anual) para construcción, así lo comento la subsecretaria de la secretaria distrital de planeación Liliana Ospina [?], esto demuestra el gran interés por parte de la administración local por hacer nuevas viviendas y aumentar la calidad de vida. Ahora, si en estos metros cuadrados que se agregan a la ciudad se implementaran espacios modulares de arquitectura flexible tanto para vivienda como para oficinas se conseguiría mermar la cantidad necesaria de m^2 al menos en un 35%, esto representaría 2'178.791 m^2 que no se agregarían a la ciudad, lo que en cinco años de continua implementación (asumiendo que la demanda de territorio se mantenga en un rango similar) equivale a 10'893.955 m^2 o en una escala comparable 9.6 parques Simón Bolívar [?]. El tiempo de construcción para este tipo de viviendas flexibles puede variar dependiendo de diferentes factores como los materiales a usar, el área con que se cuenta para la vivienda, las condiciones del terreno, etcétera.

Las tecnologías emergentes permiten retar los límites que antes solo la ciencia ficción se había atrevido a proponer, esta es una de las facilidades que brinda la ingeniería mecatrónica ya que por sus círculos informativos se esta al tanto de todas estas nuevas tecnologías que permiten integrarse entre si para crear nuevas aplicaciones, este es el caso de la realidad virtual que en un principio fue desarrollada para ampliar la inmersividad en los video juegos [20], pero que rápidamente ha adquirido mas enfoques a partir de la adaptación y integración tecnológica. La particular mezcla Realidad Virtual, software de diseño para arquitectura y domótica promete buenos resultados, siendo una forma realmente novedosa para exponer un nuevo concepto de hábitat.

1.4. ESTADO DEL ARTE

1.4.1. Living roof

De la mano de la misma empresa que ideó la capsula de experiencias *I-COCOON*, en mayo de 2011 proponen una suite autosustentable para terrazas llamada *living room*, esta permite tener múltiples ambientes en un espacio reducido sin perder la comodidad de una habitación de hotel, siendo así, una opción muy atractiva para viajeros de paso o simplemente para aquellos ciudadanos que quieran un momento de relajación en el techo de su casa. La empresa *NAU* se caracteriza por hacer espacios diferentes que cuentan historias mediante la innovación, por ello es que deciden presentar la *living room* como: espacio de relajación, espacio de trabajo y dormitorio; Todo en un mismo lugar, esto se logra gracias a un anillo central que gira cierto ángulo moviendo con él los diferentes espacios (Ver Figura 2.7)[10].

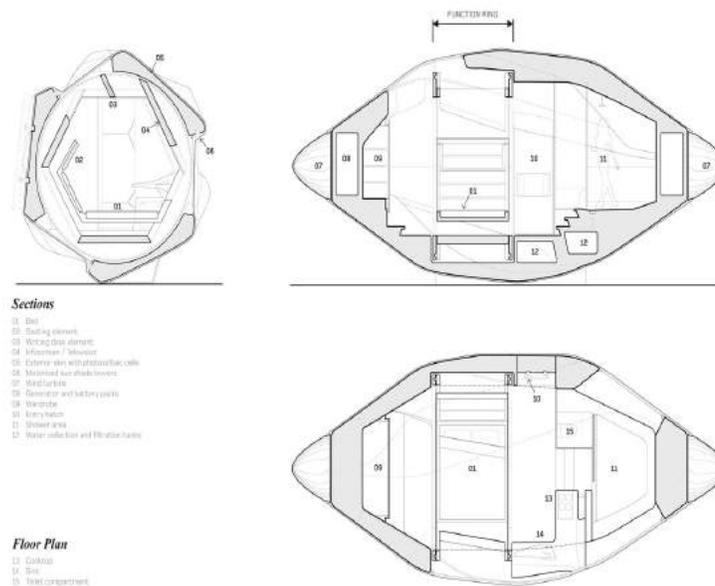
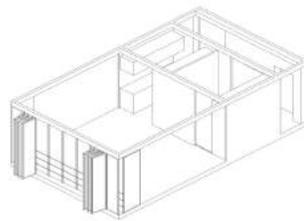


Figura 1.4: Estructura interna de un *Living room*

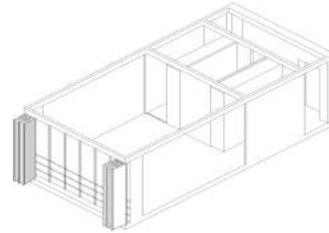
1.4.2. ComMuniQue Vivienda adaptable

En el año 2013 la empresa ecuatoriana GMA arquitectos asociados decide participar en el concurso europeo de arquitectura *EUROPAN 12: la ciudad adaptable*, este es un

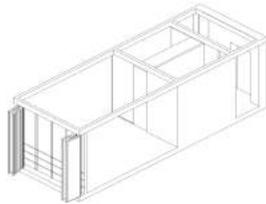
concurso que motiva a jóvenes emprendedores a participar en el y ayudar en los nuevos problemas que surgen en cada ciudad europea. GMA propone un proyecto de vivienda multifamiliar adaptable en la ciudad de Nürnberg Alemania, la vivienda propuesta se basa en un análisis previo de las necesidades de los habitantes y se plantea como objetivo adaptarse a las diferentes variables que afectan una vivienda normalmente, como la cantidad de personas que circulan por los diferentes espacios que en ella se encuentran, creando espacios lo suficientemente versátiles como para incorporar en un mismo espacio viviendas, comercios, oficinas, servicios, etc[3].



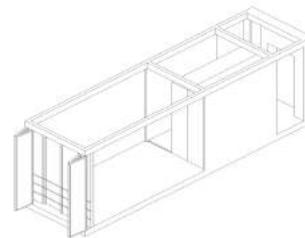
(a) Modulo Grande



(b) Modulo medio-grande



(c) Modulo medio-pequeño



(d) Modulo pequeño

Figura 1.5: Viviendas adaptables del proyecto ComMuniQue

1.4.3. 3D Print canal house

Llevando la flexibilidad a un limite nunca antes visto, en el año 2014 la empresa de arquitectos *DUS* realiza la primera casa hecha totalmente con una impresora 3D, creando la posibilidad de hacer viviendas a medida de forma modular, similares a las casas

prefabricadas pero esta tiene la posibilidad adaptarse fácilmente a cambios circunstanciales, de igual forma permite hacer múltiples piezas idénticas o hacer piezas únicas. La impresora recibe el nombre de *KamerMaker*, el material utilizado es un bioplástico granulado de nombre *Macromelt* que es llevado a 170°C para poder ser manejado, este es un tipo de pegante industrial desarrollado por Henkel el cual contiene un 80 % de aceite vegetal[2].

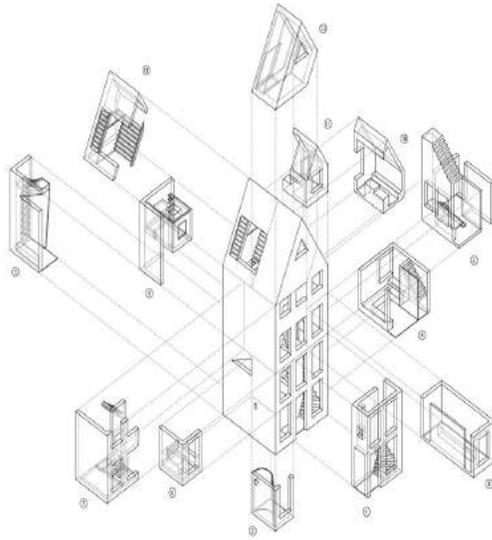


Figura 1.6: Módulos de la primer 3D print canal house

1.4.4. Primer workshop de realidad virtual en arquitectura

Uno de los temas de continuo estudio a nivel mundial es la integración de nuevas tecnologías, es por esto que la facultad de arquitectura y urbanismo de la universidad de Chile en enero de 2015 convocó a docentes, egresados y estudiantes a participar en el primer *workshop* de realidad virtual hecho en el país. Dotados con gafas de realidad virtual *oculus rift* y computadores con alta capacidad en gráficos, nueve estudiantes dirigidos por el profesor Mauricio Layola y un ex-alumno egresado experto en visualización arquitectónica Max Daiber desarrollaron llamativos modelos arquitectónicos en los cinco días que duró esta edición de *workshop*, esto se logró gracias a la integración de tecnologías propuesta por el profesor Layola (lentes de realidad virtual, programas de

navegación y programas de arquitectura) consiguiendo recrear desde lugares antiguos hasta sitios ficticios[13].

1.4.5. Micro apartamentos en Nueva York

En el año 2012 el departamento de vivienda, conservación y desarrollo de la ciudad de Nueva York lanzo un proyecto piloto, una competencia de nombre *adAPT*[18], la cual tenia como objetivo crear un nuevo modelo de vivienda para poder adaptarse a los cambios demográficos de la ciudad. En enero de 2013 fue publicado el ganador de la competencia, la propuesta vencedora fue *My Micro NY* hecha por *nACHITECTS*, *Monadnock Development LLC* y *Actors Fund Housing Development Corporation*, esta se compone de 55 apartamentos con áreas entre 25 y 35 metros cuadrados, fueron diseñados como espacios modulares prefabricados, ya que el tamaño de cada vivienda es reducido se proveerán los apartamentos con muebles multifuncionales para comedor, habitación principal y sala. la construcción de los apartamentos se inicio en la primavera de 2014 en la isla de Manhattan (Kips bay) y esta proyectado para ser entregado en septiembre de 2015[16].



Figura 1.7: Ganador proyecto adAPT Edificio My micro NY

1.5. OBJETIVOS

1.5.1. OBJETIVO GENERAL

Desarrollar una herramienta computacional mediante el uso de realidad virtual, que permita estudiar espacios modulares, con paredes plegables, wall beds, mesas abatibles entre otros, como herramienta para análisis de espacios dinámicos en el área de arquitectura.

1.5.2. OBJETIVOS ESPECÍFICOS

- Diseñar un ambiente virtual, tipo vivienda, mediante el uso de herramientas CAD con el fin de montar un espacio habitacional para la validación de la propuesta en el sistema simulado.
- Diseñar la arquitectura de Software y Hardware para la implementación en el sistema virtual.
- Simular un apartamento con espacios dinámicos, de manera que se permita tener una visualización 3D del espacio objeto estudiado.
- Conseguir una interactividad con el ambiente simulado de manera que se permita inmersión en la realidad virtual.

Capítulo 2

MARCO TEÓRICO

2.1. INTRODUCCIÓN

El proyecto de grado se ve envuelto en una serie de términos teóricos que le dan relevancia en torno al mundo académico e investigativo, estableciéndolo sobre unas bases solidas que brindan credibilidad; Lo anterior implica de igual forma una revisión histórica de los hechos que marcaron un punto de referencia en la arquitectura y por ende en la vivienda.

2.1.1. Smart cities

Uno de los conceptos que demuestran la preocupación de la sociedad por corregir los errores que por su permanencia en el planeta se han creado a lo largo de los años, de estos conceptos el que compete tratar por su relación con las viviendas es el de *Smart cities* que la Asociación española para la calidad (AEC) define como:

«aquella que utiliza los avances tecnológicos como soporte y herramienta para mejorar la calidad de vida de sus habitantes. Además tiene presente el concepto de eficiencia energética y sostenibilidad persiguiendo un equilibrio entre el medio ambiente y el consumo de los recursos naturales.»[12]

2.1.2. Domótica

El termino domótica viene del latín *domus*, casa, e *informática* y hace referencia a la automatización de viviendas mediante recursos tecnológicos, los cuales cada vez cubren una parte mayor de las necesidades del hogar empezando con el control de temperatura

en los años 70, dispositivos de seguridad y controladores de luz en interiores en los 80, la conectividad gracias al Internet en los 90 y finalmente en los últimos años se han logrado grandes avances con la eficiencia energética en las vivienda [15].

2.1.3. Flexibilidad en la vivienda

La flexibilidad vista desde la perspectiva ingenieril puede relacionarse fácilmente con el término *modulo de elasticidad*, que establece la propiedad intrínseca en cada material para resistirse a la deformación [5], es decir, la cualidad de un material que dada la necesidad de exigirse (mediante esfuerzos sobre el) puede cambiar de forma y luego volver a su estado inicial. Aplicando este termino en el interior de una vivienda los materiales flexibles pasarían a ser aquellos elementos que Brinden cierta facilidad de cambio sin ser estos permanentes en el hogar; por ejemplo: camas abatibles, asientos plegables o muros corredizos.

2.1.4. Realidad virtual

La realidad virtual es un tema que nace en la época moderna siendo definido de muchas maneras en la década de los 90's, la mayoría de las cuales establecen un punto en común, el uso de computadoras para crear esta realidad alterna con base en la manipulación de los sentidos, a la vez también son planteados los objetivos a cumplir por una buena interfaz de realidad virtual, dados por Frank Biocca y Mark Levy son:

«la inmersión completa de los canales sensomotores humanos en una experiencia vital generada por ordenador»[4]

La realidad virtual se puede clasificar en tres tipos de experiencia:

- Pasivo: Son aquellos que ofrecen inmersión virtual sin la posibilidad de interacción, similar a lo percibido en los simuladores de montañas rusas que encontramos en centros comerciales.
- Exploratorio: Permiten el desplazamiento en el entorno virtual con el fin de explorar y descubrir lo que allí se puede encontrar, ampliamente usado en museos virtuales y demostraciones arquitectónicas.
- Interactivo: La experiencia virtual interactiva no solo da la posibilidad de explorar también permite experimentar, cambiando y modificando aspectos de forma en el modelo, por ejemplo los vídeo juegos.

2.2. MARCO HISTÓRICO

La necesidad de una vivienda confortable ha sido una gran fuente de inspiración mundial, si se pensara en la cantidad de inventos novedosos que se desarrollan exclusivamente para la comodidad de las personas sería una lista muy extensa. Ahora se iniciara un recorrido histórico por los hechos que se destacan y aportan para el correcto desarrollo de esta investigación.

2.2.1. Casas Japonesas estilo *Sukiya*

La primera cultura que brindo un concepto solido de flexibilidad en la vivienda fue la civilización japonesa, específicamente en su era *edo* desarrollada entre los años 1600 y 1867, periodo de gobierno de la familia *Tokugawa*, esta etapa de la historia japonesa se caracteriza por su progreso sin influencia de países extranjeros, consiguiendo así un perfeccionamiento en su arquitectura. El modelo de hábitat que logra afanzarse en esta era y permanece fuertemente ligada a la vivienda actual tradicional japonesa, es un estilo de casa llamado *Sukiya*, la cual está hecha con pilares y piso de madera, sus paredes exteriores son móviles y reciben el nombre de *shoji* que permite el paso de luz, mientras que sus espacios internos se dividen por *fusumas* siendo puertas sobre rieles similares a los *shoji*, a diferencia de este el *fusuma* imposibilita el paso de luz y visión a través de él [6]. En la Figura 2.1. Se puede ver el interior de una casa japonesa estilo *Sukiya* donde se destaca su amplitud debido a los espacios modulares implementada en ella, convirtiendo tres cuartos individuales en un gran salón[6].



Figura 2.1: Casa Japonesa estilo *sukiya era edo*

2.2.2. Muebles multiplicación: *Murphy bed*

La aparición de muebles multiuso probablemente se dio en principios del siglo XIX con la aparición de los muebles patentados en la Inglaterra recién industrializada; Thomas Sheraton pionero en este tipo de muebles (sillas que podían convertirse en camas, mesas que podrían convertirse en escritorios altos, estanterías móviles, etcétera.) produjo y distribuyó varios modelos sin lograr mucho éxito en cuanto a moda ya que estos se vieron desplazados por los muebles europeos de la época, por ello solo hasta 1900 William Lawrence Murphy registró una patente de lo que posteriormente se conocería como *Murphy Bed* o *Wall Bed* lo que básicamente es una litera que se puede levantar a un muro y mantenerse en el, aunque se conoce de otros similares anteriormente, principalmente mostrados en exposiciones europeas de muebles. Estos muebles surgen ante la necesidad de estatus social, ya que en el siglo XIX en Europa fue un tema importante, por lo que las familias numerosas de clase media no tenían muchas veces el espacio para una sala de estar y habitaciones confortables al mismo tiempo, por ello se fabricaron estos tipos de muebles, ya que de día podría lucir como una alacena digna de una presistente familia y de noche pasar a ser una reconfortante cama para descansar (Ver figura 2.2), incluso también se llegó a combinar con otros componentes que mostraban más

estatus como un piano por ejemplo[14].



Figura 2.2: Muebles Muntifunción del siglo XVIII

2.2.3. Cubismo transformación del espacio y la arquitectura

En un contexto diferente por el año 1907 surge un nuevo tipo de arte llamado cubismo, Este se desarrolla en Francia con la ayuda de Pablo Picasso y Georges Braque quienes fueron sus principales ponentes; El cubismo y su relación con el espacio es descrita por Sigfried Giedon en su libro espacio, tiempo y arquitectura donde establece que el cubismo.

«visualiza los objetos de manera relativa: esto es, desde varios puntos de vista, ninguno de los cuales tiene autoridad exclusiva. Y diseccionando así los objetos, el cubismo los ve de forma simultanea desde todos los lados.»[14]

presentando de esta forma la multiplicidad del espacio y mostrando una nueva concepción de este. El cubismo se da por una nueva mentalidad adquirida a finales de siglo XIX y principios del siglo XX, una mentalidad de contextualización, de querer ver todo en un solo momento y apoderarse del espacio que se escapa si no se ve de esta forma absoluta, supliendo el cambio necesario de las obras renacentistas con factores planos

y poco cambiantes. En la figura ???. Se muestra una obra de Pablo Picasso llamada *Violon et guitare* perteneciente a la etapa de naturaleza muerta del año 1914, en ella se ve una correcta combinación de figuras geométricas y pocos colores vivos lo que permite ver profundidad al dejar de fondo los tonos terrosos y a medida que se acerca a la superficie resaltan las figuras más sólidas con colores fuertes. En la figura 2.3 . Se puede ver una pintura de tipo collage en óleo sobre lienzo realizada por Georges Braque en 1913 llamada *el clarinete*, en ella se aprecia figuras geométricas solidas en un ambiente claro que permite admirar las siluetas allí presentes[14].

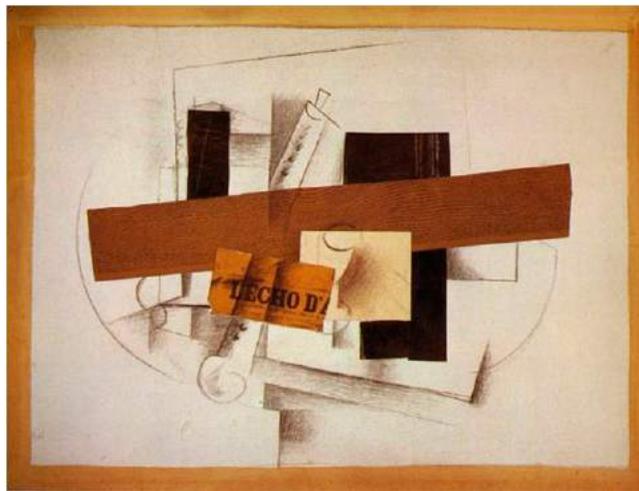


Figura 2.3: Georges Braque, *El clarinete-collage* 1913

Su aplicación en la arquitectura no se dio de forma literal, el cubismo sirvió como inspiración y así dio forma a nuevas ideas en cada lugar donde se encontrase alguien que racionalizara estas figuras; De esta manera en la segunda década del siglo XX se inicia una integración arte-arquitectura sin precedentes, como se muestra en la figura 2.4. Donde encontramos una relación planos horizontales y verticales hecha por Theo Van Doesburg arquitecto neerlandés con interés por la abstracción y geometrización, el cual utilizó este tipo de relaciones en sus construcciones que se mantendrían como legado hasta el día de hoy, también en la figura 2.5. Donde podemos ver un arquitectón hecho por el artista ruso Kazimir Malévich en el cual se pueden destacar volúmenes saliendo, mezclándose y flotando permitiendo apreciar en el su influencia cubista. Si compara este arquitecton hecho por Malévich y la obra *el clarinete* de Braque hecha

7 años antes se puede destacar varios factores, como por ejemplo una figura central sobre la que recaen todos los otros componentes, la mezcla de figuras irregulares, la importancia que toma esta pieza central haciéndola ver más grande de lo que realmente es; estos factores que más tarden se harían evidentes y con muchos ejemplos alrededor del mundo.

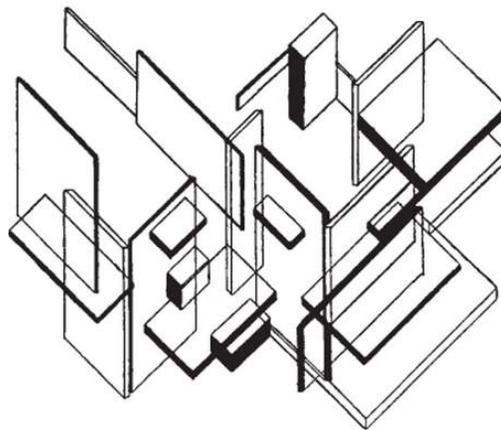


Figura 2.4: Theo Van Doesburg, relación de planos horizontales 1920.

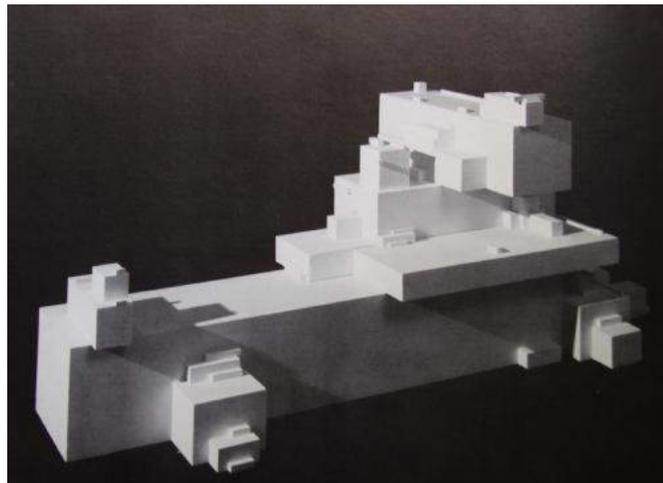


Figura 2.5: Kazimir Malevich Arquitectón con fuerte influencia cubista 1920.

2.2.4. Unidad de vivienda

Avanzando sobre la historia se encuentra el siguiente acontecimiento importante, con la primera construcción de la “*unité d’habitation*” o unidad de vivienda realizada por Le Corbusier entre los años 1947-1952 en el Boulevard Michelet, a las afueras de Marsella. Este proyecto se llevó a cabo gracias a la ayuda de Claudius Petit quien por ese entonces fue ministro de reconstrucción en Francia y defendió fuertemente el plan de construcción propuesto por Le Corbusier durante los 5 años que duro su desarrollo. La unidad de vivienda surge ante la necesidad de reconstrucción ya que se da en la etapa de post-guerra brindándole a Le Corbusier la oportunidad de ver implementados varios proyectos en los que ya venía trabajando tiempo atrás, llevando su unidad de vivienda más allá de Marsella (también se elaboró este concepto en Nantes, Berlín, Neaux, Briey-en-Fôrest). La unidad de vivienda en Marsella tiene la capacidad de acoger a 1600 personas en 337 apartamentos de 23 tipos diferentes brindando flexibilidad a los distintos tipos de familias que pueden allí vivir, se encuentran por ejemplo viviendas con una habitación hasta apartamentos para familias de 10 integrantes[19], en el proyecto desarrollado por Le Corbusier también toma gran importancia, la implicación social que se desarrolló en él, integrando de tal manera todo lo que una persona pudiera requerir en el transcurso del día como tiendas de comestibles, carnicerías, fruterías, tintorerías, peluquerías, guarderías e inclusive quiosco de periódicos. El origen de la unidad de vivienda no se da solo por la obvia necesidad de reubicar a todas las personas afectadas por el conflicto hay implicaciones más allá de eso, como el crecimiento desmedido de las ciudades, el desorden asociado a este pronto crecimiento, la poca integración de los diferentes factores en la sociedad, como el propio Le Corbusier critico en el año 1924 en su libro *la ciudad del futuro*, donde plantea un conflicto entre el desbordante crecimiento de las ciudades y su densidad población en los diferentes componentes de las mismas[7].

2.2.5. Rotor house

En el año 2004 se construye un modelo de casa denominado “*futurista*” la cual recibe el nombre de rotor house, esta llega de la mano de un diseñador alemán llamado Luigi Colani, la parte más allegada a los espacios dinámicos de este proyecto es un cilindro de

giro mecánico que contiene tres ambientes fundamentales de una vivienda (dormitorio, cocina y ducha), eliminando así espacios inutilizables como el vestíbulo. La rotor house fue diseñada para un habitante y se encuentra en una zona cuadrada de 6x6 mts un espacio relativamente pequeño para una vivienda pero que con esta implementación o “truco” como lo llama su creador, adquiere mayor amplitud. En la figura 2.6. Se muestra un boceto de la casa hecho por Luigi Colani.

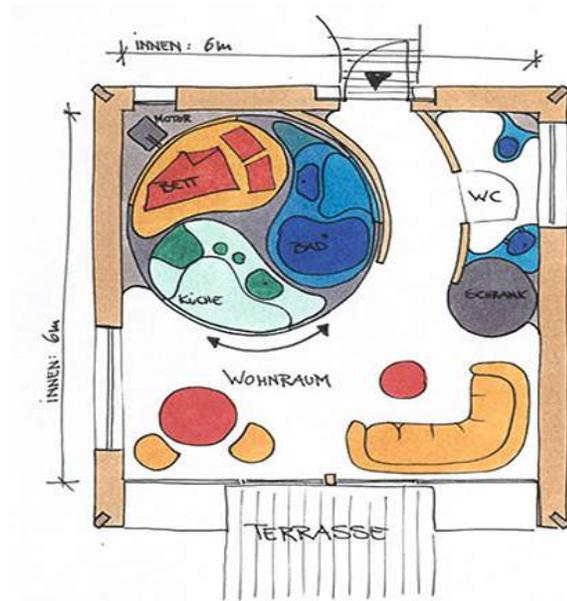


Figura 2.6: Boceto de una *Rotor House* hecho por Luigi Colani

2.2.6. Viviendas crecientes

Una moda reciente alrededor del mundo son las viviendas crecientes y decrecientes de las cuales se tiene referencia desde 2005 con un proyecto llamado living room llevado a cabo por Seifert-Stöckmann una pareja de arquitectos alemanes que hicieron una remodelación sobre una casa ubicada en el distrito histórico de Gelnhausen, lo que hace diferente a esta casa es la posibilidad de ampliar un salón central ubicado en el segundo piso (ver figura. 2.7), este puede sobresalir de la casa permitiendo ampliar el área inicial del terreno[21].



Figura 2.7: Casa living roof, Gelnhausen 2005.

Posteriormente en el año 2009 reaparecen nuevos espacios flexibles, estos ahora crecen o decrecen según la necesidad del ocupante, este proyecto fue realizado por la firma española de arquitectos Baragaño el cual tiene como objetivo brindar viviendas accesibles, cómodas y funcionales con capacidad de adaptabilidad mediante módulos, utilizando el concepto del volumen del espacio.

2.3. ESTUDIO CASO ESPACIOS DINÁMICOS

Los espacios dinámicos son aquellos lugares en donde se encuentran elementos móviles y versátiles, que permiten tomar múltiples identidades a los ambientes en el que se encuentran, esto concuerda con la definición aristotélica del espacio [1], donde se expone que un lugar se crea a partir de lo que en el se encuentra, es decir, como un único árbol no puede recibir el nombre de bosque, por que solo uno no es suficiente para crearlo.

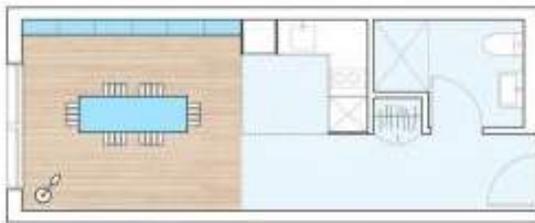
Actualmente se esta implementando de manera poco generalizada la micro vivienda, como se vio en la sección 1.4.5, donde los gobiernos metropolitanos toman estas iniciativas mediante concursos arquitectónicos para vivienda, en la Figura 2.8 se muestra el diseño ganador del concurso realizado en Nueva York, donde se puede apreciar que en la parte izquierda se encuentra un ambiente cambiante que contiene tres espacios habitacionales en el, aumentando el área en la vivienda no de forma literal pero si equivalente

(su área equivalente es alrededor de 140 %).



APT. #3A - DAY

(a) Modo día



APT. #4D - DUSK

(b) Modo anochecer



APT. #4D - NIGHT

(c) Modo noche

Figura 2.8: Vivienda equipada con espacios dinámicos

Hasta el momento solo se han encontrado referencias de espacios flexibles aplicados en la vivienda, dejando de lado otros ámbitos y sectores sobre los que también se pueden aplicar, por lo anterior se propone un cambio a esta tendencia presentando un apartamento con espacios dinámicos sobre el que puede funcionar una oficina y vivir confortablemente una familia de cinco personas, de esta forma se puede ofrecer

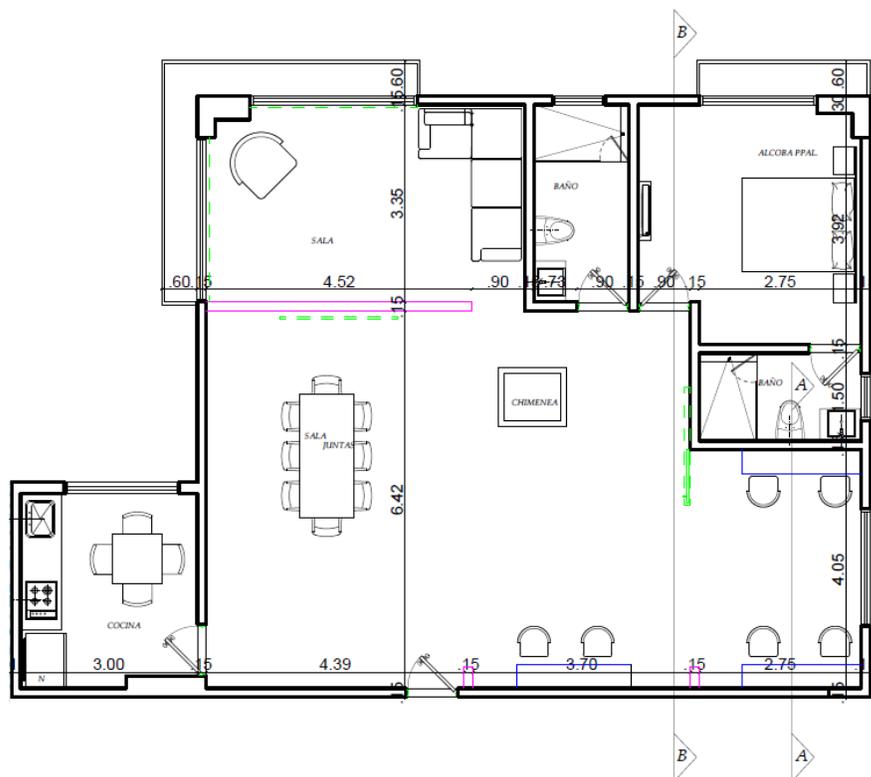
las tendencias de los espacios flexibles, se presenta un apartamento que durante el día puede funcionar como oficina y al terminar la jornada laboral se transforma en una vivienda convencional, lo anterior implica ciertas modificaciones en el plano inicial, las modificaciones cubren tres aspectos fundamentales: darle movilidad a las paredes, muebles multiuso y objetos que se pueden reducir, esto se plantea para que únicamente abarque $83,34m^2$ de la vivienda, lo que equivale al 68 % del área privada, en la figura 2.10 se pueden ver todas las zonas de la casa y en ellas se encuentra una sección achurada que corresponde a las zonas modificables propuestas en la vivienda.



Figura 2.10: Sección modular con espacios flexibles

Luego de llevar a cabo las modificaciones necesarias al apartamento se consiguen dos planos diferentes, uno para cada faceta de la vivienda (ver figura 2.11) en ellos se encuentran resaltados en colores los diferentes objetos que tienen dinamismo en la vivienda, en color morado se encuentran 3 paredes plegables, de color azul se hallan 3 literas que pueden ser levantadas sobre el muro contiguo a ellas (también conocidas como *Wall bed's*), También se encuentra una sala de juntas en color rojo ubicada en el centro del área social del apartamento, último se encuentran de color verde aquellos complementos que brindan privacidad a espacios tanto en la oficina como en la vivienda.

La parte flexible de la vivienda luego de la modificación contendría en los mismos $83,34m^2$ once ambientes diferentes al sumar los que se encuentran en la oficina y la vivienda, aumentando la productividad del espacio utilizado.



(a) Modo oficina en apartamento con espacios dinamicos



(b) Modo vivienda en apartamento con espacios dinamicos

Figura 2.11: Planos obtenidos luego de las modificaciones

2.3.1. Wall beds

En los espacios propuestos se encuentran tres camas que cumplen la función de muebles multiuso, estos pueden adquirir tres modos diferentes y cada uno para ambientes totalmente distintos, para que la transformación de cada uno se dé automáticamente se propone el uso de múltiples actuadores, el primer modo de estas camas es precisamente el de dormitorio, donde se encuentra un cubo hueco liviano en el que se halla un colchón convencional, este cubo permite ser levantado mediante dos actuadores neumáticos que realizan fuerza suficiente sobre una pestaña saliente en el lateral de la cama que esta junto a la pared para plegarla sobre el muro, con este último movimiento se da lugar al segundo modo del mueble, con él se obtiene un espacio amplio en el que no se encuentran grandes objetos permitiendo comprobar visualmente la amplitud de la vivienda y dando la posibilidad de un lugar ahora libre; El tercer modo permite desplegar un escritorio de la parte inferior de la litera, por lo que este modo únicamente es posible cuando la cama ya se encuentra sobre el muro, el escritorio se soporta sobre una bisagra que forma un triángulo donde base y altura se apoyan en la cama y escritorio respectivamente (Ver figura 2.12).

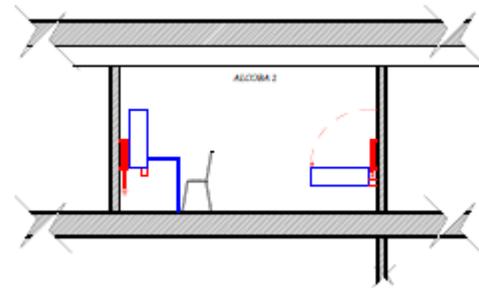


Figura 2.12: Corte del plano, funcionamiento de las Wall beds.

2.3.2. Paredes plegables

El área del apartamento elegida para implementar los espacios dinámicos cuenta con tres paredes plegables que limitan o amplían el espacio a voluntad de los ocupantes,

con el fin de lograr extender y encoger dichas divisiones se propones que, por medio de un sistema electro-mecánico que consta de rieles dentados, motorreductores, bisagras, pasadores y rodamientos, estos elementos se enlazan de la siguiente forma, en ambos extremos de cada panel a excepción del ultimo (el panel exterior no necesita bisagras en uno de sus extremos ya que es el ultimo) se deben poner las bisagras necesarias para permitir el dobléz del muro de forma segura, las partes del muro que recorren los rieles son aquellas que al estar la pared encogida se encuentran en un mismo costado sea izquierdo o derecho, de este costado elegido, los extremos de cada modulo se enlazaran en piso y techo a los rieles guía, sobre los cuales se llevara a cabo el movimiento por medio de un par de motorreductores enlazados a un piñón ubicado en los extremos del primer modulo del modulo (Ver figura 2.13).

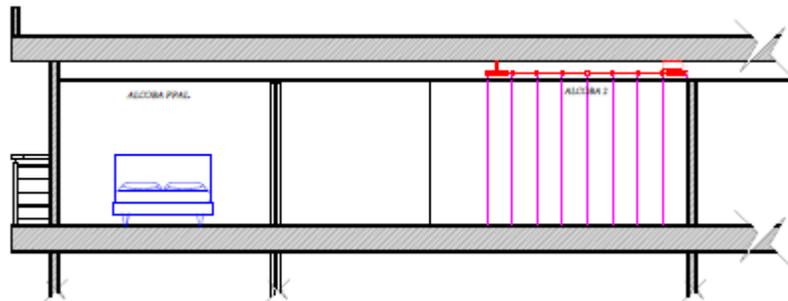


Figura 2.13: Corte del plano, funcionamiento de las paredes plegables.

2.3.3. Sala de juntas

El área adaptada con elementos flexibles incluye una mesa que tiene la capacidad de ocultarse sobre un muro de la vivienda y crecer cuando se encuentra de forma horizontal aumentando la disponibilidad de puestos adquiriendo un mayor tamaño, esto permite crear dos ambientes diferentes con una sencilla modificación, la mesa funciona por medio de una bisagra ubicada en el extremo de la mesa en contacto con la pared, al bajar la mesa y que esta tenga contacto con el suelo se activan unos motores que en contacto con unas ruedas consiguen dar movimiento a la mesa, estos motores están localizados en la parte inferior de las patas de la mesa y jalan la mesa ocasionando que se desplacen rieles interconectados entre sí, convirtiendo lo que antes era una sala de reuniones para 4 personas en un mueble para 6 o 8 personas.

2.3.4. Blackout

Con el fin de brindar comodidad y privacidad en algunas áreas del apartamento se propone usar *Blackouts* sobre las puertas ventana y entre algunas paredes internas de la vivienda; Adicional a los sistemas convencionales domóticos usados para este tipo de automatismos se propone realizar ejes guía laterales y un orificio en la base con el fin de asegurar la tela del *Blackout al suelo*.

Capítulo 3

INGENIERÍA DEL PROYECTO

3.1. METODOLOGÍA

El proyecto se desarrolla bajo un esquema con cinco etapas principales para que este pueda ser concluido satisfactoriamente, estas etapas son:

3.1.1. Análisis

Con el fin de establecer los límites del proyecto bajo las condiciones planteadas, el paso inicial del proyecto debe ser un análisis tanto de el plano entregado por el diseñador de la vivienda como de los actuadores que se pueden usar dentro de ella. En el caso de el plano se debe concluir que elementos pueden ser móviles o dinámicos (es decir que muros, muebles o electrodomésticos permitirían hacer la vivienda flexible), en cuanto a los actuadores se debe determinar un estimado de cuantos y de que tipo pueden ser usados (Electrónicos, mecánicos o neumáticos).

3.1.2. Factibilidad

La viabilidad de un proyecto se determina a partir de múltiples factores que dependen del objetivo a cumplir, unos pueden adquirir mayor importancia que otros, en este caso se opto por dar prioridad al costo de implementación y al espacio equivalente obtenido en la vivienda (el area equivalente seria la suma de los múltiples ambientes presentes en la vivienda), esto permite hacer una relación costo/beneficio y así determinar si el proyecto es factible.

3.1.3. Modelado

Para desarrollar un entorno arquitectónico apropiado como el que se requiere en el proyecto es necesario tener conocimientos en programas CAD (como los que se encuentran en la plataforma *Autodesk* [?]), esto con el fin de modelar los componentes que se mostraran en el apartamento con las gafas de realidad virtual finalmente. Para conseguir el dinamismo propuesto en el espacio habitacional es indispensable usar animaciones interactivas, estas son las que permitirán mostrar los múltiples espacios y la adaptabilidad de ellos a la necesidad del propietario.

3.1.4. Conectividad e Interacción

La integración de tecnologías es lo que hace posible este proyecto, es por esto que toma gran importancia en su desarrollo ya que va desde los diferentes programas que son necesarios para el modelo 3D, pasando por el periférico necesario para desplazarse en la plataforma virtual y hasta los diferentes accesorios con los que vienen integradas las gafas de realidad virtual.

3.1.5. Visualización y Recorrido

Como parte final solo resta disfrutar del mundo virtual interactivo y exploratorio creado (véase la sección 2.1.4), en el se deben encontrar elementos que den una impresión realista del entorno, permitiendo tener una experiencia inmersiva en el mundo de la realidad virtual.

En la figura 3.1 se muestra un esquema de la metodología anteriormente propuesta.

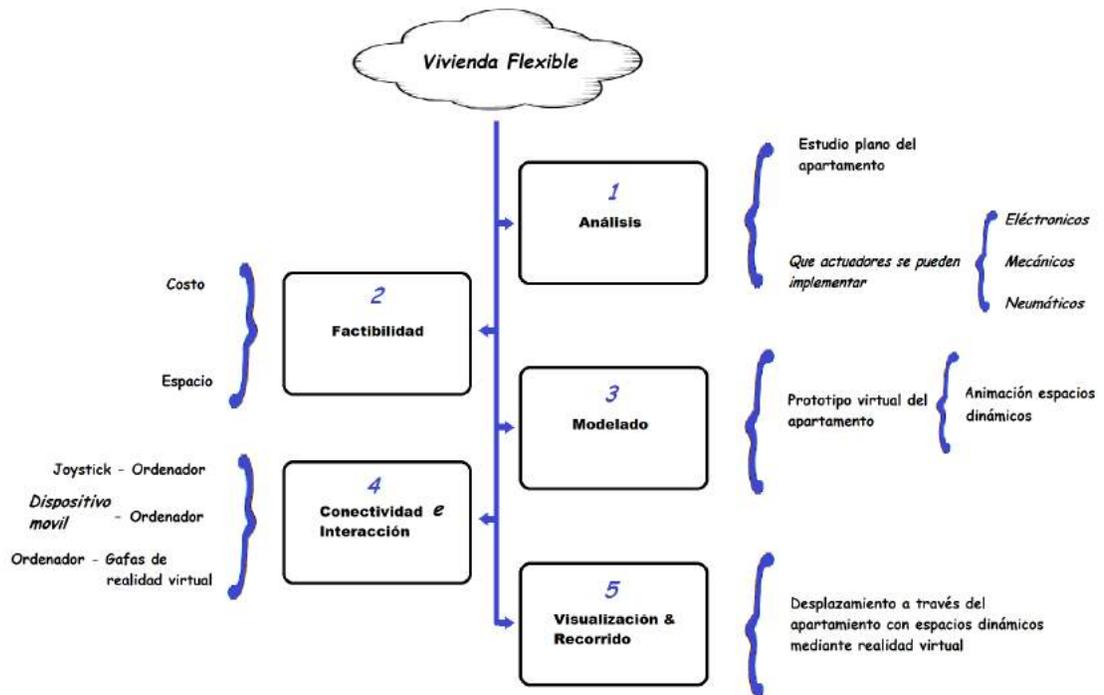


Figura 3.1: Metodología usada en el proyecto

3.2. PARÁMETROS DE ELECCIÓN DE SOFTWARE

En este apartado se explicara la razón por la cual fueron elegidos los programas computacionales que se usan en el proyecto.

3.2.1. Modelado

En la industria del diseño asistido por computadora existen muchas herramientas computacionales con la capacidad gráfica necesaria para aplicar al proyecto, la mayoría de ellas se descartan por formatos incompatibles con los motores gráficos gratuitos ó el bajo volumen de contenido instructivo en línea; Como programas de modelado de fácil acceso se contemplan tres plataformas (Programas Autodesk, Rhinoceros y Zbrush), en las cuales todas ellas se pueden hacer animaciones y modelos realistas, al final se opta por la plataforma de *autodesk* en la cual se encuentra mayor versatilidad y portabilidad

entre diferentes programas al ser la plataforma mas usada a nivel mundial (exporta en formato FBX que es soportado por los tres motores gráficos tenidos en cuenta).

3.2.2. Texturas

Para el uso de texturas realistas se contemplan dos posibles herramientas de software las cuales son: *CRAZYBUMP* y *Pixplant*, ambos son programas muy similares en cuanto a funcionamiento, características, y accesibilidad, la principal razón por la que se elige *CRAZYBUMP* es por que la herramienta *Pixplant* en su versión libre deja una marca de agua en el centro de la imagen que es usada para dar la textura esperada sobre el objeto deseado.

3.2.3. Motor gráfico

Como primer parámetro de elección se muestran los motores gráficos de libre acceso o que brindan una versión estudiantil como es el caso de *Unity*, *UDK* y *Cryengine 3D*. Entre estos tres software, Unity permite utilizar siete tipos de formatos para los modelos 3D y tres lenguajes de programación, en comparación con los motores *UDK* que permite un único formato tanto en los modelos 3D como en el lenguaje de programación y *Cryengine 3D* recibe dos tipos de modelos y la programación se da en un lenguaje básico y primitivo para las funciones gráficas que se manejan, debido a esto es elegida la plataforma *Unity* como motor gráfico.

3.2.4. Herramientas de configuración

Con el fin de tener un correcto funcionamiento de las herramientas de electrónicas a integrar para manipular el software computacional creado en el proyecto es necesario instalar un actualizador de los controladores del sistema en el ordenador, para esta tarea se tienen en cuenta los programas *Driver Toolkit* y *Driver Booster 2*, en esta ocasión se opta por el software driver Booster 2 el cual incluye adicional a su función básica una serie de plug-in que contribuyen al buen rendimiento del PC, como un limpiador de memoria.

3.3. PARÁMETROS DE ELECCIÓN DE HARDWARE

En este apartado se explicara la razón por la cual fueron elegidos los componentes electrónicos que se integran en el proyecto.

3.3.1. Gafas de realidad virtual

Actualmente existen pocos modelos de gafas de realidad virtual que cumplan con las características necesarias para un correcto desarrollo del proyecto, entre estas se encuentran las gafas Oculus Rift VR (development kit 2) y Samsung Gear VR, la principal diferencia entre estos dos modelos de realidad virtual se encuentran en el precio ya que las gafas Samsung requieren de un celular Galaxy Gear para que cumpla con la función de pantalla y sensores, por lo anterior se opta por adquirir el modelo Oculus Rift VR.

3.3.2. Controlador de movimiento

Inicialmente el movimiento se da a partir de las flechas que se ubican en el teclado y se nota la complejidad de manejo, por lo que se busca una alternativa a este problema, encontrando dos alternativas, primero se contempla el uso de un control alámbrico de Xbox 360 el cual cuenta con mas botones de los necesarios por lo que se puede afirmar que esta sobredimensionado para la tarea a cumplir, la segunda opción es un *joystick* (*Genius*) el cual cuenta con una palanca tipo stick y un disparador en la parte posterior, lo que es mas que suficiente para suplir los requerimiento en el proyecto.

3.3.3. Comunicación inalámbrica

Para poder realizar una comunicación tipo inalámbrica se plantea utilizar un modulo bluetooth, contemplando tres tipos diferentes, estos son el HC-05, HC-06 y KC-21 de los cuales por experiencia durante el desarrollo de la carrera se conoce que el modulo HC-05 presenta problemas de estabilidad luego de entablar la comunicación entre dispositivos, el Bluetooth KC-06 tiene características sobredimensionadas para la función requerida en el proyecto, el modulo HC-06 es el Bluetooth elegido para ser integrado al proyecto ya que cumple con la velocidad de transmisión requerida por la tableta *android*, tiene facilidad de configuración y es estable en la conexión.

3.4. ARQUITECTURA PROPUESTA

La estructura de operación del proyecto funciona a partir de herramientas tanto electrónicas como computacionales, dividiendo así su funcionamiento en dos secciones diferentes, la primera de ellas corresponde a la conexión e interactividad entre los software utilizados en los aspectos que corresponden a conexión y configuración de dispositivos, modelamiento de el espacio virtual, integración y compilacion; la segunda de ellas muestra el recorrido de las señales que se utilizan los componentes electrónicos al momento de utilizar la herramienta de realidad virtual, esta arquitectura se muestra en la figura 3.2 donde se puede ver cada componente que hace posible la inmersión en la realidad virtual, sea este un elemento físico o no.

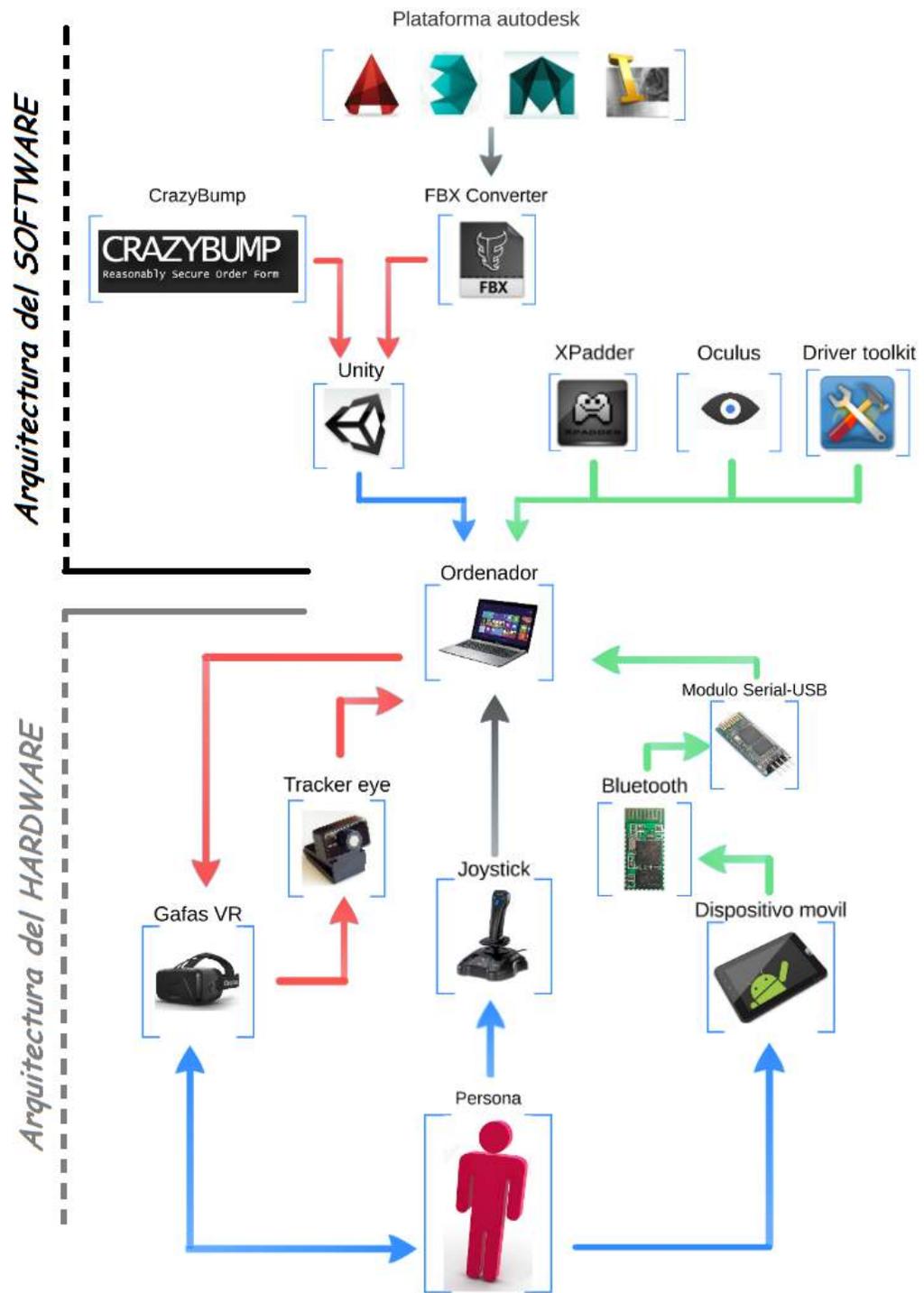


Figura 3.2: Arquitectura del proyecto

A continuación se explicara cada etapa de la arquitectura propuesta de forma detallada, partiendo de forma individual en cada sección explicando la función de sus componentes electrónicos o comunicacionales usados en el proyecto y principalmente el sentido de la señal que por estos pasa.

3.4.1. Arquitectura de software

En esta sección de la arquitectura se muestran los diferentes programas usados para desarrollar la herramienta que permite explorar e interactuar con un mundo virtual, los conectores que los unen entre si señalan el orden secuencial para usar los programas; Los diferentes software usados se pueden agrupar tres etapas de desarrollo, estas son:

Etapa de modelado del espacio virtual

En esta etapa del proceso es donde se crean y adecuan los componentes que se encontraran finalmente en el apartamento virtual, por lo que en ella se encuentra una gran variedad de programas CAD la mayoría habilitados en la plataforma de Autodesk, ya que de estos programas no siempre sale el archivo en un formato universal o compatible es necesario la herramienta *FBX converter* desarrollada también por Autodesk, esta permite convertir todos los modelos creados en archivos *.FBX* sin importar si este fue creado en *Inventor*, *Maya* o *AutoCAD* (los programas usados para crear las figura deben pertenecer a la plataforma de Autodesk). De los programas usados para modelar los componentes e imprimir en ellos texturas y lograr realismo, el único que no pertenece a la gran familia de Autodesk es *CrazyBump* en este programa lo que se busca es conseguir relieves sobre los materiales y elementos que se encuentran en un apartamento normalmente recreando la realidad, este se usa en elementos como el piso, un mesón de cocina o los labrados de la madera.

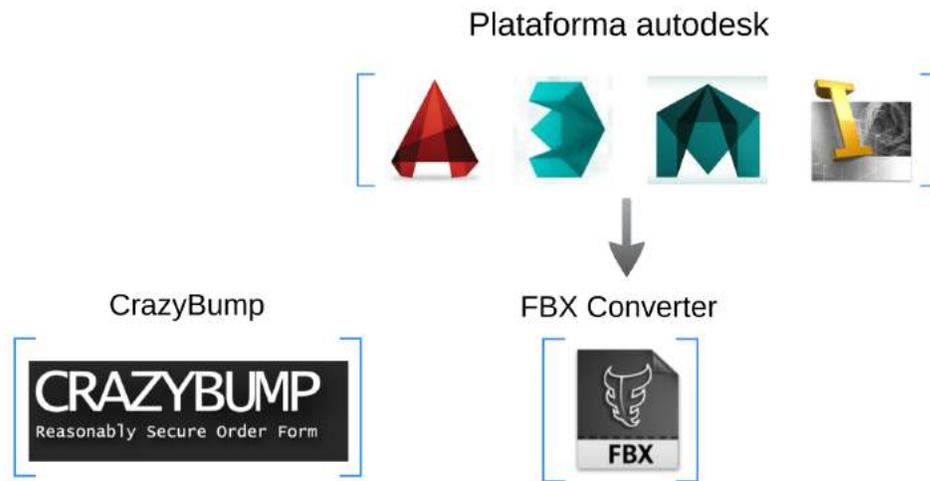


Figura 3.3: Arquitectura de software, etapa de modelamiento del espacio virtual

Etapa de conexión y configuración de dispositivos

Esta etapa hace referencia a los software que se utilizan con el fin de instalar el controlador de los dispositivos de hardware que carecen de uno o de aquellos que fácilmente se desactualizan como el *Joystick*, el módulo *serial-USB* o las gafas Oculus rift y todos sus accesorios; Para el primer caso se puede usar el programa *Xpadder*, este cuenta con un gran catálogo de controles por lo que es ampliamente usado, *Xpadder* reemplaza las funciones de el teclado y mouse por los pulsadores y sticks que se encuentran en los mandos de vídeo juegos, por eso mismo es necesario realizar la configuración y asignación de las teclas necesarias, en el mismo programa; Actualmente existen muchos software encargados de mantener los programas de un computador actualizados, sin embargo para el caso específico de este proyecto se utiliza *Driver Toolkit* ya que se probó con otras herramientas similares pero no se dio el resultado esperado. este programa mantiene actualizado el controlador del dispositivo *serial-USB* y es necesario dado que en ocasiones presenta problemas de la compatibilidad, de igual forma se debe comprobar que el módulo se encuentre asignado al puerto *COM* deseado y de no ser así proceder a configurarlo; En el caso de las gafas de RV no se da como tal problemas de afinidad entre el software y el hardware, este artefacto solo requiere de una correcta configuración en el programa *Oculus configuration utility* en el cual se modifica el encendido de las gafas, el tipo de visión y permite correr un demo de demostración, además de esto se

debe habilitar las gafas como única pantalla (este ultimo solo es necesario si la pantalla principal del computador no tiene las dimensiones apropiadas)

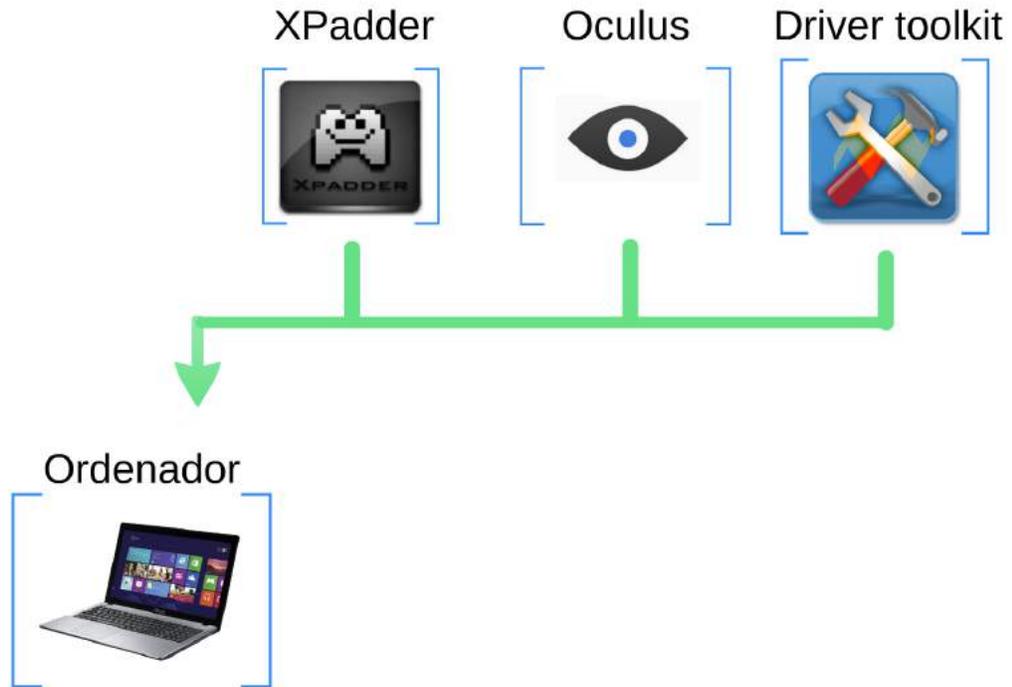


Figura 3.4: Arquitectura de software, etapa de conexión y configuración de dispositivos

Etapa de integración y compilación

Al final de la arquitectura del software encontramos la etapa de fusión entre los diferentes programas usados, para, finalmente compilar todo en un único archivo de extensión *.exe* listo para ser usado por los dispositivos de hardware, esto se logra gracias a la herramienta *unity* en la cual se realiza este último proceso, *unity* es un motor de video juegos que recientemente se ha hecho muy conocido por sus diferentes proyectos *open-source*, la fácil forma de adquirirlo, usarlo y los múltiples formatos con los que es compatible.

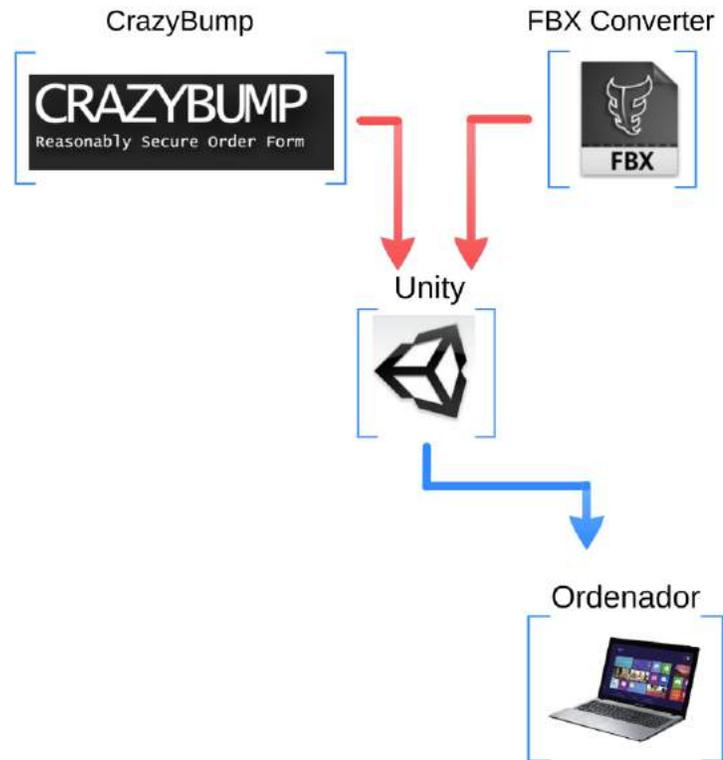


Figura 3.5: Arquitectura de software, etapa de integración y compilación

3.4.2. Arquitectura de hardware

Esta sección se construye con las diferentes conexiones existentes entre dispositivos electrónicos utilizados en el momento de la visualización en la realidad virtual y se dividen en tres ejes de señales diferentes estas son:

Eje de visualización en gafas VR

Esta rama de la arquitectura de hardware es la que permite ver el entorno virtual, se compone de dos elementos entre máquina y usuario estos requieren de dos conexiones USB y una HDMI (esta ultima se puede remplazar por una conexión DVI con un adaptador que incluye el kit de desarrollo de *Oculus VR*, pero se recomienda usar directamente el cable HDMI para una mejor experiencia). Las gafas *Oculus rift* son el

punto de inflexión en la historia de la realidad virtual gracias al sistema de seguimiento de posición en tiempo real, ya que esto, es principalmente lo que diferencia a este visor de los conocidos anteriormente. La fuente de energía del sensor de movimiento viene del ordenador, pero los datos fluyen de forma bidireccional entre el sensor y las gafas VR, que posteriormente serán analizados en una tarjeta electrónica denominada *latency testing system* integrada en las gafas. La señal de vídeo proviene del computador donde ya se ha hecho la correspondiente etapa de moldeamiento, esta señal tiene una calidad de 920*1080 HD en cada ojo y cubre un campo de visión de 100°, además de esto las gafas también incluyen giroscopio, acelerómetro y magnetómetro, con el fin de determinar posiciones que el sistema de seguimiento no puede [?].

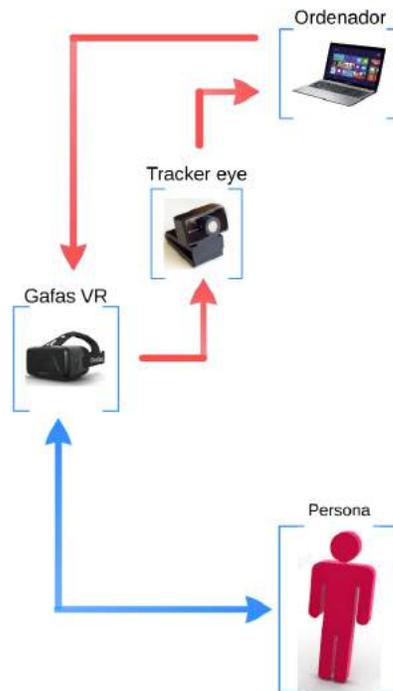


Figura 3.6: Arquitectura de hardware, eje visualización gafas VR.

Eje de movimiento con joystick

El flujo de la señal para la exploración y parte de la interacción del espacio virtual por obvias razones inicia desde el usuario hacia el ordenador ya que es quien esta viendo el

apartamento quien debe indicar hacia donde se mueve este virtualmente, el movimiento se da por medio de un mando de vídeo juegos provisto con un *stick* digital de cuatro direcciones, pulsadores y un disparador. El usuario indica la dirección del movimiento por medio de la palanca (*stick*) enviando un pulso digital estándar de una comunicación USB, estas señales son equivalentes a las flechas en el teclado de un computador; El mando a la vez también tiene otra función, esta es enviar la señal que activara las animaciones del mundo virtual brindado la interactividad necesaria para mostrar los espacios dinámicos y se dará cuando el usuario al enfocar el objeto moviendo las gafas VR accione el disparador ubicado en la parte posterior de la palanca en el mando, esta acción sobre el disparador es equivalente a dar clic sobre el botón izquierdo del mouse en el computador. En la figura 3.6 se pueden ver los componentes que se necesitan para esta comunicación y el sentido de el flujo de estas señales.



Figura 3.7: Arquitectura de hardware, eje movimiento joystick.

Eje comunicación con dispositivo móvil

Parte de la integración de tecnología hecha en el proyecto incluye una interfaz gráfica en una tableta *Android* la cual permite encender y apagar las luces de cada espacio en la casa por medio de comunicación *Bluetooth*, para que esto sea posible es necesario usar un modulo de comunicación *serial-USB* el cual se encarga de convertir la señales recibidas en el pin *Rx*, provenientes del modulo *Bluetooth* de la tableta que a su vez es manejada por el usuario. El sentido de la señal en este eje tiene siempre un solo sentido, esta dirección viene desde el usuario hacia el computador, pasando por cada componente ya antes mencionado.

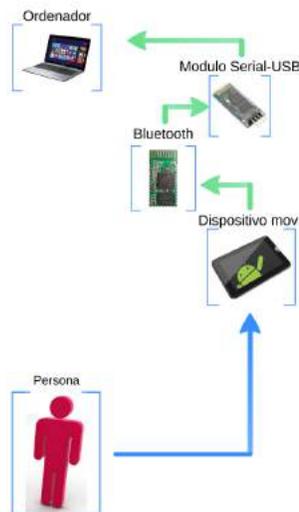


Figura 3.8: Arquitectura de hardware, eje comunicación con dispositivo móvil.

3.5. DESARROLLO DEL SOFTWARE

En la sección 3.4.1 se explico como es posible concebir un mundo virtual haciendo énfasis en los programas que se recomienda usar para lograr un resultado satisfactorio del proyecto, pero no se dio mucha profundidad en cuanto a que se debe hacer en cada uno, a continuación se desarrolla de forma ordenada el proceso seguido en cada

programa usado para conseguir el modelo arquitectónico presentado en las gafas de realidad virtual. Las herramientas computacionales usadas son:

- Autocad**- *Plataforma arquitectónica desarrollada para elaborar planos de plantas, fachadas y cortes estructurales, usado ampliamente en la industria constructiva por años.*
- Inventor**- *Software para ingenieros utilizado para generar planos bajo normas internacionales de piezas, herramientas y ensamblados de máquina, adicional a esto hace análisis de esfuerzos sobre componentes individuales o compuestos. brinda facilidades de manejo sobre su interfaz gráfica y herramientas que en ella se encuentran.*
- 3DMAX**- *Herramienta de modelado gráfico semi realista utilizada en la industria del entretenimiento para animación multimedia, diseño y modelado, con una amplia gama de herramientas especializadas para este fin.*
- FBXconverter**- *Programa realizado para facilitar la compatibilidad entre los programas de la plataforma autodesk y permitir la integración de estas brindando la posibilidad de usar las cualidades individuales de cada programa en un solo modelo.*
- CrazyBump**- *Software usado para crear texturas realistas en modelos graficos a partir de una única imagen.*
- Unity**- *Motor gráfico 3D habilitado para la creación de juegos de vídeo, aplicaciones interactivas y animaciones en 3D, con múltiples formatos de salida. La programación para crear proyectos se basa en la secuencia de scripts que usan el lenguaje C#.*

3.5.1. Diseño del mundo virtual

Inicialmente se deben reconocer las necesidades que implican los objetivos a cumplir y proyectar con esto los programas que se requieren, por lo anterior se presenta como solución utilizar los software desarrollados por la plataforma de *autodesk*, ya que cubren la mayoría de exigencias que se presentan al momento de hacer modelos 3D.

Diseño CAD

El apartamento virtual se inicia a partir del plano entregado por el arquitecto diseñador del espacio, este debe contar con las especificaciones apropiadas para poder crear una figura tridimensional a partir del diseño, esto permite delimitar el área virtual de trabajo. Para llevar a cabo la acción mencionada, es apropiado usar el programa *Autocad* ya que se facilita comparar medidas para mantener de forma rigurosa el diseño hecho por el arquitecto, al abrir el programa se debe crear una hoja de trabajo nueva, establecer la escala, crear una capa nueva y en ella pegar el plano del apartamento que fue entregado, luego crear una nueva capa y asignarle un color de capa que se diferencie de los usados en el plano, posteriormente, elegir la herramienta *rectángulo* ubicada comúnmente en la parte superior izquierda (puede cambiar debido a la versión utilizada o a la personalización de las herramientas sobre el *viewport*), luego hacer un calco de cada pared en el plano, empezar por las paredes exteriores e ir continuando hacia el centro, luego se desactiva la capa en la que se encuentra la imagen del plano y finalmente buscar en la sección de modelado 3D una herramienta llamada *extruir*, ahora seleccionar todos los rectángulos que equivalen a las paredes de la vivienda se presiona la tecla *ENTER* y se establece la altura que estas tendrán (terminado esto ver desde una perspectiva isométrica). En la figura 3.9 se puede ver el resultado esperado con el procedimiento antes descrito. Luego de comprobar que las paredes se encuentran tal como se muestran el plano guardar el archivo con un nombre fácil de reconocer.

En *Autocad* solo resta hacer una figura 3D que se convertirá en el techo de la vivienda, para esto ya se debe tener una concepción previa de todos los agujeros que requieren las bombillas y los movimientos de actuadores que son necesarios en el apartamento, Para esto se pegara nuevamente el plano del diseño en una capa que únicamente lo contenga a el, luego se realiza la figura principal del techo en nueva capa, con la herramienta *polilínea* se recorrerá la parte exterior del plano esta figura sera el cuerpo del techo, nuevamente se crea una capa aparte, en ella solo se encontraran las figuras que describen los agujeros que permanecerán en el techo, luego mediante la herramienta extrusión se dará volumen tanto a el cuerpo de el techo como a los agujeros (es necesario que los agujeros tengan mayor altura que el techo, por lo que se recomienda hacer este paso en dos etapas diferentes), en el ultimo paso para crear el techo se debe usar la herramienta sustracción ubicada en los modificadores de figuras 3D, luego de haberla elegido se deben seleccionar todas las figuras que se sustraerán (agujeros necesarios) a el cuerpo del techo, se presiona la tecla *ENTER* , luego se selecciona la figura a la cual

se le restaran estas figuras (cuerpo del techo) y se presiona la tecla *ENTER* nuevamente (terminado esto ver desde una perspectiva isométrica). En la figura 3.9 se puede ver el resultado esperado con el procedimiento antes descrito. Luego de comprobar que las dimensiones del techo y los agujeros al igual que se posición son las deseadas guardar el archivo con un nombre característico.

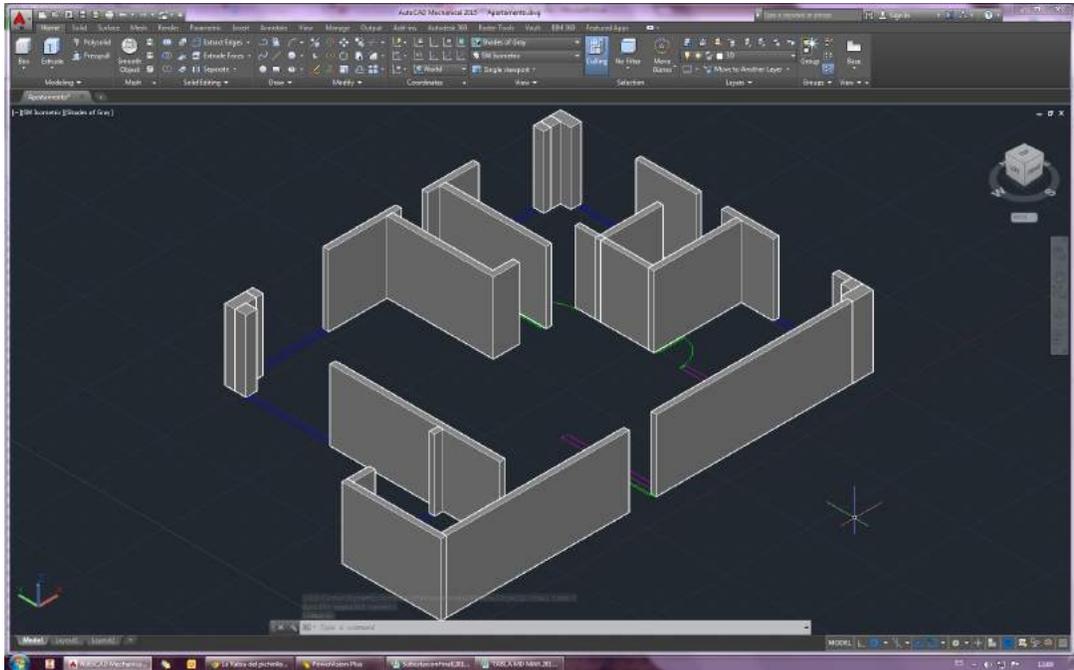


Figura 3.9: Vista isométrica de las paredes del apartamento en *Autocad*

Para poder tener los terminados de un apartamento convencional es necesario agregar puertas y ventanas al apartamento, esto se puede realizar fácilmente en la herramienta de dibujo y diseño *3D max design* ya que esta incluye plantillas predeterminadas de tres tipos de puertas, a continuación se explicara como hacer una puerta de pivote en *3D max desing*, al iniciar el programa se debe tener en cuenta que el computador que se esta usando debe tener buen rendimiento y de ser posible una tarjeta gráfica independiente ya que con las ultimas versiones del software suele colapsar después de un tiempo de no ser así. Para iniciar a hacer la puerta primero se debe ubicar la herramienta que por defecto hace la mayoría del trabajo esta se encuentra en la pestaña *create* luego se debe dirigir al recuadro desplegable y seleccionar la opción *doors*, allí elegir el tipo de puerta *pivot* (estas son las que giran al rededor de un único eje y abren

en una sola dirección) posterior a esto se dan las mediadas deseadas para la puerta estableciendo ancho, alto y profundo luego se presentan opciones adicionales para la puerta como el numero de paneles decorativos sobre ella. El único elemento que se debe agregar a la puerta para conseguir que luzca normal es una chapa, esta se hace mediante la integración de un cono, una esfera y un cilindro, cada una de estas figuras básicas se encuentran eligiendo la opción *standard primitives* que se haya en el recuadro desplegable de la pestaña *create*, después de haber llevado estas figuras al *viewport* ordenarlas de la siguiente forma, primero el cono, luego la esfera y por ultimo el cilindro, finalmente poner cada figura obtenida con esta unión sobre la superficie de la puerta. En la figura 3.10 se puede ver el modelo obtenido con el procedimiento antes descrito.

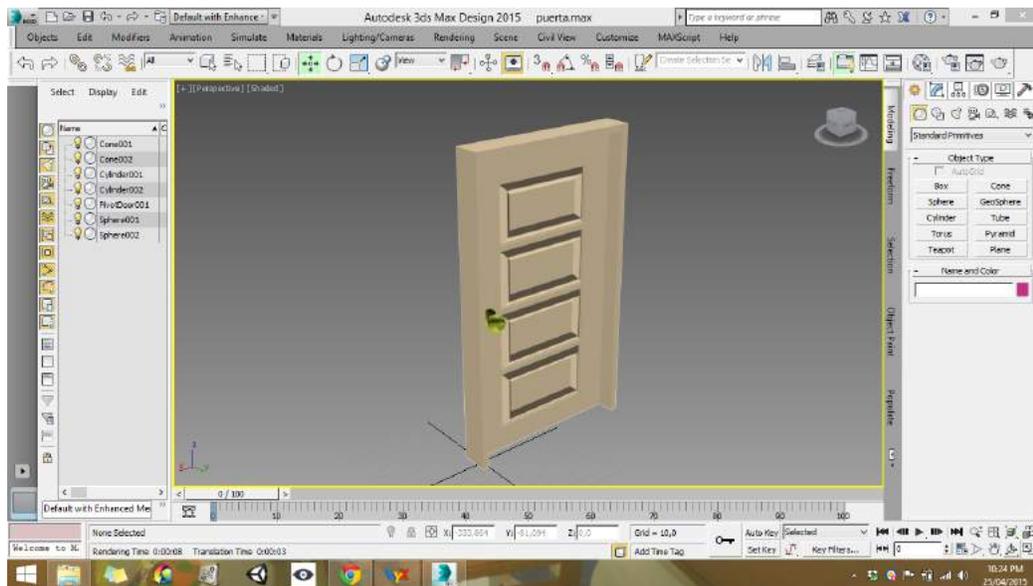
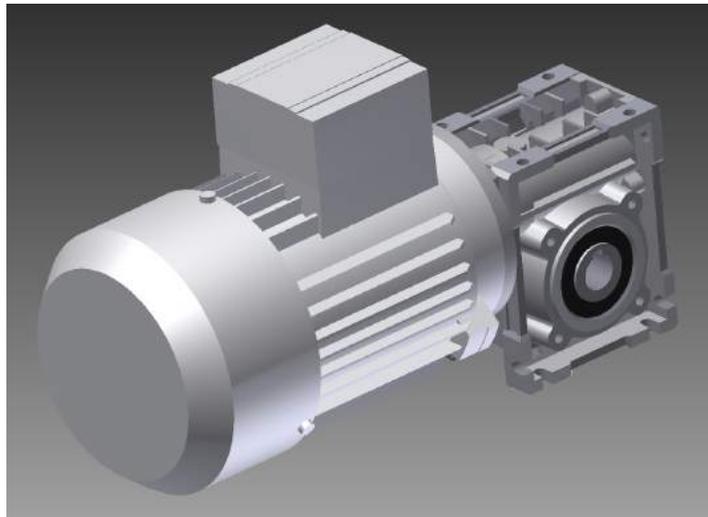


Figura 3.10: Modelo realizado 3D max para culminar la base del apartamento

Con el fin de dar credibilidad a la implementación de movimiento propuesta en el apartamento se realizaron los modelos 3D de los actuadores, para ello se recomienda utilizar el software *Inventor* por la facilidad que brinda al trabajar con elementos mecánicos, por la cantidad de pasos que un modelo mecánico requiere no se explicara como hacer este tipo de elementos, simplemente se darán los consejos que se consideran necesarios para concluir cualquier pieza mecánica a realizar satisfactoriamente. Inicialmente y por obvias razones se debe llevar a cabo el reconocimiento del programa al igual que

el de la pieza a realizar, encontrando en cada uno los componentes que se prestarían mas complejos al realizar, luego, de ser posible conseguir un plano explosionado de el elemento (en dado caso que este tenga mas de una pieza) esto es para facilitar el ensamble de cada componente al finalizarlos, posteriormente hacer las acotaciones necesarias de cada pieza esto para que luego no se estén presentando problemas de dimensionamiento y realizar cada pieza en archivos independientes manejando un sistema de unidades de fácil manejo (sean milímetros, centímetros, pies o pulgadas), finalmente crear un archivo de ensamble y llevar a el todas las piezas hechas, unir las en base al plano explosionado o de no tener uno se puede realizar en base a los conocimientos mecánicos que se tenga de el modelo. En la figura 3.11 se puede ver el resultado final luego se seguir los consejos para modelar en *inventor*.



(a) Motorreductor electrico



(b) Cilindro neumático de doble efecto

Figura 3.11: Modelos hechos en *Inventor* de los actuadores propuestos

Realismo

Para poder brindar una autentica experiencia creíble se deben realizar múltiples figuras con gran detalle visual, estas son las cosas que permiten tener la impresión de que alguien realmente ha vivido allí y son las que se deben encontrar normalmente en una casa u oficina, como vasos, lapices, cepillos, libros, cuadros, etcétera. Para llevar a cabo esta tarea el mejor software es *3D max design* ya que por su cantidad de

herramientas permite desarrollar componentes complejos en poco tiempo y con gran definición. A continuación se explicara como conseguir el modelo de una dona glaseada con chocolate, recubierta con chips de chocolate; Al encontrarse sobre la ventana principal de *3D max design* se debe arrastrar desde la pestaña *create* en la sección del recuadro desplegable *standard primitives* un toroide y darle a este las dimensiones del cuerpo de la dona, luego crear un segundo toroide con medidas similares, para darle la curvatura característica a la capa glaseada se debe dar clic en el botón derecho del mouse bajar al final del panel de opciones y seleccionar la opción *editable poly*, posteriormente seleccionar la opción *polygon* y pulsar sobre una de las caras de el toroide, luego manteniendo oprimida la tecla *shift* seleccionar la cara continua a la seleccionada anteriormente, ir a la herramienta *extrude* y disminuir el grosor de el segundo toroide, cambiar a las herramientas que permiten modificar vértices y jalar hasta curvar la superficie, posteriormente utilizar la herramienta mover para ubicar la capa del glaseado sobre el toroide numero uno, finalmente buscar en el cuadro desplegable de la pestaña *create* la opción *extended primitives* y elegir la herramienta *chamber box* crear tantas chispas de dulce como se quiera y ubicarlas sobre la dona ya glaseada.

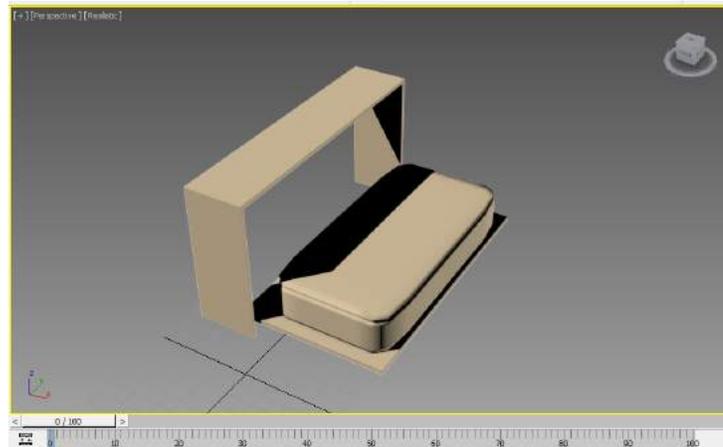


Figura 3.12: Dona glaseada con chispas de dulce

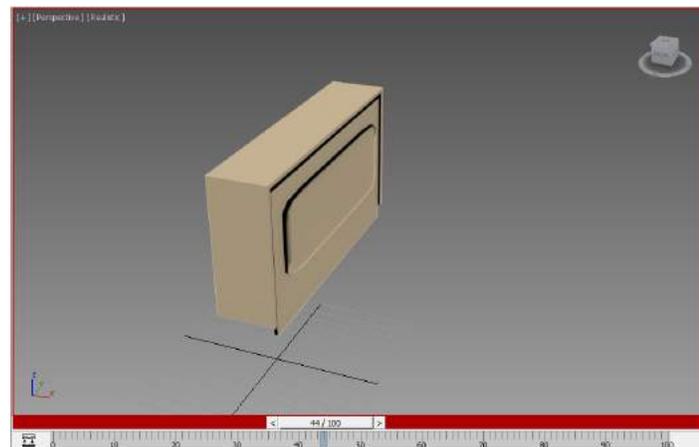
Animación

Las animaciones desarrolladas son uno de los pilares del proyecto ya que estas son las que finalmente mostraran los diferentes ambientes en la vivienda, estas se deben ver naturales, bien implementadas y fluidas. Para su creación se puede usar el programa *3D max desing* el cual permite cumplir las expectativas ya mencionadas. Inicialmente Existen dos tipos de animaciones las que se desplazan únicamente bajo movimientos lineales y aquellas que tienen movimientos rotacionales, a partir de estas dos formas básicas de animación se derivan otras maneras compuestas como la animación por trayectoria.

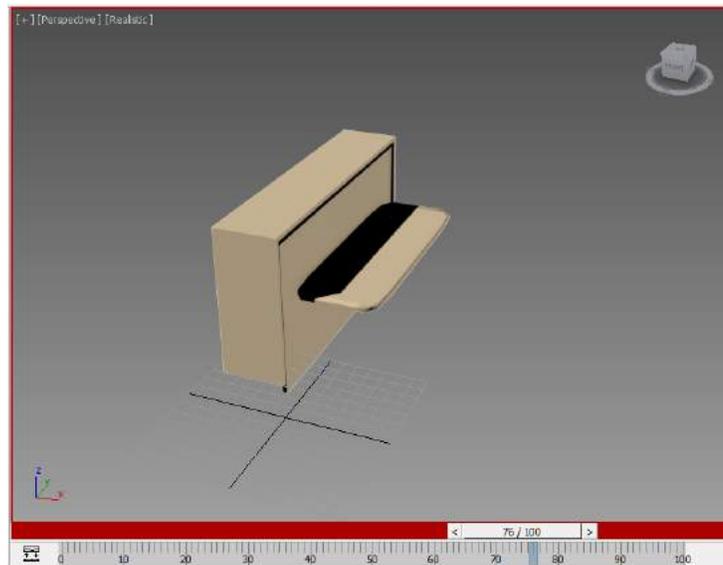
Las animaciones hechas para el proyecto se desarrollaron en dos plataformas diferentes, debido a la complejidad de algunas de ellas y con la intención de que estas se vieran de forma natural se utilizaron las herramientas *3D max design* y *Unity*, a continuación se explicara el procedimiento a seguir para elaborar la animación de una *wall bed* sencilla equipada con un escritorio que se puede desplegar al cerrar la cama, para poder iniciar la animación ya se debe tener abierto en el programa el objeto a animar (ver figura 3.13a), en el, agrupar las figuras que se moverán al mismo tiempo, esto con el fin de evitar errores o pasos innecesarios sobre los ejes coordenados o pivotes de giro auto generados por el programa. Posterior a esto se puede iniciar lo que es como tal el proceso de animar, dirigirse a la parte inferior derecha del programa hasta el recuadro *auto key* al seleccionarlo, sobre el recuadro del *viewport* debe aparecer un marco de color rojo, esto ocasiona que los objetos que se encuentren en diferentes posiciones y en *frames* distintos se unirán mediante movimiento continuo creando la animación, el tiempo que esta dure depende únicamente de la distancia entre los frames. En la figura 3.13 se puede apreciar cada estado de la animación.



(a) Modo cama abierta



(b) Modo cama sobre muro



(c) Modo escritorio

Figura 3.13: Etapas animación de Wall bed multifunción en *3D max design*.

3.5.2. Unity

Para iniciar un proyecto en Unity se debe tener en cuenta que su flexibilidad en cuanto a formas, texturas, animaciones y otros componentes es muy limitada, es una excelente herramienta para integrar dichas características desde otros programas del paquete de Autodesk (programas ya tratados en este capítulo), a unity cualquier elemento debe ingresar en formato FBX.

Nuevo proyecto

Para generar un nuevo proyecto se va a la barra de herramientas *File >> New project* aparece una nueva pantalla con un *Check list* donde se eligen los paquetes de propiedades y facilidades que tendrá tal proyecto y se define la ubicación de la carpeta madre dentro de la memoria del ordenador. en la figura 3.14 se muestra de forma gráfica cada paso a seguir para crear un nuevo proyecto

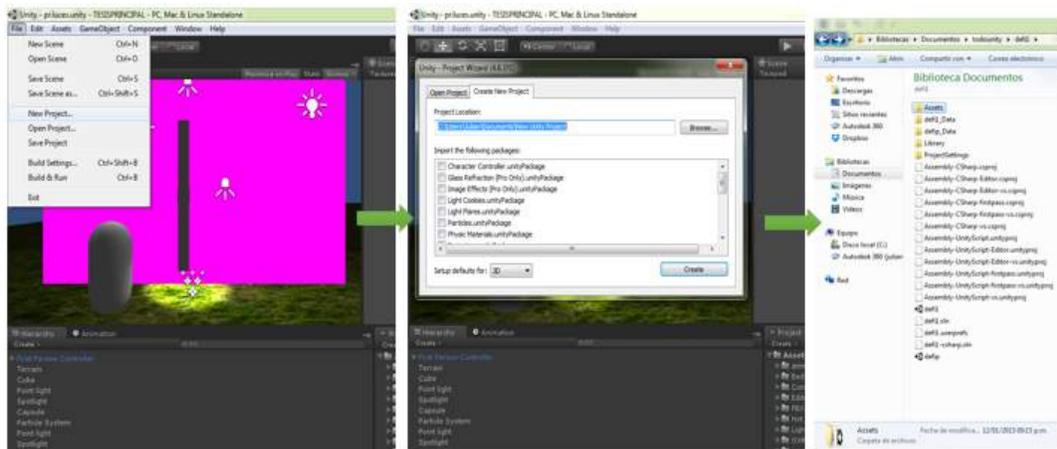


Figura 3.14: Crear Nuevo proyecto en Unity

Unity requiere múltiples pantallas de trabajo para definir las características y comportamientos de cada elemento ya sea dentro del proyecto o dentro de cada escena como se muestra en la pantalla superior izquierda *game* (ver figura 3.15) se muestra una vista preliminar de cómo va quedando la escena, en síntesis es un ejecutable del

juego, en la siguiente pantalla *scene* en la parte superior izquierda es donde se manipulan los elementos que cada escena contiene, se puede modificar el tamaño, la posición, la rotación en el espacio y las propiedades del elemento que se elige, la siguiente barra de herramientas *inspector* determina las características que cada elemento tiene y las propiedades que determinan su comportamiento dentro del juego, debido a que un proyecto puede tener varias escenas, en la parte de abajo en la pantalla se ven las dos partes dedicadas estrictamente al contenido de la escena actual *hierachy* y el contenido del proyecto en general *project*, como pestañas adicionales para el comportamiento de animaciones hechas desde Unity se tienen las ventanas *Animation* que traza la interpolación de los movimientos y *Animator* que determina en que momentos y bajo qué condiciones se reproduce una animación. en la figura 3.15 se puede ver la interfaz gráfica de unity luego de haber creado un nuevo proyecto.

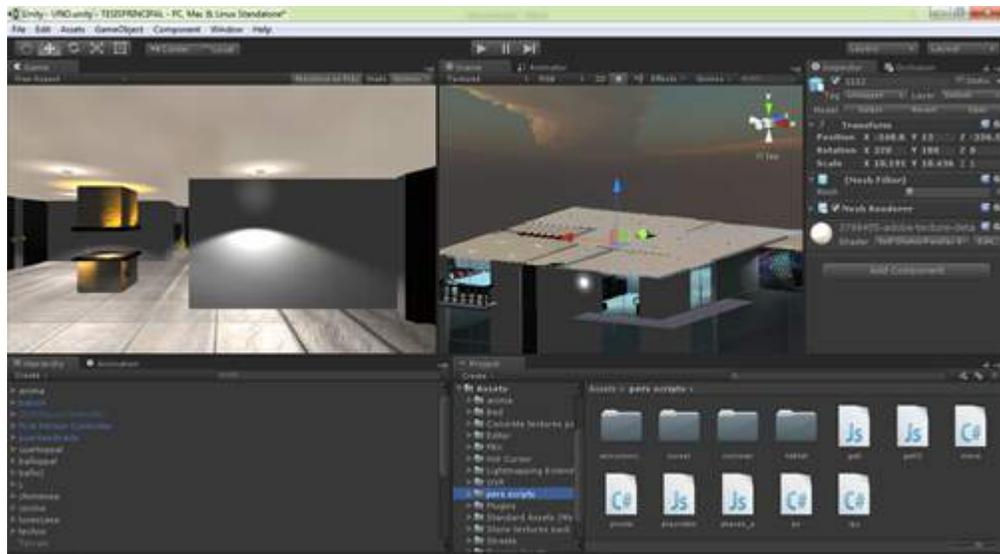


Figura 3.15: Interfaz grafica de Unity.

Exportar e importar objetos 3D

Principalmente para este proyecto se requiere levantar los muros a partir de un plano inicial, imagen1 generalmente se utiliza un modelo 3D importado de alguno de los programas de modelado para arquitectura ya sea *AutoCAD*, *3Dmax*, *Revit* entre otros, si el modelo está en algún otro formato como *3ds*, *Dae*, *Dxf*, *Obj*, se utiliza el

programa *FBX Converter* el cual modifica tales formatos hacia *.fbx* , en la Figura 3.16 se muestra el ejemplo para exportar modelos puntualmente desde *AutoCAD* en *.fbx*, ya hecho ese procedimiento el modelo en formato *.fbx* se debe trasladar hacia la carpeta madre en la ubicación que se definió al momento de crear proyecto, esto para que Unity ya lo tenga en cuenta dentro de los elementos propios del proyecto actual.

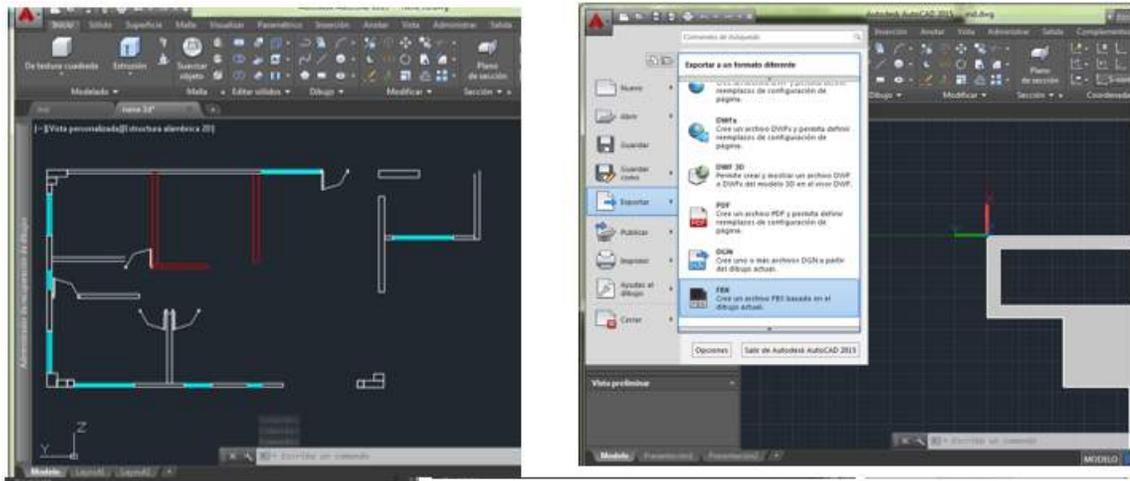
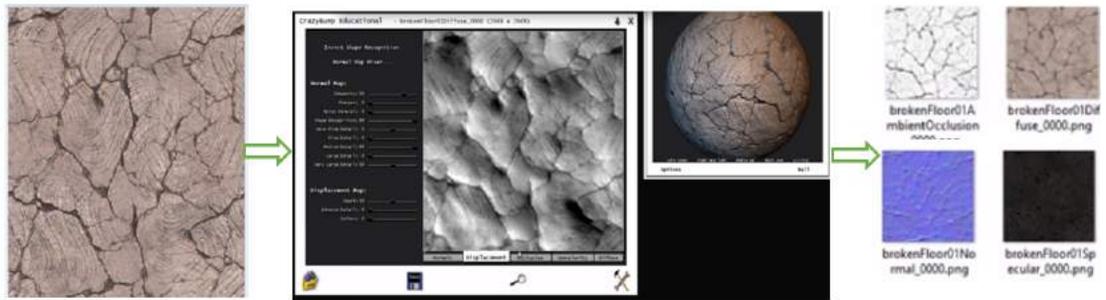


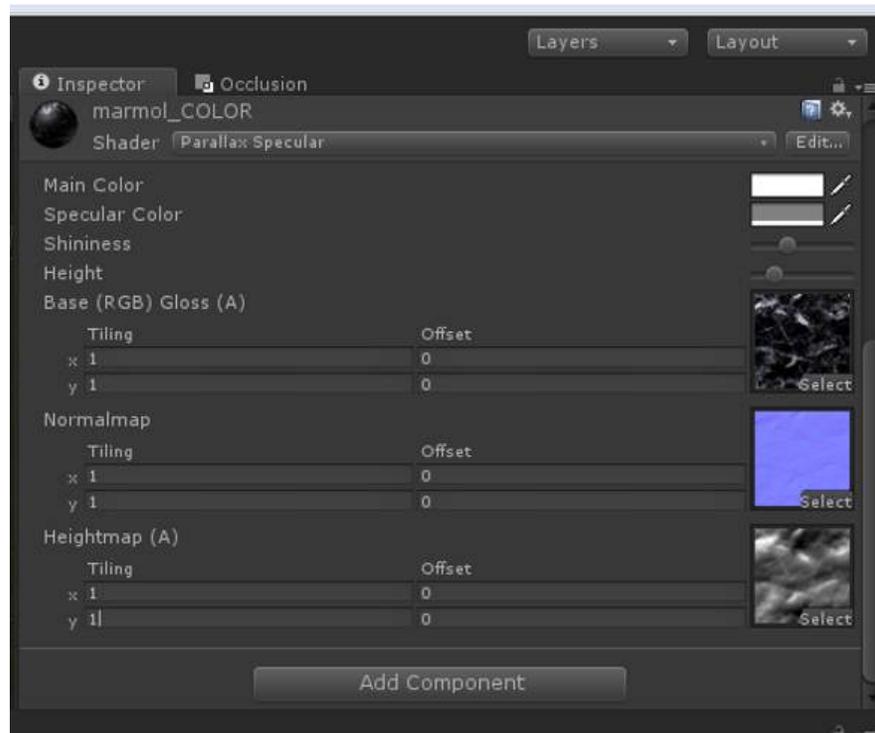
Figura 3.16: Exportar archivos en formato *FBX*.

Texturizado

Ya teniendo la figura 3D inicial del inmobiliario el siguiente paso es darle textura a los elementos dentro del modelo, para realizar esta tarea se debe tener a disposición programas como *CrazyBump* que generan tres espectros diferentes a partir de la misma imagen para darle realce y realismo a cada textura (ver figura 3.17a), esos espectros se generan para dar la propiedad *Specularity* (cantidad de reflectividad de una superficie en 3D) que determina brillo y tamaño de los reflejos sobre determinada superficie. Luego de tener los 3 espectros diferentes se dirige a la barra de herramientas *ASSETS* >> *CREATE* >> *MATERIAL* luego se dirige a la ventana *inspector* anteriormente mencionada y se configuran de la siguiente manera. En la figura 3.17b el *tiling* es la cantidad de repeticiones que tiene la imagen en el eje X y Y respectivamente en la superficie a la que se le aplica la textura, y finalmente la textura se arrastra hasta el elemento que se desee en la pestaña *SCENE*, cabe resaltar que existen paginas donde se pueden descargar las imágenes de texturas básicas listas para hacer este proceso.



(a) Espectros de la imagen en CrazyBump.



(b) Ventana inspector de Crazy Bump.

Figura 3.17: Programa CrazyBump para crear texturas.

Luego de tener el interior del apartamento totalmente texturizado se empieza a poner el mobiliario, este procedimiento se puede con modelos propios según la habilidad que se tenga para modelado 3D en los programa CAD o por recursos encontrados en Internet, actualmente existen páginas dedicadas especialmente a recrear modelos realistas 3D casi de cualquier cosa conocida en el mundo, se tienen modelos industriales, arquitectónicos, automotrices, animales, humanos, etc. , los modelos más complejos tienen costo y los más básicos son versiones free. La manera de importar todos estos modelos a Unity es

de la misma forma que se explicó anteriormente en la figura 3.16 con *AutoCAD*. Este es un ejemplo de las importaciones a escena de modelos obtenidos desde Internet, la cama, los nocheros, el mueble del TV y todo el mobiliario del baño. en la figura 3.18 se muestra una vista superior de la habitación principal donde se encuentran diferentes modelos 3D.



Figura 3.18: Habitación principal con modelos 3D en Unity.

Animaciones con Unity

La flexibilidad para animar directamente desde Unity es muy limitada, para esto se requiere manejo en programas como *Maya* o *3D max design*, este programa se especializa en animaciones fluidas y fácilmente exportables en formato *.fbx*, aunque si se quiere hacer animaciones básicas como rotaciones, traslaciones o ampliaciones desde Unity se hace teniendo en cuenta una línea de tiempo creando fotogramas clave los cuales definirán el momento en que un elemento cambia de estado, entre estos puntos existe una interpolación de movimiento la cual suaviza cambio en cada instante de tiempo. Aquí se muestra cada uno de los elementos que interfieren a la hora de crear una animación, en la figura 3.19 se puede observar en la pestaña *Animator* e *Inspector* las animaciones están reducidas a posición, rotación y escala en los 3 ejes de cada elemento, la pestaña *Animator* el hilo conductor de la forma en que funciona internamente según condiciones y variables descritas por el usuario, en la parte baja de la pantalla se tiene cada uno de los objetos que se encuentran consecutivos al elemento que se eligió, para

este caso puntual se elige la mesa y sus elementos consecutivos serán las bases de la mesa, el vidrio y el marco así es que cada uno puede tener su propia curva de animación con respecto al tiempo.

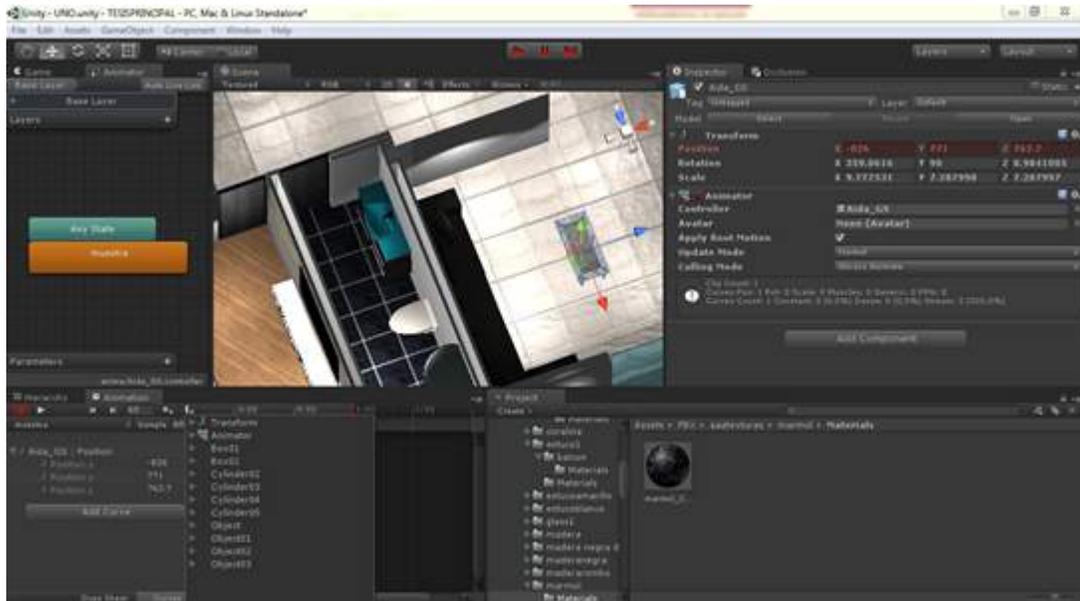


Figura 3.19: Animaciones en Unity.

Luces y sistemas de partículas

Para integrar las luces dentro del escenario se debe tener en cuenta varios factores como el uso que se le da a cada tipo de luz que Unity tiene predeterminado, es decir, El programa tiene 3 tipos de luces principales las cuales son *spot*, *point*, *directional* light, este último recrea la luz del sol en un escenario y de los tres tipos de luces es el único que genera sombras sobre los sólidos. En la figura 3.20 se encuentra un ejemplo con los tipos de luz que se pueden encontrar en Unity, donde la *spot light* como se puede evidenciar es un haz de luz directa en forma de cono que sirve para abarcar áreas de diferente tamaño, allí se puede configurar el ángulo y rango de inclusión sobre los objetos que se vayan a iluminar, la intensidad, el color del haz de luz y su ubicación en el espacio, *point light* esta luz permite recrear el destello de un bombillo reflejado en la pared donde esta instalado, es una luz focalizada en un solo punto, allí se le puede configurar el color, la intensidad, el rango de uso, a diferencia de la *directional light* esta no recrea sombras en su recorrido.

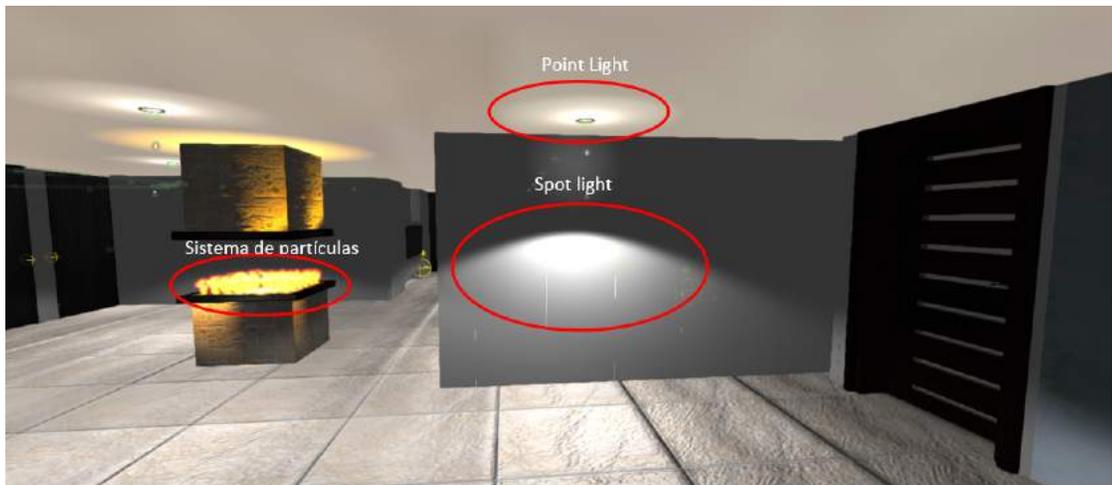


Figura 3.20: Tipos de luces presentes en Unity.

El sistema de partículas normalmente se utiliza para recrear elementos como fuego, fuentes de agua, aire y se configura dándole una textura según lo requerido a cada partícula, lo único que se debe hacer es recrear la textura para una partícula y el programa reproduce esa misma para las demás, allí puede variar las características como la intensidad, velocidad de reproducción, retardo entre partículas, en fin, se puede configurar según lo que más se adecue al sistema que se necesite. En la figura 3.21 se muestra la apariencia de un sistema de partículas implementado en Unity, en la parte derecha de la imagen se puede ver la lista de modificaciones que se puede encontrar para cada sistema.

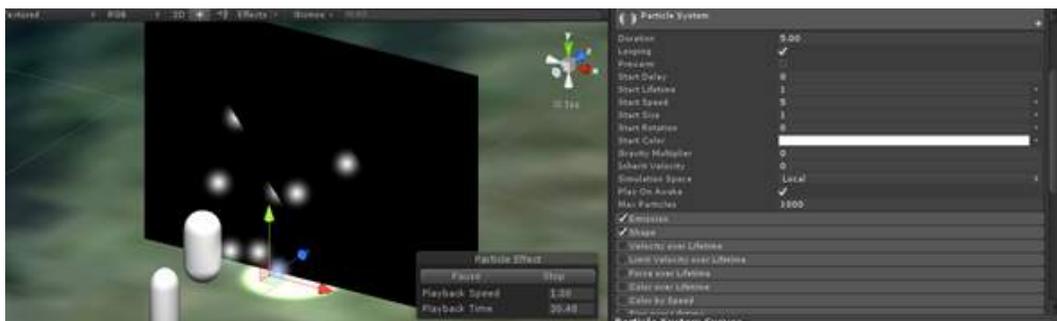


Figura 3.21: Sistema de partículas en Unity.

Sonido y vídeo

Para introducir un vídeo en Unity se utiliza el mismo procedimiento que se describió anteriormente para crear una nueva textura solo que ahora se arrastra el vídeo desde la ventana *project* hasta el espacio destinado para tener una nueva textura en el *inspector*, luego se hace lo mismo con el espectro de sonido que se sustrae del mismo vídeo, el cual por naturaleza del programa queda como un *Audio source*. En la figura 3.22 se muestran las cualidades de vídeo y sonido en el espacio virtual al ser aplicados como texturas.

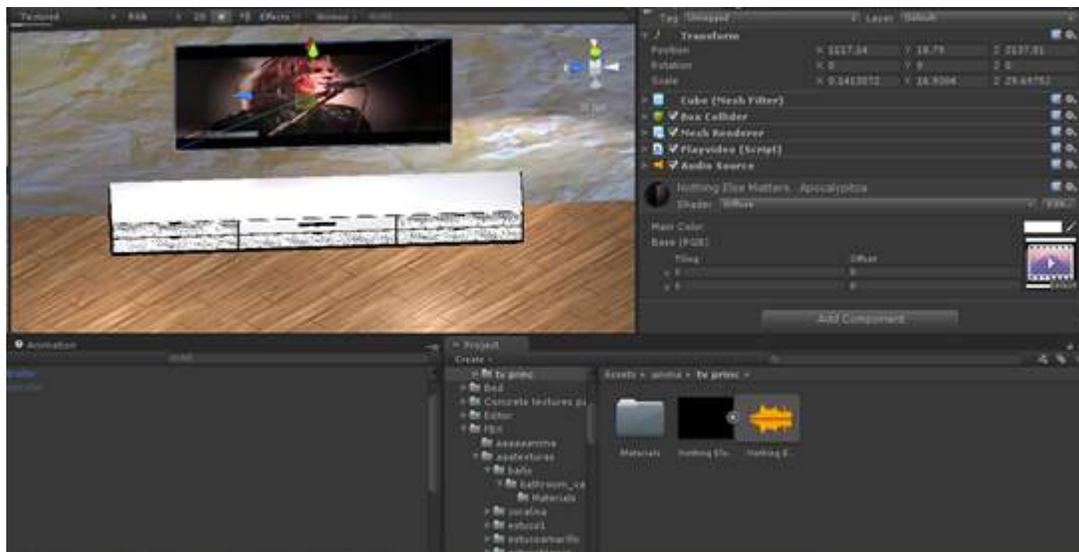


Figura 3.22: Manejo de vídeo y sonido en Unity.

Manejo del personaje

En este apartado se explica cómo maniobrar el personaje que va recorriendo físicamente el entorno generado, se resalta el hecho que tal controlador de personaje esta predeterminado en la carpeta que se genera al guardar un nuevo proyecto, y se suma a la escena requerida de la siguiente manera.

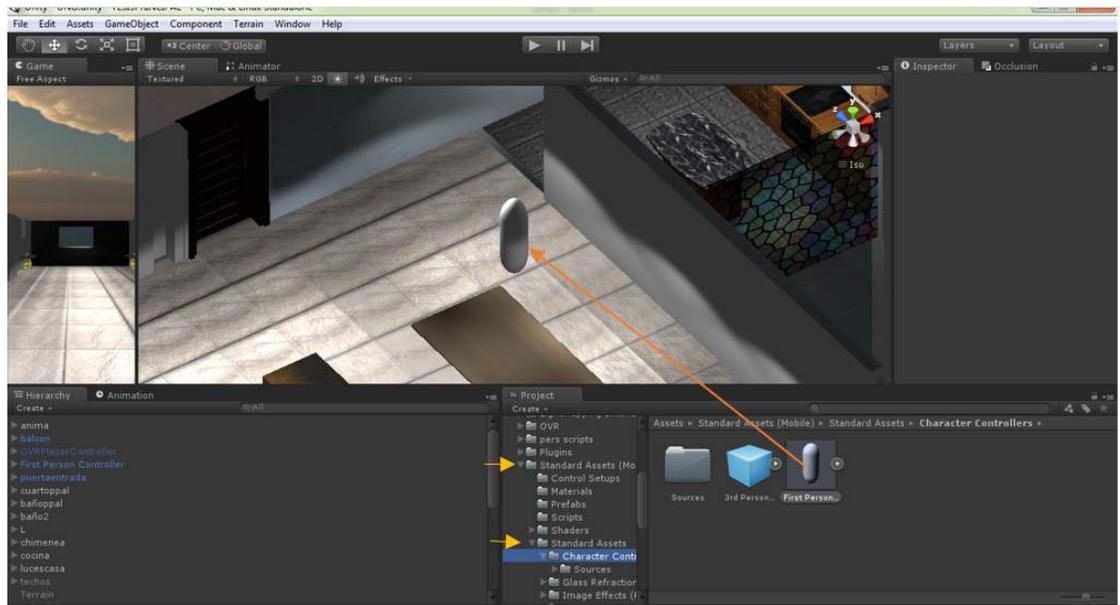


Figura 3.23: Manejo de personaje

Se busca la carpeta *STANDARD ASSETS (MOBILE) >> STANDARD ASSETS >> CHARACTER CONTROLLERS* luego se arrastra la figura que aparece como *FIRST PERSON CONTROLLER* a la escena actual ya el controlador por defecto tiene las siguientes configuraciones:

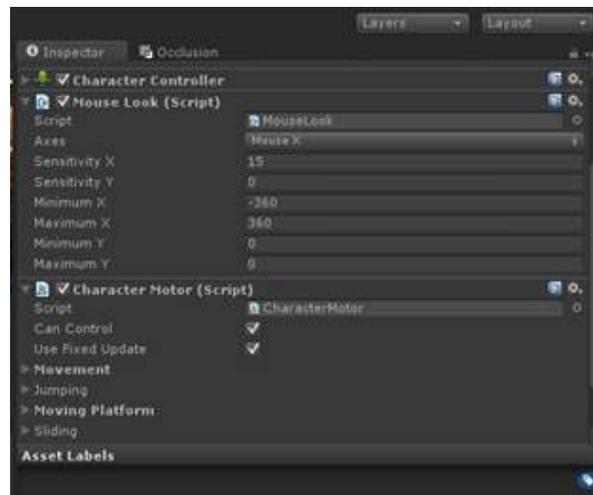


Figura 3.24: Plantilla de modificación del personaje

Donde se puede variar la sensibilidad que tiene el controlador para mirar hacia sus costados y definir variables como velocidad de movimiento sobre el plano, peso del personaje y gravedad. Para el caso puntual donde se necesita un personaje controlador utilizando OCULUS VR se debe descargar una carpeta de extensiones especializadas para este fin directamente de la página de Oculus. Y nuevamente se hace el procedimiento de copia a la carpeta madre del proyecto donde se creara un espacio destinado para las extensiones OVR. Para importar el controlador OVR a la escena se hace de la misma manera:

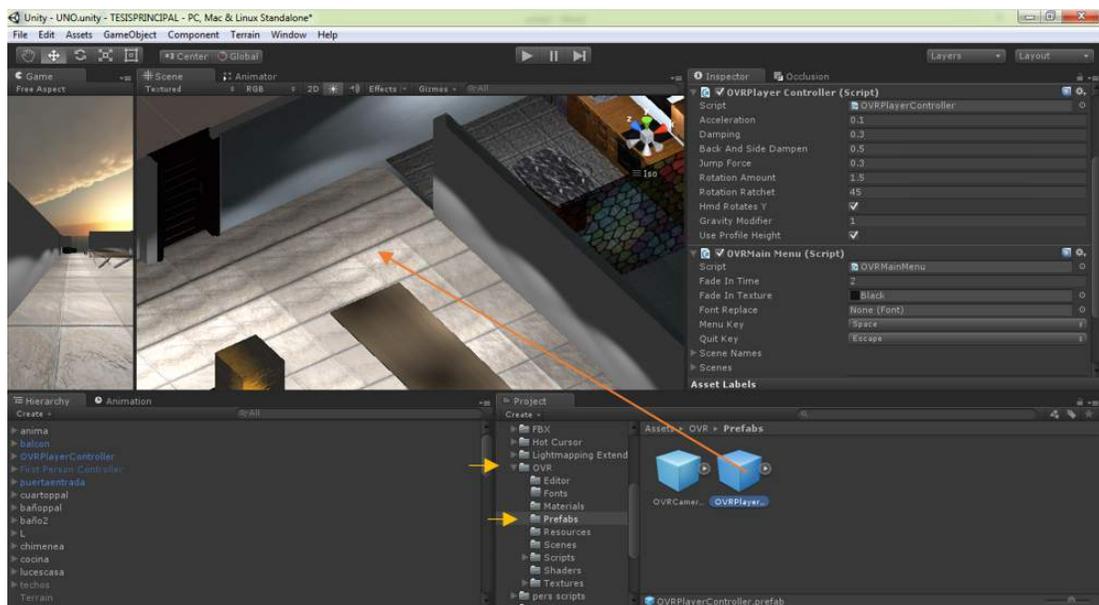


Figura 3.25: Extensiones OVR en Unity

Descripción de *Scripts*

Los *scripts* son la programación fundamental para enlazar los periféricos con el proyecto, su esencia está basada en el lenguaje de programación C# (su mayoría) aunque también se puede enlazar con JAVA y BOO, los *scripts* son indiferentes a su aplicación, puede ir desde leer puertos serial hasta resaltar un pivote de un elemento, muchos de los elementos prefabricados Unity tienen *scripts*; para este proyecto se hizo un número de *scripts* que se ejecutan al mismo tiempo dentro de la escena, a continuación se explica la manera de crear un nuevo *script* y la razón de cada uno:

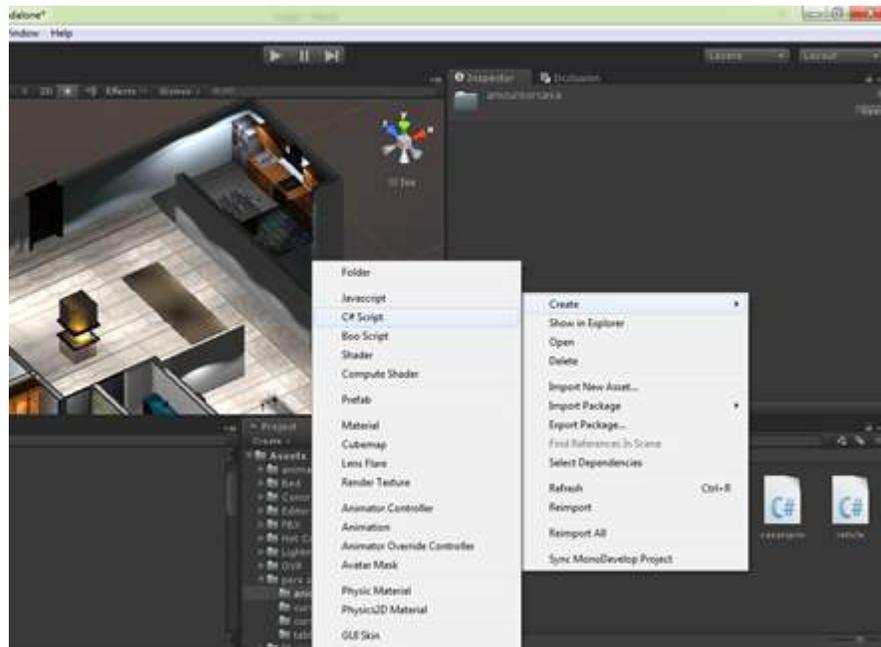


Figura 3.26: Como crear un *script*

Se da click derecho sobre un espacio vacío en la carpeta que se desee en la ventana INSPECTOR luego que sale la pestaña auxiliar se da CREATE >> C# SCRIPT (si se quiere trabajar en C#).

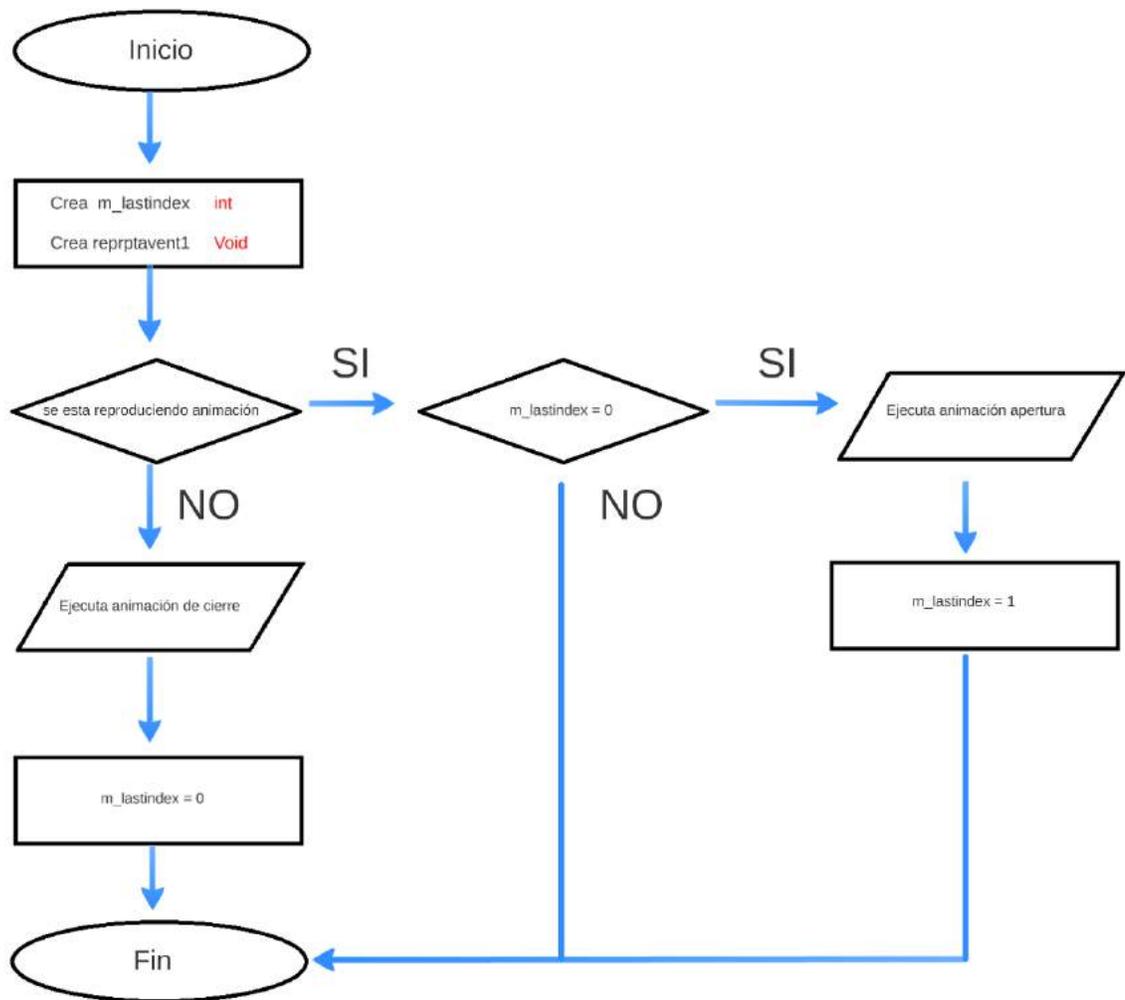


Figura 3.27: Diagrama de flujo, enlace animaciones-click del mouse

Este *script* es requerido sobre cualquier elemento animado dentro de la escena, es decir, hay uno de estos por cada animación ya que es la ejecución interna de la misma, tiene la estructura de un switch esto quiere decir que con el mismo comando o click puede ejecutar animaciones contrarias como por ejemplo aperturas y cierres evitando cruces entre las mismas, estos cruces pueden pasar por ejemplo si se da doble click a la misma animación en un tiempo corto.

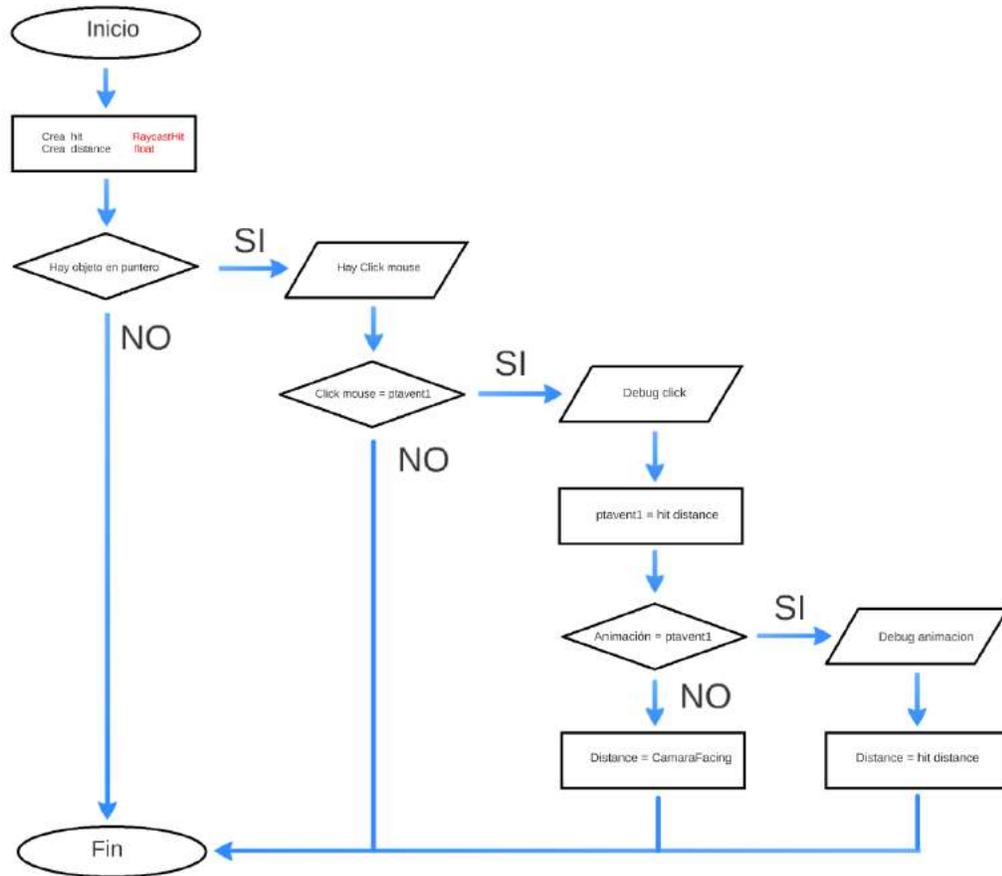


Figura 3.28: Diagrama de flujo, Controlador de animaciones

La funcionalidad de esta programación detectar el uso del botón izquierdo del mouse y ejecutar las animaciones llamando el anterior script para reproducirlas, primero pone un puntero fijo delante del controlador detectando su vector de movimiento hacia adelante, luego verifica si se utilizó el botón y si lo requiere reproduce la animación, el puntero en su programación tiene definido alejarse o acercarse (graduable) tanto como un objeto al frente se lo permite, es decir, si hay un objeto a 3 m el puntero va hasta los 3 metros si hay un objeto más cerca también el puntero lo estará,

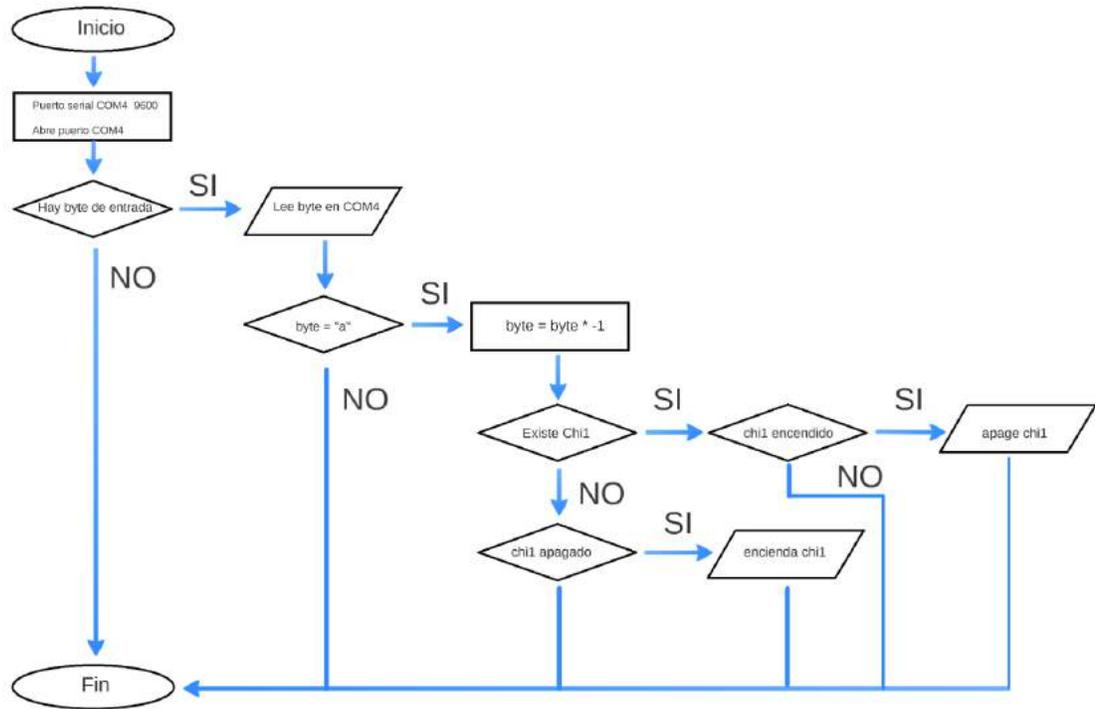


Figura 3.29: Diagrama de flujo, comunicación proyecto-tablet

Finalmente la Tablet tiene su comunicación con el programa por medio de puertos COM (serial), para lograr tal enlace se utiliza este *script* donde se configura y abre el puerto, luego si detecta que en el teclado se oprime la tecla que habilita la lectura del puerto para este caso “t”, lee el dato entrante, para este caso si este dato es “a” prende o apaga dependiendo del estado anterior dos luces llamadas crar y crab. Este principio switch se utiliza para todas las luces de cualquier tipo o sistemas de partículas existentes dentro de la escena.

Compilar en formato oculus

Finalmente para compilar una escena Unity en el formato de dos pantallas compatible con las gafas VR, se busca en la barra de herramientas superior FILE >> BUILD SETTINGS o CTRL + SHIFT + B. en la figura se muestra los pasos a seguir en la interfaz gráfica de unity.

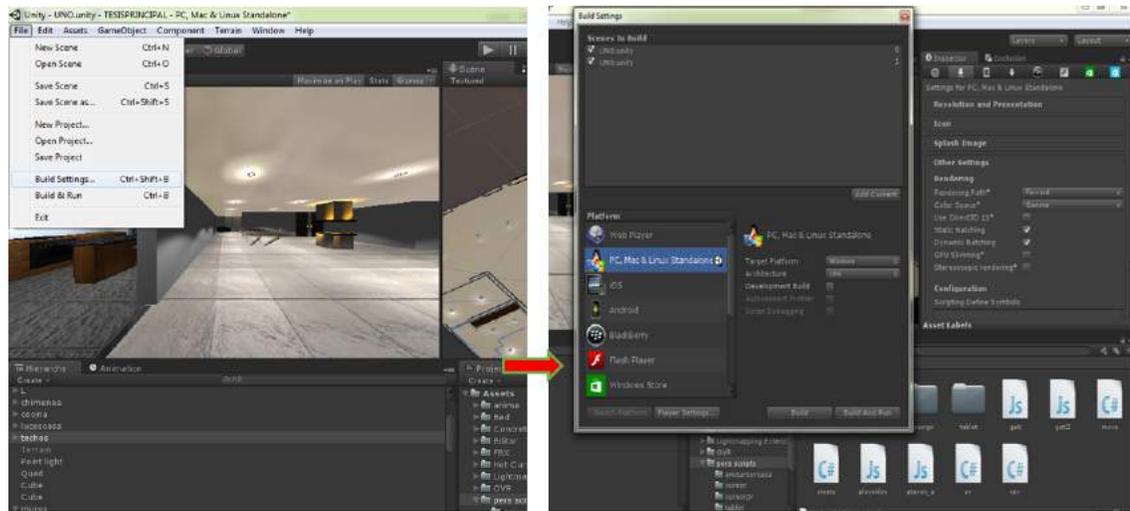


Figura 3.30: Compilar en formato Oculus

3.6. DESARROLLO DEL HARDWARE

Como tal durante el proyecto no se desarrollo un componente de hardware, en él solo se integraron diferentes herramientas tecnológicas que permiten desarrollar una estrecha relación con la realidad virtual manipulando los sentidos y la percepción del espacio. A continuación se presentaran de forma detallada los componentes de hardware usados durante el proyecto.

- OculusDK2**- Segundo kit prototipo del visor para realidad virtual creado por la empresa Oculus VR del cual solo se han vendido versiones para desarrolladores, este incluye; gafas VR, sensor tracker eye, 2 pares de lentes bifocales, adaptador de corriente y adaptador HDMI a DVI.
- Joystick**- Palanca de mando usada para dirigir el movimiento del elemento al cual esta integrada, comúnmente junto a ella se incluyen pulsadores para ampliar el campo de control.
- Tablet**- Dispositivo electrónico portable similar a un computador pero un poco menos robusto, comúnmente equipados con una pantalla táctil capacitiva de gran precisión y resolución, giroscopio, acelerómetro, cámara y micrófono.

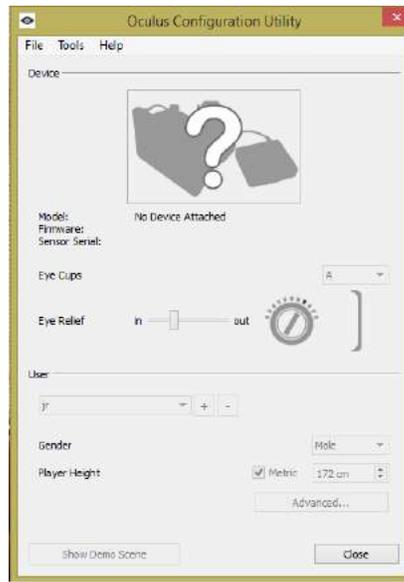
-Bluetooth- Modulo de transmisión de datos de forma inalámbrica con un rango fijo de alcance, la transmisión se produce mediante una banda de radio frecuencia de 2.4 GHz.

-Modulo Serial-USB- Tarjeta electrónica que permite la decodificación de datos, una serie de pulsos informáticos que son convertidos a señales de amplitud estándar en protocolo USB.

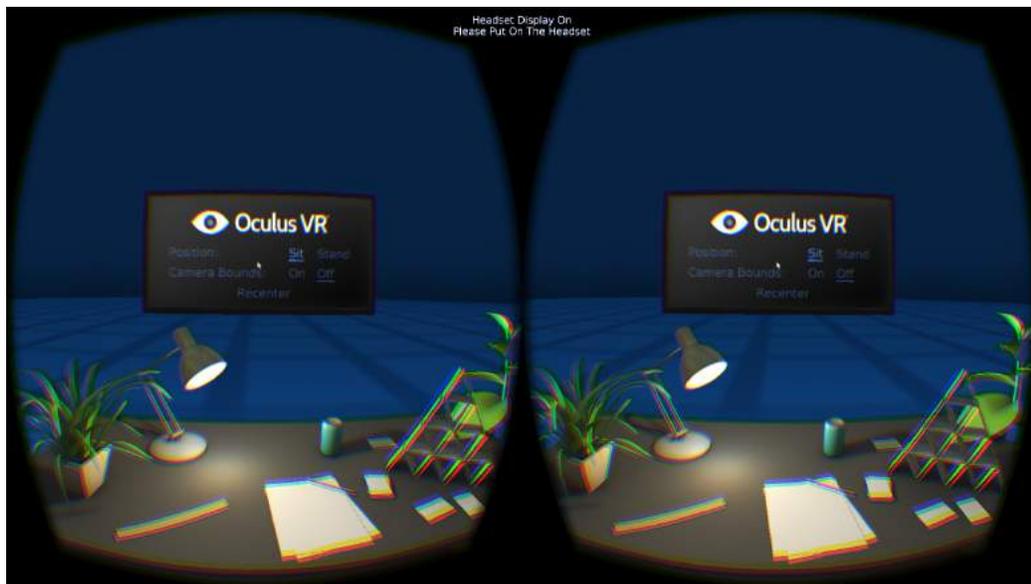
3.6.1. Caracterización Oculus rift DK2

Cuando se inicia a desarrollar un proyecto de esta categoría es esencial saber el funcionamiento interno de las herramientas con las cuales se trabaja, mas aun cuando se trata de dispositivos tecnológicos, ya que bajo algunas condiciones especiales estos componentes pueden comportarse de manera diferente saliendo de los parámetros deseados, es por esto que debe realizarse una prueba experimental con cada componente que haga parte del hardware en el proyecto.

Para iniciar la experiencia con la realidad virtual es necesario instalar el programa que permite configurar la herramienta de hardware en la que se visualiza esta realidad alterna en la figura 3.31a se puede ver la interfaz gráfica del programa antes mencionado, donde al conectar las gafas al ordenador por el puerto USB estas serán reconocidas por el programa y se mostraran allí con algunas especificaciones como el modelo, firmware y el numero serial del sensor, ahora se debe configurar el modo de ejecución de la pantalla esto se puede hacer desde la pestaña *tools* en el programa luego se selecciona la opción *Rift display mode*, esto abrirá un recuadro con múltiples opciones a elegir, en el solo debemos seleccionar las opciones *Extend desktop to the HMD* y *rift saver* posteriormente dar click en *Apply* y luego *close*. El primer contacto que se tiene directamente con las gafas *oculus rift DK2* luego de haber realizado las configuraciones previas, es a través de un demo que se encuentra en el mismo programa de configuración en la parte inferior izquierda de la interfaz principal se encuentra el botón que nos llevara allí (ver figura 3.31b).



(a) Interfaz del programa que permite configurar las gafas VR



(b) Demo, primer contacto con la realidad virtual

Figura 3.31: Primer contacto con las gafas VR y el universo virtual

Al entrar por primera vez en el mundo de la realidad virtual por medio de este demo, se percibe una sensación de libertad al ver que no existe limite sobre un horizonte infinito, en la escena de prueba creada por *Oculus VR* se permite ver diferentes elementos

que muestran el potencial de la realidad virtual para recrear lugares de la vida real, la capacidad de tener un giro de 360° es fundamental para lograr inmersión en la realidad virtual, en este aspecto no se percibe problema alguno ya que no se ven imágenes pixeladas o recuadros cortados, por ultimo, se puede ver como se integra el sistema *tracker eye* con las gafas, ya que el demo permite ver la trayectoria de los vectores que limitan el área de alcance entre el sensor y las gafas. en la figura se pueden ver los limites de operación para movimientos verticales.

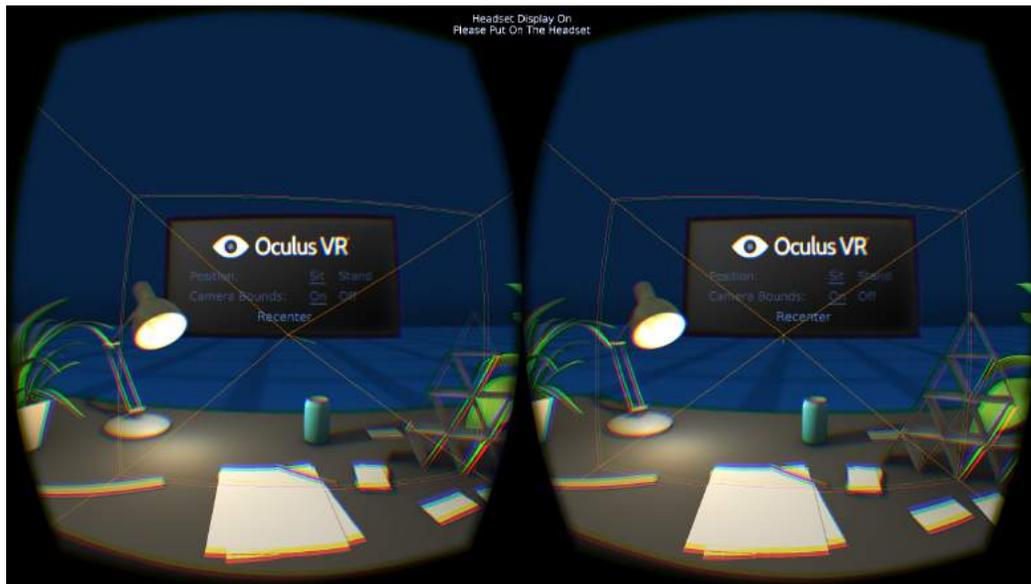


Figura 3.32: Demo, vectores limites entre sensor y gafas VR

3.7. INTEGRACIÓN

Finalmente se llega a la etapa en que se unen los diferentes componentes de software y hardware haciendo posible la concepción de el mundo virtual propuesto, siguiendo la arquitectura mostrada en la sección 3.4, los componentes electrónicos utilizan en total cuatro puertos USB y uno HDMI, de los cuales las gafas VR ocupan dos USB y el único HDMI, el joystick y modulo de comunicación serial-USB ocupan los dos puertos USB restantes, debido a esto es necesario usar un dispositivo splintter para USB que aumente la cantidad de puertos USB en el computador, ya que los existentes no son suficientes para llevar a cabo la conexión. Luego de poder llevar a cabo las conexiones físicas en

los puertos se verifica que las mismas también existan de forma virtual iniciando los programas que controlan cada dispositivo por separado (*Oculus configuration utility*, *Xpadder*, administrador de dispositivos de Windows), hecho esto iniciar la herramienta de realidad virtual desarrollada en formato soportado para las gafas Oculus rift DK2, en este punto ya es posible ver físicamente todo el trabajo desarrollado durante el proyecto. En la figura 3.33 se puede ver la conexión del hardware y software.



Figura 3.33: Integración de hardware y software, conexión física

3.7.1. Comportamiento de hardware en la realidad virtual

La primera prueba se realiza en un computador *Hewlett Packard* de bajo rendimiento, el cual cuenta con un procesador *Athlon x2* de *AMD*, memoria RAM de 2 Gb y no tiene tarjeta gráfica independiente, esto ocasiona que el primer acercamiento directo con la realidad virtual no tenga el impacto esperado; Las imágenes en las gafas VR no alcanzan las dimensiones suficientes para permitir inmersión virtual, ni se refrescan lo suficientemente rápido para permitir navegar cómodamente sobre la misma (sensación de que se encuentra saltando el personaje virtual), es difícil identificar algunas formas y figuras ya que se ven pixeladas perdiendo su forma característica, la comunicación Bluetooth establecida por medio del módulo serial-USB con el computador se lleva a

cabo pero se nota un retraso en la señal ya que no se producen los cambios esperados de forma instantánea.

La solución al problema anterior se da probando la herramienta de realidad virtual en un ordenador con mayores capacidades de procesamiento (A10 *AMD*), almacenamiento temporal (8 Gb, memoria *RAM*) y gráfica (tarjeta *Radeon Graphics* HD 8760M, 2Gb), estas nuevas características cambian por completo la experiencia anterior, en esta ocasión los gráficos se perciben con gran detalle y definición, el tamaño de la imagen se ve mucho más amplia impidiendo notar que se encuentra en una realidad virtual, al querer dar una vista de 360° del apartamento se alcanza a ver la señal un poco retrasada cuando el giro se realiza con mucha velocidad, el movimiento del personaje en el mundo virtual se da sin problemas, la comunicación tablet-ordenador en ocasiones se torna lenta, pero, aun funciona y produce el efecto esperado.

Capítulo 4

VALIDACIÓN EXPERIMENTAL

Ya concluidas todas las etapas previas es posible evaluar la herramienta arquitectónica de realidad virtual desarrollada, se recomienda el uso de una silla giratoria para evitar posibles accidentes por pérdida de equilibrio, también se sugiere, no prolongar el uso de las gafas VR más de 15 minutos sin descansar, en la figura 4.1 se muestra el punto de partida en el apartamento.

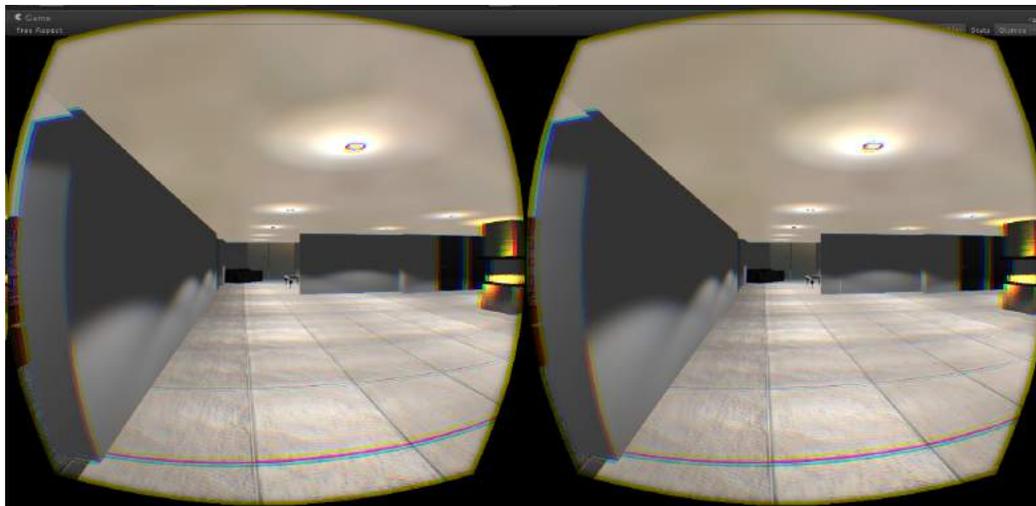


Figura 4.1: Punto de partida en apartamento virtual

4.1. RECORRIDO VIRTUAL

Al ejecutar la aplicación en formato Oculus VR y colocarse el visor de realidad virtual se puede iniciar la experiencia de exploración e interacción, inicialmente se explicara el contenido del apartamento en modo vivienda y posteriormente se llevara a cabo la explicación correspondiente en modo oficina; En el espacio virtual al encontrarse en el punto de partida (ver figura 4.1) se encuentra al lado izquierdo con el módulo de la vivienda correspondiente a la cocina (ver figura 4.2), en él se podrán observar los componentes de una cocina regular (nevera, estufa, lavaplatos, y muebles de madera), adicional a esto se encuentra una mesa auxiliar para cuatro personas.

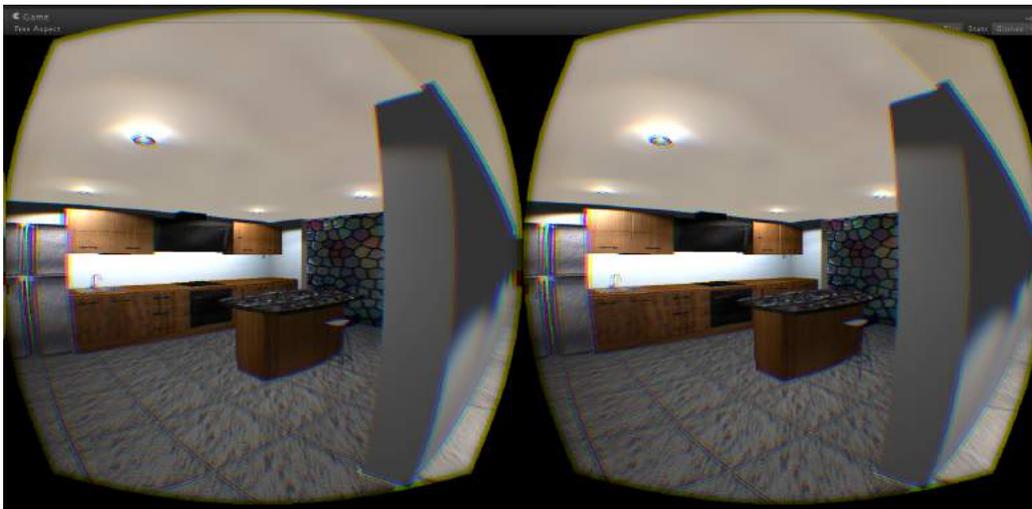


Figura 4.2: Cocina apartamento virtual

Continuando con el recorrido en modo vivienda, en la parte posterior del bloque social se encuentra una sala social en la que encontramos un sofá en L y un puff frente a este, ambos de color negro (ver figura 4.3), en medio de ellos se encuentra una mesa de centro hecha en cristal; este es el primer módulo del recorrido en el que se encuentran aplicados los espacios dinámicos, en él se encuentra una pared plegable brindando privacidad al espacio, las ventanas pueden ser cubiertas con blackout impidiendo el paso de luz hacia el apartamento.

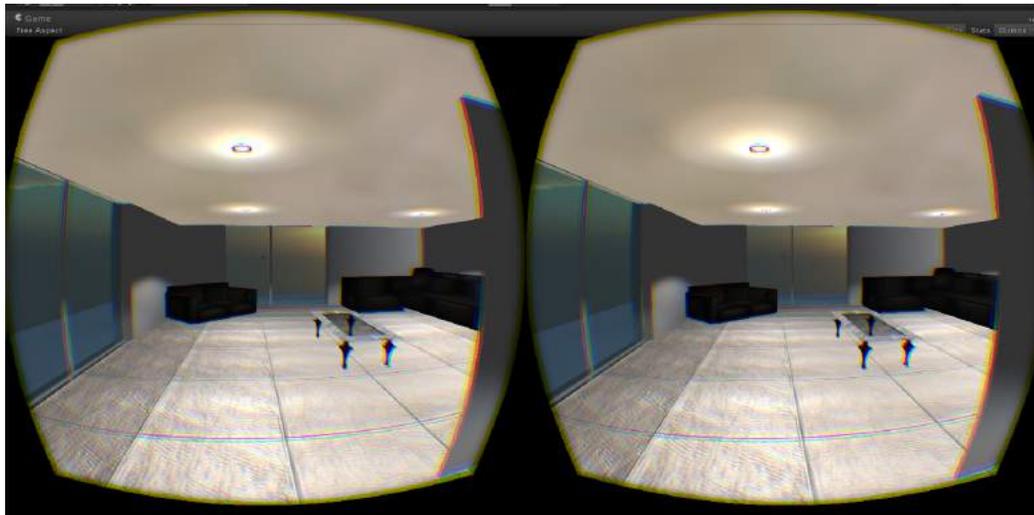


Figura 4.3: Sala social apartamento virtual

La siguiente habitación que se encuentra en la exploración es la habitación principal, en ella se encuentran los siguientes elementos, una cama doble, dos nocheros con su respectiva lámpara, una puerta ventana con salida al balcón, un televisor pantalla plana y un mueble de entretenimiento. En la figura 4.4 se muestra la habitación vista desde la puerta de ingreso a ella.

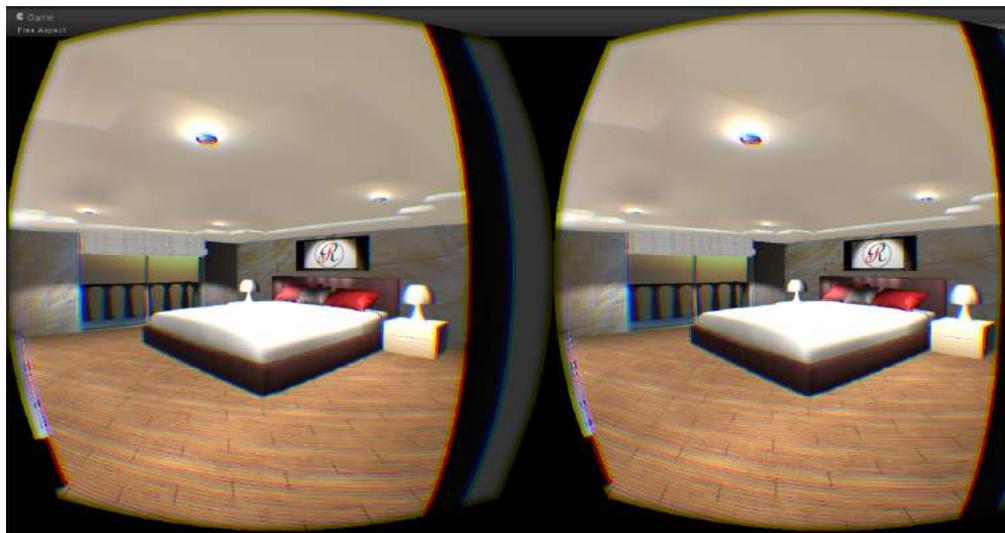
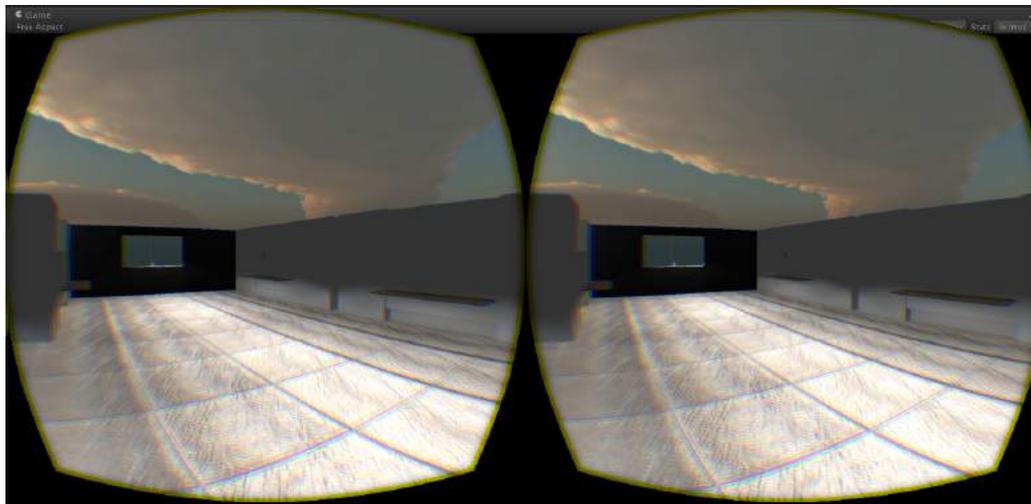
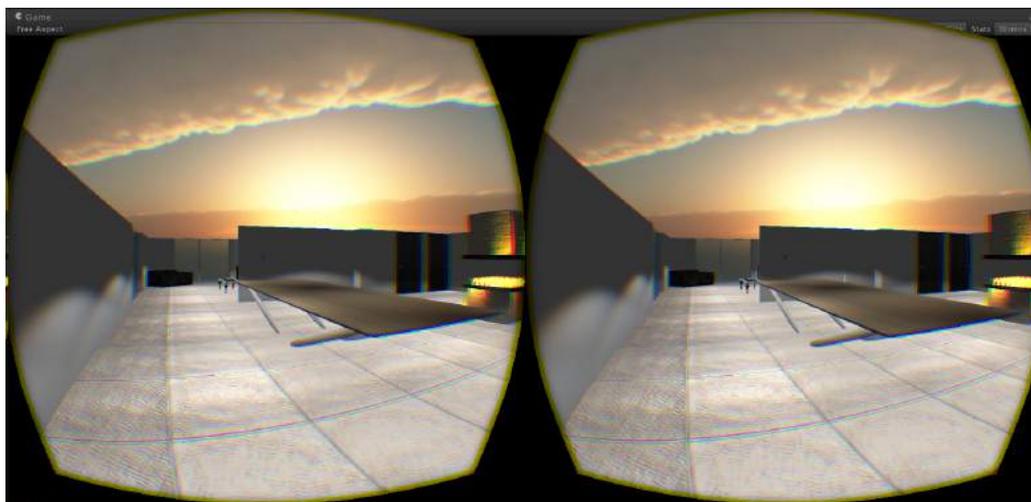


Figura 4.4: Habitación principal apartamento virtual

En la modalidad oficina del apartamento solo cambian dos módulos, ambos ubicados en la zona social e incluyen una mesa abatible que permite tener una sala de juntas ubicada en medio de la sala y la chimenea (ver figura 4.5b), el segundo espacio que se ve modificado, son las habitaciones 2 y 3 ubicadas al costado derecho del punto de partida, allí se encontraran tres escritorios amplios y cómodos, el primero se denomina como recibidor ya que se encuentra enseguida de la puerta de entrada, los dos restantes se encuentran habilitados para uso abierto, a disposición del residente del espacio habitacional. En la figura 4.5a se encuentran los escritorios antes descritos en una amplia zona de trabajo.



(a) Escritorios en el apartamento virtual



(b) Mesa abatible, sala de juntas

Figura 4.5: Área de trabajo apartamento virtual

4.2. INTERACCIÓN

La flexibilidad de los espacios en la vivienda se muestra mediante animaciones interactivas, de las cuales se pueden encontrar 20 distribuidas a lo largo y ancho de la vivienda virtual y pueden ser activadas al señalar con el puntero girando la cabeza, enfocar el objeto animado y luego accionar el disparador que se encuentra en la parte posterior del joystick. Las animaciones se pueden catalogar en cinco grupos diferentes, estos grupos son:

- *Puertas ventana y puertas normales*, se encuentran en cada lugar que requiere ser cerrado y brindar privacidad a la vivienda.
- *Paredes plegables*, usadas para separar ambientes y espacios completos, se encuentran tres de este tipo ubicadas en la sala, fin de habitación 2 y 3.
- *Wall bed y mesa abatible*, usados como muebles multiuso para brindar comodidad, espacio y versatilidad, se encuentran distribuidos entre la parte central de la vivienda y las habitaciones 1 y 2.
- *Blackout*, son cortinas enrollables usadas para impedir el paso de luz solar desde el exterior hacia el interior, se encuentran frente a las ventanas.
- *Video y sonido*, son reproducciones digitales de vídeo y sonido sobre superficies específicas, se encuentra un único vídeo sobre el televisor de la habitación principal

4.3. PERCEPCIÓN

Durante la experiencia virtual se detallan aspectos agradables y creíbles que permiten ver la realidad virtual como una herramienta de gran futuro, El apartamento recreado muestra texturas, suavizados, reflejos y movimientos realistas que permiten aumentar la inmersión en el apartamento creado, se destaca apariencia del cielo visible en el proyecto (ver figura 4.6) donde se ven matices de colores en el extenso cielo.



Figura 4.6: Cielo visible en el apartamento

Capítulo 5

RESULTADOS Y CONCLUSIONES

La razón de ser de este proyecto de grado es responder a la pregunta de investigación usando las herramientas tecnológicas que han sido acercadas por la ingeniería mecatrónica durante el curso de pregrado, y así, luego de ser culminado el proyecto concluir si los objetivos planteados inicialmente se lograron satisfactoriamente. a continuación se presentaran las conclusiones a los objetivos propuestos.

Objetivo General

La domótica puede presentarse como solución a las problemáticas sociales actuales, ampliando su campo de acción al incluir las tendencias habitacionales contemporáneas, que se muestran en desacuerdo con los riesgos ecológicos, los grandes espacios y los consumos excesivos, y por el contrario se abren a las nuevas ideas, integraciones tecnológicas y reducciones de costos tanto monetarios como ambientales, lo que permitió mostrar a los espacios dinámicos como una solución a los problemas de crecimiento actuales, comprobando que hacer viviendas adaptables con espacios flexibles permite usar el 68 % del espacio habitacional para otras funciones que no se relacionan con las labores domesticas y que son de igual forma indispensables en una ciudad.

Objetivos específicos

- Se usaron una amplia gama de herramientas computacionales para lograr el apartamento recreado en 3D, en este se puede confirmar la efectividad de los espacios modulares flexibles al usar $83,34m^2$ de el area privada del apartamento para incluir muebles y accesorios que permiten múltiples ambientes ampliando la utilidad del área.

- Los componentes necesarios para el proyecto desarrollado fueron clasificados y plasmados de forma esquemática a manera de explicar la arquitectura diseñada, de ella se puede concluir que para concevir cualquier proyecto que incluya realidad virtual, la herramienta mas importante a usar es el ordenador ya que por el pasa toda la información que permite esta experiencia inmersiva.
- Se logra de manera exitosa visualizar un modelo arquitectónico en el que se implementan espacios dinámicos con herramientas de realidad virtual, permitiendo explorar e interinar sobre él de manera practica, dando una perspectiva realista de un posible modelo fisico.
- Las herramientas tecnológicas alternativas como la realidad virtual se convierten en una novedosa manera de mostrar nuevas ideas y soluciones sin perder la impresión que deja un modelo físico real ya que como se mostró a lo largo del proyecto, la capacidad de los componentes tecnológicos actuales permiten tener una experiencia de alta inmersión en los diferentes lugares que puedan presentarse sobre la plataforma virtual.

Bibliografía

- [1] John Arango Florez et al. *El Mueble como Estructurador del Espacio en la Vivienda Moderna*. PhD thesis, Universidad Nacional de Colombia, Sede Medellín, 2012.
- [2] DUS Architects. 3d print canal house. Web site, Marzo 2014.
- [3] GMA Arquitectos asociados. *Communique*, 2013.
- [4] Frank Biocca and Mark R Levy. Virtual reality as a communication system. *Communication in the age of virtual reality*, pages 15–31, 1995.
- [5] William D Callister. *Introducción a la ciencia e ingeniería de los materiales*, volume 1. Reverté, 2002.
- [6] María Fernanda Coello Toral. La madera en la arquitectura japonesa: aplicación en paneles en el diseño de una vivienda local. 2011.
- [7] Le Corbusier and Enrique Luis Revol. *La ciudad del futuro*. Infinito, 1962.
- [8] DANE. Licencias de construcción-elic. *Boletín técnico*, 2015.
- [9] Grupo de Memoria Histórica. *¡BASTA YA! Colombia memorias de un país en guerra y dignidad*. Imprenta Nacional, 2013.
- [10] NAU: Architecture & Design. Living roof, 2011.
- [11] Rueda Enciso Jodé Eduardo. Rother leopoldo. *Gran Enciclopedia de Colombia del Círculo de lectores*, 1993.
- [12] Asociacion española para la calidad. Smart cities - ciudades inteligentes. 2012.
- [13] Universidad de Chile Facultad de arquitectura y urbanismo. Con éxito finalizó primer workshop de realidad virtual en arquitectura. Web site, Enero 2015.

-
- [14] Sigfried Giedion and Isidre Puig Boada. *Espacio, tiempo y arquitectura*. Dossat, 1982.
- [15] CL Jimeno. La domótica como solución de futuro. *Madrid Ahorra con Energía*, 1:72, 2007.
- [16] nARCHITECTS. My micro ny, Enero 2013.
- [17] E. Neufert and P. Neufert. *Arte de proyectar en arquitectura*. Gustavo Gili Diseño, 2001.
- [18] Department of Housing Preservation and Development of New York City. Winner of adapt nyc competition to develop innovative micro-unit apartment housing model, July 2012.
- [19] Gustavo Carrasco Pérez. La vivienda económica en le corbusier. *Revista INVI*, 3(4), 1988.
- [20] Marie-Laure Ryan. La narración como realidad virtual. *La inmer*, 2004.
- [21] Graziella Trovato. Definición de ámbitos de flexibilidad para una vivienda versátil, perfectible, móvil y ampliable. *Ciudad y Territorio. Estudios Territoriales*, 41(161-16):599–614, 2009.

ANEXO A

Ficha técnica Bluetooth HC-05

fluctuant in the range of 30-40mA. The mean current is about 25mA. After paring, no matter processing communication or not, the current is 8mA. There is no sleep mode. This parameter is same for all the Bluetooth modules.	fluctuant in the range of 30-40 m. The mean current is about 25mA. After paring, no matter processing communication or not, the current is 8mA. There is no sleep mode. This parameter is same for all the Bluetooth modules.
Reset: PIN11, active if it's input low level. It can be suspended in using.	Reset: PIN11, active if it's input low level. It can be suspended in using.
Level: Civil	Level: Civil

The table above that includes main parameters of two serial modules is a reference for user selection.

HC-03/HC-05 serial product is recommended.

3. Information of Package

The PIN definitions of HC-03, HC-04, HC-05 and HC-06 are kind of different, but the package size is the same: 28mm * 15mm * 2.35mm.

The following figure 1 is a picture of HC-06 and its main PINs. Figure 2 is a picture of HC-05 and its main PINs. Figure 3 is a comparative picture with one coin. Figure 4 is their package size information. When user designs the circuit, you can visit the website of Guangzhou HC Information Technology Co., Ltd. (www.wavesen.com) to download the package library of protle version.

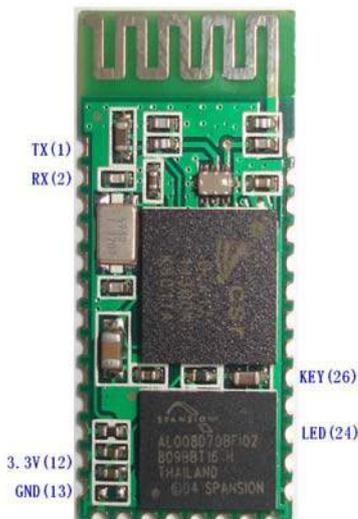


Figure 1 HC-06

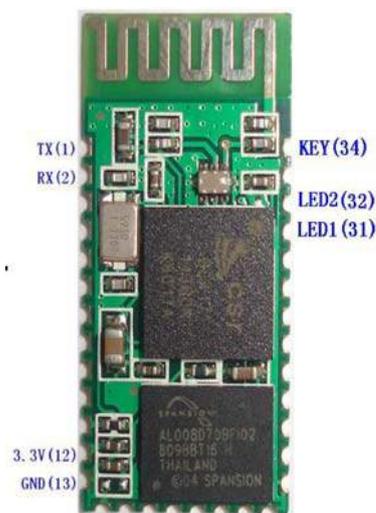


Figure 2 HC-05

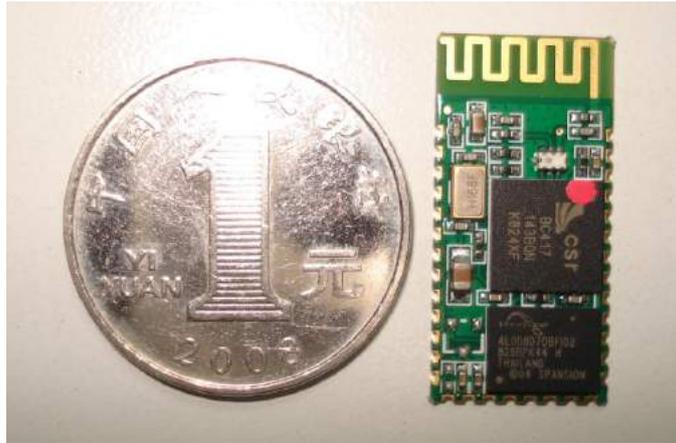


Figure 3 Comparative picture with one coin

LINVOR BLUE T
www.linvor.com

LV-BC-2.0

单位: mm

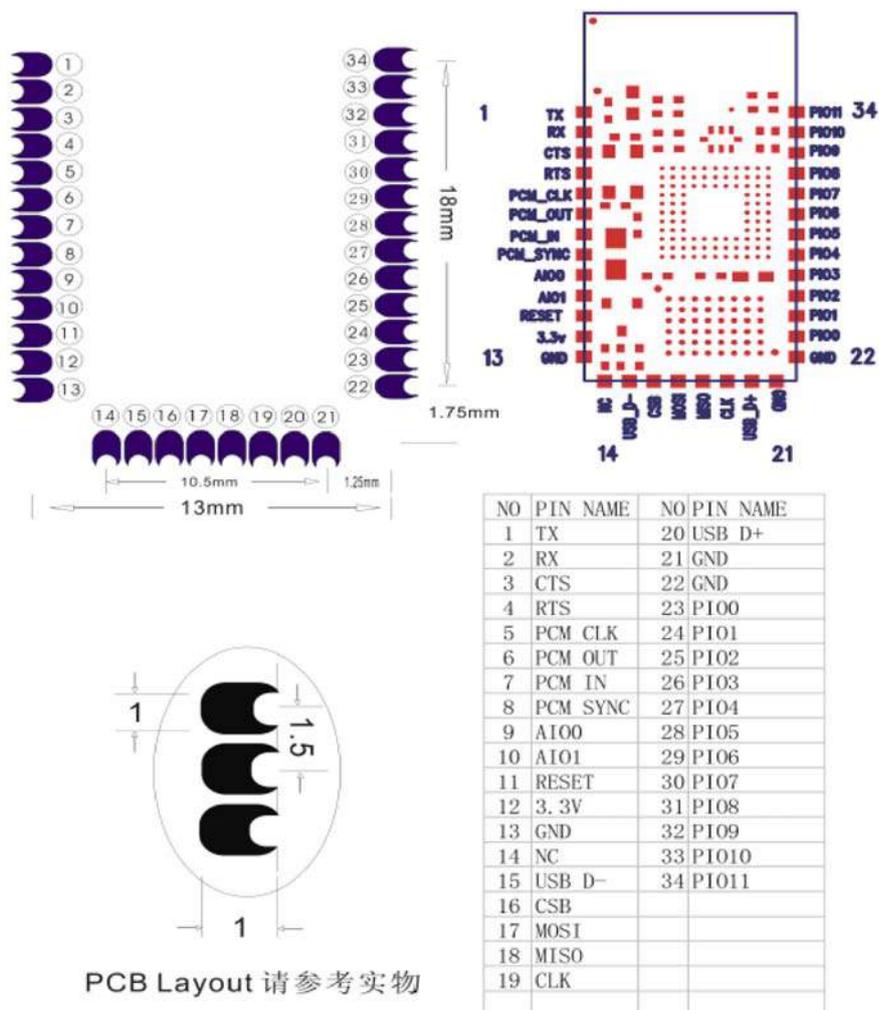


Figure 4 Package size information

4. The Using and Testing Method of HC-06 for the First Time

This chapter will introduce the using method of HC-06 in detail. User can test the module according to this chapter when he or she uses the module at the first time.

PINs description:

PIN1	UART_TXD , TTL/CMOS level, UART Data output
PIN2	UART_RXD, TTL/COMS level, s UART Data input
PIN11	RESET, the reset PIN of module, inputting low level can reset the module, when the module is in using, this PIN can connect to air.
PIN12	VCC, voltage supply for logic, the standard voltage is 3.3V, and can work at 3.0-4.2V
PIN13	GND
PIN22	GND
PIN24	LED, working mode indicator Slave device: Before paired, this PIN outputs the period of 102ms square wave. After paired, this PIN outputs high level. Master device: On the condition of having no memory of pairing with a slave device, this PIN outputs the period of 110ms square wave. On the condition of having the memory of pairing with a slave device, this PIN outputs the period of 750ms square wave. After paired, this PIN outputs high level.
PIN26	For master device, this PIN is used for emptying information about pairing. After emptying, master device will search slaver randomly, then remember the address of the new got slave device. In the next power on, master device will only search this address.

(1) The circuit 1 (connect the module to 3.3V serial port of MCU) is showed by figure 5.

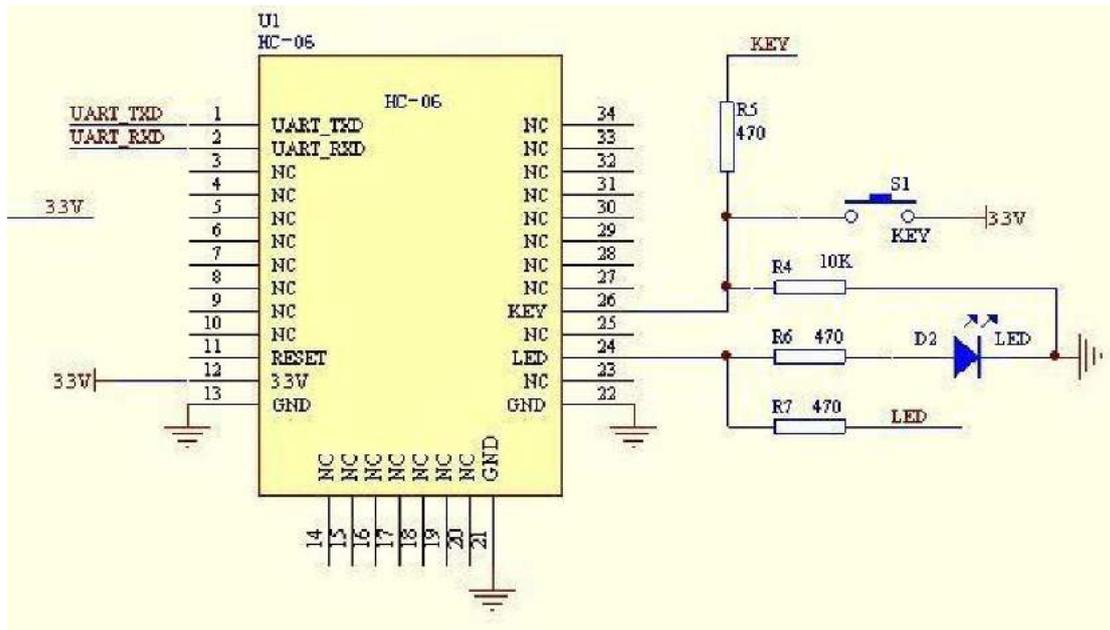


Figure 5 The circuit 1

In principle, HC-06 can work when UART_TXD, UART_RXD, VCC and GND are connected. However, for better testing results, connecting LED and KEY are recommended (when testing the master).

Where, the 3.3V TXD of MCU connects to HC-06's UART_RXD, the 3.3V RXD of MCU connects to HC-06's UART_TXD, and 3.3V power and GND should be connected. Then the minimum system is finished.

Note that, the PIN2:UART_RXD of Bluetooth module has no pull-up resistor. If the MCU TXD doesn't have pull-up function, then user should add a pull-up resistor to the UART_RXD. It may be easy to be ignored.

If there are two MCU which connect to master and slave device respectively, then before paired(LED will flicker) user can send AT commands by serial port when the system is power on. Please refer to HC-04 and HC-06's data sheet for detailed commands. In the last chapter, the command set will be introduced. Please pay attention to that the command of HC-04/HC-06 doesn't have terminator. For example, consider the call command, sending out AT is already enough, need not add the CRLF (carriage return line feed).

If the LED is constant lighting, it indicates the pairing is finished. The two MCUs can communicate with each other by serial port. User can think there is a serial port line between two MCUs.

(2) The circuit 2 (connect the module to 5V serial port of MCU) is showed by figure 6.

Figure 6 is the block diagram of Bluetooth baseboard. This kind of circuit can amplify Bluetooth module's operating voltage to 3.1-6.5V. In this diagram, the J1 port can not only be connected with MCU system of 3.3V and 5V, but also can be connected with computer serial port.

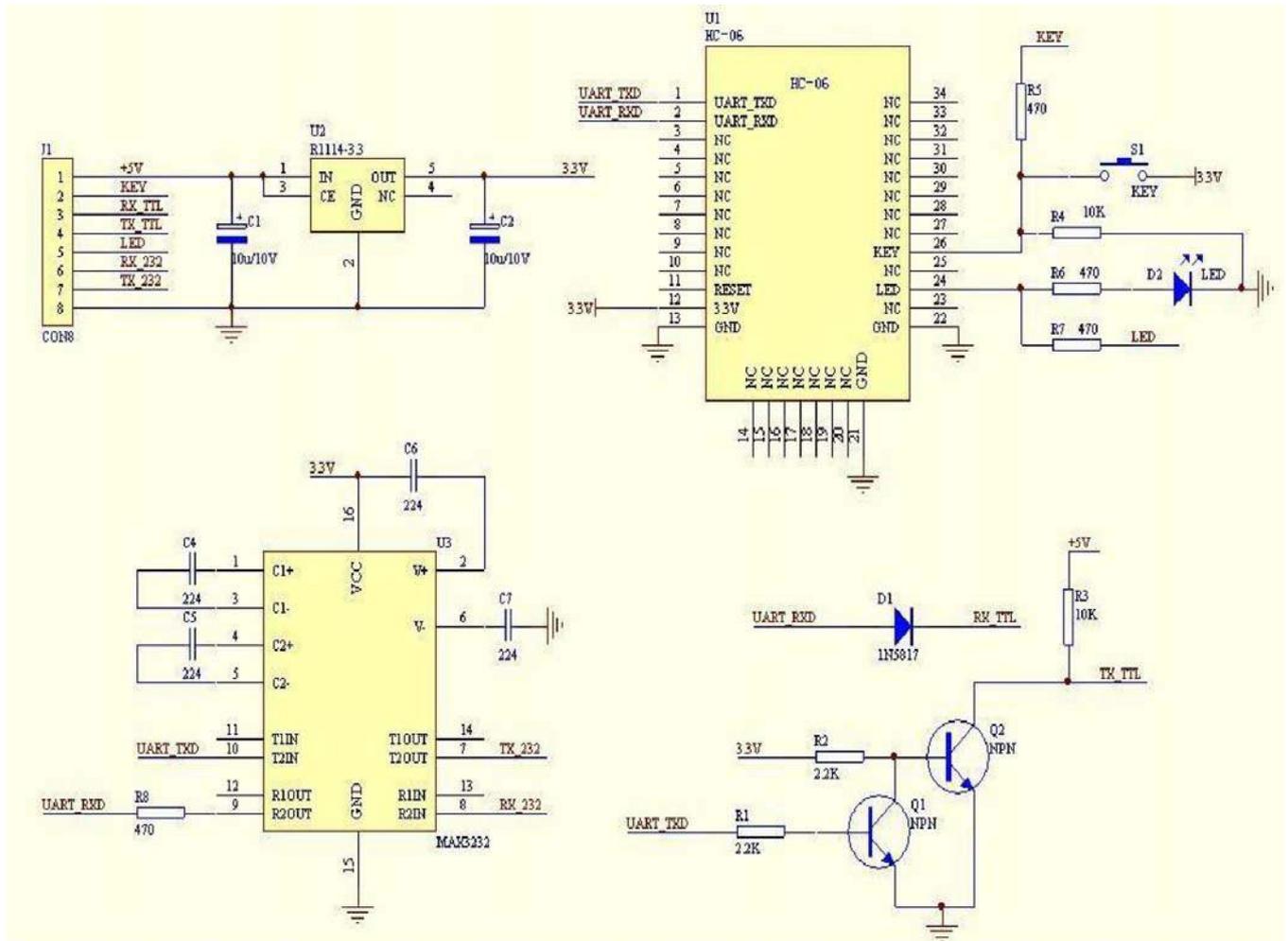


Figure 6 The circuit 2

(3) AT command test

Before paired, the mode of HC-04 and HC-06 are AT mode.

On the condition of 9600N81, OK will be received when user send the two letters AT. Please refer to the last chapter of datasheet for other commands of HC-06. Please pay attention to that sending out AT is already enough, need not add the CRLF (carriage return line feed).

The command set of Version V1.4 doesn't include parity. The version V1.5 and its later version have parity function. Moreover, there are three more commands of V1.5 than V1.4. They are:

No parity (default) AT+PN

Odd parity AT+PO

Even parity AT+PE

Do not let the sending frequency of AT command of HC-06 exceed 1Hz, because the command of HC-06 end or not is determined by the time interval.

(4) Pairing with adapter

User can refer to the download center of the company's website for "The Introduction of IVT" that introduces the Bluetooth module makes pair with computer adapter. That document taking HC-06-D for example introduces how the serial module makes pair with the adapter. That method is like to make pair with cell-phone. But the difference is that cell-phone need a third-party communication software to help. It's liked the kind of PC serial helper of and the hyper terminal. A software named "PDA serial helper" provided by our company is suitable for WM system. It has been proven that this serial module is supported by many smart phone systems' Bluetooth, such as, sybian, android, windows mobile and etc.

(5) Pairing introduction

HC-06 master device has no memory before the first use. If the password is correct, the mater device will make pair with the slave device automatically in the first use. In the following use, the master device will remember the Bluetooth address of the last paired device and search it. The searching won't stop until the device is found. If master device's PIN26 is input high level, the device will lose the memory. In that occasion, it'll search the proper slave device like the first use. Based on this function, the master device can be set to make pair with the specified address or any address by user.

(6) Reset new password introduction

User can set a new password for the HC-06 through AT+PINxxxx command. But the new password will become active after discharged all the energy of the module. If the module still has any energy, the old one is still active. In the test, for discharging all the system energy and activating the new password, we can connect the power supply PIN with GND about 20 seconds after the power is cut off. Generally, shutting down the device for 30 minutes also can discharge the energy, if there is no peripheral circuit helps discharge energy. User should make the proper way according to the specific situation.

(7) Name introduction

If the device has no name, it's better that user doesn't try to change the master device name. The name should be limited in 20 characters.

Summary: The character of HC-06: 1 not many command 2 easy for application 3 low price. It's good for some specific application. HC-04 is very similar with HC-06. Their only one difference is HC-04 is for industry, HC-06 is for civil. Except this, they don't have difference.

The following reference about HC-04 and HC-06 can be downloaded from company website

www.wavesen.com:

HC-06 datasheet .pdf	(the command set introduction is included)
HC-04 datasheet .pdf	(the command set introduction is included)
IVT BlueSoleil-2.6	(IVT Bluetooth drive test version)
Bluetooth FAQ.pdf	
HC-04-D(HD-06-D)datasheet(English).pdf	
HC-06-AT command software (test version)	(some commands in V1.5 is not supported by V1.4)
PCB package of Bluetooth key modules	(PCB package lib in protel)
IVT software manual.pdf	(introduce how to operate the modern and make pair with Bluetooth module)
PDA serial test helper.exe	(serial helper used for WM system)

5 manual for the first use of HC-05

This chapter will introduce how to test and use the HC-05 if it's the first time for user to operate it.

(1) PINs description

PIN1	UART_TXD, Bluetooth serial signal sending PIN, can connect with MCU's RXD PIN
PIN2	UART_RXD, Bluetooth serial signal receiving PIN, can connect with the MCU's TXD PIN, there is no pull-up resistor in this PIN. But It needs to be added an eternal pull-up resistor.
PIN11	RESET, the reset PIN of module, inputting low level can reset the module, when the module is in using, this PIN can connect to air.
PIN12	VCC, voltage supply for logic, the standard voltage is 3.3V, and can work at 3.0-4.2V
PIN13	GND

PIN31	<p>LED1, indicator of work mode. Has 3 modes:</p> <p>When the module is supplied power and PIN34 is input high level, PIN31 output 1Hz square wave to make the LED flicker slowly. It indicates that the module is at the AT mode, and the baud rate is 38400;</p> <p>When the module is supplied power and PIN34 is input low level, PIN31 output 2Hz square wave to make the LED flicker quickly. It indicates the module is at the pairable mode. If PIN34 is input high level, then the module will enter to AT mode, but the output of PIN31 is still 2Hz square wave.</p> <p>After the pairing, PIN31 output 2Hz square wave.</p> <p>Note: if PIN34 keep high level, all the commands in the AT command set can be in application. Otherwise, if just excite PIN34 with high level but not keep, only some command can be used. More information has provided at chapter 2.</p>
PIN32	<p>Output terminal. Before paired, it output low level. Once the pair is finished, it output high level.</p>
PIN34	<p>Mode switch input. If it is input low level, the module is at paired or communication mode. If it's input high level, the module will enter to AT mode. Even though the module is at communication, the module can enter to the AT mode if PIN34 is input high level. Then it will go back to the communication mode if PIN34 is input low level again.</p>

(2) Application circuit 1 (connect to the 3.3V system)

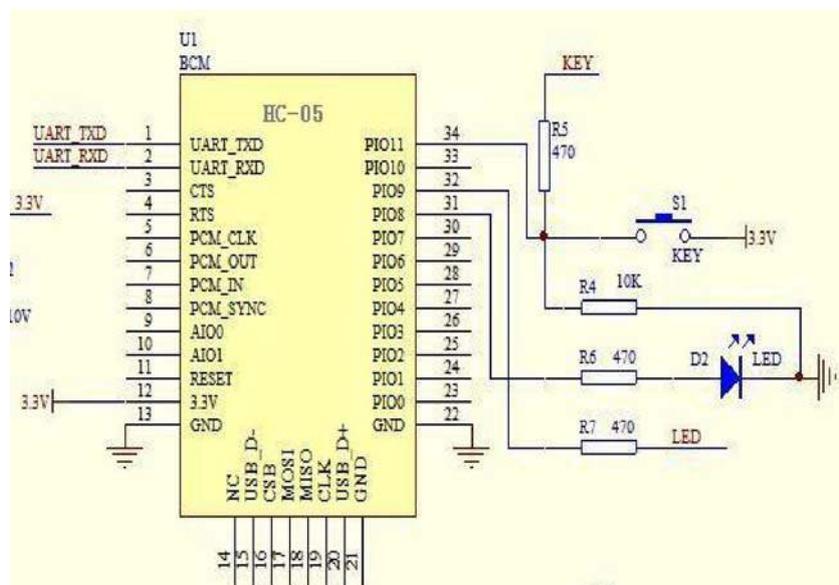


Figure 7 Application 1

(3) Application circuit 2 (connect to 5V serial system or PC serial)

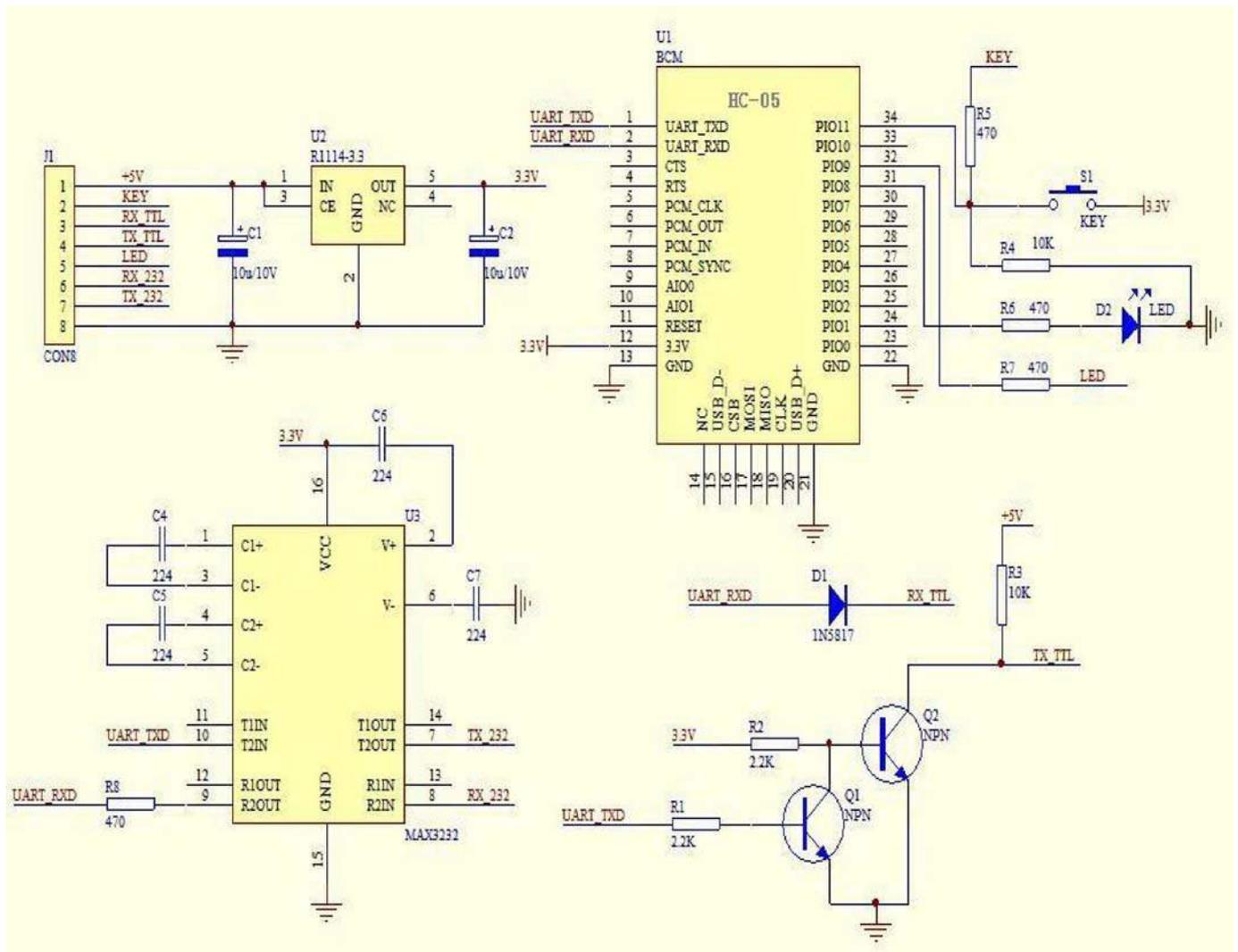


Figure 8 Application circuit 2

(4) AT command test

This chapter introduces some common commands in use. The detail introduction about HC-05 command is in HC-0305 AT command set.

Enter to AT mode:

Way1: Supply power to module and input high level to PIN34 at the same time, the module will enter to AT mode with the baud rate-38400.

Way2: In the first step, supply power to module; In the second step, input high level to PIN34. Then the module will enter to AT mode with the baud rate-9600. Way1 is recommended.

Command structure: all command should end up with “\r\n” (Hex: 0X0D X0A) as the terminator. If

ANEXO B

Ficha técnica comunicación USB-serial

Introducción

Gracias por adquirir el Adaptador USB a Puerto Serie con Retención de Puerto COM de www.StarTech.com.

Ahora usted podrá conectar dispositivos serie, como agendas electrónicas, módems, impresoras o escáneres a un ordenador Mac o PC que tenga un puerto USB disponible. Gracias a la efectiva conexión y la retención del Puerto COM, sus propiedades de configuración nunca se perderán.

Características

- Retiene las propiedades de configuración incluso en el caso de pérdida de energía o desconexión.
- Agrega un Puerto Serie de alta velocidad a un ordenador Mac o PC con puerto USB.
- Posee un buffer de 96 bytes para velocidad de transferencia de carga y descarga
- Transfiere datos a una velocidad de hasta 920 Kbits/seg
- Soporta encendido remoto y control sobre la administración de energía

Antes de Empezar

Requisitos del Sistema

- Sistema Operativo: Windows 98SE, ME, 2000, XP, Mac OS 8.6 o superior
- 1 puerto USB disponible (USB 1.1 o mayor).

Contenido de la Caja

Esta caja debería contener:

- 1 Cable USB a Puerto Serie
- 1 CD con controladores (Drivers)
- 1 Manual de Instrucciones

Instalación

Para asegurar una instalación rápida y fácil, por favor lea la siguiente sección atentamente y complete cada uno de los pasos en el orden enunciado:

La siguiente sección explica cómo instalar el ICUSB232PRO en un ordenador en funcionamiento.

Windows XP. La instalación para otros sistemas operativos de Windows (98SE, ME, 2000 etc.) es similar, sino idéntica.

Instalación para Windows XP.

1. Con su ordenador encendido, inserte el CD con controladores (Drivers), incluido en la caja, en el CD/DVD-ROM.
2. Localice el paquete de instalación de controladores (Drivers), con el nombre **2303 Driver Installer.exe**.
3. Una vez que se inicie el Asistente de Instalación, el mismo indicará que está listo para instalar los controladores (Drivers) del 2303 USB a Puerto Serie. Por favor, haga clic en **Next (Siguiente) para continuar**.
4. Al terminar la instalación, el Asistente de Instalación notificará que los controladores (Drivers) han sido instalados correctamente. Haga clic en **Finish (Finalizar)** para completar la instalación y cerrar el programa de instalación.
5. Localice en su ordenador un Puerto USB disponible. Conecte el Adaptador USB a Puerto Serie.
6. Windows debería detectar el controlador (Driver) como un **Prolific**

USB-to-Serial Comm Port. Antes de instalar el puerto COM, la siguiente ventana puede aparecer: **The device driver has not yet passed Windows XP Logo compatibility** (El controlador que desea instalar aún no ha superado la prueba de logotipo de Windows XP), o una advertencia similar. Haga clic en Continue Anyway (Continuar de todos modos). Windows comenzará a instalar el controlador (Driver) para el Puerto COM del Adaptador USB a Puerto Serie.

7. Cuando Windows haya encontrado el controlador (Driver) correcto, haga clic en **Next (Siguiente)** y luego en **Finish (Finalizar)** para completar la instalación.

Conexión de Periféricos al Adaptador

Para verificar que el software necesario ha sido instalado:

1. Con su ordenador encendido, y el Adaptador ya instalado, conecte el periférico deseado al Adaptador; provea de energía al dispositivo periférico.
2. En caso de que sea la primera vez que se utiliza este periférico, consulte el procedimiento de instalación del mismo y siga las indicaciones de instalación que aparecen en su pantalla. Si usted ya ha instalado este dispositivo periférico en otro de los puertos RC-232 (COM) de su ordenador, el mismo debería encontrar e instalar automáticamente los controladores (Drivers) pertinentes

NOTA: Si usted está conectando un modem bajo **Windows 2000**, debe especificarlo como un **Standard Modem (Modem Estándar)**.

Especificaciones

Tipo de Bus	USB 1.1
Conectores	USB Tipo A Macho RS232 Macho DB9
Velocidad Máxima de Transferencia de Datos	920 Kbits/sec
Chipset	Prolific PL2303
Sistemas Operativos Soportados	Windows 98/ME/2000/XP Mac OS 8.6 o superior
Certificaciones (Regulatorias u otras)	FCC, CE, RoHS

ANEXO C

Medidas de seguridad con Oculus Rift DK2

* These health & safety warnings are periodically updated for accuracy and completeness. Check [oculus.com/warnings](https://www.oculus.com/warnings) for the latest version.



HEALTH & SAFETY WARNINGS: Please ensure that all users of the headset read the warnings below carefully before using the headset to reduce the risk of personal injury, discomfort or property damage.



Before Using the Headset:

- Read and follow all setup and operating instructions provided with the headset.
- The headset should be configured for each individual user by using the configuration software before starting a virtual reality experience. Failure to follow this instruction may increase the risk of discomfort.
- We recommend seeing a doctor before using the headset if you are pregnant, elderly, have pre-existing binocular vision abnormalities or psychiatric disorders, or suffer from a heart condition or other serious medical condition.



Seizures: Some people (about 1 in 4000) may have severe dizziness, seizures, epileptic seizures or blackouts triggered by light flashes or patterns, and this may occur while they are watching TV, playing video games or experiencing virtual reality, even if they have never had a seizure or blackout before or have no history of seizures or epilepsy. Such seizures are more common in children and young people under the age of 20. Anyone who has had a seizure, loss of awareness, or other symptom linked to an epileptic condition should see a doctor before using the headset.



Children: This product should not be used by children under the age of 13. Adults should monitor children (age 13 and older) who are using or have used the Headset for any of the symptoms described below, and should limit the time children spend using the Headset and ensure they take breaks during use. Prolonged use should be avoided, as this could negatively impact hand-eye coordination, balance, and multi-tasking ability. Adults should monitor children closely during and after use of the headset for any decrease in these abilities.



General Instructions & Precautions: You should always follow these instructions and observe these precautions while using the headset to reduce the risk of injury or discomfort:

- **Use Only In A Safe Environment:** The headset produces an immersive virtual reality experience that distracts you from and completely blocks your view of your actual surroundings. **Always be aware of your surroundings when using the headset and remain seated at all times.** Take special care to ensure that you are not near other people, objects, stairs, balconies, windows, furniture, or other items that you can bump into or knock down when using—or immediately after using—the headset. Do not handle sharp or otherwise dangerous objects while using the headset. Never wear the headset in situations that require attention, such as walking, bicycling, or driving.
- Make sure the headset is level and secured comfortably on your head, and that you see a single, clear image.
- Ease into the use of the headset to allow your body to adjust; use for only a few minutes at a time at first, and only increase the amount of time using the headset gradually as you grow accustomed to virtual reality. Looking around when first entering virtual reality can help you adjust to any small differences between your real-world movements and the resulting virtual reality experience.
- A comfortable virtual reality experience requires an unimpaired sense of motion and balance. Do not use the headset when you are tired, need sleep, are under the influence of alcohol or drugs, are hung-over, have digestive problems, are under emotional stress or anxiety, or when suffering from cold, flu, headaches, migraines, or earaches, as this can increase your susceptibility to adverse symptoms.
- Do not use the headset while in a moving vehicle such as a car, bus, or train, as this can increase your susceptibility to adverse symptoms.
- Take at least a 10 to 15 minute break every 30 minutes, even if you don't think you need it. Each person is different, so take more frequent

and longer breaks if you feel discomfort. You should decide what works best.

- The headset may be equipped with a “passthrough” feature which permits you to temporarily see your surroundings for brief real world interaction. You should always remove the headset for any situation that requires attention or coordination. Do not use this feature for more than a few minutes at a time.
- Listening to sound at high volumes can cause irreparable damage to your hearing. Background noise, as well as continued exposure to high volume levels, can make sounds seem quieter than they actually are. Due to the immersive nature of the virtual reality experience, do not use the headset with the sound at a high volume so that you can maintain awareness of your surroundings and reduce the risk of hearing damage.

 **WARNING** Discomfort

- **Immediately discontinue use if anyone using the headset experiences any of the following symptoms: seizures; loss of awareness; eye strain; eye or muscle twitching; involuntary movements; altered, blurred, or double vision or other visual abnormalities; dizziness; disorientation; impaired balance; impaired hand-eye coordination; excessive sweating; increased salivation; nausea; lightheadedness; discomfort or pain in the head or eyes; drowsiness; fatigue; or any symptoms similar to motion sickness.**
- **Just as with the symptoms people can experience after they disembark a cruise ship, symptoms of virtual reality exposure can persist and become more apparent hours after use. These post-use symptoms can include the symptoms above, as well as excessive drowsiness and decreased ability to multi-task. These symptoms may put you at an increased risk of injury when engaging in normal activities in the real world.**

- Do not drive, operate machinery, or engage in other visually or physically demanding activities that have potentially serious consequences (i.e., activities in which experiencing any symptoms could lead to death, personal injury, or damage to property), or other activities that require unimpaired balance and hand-eye coordination (such as playing sports or riding a bicycle, etc.) until you have fully recovered from any symptoms.
- Do not use the headset until all symptoms have completely subsided for several hours. Make sure you have properly configured the headset before resuming use.
- Be mindful of the type of content that you were using prior to the onset of any symptoms because you may be more prone to symptoms based upon the content being used.
- See a doctor if you have serious and/or persistent symptoms.

 **WARNING** **Repetitive Stress Injury:** Playing video games can make your muscles, joints or skin hurt. If any part of your body becomes tired or sore while playing, or if you feel symptoms such as tingling, numbness, burning or stiffness, stop and rest for several hours before playing again. If you continue to have any of the above symptoms or other discomfort during or after play, stop playing and see a doctor.

 **WARNING** **Radio Frequency Interference:** The headset can emit radio waves that can affect the operation of nearby electronics, including cardiac pacemakers. If you have a pacemaker or other implanted medical device, do not use the headset without first consulting your doctor or the manufacturer of your medical device.

⚠ WARNING **Electrical Shock**: To reduce risk of electric shock:

- Do not modify or disassemble any of the components provided.
- Do not use the product if any cable is damaged or any wires are exposed.

If a power adapter is provided:

- Do not expose the power adapter to water or moisture.
- Unplug the power adapter before cleaning, and clean only with a dry cloth.
- Keep the power adapter away from open flames and other heat sources.
- Use only the power adapter provided with the headset.

NOTICE **Sunlight Damage**: Do not leave the headset in direct sunlight. Exposure to direct sunlight can damage the headset.

ANEXO D

Planos de piezas mecánicas



SEW
EURODRIVE

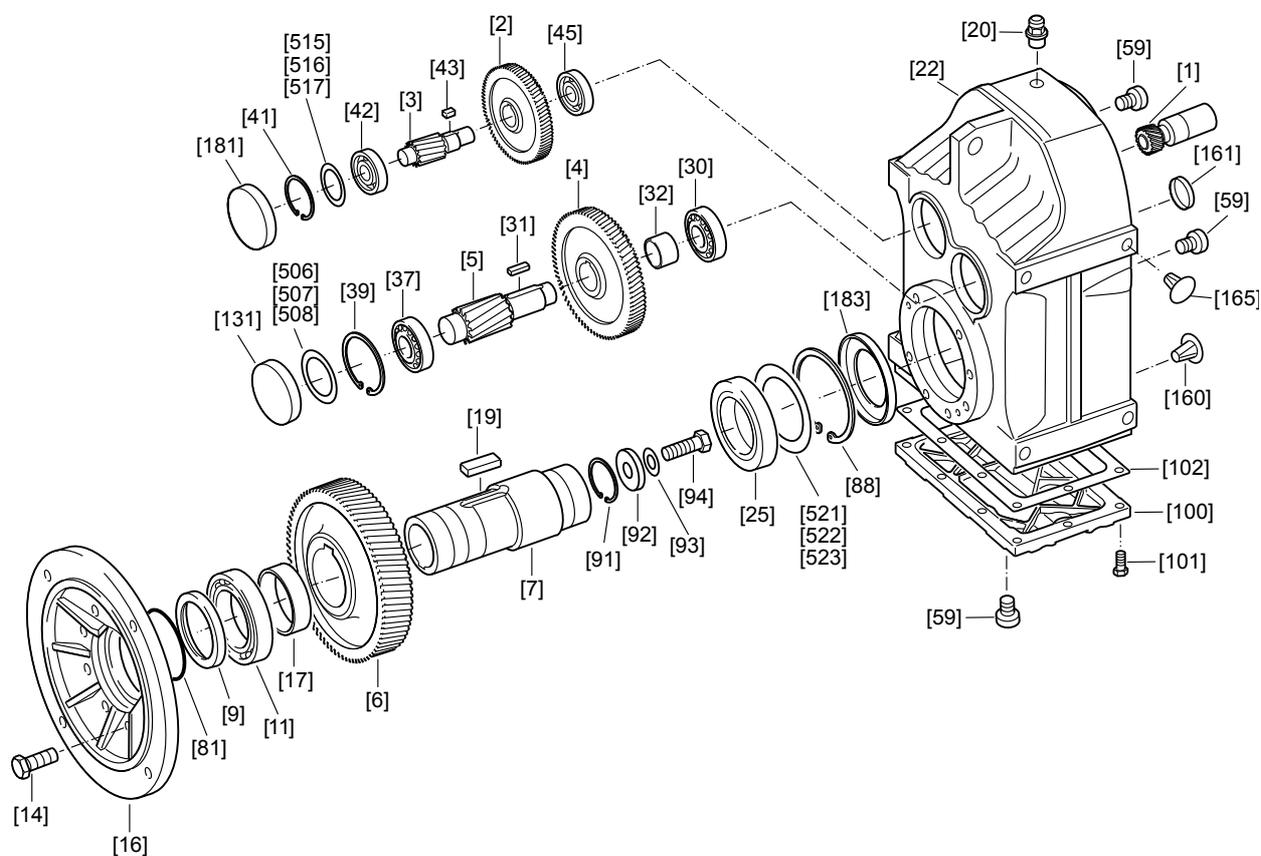
Instrucciones de montaje y funcionamiento



Reductores
Series R..7, F..7, K..7, K..9, S..7, SPIROPLAN® W



3.2 Estructura general de los reductores cilíndricos de ejes paralelos



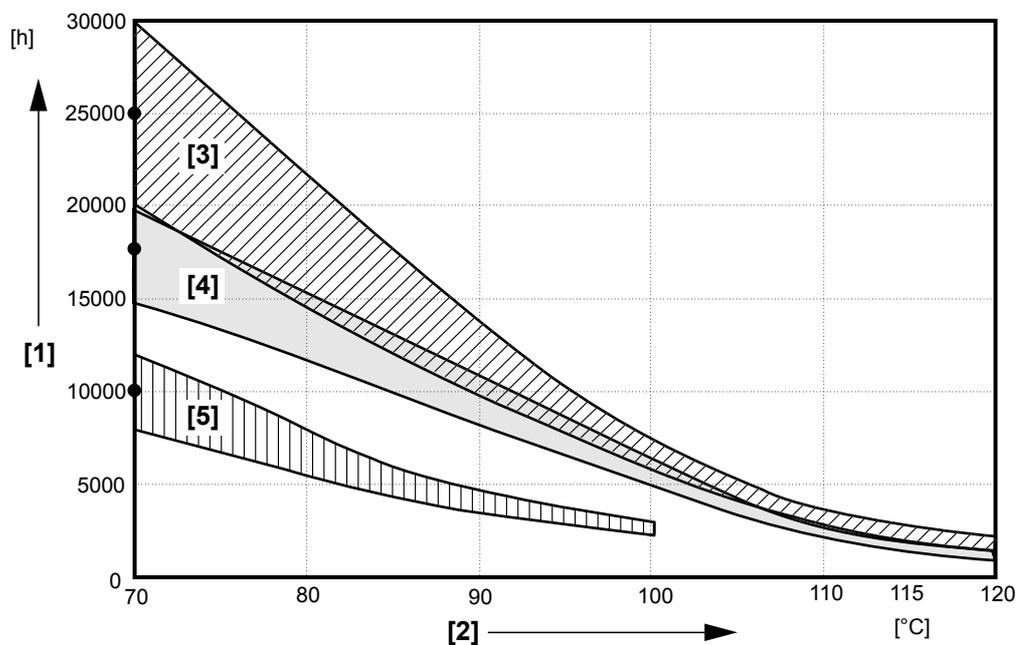
19298059

[1] Piñón	[22] Carcasa	[91] Circlip	[506] Arandela de ajuste
[2] Rueda	[25] Rodamientos	[92] Arandela	[507] Arandela de ajuste
[3] Árbol piñón	[30] Rodamientos	[93] Arandela de bloqueo	[508] Arandela de ajuste
[4] Rueda	[31] Chaveta	[94] Tornillo hexagonal	[515] Arandela de ajuste
[5] Árbol piñón	[32] Distanciadore	[100] Tapa del reductor	[516] Arandela de ajuste
[6] Rueda	[37] Rodamientos	[101] Tornillo hexagonal	[517] Arandela de ajuste
[7] Eje hueco	[39] Circlip	[102] Junta	[521] Arandela de ajuste
[9] Retén	[41] Circlip	[160] Tapón de cierre	[522] Arandela de ajuste
[11] Rodamientos	[42] Rodamientos	[161] Capuchón	[523] Arandela de ajuste
[14] Tornillo hexagonal	[43] Chaveta	[165] Tapón de cierre	
[16] Brida de salida	[45] Rodamientos	[181] Capuchón	
[17] Distanciadore	[59] Tapón roscado	[183] Retén	
[19] Chaveta	[81] Junta de estanqueidad		
[20] Tapón de salida de gases	[88] Circlip		



6.3 Intervalos de cambio de lubricantes

La siguiente imagen muestra los intervalos de cambio de los reductores estándar en condiciones normales. Cambiar el aceite con mayor frecuencia cuando se utilicen versiones especiales sometidas a condiciones ambientales más duras.



[1] Horas de servicio

[2] Temperatura constante del baño de aceite

- Valor medio según el tipo de aceite a 70 °C

[3] CLP PG

[4] CLP HC / HCE

[5] CLP / HLP / E



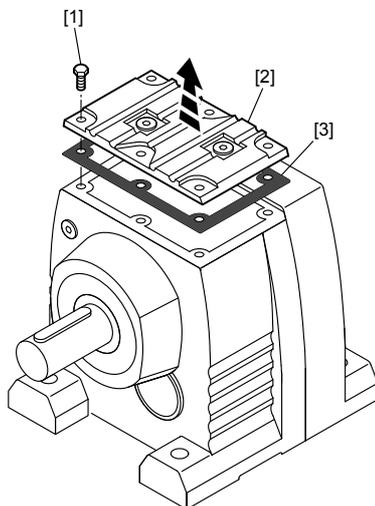


6.6.3 B: Reductores de engranajes cilíndricos, de ejes paralelos y SPIROPLAN® sin tapón de nivel de aceite y con tapa de montaje

Comprobación del nivel de aceite por la tapa de montaje

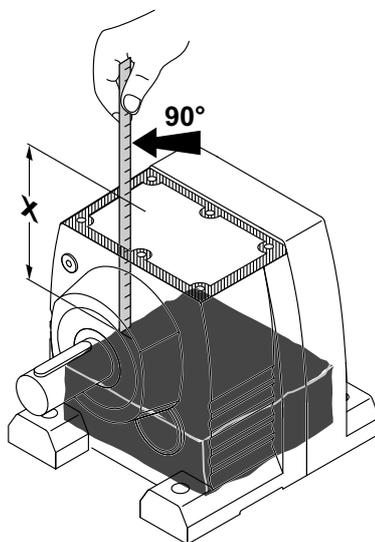
En los reductores sin orificio de nivel de aceite, el nivel de aceite se comprueba en la abertura de la tapa de montaje. Proceda como se indica a continuación:

1. Observe las indicaciones del capítulo "Trabajos previos para la inspección y el mantenimiento del reductor" (→ pág. 85).
2. Coloque el reductor en la siguiente posición de montaje para que la tapa mire hacia arriba:
 - R07 – R57 en posición de montaje M1
 - F27 en posición de montaje M3
 - W10 – W30 en posición de montaje M1
3. Suelte los tornillos [1] de la tapa de montaje [2] y retire la tapa de montaje [2] con la junta correspondiente [3] (véase la siguiente imagen).



18643211

4. Determine la distancia vertical "x" entre el nivel de aceite y la superficie de estanqueidad de la carcasa (véase la siguiente imagen).



18646283



5. Compare la distancia obtenida "x" con la distancia máxima entre el nivel de aceite y la superficie de estanqueidad de la carcasa indicada en la tabla según la posición de montaje. Dado el caso, corrija la altura de llenado.

Modelo de reductor		Distancia máxima x [mm] entre el nivel de aceite y la superficie de estanqueidad de la carcasa para posición de montaje					
		M1	M2	M3	M4	M5	M6
R07	de 2 etapas	52 ± 1	27 ± 1	27 ± 1	27 ± 1	27 ± 1	27 ± 1
	de 3 etapas	49 ± 1	21 ± 1	21 ± 1	21 ± 1	21 ± 1	21 ± 1
R17	de 2 etapas	63 ± 1	18 ± 1	46 ± 1	18 ± 1	46 ± 1	46 ± 1
	de 3 etapas	58 ± 1	11 ± 2	40 ± 2	11 ± 2	40 ± 2	40 ± 2
R27	de 2 etapas	74 ± 1	22 ± 1	45 ± 1	22 ± 1	45 ± 1	45 ± 1
	de 3 etapas	76 ± 1	19 ± 1	42 ± 1	19 ± 1	42 ± 1	42 ± 1
R47	de 2 etapas	–	–	–	–	39 ± 1	–
	de 3 etapas	–	–	–	–	32 ± 1	–
R57	de 2 etapas	–	–	–	–	32 ± 1	–
	de 3 etapas	–	–	–	–	28 ± 1	–
independiente de la posición de montaje							
F27	de 2 etapas	78 ± 1	31 ± 1	72 ± 1	56 ± 1	78 ± 1	78 ± 1
	de 3 etapas	71 ± 1	24 ± 1	70 ± 1	45 ± 1	71 ± 1	71 ± 1
W10							12 ± 1
W20							19 ± 1
W30							31 ± 1

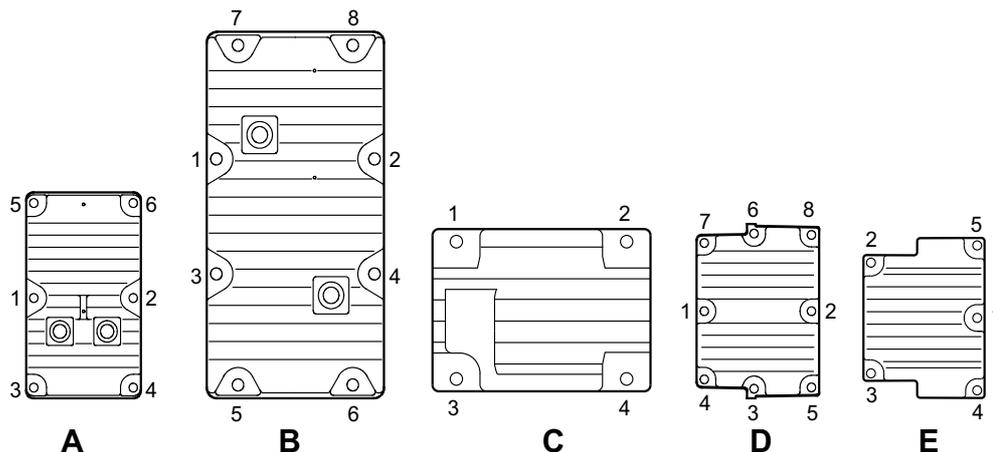


Inspección y mantenimiento

Inspección y mantenimiento del reductor

6. Selle el reductor tras comprobar el nivel de aceite:

- Vuelva a colocar la junta de la tapa de montaje. Asegúrese de que las superficies de estanqueidad están limpias y secas.
- Monte la tapa de montaje. Apriete los tornillos de la tapa desde dentro hacia fuera en el orden indicado en la imagen con los pares de apriete de la siguiente tabla. Repita los pasos de apriete hasta que los tornillos queden fijos. Para evitar daños en la tapa de montaje, sólo pueden utilizarse destornilladores de impulsos o llaves dinamométricas (no destornilladores de percusión).



18649739

Modelo de reductor	Imagen	Rosca de fijación	Par de apriete nominal T_N [Nm]	Par de apriete mínimo T_N [Nm]
R/RF07	E	M5	6	4
R/RF17/27	D	M6	11	7
R/RF47/57	A			
F27	B			
W10	C	M5	6	4
W20	C	M6	11	7
W30	A			

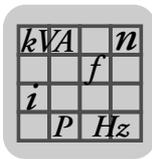


Tabla de lubricantes

01 751 09 04

	6)	DIN (ISO)	ISO, NLGI	Mobil®	Shell	BP	TEKNO	Castrol	FUCHS	TOTAL
R... K..37-187 (HK...) F...	Standard -15	CLP (CC)	VG 220	Mobilgear 600 XP 220	Shell Omala S2 G 220	BP Energol GR-XP 220	Meropra 220	Tribol 1100/220	Renolin CLP 220	Carter EP 220
	+80	CLP PG	VG 220	Mobil Glygoyle 220	Shell Omala S4 WE 220	BP Enersyn SG-XP 220	Synlube CLP 220	Optiflex A 220	Renolin PG 220	Carter SY 220
	+60	CLP HC	VG 220	Mobil SHC 630	Shell Omala S4 GX 220		Pinnacle EP 220	Optiflex Synthetic X 220	Renolin Unisyn CLP 220	Carter SH 220
	+40	CLP HC	VG 150	Mobil SHC 629	Shell Omala S4 GX 150		Pinnacle EP 150	Optiflex Synthetic X 150	Renolin Unisyn CLP 150	Carter SH 150
K..19 K..29	+25	CLP (CC)	VG 150	Mobilgear 600 XP 150	Shell Omala S2 G 150	BP Energol GR-XP 150	Meropra 150	Tribol 1100/150	Renolin CLP 150	Carter EP 150
	+20	CLP HC	VG 68	Mobil SHC 626	Shell Omala S4 GX 68		Cetus PAO 46	Optiflex BM 100	Renolin Unisyn CLP 68	
	+0	CLP HC	VG 32	Mobil SHC 624		Klubersynth HySyn FG-32		Optiflex HY 32	Renolin Unisyn DL 32	Daenis SH 32
	Standard	CLP PG	VG 460			Klubersynth GH 6-460				
S... (HS...)	+60	H1 PG	VG 460			Klubersynth UH1 6-460				
	Standard	CLP (CC)	VG 680	Mobilgear 600 XP 680	Shell Omala S2 G 680	BP Energol GR-XP 680	Meropra 680	Tribol 1100/680	Renolin SEW 680	Carter EP 680
	+60	CLP PG	VG 680	Mobil glygoyle 680	Shell Omala S4 WE 680	BP Enersyn SG-XP 680	Synlube CLP 680	Optiflex A 680	Renolin PG 680	
	+60	CLP HC	VG 460	Mobil SHC 634	Shell Omala S4 GX 460		Pinnacle EP 460	Optiflex Synthetic X 460	Renolin Unisyn CLP 460	Carter SH 460
R... K..37-187 / HK.. F.. S... / HS..	+30	CLP HC	VG 150	Mobil SHC 629	Shell Omala S4 GX 150		Pinnacle EP 150	Optiflex Synthetic X 150	Renolin Unisyn CLP 150	Carter SH 150
	+10	CLP (CC)	VG 150	Mobilgear 600 XP 150	Shell Omala S2 G 150	BP Energol GR-XP 150	Meropra 150	Optiflex BM 150	Renolin CLP 150	Carter EP 150
	+40	CLP PG	VG 220	Mobil Glygoyle 220	Shell Omala S4 WE 220	BP Enersyn SG-XP 220	Synlube CLP 220	Optiflex A 220	Renolin PG 220	Carter SY 220
	+20	CLP HC	VG 68	Mobil SHC 626	Shell Omala S4 GX 68		Cetus PAO 46	Optiflex A 220	Renolin Unisyn CLP 68	
W... (HW...)	+40	CLP HC	VG 32	Mobil SHC 624		Klubersynth HySyn FG-32		Alphasyn T32	Renolin Unisyn DL 32	Daenis SH 32
	Standard	CLP HC	VG 460			Klubersynth UH1 6-460		Optiflex GT 460	Cassida Fluid GL 460	
	+40	CLPHC NSF H1	VG 460			Klubersynth UH1 6-460		Optiflex GT 220	Cassida Fluid GL 220	
	+30	E	VG 68			Klubersynth UH1 6-460		Optiflex HY 68	Cassida Fluid HF 68	
P.S.F..	+40	SEW	VG 460			Klubersynth UH1 6-460		Optiflex HY 68	Plantogear 460 S	
	Standard	PG	VG 460							
	+10	API GL5	SAE 75W90 (-VG 100)	Mobil Synth Oil 75W90						
	+60	H1 PG	VG 460	Mobil SHC 624						
P.S.C..	+80	CLP PG	VG 220							
	+60	H1 PG	VG 460							
	Standard	CLP HC	VG 32							
	+40	CLP (CC)	VG 220	Mobilgear 600 XP 220						
B.S.F..	+40	DIN 51 818	NLGI 00	Mobilux EP 004						
	+40	DIN 51 818	NLGI 1							
	Standard	CLP HC	VG 32	Mobil SHC 624						
	+60	CLP PG	VG 220							

36028799863966091

cylinders

ISO/VDMA Series

ISO 6431 and VDMA 24562 Cylinder Line



NUMATICS



ISO/VDMA Series Metric Standard Interchangeable

NUMATICS®

How to Order

Z G 050 / 0080 C08 A1

Cylinder Series

Z = Tie-Rod Version
W = Profile Version

Cylinder Function Cylinder Type

A =  Heat Resistant Type Up to +180°C (Viton® - Seals)
Single Rod End w/o Magnet

B =  Heat Resistant Type Up to +180°C (Viton® - Seals)
Double Rod End w/o Magnet

G =  Single Rod End with Magnet

H =  Double Rod End with Magnet

Piston Diameter

032 = 32 mm (1.26)
040 = 40 mm (1.57)
050 = 50 mm (1.97)
063 = 63 mm (2.48)
080 = 80 mm (3.15)
100 = 100 mm (3.94)
125* = 125 mm (4.92)
160* = 160 mm (6.30)
200* = 200 mm (7.78)
*Only for tie-rod version.

Stroke*

0025 = 25 mm (0.98)
0050 = 50 mm (1.97)
0080 = 80 mm (3.15)
0100 = 100 mm (3.94)
0125 = 125 mm (4.92)
0160 = 160 mm (6.30)
0200 = 200 mm (7.78)
0250 = 250 mm (9.84)
0320 = 320 mm (12.60)
0400 = 400 mm (15.75)
0500 = 500 mm (19.69)
* Further strokes on request.

Options*

00 = Stainless Steel Rod and Jam Nut
A1 = Hard Piston Rod, Chrome Plated (Standard)
1A = Rod Extension (must specify length)
2A = Thread Extension (must specify length)
1R = Special Rod End
* More Options on Request

Cylinder Including Mounting Parts

C01 = Foot Bracket (MS1)
CF2 = Flange to VDMA 24562 T.2 (MF1, MF2), Front Side
CR2 = Flange to VDMA 24562 T.2 (MF1, MF2), Back Side
C03 = Foot Bracket (Plain)
CF4 = Rod Clevis, Front Side
CD4 = Rod Clevis, Both Sides (Only Type H)
CF5 = Oscillating Clevis, Front Side
CD5 = Oscillating Clevis, Both Sides (Only Type H)
C07 = Oscillating Brackets to VDMA 24562 T.2 (MP4), with Lugs
C08 = Oscillating Brackets to VDMA 24562 T.2 (MP2), Fork Type
C09 = Pivot to VDMA 24562 T.2 (MT4), Only for Cylinders of Tie-Rod Version
CV9 = Pivot to VDMA 24562 T.2 (MT4) Swivelled, Only for Tie-Rod Version
C10 = Pivot Fastening, End Cap Mounting Front Side
C13 = Oscillating Joint Bracket (Spherical)
C14 = Oscillating Brackets to VDMA 24562 T.2, Fork Type, Narrow Clevis
BOV = Piston rod extended for mounting of RL Series Rod Locking Unit
BMV = RL Series Rod Locking Unit assembled to the cylinder
BOI = *Nu*
BMI = *Nu*

Order example: ZG050/0080C08A1

Here we have a double acting tie rod version cylinder with single rod end. Piston diameter is 50 mm and stroke 80 mm. The cylinder is equipped with a magnet for switch sensing and has cushioning. The cylinder will be mounted with oscillating brackets, fork type. Optionally it has a hard chrome plated piston rod.



Technical Information

Theoretical Force Table for Double Acting Cylinders with Piston Diameters 32 mm (1.26) to 200 mm (7.78)

PISTON DIAMETER mm (inches)	PISTON ROD DIAMETER mm (inches)	EFFECTIVE PISTON SURFACE mm ² (cm ²)	PRESSURE (bar)									
			2	3	4	5	6	7	8	9	10	
32 (1.26)	12.0 (0.47)	for extend	8.0 (0.12)	141	212	282	353	424	494	565	636	706
		for retract	6.9 (0.11)	122	182	243	304	366	427	488	549	610
40 (1.57)	16.0 (0.63)	for extend	12.6 (0.20)	223	334	445	555	667	780	893	1001	1109
		for retract	10.6 (0.16)	187	281	375	468	561	655	748	843	936
50 (1.97)	16.0 (0.63)	for extend	19.6 (0.30)	346	520	692	865	1040	1207	1383	1560	1727
		for retract	17.6 (0.27)	310	464	618	772	926	1080	1234	1338	1542
63 (2.48)	20.0 (0.79)	for extend	31.2 (0.48)	551	827	1099	1373	1648	1933	2207	2482	2757
		for retract	28.1 (0.44)	495	746	991	1236	1491	1736	1982	2237	2482
80 (3.15)	25.0 (0.98)	for extend	50.3 (0.78)	889	1334	1776	2217	2668	3110	3551	4002	4444
		for retract	45.3 (0.70)	800	1197	1599	2001	2403	2806	3198	3600	4002
100 (3.94)	25.0 (0.98)	for extend	78.5 (1.22)	1383	2080	2776	3463	4159	4856	5543	6239	6926
		for retract	73.6 (1.14)	1295	1952	2600	3247	3895	4552	5199	5847	6794
125 (4.92)	30.0 (1.18)	for extend	122.7 (1.90)	2168	3247	4336	5415	6497	7583	8662	9751	10830
		for retract	115.7 (1.79)	2036	3054	4072	5090	6108	7126	8144	9162	10180
160 (6.30)	40.0 (1.57)	for extend	201.1 (3.12)	3551	5327	7102	8878	10654	12429	14205	15980	17756
		for retract	188.5 (2.92)	3326	4993	6651	8319	9987	11644	13312	14970	16638
200 (7.87)	40.0 (1.57)	for extend	314.2 (4.87)	5563	8319	11095	13871	16648	19424	22190	24966	27743
		for retract	301.6 (4.68)	5327	7985	10654	13312	15971	18639	21297	23966	26624

Friction losses are considered with 10%

Table on air consumption for double acting cylinders with piston diameter 1.26 (32mm) to 7.87 (200mm)

PISTON DIAMETER mm (inches)	PRESSURE (bar)									
	2	3	4	5	6	7	8	9	10	
	AIR CONSUMPTION (l) PER 100 MM STROKE (UNCOMPRESSED AIR)									
32 (1.26)	0.47	0.65	0.80	0.96	1.13	1.25	1.43	1.60	1.95	
40 (1.57)	0.75	1.02	1.25	1.49	1.70	2.10	2.25	2.50	2.69	
50 (1.97)	1.20	1.58	2.02	2.40	2.80	3.12	3.52	3.96	4.31	
63 (2.48)	1.94	2.53	3.16	3.78	4.39	5.08	5.60	6.18	6.86	
80 (3.15)	3.03	4.07	5.03	6.07	7.02	8.01	9.02	9.92	10.95	
100 (3.94)	4.76	6.33	7.93	9.50	11.05	12.68	14.25	15.81	17.59	
125 (4.92)	7.90	10.41	12.92	15.73	17.91	20.62	23.20	25.70	29.50	
160 (6.30)	12.51	16.60	20.85	25.80	28.90	34.39	38.62	41.95	46.72	
200 (7.87)	19.82	26.31	33.82	39.95	46.81	52.90	59.30	66.90	72.90	

Value for 1 double stroke

Weights (kg)

CYLINDER TYPE WA; OR WG, PISTON DIAMETER	32	40	50	63	80	100				
0 mm stroke	0.57	0.82	1.32	1.70	2.79	3.86				
to be added per 3.94 (100mm) stroke	0.28	0.42	0.55	0.59	0.94	1.12				
CYLINDER TYPE WB; OR WH, PISTON DIAMETER	32	40	50	63	80	100				
0 mm stroke	0.65	0.93	1.57	1.98	1.92	4.34				
to be added per 3.94 (100mm) stroke	0.36	0.57	0.79	0.83	1.31	1.49				
CYLINDER TYPE ZA; OR ZG, PISTON DIAMETER	32	40	50	63	80	100	125	160	200	250*
0 mm stroke	0.55	0.76	1.29	1.67	2.72	3.70	5.37	10.75	14.75	31.45
to be added per 3.94 (100mm) stroke	0.23	0.32	0.51	0.54	0.81	0.87	1.39	1.80	2.20	8.20
* not for cylinder type ZG										
CYLINDER TYPE ZB; OR ZH, PISTON DIAMETER mm (inches)	32 (1.26)	40 (1.57)	50 (1.97)	63 (2.48)	80 (3.15)	100 (3.94)	125 (4.92)	160 (6.30)	200 (7.78)	250* (9.87)
0 mm stroke	0.68	0.88	1.54	1.95	3.15	4.18	6.27	12.49	17.14	36.55
to be added per 3.94 (100mm) stroke	0.31	0.47	0.75	0.78	1.18	1.24	2.02	2.60	3.18	10.70
* not for cylinder type ZH										

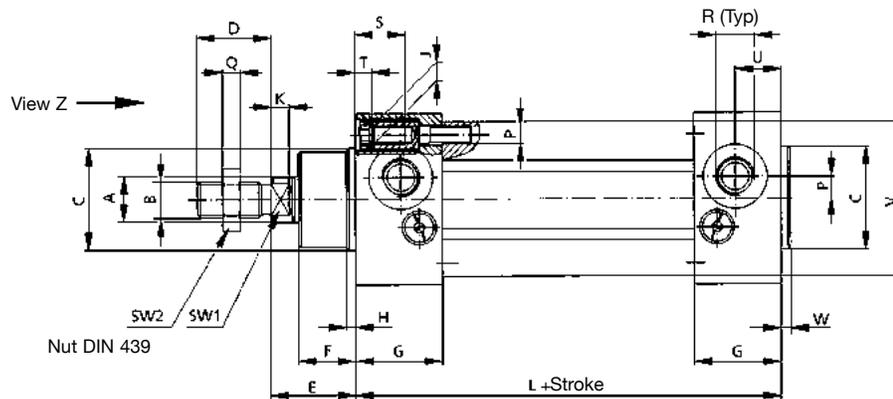


ISO/VDMA Series Metric Standard Interchangeable

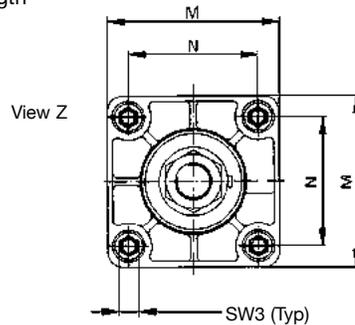
NUMATICS®

Profile Version

Double Acting Cylinder, Single Rod End, Types WA or WG



X= Internal Cushioning length



Repair Kits

CYLINDER TYPE	ORDER CODE
WA032/...	VA032/RK
WA100/...	VA100/RK
WG032/...	VG032/RK
WG100/...	VG100/RK

Dimensions (mm)

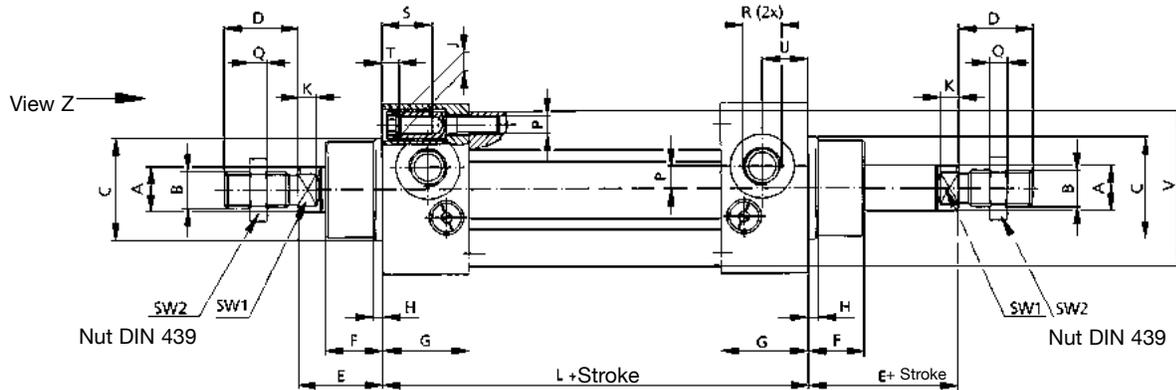
TYPE	PISTON DIAMETER	A Ø	B	C Ø	D±0.3	E±0.9	F±0.1	G	H	J	K	L	M
WA; WG	32	12.0	M10 x 1.25	30.0	22.0	26.0	16.0	33.0	4.0	M6	6.0	94 ± 0.4	47.0
WA; WG	40	16.0	M12 x 1.25	35.0	24.0	30.0	20.0	33.5	4.0	M6	6.5	105 ± 0.7	52.0
WA; WG	50	20.0	M16 x 1.5	40.0	32.0	37.0	25.0	36.0	4.0	M8	8.0	106 ± 0.7	65.0
WA; WG	63	20.0	M16 x 1.5	45.0	32.0	37.0	25.0	38.0	4.0	M8	8.0	121 ± 0.8	75.0
WA; WG	80	25.0	M20 x 1.5	45.0	40.0	46.0	30.0	41.5	4.0	M10	10.0	128 ± 0.8	95.0
WA; WG	100	25.0	M20 x 1.5	55.0	40.0	51.0	35.0	41.5	4.0	M10	10.0	138 ± 1	115.0

TYPE	PISTON DIAMETER	N	P	Q	R	S min.	T	U	V	W -0.1	X	SW1	SW2	SW3
WA; WG	32	32.5 ± 0.5	5.0	5.0	G 1/8	16.0	5.0	14.0	42.5	4.0	15.0	10	17	6
WA; WG	40	38.0 ± 0.5	6.0	6.0	G 1/4	16.0	5.0	15.0	50.5	4.0	19.0	13	19	6
WA; WG	50	46.5 ± 0.6	6.0	8.0	G 1/4	16.0	6.0	15.0	60.0	4.0	22.0	16	24	8
WA; WG	63	56.5 ± 0.7	9.5	8.0	G 3/8	16.0	6.0	20.0	70.0	4.0	30.0	16	24	8
WA; WG	80	72.0 ± 0.7	9.5	10.0	G 3/8	16.0	7.5	20.0	88.0	4.0	30.0	21	30	10
WA; WG	100	89.0 ± 0.7	12.0	10.0	G 1/2	16.0	7.5	25.0	106.0	4.0	27.0	21	30	10

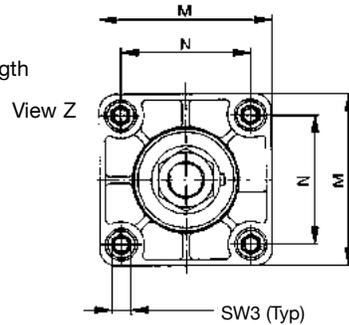


Profile Version

Double Acting Cylinder with/without Sensor, Double Rod End, Types WB or WH



X= Internal Cushioning length



Repair Kits

CYLINDER TYPE	ORDER CODE
WB032/...	VB032/RK
WB100/...	VB100/RK
WH032/...	VH032/RK
WH100/...	VH100/RK

Dimensions (mm)

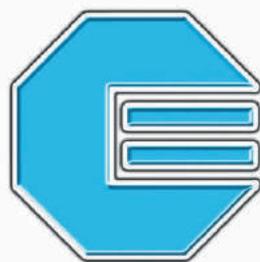
TYPE	PISTON DIAMETER	A Ø	B	C Ø	D ±0.3	E ±0.9	F ±0.1	G	H	J	K	L
WB; WH	32	12.0	M10 x 1.25	30.0	22.0	26.0	16.0	33.0	4.0	M6	6.0	94 ± 0.4
WB; WH	40	16.0	M12 x 1.25	35.0	24.0	30.0	20.0	33.5	4.0	M6	6.5	105 ± 0.7
WB; WH	50	20.0	M16 x 1.5	40.0	32.0	37.0	25.0	36.0	4.0	M8	8.0	106 ± 0.7
WB; WH	63	20.0	M16 x 1.5	45.0	32.0	37.0	25.0	38.0	4.0	M8	8.0	121 ± 0.8
WB; WH	80	25.0	M20 x 1.5	45.0	40.0	46.0	30.0	41.5	4.0	M10	10.0	128 ± 0.8
WB; WH	100	25.0	M20 x 1.5	55.0	40.0	51.0	35.0	41.5	4.0	M10	10.0	138 ± 1

TYPE	PISTON DIAMETER	M	N	P	Q	R	S min.	T	U	V	X	SW1	SW2	SW3
WB; WH	32	47.0	32.5 ± 0.5	5.0	5.0	G 1/8	16.0	5.0	14.0	42.5	15.0	10	17	6
WB; WH	40	52.0	38.0 ± 0.5	6.0	6.0	G 1/4	16.0	5.0	15.0	50.5	19.0	13	19	6
WB; WH	50	65.0	46.5 ± 0.6	6.0	8.0	G 1/4	16.0	6.0	15.0	60.0	22.0	16	24	8
WB; WH	63	75.0	56.5 ± 0.7	9.5	8.0	G 3/8	16.0	6.0	20.0	70.0	30.0	16	24	8
WB; WH	80	95.0	72.0 ± 0.7	9.5	10.0	G 3/8	16.0	7.5	20.0	88.0	30.0	21	30	10
WB; WH	100	115.0	89.0 ± 0.7	12.0	10.0	G 1/2	16.0	7.5	25.0	106.0	27.0	21	30	10



Catálogo de Extrusiones de Aluminio

PUERTAS, CURTAIN WALL, VIDRIERAS Y BARANDAS
Pintura Electrostatica



RIF.: J-07520772-6

EXTRUDAL

EXTRUSIÓN DE ALUMINIO, C.A.

Web: www.extrudal.com

ALEACIONES Y TEMPLES

Aleaciones y sus Temple en el proceso de extrusión.

Propiedades Mecánicas (Valores Típicos)

Aleación Temple	Densidad gr/cm ³	Carga de Rotura Kg/mm ²	Límite Elástico Kg/mm ²	Alargamiento % en muestra de 50 mm.	Conductividad % IACS	Módulo de Elasticidad Kg/mm ² x 10 ³
1100 - F	2,71	8,0	4,0	40	59	7,0
1350 - H111	2,71	6,5	3,0	25	62	7,0
6061 - T4	2,71	23,0	13,5	22	40	7,0
T6		29,5	26,0	12	43	
6063 - T4	2,71	17,5	9,0	22	55	7,0
T5		19,0	15,0	12		
T6		24,5	22,0	12		
6005 - T5	2,71	29,0	26,0	10	49	7,0
6101 - T6	2,71	22,5	19,5	14	56	7,0
6463 - T5	2,71	19,0	15,0	12	55	7,0
T6		24,5	22,0	12	53	

Los Temple en extrusión se denominan con la letra “T” seguida de un dígito. Los Temple mas usados en extrusión son:

T - Se aplican a los productos cuyas propiedades mecánicas se obtiene mediante tratamiento térmico. Esta letra se coloca después de la nomenclatura de la aleación de aluminio. El número indica el tratamiento térmico específico y siempre se coloca después de la “T”.

T4 - Enfriamiento por aire después de extruir y envejecimiento natural. Las propiedades mecánicas en este temple son inestables.

T5 - Enfriamiento por aire después de extruir y envejecimiento artificial.

T6 - Enfriamiento por agua después de extruir y envejecimiento artificial.

NOTA: Los Temple “F” para la aleación 1100 y H111 para la aleación 1350 representan: como extruido y estirado 1 - 3%.

PUERTA CORREDERA “SERIE 1000”

Aleación: 6063

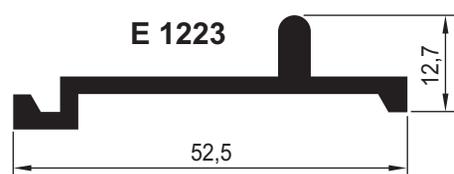
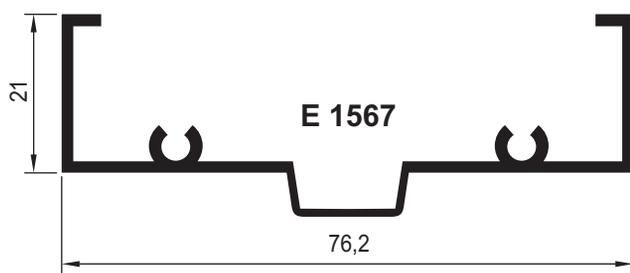
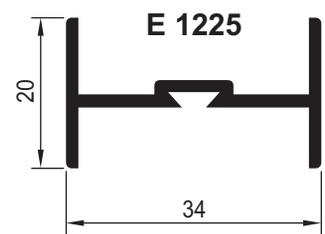
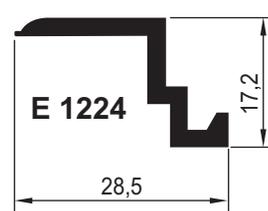
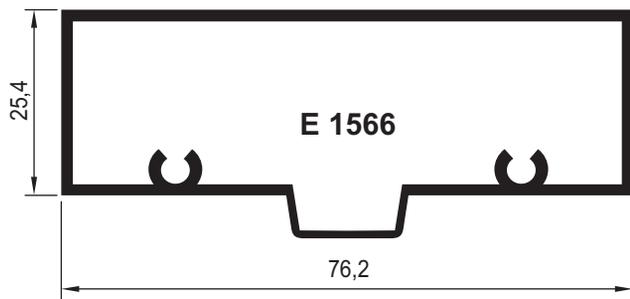
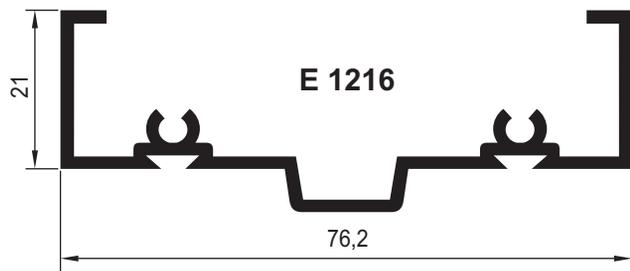
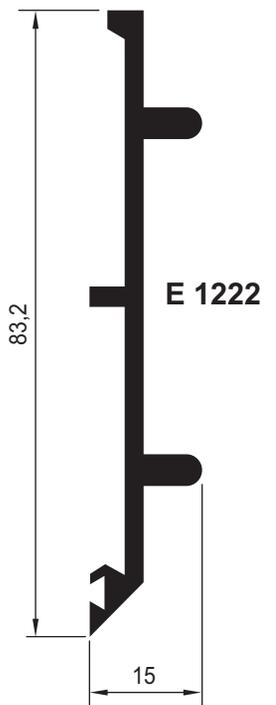
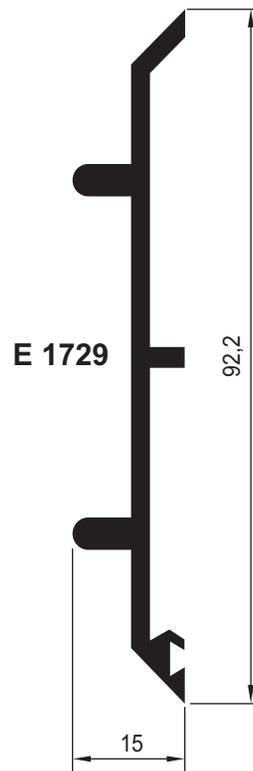
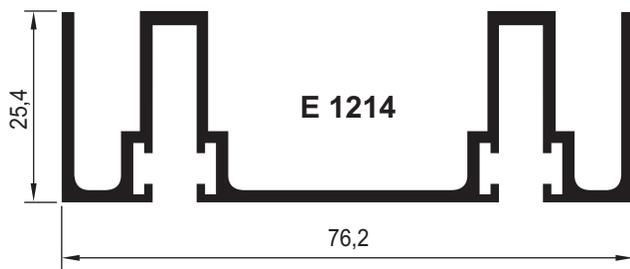
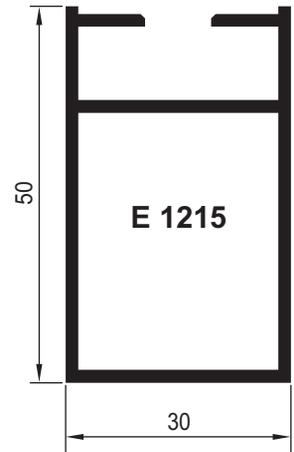
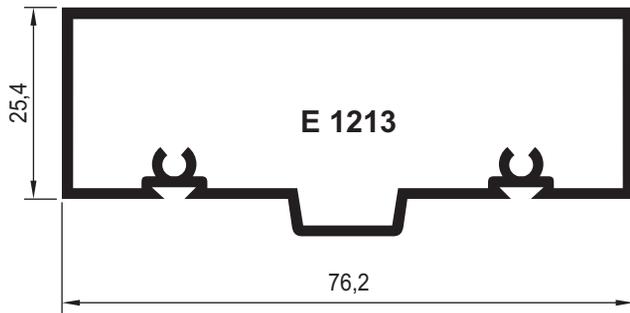
Temple: T5

Largo Normal: 6.10 Mts.

MATRIZ	DENOMINACIÓN	PESO Kg./m.	PESO Kg./Pza.	FACTOR	D.C.C. mm.	ESPESOR mm.
E 1213	CENTRAL	0.978	5.966	27	81	1.6
E 1214	CABEZAL	0.943	5.752	24	81	1.5
E 1215	VERTICAL / HOJA	0.700	4.270	28	59	1.5
E 1216	LATERAL	0.715	4.362	25	79	1.5
E 1217	COMPLEMENTO CABEZAL	0.605	3.691	30	53	1.5
E 1218	HORIZONTAL INFERIOR / HOJA	0.990	6.039	30	85	1.5
E 1219	HORIZONTAL SUPERIOR / HOJA	0.900	5.490	25	74	1.5
E 1220	VERTICAL / HOJA	0.825	5.033	29	59	1.5
E 1221	COMPLEMENTO LATERAL	0.447	2.727	28	53	1.5
E 1222	BASE / GUIA	0.795	4.850	16	83	Variable
E 1223	COMPLEMENTO BASE GUIA	0.488	2.977	18	53	Variable
E 1224	COMPLEMENTO BASE	0.295	1.800	18	32	Variable
E 1225	EMPALME / HOJA	0.314	1.915	30	39	1.5
E 1565	VERTICAL / HOJA	0.797	4.862	31	59	1.4
E 1566	CENTRAL	0.859	5.240	29	81	1.4
E 1567	LATERAL	0.607	3.703	27	81	1.4
E 1729	BASE / GUIA	0.799	4.874	16	92	Variable
E 1564	VERTICAL / HOJA	0.685	4.179	31	59	1.4
E 3164	CENTRAL	1.043	6.362	26	84	1.5

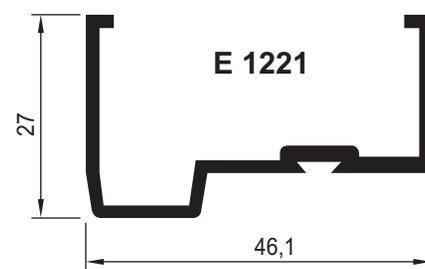
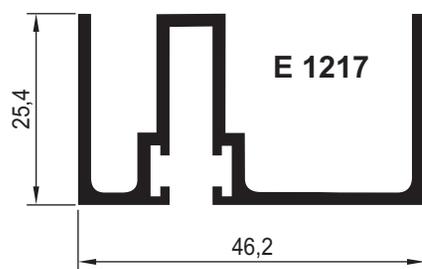
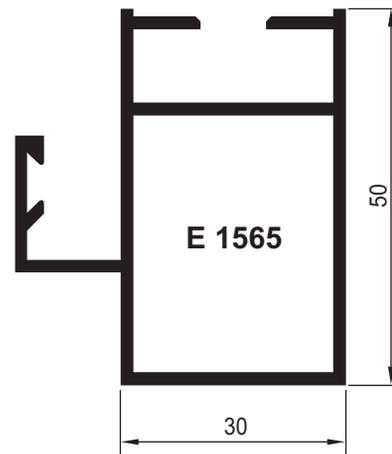
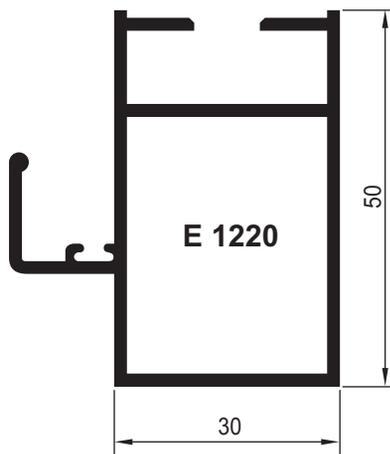
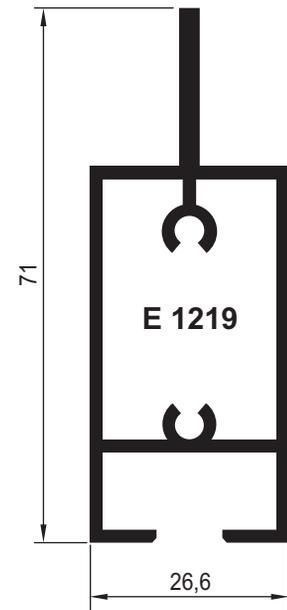
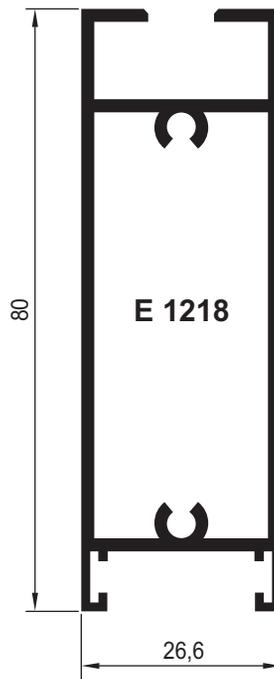
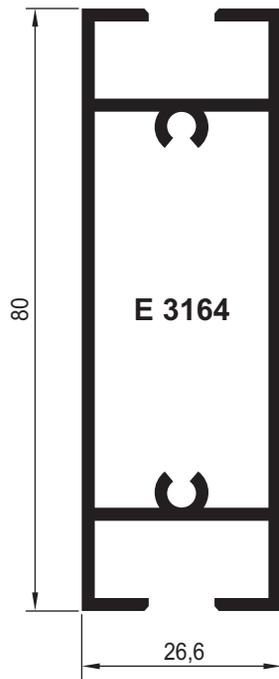
PUERTA CORREDERA "SERIE 1000"

Aleación: 6063 / Temple: T5 / Largo Normal: 6.10 Mts.



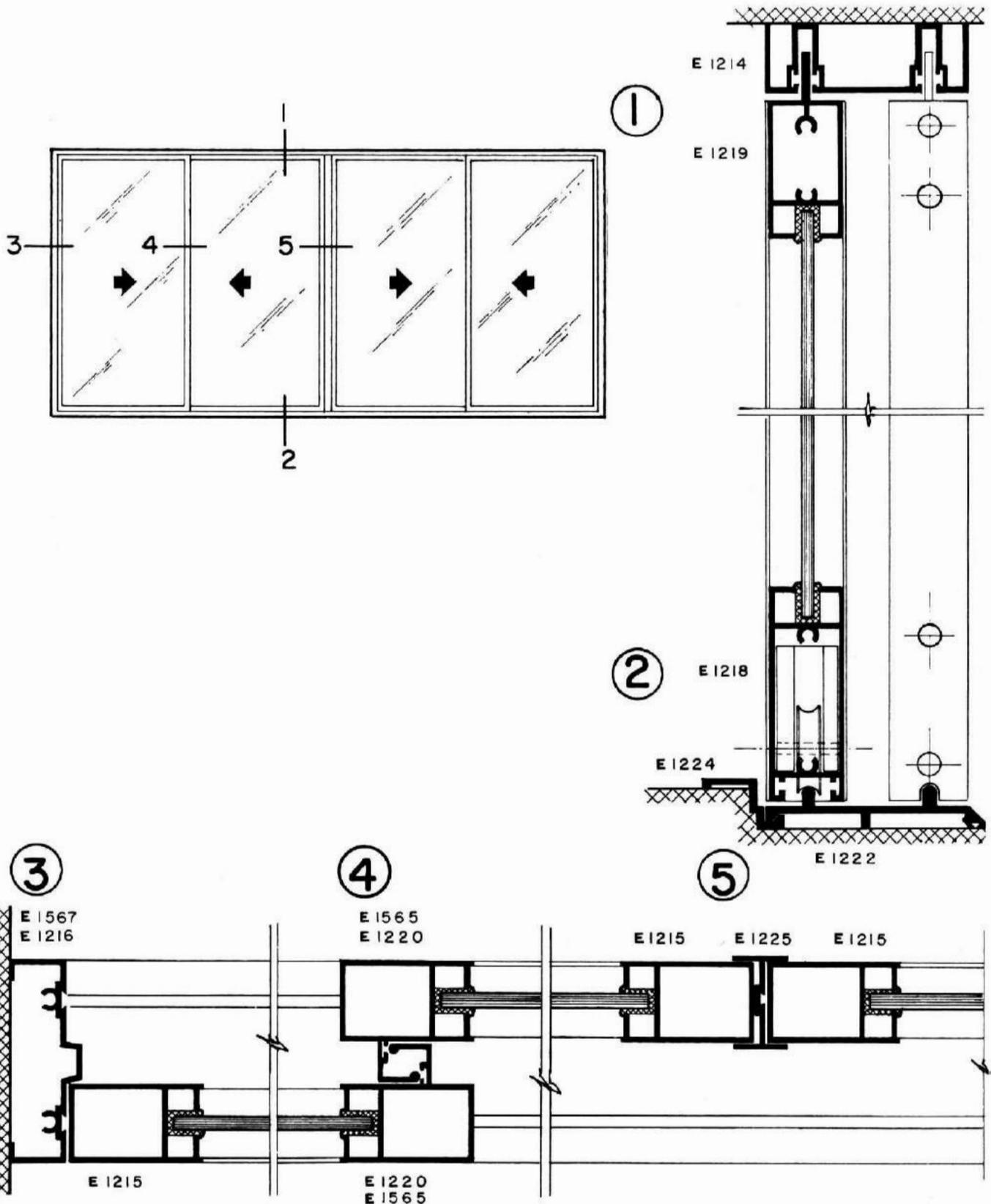
PUERTA CORREDERA "SERIE 1000"

Aleación: 6063 / Temple: T5 / Largo Normal: 6.10 Mts.



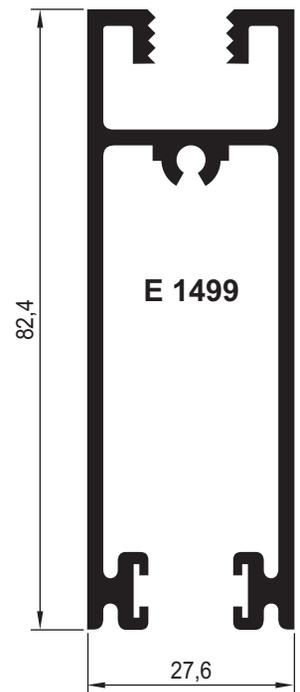
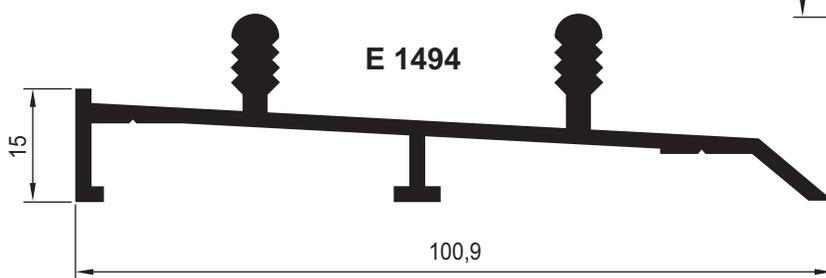
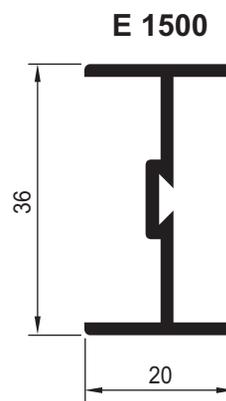
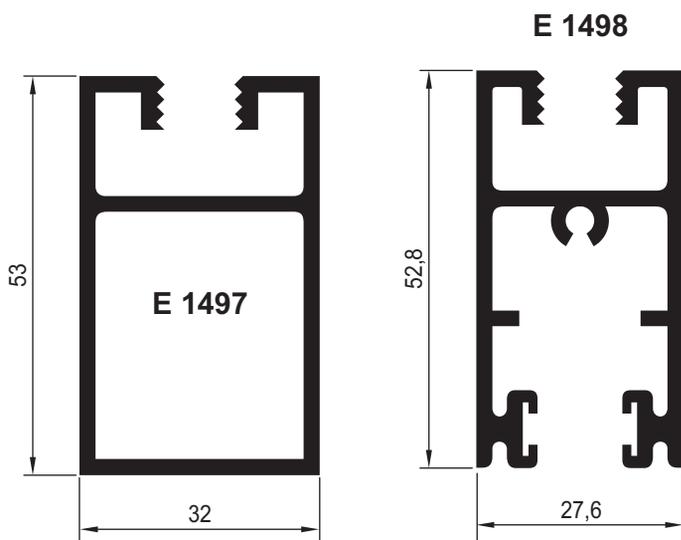
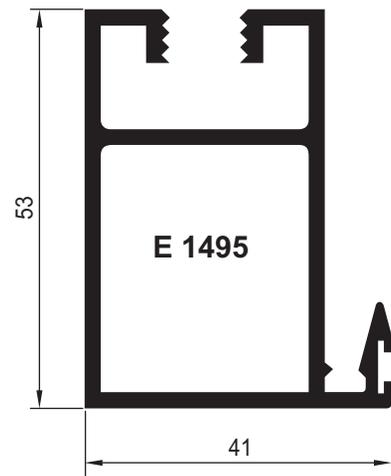
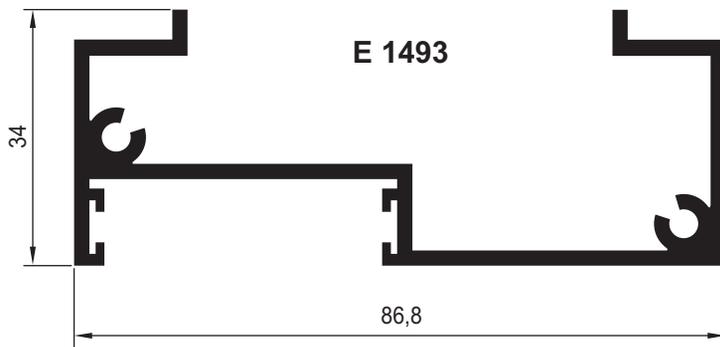
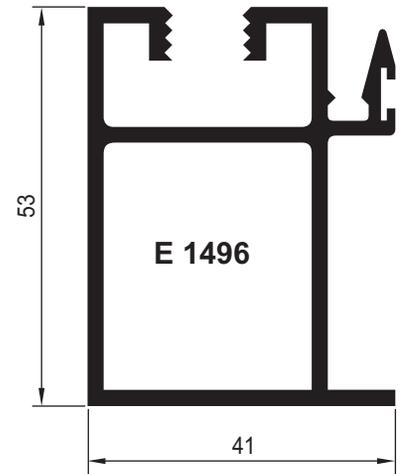
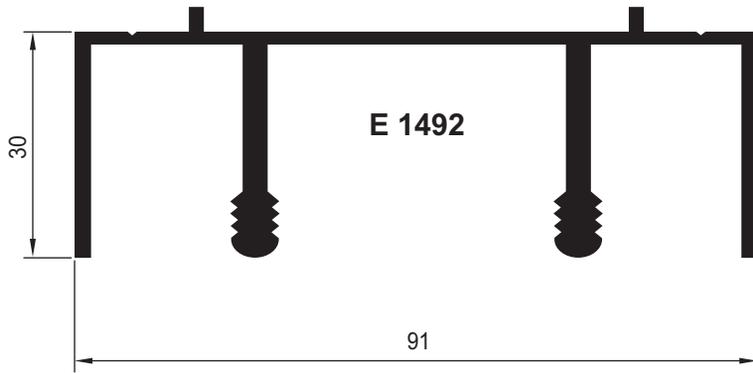
PUERTA CORREDERA "SERIE 1000"

Aleación: 6063 / Temple: T5 / Largo Normal: 6.10 Mts.



PUERTA CORREDERA "SERIE PESADA"

Aleación: 6063 / Temple: T5 / Largo Normal: 6.10 Mts.



PUERTA PARA BAÑO "SERIE A"

Aleación: 6063

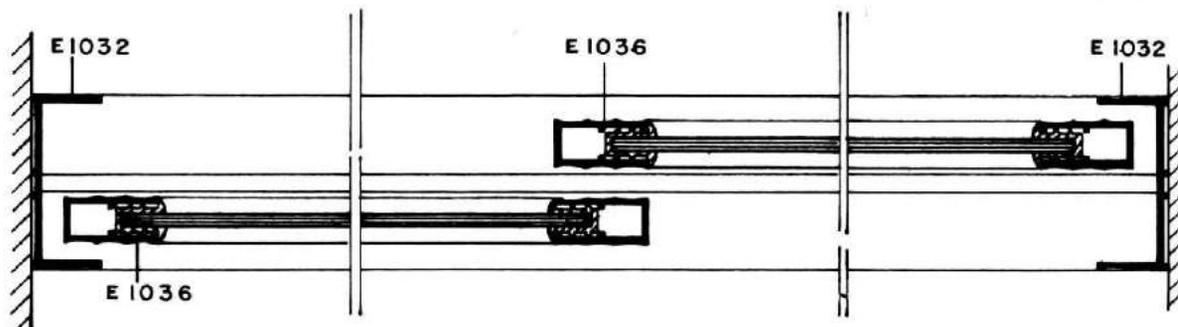
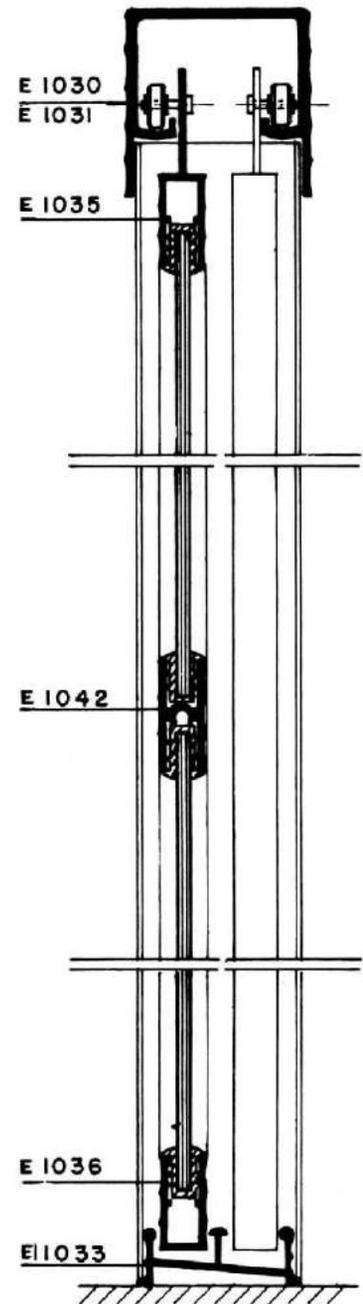
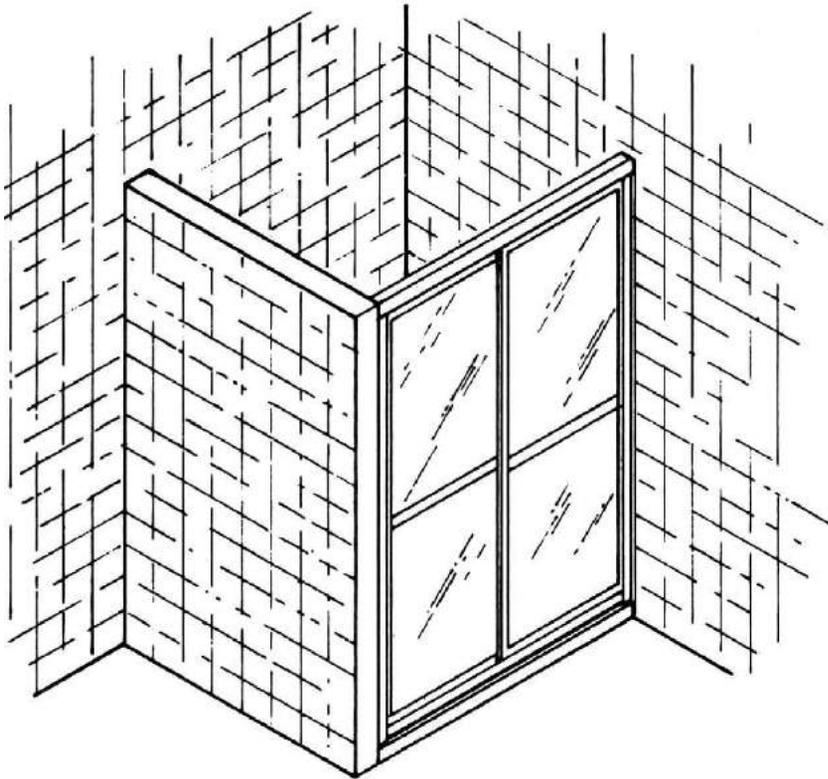
Temple: T5

Largo Normal: 6.10 Mts.

MATRIZ	DENOMINACIÓN	PESO Kg./m.	PESO Kg./Pza.	FACTOR	D.C.C. mm.	ESPESOR mm.
E 1030	CABEZAL	1.024	6.246	22	75	Variable
E 1031	CABEZAL	0.889	5.423	23	71	Variable
E 1032	LATERAL	0.330	2.013	28	52	Variable
E 1033	BASE	0.355	2.166	27	47	Variable
E 1034	CABEZAL	0.295	1.799	28	32	Variable
E 1035	MARCO SUPERIOR	0.395	2.409	28	55	Variable
E 1036	MARCO FINO	0.275	1.677	29	28	Variable
E 1037	MARCO DE CIERRE	0.398	2.428	27	37	Variable
E 1038	MARCO SUPERIOR	0.428	2.601	27	54	Variable
E 1039	MARCO FINO	0.275	1.677	29	28	Variable
E 1040	MARCO INFERIOR	0.285	1.738	27	28	Variable
E 1041	SOPORTE TOALLERO	0.528	3.221	20	70	Variable
E 1042	EMPATE HOJA	0.322	1.964	27	32	Variable
E 1043	TOALLERO	0.238	1.452	23	26	Variable
E 1044	SOPORTE TOALLERO	0.596	3.636	20	77	Variable
E 1045	MARCO	0.523	3.190	30	76	Variable
E 1046	MARCO	0.526	3.209	30	62	Variable
E 2071	BASE	0.436	2.660	25	54	Variable
E 2072	EMPATE MARCO	0.184	1.122	28	29	1.6
E 2073	CABEZAL	1.070	6.527	22	71	Variable
E 3328	MARCO FINO	0.230	1.403	34	28	Variable
E 3472	MARCO FINO	0.190	1.159	42	28	Variable

PUERTA PARA BAÑO "SERIE A"

Aleación: 6063 / Temple: T5 / Largo Normal: 6.10 Mts.



ANEXO E

Scripts de programación en C#

```
//Codigo de animacion para una puerta//
```

```
using UnityEngine;
```

```
using System.Collections;
```

```
public class anipuertabañ2 : MonoBehaviour {
```

```
    private int m_lastindex;
```

```
    // Use this for initialization
```

```
    public void reprptabañ2 ()
```

```
    {
```

```
        if (!animation.isPlaying)
```

```
        {
```

```
            if (m_lastindex == 0) {
```

```
                animation.Play ("ptabañ2open");
```

```
                m_lastindex = 1;
```

```
            } else {
```

```
                animation.Play ("ptabañ2close");
```

```
                m_lastindex = 0;
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

```
//Codigo en c# para el cursor en aplicacion //
```

```
using UnityEngine;
```

```
using System.Collections;
```

```
public class reticle : MonoBehaviour {
```

```
    public Camera CameraFacing;
```

```
    private Vector3 originalScale;
```

```
    // Use this for initialization
```

```
    void Start () {
```

```
        originalScale = transform.localScale;
```

```
    }
```

```
    // Update is called once per frame
```

```
    void Update () {
```

```
        RaycastHit hit;
```

```
        float distance;
```

```
        if (Input.GetMouseButton (0)) {
```

```
            if (Physics.Raycast (new Ray  
(CameraFacing.transform.position, CameraFacing.transform.rotation * Vector3.forward), out hit))
```

```
                Debug.Log("mouse click!");
```

```
                aniptaven1 ptavent1 =hit.transform.GetComponent<aniptaven1>();
```

```
                if(ptavent1)
```

```
                {ptavent1.reprptavent1();
```

```
                    Debug.Log("Balcon Principal");
```

```

    }

    aniptaven2 ptavent2 = hit.transform.GetComponent<aniptaven2>();
    if(ptavent2)
    {ptavent2.reprptavent2();
        Debug.Log("Balcon1 sala");
    }

    aniptaven3 ptavent3 = hit.transform.GetComponent<aniptaven3>();
    if(ptavent3)
    {ptavent3.reprptavent3();
        Debug.Log("Balcon2 sala");
    }

    distance = hit.distance;
}

else {
    distance = CameraFacing.farClipPlane * 0.2f;
}

transform.position = CameraFacing.transform.position +
    CameraFacing.transform.rotation * Vector3.forward * distance;
transform.LookAt (CameraFacing.transform.position);
transform.Rotate (0.0f, 180.0f, 0.0f);
if (distance < 10.0f) {
    distance *= 1 + 5 * Mathf.Exp (-distance);
}

transform.localScale = originalScale * distance;

```

```
}
```

```
}
```

```
//Codigo en c# para ejecucion desde tablet //
```

```
using UnityEngine;
```

```
using System.Collections;
```

```
using System.IO;
```

```
using System.IO.Ports;
```

```
public class aascript : MonoBehaviour {
```

```
    //luces alcoba principal
```

```
    /*a1*/ protected bool alc1 = true;
```

```
    public GameObject a1b1;
```

```
    public GameObject a1r1;
```

```
    public GameObject a1b2;
```

```
    public GameObject a1r2;
```

```
    public GameObject a1b3;
```

```
    public GameObject a1r3;
```

```
    public GameObject a1b4;
```

```
    public GameObject a1r4;
```

```
    public GameObject a1b5;
```

```
    public GameObject a1r5;
```

```
    public GameObject a1b6;
```

```
    public GameObject a1r6;
```

```
    public GameObject a1lamp1;
```

```
    public GameObject a1lamp2;
```

```
    public GameObject a1rcuadro;
```

```
    public GameObject a1rtv;
```

```
    //luces baño principal
```

```
/*b1*/ protected bool ban1 = true;
public GameObject b1b1;
public GameObject b1r1;
public GameObject b1b2;
public GameObject b1r2;
public GameObject b1general;
//luces alcoba 2
/*a2*/ protected bool alc2 = true;
public GameObject a2b1;
public GameObject a2r1;
public GameObject a2b2;
public GameObject a2r2;
//luces baño 2
/*b2*/ protected bool ban2 = true;
public GameObject b2b1;
public GameObject b2r1;
public GameObject b2b2;
public GameObject b2r2;
public GameObject b2general;
//luces estudio
/*e1*/ protected bool est1 = true;
public GameObject eb1;
public GameObject er1;
public GameObject eb2;
public GameObject er2;
//luces sala comedor (l)
/*sc1*/ protected bool salcom1 = true;
public GameObject lb1;
public GameObject lr1;
```

```
public GameObject lb2;
public GameObject lr2;
public GameObject lb3;
public GameObject lr3;
public GameObject lb4;
public GameObject lr4;
public GameObject lb5;
public GameObject lr5;
public GameObject lb6;
public GameObject lr6;
public GameObject lb7;
public GameObject lr7;
public GameObject lb8;
public GameObject lr8;
public GameObject lb9;
public GameObject lr9;
public GameObject lb10;
public GameObject lr10;
public GameObject lb11;
public GameObject lr11;
//luces cocina
/*coc*/ protected bool coc1 = true;
public GameObject cb1;
public GameObject cr1;
public GameObject cb2;
public GameObject cr2;
public GameObject cb3;
public GameObject cr3;
public GameObject cb4;
```

```
public GameObject cr4;
//luces y boquillas chimenea
/*c*/ protected bool chi1 = true;
public GameObject crar;
public GameObject crab;
/*c*/ public GameObject al1b;
/*c*/ public GameObject al2b;
/*c*/ public GameObject al3b;
/*c*/ public GameObject al4b;
/*c*/ public GameObject al5b;
/*c*/ public GameObject al6b;
/*c*/ public GameObject al7b;
/*c*/ public GameObject al8b;

/*c*/ public GameObject bl1b;
/*c*/ public GameObject bl2b;
/*c*/ public GameObject bl3b;
/*c*/ public GameObject bl4b;
/*c*/ public GameObject bl5b;
/*c*/ public GameObject bl6b;
/*c*/ public GameObject bl7b;
/*c*/ public GameObject bl8b;

/*c*/ public GameObject cl1b;
/*c*/ public GameObject cl2b;
/*c*/ public GameObject cl3b;
/*c*/ public GameObject cl4b;
/*c*/ public GameObject cl5b;
/*c*/ public GameObject cl6b;
```

```
/*c*/ public GameObject cl7b;
```

```
/*c*/ public GameObject cl8b;
```

```
/*c*/ public GameObject dl1b;
```

```
/*c*/ public GameObject dl2b;
```

```
/*c*/ public GameObject dl3b;
```

```
/*c*/ public GameObject dl4b;
```

```
/*c*/ public GameObject dl5b;
```

```
/*c*/ public GameObject dl6b;
```

```
/*c*/ public GameObject dl7b;
```

```
/*c*/ public GameObject dl8b;
```

```
/*c*/ public GameObject el1b;
```

```
/*c*/ public GameObject el2b;
```

```
/*c*/ public GameObject el3b;
```

```
/*c*/ public GameObject el4b;
```

```
/*c*/ public GameObject el5b;
```

```
/*c*/ public GameObject el6b;
```

```
/*c*/ public GameObject el7b;
```

```
/*c*/ public GameObject el8b;
```

```
/*c*/ public GameObject fl1b;
```

```
/*c*/ public GameObject fl2b;
```

```
/*c*/ public GameObject fl3b;
```

```
/*c*/ public GameObject fl4b;
```

```
/*c*/ public GameObject fl5b;
```

```
/*c*/ public GameObject fl6b;
```

```
/*c*/ public GameObject fl7b;
```

```
/*c*/ public GameObject fl8b;
```

```

//configuracion serial

private SerialPort puertoBT;

int serial;

int serialw;

string x;

// Use this for initialization

void Start () {

    puertoBT = new SerialPort("COM4",9600);

    puertoBT.Open();

}

// Update is called once per frame

void Update () {

    if (Input.GetKey (KeyCode.T)) {

        serial = puertoBT.ReadByte ();

        /*alcoba1*/ if (serial == 'a')

        /*alcoba1*/{alc1 = !alc1;

            /*alcoba1*/}

        /*alcoba1*/if(alc1){

            /*alcoba1*/  if(!a1b1.light.enabled)

            /*alcoba1*/  {

                /*alcoba1*/  a1b1.light.enabled = true;

                /*alcoba1*/  a1r1.light.enabled = true;

```

```
/*alcoba1*/ a1b2.light.enabled = true;
/*alcoba1*/ a1r2.light.enabled = true;
/*alcoba1*/ a1b3.light.enabled = true;
/*alcoba1*/ a1r3.light.enabled = true;
/*alcoba1*/ a1b4.light.enabled = true;
/*alcoba1*/ a1r4.light.enabled = true;
/*alcoba1*/ a1b5.light.enabled = true;
/*alcoba1*/ a1r5.light.enabled = true;
/*alcoba1*/ a1b6.light.enabled = true;
/*alcoba1*/ a1r6.light.enabled = true;
/*alcoba1*/ a1lamp1.light.enabled = true;
/*alcoba1*/ a1lamp2.light.enabled = true;
/*alcoba1*/ a1rcuadro.light.enabled = true;
/*alcoba1*/ a1rtv.light.enabled = true;

/*alcoba1*/ }}
/*alcoba1*/ else{
/*alcoba1*/ if(a1b1.light.enabled)
/*alcoba1*/ {
/*alcoba1*/ a1b1.light.enabled = false;
/*alcoba1*/ a1r1.light.enabled = false;
/*alcoba1*/ a1b2.light.enabled = false;
/*alcoba1*/ a1r2.light.enabled = false;
/*alcoba1*/ a1b3.light.enabled = false;
/*alcoba1*/ a1r3.light.enabled = false;
/*alcoba1*/ a1b4.light.enabled = false;
/*alcoba1*/ a1r4.light.enabled = false;
/*alcoba1*/ a1b5.light.enabled = false;
/*alcoba1*/ a1r5.light.enabled = false;
```

```

        /*alcoba1*/    a1b6.light.enabled = false;
        /*alcoba1*/    a1r6.light.enabled = false;
        /*alcoba1*/    a1lamp1.light.enabled = false;
        /*alcoba1*/    a1lamp2.light.enabled = false;
        /*alcoba1*/    a1rcuadro.light.enabled = false;
        /*alcoba1*/    a1rtv.light.enabled = false;
        /*alcoba1*/    }}

/*baño1*/ if (serial == 'b')
/*baño1*/{ban1 = !ban1;
        /*baño1*/}

/*baño1*/if(ban1){
        /*baño1*/    if(!b1b1.light.enabled)
        /*baño1*/    {
                /*baño1*/    b1b1.light.enabled = true;
                /*baño1*/    b1r1.light.enabled = true;
                /*baño1*/    b1b2.light.enabled = true;
                /*baño1*/    b1r2.light.enabled = true;
                /*baño1*/    b1general.light.enabled = true;
                /*baño1*/    }}

/*baño1*/    else{
        /*baño1*/    if(b1b1.light.enabled)
        /*baño1*/    {
                /*baño1*/    b1b1.light.enabled = false;
                /*baño1*/    b1r1.light.enabled = false;
                /*baño1*/    b1b2.light.enabled = false;
                /*baño1*/    b1r2.light.enabled = false;
                /*baño1*/    b1general.light.enabled = false;
                /*baño1*/    }}

```

```

/*alcoba2*/ if (serial == 'r')
/*alcoba2*/{alc2 = !alc2;
    /*alcoba2*/}
/*alcoba2*/if(alc2){
    /*alcoba2*/  if(!a2b1.light.enabled)
    /*alcoba2*/  {
        /*alcoba2*/  a2b1.light.enabled = true;
        /*alcoba2*/  a2r1.light.enabled = true;
        /*alcoba2*/  a2b2.light.enabled = true;
        /*alcoba2*/  a2r2.light.enabled = true;
        /*alcoba2*/  }}
/*alcoba2*/  else{
    /*alcoba2*/  if(a2b1.light.enabled)
    /*alcoba2*/  {
        /*alcoba2*/  a2b1.light.enabled = false;
        /*alcoba2*/  a2r1.light.enabled = false;
        /*alcoba2*/  a2b2.light.enabled = false;
        /*alcoba2*/  a2r2.light.enabled = false;
        /*alcoba2*/  }}

```

```

/*baño2*/ if (serial == 'v')
/*baño2*/{ban2 = !ban2;
    /*baño2*/}
/*baño2*/if(ban2){
    /*baño2*/  if(!b2b1.light.enabled)
    /*baño2*/  {
        /*baño2*/  b2b1.light.enabled = true;
        /*baño2*/  b2r1.light.enabled = true;
        /*baño2*/  b2b2.light.enabled = true;

```

```

        /*baño2*/    b2r2.light.enabled = true;
        /*baño2*/    b2general.light.enabled = true;
        /*baño2*/    }}
/*baño2*/    else{
        /*baño2*/    if(b2b1.light.enabled)
        /*baño2*/    {
            /*baño2*/    b2b1.light.enabled = false;
            /*baño2*/    b2r1.light.enabled = false;
            /*baño2*/    b2b2.light.enabled = false;
            /*baño2*/    b2r2.light.enabled = false;
            /*baño2*/    b2general.light.enabled = false;
            /*baño2*/    }}

```

```

/*estudio*/ if (serial == 'e')
/*estudio*/{est1 = !est1;
        /*estudio*/}
/*estudio*/if(est1){
        /*estudio*/    if(!eb1.light.enabled)
        /*estudio*/    {
            /*estudio*/    eb1.light.enabled = true;
            /*estudio*/    er1.light.enabled = true;
            /*estudio*/    eb2.light.enabled = true;
            /*estudio*/    er2.light.enabled = true;
            /*estudio*/    }}
/*estudio*/    else{
        /*estudio*/    if(eb1.light.enabled)
        /*estudio*/    {
            /*estudio*/    eb1.light.enabled = false;
            /*estudio*/    er1.light.enabled = false;

```

```
/*estudio*/ eb2.light.enabled = false;  
/*estudio*/ er2.light.enabled = false;  
/*estudio*/ }}
```

```
/*L*/ if (serial == 'l')  
/*L*/{salcom1 = !salcom1;  
/*L*/}  
/*L*/if(salcom1){  
/*L*/ if(!lb1.light.enabled)  
/*L*/ {  
/*L*/ lb1.light.enabled = true;  
/*L*/ lr1.light.enabled = true;  
/*L*/ lb2.light.enabled = true;  
/*L*/ lr2.light.enabled = true;  
/*L*/ lb3.light.enabled = true;  
/*L*/ lr3.light.enabled = true;  
/*L*/ lb4.light.enabled = true;  
/*L*/ lr4.light.enabled = true;  
/*L*/ lb5.light.enabled = true;  
/*L*/ lr5.light.enabled = true;  
/*L*/ lb6.light.enabled = true;  
/*L*/ lr6.light.enabled = true;  
/*L*/ lb7.light.enabled = true;  
/*L*/ lr7.light.enabled = true;  
/*L*/ lb8.light.enabled = true;  
/*L*/ lr8.light.enabled = true;  
/*L*/ lb9.light.enabled = true;  
/*L*/ lr9.light.enabled = true;  
/*L*/ lb10.light.enabled = true;
```

```
        /*L*/ lr10.light.enabled = true;
        /*L*/ lb11.light.enabled = true;
        /*L*/ lr11.light.enabled = true;
        /*L*/ }}
/*L*/ else{
    /*L*/ if(lb1.light.enabled)
    /*L*/ {
        /*L*/ lb1.light.enabled = false;
        /*L*/ lr1.light.enabled = false;
        /*L*/ lb2.light.enabled = false;
        /*L*/ lr2.light.enabled = false;
        /*L*/ lb3.light.enabled = false;
        /*L*/ lr3.light.enabled = false;
        /*L*/ lb4.light.enabled = false;
        /*L*/ lr4.light.enabled = false;
        /*L*/ lb5.light.enabled = false;
        /*L*/ lr5.light.enabled = false;
        /*L*/ lb6.light.enabled = false;
        /*L*/ lr6.light.enabled = false;
        /*L*/ lb7.light.enabled = false;
        /*L*/ lr7.light.enabled = false;
        /*L*/ lb8.light.enabled = false;
        /*L*/ lr8.light.enabled = false;
        /*L*/ lb9.light.enabled = false;
        /*L*/ lr9.light.enabled = false;
        /*L*/ lb10.light.enabled = false;
        /*L*/ lr10.light.enabled = false;
        /*L*/ lb11.light.enabled = false;
        /*L*/ lr11.light.enabled = false;
```

```
/*L*/  }}
```

```
/*cocina*/ if (serial == 'k')
```

```
/*cocina*/{coc1 = !coc1;
```

```
/*cocina*/}
```

```
/*cocina*/if(coc1){
```

```
/*cocina*/  if(!cb1.light.enabled)
```

```
/*cocina*/  {
```

```
/*cocina*/    cb1.light.enabled = true;
```

```
/*cocina*/    cr1.light.enabled = true;
```

```
/*cocina*/    cb2.light.enabled = true;
```

```
/*cocina*/    cr2.light.enabled = true;
```

```
/*cocina*/    cb3.light.enabled = true;
```

```
/*cocina*/    cr3.light.enabled = true;
```

```
/*cocina*/    cb4.light.enabled = true;
```

```
/*cocina*/    cr4.light.enabled = true;
```

```
/*cocina*/  }}
```

```
/*cocina*/  else{
```

```
/*cocina*/    if(cb1.light.enabled)
```

```
/*cocina*/    {
```

```
/*cocina*/      cb1.light.enabled = false;
```

```
/*cocina*/      cr1.light.enabled = false;
```

```
/*cocina*/      cb2.light.enabled = false;
```

```
/*cocina*/      cr2.light.enabled = false;
```

```
/*cocina*/      cb3.light.enabled = false;
```

```
/*cocina*/      cr3.light.enabled = false;
```

```
/*cocina*/      cb4.light.enabled = false;
```

```
/*cocina*/      cr4.light.enabled = false;
```

```
/*cocina*/    }}
```

```
/*chimenea*/ if (serial == 'h')
/*chimenea*/{ chi1 = !chi1;
    /*chimenea*/}
/*chimenea*/if(chi1){
    /*chimenea*/ if(!crar.light.enabled)
    /*chimenea*/ {
        /*chimenea*/ crar.light.enabled = true;
        /*chimenea*/ crab.light.enabled = true;

        /*B*/a11b.particleSystem.Play();
        /*B*/a12b.particleSystem.Play();
        /*B*/a13b.particleSystem.Play();
        /*B*/a14b.particleSystem.Play();
        /*B*/a15b.particleSystem.Play();
        /*B*/a16b.particleSystem.Play();
        /*B*/a17b.particleSystem.Play();
        /*B*/a18b.particleSystem.Play();

        /*B*/b11b.particleSystem.Play();
        /*B*/b12b.particleSystem.Play();
        /*B*/b13b.particleSystem.Play();
        /*B*/b14b.particleSystem.Play();
        /*B*/b15b.particleSystem.Play();
        /*B*/b16b.particleSystem.Play();
        /*B*/b17b.particleSystem.Play();
        /*B*/b18b.particleSystem.Play();

        /*B*/c11b.particleSystem.Play();
```

```
/*B*/cl2b.particleSystem.Play();  
/*B*/cl3b.particleSystem.Play();  
/*B*/cl4b.particleSystem.Play();  
/*B*/cl5b.particleSystem.Play();  
/*B*/cl6b.particleSystem.Play();  
/*B*/cl7b.particleSystem.Play();  
/*B*/cl8b.particleSystem.Play();
```

```
/*B*/dl1b.particleSystem.Play();  
/*B*/dl2b.particleSystem.Play();  
/*B*/dl3b.particleSystem.Play();  
/*B*/dl4b.particleSystem.Play();  
/*B*/dl5b.particleSystem.Play();  
/*B*/dl6b.particleSystem.Play();  
/*B*/dl7b.particleSystem.Play();  
/*B*/dl8b.particleSystem.Play();
```

```
/*B*/el1b.particleSystem.Play();  
/*B*/el2b.particleSystem.Play();  
/*B*/el3b.particleSystem.Play();  
/*B*/el4b.particleSystem.Play();  
/*B*/el5b.particleSystem.Play();  
/*B*/el6b.particleSystem.Play();  
/*B*/el7b.particleSystem.Play();  
/*B*/el8b.particleSystem.Play();
```

```
/*B*/fl1b.particleSystem.Play();  
/*B*/fl2b.particleSystem.Play();  
/*B*/fl3b.particleSystem.Play();
```

```
/*B*/fl4b.particleSystem.Play();
/*B*/fl5b.particleSystem.Play();
/*B*/fl6b.particleSystem.Play();
/*B*/fl7b.particleSystem.Play();
/*B*/fl8b.particleSystem.Play();

/*chimenea*/ }}
/*chimenea*/ else{
/*chimenea*/ if(crar.light.enabled)
/*chimenea*/ {
/*chimenea*/ crar.light.enabled = false;
/*chimenea*/ crab.light.enabled = false;

/*B*/al1b.particleSystem.Stop();
/*B*/al2b.particleSystem.Stop();
/*B*/al3b.particleSystem.Stop();
/*B*/al4b.particleSystem.Stop();
/*B*/al5b.particleSystem.Stop();
/*B*/al6b.particleSystem.Stop();
/*B*/al7b.particleSystem.Stop();
/*B*/al8b.particleSystem.Stop();

/*B*/bl1b.particleSystem.Stop();
/*B*/bl2b.particleSystem.Stop();
/*B*/bl3b.particleSystem.Stop();
/*B*/bl4b.particleSystem.Stop();
/*B*/bl5b.particleSystem.Stop();
/*B*/bl6b.particleSystem.Stop();
/*B*/bl7b.particleSystem.Stop();
```

```
/*B*/bl8b.particleSystem.Stop();
```

```
/*B*/cl1b.particleSystem.Stop();
```

```
/*B*/cl2b.particleSystem.Stop();
```

```
/*B*/cl3b.particleSystem.Stop();
```

```
/*B*/cl4b.particleSystem.Stop();
```

```
/*B*/cl5b.particleSystem.Stop();
```

```
/*B*/cl6b.particleSystem.Stop();
```

```
/*B*/cl7b.particleSystem.Stop();
```

```
/*B*/cl8b.particleSystem.Stop();
```

```
/*B*/dl1b.particleSystem.Stop();
```

```
/*B*/dl2b.particleSystem.Stop();
```

```
/*B*/dl3b.particleSystem.Stop();
```

```
/*B*/dl4b.particleSystem.Stop();
```

```
/*B*/dl5b.particleSystem.Stop();
```

```
/*B*/dl6b.particleSystem.Stop();
```

```
/*B*/dl7b.particleSystem.Stop();
```

```
/*B*/dl8b.particleSystem.Stop();
```

```
/*B*/el1b.particleSystem.Stop();
```

```
/*B*/el2b.particleSystem.Stop();
```

```
/*B*/el3b.particleSystem.Stop();
```

```
/*B*/el4b.particleSystem.Stop();
```

```
/*B*/el5b.particleSystem.Stop();
```

```
/*B*/el6b.particleSystem.Stop();
```

```
/*B*/el7b.particleSystem.Stop();
```

```
/*B*/el8b.particleSystem.Stop();
```

```
/*B*/fl1b.particleSystem.Stop();  
/*B*/fl2b.particleSystem.Stop();  
/*B*/fl3b.particleSystem.Stop();  
/*B*/fl4b.particleSystem.Stop();  
/*B*/fl5b.particleSystem.Stop();  
/*B*/fl6b.particleSystem.Stop();  
/*B*/fl7b.particleSystem.Stop();  
/*B*/fl8b.particleSystem.Stop();  
  
/*chimenea*/ }}
```

```
}}
```