



**UNIVERSIDAD DE VALLADOLID
ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN**



**TRABAJO FIN DE GRADO
GRADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN**

**JUEGO SERIO PARA ESTIMULACIÓN FÍSICA Y
COGNITIVA**

AUTOR: MARÍA V. FERNÁNDEZ SAN ROMÁN
TUTOR: MARIO MARTÍNEZ ZARZUELA

junio de 2015

Título: **Juego serio para estimulación física y cognitiva**

Autora: **María V. Fernández San Román**

Tutor: **Mario Martínez Zarzuela**

Departamento: **Departamento de Teoría de la Señal y
Comunicaciones e Ingeniería Telemática**

Tribunal

Presidente: **Míriam Antón Rodríguez**

Secretario: **David González Ortega**

Vocal: **Mario Martínez Zarzuela**

Suplente 1: **Francisco Javier Díaz Pernas**

Suplente 2: **Carlos Gómez Peña**

Fecha: **26 de junio de 2015**

Calificación

A mi familia, porque todo lo que soy es gracias a ellos.

A mis amigos, por todo el apoyo incondicional recibido y por todos esos buenos momentos vividos y por vivir.

A Mario, por su confianza, apoyo y paciencia.

A Carlos, Miguel y Sergio, por toda la ayuda recibida y por amenizar el desarrollo de todo este proyecto.

Resumen de TFG

El Trabajo Fin de Grado desarrollado consiste en la realización de un módulo de evaluación dentro de un sistema enfocado como videojuego que pretende servir como complemento en la rehabilitación de personas con problemas de movilidad tras haber sufrido un ictus y a su vez, como ayuda para los fisioterapeutas al facilitarles la tarea medición, recogida y almacenamiento de datos.

En primer lugar se realizará un estudio acerca de la naturaleza, incidencia y secuelas del ictus, para después pasar a tratar el estado del arte de los videojuegos. Más tarde se discutirán las posibles tecnologías a utilizar, para acabar explicando finalmente la solución alcanzada y las futuras líneas de desarrollo.

Palabras clave

Ictus, fisioterapeuta, Realidad Virtual, Juego Serio, Motor de Juegos, Kinect, *Joints*.

Abstract

This Final Project consists on the development of a test which will be a unit within a system. This system will be focused as a videogame and it is expected to be useful as a supplementary tool on the physical therapy that people who have mobility issues due to a brain stroke have to make. In addition, this system also aim for being useful to the physiotherapist, thanks to the ease when it comes to measure, collect and store the data.

First, it will be discussed the nature, impact and consequences of brain strokes nowadays. Afterwards, a research of the videogames state of the art will be made. Later, the possible different technologies to implement will be compared, and finally, the final solution and the future guidelines will be deeply explained.

Keywords

Brain stroke, phisiotherapist, Virtual Reality, Serious Game, Game Engine, Kinect, *Joints*.

Índice de contenidos

Capítulo 1. Introducción	16
1.1. Motivación	16
1.2. Conocimientos previos.....	17
1.3. Objetivos	18
Capítulo 2. Ictus	19
2.1. Definición, factores de riesgo y síntomas de alarma.....	19
2.2. Secuelas.....	21
2.3. Incidencia y rehabilitación	22
2.3.1. Técnicas de compensación.....	24
2.3.2. Técnicas de facilitación.....	25
2.3.3. Técnicas modernas	25
2.3.4. Recuperación del equilibrio	27
2.4. Realidad virtual en rehabilitación	28
2.4.1. Entrenamiento en cinta con realidad virtual.....	29
2.4.2. Entrenamiento con sistemas de <i>tracking</i>	30
2.4.3. Entrenamiento a través de juegos serios	31
Capítulo 3. Videojuegos.....	33
3.1. Historia de los videojuegos	33
3.1.1. Steam.....	36
3.2. Clasificación de los videojuegos.....	37
3.2.1. Acción.....	37
3.2.2. Simulación.....	38
3.2.3. Arcade	39
3.2.4. Deporte.....	39
3.2.5. Carreras	39
3.2.6. Agilidad mental.....	40
3.2.7. Educación.....	40
3.2.8. Aventura.....	40
3.2.9. Rol.....	40
3.3. Perfil de los jugadores.....	41
3.4. Gamificación.....	44

3.4.1.	Mecánicas de juego.....	44
3.4.2.	Dinámicas de juego.....	45
Capítulo 4.	Tecnologías	47
4.1.	Motores de juego y herramientas de desarrollo	47
4.1.1.	Unity	48
4.1.2.	Cocos2d-x	51
4.1.3.	Adobe AIR	54
4.1.4.	Unreal.....	57
4.1.5.	Comparativa y elección.....	59
4.2.	Creación y animación de objetos.	60
4.3.	Dispositivos de captura de movimiento	61
4.3.1.	Kinect.....	63
4.3.2.	Xtion PRO LIVE.....	65
4.3.3.	DepthSense	66
4.3.4.	Minoru 3D.....	67
4.3.5.	Comparativa y elección.....	68
4.4.	<i>Plugin</i> para Kinect	69
Capítulo 5.	Módulo desarrollado	70
5.1.	Aspectos generales.....	70
5.2.	Módulo de administración	71
5.3.	Módulo de terapeuta	72
5.4.	Módulo de juego	73
5.5.	Módulo de valoración	75
5.5.1.	Pantalla inicial.....	75
5.5.2.	Test de valoración	77
5.5.3.	Tratamiento de resultados	87
5.5.4.	Estimación presupuestaria.....	89
Capítulo 6.	Conclusiones y líneas futuras	90
6.1.	Conclusiones	90
6.2.	Líneas futuras.....	91

Índice de figuras

Figura 2.1: Distintos tipos de ictus

Figura 2.2: Entrenamiento con cinta rodante

Figura 2.3: Trayectoria del centro de masas durante 30 segundos con los ojos abiertos en un hombre de 48 años afectado por un infarto cerebral en el hemisferio derecho.

Figura 2.4: Elementos del sistema empleado en el estudio

Figura 2.5: Barra instalada en el lateral de la cinta

Figura 2.6: Alcance mediante hombros en sedestación y alcance mediante pies en bipedestación, respectivamente

Figura 2.7: Ejemplo de juego que hace uso de la tabla Wii Balance: *We Ski*

Figura 3.1: Tennis for Two de William Higinbotham.

Figura 3.2: Imagen del videojuego *Final Fantasy VII* para *PlayStation*.

Figura 3.3: Imagen del videojuego Operation Flashpoint

Figura 3.4: Preferencia de videojuegos en España

Figuras 3.5 y 3.6: Porcentajes de jugadores según sexo, edad y frecuencia

Figura 3.7: Tipos de videojuegos preferidos atendiendo a género y edad

Figura 3.8: Porcentaje de uso de distintos dispositivos para jugar

Figura 3.9: Adjetivos asociados comúnmente a los videojuegos

Figura 4.1: Herramientas de desarrollo más utilizadas

Figura 4.2: Logo de Unity

Figura 4.3: Imagen del videojuego Blackguards 2

Figura 4.4: Logo de Cocos2d-x

Figura 4.5: Imagen del juego Dragonhunter

Figura 4.6: Logo de Adobe AIR

Figura 4.7: Imagen del juego Lightstorm

Figura 4.8: Logo de Unreal Engine

Figura 4.9: Imagen del juego Infiltrator

Figura 4.10: Kinect para Windows v2

Figura 4.11: Xtion PRO LIVE

Figura 4.12: DepthSense DS325

Figura 4.13: Minoru 3D

Figura 5.1: Interfaz del módulo de administración

Figura 5.2: Interfaz del módulo de terapeuta

Figura 5.3: Interfaz del módulo de juego. De arriba a abajo y de izquierda a derecha, selección del paciente, selección de la sesión de entrenamiento, configuración de los parámetros y momento durante el juego

Figura 5.4: Paneles del módulo de valoración

Figura 5.5: Panel de selección de avatar y algunos de los avatares disponibles

Figura 5.6: Herramienta *Terrain* de Unity

Figura 5.7: Ejemplo de animación del avatar. Flexión de rodillas.

Figura 5.8: Jerarquía de los joints

Figura 5.9: Posición de los *joints*.

Figura 5.10: Joint Orientation

Figura 5.11: Fotos tomadas durante la evaluación y sus correspondientes valores

Figura 5.12: Ejemplo de gráfica generada con la Macro

Figura 5.13: Código correspondiente a la Macro creada en VBA

Figura 6.1: Diagrama de interconexión

Índice de tablas

Tabla 4.1: Relación lenguajes-plataformas soportados por Cocos2d-x

Tabla 4.2: Comparativa entre las dos versiones de Kinect

Tabla 4.3: Comparativa entre los distintos dispositivos de captura analizados.

Tabla 5.1: Relación de interacción entre módulos y actores en el sistema

Tabla 5.2: Modos de juego

Tabla 5.3: Movimientos a evaluar

Tabla 5.4: Estimación presupuestaria

Capítulo 1. Introducción

Este primer capítulo pretende servir como introducción al Trabajo Fin de Grado desarrollado. Para ello, se comentará la motivación que me llevó a elegir un proyecto de estas características y los conocimientos previos de los que disponía para asumir esta tarea, así como el punto del que se partía y qué metas pudieron alcanzarse.

1.1. Motivación

Según datos del Instituto Nacional de Estadística, en España se producen cada año unos 100.000 nuevos casos de ictus. En el Centro de Fisioterapia de la Fundación ASPAYM CyL son muy conscientes de esta situación, ya que se encargan de dar tratamiento a más de 300 pacientes mensualmente, con una media de 10 sesiones de terapia cada mes por paciente.

Es algo común que los fisioterapeutas recomienden a sus pacientes ejercicios que puedan realizar en sus domicilios como trabajo complementario para maximizar los beneficios del proceso de rehabilitación. Desgraciadamente, es igualmente común que los pacientes no sigan estas recomendaciones, habitualmente por la imposibilidad de una supervisión de estas actividades complementarias junto con una falta de motivación para la realización de las mismas.

Otra necesidad detectada radica en los períodos vacacionales. Épocas en las que los pacientes no acuden al Centro de Fisioterapia durante varias, hecho que acarrea la pérdida de muchas de las adaptaciones adquiridas a lo largo de las sesiones previas.

Asimismo, Castilla y León es la Comunidad Autónoma más extensa del territorio nacional. Este hecho hace que multitud de pacientes crónicos con discapacidad y una necesidad diaria de rehabilitación, no puedan acceder con facilidad a grandes núcleos de población en donde dispongan de un servicio de rehabilitación adecuado.

Por todos estos motivos, el Grupo de Telemática e Imagen de la Universidad de Valladolid decidió asociarse con la Fundación ASPAYM a través de los premios Desafío Universidad-Empresa. En estos premios, empresas y asociaciones de interés público lanzan una demanda tecnológica de forma anónima y los grupos de investigación de las diferentes universidades concursan para proporcionar la mejor solución técnica que lo resuelva.

En la edición 2014 el premio para las asociaciones de interés público fue otorgado a la solución tecnológica propuesta por el Grupo de Telemática e Imagen (GTI) de la Universidad de Valladolid, que ofrecía el desarrollo de un completo sistema de rehabilitación a través de interfaces de usuario con reconocimiento de gestos, denominado TIDER-3D: Tele-asistencia Interactiva en Domicilio con Ejercicios de Rehabilitación 3D, dando respuesta a las necesidades planteadas desde la Fundación ASPAYM. En la propuesta, los investigadores del GTI indicaban

cómo empleando el dispositivo Kinect de Microsoft es posible crear un sistema de tele-rehabilitación muy beneficioso y, al mismo tiempo, motivador para los pacientes.

Es necesario destacar que la inmensa mayoría de los videojuegos, y los de más éxito a la par, están dedicados únicamente al entretenimiento sin tener en cuenta algún otro fin constructivo, pero debido a que en pleno siglo XXI hay un convencimiento de que cualquier expresión del medio digital es parte de nuestra cultura y por lo tanto debe formar parte de la educación de las personas, son cada vez más las aplicaciones o videojuegos educativos que van surgiendo en el mercado (Gros, 2008).

En cuanto a los videojuegos destinados a la rehabilitación, los continuos avances en campos como la robótica, la Realidad Virtual o la Realidad Aumentada propician que cada vez haya más proyectos e investigaciones sobre esta materia (Yang, S. *et al.*, 2011; Lloréns, R. *et al.*, 2013; Deutsch, J.E. *et al.*, 2011; Ki Hun Cho *et al.*, 2012; Bower, K.J. *et al.*, 2014). Todos estos estudios se traducen en la posibilidad de complementar o incluso sustituir a las terapias convencionales por otras novedosas de al menos igual eficacia, pero que además pretenden añadir otros factores cognitivos a la rehabilitación, como la motivación y el interés.

1.2. Conocimientos previos

Este proyecto ha sido desarrollado en el lenguaje de programación C#, que aunque no se estudió como tal a lo largo de la carrera, sí se han estudiado muchos otros (ensamblador, C, C++, Java, HTML, CSS, JavaScript, etc.) que permitieron adquirir una base que facilita la tarea de aprendizaje de cualquier otro lenguaje nuevo. Por este motivo, hubo que dedicar algo de tiempo a aprender a manejar correctamente este lenguaje orientado a objetos.

Independientemente de los conocimientos adquiridos sobre C#, no disponía de ninguno sobre el motor de juegos que se iba a emplear, motivo por el que se dedicaron varias semanas a aprender a manejar Unity, del cual se hablará más adelante en profundidad. Unity es un motor de juego muy potente con múltiples herramientas, razón por la que, a pesar de los tutoriales realizados, muchas veces había que hacer un alto en el desarrollo del código para aprender nuevas funcionalidades que eran necesarias para el proyecto.

En cuanto al dispositivo de captura de movimiento a utilizar, Kinect de Microsoft, solo la conocía a nivel de usuario pero no sabía cómo funcionaba a un nivel más profundo. Para empezar a comprender su funcionamiento, se consultó el Trabajo Fin de Grado previo a este, *Juego Serio para entrenamiento de personas con discapacidad física mediante tecnología Kinect*, llevado a cabo por Ana Isabel Hervás en esta misma Universidad. Esto fue verdaderamente útil para iniciarme con este dispositivo, aunque dicho TFG fue desarrollado con

la primera versión de Kinect y en este, como ya veremos, se decidió partir de cero con la nueva versión de Kinect, lanzada en la segunda mitad del año 2014.

1.3. Objetivos

El objetivo principal del presente Trabajo Fin de Grado era el de desarrollar un módulo de evaluación dentro de una plataforma que sirviera para complementar la rehabilitación recibida por pacientes que hubieran sufrido un ictus.

Dicho módulo debía evaluar con precisión el Rango de Movimiento Máximo alcanzado por cada paciente en cada uno de los movimientos a realizar, de modo que sustituyera la evaluación tradicional llevada a cabo con goniómetros, además de permitir a los terapeutas llevar un registro y un control más sencillo de todos los datos obtenidos a raíz de cada una de las evaluaciones.

Para ello, en primer lugar se debía estudiar cómo el dispositivo Kinect de Microsoft recogía los datos obtenidos de nuestro cuerpo, y cómo tratar esos datos para poder traducirlos a algo fácilmente interpretable por los terapeutas, usando los mismos sistemas de referencia que ellos han empleado durante muchos años.

A partir de aquí nos pudimos centrar en otros objetivos en principio secundarios, como era el desarrollo de un entorno que fuera fácil de entender por cualquier paciente independientemente de su edad, y que a su vez resultara llamativo pero no contara con demasiados elementos, para poder así motivar a los pacientes pero sin afectar negativamente a los resultados que pudieran obtener si se aplicara el método de medida tradicional.

Capítulo 2. Ictus

Tanto el sistema de evaluación desarrollado que nos ocupa en este Trabajo Fin de Grado como el juego que hace uso de él está enfocado desde un primer momento a pacientes que han sufrido un ictus y necesitan de la correspondiente rehabilitación. Por eso, es importante saber en qué consiste esta patología, quién presenta un mayor riesgo de sufrirla y cómo reconocer los primeros síntomas, así como su incidencia en la población y la terapia a llevar a cabo una vez que el paciente la ha sufrido.

2.1. Definición, factores de riesgo y síntomas de alarma

Un ictus es una enfermedad cerebrovascular que se produce por una variación del flujo sanguíneo que llega al cerebro, ya sea por un déficit del mismo o por un exceso debido a la ruptura de un vaso cerebral. Al primero de los casos se le denomina ictus isquémico y supone el 85% del total. En este tipo de ictus, las células nerviosas no reciben el suficiente oxígeno debido a la falta de sangre y terminan muriendo. Por otra parte, los ictus hemorrágicos, que así se conocen a aquellos que se producen debido a un exceso de sangre, presentan un mayor riesgo de mortalidad, pero son menos frecuentes y las secuelas en los supervivientes son menos graves a medio plazo. Dentro de estos dos tipos existe una segunda subcategoría donde los tipos principales son los siguientes:

- Ictus hemodinámico. Es el más frecuente de los ictus isquémicos. En ellos, la causa del déficit de sangre es un descenso en la presión sanguínea. Este descenso puede ser debido, por ejemplo, a una hipotensión arterial grave, una arritmia o una parada cardíaca.
- Ictus trombótico o trombosis cerebral. Al igual que el anterior se trata de un ictus isquémico, y se produce cuando un trombo o coágulo de sangre bloquea el paso de sangre en una arteria importante, impidiendo que la sangre llegue a una parte del cerebro.
- Ictus embólico o embolia cerebral. También es causado por un trombo, por lo que al igual que la trombosis cerebral, se trata de un ictus isquémico. La diferencia con esta radica en que en este caso el trombo se ha originado lejos del lugar de la obstrucción, normalmente en el corazón.
- Hemorragia intracerebral. Se trata del ictus hemorrágico más frecuente, causado por la rotura de una arteria cerebral profunda. El resultado es presión y daño en el tejido cerebral cercano, debido a la salida del flujo sanguíneo. Esta presión también puede extenderse a la totalidad del encéfalo, poniendo en peligro la vida.
- Hemorragia subaracnoidea. Se produce entre la superficie del cerebro y la parte interna del cráneo como consecuencia de un aneurisma arterial, que es una pequeña protuberancia llena de sangre situada en la pared de una arteria.

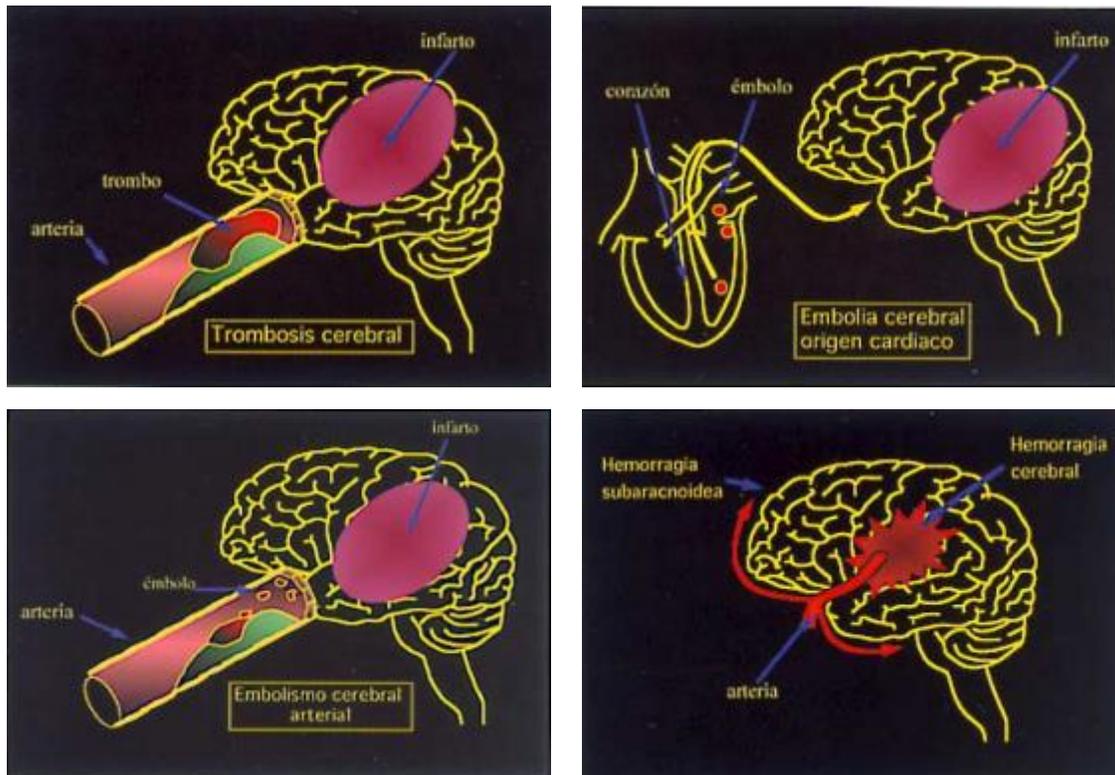


Figura 2.1: Distintos tipos de ictus

La probabilidad de padecer un ictus aumenta exponencialmente a partir de los 55 años, y el riesgo es mayor cuantas más enfermedades o hábitos de riesgo se tengan. Las patologías con riesgo de desembocar en un ictus son la hipertensión arterial, que es considerada a partir de los 139 mm Hg de sistólica u 89 mm Hg de diastólica; las arritmias cardíacas y otras enfermedades del corazón que favorecen la creación de coágulos; la diabetes mellitus, en la que el nivel de glucosa en sangre es superior a 120 mg/dl y produce daños en las arterias; y la dislipemia o exceso de “colesterol malo”, que se deposita en las arterias formando placas de ateroma. Por otra parte, los hábitos nocivos que también contribuyen a estos accidentes cerebrovasculares son el tabaquismo, el consumo excesivo de alcohol, la obesidad y el sedentarismo.

Sin embargo, aun en una persona joven y sana existe la posibilidad de padecer un ataque cerebral. Además, se ha demostrado que cuanto antes se acuda al hospital, menores son sus consecuencias. Por estos motivos es importante que la población en general conozca los síntomas que preceden a este tipo de ataques, que desafortunadamente son variados y casi siempre indoloros:

- Pérdida de fuerza de inicio brusco en la cara o en un lado del cuerpo (hemidistonía).
- Pérdida repentina de visión, ya sea total o parcial y en uno o los dos ojos.

- Alteración del habla, siendo difícil expresarse y articular el lenguaje.
- Dolor de cabeza repentino e inusualmente intenso.
- Vértigo, inestabilidad o caídas bruscas siempre y cuando vaya acompañado de alguno de los síntomas anteriores.

Estos síntomas no deben ser ignorados aunque su aparición sea repentina y de corta duración, pues precisamente en este caso podría tratarse de un Ataque Isquémico Transitorio, es decir, un “amago” de ictus que hasta en un tercio de los pacientes precede a un ictus establecido, con riesgo máximo en los días o semanas posteriores.

2.2. Secuelas

Las secuelas y complicaciones que pueden aparecer a raíz de haber sufrido un ictus se clasifican en dos tipos: psicológicas y físicas. Entre los problemas del primer tipo destacan las alteraciones del estado de ánimo durante la convalecencia y también a lo largo de la posterior rehabilitación. En particular, la depresión en el paciente es muy común, y aunque no suele ser necesario el uso de antidepresivos, sí que interfiere y ralentiza el proceso de rehabilitación. Del mismo modo, pueden aparecer:

- Ansiedad, con o sin ataques de pánico.
- Labilidad emocional, esto es, respuestas emocionales desproporcionadas o injustificadas, ya sean llantos o risas.
- Irritabilidad e incluso agresividad.
- Desmotivación.
- Falta de consciencia de las secuelas.

En ocasiones también es posible que se originen otras secuelas de carácter cognitivo, como son la disminución de la memoria, de la atención y de la orientación, así como dificultad a la hora de planificar tareas.

Por otra parte, los problemas físicos que puede acarrear un ictus son los siguientes:

- Afectación en el movimiento. Se puede manifestar como una pérdida de fuerza total (plejía o parálisis) o parcial (paresia) si se pueden efectuar movimientos pero con menor fuerza que la no afectada. La falta de coordinación y la pérdida de control también son otros dos problemas que pueden aparecer relacionados con el movimiento.
- Propensión a las caídas. Para evitarlas, es importante que el paciente realice los ejercicios pertinentes para fortalecer la musculatura y entrenar el equilibrio. Así mismo se deberán tomar medidas en el hogar, como por ejemplo usando zapatillas con suela antideslizante o retirando alfombras.

- Hemianopsia. Es una pérdida de visión de la mitad del campo visual de la cual el paciente no siempre es consciente. Por ello es importante que las personas de su entorno le recuerden que mire hacia la zona afectada.
- Trastorno en el lenguaje, ya sea en forma de afasia, disartria o mutismo. En el primer caso, el paciente no es capaz de comprender y/o de emitir el lenguaje de forma correcta. La disartria en cambio es una alteración del habla en la que existe cierta dificultad a la hora de articular las palabras. Por último, el mutismo se presenta cuando el paciente no es capaz de emitir palabra alguna.
- Trastornos de la sensibilidad: sensaciones poco agradables, hormigueos o insensibilidad al tacto. Suelen darse en el lado afectado del cuerpo.
- Espasticidad. Se define como la contracción permanente de un músculo y puede acabar ocasionando dolor, contracturas y dificultad en según qué movimientos.
- Dolor central. Se trata de un tipo de dolor superficial en el que se siente quemazón o pinchazos y que empeora al tocarlo, mojarlo, o simplemente con el movimiento. No suele ser muy común entre pacientes que han padecido un ictus, y en cualquier caso se puede tratar con antidepresivos y anticonvulsivantes.
- Dolor de hombro del brazo paralizado. Normalmente con la prescripción de algún analgésico simple es suficiente.
- Disfagia o dificultad a la hora de tragar. Por lo general, para combatir la disfagia suele ser suficiente con un cambio de dieta o con enseñar al paciente métodos seguros de alimentación, pero en los casos más graves es necesario el uso de una sonda nasal o gástrica, dependiendo de si se va a necesitar por un corto o largo periodo de tiempo, respectivamente.
- Incontinencia urinaria. Por lo general se trata de un efecto transitorio, a no ser que las secuelas del paciente sean graves.

2.3. Incidencia y rehabilitación

Una encuesta telefónica realizada en España por el Grupo de Estudio de Enfermedades Cerebrovasculares de la Sociedad Española de Neurología (GEECV-SEN) recoge que tan solo un 10% de la población conoce la naturaleza de esta enfermedad y las medidas para prevenirla. Estos porcentajes podrían hacer pensar que se trata de una patología rara y de baja incidencia en la población, pero la realidad es que es la primera causa de muerte en mujeres y la segunda en hombres, así como la primera causa de discapacidad grave y la segunda de demencia, solo por detrás del Alzheimer (Gil Núñez, A. *et al.*).

En nuestro país se dan cada año unos 100.000 casos nuevos de ictus, es decir, uno cada seis minutos y según los datos aportados por el Instituto Nacional de Estadística (INE) en el 2006, 32.887 de los afectados fallecieron (Álvarez Sabín, J., 2008). A pesar de este dato, España

presenta una de las menores tasas de mortalidad de Europa, siendo esta mayor en países como Suiza, Países Bajos, Dinamarca, Finlandia, Suecia, Noruega, Islandia o Irlanda.

En los últimos años, la tasa de mortalidad provocada por estos ataques se ha visto reducida en un 30% gracias a las Unidades de Ictus, que suponen un nuevo concepto asistencial en la atención médica de los pacientes en la fase aguda de esta enfermedad. Por lo general, estas unidades están presentes en hospitales generales y han sido desarrolladas por los servicios de neurología. Aglutinan a neurólogos, enfermeros y personal auxiliar especialmente entrenados para tratar con este tipo de pacientes, a la vez que cuentan con la colaboración de profesionales de otras especialidades como cardiólogos o rehabilitadores.

Desde el primer momento de hospitalización es muy importante empezar a trabajar en la prevención de la aparición de nuevos ictus, lo que se conoce como prevención secundaria. Estas medidas preventivas abarcan desde los medicamentos hasta el cambio de estilo de vida del paciente, y tan importante es que este tome las medicinas que le han sido recetadas, como que abandone los factores de riesgo, es decir, que deje de fumar, que adelgace en caso de obesidad, que abandone el sedentarismo, etc.

De particular importancia es un cambio en la dieta, reduciendo la ingesta de grasas saturadas y de sal en las comidas y limitando el consumo de alimentos grasos (charcutería, salchichas, leche entera, repostería...), a la vez que se debe aumentar la cantidad de alimentos ricos en fibra, frutas, verduras y legumbres. En definitiva, alcanzar un compromiso entre una dieta variada y el gusto del paciente para que este no acabe abandonándola.

En cuanto a las medidas de prevención farmacológicas, suelen ser aconsejables los antiagregantes plaquetarios, como puede ser una aspirina, y los anticoagulantes orales en pacientes con alto riesgo de sufrir embolias o con anomalías en la coagulación. De extrema importancia para un paciente bajo este tratamiento es tener especial cuidado para evitar caídas y golpes, pues los anticoagulantes incrementan el riesgo de sufrir una hemorragia.

Por desgracia y como ya hemos visto, tras un ictus no solo deben tomarse medidas preventivas para la aparición de más, sino que también se debe trabajar sobre las secuelas que dicho ictus ha provocado. Entre ellas es necesario destacar la alteración del lenguaje, dada la frecuencia con la que ocurre y lo chocante que suele resultar a los familiares la nueva situación. Por parte de estos se recomienda el uso de palabras sencillas, correctamente vocalizadas y entonadas. Del mismo modo, deben tener paciencia a la hora de dejar hablar al paciente, no interrumpiendo y prestando atención. Un paciente que no se sienta escuchado y respaldado en este aspecto puede desanimarse y caer en el mutismo.

Por otro lado, más del 30% de las personas que logran sobrevivir a un accidente cerebrovascular deben aprender a vivir con una discapacidad. Tanto la calidad de vida del paciente como la de su entorno cercano se ven bruscamente alteradas, ya que una persona que hasta entonces se valía por sí misma pasa a necesitar ayuda para llevar a cabo muchas de las

labores de su vida cotidiana. La magnitud de esta dependencia en cuanto a costes sanitarios, centros de apoyo, etc. queda de manifiesto sabiendo que el 8.5% de la población mayor de 65 años ha sufrido un ictus.

Para todas estas personas que presentan secuelas tras haber sobrevivido a un ataque cerebral, la rehabilitación es la parte más importante de su tratamiento. Generalmente suele empezarse ya desde los primeros días de estancia en el hospital, y continúa en régimen ambulatorio una vez el paciente ha sido dado de alta. Una rehabilitación adecuada es aquella que ayuda no solo físicamente al afectado, sino que también contribuye a la recuperación de su autoestima. Las cifras que se manejan tras un año de terapia rehabilitadora son las siguientes:

- El 33% de los afectados vuelve al trabajo que solía desempeñar.
- El 50% no sigue necesitando ayuda para realizar sus tareas diarias.
- El 20% necesita ayuda solo en determinadas situaciones.
- Otro 20% continúa siendo dependiente en mayor o menor grado.

Dentro de este primer año, cuando se aprecian mayores progresos es durante los dos primeros trimestres, periodo que se corresponde con la recuperación de la mayoría de los movimientos voluntarios. La restauración de otras habilidades como el lenguaje y el equilibrio suele prolongarse más a lo largo del tiempo, siendo posibles las mejoras hasta dos años después de haberse producido el ictus.

Según (Fung, J. *et al.*, 2011), los aspectos que más influyen en la eficacia del tratamiento son el tipo de rehabilitación hospitalaria junto con el momento de inicio, la intensidad y especificidad de la misma. En estos cuatro, diversos ensayos clínicos ponen de manifiesto que el mejor enfoque es aquel que presenta un tratamiento interdisciplinario que comienza en las primeras 24-48 horas, con sesiones de larga duración y dirigido a zonas específicas del cuerpo.

Históricamente, las distintas maneras de abordar cómo mejorar las secuelas físicas del paciente se han agrupado en técnicas de compensación, técnicas de facilitación y propuestas recientes, entre las que destaca el reaprendizaje motor orientado a tareas específicas. A continuación se explicará en qué consiste cada una de ellas.

2.3.1. Técnicas de compensación

Fueron las primeras en surgir y el objetivo es mejorar la función a la vez que se trata de aumentar la independencia en las tareas cotidianas. Esto se trabaja mediante el entrenamiento de las capacidades residuales, utilizando para ello el hemicuerpo no afectado. Existen detractores que sostienen que este tipo de técnicas de compensación podría dificultar la recuperación del lado afecto, aunque del mismo modo parece una terapia razonable en aquellos casos en los que la recuperación no es muy prometedora.

2.3.2. Técnicas de facilitación

Desde el año 1940 comienzan a aparecer técnicas que tratan de mejorar el movimiento del lado afecto, basándose para ello en la observación y el trato con los pacientes. Este tipo de terapias siguen siendo las más utilizadas hoy en día a pesar de que nunca se haya demostrado que sean superiores a otros enfoques. Existen cuatro tipos de técnicas de facilitación:

- Método Bobath. Desarrollado mediante ensayo-error por Berta Bobath, se basa en que el aumento del tono muscular y de la actividad refleja se deben a una falta de inhibición de un mecanismo reflejo postural dañado. Sus tres pilares fundamentales son disminuir la espasticidad, desarrollar patrones de postura y de movimiento, e incorporar el lado hemipléjico en los ejercicios para integrarlo en los movimientos.
- Método Brunnstrom. Signe Brunnstrom observó el efecto que tenían los estímulos externos y los cambios de posición de las articulaciones en los pacientes que sufrían de hemiparesia espástica, por lo que propuso emplear estímulos externos para provocar aquellos movimientos que el paciente no era capaz de realizar de manera voluntaria.
- FNP (Facilitación Neuromuscular Propioceptiva). Se basa en usar estímulos como el tacto o el estiramiento de músculos y tendones para aumentar la fuerza y la coordinación muscular, integrando al músculo débil dentro del conjunto en el que trabaja. Según el objetivo que se pretenda lograr existen diferentes tipos de ejercicios: equilibrio, coordinación, potenciación muscular y relajación.

2.3.3. Técnicas modernas

Hasta la década de 1980 no comenzaron a aparecer nuevas técnicas en la rehabilitación de pacientes afectados por un ictus, entre las que destaca el aprendizaje motor orientado a tareas, que se basa en un principio bastante simple: una persona aprende aquello que practica. De este modo, esta terapia pretende enseñar al paciente las estrategias necesarias para lograr un movimiento funcional del lado afectado a través de cinco puntos: instrucciones verbales simples, demostraciones visuales, guía manual que se ha de reducir paulatinamente, *feedback* positivo y práctica repetitiva. A raíz de la técnica del aprendizaje motor orientado a tareas han surgido otras más específicas, entre las que destacan las siguientes:

- Marcha sobre cinta rodante con suspensión parcial del peso corporal. Su efectividad ha sido demostrada en distintos ensayos clínicos y es considerada como uno de los principales avances en la rehabilitación del ictus.

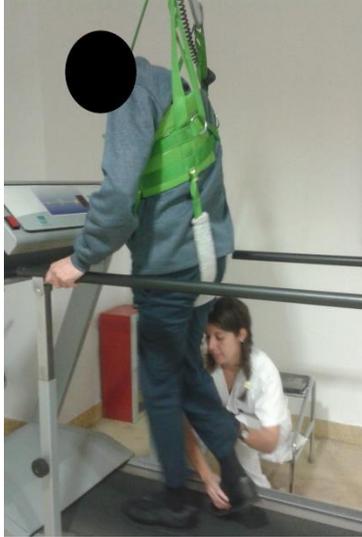


Figura 2.2: Entrenamiento con cinta rodante

- Movimiento inducido mediante la restricción del lado sano. Trata la extremidad superior afectada a través de tareas cuya dificultad aumenta progresivamente, reforzando positivamente al paciente cada vez que se logra un objetivo. Se ha demostrado que esta técnica induce una reorganización cortical en el cerebro del paciente. Sin embargo, presenta el inconveniente de solo poder ser aplicada en personas que sufren una parálisis leve y carecen de problemas cognitivos graves.
- Programas de fortalecimiento muscular y reacondicionamiento físico. A pesar de que la debilidad muscular es uno de los síntomas generales del ictus, los enfoques tradicionales no le prestan mucha atención. Esto cambió con las técnicas modernas, llegando incluso a demostrar la posibilidad de aumentar la fuerza y la capacidad funcional del paciente con ejercicios de musculación contrarresistencia sin que estos tengan efectos negativos.
- Estimulación sensitivomotora asistida con robots. Hace casi 20 años un grupo de investigadores del Hospital de Rehabilitación de Burke junto con el Instituto Tecnológico de Massachusetts crearon un robot que pretendía la reeducación del miembro superior de los pacientes a través de movimientos activos de codo y hombro personalizados para cada uno, además de refuerzos positivos visuales y acústicos. Al menos dos estudios han demostrado la eficacia de añadir una hora al día de este tipo de terapia, puesto que mejora la recuperación motora de los músculos reentrenados. Además, cuenta con otra serie de ventajas añadidas, como la posibilidad de que el propio robot analice la función motora del paciente, el aumento de la dificultad según el rendimiento del paciente, y que no sea necesaria una supervisión directa por parte del personal sanitario.

2.3.4. Recuperación del equilibrio

La pérdida del equilibrio ha sido reconocido como un gran problema de salud en las personas que han sufrido un ictus, siendo la responsable de muchas caídas tanto durante la rehabilitación como tras ella, particularmente en aquellos pacientes con déficits motores y sensoriales. Por este motivo, es importante que la recuperación del equilibrio sea tan rápida y completa como sea posible. Sin embargo, no se ha probado la superioridad de ninguna terapia tradicional en concreto, así como tampoco existen evidencias de la efectividad de otros métodos como la acupuntura o la estimulación eléctrica (Alexander C.H. *et al.*, 2005).

Para mejorar la velocidad y el alcance de la recuperación es necesario conocer completamente los mecanismos que subyacen tanto la recuperación natural del equilibrio como los mecanismos de compensación. Hasta el momento, aunque muchos estudios han identificado diferentes aspectos fisiopatológicos en el control del equilibrio en pacientes tras un accidente cerebral, pocos de ellos han aportado información acerca de cuál de esos aspectos es posible mejorar tras la fase crítica de la rehabilitación. La mayoría de estos estudios se han centrado en pacientes con secuelas bastantes graves, que aunque posiblemente sean los que requieran mayor esfuerzo y cuidados por parte de los profesionales, poco se puede decir sobre la mejoría en su equilibrio. Esto mismo es aplicable para pacientes que presenten secuelas a ambos lados del cuerpo, puesto que necesitarán un control sobre el tronco mucho mayor que aquellos en los que solo un lado de su cuerpo se ha visto afectado.

Incluso en pacientes adecuados para el estudio del equilibrio, la recuperación puede ser variable dependiendo de sus déficits iniciales. Aun así, la mayoría de estas personas ve mejorada su estabilidad en ambos planos, su habilidad para compensar perturbaciones tanto internas como externas, y su poder para controlar voluntariamente su postura. La conclusión más notable desde el punto de vista de la plasticidad neuronal es que la recuperación más sustancial del equilibrio se da también, al menos durante los tres primeros meses, en pacientes que no presentan signos de mejora en las funciones de soporte de la pierna afectada. Esta conclusión la corrobora el hecho de que la mayoría de los estudios que están investigando la relación entre el equilibrio estático o dinámico y la espasticidad o la fuerza del músculo, no están obteniendo datos significativos.

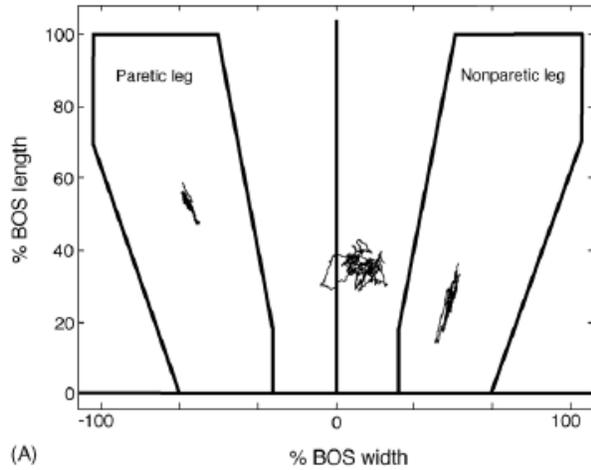
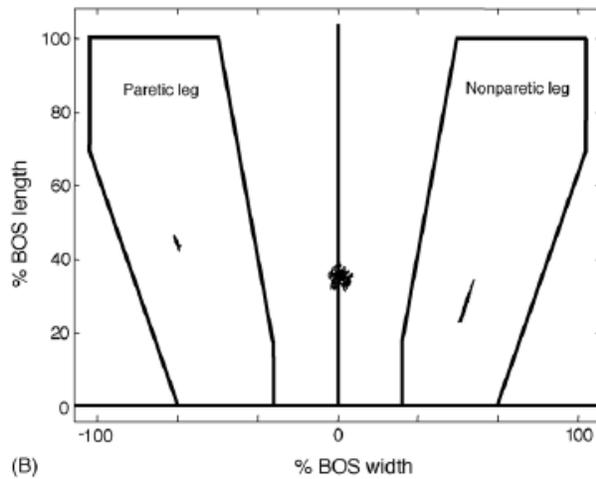


Figura 2.3: Trayectoria del centro de masas durante 30 segundos con los ojos abiertos en un hombre de 48 años afectado por un infarto cerebral en el hemisferio derecho.

(A) Un mes después de haber sufrido el ictus. Se aprecia una clara desviación hacia la derecha del centro de masas.

(B) Medio año después de haber sufrido el ictus. La asimetría ha desaparecido, aunque se sigue sobrecargando la pierna sana.

Imágenes extraídas de (Alexander C.H. *et al.*, 2005).



2.4. Realidad virtual en rehabilitación

Debido al carácter repetitivo y sistemático que presenta la rehabilitación tradicional, el paciente tiende a no tener en cuenta la mente en el proceso de recuperación, perdiendo así la motivación. Además, en la parte correspondiente a la labor del terapeuta, se hace necesaria la existencia de herramientas de análisis específicas que permitan analizar el movimiento y a su vez sean portables, de fácil acceso y con recolección de datos en tiempo real.

Un buen modo de incentivar al paciente es a través de la Realidad Virtual (RV) y/o la Realidad Aumentada (RA). La Realidad Virtual se define como un entorno de apariencia real que ha sido creado mediante el uso de la tecnología y permite al usuario la inmersión en el mismo; mientras que la Realidad Aumentada combina los escenarios del mundo real con elementos virtuales que se visualizan a través de un dispositivo, creando una realidad mixta en tiempo real.

La ventaja principal de la RV es que permite desarrollar entornos simulados de rehabilitación sobre los que el usuario interactúa de forma natural, viviendo la experiencia como si ocurriera en un entorno real (Rose *et al.*, 2005). El uso de escenarios virtuales en el ámbito de la rehabilitación pretende la mejora de las capacidades físicas, psicológicas y de adaptación social de los pacientes, teniendo como objetivo lograr un nivel óptimo de integración social. En este sentido, los ambientes virtuales han sido empleados mayoritariamente como métodos de entrenamiento de habilidades para la vida independiente y como formas de evaluación de la adquisición de tales habilidades (Pérez Martínez, 2011). Los estudios en este ámbito han demostrado que el aprendizaje adquirido por esta vía, puede ser transferido a las situaciones de la vida real en donde se requiere de tales capacidades.

A continuación se describirán los métodos planteados y los resultados obtenidos en algunos de estos estudios.

2.4.1. Entrenamiento en cinta con realidad virtual

En un estudio llevado a cabo durante el año 2011 en Taiwán (Yang, S. *et al.*, 2011), 14 pacientes fueron reclutados y aleatoriamente asignados para recibir bien entrenamiento tradicional sobre cinta, o bien incluyendo RV al mismo. El sistema de realidad virtual que se incluyó contaba con un *software* que contaba con diferentes escenas, como recorrer un parque a través de un camino o realizar actividades del hogar.

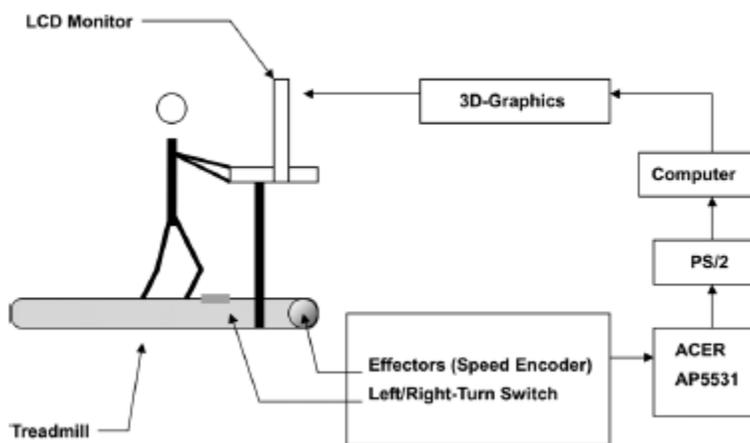


Figura 2.4: Elementos del sistema empleado en el estudio

Las conclusiones que obtuvo este estudio fueron que ni el método tradicional ni el que hacía uso de RV influían en la mejora de la postura estática. Sin embargo, el entrenamiento en cinta con RV demostró ser más efectivo que el entrenamiento tradicional a la hora de mejorar el equilibrio tanto en la dirección medial-lateral como al incorporarse estando sentado. Además, mejoró la participación del lado afecto al andar.

Desafortunadamente, solo se observaron progresos significativos en alguno de los distintos parámetros que cuantifican la capacidad para conservar el equilibrio.

Ese mismo año un sistema similar fue presentado en la 33ª Conferencia Internacional Anual de IEEE EMBS (Fung, J. *et al.*, 2011). La principal diferencia radica en que en este último, la cinta reposa sobre una plataforma gobernada por seis actuadores que permiten simular cambios de pendiente con seis grados de libertad. Por otro lado, también fue instalada una barra con el fin de proporcionar información somatosensorial a través del contacto con la yema de los dedos. En concreto, esta se encargaba de medir la presión que sobre ella ejercía la yema del dedo índice. La conclusión obtenida a lo largo de este estudio fue la influencia que un leve apoyo con la punta de los dedos ejerce sobre el equilibrio y la estabilidad al andar.



Figura 2.5: Barra instalada en el lateral de la cinta

2.4.2. Entrenamiento con sistemas de *tracking*

Otra manera de introducir la realidad virtual en una rehabilitación de estas características es a través de los sistemas de *tracking* o de seguimiento. Estos sistemas pueden ser ópticos, electromagnéticos, mediante sensores de profundidad, etc. Un ejemplo de ello es el estudio realizado en Valencia durante el año 2011 (Lloréns, R. *et al.*, 2013), que hizo uso de un sistema de seguimiento óptico formado por tres cámaras que detectaban la posición del usuario gracias a unas marcas reflectantes colocadas en posiciones concretas de su cuerpo en función de la tarea a realizar.

Los ejercicios que los pacientes debían realizar durante la sesión contaban con una dificultad regulable y consistían en alcanzar los ítems que aparecían a su alrededor, ya fuera en posición de sedestación o de bipedestación, siendo esta última postura la elegida cuando lo que se deseaba era entrenar el equilibrio estático y dinámico, el control postural y las transferencias.

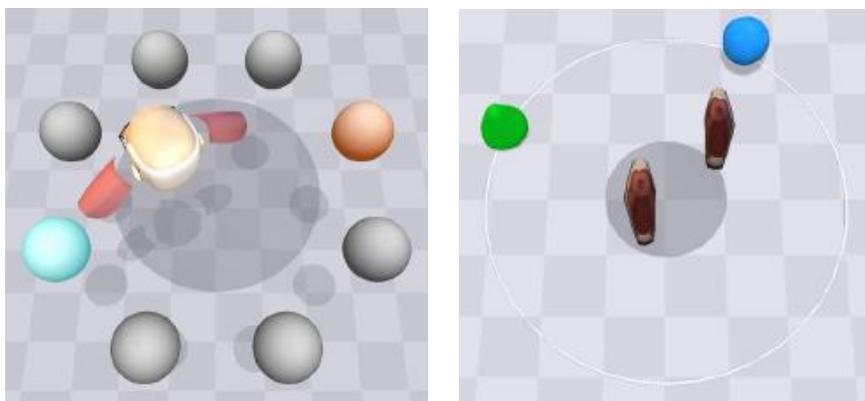


Figura 2.6: Alcance mediante hombros en sedestación y alcance mediante pies en bipedestación, respectivamente

De acuerdo con los resultados, se obtuvo una mejora significativa de las puntuaciones en dos test que evalúan el equilibrio estático y dinámico del paciente, la BBS (*Berg Balance Scale*) y la POMA (*Tinetti Performance-Oriented Mobility Assessment*), así como una mejora en el índice vestibular y en el control antero-posterior. Clínicamente, tres de los cuatro pacientes que al inicio del tratamiento empleaban un bastón cuadrúpode para una marcha segura, consiguieron una marcha sin ayudas.

2.4.3. Entrenamiento a través de juegos serios

Los juegos serios o *serious games* presentan propósitos en materia de salud cuidadosamente planeados en los que se utiliza el entretenimiento como herramienta, no como fin. Varios son los estudios que se han llevado a cabo haciendo uso de productos presentes en el mercado y de sobra conocidos por el público en general (Deutsch, J.E. *et al.*, 2011; Ki Hun Cho *et al.*, 2012; Bower, K.J. *et al.*, 2014). Quizás a la que más se ha recurrido es a la consola Nintendo Wii, y más concretamente a su accesorio Wii Balance Board, una tabla capaz de calcular la presión que es ejercida sobre ella y que en el ámbito de la rehabilitación se ha empleado para trabajar el equilibrio y la transferencia de pesos a través de ejercicios tales como yoga o ski slalom.



Figura 2.7: Ejemplo de juego que hace uso de la tabla Wii Balance: *We Ski*
Fuente: <http://img5.juegos.es/wp-content/uploads/Wii-Balance-Board-juegos.jpg>

Algunos de los beneficios reportados por este tipo de terapia complementaria son los siguientes (López, D., 2014):

- A los pacientes en condición de discapacidad les es posible realizar procesos de aprendizaje motor dentro de ambientes virtuales.
- Las frustraciones del paciente, derivadas de los entrenamientos in situ de estas habilidades, pueden ser evitadas por medio de entornos simulados a través del uso de RV y RA.
- Aquellos movimientos aprendidos por los pacientes a lo largo de las dinámicas de los videojuegos, pueden ser transferidas a la vida cotidiana.
- La rehabilitación convencional complementada con terapia que hace uso de videojuegos es más eficiente que si solo se aplica la primera de ellas.
- Hasta la fecha, no se han encontrado desventajas en las personas que han sido expuestas a este tipo de terapias.

Sin embargo, se notaron ciertos problemas que pusieron de manifiesto que si bien los juegos serios comerciales como *Wii Fit Plus* pueden ser útiles en la rehabilitación, también presentan inconvenientes al estar pensados para el público en general y no para personas con necesidades especiales. Algunos de estos puntos débiles son la limitación a la hora de ajustar la dificultad del juego y por tanto, falta de *feedback* positivo al tener que alcanzar objetivos poco realistas; la necesidad de adoptar posturas que no permiten ver la información mostrada por pantalla; o el sobreesfuerzo y consiguiente dolor generados al realizar movimientos pensados para un usuario sano.

Por tanto, parece lógico pensar en la necesidad de que todo juego serio que vaya a ser utilizado como complemento en las terapias de rehabilitación, sea desarrollado específicamente para este fin y bajo la supervisión de terapeutas especializados. Solo de este modo se conseguirá un juego que abarque las necesidades de los pacientes y vaya más allá del entretenimiento o la diversión.

Capítulo 3. Videojuegos

El presente capítulo tiene como objetivo introducir al lector en el mundo de los videojuegos, partiendo de sus inicios hasta los sofisticados videojuegos existentes en la actualidad. Tras esto, se describirán brevemente los distintos tipos de videojuegos existentes para posteriormente reflejar los datos más significativos del estudio de mercado *European Consumer Study*, llevado a cabo en noviembre de 2012. También se explicará qué es la gamificación y cómo puede ayudar al paciente a realizar su rehabilitación de un modo más ameno, para acabar centrándonos en las terapias de Realidad Virtual y Realidad Aumentada.

3.1. Historia de los videojuegos

Comúnmente, el *Pong* es considerado el primer videojuego de la historia. Sin embargo, antes de la aparición de este mítico juego surgieron otros que aunque no resultaran tan exitosos, también tienen relevancia dentro de la evolución de los videojuegos. Por ejemplo, en 1947 fue patentado un sistema electrónico de juego que consistía en simular el lanzamiento de misiles contra un objetivo haciendo uso para ello de válvulas y una pantalla de rayos catódicos. Más tarde, en 1952, surgió el *Tres en Raya* en versión digital. Funcionaba en una computadora ESDAC y el jugador interactuaba con la pantalla a través de un dial telefónico de rueda.

No obstante, ninguno de estos dos juegos presentaba movimiento en pantalla, tan solo imágenes fijas, por lo que no se les considera videojuegos sino juegos gráficos por ordenador. Al que muchos sí consideran el primer videojuego es a *Tennis for Two*, ideado en 1958 por el físico William Higinbotham. El juego se basa en un programa de cálculo de trayectorias utilizado por el ejército americano en el que William utilizó un osciloscopio como pantalla, y consistía en una línea horizontal que representaba el campo de tenis, y una pequeña línea vertical que hacía las veces de red. Los jugadores debían elegir el ángulo en el que saldría la pelota y golpearla.

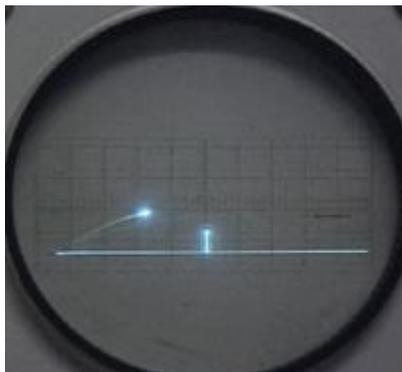


Figura 3.1: Tennis for Two de William Higinbotham.
Fuente: <http://www.gamersquarter.com/tennisfortwo>

Pero también existen detractores que opinan que *Tennis for Two* no puede ser considerado el primer videojuego ya que hace uso de circuitería en lugar de un ordenador. Para ellos, *Spacewar!* es el primer juego en cumplir los requisitos. Consiste en dos naves, cada una controlada por un juego de interruptores de consola, que tienen un depósito de combustible y un arma. Al escenario se le añadieron ciertos riesgos, como que la nave acabara quemada por el sol debido a un mal cálculo de giro o al uso de la opción “hiperespacio”, que hacía aparecer la nave en un lugar aleatorio de la pantalla.

Por otra parte, en 1966 se empezó a desarrollar un proyecto de videojuego llamado *Fox and Hounds* que supuso el inicio de los videojuegos domésticos. Esta idea evolucionaría hasta resultar en 1972 en *Magnavox Odyssey*, el primer sistema doméstico de videojuegos que permitía jugar a varios juegos pregrabados conectándolo a la televisión. Pocos años después, entre 1970 y 1979, se produjo la eclosión de los videojuegos, con la comercialización de *Computer Space*, una versión de *Spacewar!*; la aparición de la máquina recreativa *Pong*, diseñada en la recién fundada Atari; y la aparición en salones recreativos de juegos como *Space Invaders* o *Asteroids*.

La década de los 80 comenzó con un fuerte crecimiento del sector, gracias a la popularidad de la que gozaban los salones de máquinas recreativas y las primeras consolas aparecidas durante la década de los 70. En estos años Nintendo empezaba a comercializar sus primeras *Game & Watch*, así como el juego *Donkey Kong*. También llegaron al mercado numerosas consolas nuevas, como la Mattel Intellivision, Atari 5200 y Colecovision; y tantos otros ordenadores (Commodore 64, MSX, ZX Spectrum...). Desgraciadamente, toda esta variedad se tradujo en la saturación del mercado, que ya contaba con miles de juegos y decenas de consolas. Los comercios se vieron obligados a bajar drásticamente los precios para poder dar salida a todo el *stock* acumulado, dando lugar a una gran crisis en la industria del videojuego.

No sería hasta 1985 cuando la crisis empezara a desaparecer. Este fue el año en el que Nintendo lanzó *Super Mario Bros*, que llegó a vender 10 millones de copias y supuso un antes y un después en el mundo de los videojuegos. La mayoría de sus predecesores solo contenían unas pocas pantallas repetidas en bucle, con el único objetivo de conseguir la puntuación más alta posible, pero *Super Mario Bros* era un juego totalmente revolucionario en el que por primera vez existía un objetivo y un final en un videojuego. Durante los años próximos también aparecieron otros títulos míticos como el *Tetris*, *Legend of Zelda*, *Metroid*, *Arkanoid*, etc., además de consolas como *Master System* o *Mega Drive*.

Otra rama que creció con fuerza fue la de los videojuegos portátiles. Se podría decir que todo comenzó a principios de los años 70 con los primeros juegos totalmente electrónicos que Mattel lanzó al mercado, y que fueron haciéndose más populares gracias a sus versiones en máquinas recreativas. Pero sin ninguna duda, la evolución definitiva de los videojuegos portátiles llegó en 1989 de la mano del lanzamiento de la *Game Boy* de Nintendo.

En un intento de hacerse con los adeptos de la *Game Boy*, Sega sacó al mercado su consola portátil *Game Gear* en 1990. No obstante, las limitaciones en cuanto a su tamaño y a la corta duración de su batería produjeron que nunca pudiera plantarle cara a la portátil de Nintendo. Tampoco tuvieron suerte con el lanzamiento del *Mega CD* en 1991, un *add-on* que pretendía aprovechar las capacidades de CD para la *Mega Drive*, y que desapareció poco a poco debido a la falta de apoyo por parte de la compañía y de los escasos juegos que aparecieron en este formato. Sin embargo, no todo iban a ser malas noticias para Sega: el éxito llegó ese mismo año con el mítico *Sonic*, que llegó a vender más de 4 millones de copias y a ser el personaje más querido en Estados Unidos (Anon., 2014).

Los primeros años de esta década de los 90, las videoconsolas experimentaron un gran avance técnico gracias a la competición de la llamada “generación de 16 bits”, formada por la *Super Nintendo Entertainment* de Nintendo, la *Mega Drive* de SEGA, la *Turbografx* de NEC y la *CPS Changer* de Capcom.

Rápidamente los videojuegos 3D fueron abriéndose paso en el mercado, gracias a las nuevas generaciones de 32 y 64 bits. Ejemplo de estas generaciones son *Sony PlayStation* y *Sega Saturn* por un lado, y *Nintendo 64* y *Atari Jaguar* por el otro. En particular, la consola de Sony apareció tras la cancelación del proyecto *SNES PlayStation* iniciado con Nintendo, que consistía en un periférico para *SNES* con lector de CD. Finalmente, Nintendo rechazó esta propuesta al ver el poco éxito obtenido por Sega con una propuesta similar, y Sony lanzó de manera independiente su *PlayStation*, que a finales de los 90 se convirtió en la consola más popular con juegos como *Final Fantasy VII* o *Resident Evil*. Durante esa misma época, en 1998, Sega anuncia la salida de una nueva consola: *Dreamcast*. Esta consola fue la respuesta de la compañía a su fracaso con *Saturn*, y a pesar de ser la precursora de la “generación de los 128 bits”, no pudo con la gran rival que estaba a punto de salir, *PlayStation 2*, y supuso la retirada de Sega del mercado de las consolas de sobremesa.



Figura 3.2: Imagen del videojuego *Final Fantasy VII* para *PlayStation*. Fuente: <http://www.gamedev.net/topic/666934-pixel-art-sprites-an-overused-art-style>

Ya en el año 2000 aparece la consola de sobremesa más vendida de toda la historia: *PlayStation 2*, así como la *PsOne* de Sony. Un año más tarde también Microsoft entra en la industria de las consolas mediante la creación de la *Xbox*. Años más tarde, en el 2005, empieza lo que se conoce como la *Next-gen*, con la aparición de la *Xbox 360*, *PlayStation 3* y *Wii*. Todas ellas han sido criticadas por diferentes motivos: continuos fallos en el *hardware* de la consola, escasez de juegos junto con un elevado precio, y pobre gráficos, respectivamente.

En cuanto a los PCs, los juegos más populares estos años eran los de acción en primera persona y los de estrategia en tiempo real. Además, las conexiones entre ordenadores a través de Internet facilitaron el juego multijugador, que se convirtió en la opción preferida por la mayoría de los usuarios y dio lugar al nacimiento de los juegos de rol multijugador *on-line*. Es necesario destacar que el ordenador personal es la plataforma más cara de juegos pero también la que permite mayor flexibilidad, al poder añadir periféricos o componentes al mismo que mejoran constantemente, como por ejemplo tarjetas gráficas o de sonido, mandos, pedales, cámaras, etc. Además presentan la posibilidad de actualizar los juegos con parches oficiales o con nuevas características.

3.1.1. Steam

En el año 2003, la empresa estadounidense Valve Corporation lanzó al mercado Steam, una plataforma de distribución digital, gestión digital de derechos, comunicaciones y servicios multijugador. Se utiliza para la distribución de videojuegos y la de material relacionado tanto por grandes empresas como por desarrolladores independientes, además de ofrecer diferentes métodos de contacto entre los miembros de la comunidad.

Para poder hacer uso de Steam, es necesario registrarse previamente de manera gratuita. Una vez realizado el registro, todos los juegos comprados por el usuario quedarán vinculados a esa cuenta.

El primer juego en salir con Steam fue *Counter Strike 1.6*, aunque ya podía descargarse por Internet. Si se tuviera algún juego de Valve lanzado con anterioridad, este puede ser registrado en Steam con tan solo introducir su clave en la cuenta de usuario. De hecho, en un primer momento tan solo se podían ejecutar juegos de Valve, aunque a medida que la plataforma fue creciendo, se llegaron a acuerdos para la distribución de videojuegos de otras empresas.

Hoy en día no solo se pueden descargar juegos a través de esta plataforma, sino que también están disponibles herramientas como el Source SDK, que permite diseñar y modificar juegos y mapas, y programas para crear servidores dedicados.

En cuanto al número de usuarios registrados, en febrero de 2015 Steam alcanzó la cifra de 125 millones en todo el mundo, con picos de hasta 8.9 millones de usuarios conectados a la vez. Además, en la actualidad Steam cuenta con más de 4500 juegos y 400 millones de contenidos generados por los usuarios (Saed S., 2015).

3.2. Clasificación de los videojuegos.

Existen múltiples clasificaciones diferentes de videojuegos, según se atienda a su representación gráfica, tipo de interacción entre jugador y máquina, ambientación, sistema de juego, etc. A su vez, cada vez es más común que un videojuego integre distintos elementos pertenecientes a una misma clasificación, por lo que etiquetarlos cada vez es más complicado. A continuación se presenta una lista de los diferentes tipos de videojuegos existentes según su sistema de juego. Esta clasificación no debe tomarse en ningún caso como única ni inalterable, ya que existen múltiples listas diferentes aun atendiendo a un mismo elemento como es el sistema de juego. Así, se pueden encontrar otras ordenaciones en las que aparecen desde géneros clásicos como los videojuegos de cartas o de tablero, hasta otros más recientes como el *Survival Horror* o el *Cyberpunk*, dedicados respectivamente a las aventuras de terror y a la ciencia ficción futurista.

3.2.1. Acción.

Un videojuego de acción es aquel en el que el usuario hace uso de destreza, velocidad y tiempo de reacción. Este género es el más amplio dentro de esta ordenación, englobando múltiples subgéneros. Algunos de ellos son:

- *Beat 'em up*, o videojuegos de lucha a progresión. En estos videojuegos, los jugadores deben combatir contra un gran número de enemigos a la vez que progresan a lo largo de distintos niveles. Suelen ser multijugador para facilitar el avance. Algunos ejemplos son *The Warriors*, *Double Dragon* o *Battletoads*.
- Lucha. En estos se recrean combates entre avatares que pueden estar controlados tanto por un usuario como por la máquina. Generalmente, el usuario cuenta con una perspectiva lateral de los avatares, quienes suelen recrear artes marciales, boxeo o lucha libre, e incluso en ocasiones hacen uso de armas blancas. Sin duda la saga más conocida de este género es *Street Fighter*.
- Plataformas. Fueron de los primeros tipos de juegos que aparecieron en los ordenadores, y sin duda alguna, el lanzamiento de *Super Mario 64* con su sistema de cambio de perspectiva supuso una gran revolución para el género. En estos videojuegos el usuario controla un avatar que debe avanzar por un escenario a la vez que evita obstáculos y se enfrenta contra enemigos. Inicialmente, la acción transcurría a lo largo de niveles con un desarrollo horizontal. Sin embargo, tras la llegada de los gráficos 3D, este desarrollo puede ser en todas direcciones.
- Disparos. En muchas clasificaciones, este subgénero aparece como un género por sí mismo debido a la variedad existente de juegos de este tipo. Dentro de los videojuegos de disparos, nos encontramos con una segunda clasificación:
 - *First Person Shooter* (FPS) o de disparos en primera persona. En este tipo de videojuegos, la perspectiva facilita la impresión de que el usuario es el propio

personaje. Suelen predominar la jugabilidad y la calidad gráfica sobre los guiones trabajados, e impone al jugador unos buenos reflejos y precisión. Ejemplos: *Halo*, *Call of Duty*.

- *Third Person Shooter* (TPS). Se basan en alternar disparos y pelea/interacción con el entorno. El avatar es visto desde atrás y en ocasiones, desde una representación isométrica. Estos videojuegos antepone la libertad de movimientos a la precisión, e incluso se considera que ha surgido un nuevo subgénero en el que las posibilidades del personaje son prácticamente infinitas. El mayor ejemplo de este tipo es toda la saga *Grand Theft Auto*.
- *Shoot 'em up*. Este género presenta un uso continuo de un arma que comúnmente puede ser elegida entre varias disponibles, a la vez que estas pueden mejorarse a medida que se avanza en el juego. La acción suele desarrollarse en 2D aunque se incluyan elementos 3D para conseguir mayor efecto visual. Los exponentes más representativos de esta clase de videojuegos son *Gradius* y *r-Type*.
- Sigilo. Se trata de un género relativamente reciente que se basa en el sigilo, la estrategia y la furtividad en lugar de la confrontación directa. En ocasiones también pueden aparecer clasificados como juegos de estrategia. Algunos ejemplos: *Metal Gear*, *Beyond Good & Evil* y *Hitman*.

3.2.2. Simulación.

Se caracterizan por llevar a un videojuego algún aspecto de la vida real, teniendo así total control sobre lo que ocurre. Mientras que muchos toman una situación y dejan al jugador explorar las diferentes opciones, existen otros enfocados en hacer creer al usuario que lo que está pasando es real, como en las simulaciones de pilotaje. También en este género nos encontramos con una segunda clasificación:

- Simulación musical. Pueden ser de tipo karaoke, baile o con instrumentos musicales. Este tipo de videojuegos se caracteriza por la necesidad de tener accesorios especiales compatibles bien con las consolas o con el ordenador, y por contar con modo multijugador, ya que el objetivo suele ser conseguir la mayor exactitud posible en la interpretación para así obtener más puntos que el rival. Entre todos los juegos existentes dentro de esta categoría, destacaremos *Dance Central* para Xbox 360, que hace uso de la tecnología Kinect para captar los movimientos de los usuarios.
- Simulación de combate. Se caracterizan por el gran realismo que presentan, puesto que el movimiento de los avatares o el comportamiento del armamento tratan de ser completamente realistas. Destaca *Operation Flashpoint*, que cuenta con una modificación destinada al entrenamiento táctico de cuerpos de élite en Estados Unidos y Australia.



Figura 3.3: Imagen del videojuego Operation Flashpoint. Fuente: <http://www.taringa.net/comunidades/gamerscompulsivos/1972155/Aporte-Operation-Flashpoint-Dragon-Rising-Pc-MF-Espanol.html>

- Simulación de construcción. El programa proporciona al jugador todo lo necesario para construir un proyecto, en el que se consideran distintos factores como los gastos de construcción y mantenimiento, la física de los materiales, clima de la zona, etc. Probablemente el mejor ejemplo de este género sea *Age of Empires*.
- Simulación de vida. Este tipo de videojuegos se enfocan en controlar todos los aspectos de la vida de un avatar, al que se le ha dotado de capacidades y emociones humanas. El realismo que caracteriza este género provoca que muchas personas se tomen este tipo de juegos como prolongación de su propia vida, llevando a la ficción todo lo que les gustaría experimentar en su vida real, pero sin los riesgos que esto podría suponer. Sin duda, el juego más popular de este género es *The Sims*, que cuenta con numerosas expansiones.

3.2.3. Arcade

Alcanzaron su auge durante la década de 1980 y se caracterizan por la simplicidad de acción rápida de jugabilidad. No es necesaria una historia, sino que se trata de juegos largos o repetitivos. Ejemplos de juegos pertenecientes al género de arcade son *Space Invaders*, *Asteroids* o *Pac-Man*, entre otros.

3.2.4. Deporte

Simulan o imitan deportes reales tales como fútbol, golf, tenis, baloncesto, etc. Algunos títulos son las series *FIFA*, *NBA* o *Wii Sport*, así como *Deca Sports Freedom* de Xbox Kinect.

3.2.5. Carreras

Consisten en recorrer un circuito antes que los rivales, pudiendo incluir herramientas o trampas para batir a los contrincantes. Dentro de este género también se encuentran los

simuladores de carreras, que representan con exactitud circuitos de la actualidad. Ejemplos de estos últimos son *GTR* o *Forza Motorsport*.

3.2.6. Agilidad mental

Pretenden que el usuario entrene su mente para así agilizar su pensamiento a través de la resolución de ejercicios que van haciéndose más difíciles de manera progresiva. Algunos ejemplos son *Brain Age* o *Brain Academy*.

3.2.7. Educación

Aunque tradicionalmente han estado enfocados a los niños, se conoce como juego educativo a todo aquel que enseña mientras promueve diversión o entretenimiento.

3.2.8. Aventura

Este tipo de videojuegos fueron de los primeros en lanzarse al mercado y tuvieron su auge desde los años 80 hasta mediados de los 90. En ellos, el usuario encarga a un personaje que debe resolver incógnitas. En un principio, el jugador introducía las órdenes a través del teclado, como por ejemplo “ir al norte”. Más tarde, estas aventuras textuales dieron paso a las visuales, aunque se seguía haciendo uso de textos de manera introductoria. En estas aventuras gráficas, también llamadas de tipo *Point and click* se emplea el ratón en lugar de la introducir los comandos mediante teclado. Algunos títulos pertenecientes a este género son *Day of the Tentacle*, o las series *King’s Quest*, *Broken Sword* y *Leisure Suit Larry*.

3.2.9. Rol

Son conocidos por las siglas RPG (*Role-Playing Game*) y se caracterizan por presentar una historia profunda a lo largo de la cual el personaje evoluciona. Normalmente, la evolución es posible gracias a la interacción con nuevos personajes y la exploración del entorno que le rodea, donde puede conseguir aliados, armas o conocimientos de magia. En los RPG clásicos, las batallas se establecen por turnos: el usuario ataca a su contrincante haciendo uso de la sabiduría alcanzada por el personaje, y después debe esperar a recibir la respuesta en forma de contraataque bien por parte de otro jugador o del ordenador. En la actualidad, la mayoría de los videojuegos de rol ya no siguen este procedimiento, sino que se hace uso del combate en tiempo real, atacando tanto jugador como enemigo al mismo tiempo.

Dentro de esta categoría destacan los MMORPG (*Massive Multiplayer Online RPG*), un tipo de juego RPG en el que cada usuario crea un avatar y a través de una conexión a Internet, se adentra en un mundo donde debe interactuar con otros personajes diseñados a su vez por otros jugadores también conectados vía Internet. Este tipo de juegos cuenta con un componente de adicción muy amplio, además de la cultura popular que se suele generar en torno a ellos. Dentro de este subgénero, el más conocido a día a día de hoy es *World of Warcraft*.

En cuanto a los gustos de los españoles en esta materia, en 2010 Nosplay, la red social gratuita en español especializada en videojuegos, realizó un informe entre más de 3.000 jugadores españoles del cual se extrae el siguiente gráfico:

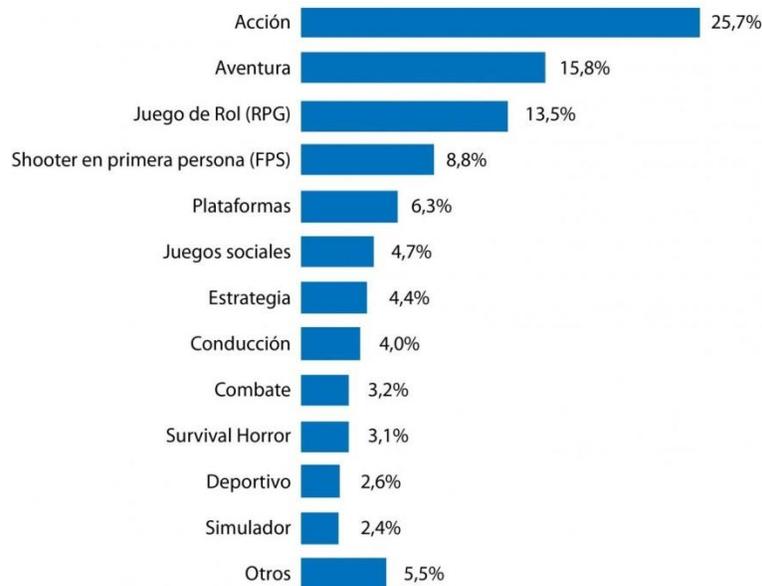


Figura 3.4: Preferencia de videojuegos en España. Fuente: Nosplay.com

3.3. Perfil de los jugadores

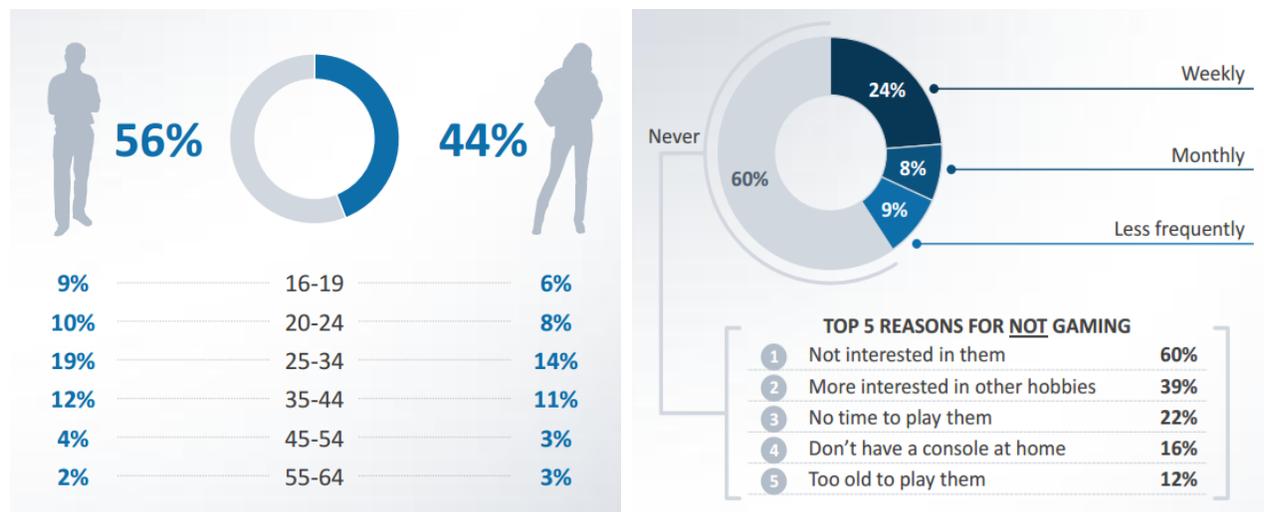
Cada persona que juega a videojuegos lo hace por distintos motivos, juega a cosas diferentes, en horarios dispares, etc. Así, una primera clasificación según las horas que el usuario pasa frente a un videojuego es la siguiente:

- Jugadores ocasionales. Se caracterizan por ser usuarios con poca experiencia que dedican pocas horas semanales a jugar, concibiendo dichas horas como un pasatiempo durante el tiempo libre. Generalmente, estos usuarios suelen preferir juegos deportivos, de arcade y sociales.
- *Gosu*. Esta palabra viene del coreano y describe a aquellos jugadores que tienen una habilidad destacable en el ámbito de los videojuegos. Dedican todo su potencial a dominar la totalidad de un videojuego con el fin de descubrir todos los trucos que este contenga, y así poder proceder de igual modo con el siguiente de su lista.
- *Gamers*. Suelen ser consumidores asiduos a los que les gusta estar informados de todas las novedades lanzadas al mercado. Además, han invertido una buena parte de su tiempo libre en jugar, y sus gustos son variados en cuanto a géneros.
- *Progamers*. Son jugadores profesionales, que viven de ello gracias a su participación en campeonatos oficiales, o incluso trabajando como *testers* para las compañías que se dedican al desarrollo de videojuegos.

Pero si queremos una información más precisa deberemos acudir a algún estudio que desglose detalladamente cuál es el perfil de los jugadores. En nuestro caso, se ha escogido el *European Consumer Study* de noviembre de 2012, una encuesta que Ipsos MediaCT e *Interactive Software Federation of Europe* (ISFE) llevaron a cabo en 16 países europeos. En España, 1.320 personas entre 16 y 64 años rellenaron esta encuesta, que cubría todos los formatos de juegos posibles en los distintos dispositivos existentes (videoconsolas, móviles, ordenadores, etc.).

Uno de los datos más significativos que se extrae de este estudio es que el 40% de los encuestados ha jugado alguna vez a lo largo del año previo al sondeo, donde el 60% de los que sí han jugado, lo han hecho con una frecuencia semanal (Figuras 3.5 y 3.6).

En una clasificación más exhaustiva en cuanto al tipo de videojuego preferido según la edad y el sexo del encuestado, vemos que los hombres entre 16 y 34 años son los jugadores que predominan, con una preferencia por los videojuegos en serie y los multijugador en línea. En cambio, los de corte social presentan prácticamente el mismo porcentaje de jugadores hombres como mujeres en ese mismo rango de edad (Figura 3.7).



Figuras 3.5 y 3.6: Porcentajes de jugadores según sexo, edad y frecuencia. Fuente: ISFE

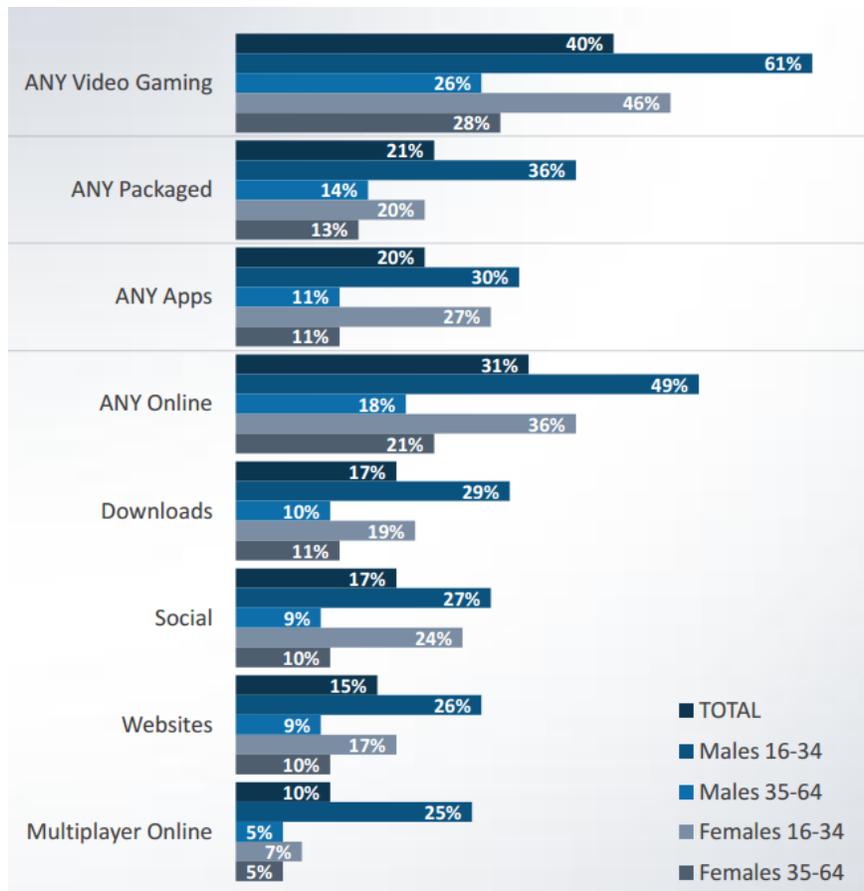


Figura 3.7: Tipos de videojuegos preferidos atendiendo a género y edad. Fuente: ISFE

En cuanto a los dispositivos más empleados para jugar vemos que el porcentaje de usuarios que utilizan un ordenador, ya sea portátil o de sobremesa, supone el 32% del total.

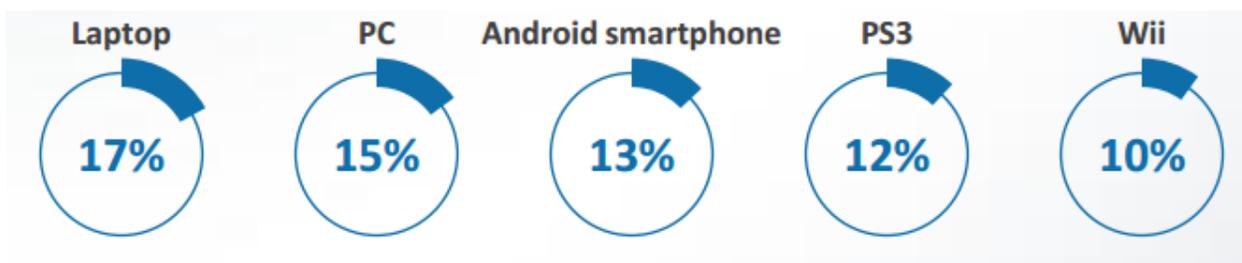


Figura 3.8: Porcentaje de uso de distintos dispositivos para jugar. Fuente: ISFE

Por último, destacaremos de este sondeo la percepción que los encuestados tienen de los videojuegos, donde los adjetivos más asociados a los mismos son: entretenidos, inmersivos, divertidos, buenos para evadirse y competitivos.

WORDS ASSOCIATED WITH GAMING: TOP 5 SELECTED			
		ALL ONLINE	GAMERS
1	Entertaining	40%	59%
2	Immersive	38%	50%
3	Fun	34%	55%
4	Good at providing escapism	32%	45%
5	Competitive	21%	28%

Figura 3.9: Adjetivos asociados comúnmente a los videojuegos. Fuente: ISFE

3.4. Gamificación

La gamificación (del inglés, *gamification*) es el uso de mecánicas y dinámicas de juego en entornos no lúdicos con el objetivo de potenciar el esfuerzo, la motivación, la concentración y la fidelización de los usuarios y así, estos adquieran nuevos hábitos y alcancen una serie de objetivos en tareas en las que con un enfoque tradicional no estarían interesados o incluso considerarían aburridas. Surge de una mezcla entre marketing, psicología y juegos, y pretende crear experiencias de usuario atractivas que provoquen la participación de los mismos de una manera dinámica y proactiva en tareas que de otro modo supondrían un esfuerzo.

La gamificación cuenta con un carácter transversal que hace que sea aplicable a cualquier tipo de actividad o rutina, siendo quizás los sectores de la educación y la salud los que encuentran más ventajoso el uso de esta técnica. Sin embargo, para una implantación efectiva de la gamificación es necesario identificar aquello que se desea incentivar, y después aplicar las mecánicas de juego más adecuadas para alcanzar la motivación y así alcanzar las dinámicas de juego.

3.4.1. Mecánicas de juego

Son una serie de reglas, herramientas y técnicas que sirven para incrementar el compromiso y la motivación de los jugadores a través de la consecución de distintos objetivos. Aplicar estas técnicas a una actividad de carácter no lúdico provoca un enriquecimiento de la actividad al aumentar su atractivo. Algunas de estas mecánicas de juego son las siguientes:

- Niveles. Son una de las mayores motivaciones de los jugadores, al aportar reconocimiento a medida que se avanza a lo largo de la actividad.

- Puntos. Pueden utilizarse para recompensar a los usuarios por los distintos logros alcanzados, servir de indicadores de estatus, ser invertidos para obtener regalos o ser gastados para desbloquear nuevos contenidos.
- Desafíos. Sirven para que los usuarios compitan entre sí para ser considerados los mejores del juego. Es posible que el jugador con la puntuación más alta sea recompensado, mientras que al resto se le otorgue un premio de consolación. Un castigo, en cambio, no suele estar recomendado por el refuerzo negativo que este supone.
- Clasificaciones. Suponen una fuerte componente motivacional al aparecer el nombre del jugador por encima de los de otros usuarios o rivales. Además, permite saber cómo de bien o de mal lo está haciendo un usuario concreto en comparación con el resto.
- Premios. Son una recompensa, ya sea física o virtual, que permite a los distintos usuarios plantearse metas. Normalmente, se trata de logros, trofeos o medallas visibles para el resto de jugadores.

3.4.2. Dinámicas de juego

Las dinámicas de juego son aquellas necesidades e inquietudes universales, atemporales e independientes de países o culturas que motivan a los seres humanos. A través de un conjunto adecuado de mecánicas de juego, se crea una experiencia que fomenta un determinado comportamiento en el usuario a través del cual consigue una o más de estas necesidades. Algunos de ellas son:

- Expresión o autoexpresión. Las personas tienden a desear ser reconocidas por el resto como seres únicos, originales y con autonomía, diferenciándose así de aquellos que les rodean. Este sentimiento se fomenta en los juegos gracias al uso de bienes virtuales, ya hayan sido obtenidos como recompensas, regalos o directamente comprados por el usuario con divisa real. Por ejemplo, el avatar empleado por el jugador normalmente sirve como expresión de sí mismo.
- Logro. Los jugadores motivados por los logros tienden a buscar retos y a establecerse metas alcanzables pero de gran dificultad. Estas metas son alcanzadas mediante la repetición de un conjunto de esfuerzos de manera prolongada en el tiempo.
- Competición. Sirve para motivar a las personas, lo que se traduce en un aumento de su rendimiento, ya que el usuario alcanza la satisfacción al compararse con otros jugadores. Para ello se emplean las tablas de clasificación, que sirven como punto de encuentro de todos los jugadores y fomentan así su motivación.
- Estatus. Generalmente el ser humano siente necesidad de comprometerse con alguna actividad que le proporcione estima, respeto, prestigio, atención, etc. Todos los

elementos de la mecánica de juego conducen a esta dinámica, siendo el aumento de niveles uno de los principales motivadores.

- **Recompensa.** El usuario se siente motivado al recibir una recompensa al realizar una determinada acción. El sistema más extendido de recompensas es aquel que se basa en ganar puntos, sin olvidar aquellos que consisten en subir de nivel u obtener bienes virtuales.

Capítulo 4. Tecnologías

Este capítulo presenta las diferentes tecnologías disponibles en el mercado para desarrollar la aplicación planteada. Se comentarán los aspectos más destacables de los distintos motores de juego y herramientas de desarrollo disponibles en el mercado, así como de los diferentes dispositivos de captura más importantes. Tras proporcionar una serie de detalles de las distintas opciones existentes en cada campo, se realizará una comparativa entre todas ellas, justificando así la elección llevada a cabo.

4.1. Motores de juego y herramientas de desarrollo

Los juegos y las aplicaciones gráficas interactivas están viviendo uno de los momentos más importantes de su historia. La cuota de mercado de la industria del videojuego ha sobrepasado ya a la del cine, y las empresas que deciden apostar por este sector crecen cada año, con más de 300 tan solo en España (Gambin, 2014). Además, la aparición de la nueva generación de dispositivos móviles (iOS, Android, Windows Phone) ha supuesto un gran incremento del mercado, puesto que convierte en clientes potenciales a prácticamente cualquier persona del planeta.

El desarrollo de aplicaciones gráficas que exigen altas prestaciones, en especial los videojuegos, necesitan herramientas de desarrollo altamente especializadas y complejas. También es importante que garanticen una serie de estrictas condiciones, siendo algunas de las más importantes:

- Obtener el máximo rendimiento gráfico de los dispositivos sobre los que se ejecuta la aplicación.
- Aprovechar las características más avanzadas de las unidades de procesamiento gráfico para así conseguir mejores resultados visuales y realismo.
- Multiplataformidad tanto en las herramientas de desarrollo como en su capacidad de exportar la aplicación final a diversas plataformas: web, escritorio, dispositivos móviles y videoconsolas.
- Proporcionar un conjunto de herramientas que permitan a los desarrolladores aprovechar las características mencionadas anteriormente con el menor esfuerzo posible, ofreciendo un potente editor, un poderoso lenguaje de programación o *scripting*, y con una amplia gama de complementos tales como importación de modelos, imágenes, sonidos, motor de físicas, edición de terrenos, sistemas de partículas, herramientas de inteligencia artificial, etcétera.

En la actualidad existen cientos de opciones que cumplen los anteriores requisitos, por lo que un estudio detallado de todos ellos resulta inviable, por lo que se ha decidido acotar el estudio atendiendo a las herramientas más importantes en la actualidad según el estudio *Developer Economics: State of the Developer 3Q 2014*.

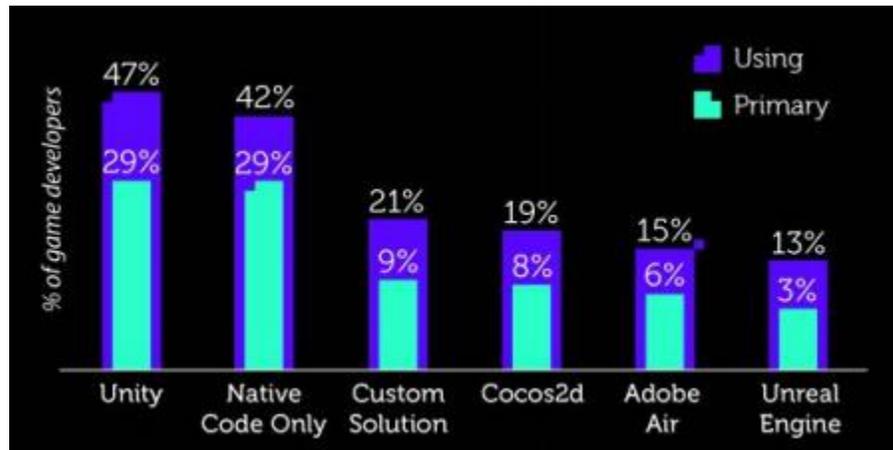


Figura 4.1: Herramientas de desarrollo más utilizadas. Fuente: Developer Economics: State of Nation Q3 2014

4.1.1. Unity

El motor de juego Unity está compuesto por un conjunto de herramientas y servicios que permite crear juegos multiplataforma o contenido interactivo. Todas estas herramientas se encuentran integradas en una misma plataforma, facilitando de este modo la creación tanto de contenido interactivo 2D como 3D, las soluciones colaborativas y el rápido desarrollo multiplataforma; así como servicios de publicidad y análisis.



Figura 4.2: Logo de Unity

Unity es propiedad de la compañía Unity Technologies, fundada en 2004 por David Helgason, Nicholas Francis y Joachim Ante en Copenhague, Dinamarca, tras crear su primer juego: *CooBall*. Este juego no tuvo ningún éxito, sin embargo, los tres estaban de acuerdo en el

gran potencial de las herramientas empleadas en el desarrollo del mismo, por lo que decidieron crear el motor para que pudiera utilizarlo todo aquel que quisiera.

En 2008, durante el gran *boom* de iPhone, Unity Technologies se convirtió en uno de los primeros desarrolladores de motores en empezar a apoyar a la plataforma. Un año después, en 2009, Unity Technologies comenzó a ofrecer una versión de su producto de forma totalmente gratuita, lo que se tradujo en un rápido crecimiento de desarrolladores registrados. Ese mismo año recibiría una financiación de 5.5 millones de dólares por parte de Sequoia Capital, a los que se sumarían otros 12 millones en 2011 por parte de WestSummit Capital e iGlobe Partners.

Actualmente Unity está teniendo un éxito muy notable en el sector, principalmente como consecuencia del enfoque hacia las necesidades de los desarrolladores independientes, que no disponen de medios para crear ni su propio motor de juego ni las herramientas necesarias. El principal objetivo de la compañía es conseguir que el desarrollo de videojuegos y aplicaciones gráficas sea lo más accesible posible a todas aquellas personas en el mundo que así lo deseen.

Gracias a esta filosofía, a día de hoy Unity abarca el 45% del mercado global de motores de juego, encontrándose su competidor más cercano 28 puntos por debajo. Además, el 45% de los desarrolladores emplean Unity como motor de juegos, que ya cuenta con 4.000.000 de profesionales registrados y 600.000.000 de jugadores (Unity Technologies, 2014).

En cuanto al diseño gráfico de los objetos que se desee que formen parte del juego, Unity puede recibir importaciones desde una amplia variedad de programas, tales como 3DS Max, Adobe Photoshop, Adobe Fireworks, Allegorithmic Substance, Blender, Cinema 4D, Cheetah3D, Maya, Modo, Softimage y ZBrush. Los cambios realizados a los objetos creados con cualquiera de estos programas se actualizan automáticamente en todas las instancias de ese objeto durante todo el proyecto sin necesidad de volver a importarlos manualmente.

A nivel de motor gráfico, Unity hace uso de Direct3D (Windows), OpenGL (Mac, Windows), OpenGL ES (Android, iOS), y APIs propietarias (Wii).

Por otro lado, el entorno de desarrollo por defecto desde la versión 3.0 es MonoDevelop, basado en la plataforma de código abierto .NET Mono. En él se pueden utilizar tres lenguajes de programación:

- C#, que en la actualidad supone más del 80% de los *scripts* creados.
- UnityScript, o JavaScript para Unity.
- Boo, que tiene una sintaxis inspirada en Python.

Cabe destacar que en la última versión recientemente lanzada al mercado, Unity 5.0, se dejará de dar soporte a la documentación hasta ahora disponible para Boo, debido a que tan solo el 0.44% de los *scripts* hacen uso de este lenguaje.

La versión 4.6 ha sido uno de los lanzamientos más esperados por los desarrolladores de este motor de juegos, ya que además de incluir mejoras en las tecnologías ya existentes, añade el nuevo sistema Unity UI. Este sistema permite crear fácilmente distintos objetos para la interfaz de usuario gracias a herramientas de diseño intuitivas. Además, se pretende que estos objetos puedan ser redimensionados, anclados o incluso animados de manera sencilla para que el control de su comportamiento vía *script* sea prácticamente innecesario.

Las plataformas soportadas en la versión 4.6 de Unity son las siguientes:

- Windows, Windows Phone 8, Windows Store Apps.
- Linux.
- Mac, iOS.
- Android.
- BlackBerry 10.
- PS3, PS4, PlayStation Vita, PlayStation Mobile.
- XBOX 360, XBOX ONE.
- Wii U.
- Tizen.

A día de hoy se dispone de dos licencias principales para desarrolladores: Unity y Unity Pro. La versión Pro se puede adquirir por 75\$ mensuales o 1500\$ si se prefiere la licencia vitalicia, y permite hacer uso de características adicionales que la versión gratuita no ofrece, tales como soporte LOD, filtrado de audio, soporte de texturas 3D, Occlusion Culling, Deferred Rendering, soporte de *plugins* en código nativo, etc. Además, la versión gratuita muestra en los juegos independientes una pantalla de bienvenida (*Splash Screen*) con el logo de Unity, o una marca de agua en los juegos web, que no es posible personalizar ni desactivar.

Ambas versiones incluyen un entorno de desarrollo, así como tutoriales y ejemplos de proyectos. También se ofrece soporte a través de un foro oficial y numerosa documentación perfectamente estructurada.

Cabe destacar que con el lanzamiento de Unity 5.0 han surgido nuevas licencias para el ámbito de la educación, empresas e industrias, siendo necesario ponerse en contacto con el Gestor de cuentas de Unity regional para conseguir más información al respecto. Por otra parte, aquellos usuarios que ya contaran con Unity Pro 4.6, pueden pasarse a esta última versión 5.0 por 750\$. Esta nueva versión proporciona, según su página web, las siguientes novedades:

- Nuevo *shader* que mejora el aspecto de los materiales sea cual sea el tipo de material o la luz ambiente.
- Unity Ads.
- Editor de 64-bit que maneja las tareas de manera más eficiente.
- Mejora de la luz en tiempo real gracias a Global Illumination.

- Contenido en navegadores que soporten WebGL sin necesidad de *plugins*
- Nuevo sistema de mezclado de audio.

Hay un gran listado de juegos desarrollados con Unity 3D, entre los que destacan: *Blackguards 2*, *Enter the Gungeon*, *Besiege*, *Grow Home*, *Infinifactory*, *Subnautica*, *Teslagrad*, *Playworld Superheroes*, *Last Life*, *Armello*, *Firewatch*, *Kalimba*, *Ziggurat*, *Stranded Deep*, *Adventure Time Game Wizard* o *Relativity*.



Figura 4.3: Imagen del videojuego Blackguards 2. Fuente: <http://www.gameit.es/blackguards-2-analisis-pc>

4.1.2. Cocos2d-x

Cocos2d es un *framework* de código abierto que puede ser usado para desarrollar juegos, aplicaciones o cualquier otro tipo de programa interactivo. Se encuentra separado en diferentes ramas, entre las que destacaremos Cocos2d-x, que es un motor de juego de código abierto bajo licencia MIT.



Figura 4.4: Logo de Cocos2d-x

La idea de Cocos2d surgió en 2005 cuando un grupo de amigos de Argentina, entre los que se encontraba Ricardo Quesada, desarrollaban juegos para un torneo. La meta de dicho torneo era crear un juego en una semana empleando para ello el lenguaje Python. El problema era que cada vez que participaban en la competición, tenían que crear un motor desde cero, lo que distaba de ser algo eficiente. Por este motivo, en 2008 decidieron crear un motor de juegos que originalmente se llamaría “Los Cocos”, y más tarde pasaría a ser conocido como Cocos2d. Basado en Pyglet y escrito en Python, tenía como objetivo servir para la creación de juegos 2D y presentaciones gráficas.

Más tarde ese mismo año, Steve Jobs anunció la apertura del iPhone SDK, lo que provocó que Ricardo Quesada quisiera embarcarse en la versión para iPhone de Cocos2d, que finalmente fue lanzada en junio de 2008 y seis meses más tarde contaba con unos 40 juegos desarrollados en la App Store. Un año más tarde se creó el primer editor de mundos, así como las primeras herramientas, que conseguían reducir el tiempo de programación drásticamente a la vez que mejoraban el resultado final.

El proyecto siguió creciendo y, en el año 2012, la familia ya contaba con Cocos2d-x, una versión de Cocos2d en C++; Cocos2d-xna, que hace uso de C# para XNA y Cocos2d-html. Asimismo, a estas alturas ya se habían desarrollado un buen número de herramientas y de editores, y se contaba con una gran cantidad de recursos, tutoriales y cursos de formación.

A finales de 2012, la empresa Chukong Technologies compra Cocos2d-x y contrata a Ricardo Quesada como arquitecto jefe, adquiriendo el compromiso de añadir nuevas funcionalidades a la plataforma, así como actualizar las ya existentes.

Se estima que en el mundo más de 400.000 desarrolladores confían en Cocos2d-x para crear sus juegos, dominando en China, Japón y Corea del Sur las listas de juegos más exitosos de las tiendas App Store y Google Play.

En cuanto a Cocos2d-x, es usado para crear juegos, apps y otros programas interactivos basados en entornos GUI multiplataforma. En su página web oficial, lo definen como una manera de ahorrar tiempo, esfuerzo y costes debido a su facilidad de uso y sus múltiples y potentes características.

Sus gráficos 2D están basados en OpenGL ES 2.0 para móvil, y OpenGL 2.1 para sobremesa.

Soporta múltiples editores, entre los que destaca CocoStudio por ser el editor de mundos oficial de Cocos2d-x. CocoStudio incluye editor UI, editor de animaciones, editor de escenas y el llamado Data Cruncher, que importa tablas de Excel y las convierte para que sean interpretables por Cocos2d-x, además de poder ser usadas como componentes por el editor de escenas.

Los lenguajes soportados por Cocos2d-x son C++, Lua y JavaScript. La relación entre estos lenguajes y las plataformas soportadas se muestran en la siguiente tabla:

	Plataformas	C++	Lua	JavaScript
Móvil	iOS	Sí	Sí	Sí
	Android	Sí	Sí	Sí
	WP8	Sí	Sí	No
Sobremesa	Win32	Sí	Sí	Sí
	Mac OS X	Sí	Sí	Sí

Tabla 4.1: Relación lenguajes-plataformas soportados por Cocos2d-x

Quizás una de las grandes ventajas de toda la familia Cocos2d es que es necesario ningún tipo de licencia de pago, estando tanto Cocos2d-x como Cocos2d-JS, Code IDE y Cocos disponibles para descargar de manera gratuita en su página web oficial.

Algunas de las características de Cocos2d-x se listan a continuación:

- En cuanto a los gráficos 2D: transiciones entre escenas, transformaciones (mover, rotar, escalar...), sistema de partículas, animaciones de esqueletos, cámara lenta / rápida integrada, etc.
- Componentes UI: vista de página, de lista o *scroll*, botones, *checkboxes*, *sliders*, barras de carga, campos de texto, etc.
- Efectos de sonido y música de fondo.
- Red: HTTP con SSL, WebSocket API y XMLHttpRequest API.

A día de hoy, la última versión de Cocos2d-x es la versión 3.4, que se lanzó en febrero de 2015 y añade, entre otras, las siguientes funcionalidades a las anteriormente citadas:

- Frustum Culling: solo aquello que se encuentra dentro del alcance de la cámara del juego es enviado a la tarjeta de memoria.
- El renderizado de objetos 2D y 3D se ha unificado.
- Debido a la gran importancia del peso de los juegos destinados a dispositivos móviles, se ha optimizado el tamaño de los paquetes cpp.
- Asignación de memoria personalizable.

Desde el año 2010, muchos desarrolladores han empleado Cocos2d-x para construir sus juegos. Algunos de estos juegos son: *Age of Yasuni*, *Badaboom*, *Bounce Classic*, *Contra: Evolution*, *Dirt Race Fury Desert*, *DragonHunter*, *I'm MT Online*, *Jewel Fight: Heroes of Legend*, *Miner Island*, *Star Traders 4X Empires*, *Smile it!*, *WOW Fish*, *ZENONIA 5: Wheel of Destiny*.



Figura 4.5: Imagen del juego Dragonhunter

4.1.3. Adobe AIR

Adobe AIR (*Adobe Integrated Runtime*) es un sistema de ejecución multiplataforma que permite a los desarrolladores hacer uso de sus conocimientos y herramientas en Flash/ActionScript o HTML/Javascript para la creación de aplicaciones RIA (*Rich Internet Applications*), las cuales se usan como aplicaciones de escritorio.



Figura 4.6: Logo de Adobe AIR

Apollo, que así se llamó la versión preliminar de Adobe AIR, fue lanzado en 2007 por Adobe Systems Incorporated, una empresa de software con sede en California, Estados Unidos. Adobe Systems Incorporated fue fundada en 1982 por John Warnock y Charles Geschke, y hoy en día es la responsable de productos tan conocidos como Adobe Acrobat, Adobe Illustrator, Adobe Photoshop, Dreamweaver, Fireworks, Adobe PDF o Adobe Flash Player, entre otros.

A lo largo de 2007, y ya bajo el nombre de Adobe AIR, se publicaron otras tres versiones beta. Finalmente, en febrero de 2008 salió la versión 1.0 para Windows y Mac OS X y hasta la versión 1.5, en diciembre de ese mismo año, no fue compatible con algunas de las diferentes distribuciones Linux: Fedora 8, Ubuntu 7.10 y openSUSE 10.3 de manera oficial.

Las estadísticas de uso más recientes publicados por la compañía acerca de Adobe AIR datan de finales de abril de 2014 (Lynda.com, Inc., 2013), y aseguran ser la tecnología multiplataforma más utilizada para crear aplicaciones tanto para móvil como para equipos PC o Mac, gracias a haber superado el billón de aplicaciones AIR instaladas en los diferentes dispositivos soportados, con más de 100.000 aplicaciones publicadas en distintas tiendas.

En (Lynda.com, Inc., 2013) también se afirma que el número de aplicaciones AIR publicadas seguía creciendo cada mes, perteneciendo más de un 70% de estas apps a las categorías de juegos, educación y entretenimiento; y siendo Android el principal sistema operativo al que iban dirigidas, seguido por iOS, Windows y en mucha menor medida, Mac.

Adobe AIR permite trabajar en los siguientes entornos:

- Flash Builder.
- Flash CS5.
- Flash Catalyst.
- HTML y JavaScript con CSS y XML.

Además de permitir el uso de archivos en formato PDF en cualquier aplicación.

Adobe AIR no cuenta con una interfaz por sí mismo, sino que sirve de puerta para acceder a un sistema operativo. Es decir, el desarrollo de la aplicación se realiza para Adobe AIR, y no para un determinado sistema operativo. Por el momento, ha sido creado para Windows, Mac, Linux y otros sistemas operativos albergados en diferentes dispositivos como *smartphones* o *tablets*.

AIR consta de dos elementos:

- HTML. Contiene HTML, JavaScript, XML y CSS, y permite la visualización de Flash y PDF.
- Flash. Contiene Flex, ActionScript, XML, audio y video, y de manera análoga a HTML, permite la visualización de HTML y PDF.

También es empleado Adobe AIR SDK, que contiene un modelo para un descriptor de archivo con información sobre los archivos (ID, nombre, versión...), así como un *framework* para las APIs y una herramienta de línea de comandos para el desarrollo de aplicaciones AIR.

En cuanto al coste económico de Adobe AIR, cabe señalar que se trata de una tecnología gratuita. Por tanto, es posible descargar e instalar su SDK sin necesidad de adquirir el producto,

si bien es cierto que sí es necesario pagar por otras herramientas como Flash Builder o Flash Catalyst.

Los proyectos creados mediante Adobe AIR pueden ser publicados a través de Flash Builder o Flash CS5, lo que crea un paquete de instalación exportable. Si se prefiere, también se puede hacer uso de manera gratuita de la herramienta ADT (AIR Developer Tool), disponible en la SDK y que permite crear un paquete de instalación nativo según el sistema operativo, por ejemplo .exe en el caso de Windows o .dmg para Mac OS X.

Por otro lado, acerca de las características que presenta Adobe AIR 16, última versión lanzada en febrero de 2015 (Epic Games, Inc., 2015) destacan las siguientes:

- Procesamiento acelerado en GPU, que aprovecha la potencia de los gráficos OpenGL y DirectX.
- Posibilidad de llamar a código nativo a través de extensiones AIR.
- Reproducción de vídeo en HD en formatos H.264, ACC y MP3.
- *Streaming* de vídeo a tasas de bit optimizadas y adaptativas, mediante el uso de los estándares HDS y RTMP.
- Aplicaciones de alto rendimiento gracias a, entre otros, el uso de LZMA como algoritmo de compresión y la posibilidad de comprimir texturas en Stage 3D.
- Mejora del renderizado.
- Interactividad y animaciones más suaves.

Algunos de los juegos desarrollados haciendo uso de Adobe AIR son: *Dimensional Drift*, *FRNZY*, *Ghost Blitz*, *Ironfell*, *KuBu Nonstop*, *Legend of Badminton*, *Lightstorm*, *Mole Warfare*, *Petit Choses*, *Void Control*, *Wonder Wings* o *Zoo Break: Countryside*.



Figura 4.7: Imagen del juego Lightstorm. Demo disponible en <http://www.flare3d.com/demos/lightstorm/>

4.1.4. Unreal

Unreal Engine es un motor de juego válido tanto para PCs como para videoconsolas, que permite crear juego de gran calidad en múltiples plataformas. La arquitectura de renderizado de este motor de juego permite crear sorprendentes gráficos, a la par que escala de manera elegante hacia sistemas de bajo nivel.



Figura 4.8: Logo de Unreal Engine

Todo empezó en 1998 cuando Epic MegaGames lanzó un *shooter* en tercera persona al que llamó Unreal y que posteriormente dio lugar a otros muchos juegos con ese mismo nombre. Debido a los buenos resultados, la compañía decidió aprovechar el motor del juego, Unreal Engine, y lo empezó a comercializar para otras empresas desarrolladoras de juegos.

El éxito que cosechan juegos realizados con este motor, como Unreal Tournament, Deus Ex o Rune, rápidamente ponen de manifiesto las ventajas que suponía hacer uso del mismo. Lejos de limitarse al apartado gráfico, también incluía comunicación en red y control de colisiones, pero sin duda la mayor característica en aquellos años era que se trataba del primer motor que incluía una arquitectura cliente-servidor y permitía gran escalabilidad en partidas en línea.

En la versión 2.0 de Unreal Engine, la estrategia de Epic MegaGames pasó a centrarse en incluir soporte para las principales videoconsolas del mercado y así convertirse en una herramienta que permitiera el desarrollo multiplataforma; mientras que la versión 3.0, lanzada en 2006, se centró en facilitar el trabajo a los diseñadores gráficos, quienes comenzaron a ser capaces de realizar escenarios sin la ayuda de programadores.

Unreal Engine es considerado uno de los motores de juego más potentes que existen en la actualidad. Su boom llegó en 2006 a raíz del videojuego Gears of War, que alcanzó un millón de ventas en las primeras semanas tras su lanzamiento. Además, consiguió más de 30 nominaciones, entre las que destacan las de “Juego del Año 2007”, “Mejor Juego para Xbox 360” y “Mejor Juego Internacional de Acción 2007”.

Una gran novedad en la versión actual, Unreal Engine 4, es el abandono de UnrealScript como lenguaje de programación y su sustitución por C++. Esta última versión es compatible con una buena cantidad de plataformas:

- iOS.
- Android.
- HTML5.
- Linux.
- Mac OS X.
- Oculus Rift.
- Playstation 4 & Morpheus.
- SteamOS
- Windows.
- Xbox One.

Esta última versión se distribuye por 19€ al mes más el 5% de las ganancias obtenidas por las ventas de los juegos desarrollados con este motor, aunque también puede conseguirse de manera gratuita para su uso académico. Los requisitos mínimos para instalar Unreal Engine 4 son: versión de Windows 7 de 64 bits o de Mac OS X 10.9.2, procesador Intel Dual-Core o AMD de 2.5 GHz, GPU compatible con DirectX 11 y 8 GB de RAM.

Este motor permite construir desde juegos muy sencillos hasta otros con efectos muy complejos como Infiltrator, gracias a sus múltiples características. A continuación se destacan algunas de estas:

- Imágenes de alto rango dinámico (HDR, *High Dynamic Range*).
- Miles de luces dinámicas por escena.
- *Anti-aliasing* temporal.
- Edición de sistemas de partículas vía Cascade, que permite a millones de partículas recibir y emitir luz dentro de una escena.
- Blueprints Visual Scripting.
- Sistema Landscape, que permite crear mayores terrenos que en versiones anteriores debido a un uso de memoria eficiente.
- Hot Reload posibilita la realización cambios a nivel de código mientras el juego está corriendo.

Además, en febrero de 2015 se ha publicado la versión 4.7. Esta versión incluye 38 novedades propuestas por los propios desarrolladores, como el rediseño del Blueprint Editor UI, o la posibilidad de combinar HTML5 + WebGL para facilitar el lanzamiento de los juegos en los navegadores.

Algunos de los juegos realizados con Unreal Engine son, además del ya mencionado *Infiltrator*, *A Tale of Two Sons*, *Daylight*, *Dreadnought*, *Epigenesis*, *Last Knight*, *Mortal Kombat X*, *Rime* o *Solus*.



Figura 4.9: Imagen del juego *Infiltrator*. Fuente: <http://pixshark.com/unreal-engine-4-infiltrator.htm>

4.1.5. Comparativa y elección

A continuación se llevará a cabo una comparativa entre los aspectos más importantes del *software* analizado en las anteriores secciones.

Partiendo del hecho de que se deseaba crear un sistema para poder ser utilizado en ordenadores, lo primero fue observar quiénes permitían esto. Como puede verse en la tabla, tanto a través de Unity como de Adobe AIR y Unreal Engine se pueden distribuir juegos para dispositivos Windows, Mac OS X y Linux. La única limitación la encontramos en Cocos2d-x, ya que no contempla la distribución en Linux.

Debido a que en el momento de la elección no se disponía de los conocimientos necesarios para crear un videojuego desde cero, era importante la existencia de documentación así como de foros de ayuda oficiales, para ir adquiriendo una base a través de la realización de pequeños proyectos de ejemplo, y teniendo a su vez la posibilidad de realizar consultas a terceras personas en caso de surgir alguna duda. En este aspecto, Unity y Unreal Engine superan a sus competidores ampliamente.

Entre Unity y Unreal Engine, era necesario conocer cuál permitía una mejor integración con Kinect, puesto que era el dispositivo de captura de movimiento elegido. Tras consultar

diversos sitios web sobre este tema en ambos motores de juego, se concluyó que ambas opciones eran válidas, puesto que la integración se realizaba de una manera casi idéntica.

Otro aspecto destacable era el lenguaje de programación. Unreal Engine trabaja con C++, mientras que Unity 3D permite el uso de C#, UnityScript y Boo, aunque este último desaparecerá en Unity 5 como ya se ha mencionado anteriormente. Es importante resaltar que UnityScript está basado en el lenguaje de programación JavaScript, del que ya se tenían conocimientos previos. Por otra parte, aunque C# era desconocido, presentaba dos ventajas: la primera, que el Trabajo Fin de Grado del que parte este trabajo está desarrollado en C# y, la segunda, que en C# no hay que preocuparse de cuestiones como la gestión de memoria o el uso de punteros.

Por estos motivos, y tras determinar que tanto Unity como Unreal Engine son dos motores de juego muy potentes y con una gran proyección, se decidió usar Unity y se comenzaron a realizar pequeños tutoriales que abarcaban los aspectos más importantes del programa para asentar la base necesaria tanto de scripting como de edición.

4.2. Creación y animación de objetos.

Una vez elegido el motor de juegos que vamos a emplear, debemos pensar en cómo se van a crear los gráficos de nuestro juego, es decir, los *gameobjects*. Unity cuenta con una tienda incorporada llamada *Asset Store* donde se pueden adquirir estos *gameobjects*, diseñados por compañías o por usuarios independientes. La variedad de objetos, escenarios y personajes es inmensa, y los podemos encontrar tanto de forma gratuita como pagando.

Todos estos modelos han sido realizados por medio de programas de diseño específicos para ello, y nosotros mismos podríamos realizar los que nos fueran necesarios mediante programas como Blender, Autodesk, 3ds Max, Maya, SketchUp, MakeHuman, Rhinoceros, Lightwave, etc.

Por otra parte, también necesitamos dotar de movimiento a los personajes de nuestro juego, lo que generalmente se lleva a cabo mediante programas de animación 3D. Estos programas de animación suelen estar ligados con los programas de diseño mencionados anteriormente, es decir, muchos programas nos permiten tanto diseñar como animar nuestras creaciones. Ejemplo de esto son Blender, 3ds Max, Maya, SketchUp o Rhinoceros.

Sin embargo, es necesario reflexionar si realmente merece la pena invertir tiempo y dinero en instalar y aprender a usar un nuevo programa para el diseño y la animación. En nuestro caso, vamos a necesitar dos avatares: uno que represente al usuario y otro que realice los movimientos a imitar por el mismo; así como un escenario en el cual ubicar a estos dos avatares.

Tras analizar las opciones proporcionadas por Unity para poder cumplir estos requisitos, se optó por lo siguiente:

- Obtener los distintos avatares de la *Asset Store*.
- Dotar a dichos avatares del movimiento necesario mediante la componente *Animation* que incluye Unity.
- Crear nuestro escenario a través de la herramienta *Terrain*, también presente en Unity. Esta herramienta nos limitará la temática, pues únicamente permite crear escenarios inspirados en la naturaleza. Sin embargo, este pequeño inconveniente es asumible si tenemos en cuenta la rapidez con la que se va a poder diseñar la escena, los grandes resultados que se van a obtener, y la optimización del renderizado mientras se esté jugando.

4.3. Dispositivos de captura de movimiento

La técnica de captura de movimiento se usa para digitalizar movimientos reales, y así poder dar vida de una manera más fácil e intuitiva a objetos y personajes. Con este fin, muchos de los programas de desarrollo 3D incluyen herramientas para transcribir la información recibida desde un dispositivo de entrada a información que permita la animación de algún personaje.

El sistema precursor de la captura de movimiento es el llamado *rotoscoping*, que aparece por primera vez en el año 1938 en la famosa película de Disney *Blancanieves y los Siete Enanitos*. Esta técnica también se emplearía años más tarde en otras producciones de películas animadas o vídeos de música.

En la década de los 90, la captura de movimiento se comenzó a utilizar con frecuencia en la industria de los videojuegos. *Highlander: The Last Of The Macleods* y *Soul Blade* son algunos ejemplos de videojuegos que hacían uso de esta técnica para conseguir mayor realismo en sus avatares. A la vista de los resultados, poco tardó la industria cinematográfica en emplear esta técnica en sus personajes, como Jar-Jar Binks de *Star Wars Episodio I: La Amenaza Fantasma*, o Gollum de la saga de *El Señor de los Anillos*.

A partir de entonces, se han creado multitud de equipos que utilizan diferentes métodos para grabar los movimientos de personas y animales. Además, las exigencias que se imponen a las empresas dedicadas al MoCap (*Motion Capture*) cada vez son mayores, siendo necesarias mejores prestaciones y una representación del movimiento casi perfecta.

En la actualidad es posible capturar datos desde dispositivos tan comunes como teclados, ratones o *joysticks*, hasta otros sistemas mucho más avanzados que permiten recopilar datos de mucha mayor complejidad y precisión. Dentro de estos sistemas avanzados existe una gran variedad de métodos para capturar los movimientos. A continuación se listan algunos de ellos:

- Sistemas electromecánicos. En ellos, el actor o intérprete debe vestir un traje especial, adaptable al cuerpo humano y compuesto de barras metálicas o plásticas unidas mediante potenciómetros que se colocan en las principales articulaciones y se encargan de recoger los datos sobre el grado de apertura de las mismas.
- Sistemas electromagnéticos. Disponen de una colección de sensores electromagnéticos que se colocan en el cuerpo y se conectan a una unidad eléctrica central mediante cables. Cada sensor está compuesto por tres espiras ortogonales que miden el flujo magnético, que permiten determinar la posición y orientación del sensor al medirse la relación espacial con un transmisor cercano.
- Sistemas ópticos. Utilizan los datos recogidos por sensores de imagen para conocer la posición de un elemento, usando una o más cámaras sincronizadas para proporcionar proyecciones simultáneas. Lo habitual hasta hace poco es que fuera necesario hacer uso de indicadores o *markers* pegados al usuario para poder recoger los datos, pero los sistemas más recientes permiten recoger datos fiables rastreando puntos del sujeto identificados dinámicamente.
- Sistemas de fibra óptica. Los primeros sistemas de este tipo son los guantes de fibra óptica, aunque en la actualidad esta técnica ya se emplea para la captura de movimientos del cuerpo entero. Para ello, los sensores se fijan sobre distintas partes del cuerpo y, cuando se dobla la fibra óptica de su interior, la luz que se está transmitiendo en dicha fibra se atenúa y permite así calcular la posición de cada parte del cuerpo en cuestión.
- Sistemas inerciales. Recogiendo la información de los distintos sensores (acelerómetro y giroscopios normalmente) es posible conocer la posición, eje de giro y velocidad angular de cualquier sensor. En estos sistemas puede darse que los errores numéricos se acumulen sobre la velocidad u orientación del sensor, problema conocido como deriva de integración, por lo que suelen combinarse con otras técnicas de captura.
- Sistemas de ultrasonidos. Hacen uso de emisores que generan pulsos ultrasónicos y de receptores situados en posiciones conocidas que capturan dichos pulsos, permitiendo averiguar la posición del emisor en el espacio. Estos sistemas son aún demasiado voluminosos y no interpretan correctamente los movimientos bruscos, por lo que se suelen combinar junto con sistemas inerciales.

Dada la naturaleza del juego serio a desarrollar, parece lógico pensar en hacer uso de un sistema que permita la mayor precisión posible sin hacer cargar al paciente con trajes y aparatos complejos y/o pesados. Por lo tanto, para este proyecto un sistema óptico parece la mejor solución a la hora de capturar los movimientos del usuario. Pasemos a analizar entonces las distintas opciones dentro de estos sistemas.

4.3.1. Kinect



Figura 4.10: Kinect para Windows v2. Fuente: <http://www.gizmag.com/kinect-v2-windows/31490/>

La idea de Microsoft de desarrollar un dispositivo de entrada para captar movimientos fue anunciada a mediados de 2009 bajo el nombre de “*Project Natal*”. Aunque muchos vieron este proyecto simplemente como una respuesta al éxito que Nintendo estaba cosechando gracias a Wii, la realidad es que Kinect fue un gran éxito de ventas en aquellos años.

Kinect permite a los usuarios interactuar con el PC o la videoconsola y controlarlos sin necesidad alguna de establecer contacto físico con periféricos de entrada, gracias a la interfaz natural de usuario (*Natural User Interface*, NUI).

Kinect 1.0 usaba un proyector infrarrojo y una cámara que escaneaba la escena y enviaba la información a un microchip especialmente preparado para capturar en tres dimensiones el movimiento de objetos y personas. Además, también contaba con una fila de micrófonos que permitía el reconocimiento de voz.

Para llevar a cabo el reconocimiento facial, de gestos y de voz, Kinect 1.0 presentaba una cámara con resolución VGA a 640x480 píxeles por defecto, aunque era posible trabajar a 1280x1024 a costa de reducir la tasa de refresco. Toda esta información apenas era procesada por el microchip, ya que era la propia consola la que se encargaba de esta tarea.

No fue hasta Junio de 2011 cuando Microsoft anunció el lanzamiento del *Software Developer Kit* (SDK) para este dispositivo, ya que anteriormente solo podía ser empleado con fines lúdicos junto con la consola Xbox 360. La versión comercial del SDK llegaría a principios de 2012, permitiendo así a los desarrolladores de todo el mundo hacer uso de Kinect en sus propias aplicaciones.

Dos años más tarde, en 2014, sale al mercado Kinect 2.0, cuya principal diferencia respecto a su predecesora reside en la nueva cámara principal: una cámara *time-of-flight* (TOF)

de alta resolución que permite capturar más detalles con gran precisión y mayor resolución. El nuevo modo de profundidad que proporciona esta cámara TOF permite reproducir una escena con tres veces más fidelidad que con Kinect 1.0.

Pero esa no es la única mejora: también logra un campo de visión un 60% mayor, lo que posibilita que más personas sean registradas a la vez (seis esqueletos completos frente a los dos que detectaba la versión anterior) y a una menor distancia del dispositivo. Además, el nuevo sensor de infrarrojos es capaz de reconocer cuerpos en una habitación completamente a oscuras.

Otras características clave que presenta el SDK 2.0 son las siguientes:

- Se incluyen nuevos *joints* a los ya existentes, que permiten reconocer diferentes gestos con las manos. Además, un mayor conocimiento del tejido conectivo humano y de la postura corporal permite obtener posiciones anatómicamente más precisas y una sensación de mayor naturalidad en los avatares.
- *Kinect Studio* proporciona una mejor grabación y reproducción, y *Visual Gesture Builder* permite a los desarrolladores construir sus propios gestos para que el sistema los reconozca.
- Reconocimiento facial avanzado, con una resolución 20 veces mayor que permite la creación de máscaras con más de 1.000 puntos para una representación más precisa del rostro humano.
- Soporte *multi-app*, que permite a varias aplicaciones acceder al mismo tiempo a un mismo sensor.
- Soporte para Unity Pro, posibilitando así la creación y el lanzamiento de aplicaciones mediante el uso de herramientas ya conocidas por el desarrollador.

Por otra parte y como es lógico, los requisitos de *hardware* de esta nueva Kinect son mayores que los de su predecesora. En la siguiente tabla podemos ver una comparación entre los requisitos mínimos necesarios en las diferentes versiones:

	Kinect 1.0	Kinect 2.0
Arquitectura	x86 o x64	x64
Procesador	Doble núcleo a 2.66 GHz	Doble núcleo a 3.1 GHz
Tarjeta gráfica	Soporte DirectX 9.0c	Soporte DirectX 11
RAM	2GB	4GB
USB	2.0	3.0
Sistema Operativo	Windows 7	Windows 8

Tabla 4.2: Comparativa entre las dos versiones de Kinect

4.3.2. Xtion PRO LIVE



Figura 4.11: Xtion PRO LIVE. Fuente: <http://vr-zone.com/articles/asus-xtion-pro-live-launching-soon/12996.html>

Xtion nace como fruto del acuerdo alcanzado entre *PrimeSense* y Asus para llevar la tecnología de detección al PC y así introducirse en el mercado del MoCap y competir directamente con Microsoft y su Kinect.

Aunque en sus orígenes este producto contaba tan solo con un sensor de profundidad, la versión Xtion PRO LIVE, orientada a desarrolladores, cuenta también con cámara RGB y micrófonos en sus extremos. La integración de todo este *hardware* permite la detección de gestos y de cuerpo entero, a la vez que reconoce sonidos con el fin de que el usuario controle todo tipo de parámetros con su voz.

Esta versión cuenta con el SDK compatible OPNI NITE *middleware* para facilitar un desarrollo simple de aplicaciones y juegos, y desde este año también existe la tienda *online* ASUS@Vibe, pensada para que los desarrolladores puedan vender sus creaciones en ella. Además, otra cualidad muy importante dado el motor de juegos escogido en el apartado 4.1.5. de este capítulo, es su compatibilidad con Unity 3D.

Entre sus especificaciones destacaremos el soporte de arquitecturas tanto x86 como AMD, así como su compatibilidad para sistemas Windows, Linux Ubuntu e incluso Android si se solicita previamente.

4.3.3. DepthSense



Figura 4.12: DepthSense DS325. Fuente: <http://www.softkinetic.com/Store/ProductID/29>

La compañía belga SoftKinetic desarrolla *hardware* y *software* para el reconocimiento gestual, y su solución *hardware* se conoce como DepthSense. Esta solución utiliza una tecnología patentada denominada CAPD (*Current Assisted Photonic Demodulation*).

En la actualidad, el modelo que presenta mejores prestaciones es la cámara DepthSense DS325, que incluye un sensor de infrarrojos *time-of-flight* capaz de transformar los datos captados en imágenes RGB 3D a tiempo real. Además, cuenta con un sensor RGB HD, dos micrófonos y una sola entrada USB.

Según la compañía, se trata de la cámara más precisa en el mercado en cuanto a profundidad. DS325 proporciona información de profundidad desde los 15 centímetros de distancia a una frecuencia de hasta 60 fps, siendo de este modo capaz de detectar el movimiento de dedos. Tiene una resolución de profundidad por QVGA de 320x240, una resolución RGB de 720p y necesita luz ambiente de interiores para funcionar correctamente, lo cual supone una desventaja frente a Kinect, que no necesita de luz ambiente.

Por otro lado, SoftKinetic permite trabajar con el *middleware* denominado *iisu SDK*. Este *middleware* dispone de interfaces y patrones de gesticulación predefinidos para reducir los ciclos de desarrollo, y es compatible con Windows 7, Windows 8, Linux y Android.

SoftKinetic pretende ser la opción preferida por los desarrolladores en el mercado de la visión 3D y el reconocimiento de gestos, dando paso de esta manera a una nueva generación de videojuegos, nuevas aplicaciones de entretenimiento educativo, experiencias mejoradas en videoconferencias, etc. Sin embargo, de momento no es posible el reconocimiento de cara y voz con sus cámaras DepthSense.

La tecnología de SoftKinect está presente en *iSec*, una videoconsola de Lenovo que salió a la venta en China a finales de 2011, y de la que en su día se dijo que podría llegar a ser una gran competidora en la próxima generación de videoconsolas.

4.3.4. Minoru 3D



Figura 4.13: Minoru 3D. Fuente: <http://www.robotshop.com/en/minoru-3d-webcam.html>

Minoru 3D Webcam fue lanzada en enero de 2009 y fue la primera cámara con tecnología 3D incorporada. Este dispositivo se conecta mediante USB al ordenador, de forma que sus dos cámaras, situadas a una distancia la una de la otra similar a la existente entre los ojos de una persona, se iluminan y comienzan a grabar vídeos anáglifos (vídeos en 3D) así como en otros formatos estereoscópicos.

El *software* de Minoru 3D cuenta con procesamiento anaglífico nativo, y la imagen 3D que genera puede ser vista a través de unas gafas 3D de lentes rojas y azules, aunque también permite ser utilizada como una cámara convencional de imagen 2D. La resolución de las imágenes que capta van desde los 320x240 hasta los 800x600 píxeles, a una frecuencia máxima de 30 fps.

Otras de sus especificaciones son un sensor VGA con tecnología CMOS, enfoque manual a partir de una distancia mínima de 10 centímetros, campo de visión de 42 grados y desviación máxima de 16,5 milisegundos entre sus dos obturadores. Además, este dispositivo dispone de software para ser utilizado en OpenCV, proporcionando tanto salida en anáglifo como en imagen de profundidad. Para que esto sea posible, implementa funciones de calibración mediante un tablero de ajedrez de dimensiones conocidas.

Entre sus inconvenientes se encuentran que para un uso en exteriores sea necesario filtrar el exceso de luz, y que tan solo sea compatible con Windows XP, Windows Vista y Windows 7.

4.3.5. Comparativa y elección

Tras mencionar los principales aspectos de las diferentes cámaras, en la siguiente tabla se realiza una comparación entre ellas:

	KINECT	Xtion	DepthSense	minoru
Resolución en profundidad	512x424	320x240	160x120	Vía <i>software</i>
FPS	30	30 o 60	30 o 60	30
Rango	0,5 – 4,5 metros	0,8 – 3.5 m	0,15 – 1 m	> 10 cm
Tipo	TOF	Luz estructurada	TOF	Estereoscópica
Compatibilidad	Windows 8	Windows XP, 7, Linux y Android	Windows 7, 8, Linux y Android	Windows XP, Vista, 7
Software	Kinect for Windows SDK	OpenNI	iisu	Librerías para OpenCV
Consumo energético	Alimentación adicional al USB	< 2,5 W	< 2,5 W	< 1,5 W
Precio	180 €	165 €	145 €	30 €

Tabla 4.3: Comparativa entre los distintos dispositivos de captura analizados.

En nuestro caso, nos interesa una cámara que sea capaz de capturar al menos el cuerpo entero de un usuario, no siendo necesaria una gran precisión en los dedos de las manos. Por este motivo, DepthSense DS325, con un alcance máximo de un metro, queda descartada como opción válida.

Por otra parte, y como ya se comentó en el apartado anterior, Minoru 3D presenta la desventaja de ser capaz de trabajar únicamente con luz ambiente. Si tenemos en cuenta que seguramente el juego vaya a ser visualizado a través de un proyector y por tanto la luz deberá ser tenue para visualizar mejor la pantalla, esta cámara queda descartada a pesar de ser una opción muy interesante debido a su bajo precio.

Por tanto, la decisión está entre Xtion PRO LIVE y Kinect para Windows. Es necesario destacar que el Trabajo Fin de Grado del que este parte fue realizado con la primera versión de Kinect para Windows, y que tanto esta como la nueva versión son las cámaras más empleadas por los usuarios (ya que se utilizan en gran cantidad de videojuegos de la consola Xbox de Microsoft) y por los desarrolladores, lo que se traduce en una mayor documentación y soporte en caso de que así fuera necesario. Por tanto, aunque ambas cámaras son una solución perfectamente válida en cuanto a prestaciones, nos decantaremos por Kinect para Windows debido a lo anteriormente comentado.

4.4. *Plugin para Kinect*

En un programa informático, un *plugin* es aquella aplicación que añade una funcionalidad adicional o una nueva característica al software. Generalmente, el *plugin* se ejecuta mediante el *software* principal, con el que interactúa a través de una interfaz. Una de las ventajas que ofrecen estos complementos es que facilitan la colaboración de desarrolladores externos con el *software*, quienes pueden realizar sus aportes a las funcionalidades extendiendo así sus funciones.

En nuestro caso, necesitamos un *plugin* para que Unity funcione con Kinect V2 y así poder aprovechar la tecnología que nos ofrece tanto el motor de juegos como el dispositivo para poder desarrollar nuestro juego. Como Kinect V2 cuenta con un *add-in* oficial para Unity Pro, escogeremos directamente esta opción. Para ello, en primer lugar es necesario descargar e instalar Kinect for Windows SDK 2.0. De este modo, nuestra Kinect será compatible con nuestro PC con sistema operativo Windows 8.

Una vez instalado el SDK, podremos acceder a algunos ejemplos y demostraciones muy útiles para empezar a entender cómo funciona nuestra Kinect V2. Entre ellos se encuentra *Kinect Configuration Verifier*, con el que podremos comprobar que efectivamente ordenador y cámara se están comunicando correctamente.

Finalmente, nos descargaremos los paquetes para Unity Pro. Estos se encuentran accesibles en <http://www.microsoft.com/en-us/kinectforwindows/develop/downloads-docs.aspx>. Si la descarga se realiza correctamente, podremos acceder al archivo *unitypackage* así como a los ejemplos que contiene.

Capítulo 5. Módulo desarrollado

En este capítulo se explicará detalladamente la solución finalmente alcanzada en este Trabajo Fin de Grado. En primer lugar, será necesaria una descripción de los diferentes actores y de las funciones que pueden llevar a cabo en los distintos módulos. A continuación, se llevará a cabo una explicación profunda y precisa del trabajo desarrollado. Al tratarse este de un módulo que se relaciona con otras partes de un juego serio, será necesario también realizar una explicación general del resto de módulos del videojuego con el fin de otorgar al lector una visión general de todo el conjunto.

5.1. Aspectos generales

El objetivo final de este juego serio es mejorar a través de una plataforma virtual la autonomía y la independencia de personas con movilidad reducida como consecuencia de un infarto cerebral. Así pues, se decidió dividir el juego en cuatro módulos bien definidos: módulo de valoración, módulo de administración, módulo de terapeuta y módulo de juego.

- Módulo de valoración. Es donde se da de alta a los usuarios, se recoge y analiza el rango máximo de movimiento (ROM) de los pacientes, y se puede consultar la evolución de los mismos a lo largo de las distintas sesiones.
- Módulo de administración. Aquí se crea la base de datos de ejercicios según el nivel de dificultad deseado, pudiendo agrupar cierto número de posturas relaciones bajo un nombre identificativo.
- Módulo de terapeuta. Permite al terapeuta diseñar una sesión de juego de acuerdo a los requisitos de cada paciente.
- Módulo de juego. El usuario deberá moverse hasta que la posición de su avatar coincida con la de una silueta que se desplaza.

A su vez, tres tipos de actores diferentes podrán interactuar con los distintos módulos. La relación de interacción entre unos y otros se muestra en la siguiente tabla:

	Módulo de valoración	Módulo de administración	Módulo de terapeuta	Módulo de juego
Paciente	<ul style="list-style-type: none"> ✔ Registro en el sistema ✔ Evaluación de su ROM ✔ Análisis de su evolución 	✘	✘	<ul style="list-style-type: none"> ✔ Realización de las sesiones de rehabilitación
Terapeuta administrador	⚠ *	<ul style="list-style-type: none"> ✔ Creación de toda la base de datos de ejercicios 	<ul style="list-style-type: none"> ✔ Diseño de las sesiones de los pacientes 	⚠ *
Terapeuta	⚠ *	✘	<ul style="list-style-type: none"> ✔ Diseño de las sesiones de los pacientes 	⚠ *

*No diseñados para terapeutas pero accesibles por ellos con fines explicativos, de supervisión o de análisis

Tabla 5.1: Relación de interacción entre módulos y actores en el sistema

5.2. Módulo de administración

Este módulo permite la creación de las diferentes posturas que más tarde serán usadas por el terapeuta para configurar las distintas sesiones de rehabilitación necesarias para los pacientes. Estas posturas deberán ser asociadas desde un primer momento a un nivel de juego determinado, atendiendo a las siguientes reglas que el terapeuta administrador deberá conocer de antemano:

- Nivel 1. Posturas con movimiento de tronco y extremidad superior unilaterales siempre en apoyo bipodal, con los pies a la altura de la cadera y las rodillas en extensión.
- Nivel 2. Posturas con movimiento de tronco y extremidad superior bilaterales simétricas siempre en apoyo bipodal, con los pies a la altura de la cadera y las rodillas en ligera flexión (nunca más de dos siluetas consecutivas en flexión).
- Nivel 3. Posturas con movimiento de tronco y extremidad superior bilaterales asimétricas en apoyo bipodal, con los pies a la altura de la cadera y las rodillas en ligera flexión (nunca más de dos consecutivas).
- Nivel 4. Posturas con movimiento de tronco y extremidad superior bilaterales simétricas y asimétricas en apoyo bipodal, con los pies juntos y las rodillas en ligera flexión (nunca más de dos siluetas consecutivas en flexión).
- Nivel 5. Posturas con movimiento de tronco y extremidad superior bilaterales simétricas y asimétricas, con diferentes apoyos bipodales y movimientos de

abducción de cadera mediante pasos laterales y movimientos de flexión o extensión de cadera mediante pasos frontales.

- Nivel 6. Posturas con movimiento de tronco y extremidad superior bilaterales simétricas y asimétricas con diferentes apoyos bipodales o monopodales. Movimientos de abducción de cadera mediante pasos laterales y movimientos de flexión o extensión de cadera mediante pasos frontales.



Figura 5.1: Interfaz del módulo de administración

Es necesario destacar que las figuras son adaptativas a cada paciente, es decir, se crean en base a porcentajes que se ajustarán a los resultados obtenidos por estos durante su valoración. Por ejemplo, una figura creada con una amplitud del 100% en el movimiento de abducción de brazo que se asigne a un paciente que haya obtenido un resultado de 60° en ese mismo movimiento durante la valoración, se adaptará a esos 60° y esa será la amplitud que el paciente deberá intentar alcanzar durante el juego. Si en vez del 100%, la figura se crea el 50% de amplitud, el paciente deberá alcanzar 30°, etc.

5.3. Módulo de terapeuta

El sistema permite configurar para cada paciente y sesión el grupo de ejercicios a realizar. El terapeuta podrá elegir varios grupos de ejercicios que el usuario deba realizar en la sesión. Estos grupos de ejercicios habrán sido creados previamente en el módulo de administración.

Cuando el terapeuta entra en este módulo, se le pregunta por el nombre del paciente. Una vez introducido se muestran todos los grupos de ejercicios disponibles en el sistema, que serán

aquellos previamente creados en el módulo de administración. Entre todos estos grupos, el terapeuta deberá seleccionar los que desee que formen parte de la sesión de entrenamiento de ese paciente concreto.



Figura 5.2: Interfaz del módulo de terapeuta

5.4. Módulo de juego

Este módulo permite al paciente jugar dentro de un mundo virtual realizando los movimientos que el terapeuta le haya programado para su sesión de juego. Estos movimientos estarán ordenados en un mismo grupo o en varios diferentes, y cada grupo deberá estar formado por siluetas pertenecientes a un mismo nivel. Además, el paciente puede jugar con una velocidad, dificultad, modo, tiempo de espera entre siluetas y duración de la sesión configurables. Todos estos parámetros son definidos por el terapeuta para cada sesión de juego del paciente en cuestión.

De entre todos los parámetros anteriores, quizás el menos intuitivo sea el modo de juego. Esta variable permite añadir un subnivel de dificultad dentro de cada nivel, y establece la exactitud con la que el paciente debe imitar la postura de la silueta durante el juego en función de su ROM máximo. Los diferentes modos de juego existentes son los siguientes:

Nombre	Porcentaje ROM máximo
Aprendiz	80%
Oficial	90%
Maestro	100%
Doctor	105%
Experto	110%

Tabla 5.2: Modos de juego

Para entender mejor cómo funciona, veamos un ejemplo: supongamos que durante el módulo de valoración (que se explicará detenidamente en el siguiente punto) el paciente consigue realizar un movimiento de abducción de hombro con una amplitud de 60° , y que el terapeuta le asigna a dicho paciente una figura con una abducción de hombro al 70%. Del mismo modo, el terapeuta decide que el paciente juegue en modo aprendiz, que como hemos visto en la anterior tabla significa un reajuste al 80% del ROM máximo. Por tanto, con esta configuración, las siluetas que le aparezcan al usuario con la figura anteriormente descrita saldrán con una amplitud en grados de $60^\circ * 0.7 * 0.8 = 33.6^\circ$. Así pues, se considerará que el paciente ha realizado correctamente el movimiento cuando la amplitud que alcance sea de $33.6^\circ \pm \text{umbral}$. Este umbral es un parámetro fijo que se define en el código fuente del sistema.



Figura 5.3: Interfaz del módulo de juego. De arriba a abajo y de izquierda a derecha, selección del paciente, selección de la sesión de entrenamiento, configuración de los parámetros y momento durante el juego

5.5. Módulo de valoración

En este apartado se explica detalladamente el funcionamiento del módulo desarrollado. El objetivo no es describir cada *script* y método implementado, sino que el lector comprenda todas las funcionalidades que dicho módulo presenta. Por este motivo, solo se mencionará a continuación el nombre de cada *script* y el tipo de funciones que podemos encontrar en cada uno, para después dar paso a la descripción del flujo de eventos del módulo.

- *gestionUITest.cs*. Gestiona la interfaz de usuario del módulo: activa o desactiva paneles, muestra títulos y valores, cambia colores de los paneles, etc.
- *movimiento.cs*. Se encarga de calcular los ángulos de la articulación o articulaciones que nos interesan, así como de las distancias entre pies.
- *evaluacion.cs*. Define cuándo y cómo se almacenan dichos ángulos o distancias, además de decidir cuándo se debe sacar una fotografía al paciente.
- *actualizarTele.cs* Gestiona la toma de fotografías, así como su almacenamiento y posterior muestra.
- *operacionesXml.cs*. Almacena los valores de ángulos o distancias deseados para su posterior tratamiento. Este *script* es compartido con otros módulos.

5.5.1. Pantalla inicial

Al entrar en ese módulo, lo primero que debe hacer el terapeuta es introducir el nombre o apodo mediante el cual se conoce o será conocido el paciente que va a realizar la valoración. En caso de que el paciente no haya sido registrado en el sistema anteriormente, en este momento y de forma transparente a los actores se creará una carpeta llamada “Valoraciones” en la ruta \Assets\XML\Pacientes que será la que almacene los resultados del test que se llevará a cabo en este módulo. Si por el contrario, el paciente ya se encuentra registrado, el terapeuta tiene la posibilidad de ayudarse de una lista de pacientes ya existentes, ordenada alfabéticamente y creada a partir de la lectura de todas las carpetas disponibles en la ruta \Assets\XML\Pacientes con el fin tanto de facilitar la búsqueda al terapeuta, como de evitar que un mismo paciente se encuentre registrado con dos o más nombres diferentes debido a equivocaciones..

Una vez introducido el nombre del paciente, se muestran tres opciones: realizar una nueva valoración, pinchando en el botón “CONTINUAR”; ver los resultados de las ya realizadas, mediante el botón “VER EVOLUCION”; o volver al menú anterior.

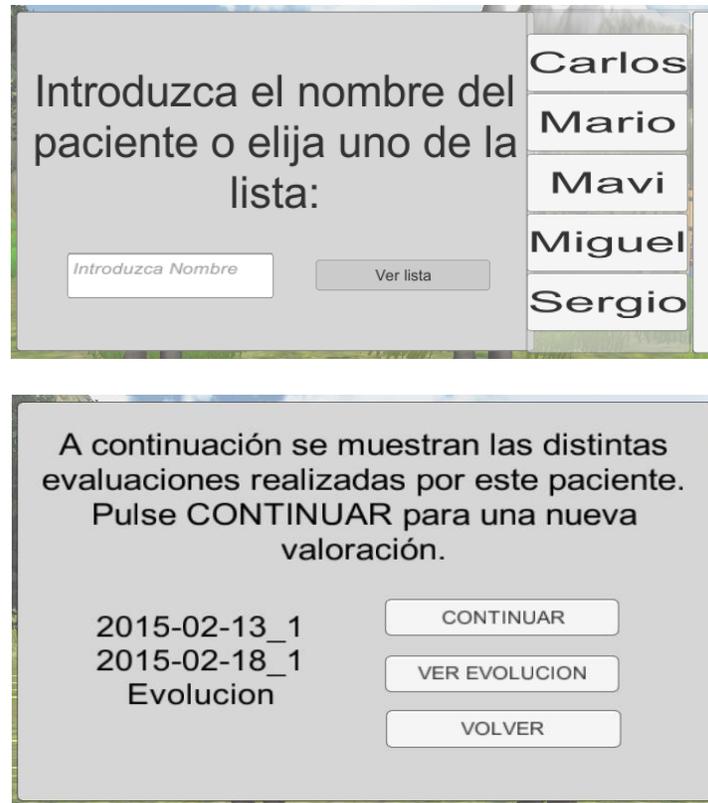


Figura 5.4: Paneles del módulo de valoración

Estos paneles se crearon con relativa facilidad gracias al editor UI de Unity, que permite hacer uso de un Game Object denominado Canvas que debe contener todos los elementos con los que un usuario pueda interactuar (paneles, imágenes, textos, *sliders*, *scrollbars*, etc.). Es decir, debe ser padre de todos estos. La ventaja de usar este Game Object radica en que durante la vista de juego por defecto se muestra como un rectángulo que se adapta automáticamente a la resolución de cada pantalla. Del mismo modo, todos los hijos que contenga el Canvas deben también reajustar su forma, por lo que se ha tenido que poner especial cuidado en fijar correctamente las anclas o *anchors*, que es el nombre por el que se pueden encontrar en Unity.

Este concepto de anclas es el encargado de fijar un Game Object hijo a su respectivo Game Object padre según convenga. Por ejemplo, el hijo puede anclarse al centro de su padre, o a una de las esquinas. De este modo, el hijo se reajustará de diferentes maneras cuando el tamaño del padre varíe. En la vista de escena, las anclas aparecen representadas como cuatro pequeños triángulos, mientras que en el Inspector podemos encontrar información más detallada sobre ellas.

5.5.2. Test de valoración

Volviendo al flujo de eventos del módulo de valoración, cuando en el segundo panel de la Figura 5.4 se hace *click* sobre el botón “CONTINUAR”, este desaparece para dejar paso a la elección del avatar que el paciente prefiera. La figura seleccionada no será recordada por el sistema, esto es, tanto en sucesivas valoraciones como en el módulo de juego aparecerá la opción de elegir un nuevo avatar. Recordemos que tal y como se comentó en el Capítulo 3 de esta memoria, el simple hecho de poder dar a elegir al paciente qué avatar quiere que le represente ayuda a satisfacer la dinámica de juego de la autoexpresión.



Figura 5.5: Panel de selección de avatar y algunos de los avatares disponibles

Tal y como se adelantaba en el Capítulo 2, no ha habido necesidad de diseñar estas figuras, sino que todas ellas se han podido descargar de manera gratuita de la Asset Store de Unity, bajo el nombre de *Male Character Pack* y *Female Character Pack*.

Para poder hacer esta selección de avatares al pulsar sobre las flechas, fue necesario implementar el método public void `seleccionAvatar(int suma)`. Cada vez que se pulsa sobre una de las flechas, se resta o se suma una unidad a un contador dependiendo de si la pulsada es la izquierda o la derecha respectivamente. Este método recoge ese valor y lo relaciona con una clase `Dictionary`, que contiene los nombres por los cuales se conoce a cada una de las figuras. Una vez que sabemos qué *Game Object* debe mostrarse, solo queda destruir el anterior e instanciar el nuevo objeto cargándolo desde la ruta en la que se encuentra.

Una vez escogido el avatar que va a representar al paciente durante la valoración, se puede ver la escena de manera más nítida debido a la desactivación de los paneles que hasta ahora interactuaban con el usuario. El paisaje que se observa es un *Game Object* tridimensional creado con la herramienta *Terrain*. Es decir, si esta escena no fuera estática el usuario podría desplazarse por el terreno creado al no tratarse de una imagen sino de un objeto al que se le puede añadir distintos suelos, árboles, flores, etc., e incluso disminuir o aumentar el desnivel de dicho terreno.

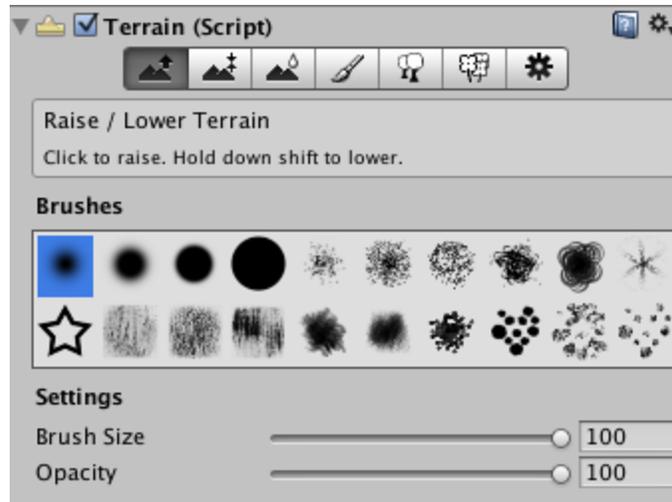


Figura 5.6: Herramienta *Terrain* de Unity

Sobre este terreno aparecen dos figuras: a la derecha, la que previamente se ha elegido y que imitará los movimientos del paciente, y a la izquierda un avatar que ningún usuario puede variar y que se encarga de realizar la secuencia de movimientos a imitar por el paciente en tiempo real, tratando de alcanzar su máximo ROM. Esta secuencia se lleva a cabo dos veces y está compuesta por 31 movimientos, cada uno de los cuales parte y finaliza en una posición base en la que el cuerpo se encuentra relajado y en reposo.

Cada uno de los movimientos a evaluar fueron ideados por especialistas de la Fundación Aspaym, y se implementaron en el *Game Object* mediante la herramienta *Animation* de Unity, que permite dotar de movimiento a la figura de manera muy precisa, pudiendo rotar todas las articulaciones tantos grados como se desee en cualquier eje.

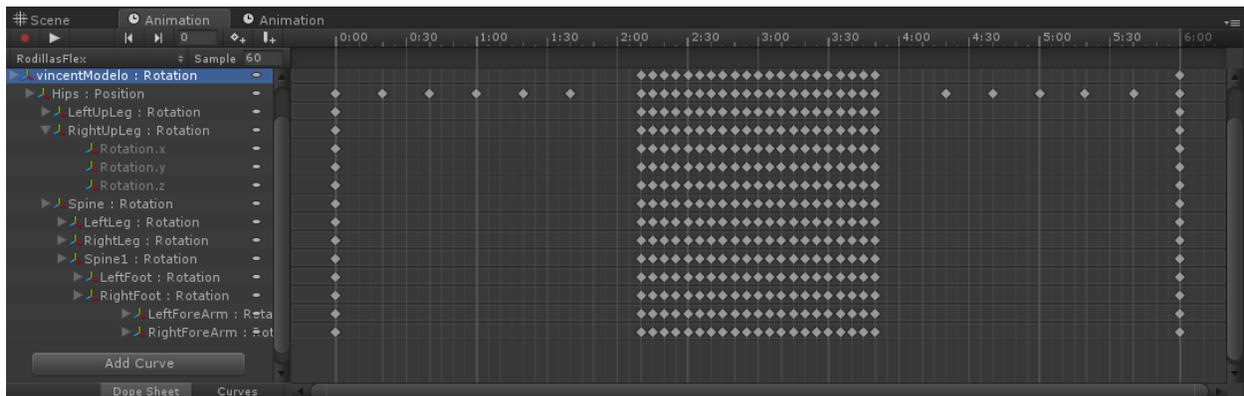


Figura 5.7: Ejemplo de animación del avatar. Flexión de rodillas.

En nuestro caso concreto, cada uno de los movimientos dura un total de seis segundos, con una pausa entre ellos implementada por código y de duración igual a un segundo, que podría modificarse fácilmente en caso de que se comprobara la necesidad de una mayor espera entre las distintas posturas a imitar por el paciente. La secuencia completa se realiza dos veces para evitar errores en la medida a causa de un despiste del paciente u otros factores, además de la posibilidad de que la segunda ronda sirva como incentivo al usuario para superar la medida obtenida en la anterior iteración. La secuencia de movimientos y su agrupación se muestra:

	Derecha	Izquierda	Bilateral
Abducción de hombro			
Flexión de hombro			
Abducción horizontal de hombro			
Extensión de hombro			

**Rotación externa de
hombro**



**Rotación interna de
hombro**



Flexión de tronco



Extensión de tronco



Abducción de cadera



Flexión de cadera			
Flexión ambas rodillas			
Paso lateral			
Paso hacia delante			

Tabla 5.3: Movimientos a evaluar

Una vez explicado cuáles son los ejercicios que se van a evaluar en el paciente, queda saber cómo se va a llevar a cabo esa evaluación, es decir, cómo hacer que el avatar siga el movimiento del paciente, y cómo saber los grados de amplitud o la distancia que este alcanza al realizar cada uno de los movimientos. Para ello necesitamos conocer cómo interpreta Kinect la información que recibe del cuerpo de paciente, y cómo traducir dicha información para obtener los ángulos o las distancias correspondientes a cada movimiento.

Como ya se comentó anteriormente, la nueva SDK de Kinect es capaz de detectar 25 articulaciones o *joints*, estructuradas de manera jerárquica del siguiente modo:

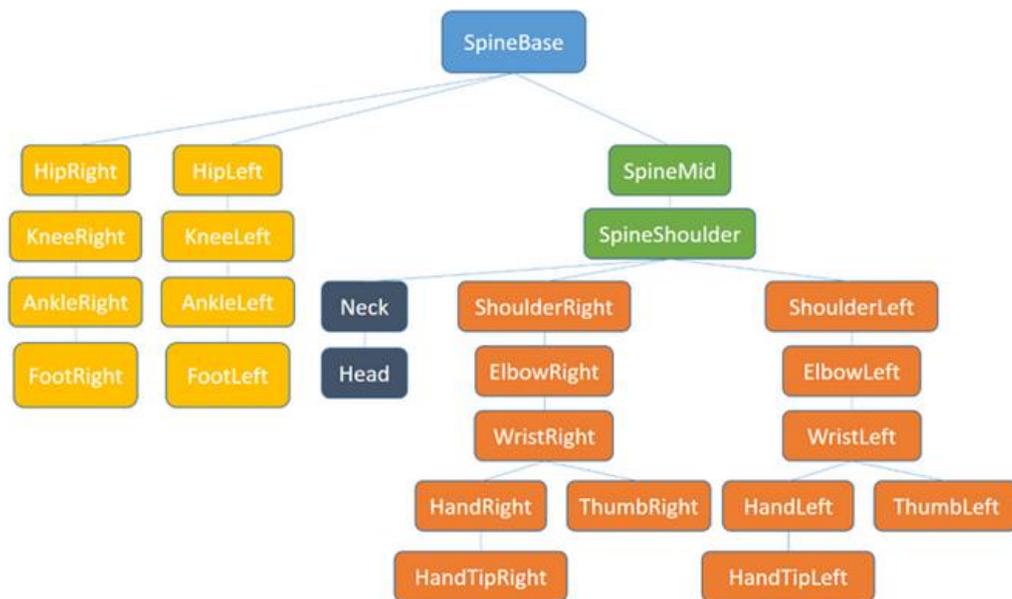


Figura 5.8: Jerarquía de los joints. Fuente: <http://peted.azurewebsites.net/wp-content/uploads/2015/04/hierarchy.png>

Y que en el cuerpo humano se corresponden con las siguientes posiciones:

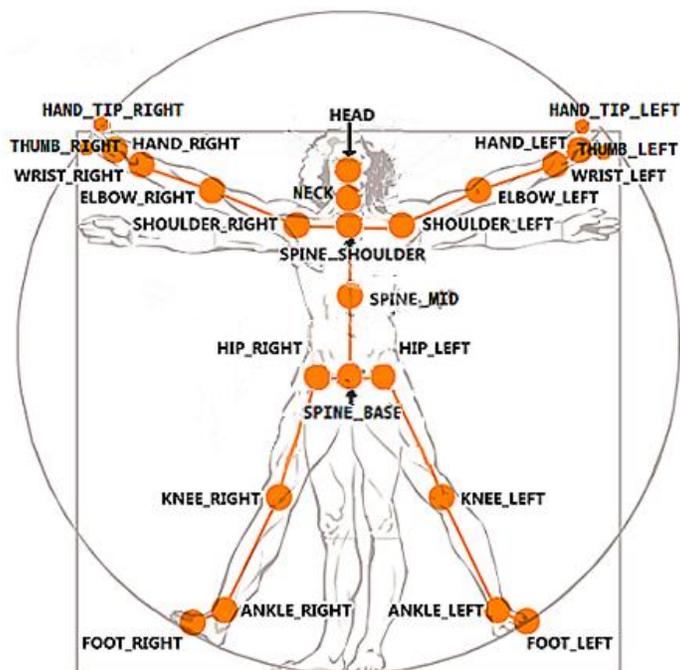


Figura 5.9: Posición de los joints. Fuente: <https://msdn.microsoft.com/en-us/library/microsoft.kinect.jointtype.aspx>

La orientación de estos *joints* viene dada en forma de cuaterniones, que se han convertido en una herramienta muy popular en el desarrollo de juegos 3D a la hora de rotar objetos. Los cuaterniones son una extensión de los números reales similar a la de los números complejos. La diferencia entre ellos reside en que mientras que los números complejos añaden a los números reales la unidad imaginaria i , tal que $i^2 = -1$, los cuaterniones añaden las unidades imaginarias i, j y k tales que $i^2 = j^2 = k^2 = ijk = -1$. Un cuaternión está formado por cuatro componentes: $q = w + xi + yj + zk$. Los tres últimos definen el eje en el que se producirá la rotación, mientras que el primero de ellos proporciona información sobre cuánto se va a rotar. Estos cuatro componentes permiten construir una matriz en la que están contenidos todos los giros posibles sin que en ellos se produzca *gimbal lock*, es decir, sin que se solapen dos ejes y por tanto se pierda un grado de libertad.

Para entender cómo realizar una rotación haciendo uso de los cuaterniones, lo más sencillo quizás sea ver un ejemplo:

Supongamos que tenemos el punto $P = (0,3,2)$ en el espacio de coordenadas y lo queremos girar 90° alrededor del eje X . La fórmula general para expresar este giro es (1) y el punto rotado P' se obtiene según la ecuación (2):

$$q = \cos(\theta/2) + (n_1\mathbf{i} + n_2\mathbf{j} + n_3\mathbf{k}) \cdot \text{sen}(\theta/2) \quad (1)$$

$$P' = q \cdot P \cdot \bar{q} \quad (2)$$

Por tanto, el primer paso es adaptar el eje y los grados para que encajen en la ecuación (1), y trasladar esos valores a dicha ecuación (5).

$$\text{Eje } X \text{ como vector: } \mathbf{n} = n_1\mathbf{i} + n_2\mathbf{j} + n_3\mathbf{k} = \mathbf{i} \quad (3)$$

$$\theta = 90^\circ \rightarrow \theta/2 = 45^\circ = \pi/4 \quad (4)$$

$$q = \cos(\pi/4) + \mathbf{i} \cdot \text{sen}(\pi/4) = \sqrt{2}/2 + \mathbf{i} \cdot \sqrt{2}/2 \quad (5)$$

A continuación, expresemos el punto $P = (0,3,2)$ como cuaternión para poder realizar el cálculo de P' , operamos y llegamos al resultado final (8).

$$P = 0 + 0\mathbf{i} + 3\mathbf{j} + 2\mathbf{k} \quad (6)$$

$$\begin{aligned}
q \cdot P &= \left(\frac{\sqrt{2}}{2} + i \cdot \frac{\sqrt{2}}{2} \right) \cdot (0 + 0i + 3j + 2k) \cdot \left(\frac{\sqrt{2}}{2} - i \cdot \frac{\sqrt{2}}{2} \right) \\
&= \left(3j \cdot \frac{\sqrt{2}}{2} + 2k \cdot \frac{\sqrt{2}}{2} + 3i \cdot j \cdot \frac{\sqrt{2}}{2} + 2i \cdot k \cdot \frac{\sqrt{2}}{2} \right) \cdot \left(\frac{\sqrt{2}}{2} - i \cdot \frac{\sqrt{2}}{2} \right) \\
&= \left(j \cdot \frac{3\sqrt{2}}{2} + k \cdot \left(\sqrt{2} + \frac{3\sqrt{2}}{2} \right) - j \cdot \sqrt{2} \right) \cdot \left(\frac{\sqrt{2}}{2} - i \cdot \frac{\sqrt{2}}{2} \right) \\
&= \left(j \cdot \frac{\sqrt{2}}{2} + k \cdot \frac{5\sqrt{2}}{2} \right) \cdot \left(\frac{\sqrt{2}}{2} - i \cdot \frac{\sqrt{2}}{2} \right) \\
&= j \cdot \frac{1}{2} + k \cdot \frac{10}{4} - i \cdot j \cdot \frac{1}{2} - i \cdot k \cdot \frac{10}{4} \\
&= j \cdot \left(\frac{1}{2} - \frac{5}{2} \right) + k \cdot \left(\frac{10}{4} + \frac{1}{2} \right) = (-2j + 3k)
\end{aligned} \tag{7}$$

$$P' = (-2j + 3k) \rightarrow \mathbf{P}' = (0, -2, 3) \tag{8}$$

En cuanto a cómo interpretar los datos que nos proporciona Kinect acerca de los *joints*, la información disponible desde el lanzamiento de la nueva SDK hasta el día de hoy es escasa y confusa. La única explicación oficial que se ha podido encontrar al respecto es que la estructura que define la orientación de un *joint*, *JointOrientation*, contiene la guiñada, el cabeceo y el alabeo (*yaw, pitch and roll*), que cada cuaternión es la orientación respecto al hueso padre, y que el fundamento de cada *joint* es el siguiente:

- Eje Y (verde): sigue la dirección del hueso.
- Eje Z (azul): perpendicular al hueso, representa el alabeo del *joint*.
- Eje X (naranja): perpendicular tanto al eje Y como al Z.

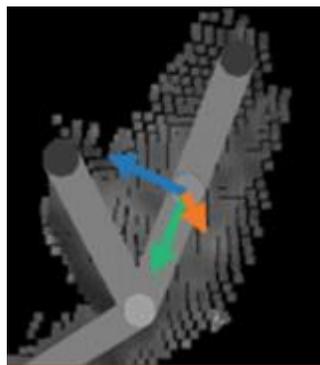


Figura 5.10: Sistema de coordenadas de cada *joint*

Sin embargo, esta información no siempre coincide con las conclusiones a las que han llegado algunos desarrolladores tras indagar por su cuenta. Sin duda, estas investigaciones y el

intercambio desinteresado de los resultados obtenidos (Microsoft Developer Network Forum, 2014) han resultado clave a la hora de poder avanzar en este tema.

Finalmente, para poder dotar de movimiento al avatar que representa al paciente, es necesario realizar una última transformación para que la información recibida por parte de Kinect sea correctamente interpretada por Unity y su sistema de coordenadas siguiendo la regla de la mano izquierda.

A la hora de hallar las distancias o los ángulos requeridos, se ha hecho uso tanto de los cuaterniones como de vectores en el espacio tridimensional. El criterio seguido para escoger una u otra opción se ha basado en elegir aquel método que proporcione una mayor estabilidad, y por tanto una mayor fiabilidad, de la medida tomada. Así pues, se comprobó que para movimientos como la flexión de la rodilla o la flexión y extensión del codo, lo más efectivo es hallar los dos vectores que tienen como origen o destino la articulación a analizar y calcular el ángulo que esos dos vectores forman en el espacio, ya que el uso de cuaterniones provocaba saltos indeseados durante la evaluación.

Volviendo al flujo de eventos en el módulo, la secuencia de movimientos se encuentra dividida en diez grupos diferentes: abducción de hombros, flexión de hombros, abducción horizontal de hombros, extensión de hombros, rotaciones externas, rotaciones internas, movimientos de tronco, movimientos de cadera, cuclillas y pasos. Cada uno de estos grupos se anuncia mediante el correspondiente título en la parte superior de la pantalla con el fin de ayudar al paciente a identificar el movimiento, y cada ángulo o distancia se muestra instantáneamente en los diferentes paneles situados a la izquierda de la pantalla. Como ya se ha mencionado anteriormente, la valoración completa consiste en la realización de dos iteraciones de la misma secuencia de movimientos, por lo que dichos paneles aparecen recuadrados con un color u otro dependiendo de en qué punto de la valoración se encuentre el paciente:

- Primera iteración:
 - Color magenta: movimiento actual en el que se encuentra el paciente.
 - Sin color: resto de movimientos.
- Segunda iteración:
 - Color verde: movimiento actual en el que se encuentra el paciente cuando la diferencia de ROM entre dicho movimiento y el correspondiente de la primera ronda es inferior al 10% .
 - Color rojo: Color verde: movimiento actual en el que se encuentra el paciente cuando la diferencia de ROM entre dicho movimiento y el correspondiente de la primera ronda es superior al 10% .
 - Sin color: movimiento aún no realizado.

Para obtener el máximo ROM alcanzado en cada movimiento, se va capturando *frame a frame* el valor del ángulo o la distancia obtenido por el paciente. Los siete primeros valores

correspondientes a los siete primeros *frames* se almacenan en un vector sin más tratamiento. A partir del séptimo resultado, cada uno de los siguientes será comparado con el mínimo valor almacenado en el *array*, y sustituido en caso de que sea mayor que dicho número. Por último, el valor máximo contenido en el vector no es considerado automáticamente como el resultado final, sino que se realiza la mediana entre los siete valores presentes en el mismo. El motivo de esta implementación es el de evitar posibles picos debido a ruido u otro tipo de errores en el sensor de Kinect, si bien en la práctica no se ha encontrado ningún salto alarmante que pudiera afectar la veracidad de los datos.

Cabe destacar que en el caso de un movimiento simétrico, serán dos los vectores que almacenen estos valores máximos, uno por articulación, y por tanto también serán dos las medianas que se calculen. Entre estas dos, nos quedaremos con la que sea menor, lo cual tiene sentido si pensamos que por lo general el paciente tendrá un lado sano y otro afecto, por lo que no tiene sentido quedarnos con el valor alcanzado por la extremidad sana al suponer una meta poco realista para la extremidad de movilidad reducida.

Por otro lado, en cada uno de los ejercicios de la evaluación se toman fotos al paciente con el objetivo de que en caso de obtener un valor numérico extraño como resultado, o una gran diferencia entre los valores de un mismo movimiento en las distintas iteraciones, el terapeuta pueda comprobar la postura y posición del paciente en cada momento y de este modo determinar si es necesaria la repetición de ese movimiento en concreto. Para ello, continuamente se compara el valor obtenido en el último *frame* con el vector de máximos que se describió anteriormente. Si este último dato es menor que todos los contenidos en el *array*, significa que el paciente ya ha alcanzado su máximo ROM y está avanzando hacia la posición de reposo, por lo que en ese momento se tomará la foto. Pudiera parecer que la foto carece de validez al no sacarse en el preciso momento en el que se alcanza el máximo ROM, sin embargo la diferencia de valores entre ambos instantes no varía más allá de las décimas de grado, precisión más que suficiente para el fin que nos ocupa.

Este total de 62 fotografías realizadas al paciente solo son accesibles una vez se haya completado totalmente la evaluación. Cuando esto ocurre, los avatares desaparecen de la escena dando paso a dos botones en formas de flechas que nos permiten desplazarnos por los distintos grupos de ejercicios. Durante este análisis, el terapeuta puede ver destacados en color rojo aquellos movimientos que difieren entre sí más de un 10%, y consultar las fotografías que se realizaron durante los mismos sin más que hacer *click* en su panel correspondiente. Si tras ver esta información se creyera que por una causa o por otra ha habido un error en la ejecución, el terapeuta podrá ordenar la repetición de ese movimiento en ese mismo instante sin necesidad de repetir toda la valoración de nuevo, y elegir entre almacenar el nuevo valor obtenido o descartarlo.



Figura 5.11: Fotos tomadas durante la evaluación y sus correspondientes valores

5.5.3. Tratamiento de resultados

Hasta ahora se ha explicado cómo hallar los resultados que nos interesan, pero nada se ha dicho acerca de qué se hace con ellos una vez obtenidos.

Una vez transcurridas las dos iteraciones del test, se comparan las medianas correspondientes al mismo movimiento y se almacena la que sea mayor en una posición determinada de un vector de longitud 31. Debido a que existe la posibilidad de que se repitan uno o más ejercicios de manera independiente, estas medianas pueden ser sustituidas por las que se obtengan en este punto si el terapeuta así lo cree oportuno.

Cuando ya se dé por finalizada la evaluación, es decir, se hayan completado las dos vueltas automáticas y se hayan repetido manualmente los movimientos deseados, este *array* de resultados definitivos se almacena en el sistema en dos archivos XML diferentes:

- El módulo de juego hará uso del primero de ellos para adaptar las siluetas creadas durante el módulo de administración al ROM máximo de cada paciente.
- El segundo de ellos, llamado *Evolucion.xml*, servirá para llevar un registro de la evolución del paciente a lo largo del tratamiento, pues almacenará los resultados definitivos de cada una de las sesiones de evaluación.

Volviendo a la Figura 5.4, vemos que existe un botón llamado “VER EVOLUCION”. Al hacer *click* en este botón, se nos abre un archivo Excel que fue copiado en la ruta perteneciente al paciente en cuestión cuando este fue agregado al sistema.

Este Excel consta dos Hojas. La primera de ellas contiene tantas gráficas como grupos de movimientos existen en la evaluación, además de un botón llamado “Actualizar”. Al presionar este botón, se ejecuta una Macro que se encarga de copiar en forma de tabla a la Hoja2 todos los datos almacenados en el archivo *Evolucion.xml*.

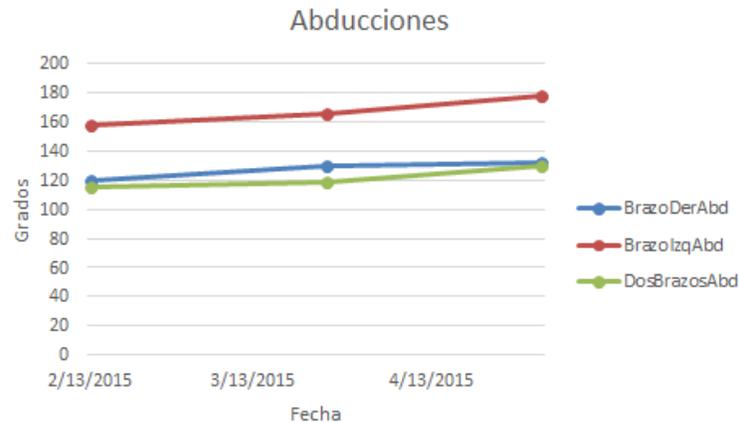


Figura 5.12: Ejemplo de gráfica generada con la Macro

Como cada una de las gráficas toma sus valores de esta segunda hoja, al presionarlo vemos que o bien se rellenan por primera vez las gráficas vacías, o bien se actualizan con los resultados de una o varias sesiones de valoración nuevas. El código concreto de la Macro desarrollado en VBA (*Visual Basic for Applications*) es el siguiente:

```
Private Sub Actualizar_Click()
Dim strTargetFile As String
Dim wb As Workbook
Dim myFile As String
Dim newSheet As Worksheet

Application.ScreenUpdating = False
Application.DisplayAlerts = False
myFile = "Evolucion.XML"
strTargetFile = ActiveWorkbook.Path & "\" & myFile
Set wb = Workbooks.OpenXML(FileName:=strTargetFile,
LoadOption:=xlXmlLoadImportToList)
Application.DisplayAlerts = True
wb.Sheets(1).UsedRange.Copy ThisWorkbook.Sheets("Hoja2").Range("A1")
wb.Close SaveChanges:=False
Application.ScreenUpdating = True

End Sub
```

Figura 5.13: Código correspondiente a la Macro creada en VBA

5.5.4. Estimación presupuestaria

La inversión necesaria para desarrollar un módulo de estas características se desglosa en:

- a) Gastos de equipamiento científico.
- b) Gastos de personal.
- c) Otros gastos.

En la tabla 5.4 se detallan las necesidades presupuestarias para llevar a cabo este proyecto. Para estimar la duración del contrato del personal, se ha tenido en cuenta el tiempo empleado en desarrollar este TFG sin contar la etapa de aprendizaje del motor de juegos Unity.

	€/unidad	Unidades	Subtotal
Equipamiento científico			1200,00
Cámara infrarroja tipo Kinect XBOX ONE	150	1	150,00
Adaptador de Kinect para Windows	50	1	50,00
PC de escritorio con tarjeta gráfica NVIDIA compatible 3D Vision	1000	1	1000,00
Personal			3450,00
Becario de investigación 3 meses con Seg.Social	3450	1	3450,00
Otros gastos (difusión, instalación en centros destino...)			200,00
SUBTOTAL			4850,00
Gastos indirectos (15%)			727,50
IMPORTE TOTAL			5577,50

Tabla 5.4: Estimación presupuestaria

Capítulo 6. Conclusiones y líneas futuras

En este último capítulo se remarcarán los objetivos que se han alcanzado tras la realización de este proyecto y se analizará si se ha logrado la finalidad requerida desde el inicio. Por otra parte, se explicarán las posibles futuras vías de mejora, puesto que a día de hoy el juego serio desarrollado es completamente funcional, pero el proyecto global es mucho más ambicioso y se pretende que pueda ser utilizado por diferentes centros de rehabilitación e incluso que los pacientes puedan ejercitarse desde casa.

6.1. Conclusiones

Todo el trabajo desarrollado ha tenido como principal objetivo el ser un complemento sólido en la rehabilitación de personas con problemas de movilidad, centrándose especialmente en aquellas que han sufrido un ictus. Es necesario recalcar el término complemento, ya que pretende ser una herramienta que ayude a los fisioterapeutas a realizar su trabajo con los pacientes, así como aliviar parte de su carga laboral al tratarse de una herramienta portable y fácil de utilizar por el usuario en su domicilio. Además, al registrar los resultados de cada uno de los pacientes tanto a lo largo del módulo de evaluación como durante el módulo de juego, facilita el manejo, almacenamiento y tratamiento de los mismos, y a su vez las gráficas que se generan permiten al terapeuta hacerse una idea de la evolución de un paciente determinado a través de un vistazo.

Por lo tanto, este es un claro ejemplo de la continua evolución de las técnicas de rehabilitación, haciendo uso cada vez más de elementos comunes que permiten abaratar los costes y como consecuencia, favorecen el desarrollo y la aceptación de estas novedosas técnicas por parte de los profesionales del sector.

En cuanto a los pacientes, la integración de complementos de rehabilitación como este juego serio permite añadir diversión e incluso cierta adicción a su terapia, influyendo así en la ilusión y la concentración a la hora de enfrentar sus sesiones. Todos estos factores difícilmente pueden lograrse siguiendo únicamente una rehabilitación tradicional.

Por todo esto, la Realidad Virtual aplicada a la rehabilitación está actualmente en auge. La Realidad Virtual nos permite desarrollar entornos atractivos orientados al paciente, y presentar en dicho entorno ejercicios que vayan más allá de los tradicionales. Sin embargo, también puede ser un arma de doble filo. Es necesario tener en cuenta que no todo el mundo está igual de familiarizado con las nuevas tecnologías, y que desde luego este porcentaje aumenta a medida que el rango de edad de las personas es mayor. Este es el motivo por el que es necesario un sistema sencillo e intuitivo y que no sature la pantalla de información para evitar despistes que afecten negativamente a la realización de la tarea encomendada al paciente.

Del mismo modo, las necesidades de los pacientes en rehabilitación condicionaron también la tecnología a utilizar para captar sus movimientos. Al tratarse de personas que presentan una movilidad reducida, inmediatamente hubo que descartar todos aquellos sistemas que obligaran al paciente a colocar en su cuerpo mecanismos que dificultaran sus movimientos, y se optó utilizar el sensor Kinect de Microsoft. Como ya se ha visto en el Capítulo 4 de esta memoria, se trata de un sensor que no necesita ni siquiera de la colocación de elementos de *tracking* en el paciente. Los únicos requisitos que se han encontrado a través de las diferentes pruebas realizadas son que el usuario no vista prendas excesivamente anchas durante la sesión, así como la necesidad de contraste entre la ropa y el fondo de la habitación.

6.2. Líneas futuras

Como ya se ha mencionado, este trabajo pretende ser la base de algo mucho más complejo que abarque múltiples aspectos relativos a los problemas de movilidad de los pacientes. Sin embargo, debido a la limitación de tiempo de un Trabajo Fin de Grado, no se han podido alcanzar.

En primer lugar, sería necesario que el sistema solicitara desde su inicio un nombre de usuario y contraseña. Esto permitiría diferenciar entre los distintos actores, pues tal y como se ha visto no todos deberían tener acceso a todos los módulos, además de evitar que personas no autorizadas tuvieran acceso a datos y estadísticas privadas de cada paciente.

Por otra parte, desde la Fundación ASPAYM se nos comunicó lo útil que sería poder calcular y proyectar el centro de gravedad de los pacientes. Desde el Grupo de Telemática e Imagen hubo un intento de desarrollar esta funcionalidad, pero de nuevo por falta de tiempo no pudo ser implementada en la solución que ahora se presenta. Por consiguiente, queda pendiente un análisis más detenido del cálculo del centro de masas, así como de comprobar su precisión para así conocer la fiabilidad de sus resultados.

En relación al centro de gravedad, aunque no resultara una herramienta válida para un seguimiento riguroso al paciente, sí podría utilizarse a la hora de desarrollar nuevos niveles en el módulo de juego. Estos niveles indicarían, además de la postura a realizar, cómo distribuir el peso entre las extremidades inferiores. Debido a que los pacientes afectados por un ictus tienden a desplazar el peso hacia el lado sano, este tipo de ejercicios serían muy útiles a la hora de retar al paciente a adoptar posturas que le generen un pequeño desequilibrio.

Siguiendo con la creación de nuevos niveles, otra dificultad añadida sería incluir cierta dificultad cognitiva, es decir, que el paciente viera dos siluetas en lugar de una y tuviera que decidir si adoptar una postura u otra dependiendo de, por ejemplo, el resultado de una sencilla operación matemática.

También se podría añadir la mecánica de juego de puntos. Esto permitiría que el paciente fuera ganando puntos en función de la precisión con la que haya imitado las siluetas durante el juego y/o el tiempo que haya dedicado a entrenar con el sistema. Dichos puntos podrían ser canjeados por el desbloqueo de nuevos escenarios o nuevos personajes, lo que ayudaría a incrementar la motivación y reducir la sensación de monotonía.

Pero sin duda, la tarea más importante que queda por desarrollar es adaptar este sistema a una plataforma completa de tele-rehabilitación. Esto permitiría mejorar la calidad de vida de un número muy elevado de pacientes con diferentes patologías, al tratarse de un sistema económico donde el paciente no tendría que desplazarse hasta una clínica, factor a tener muy en cuenta debido a la cantidad de personas que viven hoy en día alejados de los núcleos de población urbana.

Actualmente, el sistema se encuentra instalado en la Fundación ASPAYM, donde se está realizando el correspondiente estudio con pacientes. Una vez este haya finalizado, se deberán recoger las sensaciones, sugerencias, y necesidades de mejora que tanto terapeutas como pacientes crean oportunas. Solo tras este paso se podrá empezar a trabajar en la plataforma de tele-rehabilitación. Esta solución requerirá de la utilización del paradigma de comunicaciones cliente-servidor y servicios basados en Web. En la Figura 6.1 se distinguen tres partes diferenciadas:

- Servidor: contendrá el núcleo del sistema, encargado de almacenar los historiales clínicos de los pacientes, junto con toda la información recopilada durante las sesiones de rehabilitación. Este servidor estará alojado en lugar seguro a fin de garantizar la protección de los datos de los pacientes. El servidor proporcionará además redundancia de almacenamiento para protegerse ante posibles fallos de *hardware*.
- Rehabilitador: cada uno de los puestos de rehabilitador se conectará al Servidor a través de una comunicación segura. El puesto de un rehabilitador dispondrá de una aplicación para el seguimiento de los pacientes que permitirá la edición personalizada de terapias y su envío al paciente correspondiente. Opcionalmente, este puesto puede incluir una cámara Kinect.
- Paciente: el puesto de un paciente necesitará obligatoriamente de una cámara Kinect. Este puesto consistirá en un pequeño ordenador personal, que pueda ser enchufado a una televisión o proyector, a fin de lograr una experiencia de usuario más satisfactoria que si se emplease un monitor de ordenador.

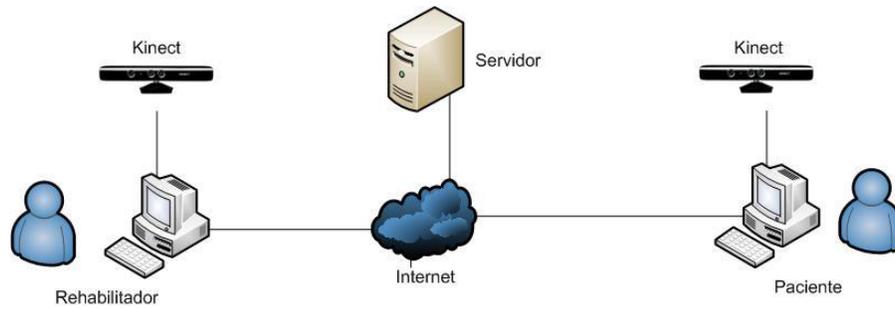


Figura 6.1: Diagrama de interconexión

Mediante el uso de este sistema, se pretende que los fisioterapeutas sean capaces de interpretar si el paciente está realizando los ejercicios correctamente en función de los movimientos de las distintas zonas del cuerpo y sin necesidad de recibir dicha información en vídeo, ya que esto supondría una importante ralentización en el envío de datos por parte del usuario, y una enorme sobrecarga para el servidor a la hora de almacenarlos.

Bibliografía

Adobe Systems Incorporated. *Adobe AIR / Features*. Recuperado el 19 de abril de 2015 de Adobe Systems Incorporated: <http://www.adobe.com/es/products/air/features.html>

Adobe Systems Incorporated. *Adobe AIR Help / Release Notes*. Recuperado el 19 de abril de 2015 de Adobe Systems Incorporated: <https://helpx.adobe.com/air/air-releasenotes.html>

Alexander C.H., Mirjam de Haart, Ilse J.W., Duysens, J.(2005). *A review of standing balance recovery from stroke*. *Gait & Posture* 22: 267-281.

Alvarado, D., Barría, S., Cordella, R., Lechuga, A. *Manual de métodos de optimización para proyectos 3D*. Recuperado el 19 de abril de 2015 de Optimización 3D: http://www.optimizacion3d.info/descargas/metodos_de_optimizacion_para_proyectos_3d_version_web.pdf

Álvarez Sabín, J. (2008). *Mortalidad hospitalaria por ictus*. Recuperado el 5 de mayo de 2015 de la Revista Española de Cardiología: <http://www.revespcardiol.org/es/in-hospital-mortality-in-stroke-patients/articulo/13126039/>

Anon. (2014). *La historia de los videojuegos*. Recuperado el 18 de abril de 2015 de El otro lado: http://www.elotrolado.net/wiki/Historia_de_los_videojuegos

Anon. (Septiembre 2014). *Documentation, Unity scripting languages and you*. Recuperado el 19 de abril de 2015 de Unity3d: <http://blogs.unity3d.com/es/2014/09/03/documentation-unity-scripting-languages-and-you/>

Anon. *Géneros de los videojuegos*. Recuperado el 18 de abril de 2015 de El otro lado: http://www.elotrolado.net/wiki/G%C3%A9neros_de_los_videojuegos

Anon. *Videojuegos: Historia, Tipos de Jugadores, etc.* Recuperado el 18 de abril de 2015 de Taringa!: <http://www.taringa.net/posts/info/17015983/Videojuegos-Historia-Tipos-de-Jugadores-etc.html>

ASUSTeK Computer Inc. *Xtion PRO LIVE. Descripción*. Recuperado el 19 de abril de 2015 de ASUS: http://www.asus.com/es/Multimedia/Xtion_PRO_LIVE/overview/

ASUSTeK Computer Inc. *Xtion PRO LIVE. Especificaciones*. Recuperado el 19 de abril de 2015 de ASUS: http://www.asus.com/es/Multimedia/Xtion_PRO_LIVE/specifications/

Bower, K.J., Clark, R.A., McGinley, J.L., Martin, C.L., Miller, K.J. (2014). *Clinical feasibility of the Nintendo Wii for balance training post-stroke: a phase II randomized controlled trial in an inpatient setting*. *Clin Rehabil* 2014 28: 912.

Cocos2d-x. *About Cocos2d-x*. Recuperado el 19 de abril de 2015 de cocos2d-x: <http://www.cocos2d-x.org/about>

Cocos2d-x. *Cocos2d-x*. Recuperado el 19 de abril de 2015 de Cocos2d-x: <http://www.cocos2d-x.org/wiki/Cocos2d-x>

Cocos2d-x. *Download*. Recuperado el 19 de abril de 2015 de Cocos2d-x: <http://www.cocos2d-x.org/download>

Cocos2d-x. *Showcase*. Recuperado el 19 de abril de 2015 de Cocos2d-x: <http://www.cocos2d-x.org/games/308>

Conneally, T. (2011). *Lenovo: next-generation video game console competitor?* Recuperado el 19 de abril de 2015 de Betanews: <http://betanews.com/2011/05/09/lenovo-next-generation-video-game-console-competitor/>

Dabingnn (Enero 2015). *Latest Stable Version of Cocs2d-x Released: V3.4 Final*. Recuperado el 19 de abril de 2015 de Cocos2d-x: <http://www.cocos2d-x.org/news/416>

Definición (2015). *Definición de plugin*. Recuperado el 19 de abril de 2015 de Definición.de: <http://definicion.de/plugin/>

Deutsch, J.E., Brettler, A., Smith, C., Welsh, J., John, R., Guarrera-Bowlby, P., Kafri, M. (2011). *Nintendo Wii Sports and Wii Fit game analysis, validation, and application to stroke rehabilitation*. Top Stroke Rehabil. 2011 Nov-Dec; 18 (6): 701-19.

Epic Games, Inc. (2015). *Frequently Asked Questions (FAQ)*. Recuperado el 19 de abril de 2015 de Unreal Engine: <https://www.unrealengine.com/faq>

Epic Games, Inc. (2015). *Unreal Engine Features*. Recuperado el 19 de abril de 2015 de Unreal Engine: <https://www.unrealengine.com/unreal-engine-4>

Epic Games, Inc. (marzo 2013). *Unreal Engine "Infiltrator" Real-Time Demo*. Recuperado el 19 de abril de 2015 de Youube: <https://www.youtube.com/watch?v=dO2rM-l-vdQ>

Facultat d'Informàtica de Barcelona. *Retro Informàtica. El pasado del futuro*. Recuperado el 18 de abril de 2015 de la Facultad de Informàtica de Barcelona: <http://www.fib.upc.edu/retro-informatica/historia/videojocs.html>

Flórez, M.T. (2000). *Intervenciones para mejorar la función motora en el paciente con ictus*. Rehabilitación (Madr) 2000; 34(6): 423-437.

Fung, J., Perez, C.F. (2011). *Sensorimotor enhancement with a mixed reality system for balance and mobility rehabilitation*. 33rd Annual International Conference of the IEEE EMBS. Boston, Massachusetts USA.

Gambin, A. (Diciembre 2014). *El videojuego lidera la industria del ocio audiovisual superando al cine o la música*. Recuperado el 19 de abril de 2015 de Blasting News España: <http://es.blastingnews.com/economia/2014/12/el-videojuego-lidera-la-industria-del-ocio-audiovisual-superando-al-cine-o-la-musica-00217253.html>

Gamificación S.L. (2013). *Gamificación*. Recuperado el 18 de abril de 2015 de Gamificación: <http://www.gamificacion.com>

Gil Núñez, A., Egio, J, Larracochea, J., Vila, N. *El ictus. ¿Qué es, cómo se previene y trata?* Recuperado el 5 de mayo de 2015 del Grupo de Estudio de Enfermedades Cerebrovasculares de la SEN: http://www.ictussen.org/files3/El_ictus_que_es_como_prevenir&tratar.pdf

ISFE. (Noviembre 2012). *Videogames in Europe: Consumer Study*. Recuperado el 18 de abril de 2015 de ISFE: http://www.isfe.eu/sites/isfe.eu/files/attachments/spain_-_isfe_consumer_study.pdf

Ki Hun Cho, Kyoung Jin Lee, Chang Ho Song (2012). *Virtual-reality balance training with a video-game system improves dynamic balance in chronic stroke patients*. *Tohoku J. Exp. Med.*, 2012 Sep, 228 (1), 69-74.

Lloréns, R., Colomer-Font, C., Alcañiz, M., Noé-Sebastián, E. (2013). *BioTrak. análisis de efectividad y satisfacción de un sistema de realidad virtual para la rehabilitación del equilibrio en pacientes con daño cerebral*. *Neurología*. 2013; 28(5): 268-275.

López, D. (2014). *Videojuegos con Realidad Virtual en Rehabilitación*. Recuperado el 18 de abril de 2015 de Slideshare: <http://es.slideshare.net/DavidLpez60/9ccc>

López, D., Marful, H., Gómez, D., González, S., Casares, R., Abella, M.A. *Motores gráficos*. Recuperado el 19 de abril de 2015 de Universidade da Coruña: <http://sabia.tic.udc.es/gc/Contenidos%20adicionales/trabajos/MotoresGraficos/MotoresGraficos/index.html>

Lynda.com, Inc. (Marzo 2013). *Understanding how Adobe AIR works*. Recuperado el 19 de abril de 2015 de YouTube: <https://www.youtube.com/watch?v=gOoG654QOwA>

Marketwire (Julio 2011). *Unity Technologies Lands \$12 Million in Series B Funding Led by WestSummit Capital and iGlobe Partners*. Recuperado el 19 de abril de 2015 de Marketwired: <http://www.marketwired.com/press-release/unity-technologies-lands-12-million-series-b-funding-led-westsummit-capital-iglobe-partners-1540593.htm>

Microsoft (2015). *Kinect for Windows features*. Recuperado el 19 de abril de 2015 de Microsoft: <http://www.microsoft.com/en-us/kinectforwindows/meetkinect/features.aspx>

Microsoft (2015). *Set up the hardware*. Recuperado el 19 de abril de 2015 de Microsoft: http://www.microsoft.com/en-us/kinectforwindows/purchase/sensor_setup.aspx

Microsoft Developer Network Forum (2014). *How to interpret JointOrientation data*. Recuperado el 19 de junio de 2015 de: <https://social.msdn.microsoft.com/Forums/en-US/a87049b5-7842-4c17-b776-3f6f4260c801/how-to-interpret-jointorientation-data?forum=k4wv2devpreview>

Microsoft. *System Requirements*. Recuperado el 19 de abril de 2015 de Microsoft Developer Network: <https://msdn.microsoft.com/en-us/library/hh855359.aspx#ID4EMB>

Minoru3D (2015). *Say Hi to the Minoru 3D Webcam! :*). Recuperado el 19 de abril de 2015 de Minoru3D: <http://www.minoru3d.com/>

Mittal, R. (Abril 2014). *AIR app installs cross a billion*. Recuperado el 19 de abril de 2015 de Adobe AIR and Adobe Flash Team Blog: <http://blogs.adobe.com/flashplayer/2014/04/air-app-installs-cross-a-billion.html>

Pérez Martínez, F. J. (2011). *Presente y futuro de la tecnología de la realidad virtual. Creatividad y sociedad*, 16.

Quesada, R. (2012). *Cocos2d: past, present, future*. Recuperado el 19 de abril de 2015 de YouTube: <https://www.youtube.com/watch?v=1sZuOOQj7gU>

Quesada, R. (2013). *Talk on Cocos2d-x v3.0*. Recuperado el 19 de abril de 2015 de YouTube: https://www.youtube.com/watch?v=wbf2_5INxEI

Ray, O. (Junio 2012). *Una docena de conceptos que deberías conocer sobre gamificación*. Recuperado el 18 de abril de 2015 de Una docena de: <http://unadocenade.com/una-docena-de-conceptos-que-deberias-conocer-sobre-gamificacion/>

Rose, F. D., Brooks, B. M., Rizzo, A. (2005). *Virtual reality in brain damage rehabilitation: Review, Cyberpsychology & Behavior*, 8(3):241-263.

Rose, M. (Marzo 2008). *Unity engine to power iPhone gaming*. Recuperado el 19 de abril de 2015 de Engadget: <http://www.tuaw.com/2008/03/31/unity-engine-to-power-iphone-gaming/>

Saed, S. (2015). *Steam has over 125 million active users, 8.9M concurrent peak*. Recuperado el 18 de junio de 2015 de VG24/7: <http://www.vg247.com/2015/02/24/steam-has-over-125-million-active-users-8-9m-concurrent-peak/>

Sieprawski, B. (Febrero 2015). *Unreal Engine 4.7 released!* Recuperado el 19 de abril de 2015 de Unreal Engine: <https://www.unrealengine.com/blog/unreal-engine-47-released>

SoftKinetic (2013). *DS325 Datasheet*. Recuperado el 19 de abril de 2015 de SoftKinetic: http://www.softkinetic.com/Portals/0/Documents/PDF/WEB_20130527_SK_DS325_Datasheet_V4.0.pdf

Taborda, W. (2013). *Clasificación-géneros de los videojuegos*. Recuperado el 18 de abril de 2015 de la Facultad de Diseño y Comunicación de Palermo: http://fido.palermo.edu/servicios_dyc/blog/docentes/detalle_tp.php?id_docente=80946&id_blog=10680

Thomsen, M. (febrero 2010). *History of the Unreal Engine*. Recuperado el 19 de abril de 2015 de IGN: <http://www.ign.com/articles/2010/02/23/history-of-the-unreal-engine>

Unidad de evaluación de Tecnologías Sanitarias de la Agencia Laín Entralgo de la Comunidad de Madrid (2009). *Consejos y cuidados tras un ictus. Información para pacientes y familiares*. Recuperado el 5 de mayo de 2015 del Servicio Gallego de Salud: http://www.sergas.es/docs/EGSPC/folleto/Ictus_AP_paciente.pdf

Unity Technologies (2014). *El software líder a nivel mundial en la industria de los juegos*. Recuperado el 19 de abril de 2015 de Unity3d: <http://unity3d.com/es/public-relations>

Unity Technologies. *Obtener Unity*. Recuperado el 19 de abril de 2015 de Unity3d: <http://unity3d.com/es/get-unity>

Unity Technologies. *Unity 4.6*. Recuperado el 19 de abril de 2015 de Unity3d: <http://unity3d.com/es/unity/whats-new/unity-4.6>

Universidade da Coruña. *MoCap (captura de movimientos)*. Recuperado el 19 de abril de 2015 del Departamento de Tecnologías de la Información y las Comunicaciones, Universidade da Coruña: <http://sabria.tic.udc.es/gc/Contenidos%20adicionales/trabajos/Peliculas/Mocap/tecnol.htm>

Xataka (octubre 2013). *La evolución de Kinect y la importancia real de Microsoft Research*. Recuperado el 19 de abril de 2015 de Xataka Windows: <http://www.xatakawindows.com/xbox/la-evolucion-de-kinect-y-la-importancia-de-microsoft-research>

Yang, S., Hwang, W-H., Tsai, Y-C., Liu, F-K., Hsieh, L-F., Chern, J-S. (2011). *Improving balance skills in patients who had stroke through virtual reality Treadmill Training*. Am J Phys Med Rehabil 2011; 90: 969-978.

Zone, R. (2013). *3DIY: Stereoscopic Moviemaking on an Indy Budget*. Burlington, MA: Focal Press.