



*ugr* | Universidad  
de **Granada**

TRABAJO FIN DE GRADO  
GRADO EN INGENIERÍA INFORMÁTICA

# Aplicación educativa para la enseñanza de formulación química

---

Prototipo de serious game para la enseñanza de formulación  
química utilizando el dispositivo Leap Motion como  
herramienta de interacción

**Autor**

Juan Ramón Gómez Berzosa

**Director**

Marcelino Cabrera Cuevas



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE  
TELECOMUNICACIÓN

—  
Granada, Junio de 2018







# **Aplicación educativa para la enseñanza de formulación química: Prototipo de serious game para la enseñanza de formulación química usando el dispositivo Leap Motion como herramienta de interacción**

Juan Ramón Gómez Berzosa

**Palabras clave:** Interacción Gestual, Educación, Formulación Química.

## **Resumen**

En pleno auge tecnológico, con miles de dispositivos de última tecnología rodeándonos, han surgido nuevos métodos de apoyo a la enseñanza que nos permiten aprovechar el gran interés y atracción que provocan estos dispositivos en todos los jóvenes para captar su atención y transmitirles conocimientos. La finalidad de este proyecto está enfocada en utilizar estas nuevas tecnologías para captar la atención y motivación de los alumnos de forma que puedan aprender formulación química de una manera más entretenida y llamativa para ellos. Se desarrollará una aplicación educativa que permita aprender y aplicar conceptos de formulación química y que pueda servir como base para un futuro “Serious game”. Para ello, usaremos el motor de juegos “Unity” e integraremos las funcionalidades necesarias para poder manejar la aplicación con el dispositivo de interacción gestual “Leap Motion”, permitiendo a los usuarios tener una interacción hombre - máquina más dinámica, natural e intuitiva usando los movimientos y gestos realizados con sus propias manos.

Al final del proyecto, lograremos alcanzar tres objetivos principalmente: desarrollo de una aplicación educativa, aprendizaje de uso del motor de juegos Unity e integración de Leap Motion como modo de interacción con una aplicación.



# **Educational application for the teaching of chemical formulation: Prototype of serious game for the teaching of chemical formulation using the Leap Motion device as an interaction tool**

Juan Ramón Gómez Berzosa

**Keywords:** Gestural Interaction, Education, Chemical Formulation

## **Abstract**

In this time of booming technology with thousands of latest technology devices around us, new methods of teaching support have emerged which allow us to take advantage of the great interest and attraction that these devices have in young people to catch their attention and transmit knowledge. The aim of this project is to use these new technologies to catch the attention and interest of students and teach them about chemical formulation in a more entertaining and appealing way. We shall develop an educational application which allows learning and applying chemical formulation concepts and that can be used as a basis for a future serious game. For this purpose, we shall use the “Unity” game engine and we shall integrate the necessary functionalities to manage the application with the most dynamic, natural and intuitive man-machine interaction device, using movements and gestures made with our own hands.

At the end of the project, we shall achieve three main goals: developing an educational application, learning how to use the game engine Unity and integrate Leap Motion as an interaction mode with an application.



---

Yo, **Juan Ramón Gómez Berzosa**, alumno de la titulación Grado en Ingeniería Informática de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI 26517239X, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: Juan Ramón Gómez Berzosa

Granada a 18 de Junio de 2018.



---

D. **Marcelino Cabrera Cuevas**, Profesor del Área de Lenguajes y Sistemas de Información (LSI) de la Universidad de Granada.

**Informa:**

Que el presente trabajo, titulado *Aplicación educativa para la enseñanza de formulación química, prototipo de serious game para la enseñanza de formulación química usando el dispositivo Leap Motion como herramienta de interacción*, ha sido realizado bajo su supervisión por **Juan Ramón Gómez Berzosa**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a 18 de Junio de 2018.

**El director:**

**Marcelino Cabrera Cuevas**



# Agradecimientos

El tiempo pasa más rápido de lo deseado, parece que fue ayer cuando empecé a cumplir el sueño que tenía desde pequeño. Después de cuatro años de mucho esfuerzo, puedo dar por concluido uno de los grandes objetivos de mi vida, graduarme en Ingeniería Informática. Puedo decir que el tiempo es capaz de todo, de lo mejor y de lo peor, pero como dicen, todo esfuerzo tiene su recompensa y hoy estamos aquí para echar la vista hacia atrás y recordar a todas esas personas que nos han acompañado de una u otra manera en el camino de alcanzar este gran paso para nuestro futuro.

En primer lugar me gustaría dar las gracias a mi madre, por estar siempre detrás mía siendo mi propia conciencia y ofreciéndome todo su apoyo de manera incondicional, a mi padre, por enseñarme a que nunca tengo que dejar de esforzarme si quiero conseguir las cosas y a mi hermano, por mirarme siempre como un ejemplo a seguir y estar a mi lado en todo momento, porque sin ellos, sin su constante apoyo y cariño, esto no hubiera sido posible. En segundo lugar, querría agradecer al resto de mi familia y especialmente a mis abuelos, por que sin ellos esto tampoco hubiera sido posible, haciendo una mención muy especial a mi abuelo Pedro, que desde el cielo estoy seguro que está muy orgulloso de mi. En tercer lugar, agradecer a mis amigos, los de siempre y todos los que he conocido a lo largo de esta aventura. Sobre todo, agradecer a mi grupo de compañeros de carrera, bueno mejor dicho mis compañeros de vida desde que nos conocimos, por ayudarnos mutuamente y pasar innumerables días y noches en vela para sacar adelante las asignaturas, por aguantarnos, por ser unos personajes y por que no los voy a olvidar jamás, han sido de lo mejor que me llevo de estos años.

Por último y no menos importante, dar las gracias a mi tutor, Marcelino, por hacer todo lo posible, incluso no encontrándose en las mejores condiciones, para que realizara un buen proyecto y porque aprendiera todo lo posible desarrollándolo. También, dar las gracias a todos los profesores del grado, porque todos me han ayudado a formarme y ser una persona diferente de la que empezó hace cuatro años, ya que una carrera no sólo nos forma como profesionales, nos invita a forjar un sin fin de vínculos, experiencias y emociones que nos ayudan a crecer como personas.



# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Descripción del proyecto y motivación . . . . .	2
1.2. Inspiración para el desarrollo del proyecto . . . . .	4
<b>2. Planificación</b>	<b>5</b>
2.1. Análisis Hardware . . . . .	5
2.1.1. Dispositivos de Interacción Gestual . . . . .	5
2.1.2. Gafas de Realidad Virtual . . . . .	14
2.2. Análisis Software . . . . .	17
2.2.1. Motor de juego . . . . .	17
2.3. Metodología para el desarrollo del sistema . . . . .	19
2.3.1. Iteraciones (sprints) . . . . .	20
<b>3. Estado del arte</b>	<b>25</b>
3.1. La química y la educación . . . . .	25
3.2. Interacción hombre - máquina . . . . .	26
3.2.1. Historia del HCI . . . . .	27
3.2.2. Tipos de HCI . . . . .	27
3.3. Los videojuegos educativos . . . . .	28
3.3.1. Factores imprescindibles en un videojuego educativo .	30
3.3.2. Ejemplos de videojuegos educativos . . . . .	31
3.4. Aplicaciones actuales . . . . .	33
3.5. Crítica al estado del arte . . . . .	34
<b>4. Análisis de requisitos</b>	<b>37</b>
4.1. Descripción del sistema . . . . .	37
4.2. Actores . . . . .	38
4.3. Objetivos . . . . .	38
4.4. Requisitos Funcionales . . . . .	39
4.5. Requisitos de Información . . . . .	40
4.6. Requisitos No Funcionales . . . . .	41
4.7. Casos de uso . . . . .	42
<b>5. Diseño de clases</b>	<b>53</b>

<b>6. Implementación</b>	<b>61</b>
6.1. Dimensión de desarrollo de la aplicación . . . . .	61
6.2. Integración de Leap Motion . . . . .	61
6.3. Gestión de la base de datos . . . . .	62
6.4. Desarrollo de la aplicación . . . . .	63
6.4.1. Escena Menú . . . . .	64
6.4.2. Escena Ejercicios . . . . .	65
6.4.3. Escena Tabla . . . . .	66
6.4.4. Escena Grupos Tabla Periódica . . . . .	67
6.4.5. Escena Formulación . . . . .	69
<b>7. Pruebas</b>	<b>73</b>
7.1. Entorno de desarrollo . . . . .	73
7.2. Pruebas internas . . . . .	74
7.3. Pruebas externas . . . . .	75
<b>8. Conclusiones</b>	<b>77</b>
8.1. Asignaturas relacionadas . . . . .	77
8.2. Valoración de los objetivos . . . . .	78
8.3. Conclusiones del desarrollo y la aplicación . . . . .	79
8.4. Valoración personal . . . . .	80
<b>9. Trabajos Futuros</b>	<b>81</b>
9.1. Mejoras en la aplicación . . . . .	81
9.2. Realidad Virtual . . . . .	82
9.3. Serious Game . . . . .	82
<b>Bibliografía</b>	<b>87</b>
<b>A. Manual de usuario</b>	<b>89</b>
A.1. Ejemplo de uso . . . . .	91
A.2. Diagrama de flujo de escenas de la aplicación . . . . .	97
<b>B. Autorizaciones para la grabación del vídeo</b>	<b>99</b>
<b>C. Estructura de los ficheros JSON de datos</b>	<b>115</b>

# Índice de cuadros

2.1. Tabla comparativa de dispositivos de interacción gestual . . . . .	12
2.2. Comparativa entre Unity y Unreal . . . . .	18
4.1. Actor Alumno . . . . .	38
4.2. CU_01: Seleccionar elemento de la tabla periódica . . . . .	43
4.3. CU_02: Seleccionar un ejercicio . . . . .	44
4.4. CU_03: Sacar átomo del inventario . . . . .	45
4.5. CU_04: Meter átomo en el inventario . . . . .	46
4.6. CU_05: Eliminar átomo . . . . .	47
4.7. CU_06: Enlazar átomos . . . . .	48
4.8. CU_07: Romper enlace átomos . . . . .	49
4.9. CU_08: Comprobar solución ejercicio . . . . .	50
4.10. CU_09: Comprobar solución libre . . . . .	51



# Capítulo 1

## Introducción

Normalmente cuando una persona habla de juegos y en especial de videojuegos, tenemos asociados a este concepto una imagen de diversión o entretenimiento, por lo que podemos llegar a pensar que son malos o incluso una pérdida de tiempo. Sin embargo, en la actualidad se ha demostrado que no sólo sirven como pasa tiempo, sino que también nos pueden ayudar a desarrollar nuestras habilidades. Se ha podido comprobar que las personas que juegan a algunos juegos tienen más desarrolladas habilidades como la rapidez de sus reflejos, coordinación o el sentido de liderazgo en comparación a otras personas que no juegan, por ejemplo en los juegos de shooters. [1]

Siguiendo en esta línea, en la actualidad es cada vez más frecuente utilizar los videojuegos como objeto de aprendizaje o la práctica de habilidades, sin tener como objetivo principal la diversión o el entretenimiento. Este tipo de juegos se denominan “**serious games**”, o en español, juegos educativos.

Hoy en día, la mayoría de los jóvenes “nacen con una videoconsola debajo del brazo” o con dispositivos como “tablets” o “smartphones”, lo que les permite estar familiarizados con los videojuegos desde muy pequeños y por tanto captar su atención. Por tanto, los serious game intentan aprovechar esta devoción de los jóvenes, e incluso de los mayores, para llegar a ellos e intentar así captar su atención para el aprendizaje o la práctica de habilidades desde un modo más “familiar” para ellos. A este método de enseñanza se le conoce como “**game-based learning**” y su uso junto a los serious game se está extendiendo en escuelas de primaria, universidades e incluso grandes empresas. [2]

Nuestro proyecto consiste en el desarrollo de **un prototipo de serious game** que implementa algunas de las mecánicas básicas del futuro serious

game, por lo que lo consideraremos a partir de ahora como una aplicación educativa. Una **aplicación educativa** es un recurso, material o programa multimedia, que es ejecutado en un dispositivo electrónico y tiene como fin que se pueda usar como una herramienta de apoyo en la enseñanza.[3]

La parte que hará diferir nuestro proyecto de otros juegos es el enfoque más dinámico e interactivo que le daremos al emplear un **dispositivo de interacción hombre máquina** capaz de detectar nuestros gestos en 3D, lo que hará más natural e intuitiva la interacción.

Cabe destacar se ha optado por realizar una aplicación educativa en vez de el desarrollo completo del juego educativo por que se necesitarían cumplir una serie de requisitos adicionales, hacer un gran desembolso económico y aportarle una gran cantidad de tiempo extra.

### 1.1. Descripción del proyecto y motivación

Normalmente, cuando empezamos en el colegio tenemos asignaturas como lengua, matemáticas, dibujo, música, etc. Cada una de ellas tiene su nivel de dificultad, sin embargo somos pequeños y parece que no nos damos del todo cuenta ya que lo vamos aprendiendo poco a poco con el paso de los años. Sin embargo, ¿quién no ha llegado a los cursos de educación secundaria obligatoria (ESO) y cuando le dicen que al año siguiente tiene que cursar la asignatura de física y química no ha pensado en la tremenda dificultad que iba a conllevar?, campos gravitatorios, fuerzas de gravedad, tabla periódica, elementos químicos, **formulación química**, etc. Sin embargo, cuando el profesor o la profesora hacía dibujos en la pizarra para explicar como afectaban las fuerzas a un objeto en una rampa o como se formaba una molécula de agua, ya nos parecía un poco más fácil, ¿verdad?.

El principal **objetivo de este proyecto** era usar esa ventaja que aporta ver las cosas de forma más gráfica y desarrollar un prototipo de juego educativo que facilitará la enseñanza y el aprendizaje de la química, permitiendo así a los alumnos darles un enfoque más dinámico y llamativo a la hora de introducirse en este campo de estudio, haciendo que todo parezca un poco más fácil al permitirles interactuar a ellos mismos de forma directa. Algunos de los campos que se han tratado son:

- Aprender los diferentes elementos químicos que existen tanto en la naturaleza como los obtenidos de forma artificial.
- Diferenciar las familias a las que pertenecen cada uno de los elementos

químicos y a su posición en la tabla debido a algunas características como su número atómico.

- Aprender a formular.
- Diferenciar entre los distintos grupos que existen dentro de la formulación inorgánica.
- Aprender la estructura química que se corresponde con una determinada formula

El primer **objetivo tecnológico**, era aprender a programar el sensor de movimiento **Leap Motion** y utilizarlo para conseguir que toda la interacción con el sistema se realizara a través de este. Leap Motion es un dispositivo de interacción hombre-máquina que detecta el movimiento, posición y gestos de las manos en 3D, de modo que permite controlar una interfaz de usuario. Esto permitiría que el alumno se adentrara mucho más en el juego y darle un rol más dinámico al poder interactuar con los elementos y las estructuras químicas con sus propias manos.

Una vez elegido el dispositivo de interacción y el campo de enfoque del futuro juego, había que decidir sobre que motor de juego se realizaría el proyecto. El **segundo objetivo tecnológico** del proyecto era aprender a trabajar con el motor de juegos **Unity 3D** y aprender a integrar en este el dispositivo Leap Motion para poder desarrollar el prototipo del videojuego educativo.

Otro de los objetivos tecnológicos que planteamos inicialmente, fue el desarrollo del proyecto usando realidad virtual para permitir al alumno adentrarse por completo de forma virtual en el entorno, haciendo aún más natural la interacción y evitando así ciertas limitaciones que comentaremos posteriormente de Leap Motion. El problema era el gran desembolso económico y la gran capacidad gráfica y computacional que requería unas gafas como Oculus Rift o HTC Vive, resultando totalmente imposible el desarrollo del proyecto usándolas.

Aunque este no era sólo el problema, ya que no poder desarrollar para este tipo de gafas de realidad virtual tampoco resultaba ser un punto tan negativo, ya que si nos fijamos en la viabilidad futura de implantación de la aplicación en las aulas, no se puede exigir a un instituto que comprara unas gafas para cada alumno. Por tanto se pensó en el uso de las gafas Cardboard usando dispositivos android. Sin embargo, la empresa desarrolladora de Leap Motion aunque había publicado que estaba trabajando en el desarrollo de un SDK para poder integrarlo con android, no lo llegaron a publicar con

tiempo de permitirnos trabajar para android, de hecho a día de hoy sigue sin estar disponible [4] por lo que decidimos establecerlo como un trabajo futuro.

## **1.2. Inspiración para el desarrollo del proyecto**

Hace unos años en la Universidad de Granada y concretamente en la Escuela Técnica Superior de Ingeniería Informática y Telecomunicaciones, un grupo de 3 alumnos tutelados por el mismo director de este proyecto, presentaron un proyecto que se clasificó para la final nacional de la competición Imagine Cup, donde quedó en segunda posición. Esta competición la realiza Microsoft para buscar y promocionar las nuevas generaciones de talentos, sus ideas y proyectos.

El proyecto, llamado **EducAr** consiste en una herramienta educativa que mezcla la realidad aumentada con Kinect, un dispositivo de detección de movimientos, para ofrecer una nueva forma de dar clase. El usuario puede interactuar para aprender y enseñar en diferentes áreas de estudio. La aplicación permite al profesor crear sus propios ejercicios y a través de Azure compartir con otros compañeros[5]. A continuación se adjunta el enlace a un vídeo detallando su funcionamiento[6]:

La aplicación tiene dos modos de funcionamiento:

- El primer modo funciona de tal manera que el usuario acerca etiquetas en el mundo real que la aplicación reconoce mediante la realidad aumentada y generan un resultado.
- En el segundo modo se podían usar modelos directamente en 3D interactuando con Kinect y realizar ejercicios de forma similar al uso de las etiquetas.

EducAr ha servido de inspiración para la realización de este proyecto, ya que este también busca dar las clases de una forma diferente pero usando la realidad virtual y un dispositivo de detección de gestos distinto, que nos permita interactuar directamente con nuestras propias manos con un poco más de precisión.

## **Capítulo 2**

# **Planificación**

Cuando se quiere realizar un proyecto, hay que plantear desde un principio el modo de trabajo que se va a seguir para poder cumplir con los objetivos iniciales en el tiempo que se tiene disponible. La primera tarea por tanto será realizar un análisis previo del proyecto, definiendo de forma exacta qué objetivos buscamos perseguir.

### **2.1. Análisis Hardware**

En nuestro caso, en primer lugar tendremos que realizar un estudio acerca de los dispositivos hardware que existen en el mercado actualmente y que vamos a utilizar en la realización del proyecto, para ello seleccionaremos los que más nos convengan tanto para el estado final como para la viabilidad de un posible futuro. Para nuestro proyecto, este análisis tendrá que centrarse fundamentalmente en valorar las características de los dispositivos de interacción gestual que existen en el mercado y decidir con cual de ellos vamos a trabajar. Vamos a comentar las características de algunos dispositivos de interacción gestual que planteamos para la realización del proyecto y comentar la decisión tomada al final. A continuación se hará un estudio de los diferentes dispositivos de realidad virtual que planteamos y comentaremos la decisión que más viable sería para llevar el proyecto educacional al mercado.

#### **2.1.1. Dispositivos de Interacción Gestual**

Para la interacción gestual, decidimos plantear el uso de uno de los tres dispositivos que se habían visto en la asignatura “Nuevos Paradigmas de Interacción”. Para decidir cual, a parte de la preferencia propia, hemos realizado un estudio de las características de cada dispositivo:

### Kinect

Kinect es un sensor de movimiento que desarrolló por Microsoft para sus videoconsolas Xbox 360 y Xbox One y para ordenadores con sistema operativo Windows. Básicamente, Kinect es un dispositivo de interacción hombre-máquina similar a una cámara que permite a los usuarios controlar su videoconsola u ordenador a través de una interfaz natural usando gestos con su propio cuerpo y comandos de voz [7].

Al poco tiempo de surgir, se publicó una SDK para Windows que permitía el desarrollo de aplicaciones utilizando Kinect. Con la llegada de la Xbox One, salió a la venta un nuevo Kinect con prestaciones totalmente mejoradas respecto al anterior y un nuevo SDK para el desarrollo de aplicaciones utilizando esta nueva herramienta. A partir de ahora nos centraremos en el Kinect 360, ya que tenía mucha más documentación al ser más antiguo y realmente para nuestro proyecto hubiera servido tanto un como otro.

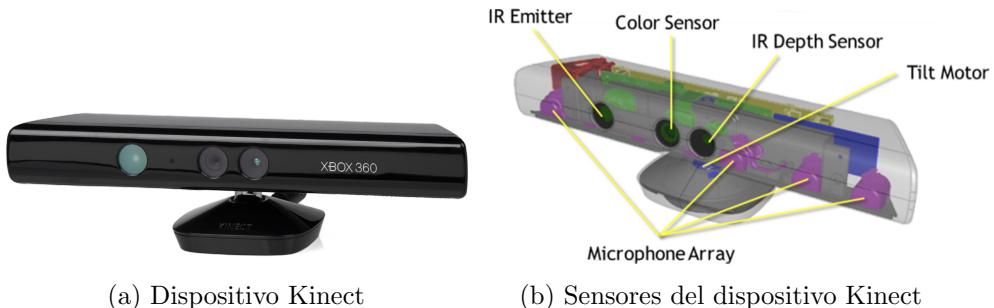


Figura 2.1: Kinect

Kinect contiene una serie de dispositivos de detección que le permiten capturar el movimiento tridimensional de todo el cuerpo del sujeto y también el reconocimiento facial y de voz. Estos dispositivos son: [8]

- Color Sensor: Un cámara a color RGB.
- IR Depth Sensor: Un sensor de profundidad.
- Micrhophone Array: Consiste en un array de 4 micrófonos que permiten el reconocimiento de voz.
- IR Emitter: Un proyector infrarrojos que se combina con el sensor de profundidad para la detección tridimensional de los movimientos. Básicamente el funcionamiento de detección es que se emite a través de este sensor unos rayos infrarrojos que son reflejados por la persona (u objeto) y que es captado por el sensor de profundidad.

- Tilt Motor: Permite subir y bajar la el dispositivo con respecto a la base.

El rango de detección de Kinect es el siguiente, siendo la zona de color azul el rango donde mejor detección tiene:

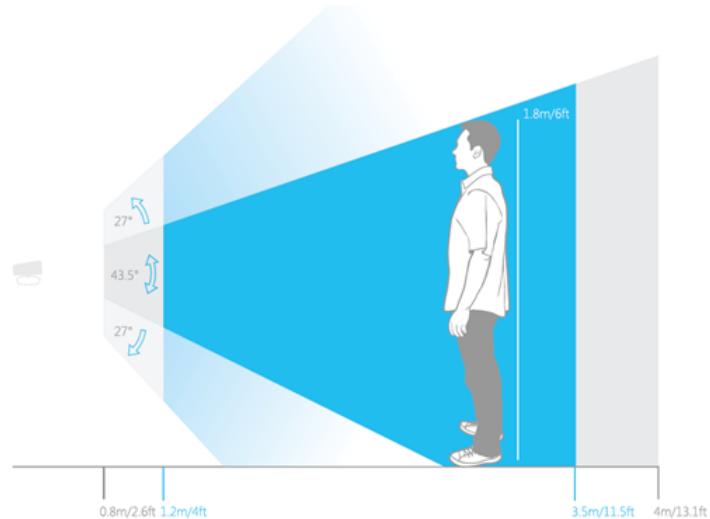


Figura 2.2: Rango de detección de Kinect

Una de las grandes ventajas que tiene Kinect es que permite detectar el movimiento realizado con todo el cuerpo, aunque tiene una determinada precisión. Por ejemplo, es capaz de captar el movimiento de la mano, si cerramos o abrimos, pero no es capaz de detectar con precisión que dedo estamos moviendo de la mano.

### Leap Motion

Leap Motion es un dispositivo de interacción hombre - máquina que detecta los movimientos tridimensionales que el usuario hace con sus manos y dedos, permitiendo controlar una interfaz de usuario sin usar ningún otro dispositivo. Tiene una precisión de menos de 1 milímetro, siendo capaz de detectar las dos manos del usuario y sus diez dedos, interpretando los movimientos y los gestos en 3D que se realicen. [9]



(a) Dispositivo Leap Motion Controller (b) Sensores del dispositivo Leap Motion

Figura 2.3: Leap Motion

El dispositivo tiene una serie de sensores que le permiten captar un área aproximadamente semiesférica sobre él mismo. Está formado por:

- IR Leds: Hay 3 repartidos por el dispositivo y emiten rayos infrarrojos.
- Sensores: Son dos cámaras que captan casi 200 fps de datos reflejados.

El dispositivo funciona de modo que los sensores IR leds emiten rayos infrarrojos que rebotan contra las manos y gracias a los sensores (cámaras) se captan cada uno de estos datos reflejados. Posteriormente el software interno de Leap Motion genera una síntesis de datos en 3D de los frames obtenidos. El dispositivo es compatible ahora mismo solo con Windows y Mac OS, aunque está confirmado que próximamente también estará para Android.

Inicialmente, Leap Motion Controller se pensó para poder controlar las interfaces de usuario como si fuera un propio ratón y se publicó el SDK (V2 desktop) para poder desarrollar aplicaciones que utilizaran dicho dispositivo. Para ello, el dispositivo se coloca sobre una mesa, preferiblemente delante de nuestra pantalla, y detecta las manos que colocamos por encima de este dentro de un rango determinado. Sin embargo, aunque parezca muy natural puede llegar a resultar bastante incómodo tener que estar 10 minutos con

las manos encima del dispositivo, lo cual no es una posición muy natural. Además, esto sumado a que el rango óptimo de detección es pequeño, consigue que el usuario se llegue a cansar y prefiera usar un ratón para controlar la interfaz.

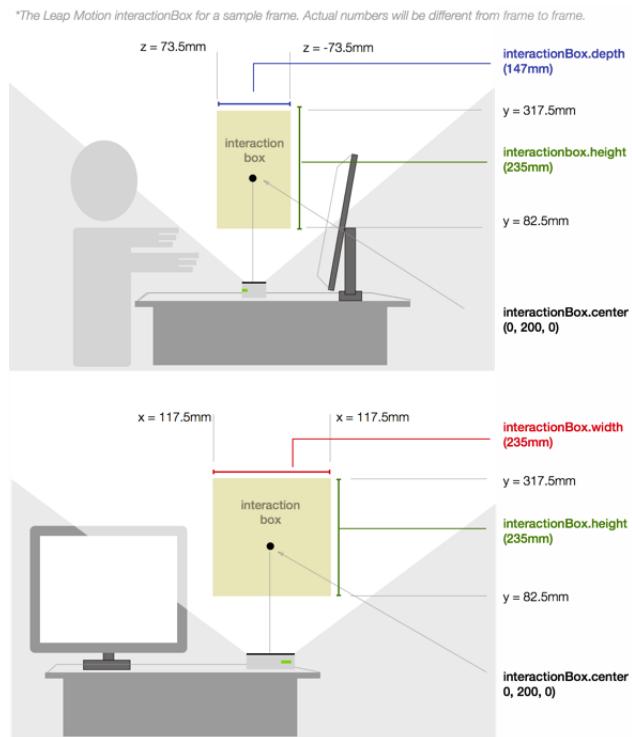


Figura 2.4: Interaction Box de Leap Motion

El área blanca que sale en forma de cono desde el dispositivo es donde emite los rayos infrarrojos los sensores del Leap Motion, mientras que la caja marrón dónde pone Interaction Box es realmente el área donde se detecta con precisión. Esta posición puede llegar a ser realmente incómoda, por lo que se decidió dejar la idea inicial y dieron un salto a la realidad virtual, creando el soporte necesario para poder integrar Leap Motion con unas gafas de realidad virtual, teniendo su propia SDK para el desarrollo de aplicaciones para realidad virtual (Orion).

Para integrar el dispositivo con la realidad virtual, sólo hace falta colocar el dispositivo encima de unas gafas de realidad virtual y conectar ambos dispositivos a un ordenador. De este modo tendremos una interacción mucho más natural ya que podremos interactuar con nuestras manos con las gafas puestas y realizando delante de ellas los mismos gestos y movimientos que realizaríamos si no las tuviéramos puestas. Es decir, permitiría ver gracias al

dispositivo la representación real de tus propias manos en formato 3D en el mundo virtual, haciendo una experiencia más real y natural para el usuario.



(a) Dispositivo Leap Motion integrado (b) Área de detección de Leap Motion con con unas gafas de realidad virtual              gafas de realidad virtual

Figura 2.5: Área de detección de Leap Motion con realidad virtual

La principal diferencia con Kinect es que este está pensado para captar el movimiento de todo el cuerpo en un espacio más grande como una sala, en cambio en Leap Motion el área de observación del dispositivo es más pequeña, pero tiene una mayor precisión para poder detectar por ejemplo el movimiento de los dedos.

Tanto Kinect como Leap Motion tienen problemas con la luz solar, generando ruido y provocando imprecisión en la detección, de modo que la zona óptima de funcionamiento sería una zona sin luz solar y utilizando ropas de colores oscuros, ya que esto también influye en la detección.

## Myo

El brazalete Myo es un dispositivo de interacción hombre - máquina que es capaz de reconocer los gestos de las manos y el movimiento de los brazos en base a los impulsos eléctricos generados por los músculos. Se coloca debajo del codo y se puede usar para interactuar con interfaces de usuario, por ejemplo para pasar diapositivas en una presentación o jugar a algún juego donde se pueda subir y bajar con el movimiento del brazo.

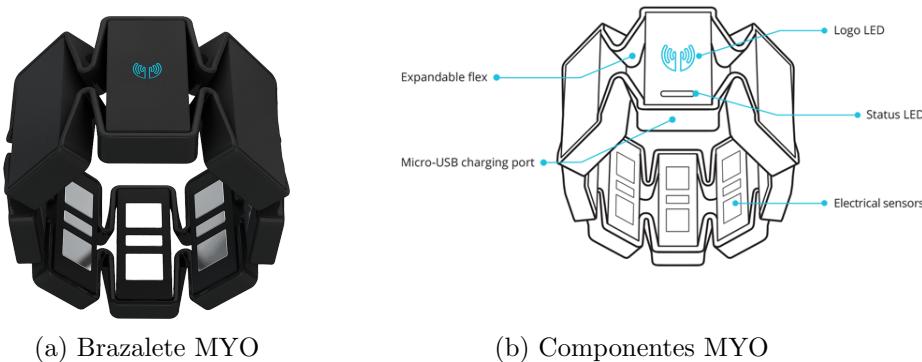


Figura 2.6: Brazalete Myo

El brazalete está equipado con una serie de sensores para poder reconocer los gestos y movimientos de manos y brazos. Para ello utilizan un proceso llamado electromiografía (EMG), técnica que evalúa el estado de salud de los músculos y las células nerviosas que los controlan, las cuales transmiten señales eléctricas que hacen que los músculos se contraigan y relajen.[10]. Cada persona tiene un diferente tamaño muscular, diferente piel u otras diferencias significativas, por lo que al principio de usar el dispositivo es necesario realizar una calibración para que la detección de los sensores EMG sea más precisa.

Los componentes del brazalete Myo son los siguientes:

- Electrical sensors: Sensores EMG, el dispositivo tiene ocho repartidos en el interior de su estructura. Son encargados de reconocer cada gesto.
- Expandable flex: Sirve para adaptar el brazalete a los diferentes tamaños de brazos de los usuarios.
- Micro-USB charging port: Sirve para cargar y programar el dispositivo.
- Logo y status LED: Indicadores de si está el dispositivo conectado y del estado de batería en el que se encuentra.

Además de esto, también cuenta con una unidad de medición inercial (IMU), compuesta por un giroscopio, un acelerómetro y un magnetómetro, que permiten detectar los movimientos realizados con el brazo. También cabe destacar que es capaz de emitir una serie de vibraciones para que sirvan al usuario como retroalimentación. [11]

Gracias a todas estas características, Myo es un dispositivo que se puede aplicar en un sin fin de campos, entre ellos en técnicas de fisioterapia, no sólo permitiendo detectar con precisión los gestos realizados con las manos y los movimientos de los brazos, si no hasta personas que les han amputado un brazo podrían combinarse con prótesis que fuesen controladas por los impulsos nerviosos captados por el Myo.

### Comparativa de dispositivos de interacción gestual y decisión

A continuación vamos a realizar un estudio comparativo de las propiedades más significantes de cada uno de los 3 dispositivos, justificando a continuación porque hemos elegido Leap Motion como herramienta de interacción.

Cuadro 2.1: Tabla comparativa de dispositivos de interacción gestual

Características / Dispositivo	Kinect 360	Leap Motion	Myo
Detección del movimiento y los gestos de las manos	Si	Si	Si
Detección precisa de las manos	No, no permite reconocer con precisión el movimiento de los dedos de la mano	Si	Si
Detección corporal	Permite detectar los gestos y movimientos realizados por el usuario	Sólo se limita al reconocimiento de los movimientos de las manos	Sólo reconoce los movimientos realizados con los brazos y las manos
Reconocimiento oral	Si	No	No
Integración con realidad virtual	Si	Si	Si
Plataforma de desarrollo disponibles	Ordenadores con SO Windows y Xbox 360	Ordenadores con SO Windows/Mac OS y en desarrollo para dispositivos Android	Ordenadores con SO Windows/Mac OS y dispositivos Android/iOS.
Precio	Aproximadamente 130 € con cable de conexión para ordenador	68 €	170 €

Como podemos comprobar, prácticamente cualquiera de los tres dispositivos nos hubiera servido para el desarrollo de la interacción del proyecto. Sin embargo, hay características y situaciones que nos han llevado a elegir el dispositivo Leap Motion. Esto se debe a varios factores:

1. Kinect nos proporciona un reconocimiento gestual de todo el cuerpo, incluidas las manos, sin embargo para llevar la aplicación a las aulas nos interesaría utilizar dispositivos que nos permitieran ser más

portátiles y nos permitieran tener un rango de visión o detección más personalizado (o pequeño), como podrían ser el Leap Motion y el Myo. El Leap Motion detecta solo el movimiento de las manos y tiene una zona de detección más reducida, luego sería útil en caso de que un alumno en su sitio quisiera utilizar la aplicación sin que sus movimientos interfiriesen con la actividad de otro alumno. El Myo también detecta el movimiento de las manos y los brazos, además responde sólo a los impulsos del usuario portador, lo que también solventaría incluso con mejor rendimiento que Leap Motion el problema de Kinect. Sin embargo, Leap Motion es capaz de detectar los movimientos realizados con ambas manos mientras que necesitaríamos dos dispositivos Myo para igualar la misma detección.

2. Aunque Kinect nos aporte reconocimiento oral, el uso en el aula podría suponer un problema si cada niño estuviera interactuando oralmente todo el rato con la aplicación.
3. Todos tienen soporte para realidad virtual, sin embargo Kinect nos limita el desarrollo de aplicaciones para ordenadores con sistema operativo Windows o consolas Xbox 360, lo cual también es un problema ya que en un futuro lo ideal sería poder exportarlo a dispositivos móviles para poder usar la aplicación en cualquier parte sólo disponiendo de un dispositivo móvil. En este aspecto Myo, es el que más ventaja tiene ya que tiene soporte para ordenadores con sistema operativo tanto Mac OS como Windows, además de dispositivos móviles Android o iOS. Leap Motion también tiene soporte tanto para ordenadores con sistema operativo Windows como Mac OS, sin embargo está en desarrollo el soporte por Android aunque estará próximamente, así que también nos serviría.
4. En precio la diferencia es abismal, Kinect cuesta casi el doble que el Leap Motion, mientras que Myo cuesta casi el triple (sin contar que necesitaríamos dos brazaletes). Para el desarrollo de una aplicación con finalidad como la nuestra, es inviable seleccionar como base de interacción un dispositivo de tal valor económico. Cuanto más barato sea, más dispositivos podrían adquirir los institutos para prestarlos a los alumnos, así como comprar los propios alumnos.
5. Todos los dispositivos tienen una buena documentación y gran apoyo de la comunidad, aunque sobre todo Kinect y Leap Motion por antigüedad.

Por todos estos motivos principalmente, decidí utilizar el dispositivo Leap Motion. También cabe destacar que uno de mis objetivos principales era aprender a usar este dispositivo e integrarlo con un motor de juegos, así que este motivo también ayudó.

### 2.1.2. Gafas de Realidad Virtual

Aunque para nuestro proyecto no se usarán gafas de realidad virtual ni se exportará el proyecto para que sea compatible, hemos realizado un estudio comparativo de diferentes modelos de gafas de realidad virtual que están en el mercado para ver la viabilidad del futuro desarrollo del proyecto con realidad virtual.

#### Oculus Rift

Oculus Rift es un casco de realidad virtual que permite al usuario sumergirse en un mundo virtual y poder interactuar con este. Consta de dos lentes que nos permiten tener una resolución de 1080x1200px por ojo, con una frecuencia de refresco de 90 Hz y permiten un campo de visión de 110° [12]. Tiene auriculares integrados que proporcionan un efecto de sonido 3D. Permite un seguimiento de posición y rotación, para ello se ayuda de un sistema denominado “constelación” o sensor oculus que es un sensor de infrarrojos que recoge la luz que es emitida por los LED de IR que lleva el casco. Esto permite generar un espacio 3D. El área de seguimiento es de 1,5 x 3,3 m. [13]



(a) Oculus Rift

(b) Sensor Constelación

Figura 2.7: Oculus Rift y sensor Constelación

Oculus creó un SDK para que los desarrolladores pudieran integrar en sus juegos el uso de Oculus Rift. Actualmente sólo está disponible, tanto el casco como el SDK, para consolas Xbox One u ordenadores con sistema operativo Windows, ya que, entre otras cosas, necesita una serie de especificaciones hardware mínimas para poder rendir. Actualmente es de los mejores productos en renderizado de realidad virtual que hay en el mercado, aunque su precio de mercado es de unos 450 €. Este precio incluye, además de las gafas y el sensor constelación, dos mandos (uno para cada mano) que se pueden usar dentro de juegos como por ejemplo para disparar o interactuar

con los elementos dentro del juego.

### HTC Vive

HTC Vive es un casco de realidad virtual que permite al usuario sumergirse en un mundo virtual y poder interactuar con este. Consta de dos lentes que nos permiten tener una resolución de 1080x1200px por ojo, con una frecuencia de refresco de 90 Hz y permiten un campo de visión de 110°. No tiene audio integrado aunque se pueden conectar unos auriculares denominadas “Vive Deluxe Audio Strap”. Para el seguimiento de la posición, usan unos dispositivos especiales denominados “Vive Base Stations” que generan un espacio virtual de 360°. Tienen un área de seguimiento de 4,5 x 4,5 m. [14]

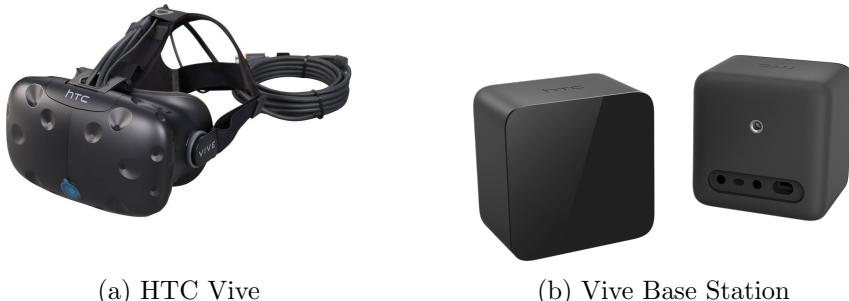


Figura 2.8: HTC Vive y Vive Base Station

Estas gafas también publicaron su propio SDK para permitir a los desarrolladores crear sus propios juegos y dar soporte para ordenadores con sistema operativo Windows, MacOS o Linux. HTC Vive también tiene sus propios dispositivos que permiten la interacción con el mundo virtual y que también se venden junto al paquete principal. Su precio de mercado es de 600 €.

Tanto HTC Vive como Oculus Rift son unos cascos de realidad virtual con excelentes prestaciones, cualquiera de las dos puede hacer que nos sumerjamos en un entorno de realidad virtual completa donde podamos interactuar con todos los elementos del mundo, hasta parecer que de verdad estamos allí. La calidad de imagen y el número de fps que son capaces de captar hacen que la experiencia del usuario sea increíble. A la hora de elegir una u otra sería una difícil decisión, ya que ambas rinden muy bien, aunque a la hora de usarlo para nuestra aplicación sería una locura optar por uno de estos modelos. Estos tipos de gafas de realidad virtual necesitan unos re-

quisitos mínimos que implican ordenadores de altas prestaciones para poder utilizarlos, sobre todo para obtener el máximo rendimiento.

Nuestro proyecto se trata de una aplicación educativa que en un futuro pueda convertirse en un Serious Game y poder llevarse a las aulas de modo que cualquier usuario pueda acceder a ella e incluso poder usarla desde su casa. Por tanto, no podemos obligar a los niños a adquirir un ordenador de determinadas prestaciones y un casco de realidad virtual de estas características con el desembolso económico que esto conlleva. Así que en el futuro se plantearía para poder usarlo con gafas de realidad virtual como lo son las **Google Cardboard** junto a un dispositivo móvil Android, lo cual es accesible a todos los usuarios.

### Google CardBoard

Las Google Cardboard son una plataforma de realidad virtual que Google desarrolló de modo que cualquier usuario a través de un dispositivo Android o iOS pudiera introducirse en el mundo de la realidad virtual.



Figura 2.9: Google CardBoard

Las gafas constan de un par de lentes, unos imanes, velcro, una banda de goma y una etiqueta NFC. Para utilizarlo colocamos el móvil en el soporte delantero de las gafas (como en la imagen) y nos colocamos las gafas en los ojos, utilizando en nuestro móvil cualquier aplicación compatible y pudiendo interactuar con un botón que tienen las gafas. Sólo con estas características el usuario podrá introducirse en un mundo virtual, pudiendo adquirir este tipo de dispositivos cualquier persona (cuesta aproximadamente 5 €) o incluso fabricarlas tú, ya que se suministra desde la web oficial las instrucciones necesarias para llevarlas a cabo [15].

Como podemos observar la tecnología no reside en las gafas, si no en el

software de la aplicación. La idea de las Google no era competir con Oculus o Vive, ya que era imposible con este tipo de dispositivo ya que nunca se podría comparar ni en calidad de imagen ni con el envolvente sonido 3D que si que se puede llegar a conseguir con las otras gafas, si no que lo que quería era hacer un dispositivo que estuviera al alcance de todos y que permitiera captar la atención de los usuarios para introducirlos poco a poco en el mundo de la realidad virtual. Cabe destacar que Google ofrece un kit de desarrollo para el diseño de aplicaciones para dispositivos Android e iOS.

## Conclusión

Por todo lo comentado anteriormente, es evidente que para la finalidad que queremos que tenga nuestro proyecto en un futuro y la viabilidad económica que este supondría para su implantación en los institutos, utilizaríamos las gafas de Google ya que supondría que cualquier alumno tuviera disponible este tipo de gafas en sus casas y en el colegio para poder interactuar con la aplicación. Además, Leap Motion es un dispositivo económico en relación a los similares y próximamente darán soporte para poder usarlo con Android, por lo que podríamos integrarlo con las Google Cardboard, por lo que los institutos podrían adquirir un cierto número de unidades y establecer un sistema de prestamos en el instituto, sin contar con que sería también un desembolso mucho menor para los padres de los niños.

## 2.2. Análisis Software

El proyecto, pretende ser la base para un futuro serious game, por lo que que es lógico y necesario el uso de un motor de juegos que nos facilite el desarrollo del proyecto. Así, la siguiente parte en la que se centrará nuestro proyecto será analizar los motores de juego existentes y valorar cual de ellos se adapta más a nuestros gustos y necesidades.

### 2.2.1. Motor de juego

Un motor de juegos (engine) es un software orientado a diseñar, crear y representar un videojuego, así como la funcionalidad para poder portarlo a múltiples plataformas.[16] Algunas funcionalidades son:

- Renderizado 2D y 3D.
- Motor de físicas.

- Detector de colisiones.
- Animación.
- Scripting
- Sonido
- Inteligencia Artificial

Dos de los principales motores de juego que existen en la actualidad son **Unity 3D** [17] y **Unreal Engine**[18]. A continuación vamos a comparar los dos motores y justificar la decisión que hemos tomado para desarrollar el proyecto.

Tanto Unity como Unreal son los motores de juegos más potentes que existen en la actualidad. Unity ha sido el motor de juegos pionero por excelencia, ya que fue el primero en permitir una gran versatilidad y funciones para poder crear un videojuego con apenas programación gracias a las diversas funcionalidades de su IDE que facilitan el desarrollo rápido. Sin embargo, con la llegada de Unreal 4 (su última versión) este motor ha sido capaz de ponerse a la altura de Unity gracias a su sistema de “Blueprints”. Este sistema permite que aunque no tengamos ninguna noción de programación podamos desarrollar un videojuego utilizando una programación basada en nodos y componentes, siendo similar a la creación de diagramas de flujo. [19]

Cuadro 2.2: Comparativa entre Unity y Unreal

Características \Engine	Unity 3D	Unreal Engine
Sistema Operativo	Windows/OS X	Windows/OS X
Multiplataforma	Si	Si
Lenguajes	C#, Javascript, Boo	C++
Precio	Tiene varias licencias. Una de ellas gratuita hasta cierta cantidad de dinero.	Licencia gratuita hasta cierta cantidad de ventas, a partir de entonces se cobra una comisión del 5%.
Documentación y comunidad	Amplia comunidad de usuarios y documentación.	Amplia comunidad de usuarios y documentación, aunque menor que la de Unity
Compatibilidad con Leap Motion	Si	Si

Unity es relativamente fácil de aprender a utilizar y tiene muchos plugins gratuitos. En cambio, Unreal es más complejo de aprender a utilizarlo y no existen tantos plugins gratuitos disponibles como Unity, sin embargo es más potente gráficamente.

Unity y Unreal tienen sus pros y sus contras, aunque ambos tienen soporte para integrar Leap Motion así que realmente daba un poco igual cuál de los dos seleccionáramos. Debido a que el número de plugins gratuitos disponibles, así como que la documentación y la comunidad de Unity es mucho mayor que la de Unreal y la enorme facilidad que aporta Unity para el desarrollo de juegos 2D, decidí utilizar dicho Engine. También cabe destacar que era uno de los objetivos iniciales del proyecto, lo cual esto también me ayudó a decantarme por este, usando para el desarrollo la versión: **Unity 2017.3.1.**

### 2.3. Metodología para el desarrollo del sistema

Una vez planteado el proyecto, decididas las herramientas que se van a emplear y habiéndose documentado de ellas, el siguiente paso consistirá en ponerse con el software. Sin embargo, para realizar el desarrollo software de una aplicación tenemos que seguir una serie de principios para analizar en primer lugar el sistema que deseamos desarrollar y que posteriormente la implementación resulte más sencilla y guiada a través de estos principios. Por tanto, tendremos que realizar un buen análisis software del sistema para poder ver reflejados los objetivos que se plantearon en el estado final del proyecto.

Al tratarse de un proyecto de fin de grado, el desarrollo software recae totalmente por el alumno y no en un equipo de desarrollo. Por lo tanto, tenemos que optar por métodos de desarrollo que permitan ajustarnos al máximo al poco tiempo disponible y cumplir con los objetivos, pudiendo adaptarnos a los complicaciones que puedan ir ocurriendo a lo largo del proyecto, como por ejemplo que vayan surgiendo nuevos requisitos según avanza el desarrollo. Por tanto, hemos decidido seguir un método de desarrollo ágil iterativo, más en concreto el método SCRUM.

El método SCRUM consiste básicamente en realizar el desarrollo del software mediante iteraciones denominadas “Sprint”, de forma que dividimos el proyecto en pequeños pasos y se les asigne un periodo de tiempo determinado[20]. Atendiendo a los principios básicos de SCRUM, nuestro desarrollo se basará en este tipo de metodología ya que iremos dividiendo el proyecto en pequeños objetivos que iremos cumpliendo poco a poco en determinados períodos de tiempo, revisados de forma frecuente a través de reuniones con el tutor y ajustándonos a la planificación para terminar el proyecto en los plazos establecidos y con los objetivos iniciales conseguidos.

Por tanto, dividimos inicialmente el proyecto en iteraciones a las que a cada una le asignamos un marco estricto de tiempo, de modo que cuando pasásemos a una nueva iteración, la anterior tenía que estar totalmente terminada y probada. De este modo, nos asegurábamos ir cumpliendo con los objetivos de cada iteración y restringirnos al tiempo estimado para poder llegar a la fecha de entrega.

### **2.3.1. Iteraciones (sprints)**

A continuación definiremos las iteraciones que llevaremos a cabo.

#### **1<sup>a</sup> Iteración: Escena formulación**

**Tiempo estimado:** Dos semanas.

En esta iteración se creará la escena dónde los usuarios interactuarán con los átomos, la escena llamada: “Formulación”. Aunque en esta parte sólo incluiremos algunas funciones:

- Creación del objeto átomo que permita ser manipulado por las manos.
- Creación de los límites que nos restrinjan el movimiento de los átomos a las zonas dónde podemos interactuar con las manos de forma correcta.
- Creación de los script controladores necesarios que nos permitan enlazar los átomos.

#### **2<sup>a</sup> Iteración: Escena menú inicial**

**Tiempo estimado:** Una semana.

Una vez creada las funcionalidades básicas de la escena principal y habiendo podido familiarizarnos ya con el uso de Leap y Unity, diseñaremos el menú inicial. Se llevarán a cabo las siguientes tareas:

- Crear un menú de interacción 2D con los botones necesarios para acceder a las diferentes funciones de la aplicación.
- Creación de scripts controladores de Leap Motion que nos permitan detectar los gestos que necesitamos y poder tratarlos.
- Creación de puntero que pueda ser movido moviendo nuestra propia mano sobre Leap Motion.
- Enlazar esta escena con la principal.

- Permitir la interacción realizando gestos con las manos detectados por Leap Motion.

### 3<sup>a</sup> Iteración: Menú Tabla periódica

**Tiempo estimado:** Una semana.

Diseño de una escena que contenga la tabla periódica y podamos interactuar con ella:

- Creación de los sprites para los botones que representarán los grupos de la tabla periódica.
- Gestión y detección del gesto que nos permita ir hacia la escena de formulación.
- Permitir la interacción realizando gestos con las manos detectados por Leap Motion.

### 4<sup>a</sup> Iteración: Escenas Grupos

**Tiempo estimado:** Dos semanas.

- Crear una escena para cada grupo de la tabla periódica, distribuyendo en su interior los elementos de la tabla periódica pertenecientes a dicho grupo.
- Pensar en algún mecanismo que permita no romper demasiado con la distribución de los elementos de dicho grupo pero adaptándonos a la pantalla.
- Permitir interacción con los elementos usando Leap Motion.

### 5<sup>a</sup> Iteración: Panel Calculadora

**Tiempo estimado:** Dos semanas.

Debemos de crear la herramienta necesaria para poder indicar el número de átomos que queremos de cada elemento y almacenarlos de alguna forma para compartirlos entre escenas.

- Creación del panel calculadora que nos permita seleccionar el número de átomos que queremos del elemento seleccionado.
- Creación de una clase serializable que nos permita almacenar la información, en un fichero JSON, de los átomos seleccionados para posteriormente poder utilizarlos en la escena de formulación.

**6<sup>a</sup> Iteración: Creación y gestión del inventario**

**Tiempo estimado:** Dos semanas.

Una vez consigamos almacenar la información entre escenas, tenemos que crear un inventario que nos permita seleccionar los átomos que queremos utilizar para realizar los enlaces.

- Crear los botones que nos permitan ir a la escena de la tabla periódica o desplegar el inventario en la escena de formulación.
- Diseñar el inventario con el script necesario que cargue la clase serializable almacenada en el fichero JSON e instancie en la escena los átomos seleccionados.
- Creación de los scripts necesarios que nos permitan sacar átomos del inventario y llevarlos a la escena de forma que se fijen al soltarlos y permitir enlazarlos.

**7<sup>a</sup> Iteración: Escena ejercicios**

**Tiempo estimado:** Una semana.

Una vez realizado esto, tenemos que pensar en la creación de una base de datos con fórmulas que serán reconocidas por nuestra aplicación y la creación de la escena de ejercicios.

- Crear clase serializable que permita leer los datos introducidos en un archivo JSON referentes a cada fórmula.
- Desarrollar escena ejercicios de modo que permitiera seleccionar algún tipo de ejercicio y cargar fórmulas de la base de datos correspondientes con el tipo seleccionado para ser resueltas en la escena de formulación.

**8<sup>a</sup> Iteración: Gestión de solución**

**Tiempo estimado:** Dos semanas.

Una vez realizados los pasos anteriores, tenemos que dotar de funcionalidad a la finalidad de esta aplicación que es poder aprender. Por tanto, tendremos que introducir un método que nos permita comprobar resultados:

- Introducir un botón que nos permita encontrar la solución.
- Generar un script que nos permita comprobar si los enlaces realizados en la escena se corresponden con los del ejercicio que seleccionemos o con alguno de nuestra base de datos.

Esta iteración parecerá más complicada de lo que parece, ya que requerirá definir como vamos almacenar los enlaces y buscar en el archivo de la base de datos.

### Diagrama de Gantt

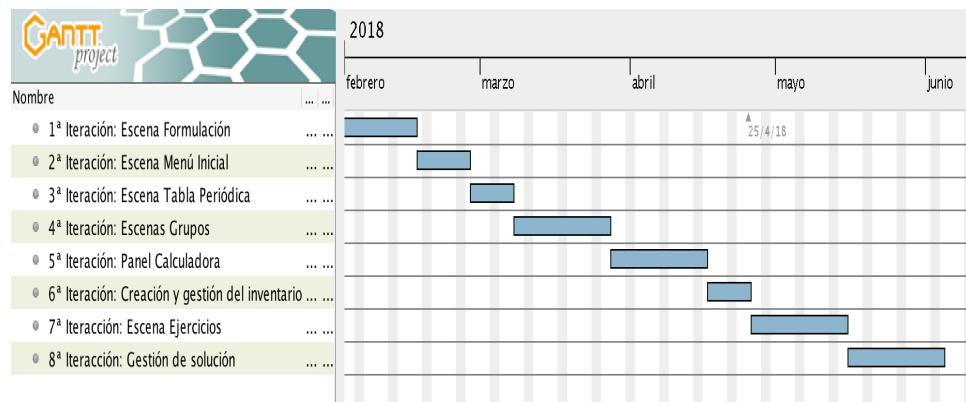


Figura 2.10: Diagrama de Gantt



# **Capítulo 3**

## **Estado del arte**

En este apartado vamos a introducirnos históricamente en la situación tecnológica actual, comentando los antecedentes históricos que han precedido al proyecto, la crítica hacia ellos y las diferencias fundamentales con el trabajo que se ha realizado y que pueden suponer un punto de inflexión en la actualidad.

### **3.1. La química y la educación**

La química es la ciencia que estudia la composición, estructura y propiedades de la materia, así como los cambios que esta experimenta durante reacciones químicas. La química está en todas partes: el agua tiene una estructura química que se compone a través de elementos que se pueden encontrar en la naturaleza como son el oxígeno o el hidrógeno, la sal que usamos para las comidas tiene una estructura química determinada también, el agua oxigenada con la que nos curamos las heridas está hecha a partir de una mezcla de oxígeno e hidrógeno... etc. [21]

Desde el principio de los tiempos hemos estado empleando la química de una manera consciente o inconsciente, como cuando el hombre empezó a usar el fuego para transformar la materia u obtuvo hierro a partir de mineral o el vidrio a partir de arena. Hoy en día, el hombre ha investigado y desarrollado este campo de estudio hasta un punto en el que casi todo lo que nos rodea o podemos observar está compuesto por química: materiales, energía, alimentos, cosmética, medicina ... etc. [22]

En general la química ha supuesto una aportación esencial para el bienestar del ser humano, sin embargo cada vez es menos apreciada por la sociedad. Esta baja percepción social se debe en parte a la educación de los niños.

Con el cambio de la educación, la física y la química perdieron peso en el sistema educativo de los institutos, de modo que anteriormente se cursaba de manera obligatoria y como una única asignatura en más cursos y ahora es todo lo contrario siendo solo obligatoria en 3 cursos y dos de ellos impariéndose de manera conjunta como asignatura con geología y biología (Ciencias de la Naturaleza), lo cuál no permite especificar mucho en el área. A partir de ahí, se vuelve de carácter optativo y debido a su dificultad y a la mínima base que tienen los estudiantes prefieren por lo general optar por otras materias más fáciles. [23]

Además de lo mencionado anteriormente, los programas educativos en estos niveles intentan transmitir demasiados conocimientos teóricos en muy pocas horas lectivas y no se realizan prácticas de laboratorio, no se aplica a la vida cotidiana, no se explica de una manera más visual... etc. Todo esto hace que la física y la química se vaya dejando un poco de lado entre nuestros jóvenes y que gran cantidad de ellos pierdan su interés en este campo, siendo estos jóvenes la base del futuro del país.

[22]

El sistema educativo es difícil que vaya a cambiar, por lo menos a priori, sin embargo sí que podríamos hacer que los jóvenes adquieran un poco más de interés en el campo de la química. Para ello, tal vez lo que se necesita es cambiar la forma en la que se enseña, de modo que no todo sea tan abstracto y tan teórico como las clases que se dan en la actualidad, si no que se intentará que todo fuera mucho más dinámico y visual, de modo que los alumnos participaran más en las clases utilizando métodos de enseñanza que sean más interesantes para ellos, que impliquen utilizar lo que hoy en día capta la mayor parte de su atención, la tecnología.

### **3.2. Interacción hombre - máquina**

La interacción hombre - máquina (Human Computer Interaction - HCI) se define como la disciplina orientada al diseño, evaluación e implementación de sistemas computacionales interactivos para ser usados por humanos.

- El hombre se trata de uno o varios usuarios que interactúan de modo competitivo o cooperativo, teniendo cada uno diversas capacidades físicas o mentales.
- La máquina es cualquier dispositivo que tenga capacidad de cómputo.

### 3.2.1. Historia del HCI

Realizaremos un recorrido desde la antigüedad hasta la actualidad de la evolución de la interacción hombre - máquina, hablando de las interfaces gestuales, que son las que realmente interesan en este proyecto.

En la década de los 60 no existía la HCI como tal. Fue a partir de los 70 cuando con la llegada de los monitores se aceleró el intercambio de información, aunque la interacción seguía siendo muy pobre. Tenían interfaces de usuario muy pobres, mal diseñadas y difíciles de aprender a usar.

En la década de los 80, llegaron los ordenadores personales (PC) que eran muchos más accesibles a todo el mundo. La interacción mejoró, ya que al tener muchos usuarios con limitaciones distintas tuvieron que mejorar las interfaces de modo que fueran más simples y eficientes. Con el PC nació la GUI (Graphical User Interface) y con ella se empezó a desarrollar la interacción hombre - máquina, surgiendo los monitores los menús y el ratón como herramienta de control.

En la década de los 90 el diseño de interfaces se convierte en una disciplina tratada científicamente, convirtiéndose la interfaz en un sistema centrado en el usuario.

En la actualidad la interacción hombre - máquina se ve como una interacción desarrollada en contextos sociales y de organización, donde varios dispositivos intentan satisfacer las necesidades de los humanos. El componente humano de la interacción se estudia desde sus limitaciones fisiológicas, sus habilidades y la psicología.

### 3.2.2. Tipos de HCI

Podemos distinguir dos tipos básicos de interacción hombre - máquina en función de como interactúe el usuario con el dispositivo:

- **HCI explícita:** Puede realizarse mediante comandos o por acciones explícitas. Pone al usuario en el centro del proceso, de modo que es este el que controla todas las operaciones del sistema. Sin embargo, esto puede llevar a abrumar al usuario ya que en un ambiente con muchos dispositivos el usuario tendría que controlarlos a todos ellos. Para que sea efectiva debe tener un buen diseño de la interfaz de usuario, ya que muchos usuarios es la única parte del sistema que ven.

- **HCI implícita:** Se da cuando el usuario realiza una acción cuya intención no es interaccionar con el sistema pero que el sistema interpreta como una entrada por parte del usuario. Sin embargo, depende mucho del contexto humano, por lo que su efectividad depende de la capacidad del sistema de captarlo e interpretarlo. Existen muchos tipos de interfaces de usuario implícitas: multi modal, **gestual**, tangibles, auditivas... etc.

Las **interfaces gestuales** captan información de todo el cuerpo humano, de diferentes tipos de sensores y de nuevos dispositivos de entrada. En películas como en “Minority Report”, publicada en 2002, se empezaron a mostrar rasgos de lo que en un futuro pueden llegar a convertirse las interfaces gestuales, mostrando como con unos guantes permitían al usuario controlar pantallas y ordenadores interactuando a través de hologramas.



Figura 3.1: Escena de la película Minority Report mostrando una interacción gestual.

En la actualidad, existen muchas herramientas de interacción que nos permiten captar este tipo de información para después utilizarla, como por ejemplo: Kinect, **Leap Motion** o Myo. Gracias a estos tipos de dispositivos podemos hacer que los usuarios tengan una interacción con los sistemas de una forma más intuitiva, transparente y natural.

La documentación de esta sección ha sido extraída de: [24]

### 3.3. Los videojuegos educativos

Con el paso del tiempo, la industria de los videojuegos ha crecido enormemente. Los juegos han ido evolucionando hasta comprobarse que no sólo

sirven para divertirse o entretenerte, que con ellos se pueden desarrollar las capacidades cognitivas de los seres humanos y poder aplicarlas en la realidad. Los educadores y las empresas de marketing lo han aprovechado para utilizar las capacidades de los desarrolladores de juegos fuera de la industria del entretenimiento, intentando darles un enfoque didáctico y usarlos para el aprendizaje y el entrenamiento de habilidades. [25]

Los juegos, para los niños, están relacionados directamente con el aprendizaje. A través de ellos, los niños se adaptan al mundo y lo exploran, ampliando sus habilidades y competencias. Sin embargo, cuando van creciendo el camino entre el entretenimiento y el aprendizaje comienza a separarse. Cuando los niños son mayores, empiezan a pensar en los juegos como algo completamente alejado de la educación, pensando en estos sólo como una forma de pasar el tiempo y entretenerte.

Los medios de comunicación han demostrado que son capaces de reunir estos dos propósitos y transmitir al usuario habilidades y contenidos: podemos ver programas educativos de televisión o de radio, música, cómics, medio digitales interactivos... etc. Un claro ejemplo sería la serie de dibujos animados “Barrio Sésamo” [26], considerada como pionera de los programas infantiles en combinar educación y entretenimiento .



Figura 3.2: Personaje de la serie Barrio Sésamo mostrando la diferencia entre lo que significa 1 y 5.

Sin embargo, a menudo los proyectos de educación y de entretenimiento no son tan buenos como esperábamos. Esto se debe a que el éxito de esta unión pasa por combinar una serie de factores: captar la atención del usuario, mostrar de forma visual el cambio que implica lo que se quiere transmitir y reforzar mediante recompensas o ventajas lo que supone adoptar dichos cambios. También es importante llegar a los espectadores mediante

una historia, que se identifiquen con los personajes o tengan empatía con ellos, provocando en ellos la motivación necesaria para que posteriormente busquen información relacionada o hablen con sus amigos sobre los temas tratados. [27]

De esta visión y aprovechando el exponente actual que suponen en la sociedad los videojuegos, surgieron los serious game (videojuegos educativos) que son aquellos cuyo objetivo principal no es la diversión, sino el aprendizaje o la práctica de habilidades, aunque no quitan que dejen de captar la atención al usuario y entretenarlo. Esta técnica de aprendizaje se conoce como **game-based learning**, la cuál se está empleando cada vez más en escuelas de primaria, universidades e incluso en las empresas. [2]

### 3.3.1. Factores imprescindibles en un videojuego educativo

Como hemos comentado antes, para que un contenido educativo y de entretenimiento tenga éxito, se deben de dar una serie de factores. Las características que todo videojuego educativo tiene que tener son:

- **Una historia:** No es imprescindible, pero la mayoría tienen una trama o historia principal. Cuanto más sofisticado sea el argumento y los personajes, más fácil será captar la atención de los jugadores y motivarlos.
- **Gamificación:** Son esenciales las dinámicas de juego como los ranking, sistemas de puntuaciones o las recompensas. Estas dinámicas animan y motivan a los jugadores, provocando una competencia sana con los demás jugadores (o compañeros de clase o trabajo) para esforzarnos más y superarlos.
- **Feedback inmediato e individualizado:** Es importante que el jugador interactúe directamente con el juego teniendo una retroalimentación directa que nos indique si hemos hecho algo bien o mal, incluso detallarnos el error para mejorarlo. Esto es una diferencia esencial con respecto a cualquier clase presencial, ya que en una clase hay muchos alumnos y un solo profesor por lo que el feedback no es tan directo ni personalizado.
- **Simulación:** En la mayoría de los juegos, es importante que haya una relación entre la realidad del juego y la vida real, ya que así el jugador se verá inmerso en un mundo similar al que encontrará en el mundo real. De este modo, se permite a los usuarios entrenar habilidades y adquirir conceptos que luego les serán útiles fuera del juego.

- **El objetivo es aprender:** Todos los elementos mencionados anteriormente se pueden dar en otro tipo de videojuegos que no tengan nada que ver con los serious games, la diferencia esencial reside en la finalidad educativa, a través de aspectos educativos o de captación, que buscan los juegos educativos.

El diseño estético también es importante en un juego, ya que si algo no entra por los ojos ni es visual para el usuario, este lo rechazará por muy útil que sea.

Desarrollar un videojuego de este calibre implica un gran coste, ya que hay que invertir mucho tiempo, dinero y esfuerzo en desarrollar e implementar esta serie de características en cualquier aplicación. Por todos estos temas de viabilidad, desde un principio nuestra idea era el desarrollo de una **aplicación educativa**, buscando aprovechar el pleno auge de la tecnología en el que nos encontramos para captar la atención de los niños y darles un modo **más dinámico, natural e intuitivo** de aprender los conceptos, pero **que sirviera como base** (o prototipo) de lo que realmente podría ser en un futuro un serious game que integrara de forma idónea todos estos factores comentados anteriormente para que los niños realmente quisieran interesarse por saber más.

La documentación de esta sección ha sido extraída de: [2]

### 3.3.2. Ejemplos de videojuegos educativos

A continuación enumeraremos un par de ejemplos de buenos serious games que podemos encontrar en la actualidad:

#### Dragon Box Elements

Es uno de los juegos más sencillos y efectivos en este campo. Se inspira en el libro escrito por Euclides, “Elementos”, la historia se basa en que los jugadores tienen que construir un ejército para derrotar a un enemigo final para poder salvar la isla del propio Euclides. Para ello, los niños tendrán que aprender las bases de la geometría y los teoremas de Euclides, siendo un gran apoyo didáctico para niños a partir de cierta edad. [2] [28]



Figura 3.3: Escena del juego Dragon Box Elements donde se tiene que resolver una prueba geométrica para llegar al enemigo.

### Pulse!!

Este videojuego es un claro ejemplo de lo que significaría la aplicación de serious game al desarrollo de habilidades, ya que este videojuego simula las condiciones de una sala de emergencias de un hospital y tendremos que identificar los problemas de los pacientes, priorizar según la gravedad o aplicar las medidas oportunas. Gracias a este, los futuros enfermeros podrían practicar los conocimientos teóricos adquiridos y ganar experiencia enfrentándose a simulaciones de situaciones reales. [2]



Figura 3.4: Escena del juego Pulse!!.

### 3.4. Aplicaciones actuales

En la actualidad podemos encontrar algunas aplicaciones similares a nuestro proyecto:

- **Tabla Periódica 2018:** Aplicación para dispositivos móviles gratuita que te permite aprender la tabla periódica, teniendo todos los elementos distribuidos por el grupo al que pertenecen y aportando información más específica como su configuración electrónica y enlaces directos a más información. [29]
- **Suite Química Gratuita:** Aplicación para dispositivos móviles gratuita de formulación química bastante completa que te permite aprender la tabla periódica y sus diferentes grupos, aprender detalles de los elementos, calculadora de pesos atómicos a partir de fórmulas, realizar test de formulación orgánica, balanceo de ecuaciones, entre otras funciones. [30]
- **YoFormulo:** Aplicación orientada a aprender a nombrar y a formular algunos compuestos químicos inorgánicos. Se pueden buscar fórmulas y te dirá cual es su nomenclatura en cada uno de los 3 tipos aceptados por la IUPAC ( Unión Internacional de Química Pura y Aplicada ), aprendiendo así la nomenclatura de dichas fórmulas. En la versión pro el usuario puede realizar una serie de test de modo que te dicen la fórmula y tienes que introducir cuál es su nomenclatura según el tipo que hayas seleccionado.[31]
- **Arloon chemistry:** Esta aplicación es de las más completas, es una aplicación educativa para dispositivos móviles y ordenadores portátiles con sistema operativo Windows que tiene como finalidad ser utilizada para aprender a formular y nombrar los compuestos químicos. Permite estudiar la tabla periódica y sus elementos repartidos por grupos, tiene tutoriales que permiten aprender las bases teóricas y reglas de formulación y permite practicar mediante una serie de ejercicios. La aplicación, permite observar las moléculas en 3D conociendo así la disposición espacial de los átomos y el tipo de enlace que forman entre ellos. Además, permite a través de realidad aumentada visualizar el compuesto en 3D sobre una superficie. [32][33]
- **EducAR:** Fue comentada al principio, es la aplicación que ha servido de inspiración para este proyecto, ya que la mecánica es muy similar aunque no sea orientada solo a la formulación química. Esta aplicación es general para más materias de estudio y también permite utilizar la realidad aumentada en uno de sus modos de funcionamiento. También

tiene otro modo de funcionamiento donde usa el dispositivo de interacción gestual Kinect para realizar los ejercicios, entre ellos las fórmulas.

### 3.5. Crítica al estado del arte

Como hemos comentado en una sección anterior, la importancia de la química está perdiendo peso en la sociedad y parte de la culpa se debe a la educación actual.

La interacción hombre - máquina ha evolucionado enormemente desde que se escuchó por primera vez el concepto. Gracias a esto se han ido desarrollando una gran cantidad de dispositivos que facilitan al componente humano la labor de interacción con las máquinas, entre ellos se han desarrollado los dispositivos de interacción gestual. Estos dispositivos permiten interactuar con las máquinas de forma más dinámica y natural gracias a su reconocimiento gestual, sin embargo se tiene que pulir el funcionamiento de todos estos dispositivos de modo que se puedan integrar perfectamente en cualquier sitio sustituyendo a los propios mecanismos actuales.

Como hemos visto, los serious games son una medida que si se consiguen cumplir todos sus factores pueden ayudar enormemente en la educación. Sin embargo, en la actualidad aunque hemos visto que existen muchas aplicaciones educativas para la enseñanza de algunos conceptos de la formulación química, no tenemos un serious game de dicho ámbito.

Este proyecto surge con motivo de realizar una aplicación educativa que sirva como base para un futuro serious game que permita a los jóvenes aprender los diferentes conceptos de la formulación química. Aunque existen muchas aplicaciones que sirven como apoyo didáctico en este campo, una de las ventajas que tiene este proyecto es el uso de un dispositivo de interacción gestual, Leap Motion, que permite que el usuario interactúe directamente con sus propias manos con los diferentes elementos existentes y pueda realizar estructuras químicas haciendo el mismo los enlaces. También permite aprender la nomenclatura correspondiente a la estructura química que queremos crear, así como aprender a situar los elementos en la tabla periódica y conocer a qué grupo pertenecen.

La gran diferencia que tiene con Chemistry es el poder interactuar de forma directa con la aplicación, de modo que toda la interacción se realice haciendo gestos tridimensionales con las manos y permitiendo manipular

así los distintos elementos y enlazarlos para crear las fórmulas químicas. En cambio, la principal diferencia con EducAR viene dada por la herramienta de interacción, ya que aquí toda la interacción se realiza con las manos, permitiendo gracias a Leap Motion una mayor precisión en la interacción.

Sin embargo, el verdadero potencial de este proyecto será si en un futuro se integra con la realidad virtual, donde la interacción con Leap Motion es mucho mejor y más natural, debido a limitaciones que comentaremos en el siguiente capítulo.



## Capítulo 4

# Análisis de requisitos

Como en cualquier proyecto software, tenemos que realizar un análisis del sistema que queremos abordar. Para ello, aunque hemos realizado a lo largo de la memoria una descripción del problema que vamos a tratar sintetizaremos lo más importante en una descripción del sistema y sus objetivos. Despu s identificaremos los actores implicados y la lista de requisitos que deseamos que cumpla nuestro proyecto.

Para la realizaci n del an lisis me he basado en los siguientes apuntes:  
[34]

### 4.1. Descripci n del sistema

Nuestra aplicaci n trata de un sistema que ayudar  a los alumnos a aprender formulaci n qu mica inorg nica. El usuario interactuar  con el sistema usando sus propias manos a trav s del dispositivo Leap Motion.

El sistema ser  capaz de permitir a los usuarios realizar estructuras qu micas seg n convengan, pudiendo navegar a trav s de la tabla peridica y seleccionar los elementos que deseen para posteriormente poder enlazarlos y comprobar si existe dicha estructura qu mica en nuestra base de datos, mostr ndole la nomenclatura correspondiente a dicha estructura.

El sistema tambi n ser  capaz de permitir al usuario elegir un ejercicio a realizar, seleccionando un grupo seg n la clasificaci n en las que se dividen la formulaci n inorg nica y mostr ndole al usuario la nomenclatura de una f rmula perteneciente a dicho grupo. El usuario tendr  que realizar una estructura qu mica seleccionando de nuevo los elementos que crea con-

venientes y enlazándolos, comprobando si la estructura química es correcta o no y si se corresponde con la fórmula del ejercicio propuesto por el sistema.

## 4.2. Actores

A continuación describiremos los actores que interactuarán con el sistema, inicialmente sólo habrá un tipo de actor que será el usuario principal de la aplicación, el alumno, aunque podría estar guiado por un profesor aunque este no interactúe directamente con el sistema, con lo cual no lo consideraremos en la lista de actores. En un futuro se pueden plantear la introducción de actores profesores que sí que interactúen de forma directa con el sistema.

Cuadro 4.1: Actor Alumno

Nombre	Descripción	Tipo	Responsabilidad
Alumno	Representa un usuario del sistema que tiene como finalidad aprender formulación	Usuario del sistema	Interactuar con el sistema realizando estructuras químicas de forma libre o a través de ejercicios guiados

## 4.3. Objetivos

Los objetivos principales que debe cumplir nuestra aplicación son los siguientes:

- **Obj-1:** El sistema tendrá una base de datos con ejemplos de fórmulas y la estructura química que la representa.
- **Obj-2:** El sistema permitirá seleccionar a un usuario un determinado número de elementos de la tabla periódica y almacenarlos en el inventario para que posteriormente el usuario pueda trabajar con ellos.
- **Obj-3:** El sistema generará un archivo con el ejercicio seleccionado por el usuario para posteriormente comprobar si la solución es correcta.
- **Obj-4:** El usuario será capaz de trabajar con los elementos del inventario seleccionados y poder enlazar los distintos elementos.
- **Obj-5:** El usuario podrá comprobar si la estructura química se corresponde con alguna fórmula existente o comprobar si ha realizado correctamente el ejercicio.

## 4.4. Requisitos Funcionales

Estos requisitos describen los aspectos más importantes a nivel de funciones que debe incluir el sistema, realizando una clasificación en categorías. Se les han asignado un código y un nombre, con el fin de identificarlo fácilmente a lo largo del análisis.

**RF-1. Gestión del inventario.** El sistema llevará un control de los elementos que hay en el inventario.

- **RF-1.1. Añadir átomos al inventario.** El usuario será capaz de añadir átomos al inventario seleccionando de la tabla periódica el número de átomos seleccionado. El sistema los almacenará en el inventario.
- **RF-1.2. Eliminar átomos al inventario.** El usuario tiene que ser capaz de eliminar átomos del inventario. El sistema borrará los átomos del inventario.
- **RF-1.3. Sacar átomos del inventario.** El usuario será capaz de sacar átomos del inventario para trabajar con ellos, el sistema tendrá que actualizar el contador del inventario cada vez que se realice una acción de este tipo.
- **RF-1.4. Meter átomos del inventario.** El usuario será capaz de meter átomos del inventario, el sistema tendrá que actualizar el contador del inventario cada vez que se realice una acción de este tipo.
- **RF-1.5. Almacenar datos del inventario.** El sistema almacenará los datos de los elementos del inventario en un archivo, para poder trabajar con ellos entre las diferentes escenas.

**RF-2. Gestión de ejercicios.** El sistema permitirá al usuario seleccionar ejercicios y se encargará del control que implica.

- **RF-2.1. Seleccionar ejercicio.** El sistema ofrecerá al usuario una serie de ejercicios disponibles, de los cuales el usuario podrá seleccionar uno cada vez.
- **RF-2.2. Almacenar datos del ejercicio.** El sistema recogerá la selección del usuario y cargará los datos referentes al ejercicio en un archivo para posteriormente poder usarlo para comprobar de forma directa el estado de la solución.

- **RF-2.3. Mostrar ejercicio.** El sistema buscará el ejercicio y mostrará al usuario la nomenclatura de la fórmula perteneciente al ejercicio seleccionado que tendrá que realizar.

**RF-3. Gestión de soluciones.** El sistema comprobará si las estructuras químicas son correctas.

- **RF-3.1. Comprobar solución ejercicio.** El sistema será capaz de comprobar si la estructura química realizada por el usuario se corresponde o no con la del ejercicio seleccionado.
- **RF-3.2. Comprobar solución formulación libre.** El sistema será capaz de comprobar si la estructura química se corresponde o no con alguna de las existentes en nuestra base de datos.

**RF-4. Gestión de átomos.** El sistema será capaz de gestionar los enlaces realizados por el usuario entre los átomos.

- **RF-4.1. Enlazar átomos.** El sistema será capaz representar que dos átomos están enlazados al ser acercados.
- **RF-4.2. Almacenar enlaces de los átomos.** El sistema será capaz de almacenar los enlaces que relacionan cada átomo.
- **RF-4.3. Romper enlace átomos.** El sistema será capaz de representar la ruptura del enlace entre dos átomos al separarse cierta distancia.
- **RF-4.4. Eliminar enlaces de los átomos desenlazados.** El sistema será capaz de borrar los datos de aquellos enlaces que se hayan eliminado.

**RF-5. Gestión de fórmulas.** El sistema podrá consultar en cualquier momento la información relacionada con las fórmulas almacenadas en la base de datos.

**RF-6. Gestión de archivos.** El sistema tendrá que encargarse de eliminar el contenido de los archivos de inventario y ejercicios al salir de la aplicación o volver al menú inicial.

## 4.5. Requisitos de Información

Estos requisitos indican la información que es necesaria almacenar en el sistema:

**RI-1. Inventario.** El sistema almacenará los datos de los átomos que existen en el inventario en cada momento.

- **Contenido:** Tamaño máximo del inventario y una lista de átomos. Cada elemento de la lista contendrá: tipo de átomo, número de unidades de dicho átomo, grupo al que pertenece y símbolo químico.
- **Requisitos asociados:** RF-1, RF-6.

**RI-2. Base de datos de fórmulas.** El sistema tendrá una base de datos con información de las fórmulas y las estructuras químicas que las representan.

- **Contenido:** Contiene una lista de fórmulas. Cada fórmula contiene: grupo al que pertenece la fórmula, nomenclatura de la fórmula y enlaces necesarios para cumplir su estructura química.
- **Requisitos asociados:** RF-2, RF-3.2, RF-5.

**RI-3. Ejercicio.** El sistema almacenará los datos relacionados con el ejercicio seleccionado por el usuario.

- **Contenido:** El contenido igual al requisito RI-2 pero sólo conteniendo aquellas fórmulas correspondientes con el ejercicio seleccionado por el usuario.
- **Requisitos asociados:** RF-2, RF-3.1, RF-6.

**RI-4. Enlaces.** El sistema tendrá una lista de átomos y los enlaces que tienen.

- **Contenido:** Lista de átomos enlazados. Cada átomo enlazado contiene una lista con el nombre del átomo con el que está enlazado.
- **Requisitos asociados:** RF-3, RF-4.

## 4.6. Requisitos No Funcionales

Estos requisitos incluyen algunas restricciones que afectarán a los requisitos anteriores:

**RNF-1. Interacción.** La interacción del usuario con el sistema se realizará con el dispositivo Leap Motion.

**RNF-2. Número máximo de elementos en el inventario.** Se tendrá que controlar el número máximo de elementos que se guardan en el inventario.

**RNF-3. Conexión con el dispositivo.** Se tendrá que controlar el sistema para que sólo se pueda interactuar con el en caso de estar disponible el dispositivo Leap Motion.

**RNF-4. Plataformas disponibles.** El sistema sólo podrá ser ejecutado en dispositivos con sistema operativo Mac OS o Windows.

**RNF-5. Fiabilidad de la solución.** Cuando se compruebe un resultado los datos deben estar correctamente almacenados sin fallos en la base de datos para poder dar una solución fiable.

## 4.7. Casos de uso

Un caso de uso especifica la secuencia de acciones, incluidas secuencias variantes y de error, que un sistema o subsistema puede realizar al interactuar con actores externos. En la ingeniería del software se utilizan para representar los objetivos que nos interesa que cumpla nuestro sistema. Algunas de sus funciones son:

- Obtener, analizar y especificar requisitos.
- Ser una base para el proceso de diseño y validación.
- Guiar el diseño de la interfaz de usuario.

A continuación vamos a establecer los casos de uso que van a ocurrir en nuestro sistema, para ello usaremos las plantillas de descripción de casos de uso básica.

Caso de uso	Seleccionar elemento de la tabla periódica	CU_01
Actores	Alumno (principal)	
Tipo	Primario — Extendido — Real	
Referencias	RF-1.1	
Precondición	Tener espacio en el inventario.	
Poscondición	Se añade un número determinado átomos de un elemento en el inventario.	
Autor	Juan Ramón Gómez Berzosa	Fecha   03-04-2018   Versión   1.0

**Propósito**

Añadir un número determinados de átomos de un elemento en el inventario para poder trabajar con ellos.

**Resumen**

El alumno abre la tabla periódica y selecciona el grupo de dónde desea seleccionar el elemento. Selecciona el elemento y establece en la calculadora el número de átomos que desea del mismo.

**Curso Normal (básico) de eventos**

Actor	Sistema		
1	El alumno hace un círculo sobre el botón de la tabla periódica con su mano.	2	El sistema muestra la tabla periódica organizada en los distintos grupos que la componen.
3.	El alumno selecciona un grupo.	4.	El sistema muestra los elementos del grupo que hemos seleccionado.
5.	El alumno selecciona el elemento.	6.	El sistema muestra la calculadora para seleccionar el número deseado.
7.	El alumno selecciona una cantidad y le da a OK.	8.	El sistema almacena el número de átomos del elemento seleccionado en el inventario.

**Curso alterno de eventos**

- 7.a El espacio del inventario está lleno.
  - 1. El sistema no almacena el nuevo elemento y no lo muestra en el inventario.
  - 7.b El elemento ya existía en el inventario y al sumarle las nuevas unidades supera el límite.
    - 1. El sistema almacena sólo el número de elementos hasta el límite.
    - 7.c El archivo inventario no existe
      - 1. El sistema no hará nada hasta que lo detecte.

Cuadro 4.2: CU\_01: Seleccionar elemento de la tabla periódica

Caso de uso	Seleccionar un ejercicio	CU_02
Actores	Alumno (principal)	
Tipo	Primario — Extendido — Real	
Referencias	RF-2	
Precondición	Estar en el menú inicial	
Poscondición	Se establece un ejercicio el cuál debe realizar el usuario.	
Autor	Juan Ramón Gómez Berzosa	Fecha   03-04-2018   Versión   1.0

**Propósito**

El alumno selecciona un ejercicio para posteriormente poder realizarlo y comprobar su solución si lo desea.

**Resumen**

El alumno selecciona la opción de Ejercicios y selecciona uno de los distintos grupos en los que se clasifican las fórmulas de la formulación inorgánica para guardarla y posteriormente realizarlo y comprobar si la solución es correcta.

Curso Normal (básico) de eventos			
Actor		Sistema	
1	El alumno hace un círculo sobre el botón ejercicio con su mano.	2	El sistema muestra el menú de selección de ejercicios.
3.	El alumno selecciona un grupo.	4.	El sistema busca en la base de datos una fórmula que pertenezca a dicho grupo
		5.	El sistema almacena en el archivo ejercicio los datos relacionados con la fórmula correspondiente al grupo elegido.
		6.	El sistema carga la escena de formulación poniendo en un panel la nomenclatura de la fórmula a resolver.

**Curso alterno de eventos**

5.a El archivo ejercicio no existe

1. El sistema no hará nada hasta que lo detecte.

Cuadro 4.3: CU\_02: Seleccionar un ejercicio

Caso de uso	Sacar átomo del inventario	CU_03
Actores	Alumno (principal)	
Tipo	Primario — Extendido — Real	
Referencias	RF-1.3	
Precondición	Tener átomos en el inventario	
Poscondición	El átomo estará disponible en la paleta de formulación para trabajar con este.	
Autor	Juan Ramón Gómez Berzosa	Fecha   03-04-2018   Versión   1.0

**Propósito**

El alumno coge un átomo y lo arrastra fuera del inventario para poder trabajar con este.

**Resumen**

El alumno abre el inventario, coge el átomo que necesita y lo arrastra fuera del inventario. Esto permitirá al alumno poder trabajar con este a partir de ahora, permitiendo que enlace con otros átomos.

Curso Normal (básico) de eventos			
Actor	Sistema		
1	El alumno hace un círculo sobre el botón inventario con su mano.	2	El sistema muestra el inventario y los átomos disponibles en este.
3.	El alumno agarra un átomo haciendo el gesto de cogerlo con su mano y lo arrastra fuera del inventario.	4.	El sistema saca el elemento del inventario, actualizando el contador de elementos disponibles en el inventario.

**Curso alterno de eventos**

Cuadro 4.4: CU\_03: Sacar átomo del inventario

Caso de uso	Meter átomo en el inventario	CU_04
Actores	Alumno (principal)	
Tipo	Primario — Extendido — Real	
Referencias	RF-1.5	
Precondición	Tener átomos fuera del inventario	
Poscondición	El átomo estará almacenado de nuevo en el inventario.	
Autor	Juan Ramón Gómez Berzosa	Fecha   03-04-2018   Versión   1.0

**Propósito**

El alumno coge un átomo y lo arrastra dentro del inventario para guardarlo.

**Resumen**

El alumno abre el inventario, coge el átomo de la escena que necesita y lo arrastra dentro del inventario. Esto hará que se guarde el átomo aumentando de nuevo el número de elementos disponibles en el inventario.

Curso Normal (básico) de eventos			
Actor		Sistema	
1	El alumno hace un círculo sobre el botón inventario con su mano.	2	El sistema muestra el inventario y los átomos disponibles en este.
3.	El alumno agarra un átomo haciendo el gesto de cogerlo con su mano y lo arrastra dentro del inventario.	4.	El sistema guarda el elemento en el inventario, actualizando el contador de elementos disponibles en el inventario.

**Curso alterno de eventos**

Cuadro 4.5: CU\_04: Meter átomo en el inventario

Caso de uso	Eliminar átomo	CU_05
Actores	Alumno (principal)	
Tipo	Primario — Extendido — Real	
Referencias	RF-1.2	
Precondición	Tener átomos fuera del inventario y que el inventario esté oculto.	
Poscondición	El átomo será eliminado de la paleta de formulación y del inventario.	
Autor	Juan Ramón Gómez Berzosa	Fecha 03-04-2018 Versión 1.0

**Propósito**

El alumno coge un átomo fuera del inventario y lo arrastra hacia el botón de la papelera para eliminarlo.

**Resumen**

El alumno coge un átomo fuera del inventario y lo arrastra hacia el botón de la papelera para eliminarlo. El átomo será eliminado de la paleta de formulación y del inventario, ya no estando disponible para trabajar con este.

**Curso Normal (básico) de eventos**

Actor	Sistema
1 El alumno arrastra el átomo hasta el botón papelera.	2 El sistema elimina el átomo y actualiza el inventario y el archivo inventario.

**Curso alterno de eventos**

- 1.a El inventario está abierto.
- 1. El usuario saca las manos del inventario y este se ocultará automáticamente cuando pasen 5 segundos.
- 2.a El archivo inventario no existe
  - 1. El sistema no hará nada hasta que lo detecte.
- 2.b El átomo está enlazado con otro
  - 1. Se rompe primero el enlace entre los dos átomos. (CU\_07)

Cuadro 4.6: CU\_05: Eliminar átomo

Caso de uso	Enlazar átomos	CU_06
Actores	Alumno (principal)	
Tipo	Primario — Extendido — Real	
Referencias	RF-4.1	
Precondición	Tener al menos dos átomos fuera del inventario que no estén enlazados entre sí.	
Poscondición	Se creará un enlace entre los dos átomos.	
Autor	Juan Ramón Gómez Berzosa	Fecha 03-04-2018 Versión 1.0

**Propósito**

El alumno coge un átomo fuera del inventario y lo arrastra hacia el otro. Los átomos se enlazan

**Resumen**

El alumno coge un átomo fuera del inventario con sus manos y lo arrastra hacia la posición de otro, aunque puede coger el otro con la otra mano y acercar los dos. El resultado será la generación de un enlace entre los dos átomos.

**Curso Normal (básico) de eventos**

Actor	Sistema		
1	El alumno arrastra un átomo hacia la posición de otro.	2	El sistema creará un enlace entre ambos átomos.
		3.	El sistema almacenará en la lista de enlaces los nuevos enlaces.

**Curso alterno de eventos**

2.a. El enlace genera un ciclo al también enlazar con otro elemento próximo.

1. El enlace no se producirá.

Cuadro 4.7: CU\_06: Enlazar átomos

Caso de uso	Romper enlaces átomos	CU_07
Actores	Alumno (principal)	
Tipo	Primario — Extendido — Real	
Referencias	RF-4.3	
Precondición	Tener dos átomos fuera del inventario que estén enlazados entre sí.	
Poscondición	Se eliminará el enlace entre los dos átomos.	
Autor	Juan Ramón Gómez Berzosa	Fecha   03-04-2018   Versión   1.0

**Propósito**

El alumno coge uno de los dos átomos enlazados y los separa cierta distancia. El enlace entre los dos átomos se rompe.

**Resumen**

El alumno coge uno de los dos átomos enlazados fuera del inventario con sus manos y los separa cierta distancia, aunque puede coger el otro con la otra mano y separar los dos. El resultado será la destrucción del enlace de los dos átomos.

**Curso Normal (básico) de eventos**

Actor	Sistema		
1	El alumno arrastra un átomo lejos del otro al que está enlazado.	2	El sistema eliminará el enlace entre ambos átomos.
		3.	El sistema eliminará de la lista de enlaces los antiguos enlaces.

**Curso alterno de eventos**

Cuadro 4.8: CU\_07: Romper enlace átomos

Caso de uso	Comprobar solución ejercicio	CU_08
Actores	Alumno (principal)	
Tipo	Primario — Extendido — Real	
Referencias	RF-3.1	
Precondición	Tener al menos dos átomos enlazados y haber elegido un ejercicio.	
Poscondición	El sistema dirá si la estructura se corresponde con la del ejercicio o no.	
Autor	Juan Ramón Gómez Berzosa	Fecha   03-04-2018   Versión   1.0

**Propósito**

El alumno comprueba si la estructura química realizada se corresponde con la del ejercicio que hemos seleccionado.

**Resumen**

El alumno comprueba si la estructura química que ha realizado se corresponde con la del ejercicio que hemos seleccionado. En caso de haber acertado dirá que es correcta, en caso contrario dirá que no lo es.

**Curso Normal (básico) de eventos**

Actor	Sistema
1 El alumno hará un círculo con su mano encima del botón solución.	2 El sistema comprobará si la estructura se corresponde con la de la fórmula que hemos seleccionado para el ejercicio. Para ello leerá el archivo ejercicio.
	3. El sistema mostrará si nos hemos equivocado o no.

**Curso alterno de eventos**

2.a No hay átomos enlazados por tanto no hay una estructura química.

1. El sistema nos dirá que no hemos enlazado ningún átomo.

2.b No existe el archivo ejercicio.

1. El sistema no hará nada hasta que lo detecte.

Cuadro 4.9: CU\_08: Comprobar solución ejercicio

Caso de uso	Comprobar solución libre	CU_09
Actores	Alumno (principal)	
Tipo	Primario — Extendido — Real	
Referencias	RF-3.2	
Precondición	Tener al menos dos átomos enlazados.	
Poscondición	El sistema dirá si la estructura se corresponde con la de alguna fórmula de la base de datos.	
Autor	Juan Ramón Gómez Berzosa	Fecha   03-04-2018   Versión   1.0

**Propósito**

El alumno comprueba si la estructura química realizada se corresponde con la del alguna fórmula de la base de datos.

**Resumen**

El alumno comprueba si la estructura química que ha realizado se corresponde con la de alguna fórmula de nuestra base de datos. En caso de ser correcta mostrará la nomenclatura de la fórmula , en caso contrario mostrará que no la ha encontrado.

**Curso Normal (básico) de eventos**

Actor	Sistema
1 El alumno hará un círculo con su mano encima del botón solución.	2 El sistema comprobará si la estructura se corresponde con la de alguna fórmula de las de la base de datos.
	3. El sistema mostrará si existe o no y en caso correcto, mostrará su nomenclatura.

**Curso alterno de eventos**

2.a No hay átomos enlazados por tanto no hay una estructura química.

1. El sistema nos dirá que no hemos enlazado ningún átomo.

2.b No existe el archivo que contiene la base de datos.

1. El sistema no hará nada hasta que lo detecte.

Cuadro 4.10: CU\_09: Comprobar solución libre



## Capítulo 5

# Diseño de clases

Como buen proyecto software, toda aplicación tiene que tener un buen diseño lógico. En este apartado, vamos a mostrar las diferentes clases que componen la aplicación, así como las relaciones existentes entre ellas. Cabe destacar, que algunas clases no estarán relacionadas con otras ya que son controladores y gestionan internamente el funcionamiento de la aplicación, lo cuál especificaré en el apartado de implementación.

En primer lugar, voy a dividir la estructura de clases para que no se solapen todas, de tal forma que las voy a organizar por la forma en la que se utilizan en el sistema. Los proyectos de Unity se basan en escenas las cuales están formadas por un conjunto de objetos y tienen una determinada funcionalidad. Así, vamos a dividir los diagramas de clases de nuestra aplicación en primer lugar por elementos comunes a todas ellas y después por la relevancia de cada clase en una escena.

Empezaremos mostrando las clases predefinidas por los “Assets” de Leap Motion para Unity, las cuales son utilizadas en todas las escenas.

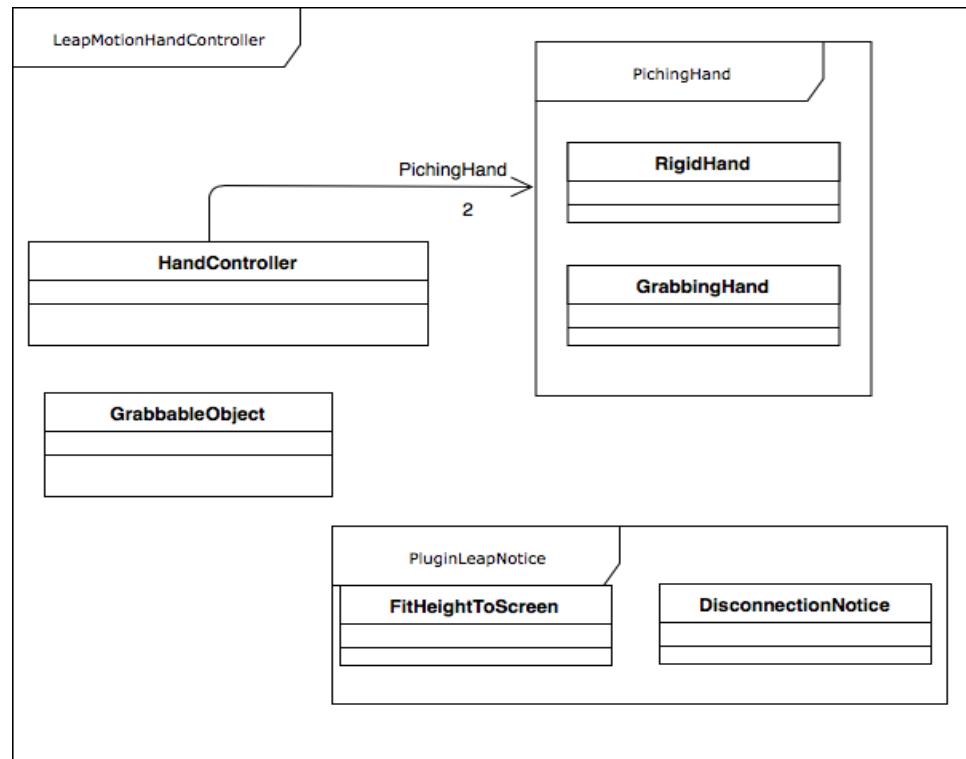


Figura 5.1: `LeapMotionHandController`

A continuación, mostraremos la jerarquía de las clases que controlan los gestos en cada escena. Cabe destacar que todas las clases de la aplicación heredan de una de las clases principales de Unity, Monobehaviour, menos aquellas que heredan primero de otras que ya heredaban de esta.

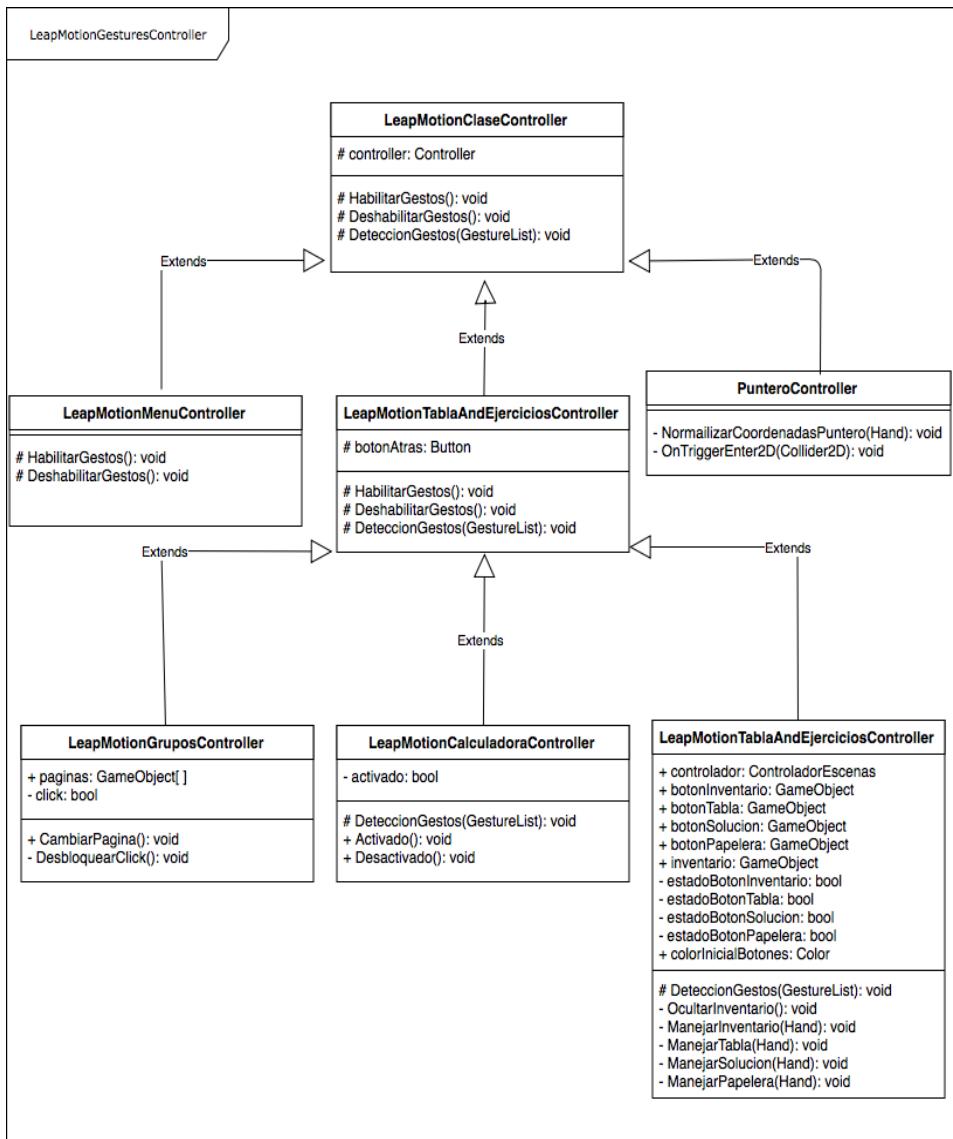


Figura 5.2: LeapMotionGesturesController

Ahora vamos a pasaremos a mostrar el diagrama de clases utilizado por cada escena:

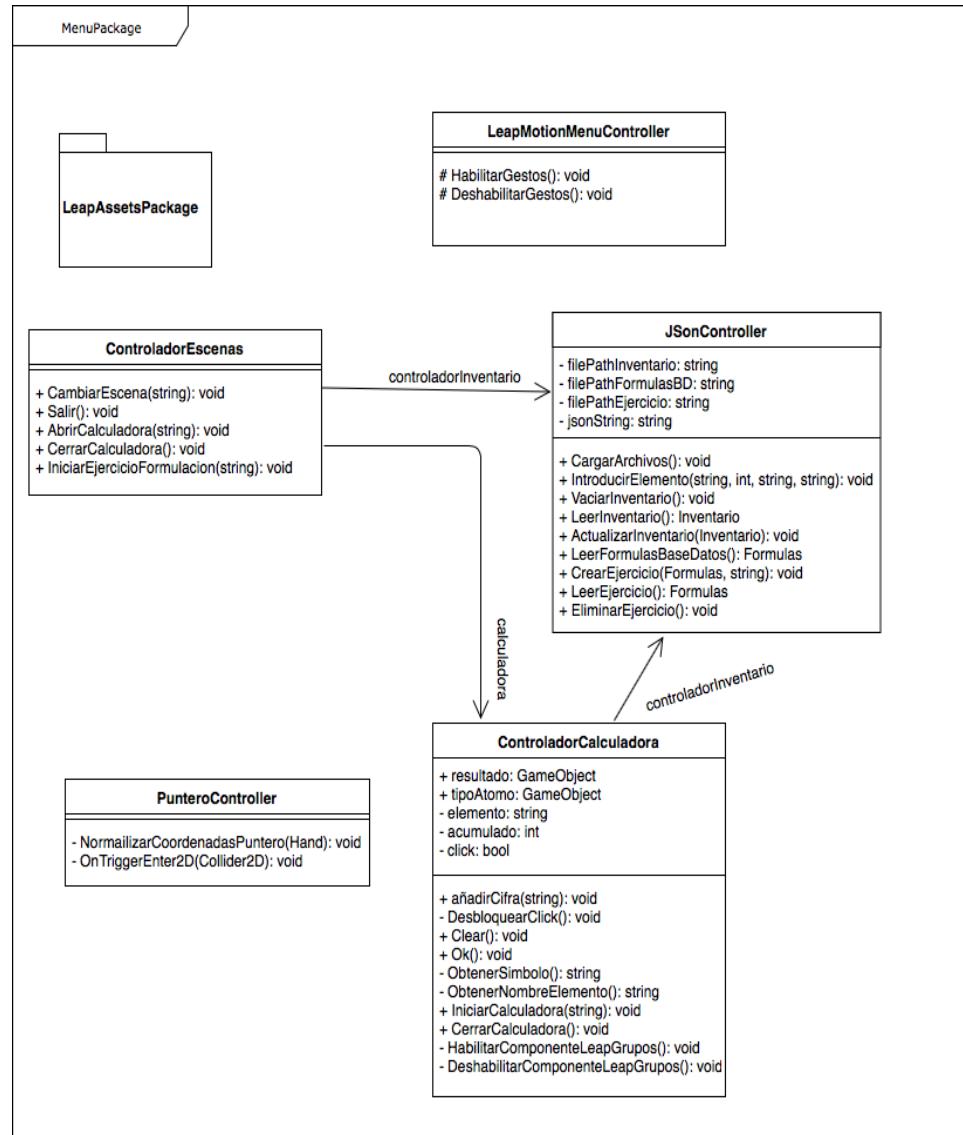


Figura 5.3: MenuPackage

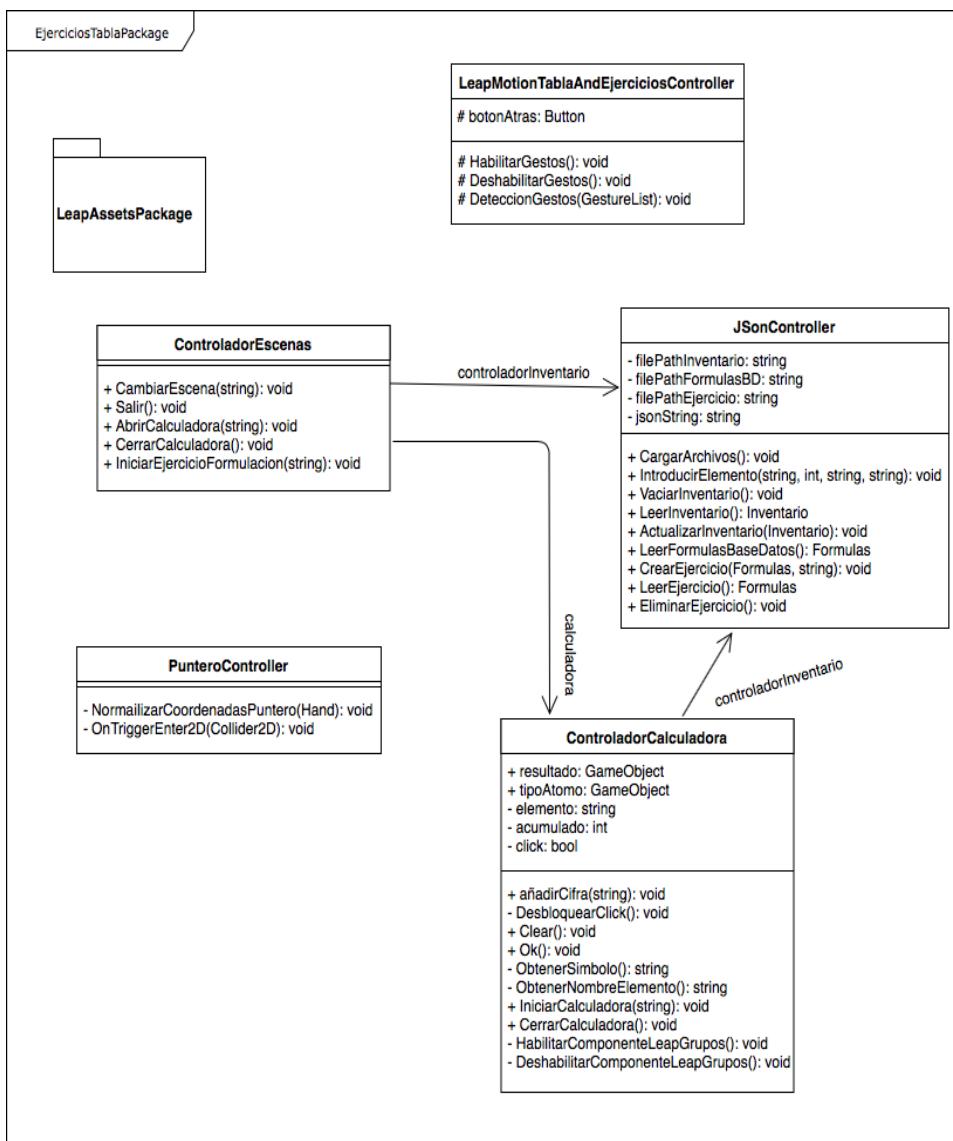


Figura 5.4: EjerciciosTablaPackage

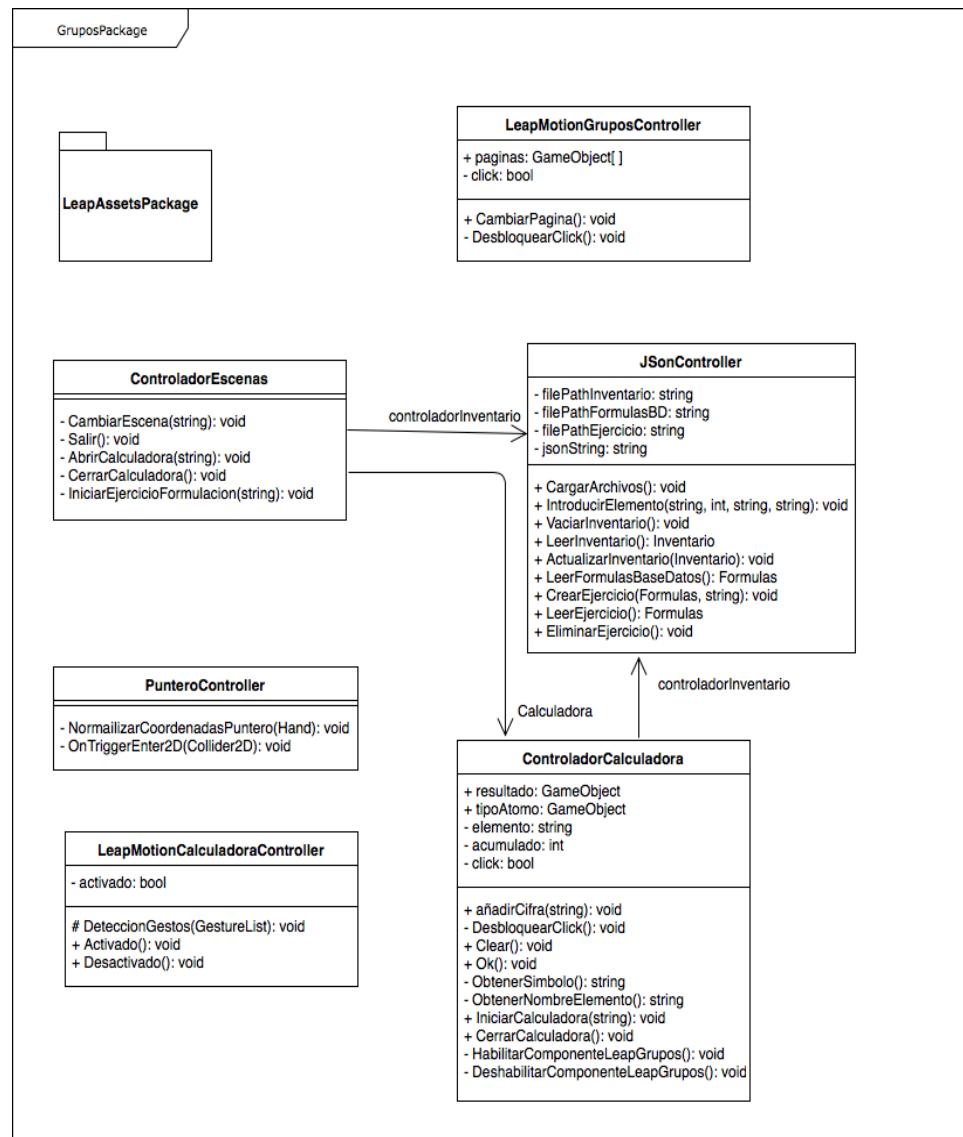


Figura 5.5: GruposPackage

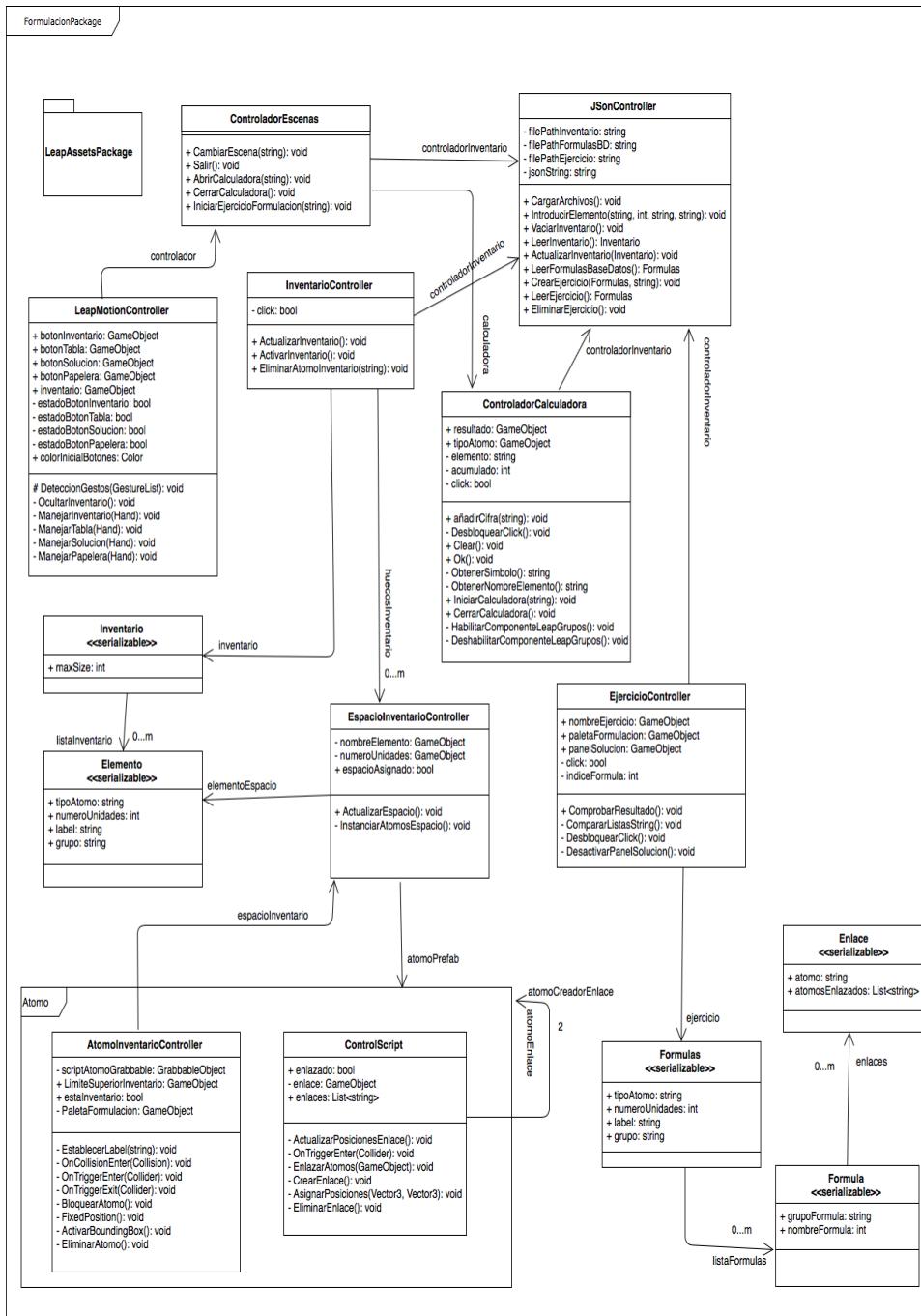


Figura 5.6: FormulacionPackage



# Capítulo 6

## Implementación

Llegamos a una de las partes más importantes de un proyecto, la implementación. ¿Qué son todos estos requisitos, casos de uso o diagramas si no conseguimos aplicarlos y desarrollar una aplicación que los implemente y cumpla con los objetivos que nos planteamos inicialmente?.

### 6.1. Dimensión de desarrollo de la aplicación

En cuanto a la dimensión de desarrollo de nuestra aplicación, todos los menús así como la tabla periódica y los diferentes grupos están en 2D, ya que son básicamente escenas de interfaz de usuario 2D donde la navegación consiste en ir seleccionando opciones o elementos. En cambio, en la escena de formulación tenemos un Pseudo-3D o en otras palabras 2.5D. En esta escena parece que los objetos tienen profundidad, de hecho las propias manos captadas por Leap Motion tienen profundidad, sin embargo se ha limitado a la interacción en 2D (eje x e y) debido a las dificultades de detección que se tenían con Leap Motion y a la complicación que añadiría a los usuarios tener que coger átomos del fondo de una habitación en lugar de una pantalla.

### 6.2. Integración de Leap Motion

Toda la interacción de la aplicación se realiza con Leap Motion, incluida en las pantallas donde “no se ven las manos”, ya que para las pantallas en 2D utilizamos un puntero que se controla con nuestra propia mano.

Los gestos que nuestra aplicación es capaz de detectar se limitan a los

“**Circle**”, hacer un círculo con el dedo, y los “**Swipe**”, movimiento de la mano de un lado a otra de la pantalla, ya que los demás eran más complicados de detectar. El gesto circle está disponible en todas las pantallas, al contrario que el Swipe que sólo está habilitado en determinadas escenas debido a que es más sensible.

### 6.3. Gestión de la base de datos

Para el desarrollo de la aplicación hemos necesitado utilizar archivos que nos sirvieran como base de datos para almacenar datos importantes entre cada pantalla. Para ello, hemos decidido emplear archivos en formato JSON, ya que para nuestra aplicación y el pequeño conjunto de datos que se manejaban y se han introducido en la base de datos de fórmulas, no era necesario el uso de un sistema de gestión de base de datos, aunque en un futuro habría que introducirla. Además, decidimos usar esta manera gracias a las facilidades que nos aportaba Unity para realizar una serialización de clases tanto para guardar datos en este formato o para leerlos.

Por tanto, tenemos 3 archivos esenciales para el correcto funcionamiento de nuestra aplicación:

- **Inventario:** Es el archivo esencial de nuestra aplicación, en este se guardan una lista de elementos que se corresponden con los que vamos seleccionando en nuestra tabla periódica. Almacena el número de átomos de cada elemento, así como propiedades como el nombre o el grupo al que pertenece. Tiene un número limitado de capacidad establecido a la hora del desarrollo.
- **BaseDatosFormulas:** En este archivo sólo se usa como lectura, se guarda una lista de fórmulas químicas inorgánicas las cuales contienen cada una su nombre, grupo de la formulación inorgánica al que pertenecen y una lista de enlaces con otros elementos que se tiene que dar para que se cumpla su estructura química.
- **Ejercicio:** Este archivo almacenará una lista de fórmulas que se correspondan con el grupo de la formulación química que se haya seleccionado en el modo ejercicio. Inicialmente como sólo tenemos una fórmula por grupo pues sólo tendremos una fórmula para la realización del ejercicio, pero en un futuro podremos introducir más.

En el apéndice 3 se muestra una estructura de ejemplo de lo que sería el contenido de cada fichero.

## 6.4. Desarrollo de la aplicación

Unity es un engine que nos permite desarrollar juegos o aplicaciones de una forma bastante intuitiva gracias a su interfaz y el uso de scripting para gestionar el comportamiento de los objetos que se encuentran en las escenas. Las escenas están formadas por un conjunto de objetos que tienen una determinada funcionalidad, por ejemplo: una escena sería un menú de un juego y otra distinta la pantalla del primer nivel. Los objetos se denominan GameObject y se les pueden asignar propiedades como: "Rigidbody" para aplicar físicas, "Colliders" para detectar colisiones, "Scripts" que permitan manejar el comportamiento de los objetos... etc.

En base a todo lo comentado anteriormente, vamos a explicar algunas de las clases más importantes que hemos desarrollado en nuestra aplicación centrándonos en una organización por escenas, ya que esta misma estructura hemos seguido para definir los diagramas de clases.

En primer lugar, antes de centrarnos en cada escena, vamos a comentar de forma genérica las clases de la figura 5.1 y 5.2:

- **Figura 5.1:** Está formada por una serie de clases predefinidas en los paquetes de Leap Motion, que son las que se encargan del renderizado de las manos, tanto físico como gráfico, y de controlar que tengamos el LeapMotion conectado a nuestro ordenador mientras usamos la aplicación. Gracias a la clase PichingHand y a las colisiones con las manos también nos permiten agarrar objetos que tengan la clase GrabbableObject asignada. Estas clases se incluyen en todas nuestras escenas, ya que todo el funcionamiento es a través de las manos.
  
- **Figura 5.2:** En la figura podemos apreciar una jerarquía de clases dónde todas heredan de LeapMotionClaseController. Básicamente esta clase gestiona los gestos que se producen con nuestras manos cuando estamos en una escena y que son captados por Leap Motion, así como las acciones que se deben llevar a cabo cuando ocurren. Sus métodos principales son HabilitarGestos, DeshabilitarGestos y DeteccionGestos, siendo sobreescritos por la mayoría de los hijos ya que en algunas escenas se usan algunos gestos y se gestionan acciones específicas a cada una, por ello también existe un controlador para cada una. Al igual que las de la figura anterior, en cada escena se usará al menos una clase de esta jerarquía, allí comentarán un poco más sus funcionalidades.

### 6.4.1. Escena Menú

Esta será la escena inicial que veremos cuando iniciemos la aplicación. Las clases que se utilizan en esta son las de la figura: 5.3.



Figura 6.1: Escena Menú dónde podemos apreciar el puntero

Principalmente podremos observar tres botones: uno nos llevará al modo de formulación libre (Formular), otro al modo de formulación guiado por un ejercicio (Ejercicios) y el último a cerrar la aplicación (Salir). Vamos a comentar las clases relevantes en esta escena:

- **LeapMotionMenuController:** En esta escena sólo se manejan gestos de tipo círculo, básicamente su funcionamiento consta de detectar círculos y si se hacen sobre un botón, hacer click en él.
- **PunteroController:** Esta clase está asociada a el objeto que representa el puntero dentro de nuestra aplicación. La funcionalidad de esta clase es captar la coordenada x y la coordenada y de la posición de la palma de nuestra mano y hacer una normalización de dichos valores en coordenadas de la escena. Para ello se usa la función “NormalizarCoordenadasPuntero“, la cual hace la normalización comentada usando como referencia las unidades en las que está dividida la escena.
- **ControladorEscenas:** Esta clase gestiona principalmente el cambio de escena, salir del juego, desplegar calculadora y otras funciones según la escena a la que se encuentre asociada.

- **ControladorCalculadora:** La funcionalidad consiste básicamente en controlar el objeto calculadora: abrirla, cerrarla, guardar resultado..., aunque ahora no la explicaré ya que en esta escena no es relevante, sólo en las escenas de los grupos de la tabla periódica que es donde la usaremos.
- **JSonController:** Esta clase gestiona la lectura de archivos en formato JSON que usamos como base de datos y los convierte en objetos de clases serializables. En esta escena se usa al iniciar la escena y al salir, de modo que se borran los archivos de ejercicios y se reinicia el inventario.

#### 6.4.2. Escena Ejercicios

Esta escena nos permitirá seleccionar un grupo de la formulación inorgánica para realizar la estructura química de una fórmula de dicho grupo. Las clases que se utilizan en esta son las de la figura: 5.4.



Figura 6.2: Escena Ejercicios

Básicamente observamos una serie de botones que hacen referencia a distintos grupos de la formulación inorgánica y otro que nos permite ir hacia atrás. Ahora comentaremos las clases relevantes de esta escena, aunque desde este momento las clases que ya hemos comentado su funcionamiento en otras escenas, no las volveremos a comentar:

- **LeapMotionTablaAndEjerciciosController:** Esta clase se denomina así porque es el mismo controlador que usaremos para la siguiente escena, ya que lo que tiene en especial con respecto al anterior es que es capaz de detectar además del gesto circle, el gesto swipe y de realizar una gestión de dicho evento.
- **ControladorEscena:** En esta clase se usa el método IniciarEjercicioFormulación, que además de cargar la siguiente escena, busca las fórmulas que se corresponden con el grupo de formulación inorgánica seleccionado en la base de datos y se encarga de crear el ejercicio para el usuario, usando dos funciones de la clase JSONController. Se utiliza la función “LeerFormulasBaseDatos” se encargada de leer del archivo serializable donde guardamos la base de datos de fórmulas (BaseDatosFormulas) y posteriormente la función CrearEjercicio para seleccionar las fórmulas deseadas y crear la información en el archivo Ejercicio.

#### 6.4.3. Escena Tabla

Esta escena nos mostrará una representación 2D de la tabla periódica dividida por los grupos que la forman. Permitirá seleccionar un grupo de la tabla periódica para consultar los elementos que pertenecen a dicho grupo. Las clases que se utilizan en esta son las de la figura: 5.4, la misma que para la escena Ejercicios, por lo tanto no vamos a volver a explicar su funcionamiento. La única diferencia es que aquí no se llaman a las funciones de la clase JSONController, ya que aquí no tenemos que generar un ejercicio.



Figura 6.3: Escena Tabla

Básicamente observamos una representación de la tabla dividida en varios botones que representan cada uno de los grupos en los que esta se divide. También vemos un botón atrás para poder navegar hacia atrás.

Cabe destacar, que los botones de la tabla han sido creados por nosotros usando la herramienta Photoshop, para que tuvieran un aspecto similar a la representación que tienen los grupos en una tabla periódica real.

#### 6.4.4. Escena Grupos Tabla Periódica

Como hemos comentado, cada vez que seleccionemos un botón de la tabla periódica navegaremos a dicho grupo de la tabla. Cada uno es una escena distinta, aunque su funcionamiento es prácticamente el mismo. La periódica para consultar los elementos que pertenecen a dicho grupo. Las clases que se utilizan en esta son las de la figura: 5.5.



Figura 6.4: Escena del grupo de Metales de Transición

Consta de un botón por cada elemento del grupo de la tabla periódica. En algunos casos cuando todos los elementos no caben en pantalla, existe un botón que nos permite navegar hasta la siguiente página y volver, como si fuera un carrusel. También existe un botón para volver atrás, a continuación comentaremos el funcionamiento de las clases implicadas en la escena:

- **LeapMotionGruposController:** Hereda de la clase LeapMotionTa-

blaAndEjerciciosController. Gestiona los mismos gestos, la diferencia básica está en que permite cambiar cambiar los elementos de la vista al hacer el gesto circle sobre el botón siguiente en caso de no caber en la pantalla.

- **ControladorEscenas:** En esta escena, esta clase no sólo maneja el cambio de escena, si no que también se encarga de cada vez que pulsamos un elemento desplegar y ocultar a calculadora con la función “AbrirCalculadora” y “CerrarCalculadora”. Esto hará uso de la clase **controladorCalculadora**, con el cual usaremos su función “IniciarCalculadora” para desplegarla para poder seleccionar el número de átomos que deseamos de un elemento y la función “CerrarCalculadora” para volver a cerrarla.
- **ControladorCalculadora:** Cuando la calculadora es iniciada, esta clase se encarga de recoger el número que hayamos seleccionado y mostrarlo en ella con la función “AñadirCifra”. Cuando hemos establecido el número y queremos cerrar la calculadora guardando el resultado, la función “Ok” es llamada. Esta función se encarga de usar la clase JSonController para almacenar en el archivo “Inventario” el número de átomos que hemos seleccionado de dicho elemento, usando la función “IntroducirElemento” de la clase JSonController.
- **JSonController:** Esta clase se encarga de almacenar los elementos en el inventario, buscando en primer lugar si ya hay algún átomo de dicho elemento y limitando el número de átomos del mismo elemento que se pueden almacenar.

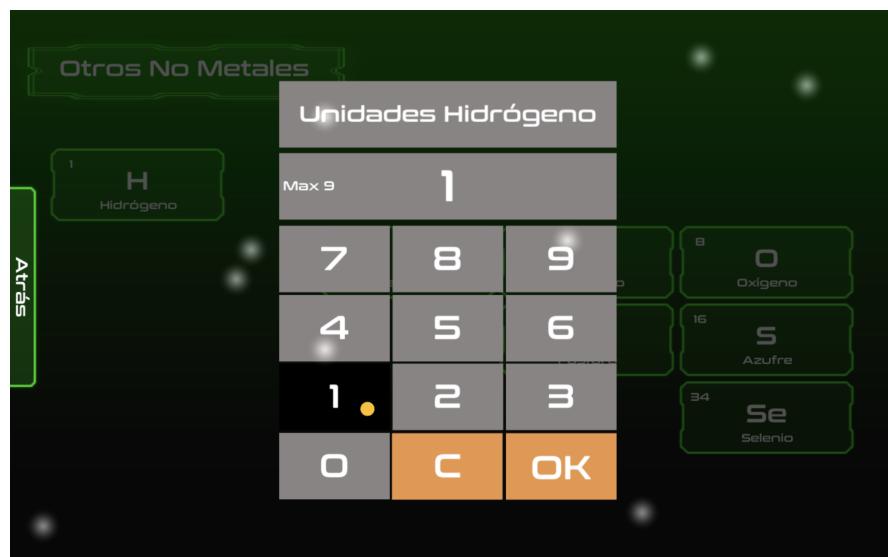


Figura 6.5: Calculadora desplegada

Cabe destacar que todos los fondos están realizados con la herramienta Photoshop.

#### 6.4.5. Escena Formulación

Es la escena más importante de la aplicación. En ella el usuario puede interactuar con sus propias manos agarrando átomos, enlazarlos, desenlazarlos... entre otras funcionalidades. Las clases que se utilizan en esta son las de la figura: 5.6.

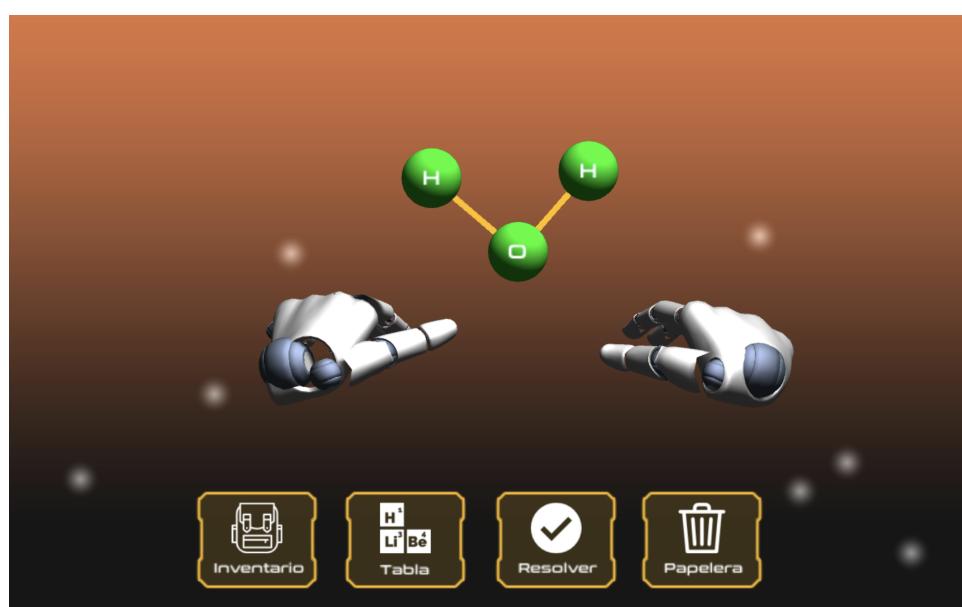


Figura 6.6: Escena formulación con la estructura química de una molécula de agua y nuestras manos captadas por LeapMotion

En la escena podemos observar cuatro botones que nos permiten ver el inventario, acceder a la tabla periódica, comprobar la formula o eliminar un átomo. Además observamos como están nuestras manos renderizadas gracias al Leap Motion y sus assets para Unity, los cuales también nos permiten agarrar los átomos de la escena. Por último, podemos ver como están enlazados una serie de átomos que componen una estructura química. Ahora vamos a comentar las clases que hacen posible todo esto:

- **LeapMotionController:** Hereda también de la clase LeapMotion-TablaAndEjerciciosController. Detecta y gestiona los gestos circle para seleccionar los botones y swipe de izquierda a derecha para volver al menú. Controla el estado de los botones para que cuando nuestras

manos estén en contacto con ellos se iluminen. También controla los cambios de escena o las selecciones de botones con el circle, incluyendo la ocultación del inventario. El inventario se ocultará automáticamente tras 5 segundos cuando saquemos las manos de este o no detecte Leap ninguna, a no ser que volvamos a interactuar con el inventario en ese determinado tiempo.

- **InventarioController:** Esta clase es de las más importantes de la escena, ya que se encarga a través de la clase JSONController, mediante la función “LeerInventario”, de leer el contenido del archivo Inventario para indicarle a cada hueco del inventario de nuestra escena que elemento contiene y cuantas unidades, usando la función “ActualizarInventario”. También es el encargado de desplegar el inventario y de usar la clase JSONController para eliminar un átomo del inventario cuando lo hayamos arrastrado a la papelera.
- **EspacioInventarioController:** Básicamente, su funcionalidad reside en crear el número de átomos del elemento que le ha sido asignado por la clase InventarioController, con la función “InstanciarAtomosEspacio”. También, lleva una gestión del inventario que básicamente controla el contador de átomos en el hueco del inventario.
- **AtomoInventarioController:** Esta clase es una de las que aportan la funcionalidad a cada objeto átomo de la escena, además de la clase **ControlScript**, la cual gestiona el enlazado con otros átomos básicamente, y de **GrabbableObject**, que es una clase de los assets de Leap Motion que permite agarrar objetos que la posean con las manos “PichingHand”. Esta clase controla básicamente si el átomo está en el inventario o en la “paleta de formulación”, permitiendo que se pueda enlazar o que se guarde en el inventario. Todo esto se gestiona a través de colisiones. Además, controla que el átomo se quede fijado en la escena mientras no es agarrado para que no divague por esta debido a la física del objeto, con la función “BloquearAtomo” y permite eliminar el átomo que controla, con la función “EliminarAtomo”.
- **ControlScript:** Esta clase se encarga de enlazar y desenlazar los átomos de la escena. Básicamente realiza un enlace con el átomo que colisiona con la función “EnlazarAtomos”, creando un GameObject enlace entre ambos átomos y actualizando en ellos su lista de enlaces. También permite eliminar enlaces, con la función EliminarEnlace, cuando los átomos se separan más de cierta distancia, para este cálculo se comprueba la distancia euclíadiana entre ambos átomos.
- **EjercicioController:** Esta clase es esencial en la aplicación, es la que se encarga de comprobar si la estructura química que tenemos en la escena se corresponde con la del ejercicio que hemos seleccionado o

con alguna de nuestra base de datos en caso de la selección de la formulación libre. Esto se realiza con la función “ComprobarResultado” la cual extrae los átomos que hay en la escena que están enlazados y genera una lista de enlaces, después utiliza la clase JSonController para leer el archivo ejercicio ( si hemos seleccionado un ejercicio) o el archivo de base de datos de formulas (en caso de elegir formulación libre). A continuación, si hemos elegido realizar un ejercicio comprobará que las listas de enlaces del ejercicio y de la estructura que tenemos en la paleta de formulación son exactamente las mismas, ni más ni menos. En cambio, si hemos optado por la formulación libre, lo que se comprobará es si los enlaces de nuestra estructura química se corresponden exactamente con los enlaces de alguna de nuestras fórmulas de la base de datos. Finalmente mostrará si el resultado es acertado o no activando un GameObject en la escena.

Cabe destacar que todos los iconos que han sido utilizados dentro de los botones en la aplicación han sido extraídos de la página: [35], siendo todos propiedad de esta y permitidos utilizarse referenciándolos. Igualmente la aplicación no se ha usado para lucrarse por lo que tampoco hay problemas.

También hay que mencionar que los sprites que hemos utilizado para los botones forman parte de unos assets gratuitos de la Assets Store: [36].



# Capítulo 7

## Pruebas

Una vez desarrollado el proyecto, es necesario probarlo y comprobar su funcionamiento, obteniendo una serie de datos que nos sirvan para especificar nuestras conclusiones y futuras mejoras. Cabe destacar que en el apéndice A se adjuntará un manual de usuario donde se explique el funcionamiento y una demostración guiada del uso de la aplicación.

### 7.1. Entorno de desarrollo

En primer lugar comentaremos las especificaciones del ordenador donde se ha desarrollado el proyecto y dónde se han realizado todas las de funcionamiento. El ordenador con el que hemos trabajado se trata de el **Mackbook Pro 2016** de 13”, teniendo como características principales:

- Procesador Intel Core i5 de 6a generación de doble núcleo a 2 GHz.
- Memoria RAM de 8 GB DDR3 a 1867 MHz.
- Intel Iris Graphics 540 1536 MB.
- Sistema operativo: MacOS High Siera 10.13.14

Cabe destacar que para el trabajo con Leap Motion hemos utilizado el **V2 Desktop SDK 2.3.1**, el más actual de Leap Motion en su versión para Mac OS. La versión de Unity en la que hemos trabajado ha sido: **2017.3.0.f3 personal** y la versión de los assets de Leap Motion para Unity que hemos usado es la **versión 3.0.0**.

## 7.2. Pruebas internas

A continuación expondremos los datos más relevantes que hemos recogido como objeto de las pruebas realizadas con la aplicación, destacando que han sido de carácter no formal:

- La aplicación es llamativa, los colores son vivos, lo que también atrae la atención, y la interacción con las manos hace que sea más vistosa.
- La implementación del puntero controlado con nuestra propia mano como método de interacción en las escenas de menús y selecciones ha sido una buena idea, ya que si tuviéramos las manos en pantalla en ese momento nos restaría mucha visión de la escena y nos quitaría eficacia a la hora de la realización de gestos. También cabe destacar que así la aplicación se ve más interactiva al realizar todo con el movimiento de nuestras propias manos.
- El hecho de limitar la interacción de nuestras manos en la escena de formulación a los planos x e y, nos aporta una mayor facilidad a la hora de trabajar, ya que de otra manera los átomos se podrían ir al fondo de la escena en la coordenada z y el usuario le costaría más por tanto agarrarlos y manipularlos, ya que habría que gestionar profundidad. Aún así, cabe destacar que las manos si tienen profundidad en z por lo que la primera vez que usemos la aplicación nos costará un poco acostumbrarnos porque sólo podremos interactuar con los átomos si estamos a su mismo nivel de profundidad.
- Leap Motion es un dispositivo que es muy sensible a la luz solar, ya que provoca que se detecten peor los rayos emitidos por su sensor, lo que el entorno ideal de ejecución para una detección idónea sería en una habitación a oscuras. Como hemos podido comprobar en las pruebas, en entornos con luz las manos temblaban más o desaparecían por momentos. También es muy aconsejable el uso de ropa oscura y mangas largas mientras se utiliza.
- Cabe destacar que el rango óptimo de interacción con leap, la interaction box (figura 2.4), a partir de ciertos márgenes la interacción con Leap se dificulta, perdiendo precisión a la hora del renderizado de las manos y de la detección de gestos. Con lo cual, acumular la parte gráfica de la aplicación en sólo estas zona le quita espacio y jugabilidad, dando una impresión no muy buena al tenerlo todo comprimido. Todo esto sumado a lo anterior provoca que el manejo de la aplicación con Leap Motion se dificulte.

### 7.3. Pruebas externas

La finalidad de nuestro proyecto era realizar una aplicación educativa, así que no hay mejor prueba que llevarla directamente a un instituto de secundaria y dejársela utilizar a los alumnos para que valoren por ellos mismos y nos cuenten sus impresiones. Cabe destacar, que al igual que en el apartado anterior las pruebas son de carácter no formal.

Por tanto, el proyecto se llevó al Colegio Cristo Rey, en Villanueva Del Arzobispo (Jaén), para exponer a los alumnos de 4º de la ESO y a su profesora de la asignatura Física y Química la aplicación realizada y permitir que interactuarán con ella para ver sus reacciones y opiniones. Se pidieron los permisos necesarios y nos dejaron grabar tanto la exposición, como a los alumnos interactuando con ella y las opiniones tanto de los alumnos como de la profesora.

La finalidad de este vídeo es ser expuesto ante el tribunal el día de la defensa, ya que uno de los requisitos que se nos pedía de forma estricta en las autorizaciones para que nos permitieran hacer el vídeo con los alumnos, era que el vídeo no se difundiera por Internet. Las autorizaciones están adjuntas en el apéndice B.

En primer lugar, se realizó una presentación de diez minutos aproximadamente sobre los aspectos generales del proyecto, incluyendo motivación, precedentes, descripción de la aplicación y planteamiento de la idea de futuro. Posteriormente se hizo una demostración de uso explicándoles a los alumnos como se debía utilizar el sistema y que se podía hacer, para a continuación dejar probarlo a todo el que quiso. Para finalizar se pidió la opinión a los alumnos y a su profesora acerca de la aplicación realizada quedándose todo esto resumido en el vídeo que se expondrá el día de la defensa ya que no se podía exponer el vídeo de la hora entera que duró el acto.

Algunas de las impresiones extraídas de la prueba son las siguientes:

- En primer lugar, la aplicación llamó mucho la atención a los alumnos, ya que la idea de formular utilizando un aparato que representaba el movimiento de sus manos en la pantalla era muy llamativo para ellos. Todos ellos querían probarla.
- Adaptarse a la sensibilidad con la que había que realizar los gestos o interactuar usando el Leap Motion les costaba bastante al principio, ya que al no estar acostumbrados a la áreas óptimas de detección y

a tener que poner las manos en la posición encima del dispositivo les resultaba un poco incómodo.

- Todos coincidieron en que la forma de interactuar usando gafas de realidad virtual sería mucho más natural e intuitivo.

# Capítulo 8

## Conclusiones

Hace falta echar un vistazo hacia atrás cuando estamos llegando al final para realmente darnos cuenta de todo el trabajo que hemos realizado. El trabajo que implica un TFG parte desde el inicio, desde la duda que genera la elección de un tutor como del planteamiento de una idea que sea válida para el proyecto y que te permita poder realizarlo en el tiempo que tienes previsto, hasta cuando das el último retoque gramatical y ortográfico a la memoria del proyecto.

A medida que hemos ido desarrollando el TFG, hemos pasado por diferentes fases de la misma importancia, tanto de documentación como de desarrollo software e implementación, sin contar con todas las pruebas de ensayo y error y depuraciones que han tenido que ser realizadas para logar llegar a un estado funcional de la aplicación. Por no contar lo importante que era que al final se consiguieran cumplir los objetivos iniciales, consiguiendo presentar un proyecto de calidad que no sólo fuera bueno, si no que nos permitiera aprender a lo largo de su desarrollo.

### 8.1. Asignaturas relacionadas

Ahora comentaremos cuáles han sido las asignaturas cursadas en la carrera que más nos han servido como apoyo para la realización del proyecto. Aunque cabe destacar, que realmente todas las asignaturas que se han cursado han aportado algo, ya que gracias a todas ellas hemos aprendido a enfrentarnos a situaciones complicadas y buscar soluciones hasta la saciedad, hemos aprendido a documentarnos de forma correcta o a saber redactar memorias de forma correcta, entre otras cosas. Las asignaturas más utilizadas han sido las siguientes:

- **Nuevos Paradigmas de Interacción (NPI):** Gracias a esta asignatura conocimos la existencia de los dispositivos de interacción gestual comentados a lo largo de la memoria y empezamos a manejarlos en otros entornos de desarrollo, pero sirviéndonos como base.
- **Computación Ubicua e Inteligencia Ambiental (CUIA):** Esta asignatura nos ha permitido conocer más a fondo los conceptos de la interacción hombre máquina y todo lo que conlleva.
- **Programación Lúdica (PLD):** Con ella, hemos aprendido conceptos básicos para la realización de un videojuego, además se plantea la realización de un videojuego para las prácticas de la asignatura usando un motor de juegos, lo que le da una mayor importancia ya que lo realizamos también con Unity.
- **Fundamentos de Ingeniería del Software (FIS):** La asignatura nos ha permitido realizar un buen análisis software del proyecto, especificando de forma correcta los requisitos, casos de uso y diagramas de clases que hemos implementado.
- **Programación y diseño orientado a objetos (PDOO: )** En ella realizamos un videojuego del que nos ha servido a la hora de pensar en el proyecto y en como estructurararlo.
- **Estructura de datos (ED):** Nos ha servido fundamentalmente para la definición de la estructura de datos para almacenar la información y la experiencia de programación en lenguaje C++, que aunque Unity trabaje con C# realmente es muy similar a C++. Aunque cabe destacar que no sólo está, si no también todas las asignaturas anteriores y posteriores donde hemos usado este tipo de lenguajes como Fundamentos de programación o Metodología de programación entre otras, las cuales no sólo nos han aportado una base de programación en C++, si no que nos han ayudado a aprender a estructurar de forma correcta las clases y seguir una cierta metodología.

## 8.2. Valoración de los objetivos

Desde el principio planteamos tres objetivos principales:

- Desarrollar una aplicación educativa que facilitará la enseñanza y el aprendizaje de la química, permitiendo a los alumnos darles un enfoque más dinámico y llamativo a la hora de introducirse en este campo de estudio.

- Aprender a usar el dispositivo de interacción gestual Leap Motion y conseguir que toda la interacción con la aplicación se realizara a través de este.
- Aprender a desarrollar aplicaciones o videojuegos con el motor de juegos Unity3D, consiguiendo integrar Leap Motion para desarrollar toda la funcionalidad de la aplicación educativa a través de esta y permitiera al usuario darle un enfoque más gráfico.

Podemos concluir asegurando que hemos cumplido nuestros objetivos iniciales, ya que en primer lugar hemos conseguido realizar una aplicación educativa que pueda ayudar a aprender química con más o menos nivel, pero permitiendo realizar todas las funciones que planteamos inicialmente a los usuarios. Se ha conseguido utilizar Leap Motion con el desarrollo de la aplicación, comprobando el amplio abanico de oportunidades que nos ofrece y estudiando sus limitaciones. Por último, hemos aprendido a integrar todo esto y desarrollar la aplicación usando el engine Unity3D, comprobando la infinitud de alternativas, facilidades y soluciones que nos aporta a la hora del desarrollo de juegos o aplicaciones.

### 8.3. Conclusiones del desarrollo y la aplicación

En este apartado, vamos a valorar el estado final del proyecto. Gracias a las pruebas, sobre todo gracias a la exposición en el instituto, hemos podido concluir que la aplicación llama mucho la atención de los alumnos y hace que de verdad se interesen por utilizarla, olvidándose de que realmente están aprendiendo química interreactuando con la aplicación.

Sin duda, el uso de una tecnología de detección gestual como lo es Leap Motion ha provocado uno de los mayores focos de atención del proyecto, ya que el poder interactuar con la aplicación con una representación gráfica de lo que realmente estás intentando con tus manos hace que sea mucho más natural y que parezca que realmente estás haciendo tu dichas fórmulas en la realidad. Sin embargo, colocar las manos de modo que siempre se sitúen dentro del pequeño rango de la interaction box del dispositivo, así como la incomodidad de tener las manos en una posición incómoda encima del dispositivo teniendo que guardar ciertas distancias provocando una posición antinatural, nos ha llevado a la conclusión de que el futuro de la aplicación y por tanto del uso de Leap Motion está en el empleo de la realidad virtual. El empleo de la realidad virtual permitiría que lleváramos el dispositivo montado en las gafas de realidad virtual y así pudieramos interactuar de forma inmediata con nuestras manos de frente en el mundo virtual.

A pesar de todo esto, la aplicación fue un éxito entre los alumnos y entre los profesores que estuvieron presentes, incluida la profesora de Física y Química del colegio. Es una manera diferente de dar las clases permitiendo que sean mucho más dinámicas por lo que verían bien su introducción, aunque para ello deberían de pulirse algunas dinámicas e introducir la realidad virtual para aumentar la naturalidad de la interacción.

#### **8.4. Valoración personal**

En cuanto a nuestra valoración personal, el proyecto ha resultado ser más complejo de lo que pensamos al principio. Ha implicado mucho tiempo para formarnos en el desarrollo de juegos con Unity, consultando mucha documentación y tutoriales, ya que en ninguna asignatura te enseñan a usarlo. También ha llevado mucho tiempo entender el funcionamiento de Leap Motion, acostumbrarse a trabajar con él y saber exactamente las clases y funciones que nos interesaban utilizar de este, ya que tiene muchísimas clases, funciones y datos que no hemos utilizado pero que están disponibles.

Por todo esto y múltiples errores y contratiempos que siempre surgen, al final hizo que nos retrasáramos más de lo esperado, pero siempre hemos tenido como meta cumplir una serie de objetivos y hacerlo en unos cuadros de tiempo específicos, Junio.

Para concluir, hemos conseguido realizar, siempre hablando desde nuestra opinión, un proyecto consistente que cumple con las bases de lo planteado inicialmente, permitiendo llegar al alumno desde un enfoque distinto de la asignatura de modo que resulte ser mucho más dinámico y natural, y que deja abiertos una inmensidad de caminos para su mejora en un futuro.

# Capítulo 9

## Trabajos Futuros

Como hemos mencionado en la conclusión, el planteamiento del proyecto y el campo que abarca deja abiertas múltiples vías de mejora para el futuro, permitiendo añadir los elementos necesarios para convertirse en un completo serious game o incluso seguir su camino como aplicación educativa. A continuación, vamos a comentar algunos de los trabajos que se pueden realizar en el futuro, aunque puede haber muchos más que aún no hemos pensado.

### 9.1. Mejoras en la aplicación

Aunque la aplicación se encuentra en un estado funcional, en caso de querer llevarla al mercado e incluso implantarla en una aula, habría que modificar algunas partes para darle más funcionalidad.

- **Base de datos:** Actualmente tenemos una base de datos limitada, con 7 fórmulas almacenadas. En un futuro se podría gestionar la idea de utilizar una base de datos propia en la nube o usando algunas de las existentes, permitiendo hacer búsquedas eficientes mediante selectores.
- **Gestión de usuarios:** Se podría tener un acceso a la aplicación por usuarios, distinguiendo entre alumnos y profesores. De este modo los profesores podrían generar ejercicios de formulación y subirlos a la nube para que los alumnos se los pudieran descargar y realizarlos desde sus propias casas o desde las propias aulas. También, así permitiría que los alumnos tuvieran sus ejercicios personalizados por usuario.
- **Gestión de notas para los ejercicios:** Permitiría a los alumnos asignarles unas notas en función de los ejercicios realizados.

- **Introducción de formulación orgánica :** Modificar el funcionamiento para la introducción de formulación orgánica y no sólo inorgánica.
- **Mejora de las funcionalidades básicas:** Introducir algunas mejoras a la aplicación como obligar al usuario a realizar los enlaces de forma que se formen los ángulos correctos entre los átomos. También permitir rotar la molécula por ejemplo con el movimiento de nuestra mano para posteriormente hacer el enlace por esa zona.

## 9.2. Realidad Virtual

Uno de los trabajos futuros más importantes y que inicialmente se planteó para el desarrollo de este, fue la realización de la aplicación en realidad virtual.

El problema, como comentamos inicialmente, era el gran desembolso económico y la gran capacidad gráfica y computacional que requería unas gafas como Oculus Rift o HTC Vive, resultando totalmente imposible el desarrollo del proyecto usándolas. Además de no poder de usar gafas de corte más económico como lo son CardBoard al no poder desarrollar para dispositivos Android por no estar disponible el sdk de Leap Motion en la actualidad, así que se planteo dejar como un trabajo futuro.

Sin embargo, este trabajo futuro resulta esencial para que la aplicación crezca, ya que nos permitiría romper con la barrera que suponen las limitaciones y las posiciones antinaturales a las que nos fuerza la versión de escritorio de Leap Motion.

Cabe destacar que para su desarrollo habría que esperar a que Leap Motion terminará de sacar el SDK para poder trabajar con dispositivos Android.

## 9.3. Serious Game

Como comentamos al principio, la aplicación se puede considerar como un prototipo de serious game al que le faltan los factores principales que permiten atribuirle dicho rango. Por tanto, como trabajo futuro quedaría decidir toda esta serie de factores, ya que la mecánica básica de la aplicación si que está desarrollada.

Se podría desarrollar una historia que permita a los jugadores adentrarse en el juego y captar la atención de estos y motivarlos. Por ejemplo, se podrían poner niveles del juego dónde hubiera fuego y para apagarlo necesitas generar una molécula de agua.

También se tendría que permitir al usuario ganar una serie de puntos en función del número de fórmulas correctas o de la rapidez en la que se resuelve, así como al superar niveles, permitiendo provocar una competencia sana entre los jugadores.



# Bibliografía

- [1] Estudio sobre el uso de los videojuegos como herramienta para el desarrollo de habilidades cognitivas. <http://ride.org.mx/1-11/index.php/RIDESECUNDARIO/article/viewFile/789/771>.
- [2] Definición y propiedades de los serios game. <https://www.game-learn.com/lo-que-necesitas-saber-serious-games-game-based-learning-ejemplos/>.
- [3] Definición de aplicación educativa. <https://www.aula1.com/apps-educativas/>.
- [4] Plataforma de leap motion con información del futuro desarrollo del sdk para dispositivos android. <https://developer.leapmotion.com/android#107>.
- [5] Información sobre el proyecto educar. enlace.
- [6] Video de presentación de educar. <http://media.ugr.es/index.php/categorias/28-ingeneria-y-arquitectura/440-que-es-educar>.
- [7] Información de kinect - projectnatal. <https://web.archive.org/web/20120121223600/http://download.microsoft.com/download/A/4/A/A4A457B3-DF5D-4BF2-AD4E-963454BA0BCC/ProjectNatalFactSheetMay09.zip>.
- [8] Artículo sobre microsoft kinect y sus efectos en la sociedad . <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6190806>.
- [9] Artículo sobre la precisión de leap motion . <http://www.mdpi.com/1424-8220/13/5/6380/htm>.
- [10] Definición de electromiografía. <https://www.healthline.com/health/electromyography#purpose>.
- [11] Estudio del uso del brazalete myo para fisioterapia. <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7439933>.

- [12] Información sobre oculus rift 1. <https://www.oculus.com/blog/powering-the-rift/>.
- [13] Información sobre oculus rift 2. <https://www.oculus.com/blog/the-oculus-rift-oculus-touch-and-vr-games-at-e3/>.
- [14] Información sobre htc vive. [http://alex.vlachos.com/graphics/Alex\\_Vlachos\\_Advanced\\_VR\\_Rendering\\_GDC2015.pdf](http://alex.vlachos.com/graphics/Alex_Vlachos_Advanced_VR_Rendering_GDC2015.pdf).
- [15] Artículo sobre google cardboard. <https://www.cnet.com/news/facebook-has-oculus-google-has-cardboard/>.
- [16] Antonio Bautista Bailón Morillas. Programación lúdica: Motores de juego. <http://decsai.ugr.es/~bailon/clases/PL/desarrollo.html#/32>.
- [17] Página web oficial de unity. <https://unity3d.com/es>.
- [18] Página web oficial de unreal. <https://www.unrealengine.com/en-US/what-is-unreal-engine-4>.
- [19] Unity vs. unreal. <https://www.deustoformacion.com/blog/diseño-producción-audiovisual/pros-contras-programar-unity-vs-unreal-engine>.
- [20] Artículo sobre las metodologías de desarrollo ágil como una oportunidad para la ingeniería del software educativo <http://www.redalyc.org/html/1331/133115027022/>.
- [21] Definición de química. [http://www.educa.madrid.org/web/ies.mateoaleman.alcala/TEORIA\\_AQ\\_alumnos.pdf](http://www.educa.madrid.org/web/ies.mateoaleman.alcala/TEORIA_AQ_alumnos.pdf).
- [22] Historia de la química como herramienta didáctica. [http://www.eduquim.com/pdfs/herradon\\_historia\\_ensenanza\\_rev.pdf](http://www.eduquim.com/pdfs/herradon_historia_ensenanza_rev.pdf).
- [23] Física y química en la educación. <http://www.rsme.es/comis/educ/senado/f3.pdf>.
- [24] Antonio Bautista Bailón Morillas. Computación ubicua y ambiental: Interacción hombre - máquina.
- [25] Información acerca de los serious games 1. <http://www.citeulike.org/group/3106/article/1426623>.
- [26] Sinopsis de barrio sésamo. <https://www.filmaffinity.com/es/film788960.html>.

- [27] Información acerca de los serious games 2. [https://books.google.es/books?hl=en&lr=&id=eGORAgAAQBAJ&oi=fnd&pg=PP1&dq=serious+game&ots=1v3us1RkZB&sig=b6Kq4JvSIdrXb2fiTzVDPrpZ4-Y&redir\\_esc=y#v=onepage&q=serious%20game&f=false](https://books.google.es/books?hl=en&lr=&id=eGORAgAAQBAJ&oi=fnd&pg=PP1&dq=serious+game&ots=1v3us1RkZB&sig=b6Kq4JvSIdrXb2fiTzVDPrpZ4-Y&redir_esc=y#v=onepage&q=serious%20game&f=false).
- [28] Página oficial de dragon box elements . <https://dragonbox.com/products/elements>.
- [29] Aplicación para android tabla periódica 18. <https://play.google.com/store/apps/details?id=mendeleev.redlime>.
- [30] Aplicación para android suite química 18. [https://play.google.com/store/apps/details?id=es.mafn.chemdroidcp&hl=es\\_419](https://play.google.com/store/apps/details?id=es.mafn.chemdroidcp&hl=es_419).
- [31] Aplicación para android yoformulo. <https://play.google.com/store/apps/details?id=es.jarodriguez>.
- [32] Información de la aplicación chemistry de la empresa arloon. <https://www.educaciontrespuntocero.com/recursos/chemistry-la-app-para-transformar-las-clases-de-quimica-en-secundaria/24728.html>.
- [33] Página web oficial de la aplicación chemistry de la empresa arloon. <http://www.arloon.com/apps/arloon-chemistry/>.
- [34] Ana Anaya Morito. Fundamentos de ingeniería del software: Ingeniería de requisitos.
- [35] Referencia de la página de dónde hemos extraído los iconos para nuestra aplicación. <https://www.flaticon.com>.
- [36] Assets para los botones de nuestra interfaz. <https://assetstore.unity.com/packages/essentials/unity-samples-ui-25468>.



## Apéndice A

# Manual de usuario

A continuación, se va a comentar el funcionamiento de la aplicación y lo que se puede realizar con ella. Nuestra aplicación consta de dos modos de funcionamiento:

- **Formulación Libre:** Permite realizar estructuras químicas enlazando los elementos que se deseen de la tabla periódica, pudiendo comprobar si dicha estructura se corresponde o no con una formula química existente en nuestra base de datos.
- **Ejercicios:** Permite seleccionar uno de los diferentes grupos en los que se dividen las formulas pertenecientes a la formulación inorgánica según su estructura. Cuando sea seleccionada, pedirá que se realice la estructura química correspondiente a una formula perteneciente a dicho grupo y se podrá comprobar si la solución es o no correcta.

La aplicación permite navegar por los diferentes grupos de la tabla periódica, seleccionando los elementos que necesitamos y almacenándolos en el inventario para más tarde poder enlazarlos en la escena de formulación. También, permite romper los enlaces si se separan dos átomos lo suficiente e incluso eliminar los elementos que no se necesiten del inventario o de la escena de formulación. Toda esta interacción se realiza a través de las propias manos del usuario, usando sus movimiento y gestos captados por Leap Motion.

Los gestos con los que podemos interactuar usando nuestras manos son los siguientes:

- **Circle:** Encima de los botones para seleccionar algo. Permite seleccionar cualquier botón de la interfaz usando este gesto.

- **Swipe:** En todas las pantallas para ir hacia atrás o volver al menú. Aunque esto también se puede realizar en la mayoría con circle sobre el botón.
- **Grabb:** Que aunque no sea un gesto, al cerrar la mano imitando el gesto de coger nos permitirá manipular los átomos.

Para la realización del proyecto se han elegido sólo formulas correspondientes a la formulación inorgánica. Dentro de los grupos de formulación inorgánica se han añadido fórmulas correspondientes a:

- Óxidos no metálicos
- Sales neutras
- Sales volátiles
- Peróxidos
- Oxoácidos
- Hidruros Volátiles

Además de estas se ha añadido el agua como sustancia no clasificada.

## A.1. Ejemplo de uso

En primer lugar mostraremos la escena inicial, la escena menú. En esta podemos elegir uno de los dos modos de funcionamiento o salir de la aplicación haciendo uso del gesto circle con nuestra mano situando el puntero sobre lo que queramos seleccionar.



Figura A.1: Escena menú.

En el menú de ejercicios se permite seleccionar uno de los grupos anteriormente mencionados para poder realizar el ejercicio, de este modo la aplicación busca la fórmula correspondiente a dicho ejercicio en la base de datos y la carga, mostrándole al usuario la nomenclatura de la fórmula a resolver. Para seleccionar un grupo podemos utilizar el gesto circle sobre este.



Figura A.2: Escena ejercicios.

Una vez realizado este paso, iremos a la escena de formulación que estará totalmente vacía.



Figura A.3: Escena formulación con ejercicio de peróxidos.

Ahora iremos desplegaremos la tabla periódica llevando nuestra mano hasta la posición del botón Tabla y haciendo un gesto circle.



Figura A.4: Escena tabla.

A continuación navegaremos con el puntero hasta el grupo de Otros no metales para seleccionar los átomos de hidrógeno y de oxígeno para realizar el agua oxigenada. Para ello haremos un gesto circle sobre el grupo con el puntero.

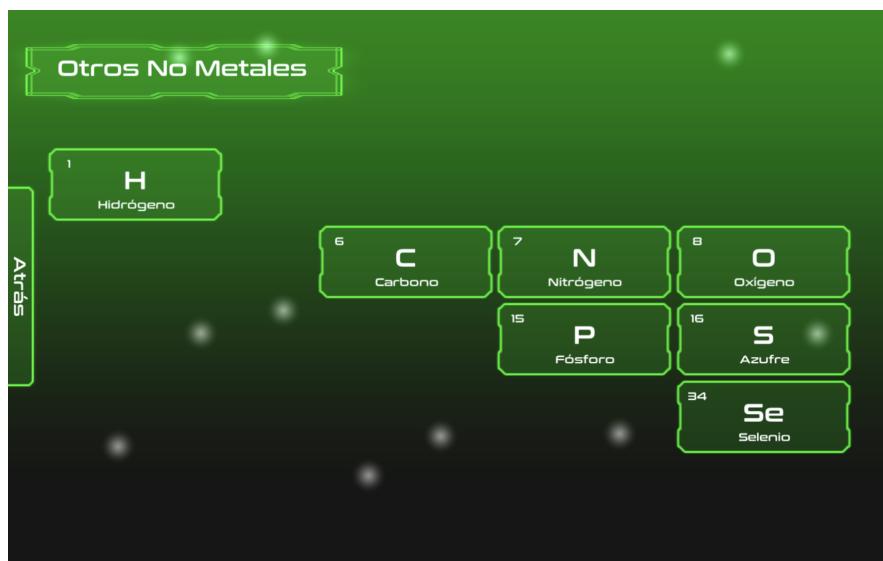


Figura A.5: Escena Otros No Metales.

Seleccionamos el oxígeno e introducimos en la calculadora el número que necesitamos para llevar a cabo la fórmula (2), haciendo un gesto circle sobre

el número correspondiente.



Figura A.6: Seleccionando dos átomos de oxígeno.

Después haremos lo propio con el hidrógeno y volveremos a la escena de formulación. Cuando despleguemos el inventario haciendo swipe sobre este, nos aparecerán cargados los elementos que hemos seleccionado.

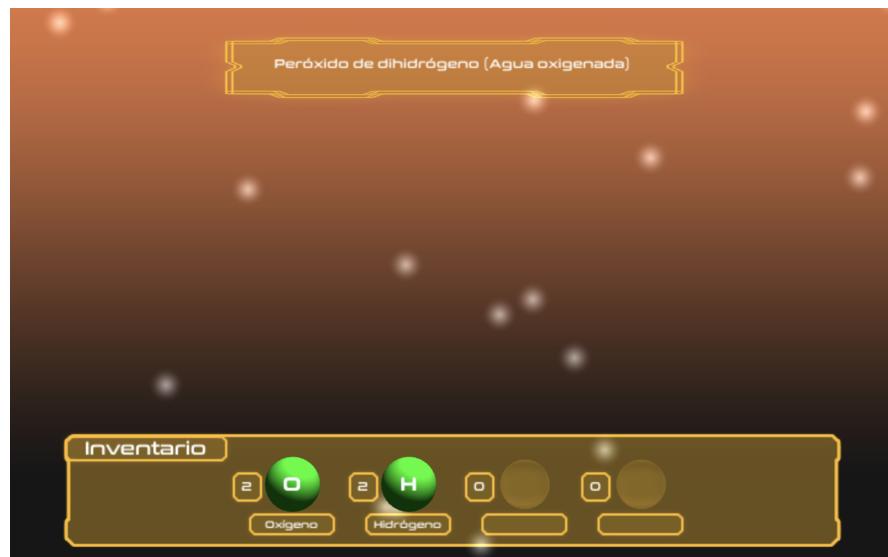


Figura A.7: Inventario con átomos seleccionados.

Ahora realizaremos la estructura.



Figura A.8: Estructura química del agua oxigenada.

Una vez realizada la estructura química que deseamos, podemos comprobar si el ejercicio es correcto usando el botón solución, haciendo un gesto circle.

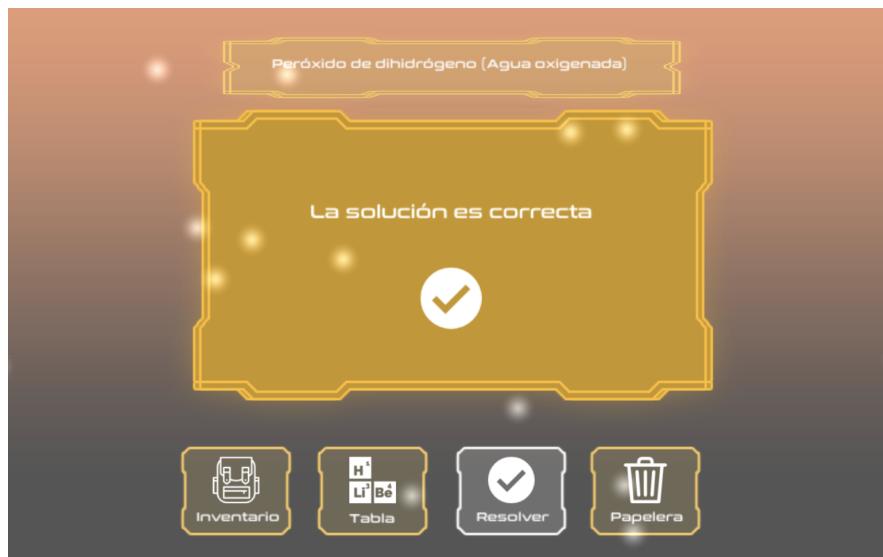


Figura A.9: Solución correcta.

En caso de haber elegido el modo de formulación libre, el botón solución nos dirá si la estructura química se corresponde con alguna existente en nuestra base de datos, mostrando su nomenclatura.

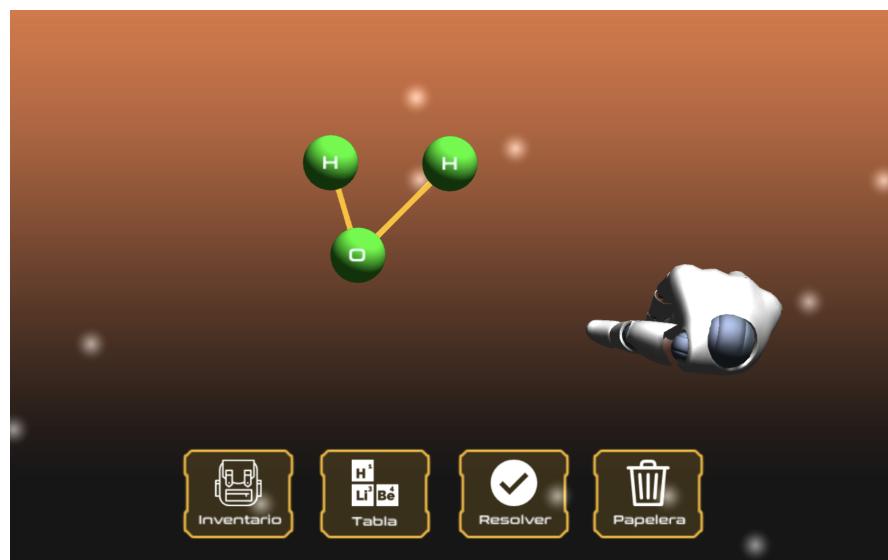


Figura A.10: Estructura química del agua.

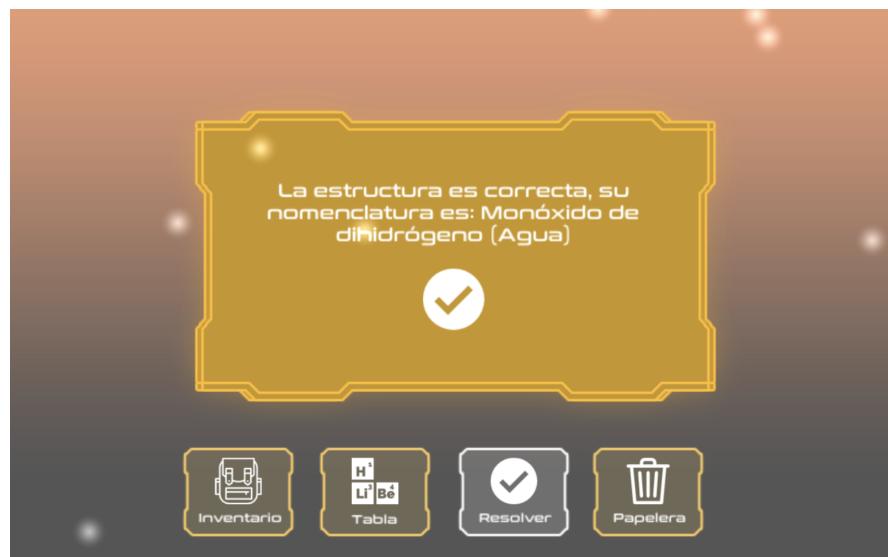


Figura A.11: Solución encontrada.

## A.2. Diagrama de flujo de escenas de la aplicación

Ahora adjuntaremos un diagrama de flujo que muestre la conexión entre escenas de nuestra aplicación.

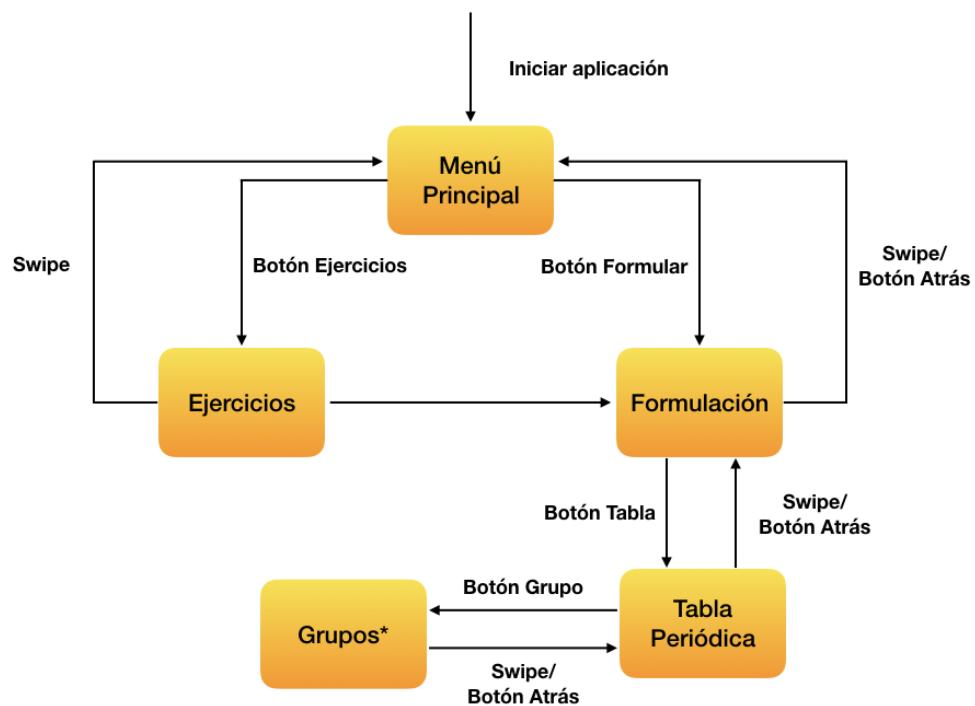


Figura A.12: Diagrama de flujo de escenas.



## **Apéndice B**

### **Estructura de los ficheros JSON de datos**

```
{
  "maxSize":4,
  "listaInventario": [{"tipoAtomo":"Hidrógeno","numeroUnidades":2,"label":"H","grupo":"OtrosNoMetales"}, {"tipoAtomo":"Oxígeno","numeroUnidades":1,"label":"O","grupo":"OtrosNoMetales"}]
}
}
```

Figura B.1: Archivo Inventario

```
{"listaFormulas":[
  {"grupoFormula":"Peroxidos","nombreFormula":"Peróxido de dihidrógeno (Agua oxigenada)", "enlaces":[
    {"atomo":"Hidrógeno","atomosEnlazados":["Oxígeno"]}, {"atomo":"Hidrógeno","atomosEnlazados":["Oxígeno"]}, {"atomo":"Oxígeno","atomosEnlazados":["Hidrógeno","Oxígeno"]}, {"atomo":"Oxígeno","atomosEnlazados":["Hidrógeno","Oxígeno"]}]
  ]
}
]
```

Figura B.2: Archivo Ejercicio

```
{"listaFormulas":[
  {
    "grupoFormula":"NoCatalogados",
    "nombreFormula":"Monóxido de dihidrógeno (Agua)",
    "enlaces": [
      {
        "atomo":"Hidrógeno",
        "atomosEnlazados": [
          "Oxígeno"
        ]
      },
      {
        "atomo":"Hidrógeno",
        "atomosEnlazados": [
          "Oxígeno"
        ]
      },
      {
        "atomo":"Oxígeno",
        "atomosEnlazados": [
          "Hidrógeno",
          "Hidrógeno"
        ]
      }
    ],
  }
},
```

Figura B.3: Ejemplo de una fórmula del fichero baseDatosFormulas

