Proyecto Fin de Grado

# KVLeap

An application for touchless computer interaction

Kevin Villalobos Rodríguez

21 de junio de 2015

Directora:

María Aranzazu Illarramendi Echabe


# Agradecimientos

Como se suele decir, lo importante es lo primero, y creo que no hay nada más importante que reconocer y agradecer su ayuda a todos aquellos que nos la han prestado. Es por eso que me gustaría empezar esta memoria dando las gracias a todos los que de una forma u otra me han ayudado tanto en el desarrollo de este proyecto, como en mi paso por el grado en Ingeniería Informática.

Gracias a mi directora de proyecto María Aranzazu Illarramendi Echabe por haberme brindado la oportunidad de realizar este proyecto con ella, por la confianza que ha depositado en mí desde el principio del proyecto y por lo involucrada que ha estado durante todo el proyecto.

Gracias a David Antón, que sin conocernos siquiera, se ofreció a ayudarme de forma completamente desinteresada, y que finalmente acabó ejerciendo de subdirector y de cliente del proyecto.

Gracias a Alfredo Goñi por habernos ayudado con el artículo enviado a las JISBD, a los *testers* de KVLeap por sus aportaciones e ideas que han ayudado a mejorar la aplicación, y a todos los que en mayor o menor medida han intervenido en este proyecto.

Gracias a mis compañeros, tanto a los de Vitoria como a los de Donostia, y en especial a aquellos con los que más trato he tenido, por los momentos que hemos compartido dentro y fuera de las aulas.

Gracias a los profesores con los que me he encontrado durante estos años, por enseñarme las materias de sus asignaturas, y lo que es más importante, a aprender por mí mismo.

Gracias a María Aranzazu Illarramendi Echabe y a José Miguel Blanco Arbe, por orientarme y ayudarme cuando he tenido alguna duda o algún problema en mi paso por la facultad.

Gracias a mi padre y a mi madre por todas las facilidades y el apoyo que me han dado para poder estudiar este grado en Ingeniería Informática.

Y por último, gracias a la persona que está leyendo esta memoria, por el interés mostrado en el trabajo que hemos realizado y por el tiempo empleado en leer este documento.

## Resumen

Hoy en día existen diferentes alternativas para interactuar con los ordenadores. Sin embargo, las más extendidas y utilizadas son el teclado y el ratón. En ambos casos resulta necesario que las manos del usuario entren en contacto con algún dispositivo, ya sea un teclado físico o un ratón. En determinadas circunstancias en las que la higiene de las manos es un factor importante, este hecho puede suponer un inconveniente. En este proyecto de fin de grado se ha desarrollado KVLeap, una aplicación de escritorio para los sistemas Windows, que usando el controlador Leap Motion, un dispositivo que detecta y rastrea la posición y los movimientos de las manos en el aire, permite interactuar con un ordenador sin que las manos del usuario tengan que entrar en contacto con ningún dispositivo.



**Palabras clave:** Interacción sin contacto, Leap Motion, Sistemas de reconocimiento gestual

## Abstract

Nowadays there are different ways to interact with computers. However, the most widespread and used are keyboard and mouse. In both cases it is necessary that the user´s hands enter in contact with a device, either a physical keyboard, a mouse or a touch screen. In certain circumstances where hands hygiene is an important factor, this may be an inconvenient. In this final degree project it has been developed KVLeap, a desktop application for Windows systems, which using Leap Motion controller, a device that detects and tracks hands position and movements in the air, allows users to interact with a computer without having the hands in contact with any device.



**Key words:** Touchless computer interaction, Leap Motion, Gesture recognition systems

# Índice general

# Índice de figuras

# Índice de tablas

# Capítulo 1

# Introducción

La forma en la que los seres humanos se comunican e interactúan con los ordenadores ha ido evolucionando a lo largo de los años. Esta evolución viene precedida por la evolución de las interfaces de usuario, la parte del ordenador que se encarga de gestionar la interacción con el usuario, ya que a medida que se han ido desarrollando nuevas interfaces de usuario, los usuarios se han ido adaptando a ellas.

Las interfaces de usuario que han aparecido a lo largo de este proceso evolutivo pueden ser clasificadas en generaciones según Nielsen (Nielsen, 1993 [24]). La primera de estas generaciones, a la que denomina generación prehistórica abarca hasta 1945, en esta generación no existía el software y los usuarios interactuaban directamente con el hardware de las máquinas. Entre 1945 y 1955 aproximadamente, estaría la generación del procesamiento por lotes. Esta generación se caracteriza porque la ejecución de los programas, no necesita ningún tipo de interacción con el usuario. Los 10 próximos años le corresponderían a la generación de la interacción por línea de comandos o CLI (Command Line Interface), en esta generación el usuario interactúa con el ordenador por medio de comandos. Las siguientes dos generaciones, conocidas como la tradicional y la moderna, pertenecerían a las GUI (Graphical User Interface), e irían desde 1965 hasta 1980 y desde 1980 hasta 1995 respectivamente. La primera de ellas tiene menús y formularios, y en la segunda aparecen también las ventanas, los iconos y el puntero. La generación moderna también es conocida como WIMP (Windows Icons Menus and Pointer). Finalmente, quedaría la generación actual, a la que Nielsen denomina como generación futura.

En esta generación, predomina el uso de las interfaces gráficas junto con el uso del teclado y el ratón como forma de interactuar con las personas. No obstante, en los últimos años han ido apareciendo interfaces de usuario alternativas que buscan romper con el patrón clásico del uso del teclado y el ratón. A las interfaces de usuario de esta nueva generación se las conoce como NUI (Natural User Interface). Estas interfaces permiten al usuario interactuar con un ordenador sin la necesidad de utilizar los sistemas de mando o los dispositivos de entrada de las GUI tales como el teclado y el ratón. En su lugar hacen uso de diferentes alternativas como las pantallas táctiles, sis-

temas de reconocimiento de voz... Entre estas alternativas podemos encontrar los sistemas de reconocimiento de gestos, una tecnología en auge que permite al usuario interactuar con el ordenador mediante movimientos y gestos.

En este contexto es donde surge KVLeap, una aplicación que permite a los usuarios interactuar con un ordenador utilizando simplemente los movimientos y gestos de sus manos.

KVLeap es el producto obtenido del proyecto realizado por Kevin Villalobos Rodríguez bajo la dirección de María Aranzazu Illarramendi Echabe durante el curso académico 2014/2015, y este documento es la memoria que recoge todo el trabajo realizado durante el ciclo de vida del proyecto, que está estructurado de la siguiente forma:

1- Introducción y contextualización del proyecto, mediante un paseo a lo largo de la historia de las interfaces de usuario.

2- Los objetivos del proyecto junto con la necesidad que satisface y el alcance del mismo.

3- El contexto tecnológico en el que se enmarca el proyecto, algunos de los sistemas de reconocimiento de gestos, y en particular Leap Motion, que es el que se utiliza en este proyecto.

4- Las tecnologías que se han utilizado para desarrollar el producto, así como las alternativas a éstas y las razones que llevaron a su elección frente a las alternativas existentes.

5- Las funcionalidades y las características del sistema desarrollado.

6- La arquitectura y la estructura del sistema desarrollado.

7- Como se ha gestionado el proyecto.

8- Conclusiones del trabajo realizado.

9- Líneas futuras de la aplicación, de los sistemas de reconocimiento gestuales y de las interfaces de usuario.

# Capítulo 2
# Objetivo del proyecto

En este capítulo se explica el objetivo del proyecto, así como la necesidad que ha impulsado el desarrollo del mismo, y se determina cual es el alcance de éste, detallando qué es lo que incluye el proyecto y lo que no.

## 2.1 Problema a resolver

Hoy en día existen diferentes alternativas para interactuar con los ordenadores. Las alternativas más extendidas y más utilizadas son el teclado y el ratón. Sin embargo en ambos casos para poder utilizar estas alternativas, es necesario que las manos del usuario entren en contacto con algún dispositivo, ya sea un teclado físico, un ratón, una pantalla táctil... En determinadas circunstancias, en las que la higiene de las manos es un factor importante, esto puede ser un inconveniente.

Un contexto en el que la higiene de las manos es un factor importante es un quirófano. Mediante esta aplicación los médicos podrían utilizar el ordenador sin tener que entrar en contacto con el ratón ni con el teclado. Pero también puede darse el caso contrario, en el que el usuario tiene las manos sucias y tiene que limpiárselas para poder utilizar el ordenador, como puede ser en un taller de coches, donde los mecánicos se manchan las manos con grasa continuamente.

Enmarcado en este tipo de contextos, se ha planteado crear una aplicación que permitiera utilizar los datos ofrecidos por el controlador Leap Motion para interactuar con un ordenador.

## 2.2 Objetivo del proyecto

El objetivo de este proyecto es utilizar el dispositivo Leap Motion como un dispositivo de entrada de datos que permita al usuario interactuar con un ordenador de la misma forma que lo haría con un teclado y un ratón, utilizando únicamente los gestos y movimientos de sus manos. Para ello KVLeap interpretará los datos capturados por el controlador Leap Motion y realizará las acciones equivalentes a las que un usuario realizaría con el teclado o el ratón. No es objetivo de este proyecto reemplazar el teclado y el ratón, sino desarrollar una aplicación *touchless* que permita interactuar con el ordenador en entornos en los que el contacto de las manos del usuario con dispositivos físicos pueda suponer un inconveniente.

## 2.3 Alcance del proyecto

El alcance del proyecto está orientado a ofrecer al usuario la funcionalidad necesaria para realizar mediante movimientos y gestos de sus manos todas las *acciones básicas que puede realizar con un ratón y un teclado normal*. Por una parte, con el objetivo de simular la funcionalidad del ratón, la aplicación incluye las siguientes funcionalidades:

- Establecer una conexión con el controlador Leap Motion y recibir los datos de las posiciones, movimientos y gestos de las manos que éste ha rastreado e interpretarlos para generar los eventos que permitirán interactuar con el ordenador.
- Interpretar los datos recibidos para dibujar y mover mediante el movimiento de la mano, un puntero que simula el cursor del ratón a través de la pantalla.
- Realizar cualquiera de los tipos de clic básicos que un ratón normal permite realizar (clic con el botón izquierdo, clic con el botón derecho, doble clic con el botón izquierdo y clic y arrastrar con el botón izquierdo) mediante gestos y movimientos de las manos del usuario.
- Realizar la acción asociada al tipo de clic realizado, abrir/ejecutar ficheros, seleccionar ficheros, mostrar el menú contextual, mover un fichero...
- Realizar la acción asociada al movimiento de la rueda del ratón mediante gestos del usuario.

Por otra parte, la aplicación ofrece las siguientes funcionalidades con el objetivo de simular la funcionalidad del teclado.

- La aplicación cuenta con un teclado virtual que el usuario podrá mostrar u ocultar.
- La aplicación permite utilizar el teclado virtual para escribir los caracteres que aparecen en la tabla de caracteres de la aplicación *(Tabla 9)*, así como para generar los comandos asociados a las combinaciones de teclas que aparecen en esa tabla.
- La aplicación permite utilizar el teclado virtual como dispositivo de entrada para escribir sobre otras aplicaciones como el navegador, Microsoft Word, el bloc de notas...

### 2.3.1 Inclusiones del alcance

En el alcance del proyecto, además de la funcionalidad necesaria para poder realizar las acciones básicas que se pueden realizar con el teclado y el ratón, se incluyen los siguientes puntos:

- Desarrollo e implantación de un algoritmo *(Apéndice A)* basado en el algoritmo T9 para la predicción de texto.
- La posibilidad de mostrar/ocultar una barra con palabras predichas por el algoritmo de predicción de texto en función de las teclas que el usuario ha pulsado.
- Un fichero ejecutable para realizar la instalación de la aplicación.
- Una ventana de configuración mediante la que el usuario puede activar o desactivar algunas de las funcionalidades de la aplicación y ajustar algunos parámetros.
- 3 formas diferentes de realizar los diferentes tipos de clic admitidos por la aplicación, dos de ellas mediante la detección de gestos que realiza el usuario con las manos, y la tercera en función del tiempo que pasa el puntero en una determinada posición.
- Una base de datos con un total de 885.418 palabras para la predicción de texto.
- Un manual de usuario que recoge las instrucciones necesarias para poder utilizar la aplicación.
- El desarrollo de un sitio web con el objetivo de dar visibilidad al proyecto.

### 2.3.2 Exclusiones del alcance

No se incluyen en el alcance del proyecto ninguno de los siguientes puntos:

- Corrector ortográfico para las palabras que el usuario ha escrito.
- El desarrollo e implementación de cualquier otro algoritmo que no sea el especificado, ni el uso de servicios o aplicaciones externas para la predicción de palabras.
- Inclusión de paquetes de idiomas para la predicción de palabras.
- Instalación y configuración de los programas requeridos por la aplicación para el correcto funcionamiento de la misma.
- El desarrollo de la aplicación para otros sistemas operativos que no sean Windows ni garantías de compatibilidad con versiones anteriores a Windows 7.
- Por motivos ajenos al proyecto, queda excluido del alcance un buen rendimiento de la aplicación y el correcto funcionamiento de la misma en equipos que no cumplan con las especificaciones mínimas requeridas por el software de Leap Motion *(Figura 1)*.
- Queda excluido del alcance la solución a cualquiera de los problemas conocidos del SDK (Software Development Kit) de Leap (Leap Motion [17]).
- Cualquier otra funcionalidad que un usuario pueda realizar con el ratón o el teclado y que no esté en los puntos del alcance del proyecto.

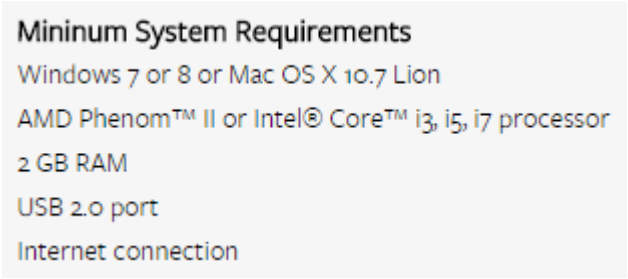


*Figura 1: Especificaciones mínimas requeridas por el software de Leap Motion*

# Capítulo 3

# Contexto tecnológico

Este proyecto se enmarca en el contexto tecnológico de los sistemas de reconocimiento de gestos. Estos sistemas permiten a los seres humanos comunicarse con máquinas e interactuar con ellas de forma natural, sin la necesidad de entrar en contacto con ningún tipo de dispositivo, simplemente mediante gestos y movimientos de su cuerpo.

Los gestos que se pueden utilizar para interactuar varían en función del sistema escogido, algunos como Kinect (Kinect [13]) capturan gestos y movimientos de todo el cuerpo del usuario, mientras que otros se centran en partes específicas del cuerpo tales como la cara y las expresiones faciales o en el caso de Leap Motion, las manos.

La tecnología subyacente en estos sistemas también es heterogénea, aunque la tendencia está en el uso de cámaras y técnicas de visión por computador para la detección de los gestos.

En este capítulo se presenta el sistema de reconocimiento gestual escogido para este proyecto que es Leap Motion, y las posibilidades que ofrece este dispositivo, así como las tecnologías que se han utilizado para desarrollar el proyecto.

## 3.1 Leap Motion

El controlador Leap Motion es un dispositivo que se conecta a un ordenador mediante el puerto USB (Universal Serial Bus) y permite detectar el movimiento natural de las manos en el aire, rastreando las posiciones de los dedos y las manos con una precisión detallada. Además de los dedos y las manos Leap también rastrea objetos que puedan servir como punteros, como por ejemplo un lápiz.

Este controlador cuenta con dos cámaras y tres LEDs (Light Emitting Diode) infrarrojos que rastrean la luz infrarroja con una longitud de onda de 850 nanómetros (fuera del espectro de luz visible).

Gracias a sus lentes de gran angular, el dispositivo cuenta con un amplio espacio de interacción de unos 226.534 centímetros cúbicos (8 pies cúbicos) como el que se ve en la *Figura 2*. El rango de visión del controlador está limitado a unos 2 pies por encima del dispositivo, por la propagación de la luz LED a través del espacio, ya que es mucho más difícil inferir la posición de la mano en 3D más allá de una cierta distancia. En última instancia, la intensidad de la luz LED está limitada por la corriente máxima que recibe a través de la conexión USB.

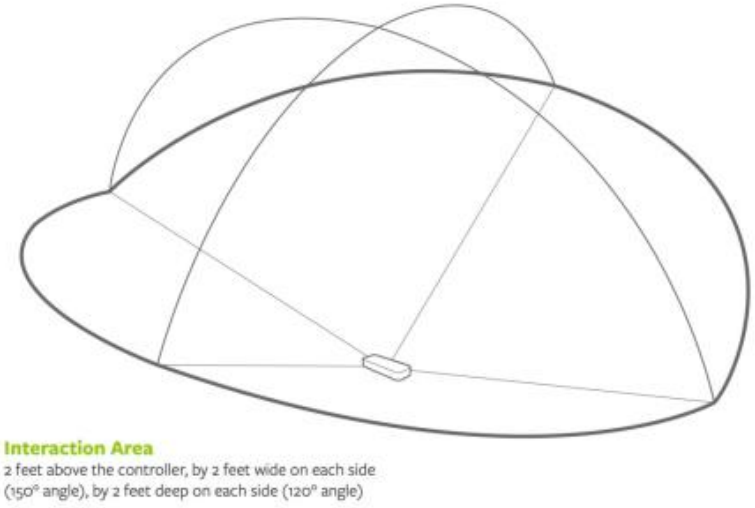


*Figura 2: Área de interacción Leap Motion: 2 pies (60 centímetros) por encima del controlador, por 2 pies de ancho a cada lado (con un ángulo de 150º), por 2 pies de profundidad a cada lado (con un ángulo de 120º)*

El controlador lee los datos que recibe de los sensores junto con los que generan las cámaras (capaces de generar hasta 300 *frames* por segundo, en los que se incluyen datos e imágenes) y los almacena en su propia memoria, donde realiza los ajustes necesarios antes de enviarlos a través de la conexión USB al software de Leap Motion instalado en el ordenador al que está conectado el controlador.

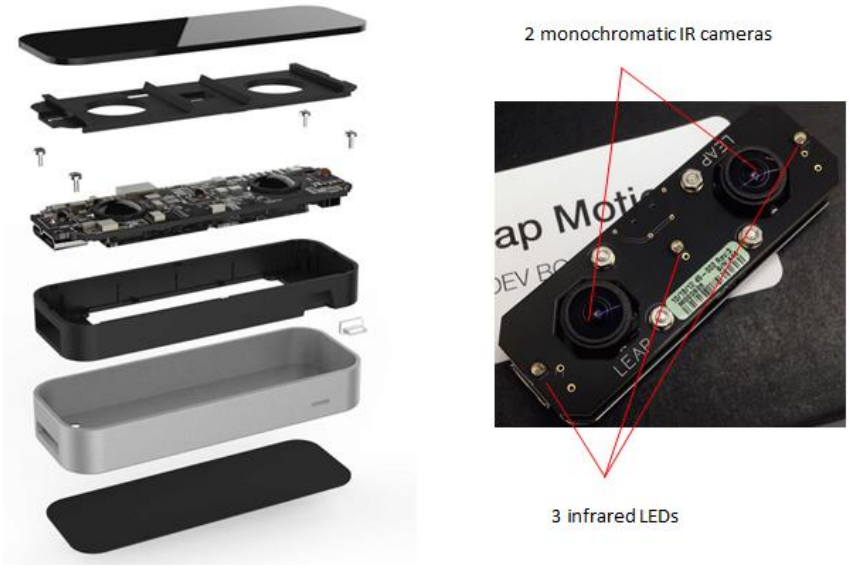


*Figura 3: Componentes hardware del controlador Leap Motion*

El software instalado en el ordenador que recibe los datos es el servicio Leap Motion *(Leap Motion Service)* y es el encargado de procesar los datos y las imágenes que recibe del controlador.

Lo primero que hace este servicio es aplicar sobre los datos, lo que los propios creadores de Leap denominan como "algoritmos matemáticos avanzados" (Colgan, Sitio web de Leap Motion, 2014 [2]). A continuación, realizan una compensación de los objetos de fondo capturados por las cámaras (como la cabeza del usuario) y de otros elementos como la iluminación ambiental para reconstruir una representación en 3D de lo que el dispositivo percibe.

Después de reconstruir esta representación en 3D, los algoritmos de rastreo del servicio analizan la información para extraer datos de rastreo de dedos y herramientas e interpretan la representación 3D de lo percibido por el controlador, para inferir las posiciones de los dedos y los objetos. A estos resultados se les aplica técnicas de filtrado para mantener la coherencia de los datos. Estas técnicas permiten *"suavizar"* la representación de los movimientos de las manos del usuario.

Posteriormente, el servicio Leap Motion prepara los resultados, expresándolos como una serie de *frames* que contienen todos los datos de rastreo, para enviarlos a través del protocolo de transporte.

El servicio utiliza este protocolo para comunicarse con el panel de control de Leap Motion *(Leap Motion Control Panel*) así como con las aplicaciones que utilizan las librerías nativas de Leap o las que utilizan las librerías del cliente web, a través de una conexión de *socket local* (TCP (Transmission Control Protocol) para aplicaciones nativas y *WebSocket* para aplicaciones web).

Finalmente, cuando la aplicación ha recibido los datos, la librería organiza los datos en una estructura de API (Application Programming Interface) orientada a objetos *(Figura 4)*, gestiona el historial de *frames* y proporciona una colección de clases y funciones para que utilicen los desarrolladores.

## 3.2 Precisión de Leap Motion

Guna (Guna et al. [10]) y Weichert (Weichert et al. [34]) han analizado Leap Motion desde el punto de vista del hardware con el fin de comprobar la precisión y fiabilidad de los datos que ofrece. Para medidas de puntos estáticos, en ambos trabajos se obtuvo un nivel de precisión alto, por debajo de los 0,2 mm. Para medidas de objetos dinámicos, Guna (Guna et al. [10]) detectaron que los datos de Leap presentan una importante dependencia espacial y que la desviación estándar de los datos se incrementa al alejarse del dispositivo, ya sea hacia arriba o hacia los lados. Weichert (Weichert et al. [34]) obtuvieron una precisión media de 0,7 mm en ese contexto dinámico.

Esta dependencia espacial detectada por Guna (Guna et al. [10]) afecta también al proyecto, ya que las posiciones de los punteros que se muestran están ligadas a las posiciones de los dedos que el servicio de Leap detecta y envía a la aplicación. Es por esto que cuanto más se aleja el usuario del dispositivo, ya sea hacia los límites laterales o hacia los límites superior-inferior, la precisión de la aplicación a la hora de representar el puntero disminuye notablemente.

## 3.3 Aplicaciones para Leap

Leap fue lanzado el 21 de mayo de 2012 y en octubre de ese mismo año se lanzó una primera versión del SDK para los desarrolladores. A pesar de llevar 4 años en el mercado, sigue siendo una tecnología en auge y es por eso que todavía no hay muchas empresas dedicadas al desarrollo de grandes aplicaciones para Leap Motion. Aun así, Leap cuenta con su propia tienda online de aplicaciones, la Leap Motion App Store (Leap Motion [16]), en la que los usuarios pueden comprar las aplicaciones que las empresas o incluso desarrolladores independientes han desarrollado.

Con el objetivo de promover el desarrollo de aplicaciones para la Leap Motion App Store, Leap ofrece un SDK, cuya segunda versión fue lanzada en mayo de 2014, que permite desarrollar aplicaciones para Leap con diferentes plataformas y lenguajes de programación.

El tipo de aplicaciones para Leap es muy heterogéneo. Uno de los campos en el que más interés pueda tener esta tecnología es el mundo de los videojuegos, para mejorar la experiencia del usuario, pero también existen aplicaciones pensadas para entornos más serios como hospitales, talleres... En los siguientes puntos, se presentan algunas de las aplicaciones para Leap.

### 3.3.1 Aplicaciones de la Leap Motion App Store

En la tienda de aplicaciones de Leap a día de hoy se pueden encontrar un total de 227 aplicaciones, entre las cuales algunas a destacar serían las siguientes:

- Form and Function 3D (Polley [29]), una aplicación educativa que permitiría mejorar la parte práctica de los estudiantes de asignaturas relacionadas con la biología y la medicina. Esta aplicación muestra un corazón y permite interactuar con él, con el objetivo de dar la sensación al usuario de que es casi real.

- Enmarcada en el mismo contexto que la aplicación anterior se encuentra Frog Dissection (Ematras [7]), una aplicación que permitiría a los estudiantes vivir la experiencia de diseccionar una rana sin la necesidad de utilizar una rana real.

- Además de aplicaciones también se pueden adquirir complementos o *plug-ins* para otras aplicaciones como es el caso del *plug-in* para el software de modelado 3D Maya (Autodesk [1]).

### 3.3.2 Otras aplicaciones

Aunque Leap Motion ofrece una tienda para que los desarrolladores pongan en venta sus aplicaciones, algunos prefieren ofrecerlas en sus propios sitios web, o incluso no dejarlas al alcance de cualquier usuario, sino desarrollarlas para clientes específicos. En los siguientes puntos se muestran algunas aplicaciones y proyectos e iniciativas de desarrolladores de Leap.

- Una iniciativa interesante es la que ha llevado acabo TedCas (TedCas [32]). Una *start-tup* centrada en la salud, que proporciona interfaces naturales de usuario que permiten acceder, en tiempo real y sin ningún tipo de contacto físico, a la información digital e imágenes de un paciente. TedCas ha desarrollado un sistema que permite utilizar el controlador Leap Motion para interactuar con los ordenadores de un modo muy sencillo.

- Otro proyecto interesante es el de InMoov (InMoov [11]), que permite controlar un brazo robótico con el dispositivo Leap Motion. El usuario realiza gestos y movimientos con las manos sobre el controlador Leap y el brazo robótico copia esos movimientos.

- En el blog oficial de Leap Motion se muestra una lista de aplicaciones que hacen uso del dispositivo Leap en uno de los posibles escenarios que debido a la importancia de la higiene de las manos enmarcaría este proyecto, el mundo de la medicina. En la entrada del blog (Colgan, Sitio web de Leap Motion, 2015, [3]), se listan 5 aplicaciones, proyectos e iniciativas que relacionan Leap con el sector de la medicina. Entre éstas se encuentran la iniciativa de TedCas ya comentada, algunas iniciativas como Hand Tremors (Smith, 2014 [30]) para el tratamiento de enfermedades que ocasionan temblores en las manos (como por ejemplo el Parkinson) y una iniciativa para el reconocimiento del lenguaje de los signos. En esta entrada del blog, también aparecen aplicaciones para realizar terapia de rehabilitación en las manos como es el caso de la aplicación Visual Touch Therapy de Ten Toy Ray Gun (Ten Toy Ray Gun [33]).

- Enmarcados en otro de los posibles escenarios del proyecto como sería el sector de la industria también existen proyectos que utilizan este dispositivo. En el apartado de

soluciones del portal de Leap Motion (Leap Motion [19]) se muestran algunos posibles usos de Leap en el sector de la industria, en el cual se puede utilizar Leap para controlar máquinas o robots, como en dos de las soluciones del portal de Leap en las cuales controlan un robot para desactivar bombas y un robot de la NASA por medio de este dispositivo. Otra de las soluciones que aparecen en el portal es la iniciativa llevada a cabo por Design&Systems (Design&Systems [4]) que permite explorar un proceso industrial utilizando el controlador Leap.

- En el sector de la industria también existen aplicaciones que integran el controlador Leap con el diseño de piezas y modelado 3D como es el caso del ya comentado *plug-in* para Maya desarrollado por Autodesk.

### 3.3.3 Aplicaciones similares

Existen diversas aplicaciones y proyectos orientados a la interacción con los ordenadores por medio del dispositivo Leap Motion. Sin embargo no todas comparten el objetivo de este proyecto, que es permitir al usuario realizar las mismas acciones que realizaría con un ratón y un teclado, utilizando únicamente el controlador Leap Motion.

Entre las aplicaciones que sí comparten el objetivo de este proyecto, algunas de ellas se pueden encontrar en la Leap Motion App Store y otras en los sitios web de los desarrolladores. A continuación, se muestran algunos de estos proyectos y aplicaciones junto con los principales aspectos en los que se diferencian de este proyecto.

- DexType (DexType [5]) es el proyecto que más se asemeja, sin duda alguna, a KVLeap. Ambos permiten realizar las mismas acciones que un usuario realizaría con un ratón por medio del dispositivo Leap, y ambos cuentan con un teclado virtual que se muestra en pantalla para la introducción de texto por medio de movimientos y gestos de las manos. DexType actualmente está disponible como extensión para el navegador Google Chrome. DexType también reconoce gestos que interpreta para llevar a cabo determinadas acciones y permite escribir dibujando números en el aire. La principal diferencia entre KVLeap y DexType es que KVLeap permite interactuar con el sistema operativo y todas las aplicaciones del mismo, mientras que DexType está limitado únicamente a Google Chrome.

- Minuun (Minuun [22]) muestra en su página web un teclado con la posibilidad de escribir sobre él mediante los movimientos de las manos. Este proyecto todavía no ha

sido desarrollado y no se ha podido determinar cuáles son las posibilidades del mismo, pero en el vídeo de la página web de Minuun muestran un teclado sobre el que se puede escribir de forma similar al de KVLeap.

- Varias aplicaciones que disponibles en la Leap Motion App Store, las cuales permiten controlar el ratón o determinadas aplicaciones o apartados del sistema (como el volumen o el brillo). Un ejemplo sería Pointable (Pointable [28]) una aplicación de escritorio para sistemas Windows que permite al usuario realizar todo lo que haría con un ratón. En el aspecto del control del ratón Pointable es muy similar a KVLeap, aunque no ofrece la funcionalidad del teclado.

## 3.4 Alternativas a Leap

Leap Motion no es el único dispositivo que permite interactuar con un ordenador por medio de los gestos, son ya varias las empresas que antes o después de Leap han decidido probar suerte en esta área que está todavía en vías de desarrollo. Grandes empresas como Microsoft han realizado sus puestas en este tipo de tecnologías mediante Kinect, aunque no sería justo comparar Kinect con Leap, ya que la primera permite rastrear las posiciones y movimientos de todo el cuerpo y la segunda se centra simplemente en las manos, y es por eso que Kinect no ofrece tantas posibilidades en cuanto al rastreo de los datos de las manos.

En este apartado se presentan algunas tecnologías que comparten el mismo objetivo que Leap.

Nimble VR (Nimble VR [25]), que aunque todavía es un prototipo, es sin duda la alternativa más parecida a Leap. Al igual que Leap es un dispositivo pequeño que rastrea las posiciones y movimientos de las manos. Además, los datos que rastrea son muy similares a los de Leap. En uno de sus vídeos oficiales (Nimble VR [25]), muestra una comparación entre los dos dispositivos.

Otra alternativa, aunque no tan similar a Leap, es Myo (Myo [23]), un brazalete que lee la actividad muscular del usuario para permitirle interactuar con un ordenador por medio de los gestos de su brazo y sus manos.

# Capítulo 4

# Tecnologías utilizadas

Para poder llevar a cabo este proyecto ha sido necesario adquirir e integrar diferentes tecnologías que en mayor o menor medida han ayudado a desarrollarlo. Algunas de estas tecnologías, como WPF (Windows Presentation Foundation) o el SDK de Leap, han jugado un papel determinante en el proyecto y sin ellas el proyecto hubiera sido muy diferente, o directamente no hubiera sido posible. Sin embargo, otras han jugado un papel menos crucial y podrían ser fácilmente reemplazadas por otras tecnologías, como es el caso de la base de datos SQLite.

Este capítulo presenta las diferentes tecnologías que conforman el proyecto y el papel que juegan dentro de éste, así como las alternativas consideradas, y los factores que han determinado la elección de la tecnología.

## 4.1 SDK de Leap Motion

El SDK de Leap Motion es el kit de desarrollo que permite a los desarrolladores utilizar los datos que genera el software de Leap Motion ya mencionado en el capítulo 3. Este kit para desarrolladores ofrece una API *(Figura 4)* para facilitar el acceso a los datos generados por el software de Leap.

La versión actual del SDK de Leap es la 2 (concretamente la 2.2.6). Cuando se comenzó a desarrollar este proyecto, se comenzó a utilizar la última versión disponible del SDK que era la 2.1.2. Esta versión todavía estaba en fase beta, pero a medida que el proyecto fue avanzando se pasó a la versión 2.2.2 que ya era una versión estable y con la última actualización del SKD finalmente se pasó a la versión 2.2.6. Las diferencias que hay entre estas versiones principalmente son soluciones de problemas y cuestiones de optimización en algunos aspectos como el uso de CPU (Central Processing Unit) para el procesamiento de los *frames*.

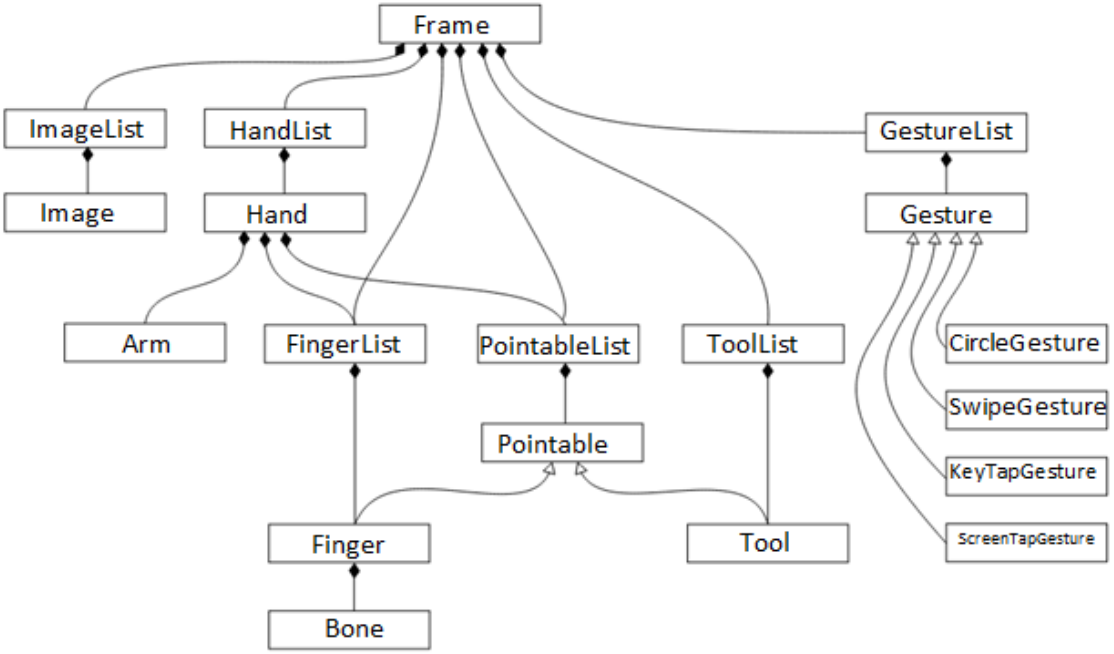


*Figura 4: Leap Motion API: Tracking Model*

## 4.2 WPF

El objetivo de este proyecto es obtener como producto una aplicación de escritorio que los usuarios podrán utilizar para interactuar con su ordenador mediante gestos que reconoce el controlador Leap. Actualmente, la familia de sistemas operativos más utilizada es Windows, y es por esto que la aplicación ha sido desarrollada como una aplicación de escritorio para los sistemas operativos Windows.

WPF es una tecnología de Microsoft que permite el desarrollo de aplicaciones e interfaces de interacción para los sistemas Windows. WPF ofrece una amplia infraestructura y potencia gráfica con la que es posible desarrollar aplicaciones visualmente atractivas. Esta tecnología separa mediante el lenguaje declarativo XAML (eXtensible Application Markup Language) y los leguajes de programación del *framework* .NET, la interfaz de interacción de la lógica del negocio, proporcionando a las aplicaciones una arquitectura MVC (Modelo Vista Controlador).

## Alternativas a WPF

Entre las múltiples tecnologías que permiten desarrollar aplicaciones de escritorio, este proyecto está limitado a utilizar únicamente aquellas soportadas por el SDK de Leap. Entre estas tecnologías, la primera alternativa considerada fue desarrollar la aplicación con Java, de esta forma sería compatible con la mayoría de los sistemas operativos. Sin embargo, se prefirió utilizar WPF, porque permitía crear interfaces más atractivas para el usuario.

La segunda alternativa considerada fue utilizar el motor de videojuegos Unity. Las posibilidades que ofrece Unity con Leap son enormes, y permite crear aplicaciones en 3D mucho más visuales. En las fases tempranas del proyecto, Unity era la tecnología escogida para realizar el proyecto. Con esta tecnología el proyecto hubiera sido completamente diferente, ya que el objetivo era crear un modelo de un teclado en 3D e integrarlo con el *HandControler* de los a*ssets* de Leap para escribir.

A pesar de que la aplicación visualmente sería mucho más atractiva, las posibilidades de la misma quedarían mucho más reducidas, ya que no permitirían mostrar el teclado y las manos directamente sobre otras aplicaciones como Word, Chrome... Para cumplir con el objetivo, se podrían utilizar dos pantallas, una que mostrase el teclado y la recreación de las manos del usuario en 3D y otra que mostrase el sistema y las aplicaciones del usuario. La aplicación de Unity captaría los gestos y movimiento y generaría los eventos correspondientes a pulsar teclas, hacer clic... pero aun así, el proceso de escritura seguía siendo más complicado.

Por último antes de descartar esta opción se les presentó a un grupo reducido de 12 médicos MIR (Médico Interno Residente) del Hospital de Donostia, 2 diferentes enfoques para llevar a cabo el objetivo, el enfoque de DexType y Minuun y el enfoque de Unity. Todos ellos coincidieron en que les parecía más fácil y rápido a la hora de escribir el enfoque de Minuun, y finalmente, se optó por darle este enfoque a la parte del teclado.

Una vez determinado el enfoque y descartadas las opciones de Java y Unity, solo quedaba elegir cuál de las tecnologías de .NET iba a ser la que se utilizaría para desarrollar el proyecto. Las alternativas eran Windows Presentation Foundation o Windows Forms. Ante la inexperiencia en cualquiera de las dos opciones y la indiferencia de cual utilizar, ya que ambas permitirían desarrollar la aplicación que se tenía en mente, finalmente se decidió utilizar WPF, porque por un lado, el tipo de ventanas con las que se quería trabajar eran más fáciles de conseguir con WPF y por otro lado, por ser una

tecnología más novedosa que está integrada completamente con el Modelo Vista Controlador de .NET.

## 4.3 SQLite

KVLeap cuenta con un diccionario de palabras para poder predecir palabras en función de las teclas pulsadas por el usuario. Para almacenar las palabras que componen este diccionario, utiliza una base de datos SQLite. KVLeap utiliza la base de datos también para almacenar la configuración establecida por los usuarios y poder volver a cargarla cuando se vuelva a ejecutar la aplicación.

SQLite es un sistema de gestión de bases de datos relacional contenido en una librería escrita en C. La biblioteca de SQLite se enlaza con el programa que la utiliza, pasando a ser parte del mismo, lo que permite al programa funcionar de forma independiente sin la necesidad de otros programas o servicios. SQLite no requiere ninguna instalación ni configuración de la base de datos.

### Alternativas SQLite

El modelo de datos de la aplicación es un modelo de datos sencillo, que no añade ninguna restricción a la hora de elegir un SGBD (Sistema Gestor de Bases de Datos). Sin embargo, sí lo hacen las características de la aplicación. La aplicación está pensada para funcionar de forma independiente, sin conexión a Internet y sin requerir la instalación previa de otros programas, por lo que la base de datos a utilizar tiene que poder ser incrustada en la aplicación para funcionar correctamente sin conexiones a Internet ni instalaciones previas de sistemas gestores de bases de datos. Este requisito establece una restricción que limita las opciones en cuanto al SGBD a utilizar.

La primera alternativa que se considero fue la de no usar un SGBD. Con esta alternativa se usaría un diccionario de datos, o servicio de predicción de palabras de una aplicación o de un servicio externo a la aplicación desarrollada, y la configuración se almacenaría en un fichero de configuración. No obstante, como uno de los objetivos era que la aplicación funcionase sin dependencias de otras aplicaciones, se decidió utilizar una base de datos que pudiese integrarse completamente en el proyecto sin necesidad de

otras instalaciones, para almacenar el diccionario y se aprovechó también para guardar la configuración de la aplicación.

Una vez decidido el uso de la base de datos, se eligió el sistema gestor de bases de datos que utilizaría la aplicación. Entre los diferentes sistemas gestores que existían en el mercado, solo se consideraron aquellos que pudiesen hacer que la aplicación fuese independiente del software instalado en el ordenador del cliente. Entre estas alternativas, además de SQLite se consideró también la posibilidad de utilizar Microsoft SQL Compact.

SQL Compact al igual que SQLite es un sistema de gestión de bases de datos que se ejecuta bajo el proceso de la aplicación que la consume, permitiendo a la aplicación utilizar una base de datos sin la necesidad de tener instalado software adicional, ni establecer conexiones con un servidor a través de Internet.

Para una base de datos tan sencilla como la que se utilizará en este proyecto, cualquiera de las dos opciones consideradas era válida. Sin embargo, se optó por utilizar SQLite, ya que es compatible con un mayor número de sistemas operativos, y a pesar de que la aplicación está hecha para Windows, en caso de que en un futuro se quisiera extender a otros sistemas, la base de datos será compatible con éstos.

## 4.4 Visual Studio

El entorno de desarrollo utilizado para la implementación del código de la aplicación ha sido Visual Studio, concretamente la versión 2013 profesional. Visual Studio es un entorno de desarrollo para sistemas operativos Windows que permite crear diferentes tipos de aplicaciones, ya sean aplicaciones para *smartphones*, aplicaciones o servicios web, o aplicaciones de escritorio, utilizando las herramientas del *framework* .NET.

Visual Studio además soporta varios lenguajes para el desarrollo de las aplicaciones, tales como c#, c++ y Visual Basic, pero además, soporta otros lenguajes como Java, PHP, Python... mediante extensiones que se pueden añadir al entorno de desarrollo.

## Alternativas Visual Studio

Visual Studio no es el único IDE (Integrated Development Environment) que permite trabajar con las tecnologías de .NET. Otros como MonoDevelop o SharpDevelop, ofrecen características muy similares y además son *open source*, por lo que no sería necesario pagar por la licencia de Visual Studio. No obstante como ya se había utilizado Visual Studio, y como se disponía de una licencia (fruto del acuerdo entre Microsoft y la UPV/EHU) se decidió utilizar este entorno de desarrollo sin considerar ninguna otra alternativa.

No obstante, durante el periodo formativo de este proyecto, se realizaron algunas pruebas con Unity, que por defecto venía con el entorno de desarrollo MonoDevelop. Por esta razón durante algunas pruebas se utilizó este IDE, pero rápidamente se cambió a Visual Studio, porque el *feedback* ofrecido para la corrección de errores era mejor, y porque se tenía más experiencia con éste.

En cuanto al lenguaje de programación a utilizar, se decidió utilizar c# por su gran parecido con Java que es el lenguaje con el que más experiencia se tenía, minimizando así los costes del aprendizaje de un nuevo lenguaje de programación.

# Capítulo 5

# Funcionalidades y características del sistema

En este capítulo se presentan las funcionalidades y características principales de la aplicación desarrollada en este proyecto.

## 5.1 Funcionalidades de KVLeap

Como ya se ha mencionado, el objetivo de este proyecto está principalmente orientado a ofrecer al usuario la posibilidad de realizar todas las acciones que puede realizar con un teclado y un ratón normal mediante los gestos y movimientos de sus manos que el controlador Leap Motion recoge. En este apartado se presentan las funcionalidades que ofrece KVLeap para cumplir con su objetivo. Por un lado se presenta la funcionalidad encargada de interpretar las posiciones y gestos de los dedos y las manos, y por otro lado, las funcionalidades que permiten realizar las acciones del teclado y el ratón.

### 5.1.1 Posiciones de los dedos

El controlador de Leap Motion rastrea las posiciones de las manos en tiempo real y envía los datos al software de Leap Motion que se encarga de interpretarlos y enviarlos a las aplicaciones que los soliciten. Entre esos datos se encuentran las posiciones de los dedos que KVLeap utilizará para representar los punteros de la aplicación. Sin embargo esos datos no pueden ser utilizados directamente como coordenadas de la pantalla de un ordenador, ya que son puntos pertenecientes a un espacio tridimensional

como lo es el área de interacción del controlador Leap. Para poder utilizar esos datos, KVLeap realiza un proceso de *mapping* que se encarga de transformar los puntos tridimensionales del área de interacción de Leap, al sistema de coordenadas de la pantalla de un ordenador.

### 5.1.2 Detección de gestos

Como ya se ha comentado el software de Leap Motion es capaz de reconocer gestos en los datos que recibe del controlador. Por defecto, reconoce una serie de gestos predefinidos, pero también permite a los desarrolladores definir nuevos gestos que luego el software de Leap Motion reconocerá. Cuando el software de Leap Motion reconoce un gesto, genera un objeto de la clase *Gesture* y lo añade al *frame* que contiene los otros datos que se les enviará a las aplicaciones que los soliciten.

Entre los diferentes gestos que el software de Leap Motion es capaz de reconocer por defecto, KVLeap utiliza los siguientes:

- Swipe gesture: Este gesto se genera cuando el usuario desliza la mano realizando un barrido como el que se muestra en la *Figura 5 gesto a),* en cualquiera de las direcciones (positiva y negativa), y en cualquiera de los dos ejes (horizontal y vertical).

- Circle gesture: Este gesto se genera cuando el usuario dibuja un círculo con alguno de sus dedos, tal y como se muestra en la *Figura 5 gesto b)*.

- Screen tap gesture: Este gesto se genera cuando el usuario mueve alguno de sus dedos hacia la pantalla, como si fuese a tocarla, como se puede observar en la *Figura 5 gesto c)*.

- Key tap gesture: Este gesto se genera cuando el usuario mueve alguno de sus dedos como si estuviese pulsando una tecla, tal y como muestra la *Figura 5 gesto d)*.

*Figura 5: Gestos KVLeap: a) Swipe gesture; b) Circle gesture; c) Screen tap gesture; d) Key tap gesture*

### 5.1.3 Funcionalidades relacionadas con el ratón

Las funcionalidades de KVLeap equivalentes a las ofrecidas por un ratón son las siguientes:

- Detectar y mover el cursor: El dispositivo Leap Motion permite detectar en tiempo real las posiciones de las manos y dedos del usuario y representarlos en pantalla. El usuario sitúa sus manos a una cierta distancia por encima del controlador Leap Motion, y la aplicación KVLeap muestra un puntero asociado a cada dedo índice que el controlador reconoce. Este puntero se mueve siguiendo los movimientos del dedo que tiene asociado. De los dos punteros que aparecen en pantalla, el puntero ligado al dedo que primero reconoce el controlador es el puntero principal, y el otro, el puntero secundario.

- Realizar diferentes tipos de clic en la posición del puntero principal: KVLeap ofrece la posibilidad de generar los eventos asociados a los principales tipos de clic que un ratón permite realizar, de tres formas distintas, para que el usuario elija la que más cómoda le parezca. Entre los tipos de clic que se pueden realizar tenemos los siguientes: clic izquierdo, clic derecho, doble clic izquierdo y clic y arrastrar. Para indicar el tipo de clic que se quiere realizar se utilizará la mano asociada al puntero principal tal y como se especifica en el apartado 5.1.4 *Tipos de clic* y para generar el evento que realiza el tipo de clic indicado, se realiza de alguna de las siguientes formas:

  - Realizando el gesto *Screen Tap Gesture (Figura 5 gesto c))*: Si el usuario realiza este gesto con el dedo índice de la mano asociada al puntero secundario, se generará el evento correspondiente a realizar el tipo de clic indicado por la mano asociada al puntero principal.

- Realizando el gesto *Key Tap Gesture (Figura 5 gesto d))*: Cuando el usuario realiza el gesto de pulsar una tecla con el dedo índice de la mano asociada al puntero secundario, se comprueba el tipo de clic indicado por la mano asociada al puntero principal, y se genera evento correspondiente a realizar ese tipo de clic.
- Dejando el puntero fijo durante un tiempo: Esta funcionalidad permite realizar clic con el puntero utilizando únicamente la mano asociada al puntero principal. Para hacer clic, el usuario indicará que quiere hacer clic extendiendo el pulgar de esa mano, y el tipo de clic con el resto de los dedos. Finalmente, posicionará el cursor sobre el elemento que quiere hacer clic durante un breve periodo de tiempo.

- Girar la rueda de un ratón: Cuando el usuario realiza un círculo con alguno de los dedos asociados a los punteros, la aplicación detecta el gesto *Circle gesture (Figura 5 gesto b))*. Si el círculo realizado es en el sentido de las agujas del reloj, la aplicación produce el evento correspondiente a girar la rueda del ratón hacia adelante. Si por el contrario el círculo es en el sentido contrario a las agujas del reloj, la aplicación genera el evento correspondiente a girar la rueda del ratón hacia atrás.

Con el objetivo de mejorar la funcionalidad de hacer clic, el usuario puede activar o desactivar el modo de clic preciso desde la ventana de configuración. Con este modo activado, cuando el usuario va a realizar clic, el movimiento del puntero queda reducido a un área cercano a la posición del puntero, mejorando así la precisión del usuario para hacer clic sobre un elemento dado.

### 5.1.4 Tipos de clic

KVLeap permite realizar diversos tipos de clic. Para poder hacer clic es preciso indicar cuál es el tipo de clic que el usuario quiere hacer. Para indicar el tipo de clic independientemente de la opción activa para hacer clic (ya sea mediante algún gesto o en función del tiempo) siempre se realizará de la misma manera: Primero el usuario indicará con la mano asociada al puntero principal el tipo de clic que quiere realizar, y luego con realizará el clic de alguna de las tres maneras indicadas en el apartado 5.1.3 *Funcionalidades relacionadas con el ratón*.

El tipo de clic a realizar varía en función de los dedos extendidos de la mano asociada al puntero principal:

- Si el dedo índice está extendido, el tipo de clic es el equivalente a realizar un clic con el botón izquierdo del ratón.
- Si además del dedo índice el dedo medio está extendido, el tipo de clic a realizar es el equivalente al que realiza el botón derecho del ratón.
- Sí además de estos dos dedos el dedo anular se encuentra extendido, entonces el tipo de clic será el equivalente a realizar un doble clic con el botón izquierdo del ratón.
- Por último si además de estos tres dedos el dedo pequeño se encuentra extendido, entonces el tipo de clic que se realizará, será el equivalente a hacer clic y arrastrar el ratón. Para soltar después de arrastrar, habrá que realizar otro clic de este tipo.

En la *Figura 6* se puede ver de forma gráfica cada tipo de clic de la aplicación.

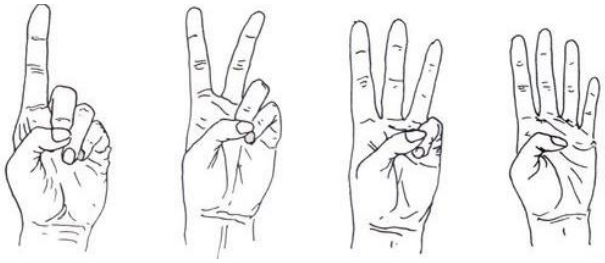

*Figura 6: Left click; right click; double left click; click and drag.*

## 5.1.5 Funcionalidad relacionada con el teclado

KVLeap dispone de un teclado virtual que permite al usuario realizar las funciones básicas que realizaría con un teclado normal. El teclado virtual aparecerá en pantalla si el usuario desliza la palma de la mano con la mano abierta desde abajo hacia arriba *(Swipe gesture (Figura 5 gesto a)))*. Una vez que el teclado virtual esté visible, para poder escribir el usuario tendrá que deslizar el puntero principal sobre las teclas del teclado virtual. Al terminar, con un movimiento de la mano de arriba hacia abajo, la aplicación ocultará el teclado.

Con el objetivo de acelerar el proceso de escritura, se ha añadido en la aplicación la funcionalidad de predecir texto. El usuario puede activar/desactivar la predicción de texto cuando lo considere oportuno. Cuando está activada, aparece una barra en la parte superior de la pantalla sobre la que se van mostrando las palabras sugeridas por la aplicación en función de las teclas que el usuario ha pulsado (excepto la última palabra que muestra los caracteres que el usuario ha pulsado). Si el usuario mueve la mano de izquierda a derecha, la aplicación detecta el gesto *Swipe gesture* *(Figura 5 gesto a))* y actualiza la lista de sugerencias para mostrar nuevas palabras. Si el gesto es en dirección contraria, la aplicación vacía la lista de sugerencias.

## 5.2 Características de KVLeap

En este apartado se presentan las características principales de la aplicación así como algunos de los detalles que aportan algún tipo de valor al proyecto.

### 5.2.1 Ventana principal

La ventana principal de KVLeap es la que se puede ver en la *Figura 7*. Esta ventana tiene la propiedad *TopMos*t, que hace que al ejecutarse se ponga por encima de cualquier otra ventana en ejecución, para poder mostrar tanto los punteros como el teclado sobre otras aplicaciones. KVLeap no es la única aplicación que usa esta propiedad, por lo que al usar los punteros para lanzar otra aplicación si ésta es *TopMost* KVLeap pasaría a estar en un segundo plano y no se verían los punteros ni el teclado. Para evitar que esto ocurra, cuando se realiza un clic desde la aplicación, ésta se encarga de ponerse otra vez en primer plano. Un caso concreto de esta propiedad se puede ver cuando se muestra un menú *(Figura 8)*, al mostrar el menú contextual la aplicación quedaría en segundo plano y no se vería el puntero, sin embargo, de esta forma el puntero seguirá estando sobre esos menús.

La ventana principal además está siempre maximizada y tiene un fondo transparente que permite ver todo lo que hay detrás de ésta para poder interactuar con ello. Además, se ha omitido el marco de la ventana para maximizar el área de interacción. Para

poder cerrar la ventana sin necesidad de hacerlo desde la barra de tareas, se ha añadido un botón *(Figura 7 cuadro 1)*, para cerrar la aplicación, que se muestra cuando el teclado está oculto.

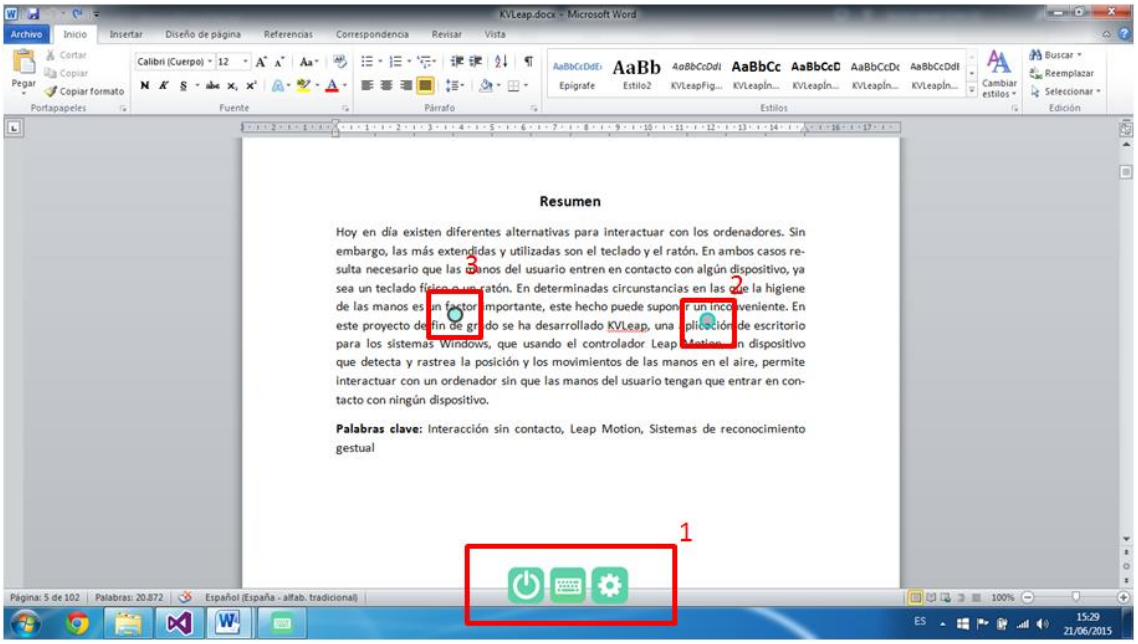


*Figura 7: KVLeap ventana principal*


*Figura 8: KVLeap propiedad Topmost: a) Sobre el menú de inicio; b) Sobre el menú contextual*

## 5.2.2 Punteros

KVLeap cuenta con dos punteros que juegan el papel del cursor del ratón. Estos punteros están asociados al dedo índice de cada mano del usuario y siguen los movimientos

de éstas. De estos dos punteros el que está asociado al dedo índice que primero reconoce Leap es el puntero principal *(Figura 7 cuadro 2)*, y el otro el secundario *(Figura 7 cuadro 3)*. Además, los punteros aparecen y desaparecen en función del número de dedos índices que ha reconocido el controlador.

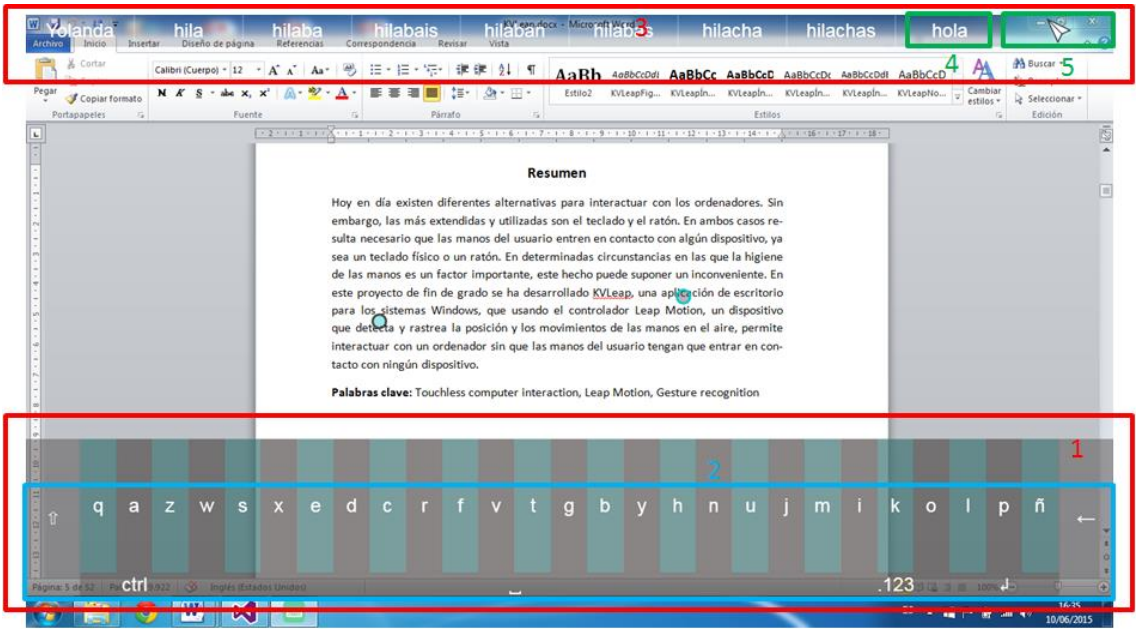


*Figura 9: KVLeap en ejecución*

### 5.2.3 Teclado

KVLeap cuenta con un teclado virtual *(Figura 9 cuadro 1)* que se puede mostrar/ocultar mediante el gesto *Swipe gesture* en el eje vertical. Este teclado virtual tiene tres juegos de caracteres que se pueden alternar pulsando la tecla de alternar teclado. Además, también permite alternar entre mayúsculas y minúsculas y enviar algunos comandos simples.

Cuando el puntero está sobre el área de colisión del teclado virtual *(Figura 9 cuadro 2)*, éste detecta una colisión y envía a la aplicación activa la tecla pulsada. El área de colisión de una tecla está definido en el 66% de la altura de la tecla, una vez que el puntero entra en este área se congela el movimiento del mismo en el eje horizontal para evitar que éste se mueva de forma involuntaria a las teclas adyacentes. Cuando el puntero abandona este área es cuando se escribe la letra asociada a esa tecla, salvo que el puntero haya bajado hasta algunas de las teclas de la parte inferior del teclado, en ese caso se generará el evento asociado a pulsar esa tecla.

### 5.2.4 Texto predictivo

La escritura con el teclado virtual puede llegar a resultar un poco lenta. Es por eso que KVLeap cuenta con la posibilidad de mostrar/ocultar una barra *(Figura 9 cuadro 3)* con palabras que la aplicación predice en función de las teclas pulsadas utilizando un algoritmo similar al algoritmo de texto predictivo T9. La barra puede llegar a mostrar hasta un total de ocho palabras predichas por el predictor de texto, y una novena palabra *(Figura 9 cuadro 4)* que muestra las teclas que el usuario ha pulsado, por si la palabra no estuviese en el diccionario.

Para escribir una palabra el usuario deberá pasar con el puntero por encima de la palabra que quiere escribir y esta quedará seleccionada (se pone el fondo más claro para que se diferencie del resto) para que cuando el usuario pase con el puntero por el botón de enviar *(Figura 9 cuadro 5)* se escriba esta palabra en la aplicación que esté activa. Cuando se escribe una palabra, si ésta no estaba en la base de datos, queda añadida, y si ya estaba, entonces se actualiza su contador para darle mayor prioridad en las próximas predicciones.

Si el usuario quiere escribir una palabra y no se muestra entre las ocho que predice el predictor de texto, puede mostrar nuevas palabras realizando el gesto *Swipe Gesture* en el eje horizontal y con dirección positiva. Si por el contrario la dirección del gesto es negativa, en lugar de mostrar nuevas palabras borrará la palabra que se está escribiendo y se vaciará la lista de palabras.

### 5.2.5 Diccionario

Para poder predecir las palabras que el usuario está escribiendo, KVLeap cuenta con un diccionario que consta de 885.418 palabras y que va incrementando a medida que el usuario escribe nuevas palabras.

Este diccionario de palabras se obtuvo a partir del diccionario que tiene el corrector ortográfico GNU Aspell (versión española) mediante el siguiente comando:

```
aspell –l es dump master | aspell –l es expand > words.txt.
```

Una vez obtenida la lista de palabras se introdujeron en la base de datos de la aplicación.

### 5.2.6 Optimización en el módulo de acceso a datos

El algoritmo de predicción de texto predice el texto en base al código que tienen asociadas cada una de las letras del teclado. Debido al elevado número de palabras que contiene el diccionario, el número de palabras que recogen las consultas a la base de datos para las palabras que coinciden con ese código también es bastante elevado y el tiempo en el que se realiza la consulta y en el que se recogen los datos afectaba al rendimiento de la aplicación, haciendo que los punteros dejasen de moverse (mientras se hacía la consulta), y dando la sensación de que ésta se paraba.

Para evitar que los punteros se dejasen de mover debido a que la aplicación está interactuando con la base de datos, se hizo que las llamadas a la base de datos para la obtención y actualización de las palabras del diccionario se realizasen en un *thread* aparte, de modo que el usuario podría seguir usando la aplicación mientras esta carga las palabras.

Por último, se limitó el número de palabras que devolvía la consulta como resultado a 32 para evitar tener largas listas de cadenas cargadas en memoria. De esta forma si el usuario quiere mostrar nuevas palabras a las predichas por el predictor realiza el gesto correspondiente y se le mostrarían 8 nuevas palabras hasta tres veces consecutivas sin volver a consultar la base de datos, a la cuarta petición la aplicación volvería a realizar una consulta a la base de datos para obtener otras 32 palabras.

Además de las mejoras mencionadas para acceder a los datos de forma más rápida y transparente al usuario, también se aplicó una mejora para la reducción del número de palabras almacenadas en la base de datos. La mejora se aplica a la hora de introducir nuevas palabras en la base de datos. Si la palabra escrita por el usuario no está en la base de datos, ésta se añade, pero solo si no existe una variante de la misma palabra (considerando como variante la misma palabra con variaciones de minúsculas y mayúsculas en sus caracteres) en la base de datos. Para poder escribir variantes de una palabra que ya está almacenada en un formato, KVLeap va aplicándole una máscara a las palabras obtenidas de la base de datos para que se corresponda con el formato (de minúsculas y mayúsculas) establecido por el usuario.

### 5.2.7 Ventana de configuración

KVLeap cuenta con una ventana de configuración *(Figura 10)* que permite a los usuarios configurar la aplicación y personalizarla como más se ajuste a sus gustos y preferencias personales. Esta configuración se guarda en la base de datos al cerrar la ventana y se carga cuando la aplicación se inicia.

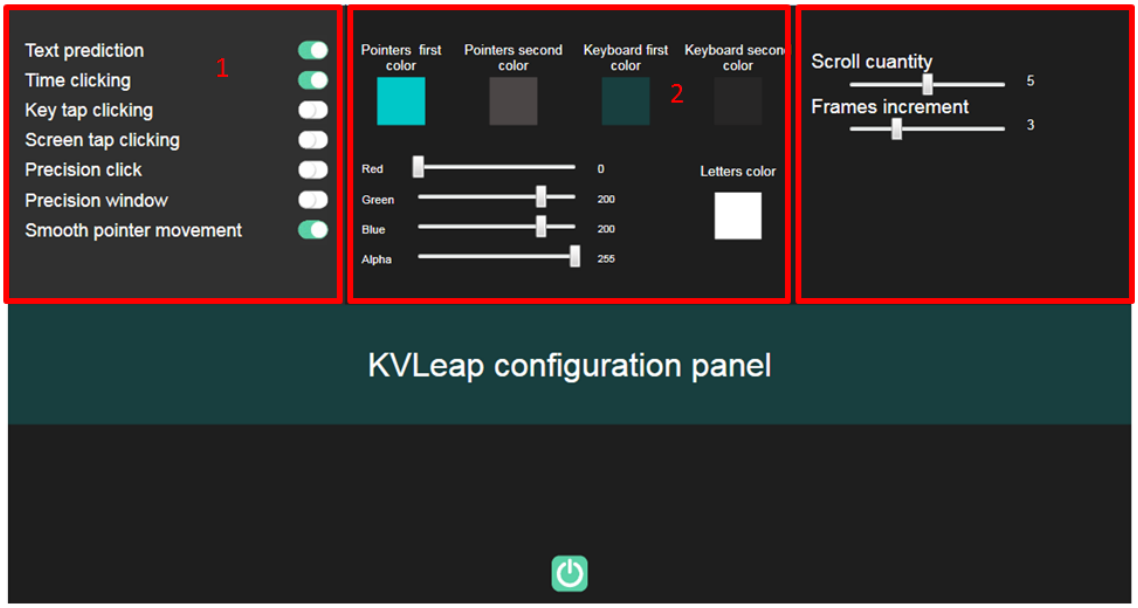


*Figura 10: KVLeap ventana de configuración*

Esta ventana está dividida en tres bloques, el primero de ellos *(Figura 10 cuadro 1)* es el bloque de opciones que permite activar y desactivar determinadas opciones para mejorar la interacción con el usuario; el segundo de ellos *(Figura 10 cuadro 2)* es el bloque de los colores, que permite al usuario personalizar algunos de los colores que se utilizan en la aplicación; y el tercer bloque *(Figura 10 cuadro 3)* contiene algunos parámetros relacionados con la funcionalidad del ratón ofrecida por KVLeap. A continuación, se describen las diferentes funciones y parámetros de estos bloques.

#### 5.2.7.1 Opciones

Las opciones del panel de configuración están orientadas a activar/desactivar funcionalidades de la aplicación para mejorar y facilitar la interacción con el usuario, estas opciones son las siguientes:

- Text prediction: Activa/desactiva la predicción de texto para la aplicación.
- Time clicking: Activa/desactiva la funcionalidad que permite realizar clic en función del tiempo que pasa el puntero sobre un área de la pantalla.
- Key tap clicking: Activa/desactiva la funcionalidad que permite realizar clic mediante el gesto *Key Tap Gesture*.
- Screen tap clicking: Activa/desactiva la funcionalidad que permite realizar clic mediante el gesto *Screen Tap Gesture.*
- Precisión click: Activa/desactiva el modo de precisión que reduce el movimiento del puntero en un área cercana a donde se quiere hacer clic para que la aplicación sea más precisa a la hora de hacer clic.
- Precisión window: Muestra/oculta una pequeña ventana *(Figura 11 cuadro 1)* que muestra un área cercana al cursor en tamaño aumentado.
- Smooth pointer movement: "Suaviza" el movimiento del puntero en la pantalla escalando a la pantalla directamente las posiciones de las manos que ha rastreado el dispositivo en su área de rastreo, logrando que la interacción con la aplicación sea mucho más agradable para los usuarios.

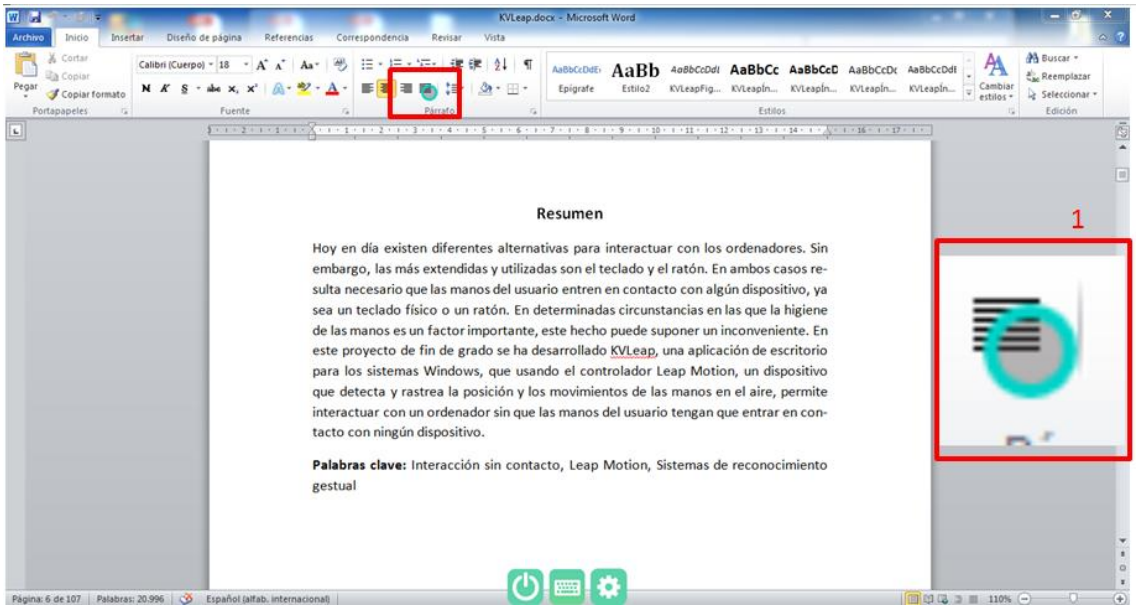


*Figura 11: Precision window*

### 5.2.7.2 Colores

Los colores utilizados en la aplicación son claros y se les ha dado un toque de transparencia para que tanto el teclado como los punteros dejen ver lo que está por detrás de ellos en segundo plano. Este efecto hace que sobre algunas aplicaciones no se distingan bien los elementos ya sea por el toque transparente que se les ha dado a los colores de la aplicación o porque los colores de la aplicación que está en segundo plano y los de KVLeap son muy similares. Por esta razón, más que por ofrecer al usuario la posibilidad de personalizar a su gusto los colores de la aplicación, se decidió darle la posibilidad de cambiar los colores de algunos de los elementos de KVLeap para que se vean bien sobre diferentes aplicaciones. Los colores que se pueden cambiar son los siguientes:

- Pointers first color: El color del borde del puntero principal que es el mismo que el del relleno del puntero secundario, salvo por la diferencia de que al relleno del puntero secundario se le da un toque de transparencia.
- Pointers second color: Al igual que el anterior, pero a la inversa el color del borde del puntero secundario y el de relleno del principal.
- Keyboard first color: El color de algunas de las teclas del teclado.
- Keyboard second color: El color de las teclas a las que no se les aplica el color de *Keyboard first color*.
- Letters color: El color de las letras de las teclas del teclado.

### 5.2.7.3 Parámetros

Por último se decidió dar la posibilidad al usuario de modificar algunos parámetros relacionados con la funcionalidad del ratón:

- Scroll: El valor asociado al movimiento de la ruleta del ratón.
- Frames increment: Una de las diferentes formas que ofrece la aplicación para hacer clic es en función del tiempo. Mediante este parámetro el usuario puede establecer el tiempo necesario para hacer clic. La aplicación hace clic cuando consigue más de 100 *frames* en un área reducida de la pantalla. Este parámetro establece el incremento de *frames* que se van sumando cada vez que se recibe un *frame* del controlador dentro de esa área. Por ejemplo si el valor es 4 cada

*frame* recibido dentro del área establecida se incrementa un contador en 4 y al llegar a 100 se genera un clic.

## 5.3 Características sitio web

Con el objetico de dar visibilidad al proyecto, se ha desarrollado un sitio web *(http://kvleap.es/KVLeap)* al que los usuarios podrán acceder para descargar la aplicación y en el que podrán ver información relacionada con ésta (vídeos, un manual de usuario, imágenes...). El sitio web no es uno de los elementos clave del proyecto, sin embargo, se ha desarrollado dentro de este proyecto y con el objetivo de aportar valor al mismo, por lo que se ha decidido dedicarle un apartado para comentar sus características más destacables.

### 5.3.1 Frameworks y tecnologías utilizadas

En este apartado se presentan los diferentes *frameworks* y tecnologías utilizadas para desarrollar el sitio web. La elección de estos *frameworks* y tecnologías estaba prefijada antes de comenzar el proyecto, ya que para el desarrollo del sitio web se partió de otra aplicación web desarrollada como práctica de la asignatura de Sistemas de Gestión de la Seguridad en los Sistemas de Información.

- Symfony: Symfony es un *framework* PHP que facilita la creación de proyectos web siguiendo la arquitectura MVC. Symfony fue diseñado para optimizar el desarrollo de aplicaciones web, proporcionando herramientas para agilizar aplicaciones complejas y guiando al desarrollador a acostumbrarse al orden y buenas prácticas dentro del proyecto.

- Bootstrap: Bootstrap, es un *framework* de software libre para el diseño de aplicaciones web *responsive.*

- Leapstrap: Leapstrap es el primer *framework* para el desarrollo de aplicaciones web integradas con Leap. Leapstrap está basado en Bootstrap, al que le añade

las funcionalidades necesarias para otorgar a un sitio web la capacidad de interactuar con un dispositivo Leap.

- SDK de Leap: Tanto el sitio web como Leapstrap utilizan la versión para Javascript del SDK de Leap Motion.

### 5.3.2 Secciones del sitio web

El sitio web es un sitio web sencillo que simplemente ofrece información sobre la aplicación desarrollada y el enlace para descargarla. Esta información está estructurada en los siguientes apartados:

- KVLeap: Página principal del sitio web, contiene información e imágenes de la aplicación.
- Proyecto: El objetivo del proyecto y sus características.
- Autor: Información sobre el autor del proyecto.
- Tecnologías: Una presentación de las tecnologías utilizadas para desarrollar el proyecto.
- Aplicación: Vídeo que presenta el proyecto y las posibilidades del mismo, información sobre la aplicación y enlaces de descarga para la aplicación y las instrucciones.
- Enlaces: Algunos enlaces de interés.
- Leap Controler: Indicador de si el sitio web detecta el controlador Leap Motion.

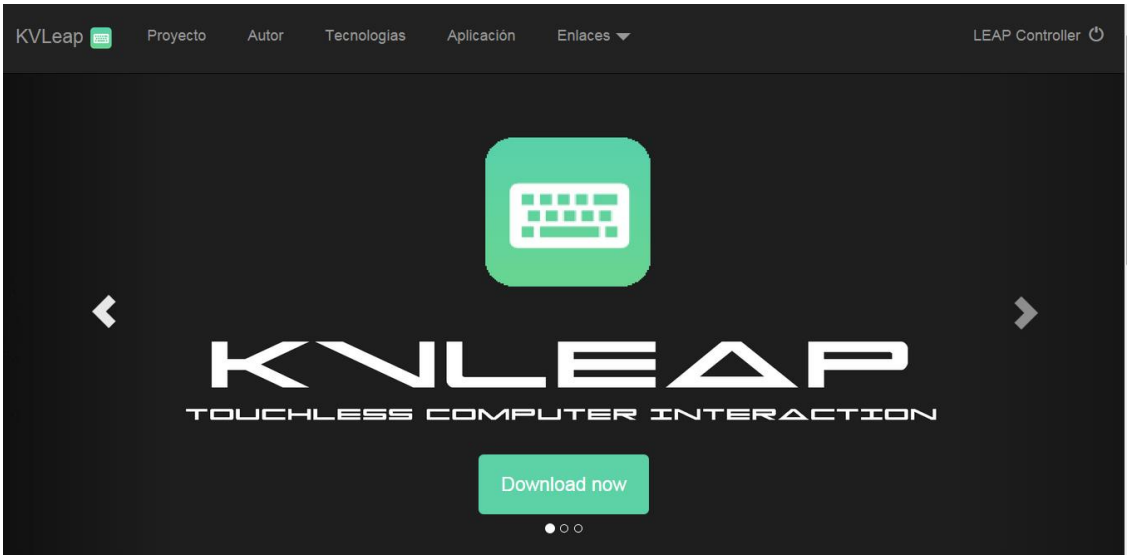


*Figura 12: Sitio web del proyecto*

# Capítulo 6

# Arquitectura del sistema

En este apartado se presenta la arquitectura del sistema desarrollado junto con los diferentes módulos que lo componen, con el objetivo de hacer que sea más fácil la comprensión de cómo funciona el mismo.

## 6.1 Arquitectura KVLeap

La arquitectura del sistema KVLeap está establecida por el patrón MVC *(Figura 13)*. Éste es un patrón de arquitectura software que separa los datos, de la interfaz de usuario y del controlador.

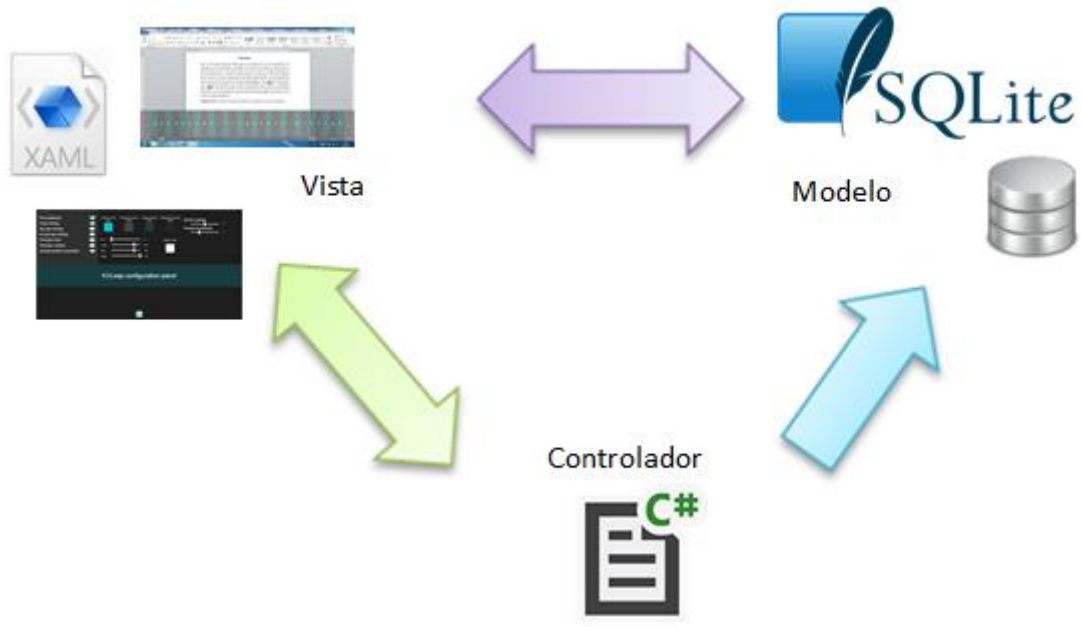


*Figura 13: Patrón MVC de KVLeap*

### 6.1.1 Modelo

El modelo es la representación de la información con la que el sistema opera. En KVLeap el modelo es muy sencillo, ya que la mayor parte de la funcionalidad recae sobre la vista y el controlador, y es por eso que para representarlo se utilizarán directamente las tablas de la base de datos:

La tabla Dictionary *(Tabla 1)* que contiene las palabras que forman el diccionario, junto con el código asociado a éstas y un contador que indica el número de veces que se ha escrito esa palabra y que permite dar prioridad a ciertas palabras sobre otras.

| **Dictinoary** | | |
|---|---|---|
| WORD (VARCHAR) | CODE (VARCHAR) | COUNT (INT) |

*Tabla 1: Tabla Dictionary*

La tabla Colors *(Tabla 2)* almacena los colores configurables de la aplicación. Para cada color se almacena el nombre del color junto con sus valores A (*alpha*), R (*red)*, G (*green*) y B (*blue*).

| **Colors** | | | | |
|---|---|---|---|---|
| COLOR (VARCHAR) | A (INT) | R (INT) | G (INT) | B (INT) |

*Tabla 2: Tabla Colors*

La tabla Parameters *(Tabla 3)* almacena los parámetros de configuración junto con sus valores.

| **Parameters** | |
|---|---|
| PARAMETER (VARCHAR) | VALUE (INT) |

*Tabla 3: Tabla Parameters*

Y por último, la tabla Options *(Tabla 4)*, que almacena las opciones configurables de KVLeap, junto con un valor que indica si está activada (1) o si está desactivada (0).

| **Options** | |
|---|---|
| OPTION (VARCHAR) | ACTIVATED (INT) |

*Tabla 4: Tabla Options*

### 6.1.2 Vista

La vista en KVLeap englobaría todo lo referente a las interfaces gráficas que se le presentan al usuario y cómo estás son tratadas durante la aplicación. KVLeap cuenta con numerosos elementos de interfaz de usuario que se van creado y modificando dinámicamente. Algunos ejemplos de estos elementos que componen la vista serían los elipses que se muestran en pantalla representando los punteros de la aplicación, las ventanas que se utilizan, los paneles que representan el teclado...

WPF permite diferenciar claramente la vista de los otros componentes del patrón MVC mediante el lenguaje declarativo de marcado, XAML. Aunque en este proyecto, como muchos de los elementos de la interfaz de usuario se generan automáticamente desde el código, no se ha hecho mucho uso de este lenguaje.

### 6.1.3 Controlador

El controlador en KVLeap es el componente del patrón MVC que más complejidad tiene y que más tareas realiza. Al ser una aplicación que permite a los usuarios interactuar con otras aplicaciones, está constantemente atendiendo a los diferentes eventos y acciones que hace el usuario, gestos, movimientos, clics...

El controlador es el que se encarga de recibir y procesar toda la información recibida del software de Leap, actualizar la vista (por ejemplo mover los punteros, ocultar/mostrar el teclado...), e interactuar con el modelo (realizar consultas a la base de datos, consultar la configuración, actualizar la base de datos...).

## 6.2 Módulos de KVLeap

Las clases que componen la aplicación KVLeap se han dividido en módulos diferenciados, de modo que los cambios en un módulo no tengan que afectar a los demás y que

no suponga ningún o casi ningún esfuerzo adaptar un nuevo módulo a la aplicación. En los siguientes apartados se verán los diferentes módulos que componen KVLeap.

### 6.2.1 Módulo de acceso a datos

Como KVLeap utiliza una base de datos, se ha decidido integrar en un módulo todo lo referente a la base de datos, de modo que sí por algún casual se decidiese cambiar la base de datos, solo hubiese que cambiar las clases que componen ese módulo.

Como la interacción con la base de datos que realiza KVLeap es escasa, se ha decidido crear una única clase que se encargue de todos los accesos y consultas a la base de datos. Esta clase se encarga tanto de la interacción con el diccionario de palabras, como con la configuración de KVLeap almacenada en la base de datos.

### 6.2.2 Módulo del teclado

Una de las principales funcionalidades de KVLeap es permitir realizar las acciones que se puedan realizar con un teclado, es por eso que la aplicación cuenta con un módulo dedicado exclusivamente a esto. Este módulo a su vez está dividido en dos partes, por un lado está lo referente a la parte visual del teclado, las teclas del mismo, los grupos de teclas que comparten código *(Figura 19 apéndice A)*, la gestión de los diferentes tipos de teclas... y por otro está la parte que se encarga exclusivamente de las escritura sobre la aplicación que esté activa.

### 6.2.3 Módulo del ratón

La otra funcionalidad principal de KVLeap es permitir realizar las acciones que se puedan realizar con un ratón, y al igual que se ha hecho con el teclado, la aplicación cuenta con un módulo dedicado a cumplir con esta funcionalidad.

Este módulo también está dividido en dos partes, la primera de ellas es la que se encarga de la gestión de los punteros, las posiciones de éstos, sus movimientos, la detección de colisiones... y la segunda de ellas es la que se encarga de generar todos los eventos del ratón cuando es preciso.

### 6.2.4 Módulo de predicción de texto

Aunque está muy relacionado con el teclado se decidió enmarcar en un módulo distinto la funcionalidad de la predicción de texto. Este módulo se compone de una sola clase, que es la encargada de mostrar y gestionar la barra de las palabras predichas por la aplicación, de aplicar el algoritmo de predicción de texto y de realizar las solicitudes al módulo de acceso a datos para obtener la lista de palabras predicha.

Al haber realizado esta funcionalidad en un módulo aparte, si en un futuro, se quisiera cambiar el algoritmo de predicción de texto por uno más complejo y más preciso (o utilizar servicios y aplicaciones externas para la predicción de texto), únicamente sería necesario modificar una parte de este módulo.

### 6.2.5 Módulo de interacción con Leap

Uno de los módulos más importantes de la aplicación, sin duda alguna, es el que se encarga de toda la interacción con el software de rastreo de Leap y del procesamiento de la información que éste envía. Este módulo es el que se encarga de realizar la configuración del controlador de Leap, recibir los datos que éste envía y escuchar los eventos que se generen como los gestos, conexión/desconexión del dispositivo Leap...

Este módulo también podría ser fácilmente sustituido dado el caso de que se quisiera utilizar otro dispositivo de reconocimiento gestual para las manos como podría ser el caso de Nimble VR (Nimble VR [25]).

### 6.2.6 Módulo de configuración

Este módulo es el que se encarga de todo lo referente a la configuración de la aplicación. En él se incluye a la ventana de configuración y a la clase que se encarga de cargar, establecer, gestionar y guardar la configuración establecida por el usuario.

### 6.2.7 Otros componentes

Aparte de los módulos comentados en los apartados anteriores, existen otros componentes a los que no se les ha otorgado la categoría de módulos porque tienen un objetivo muy específico que no se comparte con ningún otro componente. Entre estos componentes se encuentran los siguientes:

- La ventana principal: Es el componente principal de la aplicación y se encarga de inicializar y crear el resto de componentes y módulos.
- KVLeapColors: Una clase dedicada exclusivamente a la gestión de los colores de la aplicación. Los elementos gráficos de la aplicación utilizan los colores definidos en esta clase, de esta manera quedan abstraídos y se pueden cambiar con mayor facilidad.
- UIElementsGenerator: Una clase de apoyo con el objetivo de abstraer las partes comunes de la generación de diferentes elementos visuales. Ésta contiene métodos que crean y devuelven elementos visuales que luego la clase que ha realizado la llamada modificará para que cumpla con las necesidades específicas de ésta.

### 6.2.8 Interacción entre componentes

A pesar de que los componentes no comparten objetivos ni funcionalidades, interactúan entre ellos para realizar tareas más complejas que uno solo no puede realizar. A continuación se describe cómo interactúan los módulos que componen la aplicación.

- La ventana principal y el módulo de configuración están relacionados con todos (o casi todos) los módulos, debido a que la ventana principal es la encargada de inicializarlos y a que muchos de los módulos tienen parámetros configurables que se configuran en el módulo de configuración.

- El módulo de interacción con el software de Leap Motion también está relacionado con muchos módulos, ya que algunos como el módulo del teclado, el de predicción de texto o el módulo del ratón atienden a gestos y movimientos de los dedos que son detectados y gestionados por este módulo.

- El módulo del ratón contiene los punteros, por lo que interactúa con el módulo del teclado y el de predicción de texto para la detección de colisiones (además de los mencionados en los puntos anteriores).

- El módulo de acceso a datos interactúa con el módulo de predicción de texto para realizar consultas al diccionario de palabras de la base de datos, y con el módulo de configuración para guardar/cargar la configuración establecida por el usuario. Por último también interactúa con la ventana principal, ya que ésta es la encargada de inicializar el módulo de acceso a datos.

- Los módulos del teclado y de predicción de texto interactúan con los módulos que ya se han comentado en los puntos anteriores.

## 6.3 Estructura KVLeap

La estructura de KVLeap es un poco compleja, ya que tiene bastantes clases y muchas relaciones entre éstas. Es por eso que para poder ver mejor como están estructuradas las clases que componen la aplicación se utilizarán dos diagramas de clases. El primero de ellos *(Figura 14)* muestra las clases que componen los módulos de: acceso a datos,

teclado, ratón, predicción de texto e interacción con Leap. Omitiendo el módulo de configuración y la ventana principal, porque las clases de estos módulos están relacionadas con muchas de las otras clases y complicarían la comprensión del diagrama. Para ver la interacción con estas clases, se ha hecho un diagrama *(Figura 15)* especifico que muestra las relaciones de las otras clases y módulos con el módulo de configuración y la ventana principal. En ninguno de los dos diagramas se muestran las clases *KVLeapColors* ni *UIElementsGenerator*¸ por ser clases que únicamente se han utilizado como clases de apoyo para la abstracción de código y para tener un poco más estructurada la aplicación.

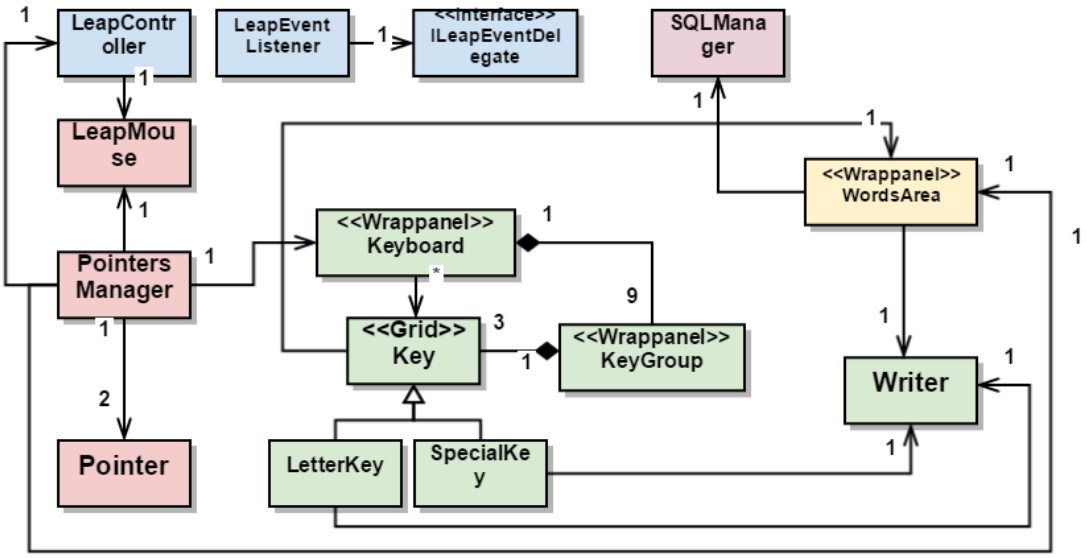


*Figura 14: Diagrama de clases 1: Interacción entre los módulos. En rojo el módulo del ratón, en verde el módulo del teclado, en amarillo el módulo de predicción de palabras, en azul el módulo de interacción con Leap y en rosa el módulo de acceso a datos*

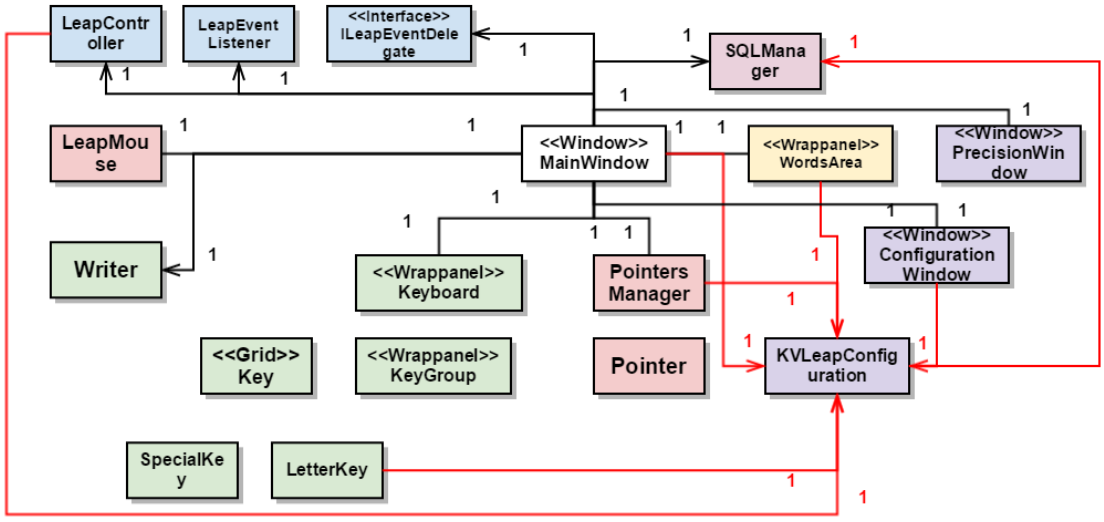


*Figura 15: Diagrama de clases 2: Interacción con MainWindow y módulo de configuración. En rojo el módulo del ratón, en verde el módulo del teclado, en amarillo el módulo de acceso a datos, en azul el módulo de interacción con Leap, en rosa el módulo de acceso a datos, en morado el módulo de configuración y en blanco la ventana principal. Las líneas rojas indican la interacción con el módulo de configuración, y las negras la interacción con la ventana principal.*

# Capítulo 7

# Gestión del proyecto

El desarrollo de este proyecto se ha realizado de forma incremental, siguiendo la metodología ágil SCRUM. El desarrollo del proyecto ha estado marcado por una serie de *sprints* en los que se ha incrementado el producto software obtenido del *sprint* anterior, obteniendo un nuevo producto potencialmente entregable.

Una vez obtenido cada uno de los productos potencialmente entregables, se realizaba una reunión de *sprint planning* con la directora del proyecto y con el cliente, en la que se obtenía como resultado una serie de requisitos (*Product Backlog Item)* que definían el trabajo a realizar durante el siguiente *sprint*.

## 7.1 Gestión del alcance

El alcance final del proyecto queda definido como la suma y la integración de los alcances de cada uno de los *sprints* realizados durante el ciclo de vida del proyecto. Es por esta razón, por la que el alcance de este proyecto se ha ido incrementando a medida que se han realizado nuevos *sprints*.

### 7.1.1 Sprint 1

El alcance del primer *sprint* está orientado a obtener un mínimo producto viable que permita mostrar al cliente el enfoque que se le ha dado al proyecto, de modo que si el cliente está de acuerdo con el enfoque propuesto, poder continuar en esa línea, o si por el contrario el cliente no está de acuerdo con el enfoque propuesto, cambiar éste. Es por esto que el alcance de este primer *sprint* únicamente cubre algunas de las funcionalidades básicas del producto final, que son las siguientes:

- Establecer una conexión con el controlador Leap Motion y recibir los datos de las posiciones, movimientos y gestos de las manos que éste ha rastreado e interpretarlos, para generar los eventos que permitirán interactuar con el ordenador.

- Interpretar los datos recibidos para dibujar y mover con el movimiento de la mano, un puntero que simula el cursor del ratón a través de la pantalla.
- Desarrollo de un teclado virtual que el usuario podrá mostrar/ocultar con un simple gesto.

### 7.1.2 Sprint 2

Recibido el visto bueno del cliente y de la directora del proyecto para poder continuar con el enfoque propuesto, se procedió a desarrollar un primer producto funcional que permitiera a un usuario interactuar con un ordenador. El alcance de este *sprint* está limitado a ofrecer al usuario la posibilidad de interactuar con un ordenador de la misma forma que puede hacerlo con el teclado y el ratón. En este alcance se incluyen los siguientes puntos:

- Ofrecer la funcionalidad básica de un ratón: La posibilidad de realizar cualquiera de los tipos de clic básicos de un ratón normal y realizar la acción asociada al tipo de clic realizado o la acción asociada al movimiento de la rueda del ratón. Todo esto mediante movimientos y gestos de las manos del usuario. La forma de hacer clic con el ratón en este *sprint* es únicamente mediante el gesto *Screen tap gesture*.

- La aplicación permite utilizar el teclado virtual para escribir caracteres (un único carácter (“a”) a modo de prueba) sobre otras aplicaciones.

### 7.1.3 Sprint 3

Una vez obtenido el primer producto funcional como *output* del segundo *sprint* se les presentó tanto al cliente como a la directora en la reunión de *sprint planning.* Tras rea-

lizar pruebas sobre la aplicación ambos coincidieron en que cumplía con los objetivos establecidos, pero que no era para nada *user friendly*, por lo que se añadió un nuevo requisito a la lista de requisitos para el tercer *sprint.* El alcance de este tercer *sprint* viene determinado por los siguientes puntos:

- Cambiar el algoritmo que se encarga de *mapear* las posiciones de los dedos que envía el Software de Leap Motion para obtener las posiciones en las que se representarán los punteros en la pantalla.
- La posibilidad de mostrar/ocultar una barra con palabras predichas por un algoritmo de predicción de texto en función de las teclas que el usuario ha pulsado.

### 7.1.4 Sprint 4

Al finalizar el tercer *sprint,* se les volvió a presentar tanto al cliente como a la directora del proyecto el producto obtenido y ambos volvieron a coincidir en que a pesar de notar mejoras en cuanto a la interacción con el usuario, todavía no era muy *user friendly* a la hora de realizar clic mediante gestos, ya que era bastante complicado. Tomado como prioridad este requisito, se procedió a definir el alcance del cuarto *sprint*:

- Añadir dos nuevas formas de hacer clic, una mediante el gesto *Key tap gesture* y otra basada en el tiempo que pasa el ratón en un área reducida cercana al punto sobre el que quiere hacer clic.
- Añadir la base de datos que contiene el diccionario de palabras al producto.
- Añadir una ventana de configuración en la que el usuario podrá activar/desactivar algunas opciones relacionadas con la forma en la que interactúa con el ordenador.

### 7.1.5 Sprint 5

Tras lograr la conformidad del cliente con el producto, se realizó un último *sprint* orientado a atar todos los cabos sueltos del proyecto para poder darlo por finalizado y entregárselo al cliente. El alcance de este último *sprint* incluye lo siguiente:

- Terminar la ventana de configuración añadiendo algunas nuevas funcionalidades para personalizarla así como la opción de cambiar los colores, y algunos parámetros, y una nueva opción a destacar que permite al usuario "suavizar" los movimientos de los punteros para que sea más *user friendly*.
- Aunque en todos los *sprints* se ha realizado una revisión y corrección de errores, en este último se ha realizado una general para que todo funcione correctamente.
- Guardar la configuración en la base de datos.

### 7.1.6 Fase de cierre

Los *sprints* comentados anteriormente están únicamente orientados a la aplicación desarrollada en este proyecto KVLeap. Sin embargo, el alcance del proyecto no está formado únicamente por esta aplicación, por lo que tras los *sprint*s realizados durante el desarrollo del proyecto se ha añadido una nueva fase orientada a cumplir con los puntos del alcance que no cubre la aplicación, y son los siguientes.

- Desarrollo de un sitio web con el objetivo de dar visibilidad al proyecto.
- Desarrollo de un ejecutable para la instalación de KVLeap.
- Manual de instrucciones de KVLeap.
- Redacción de un artículo para enviar a las JISBD (Jornadas de Ingeniería del Software y Bases de Datos) 2015 (JISBD, 2015 [12])

## 7.2 Gestión del tiempo

Un proyecto como éste está compuesto de una serie de hitos que tienen lugar en fechas específicas. El siguiente diagrama de hitos, recoge los más significativos del proyecto junto con la fecha en la que tuvieron lugar.

| Fecha | Hito |
|---|---|
| **21/12/2014** | Inicio del proyecto |
| **03/02/2015** | Reunión con la dirección y con el cliente para determinar el enfoque del proyecto, e inicio del primer *sprint* |
| **03/03/2015** | Primera reunión del *sprint planning* e inicio del segundo *sprint* |
| **31/03/2015** | Segunda reunión del *sprint planning* e inicio del tercer *sprint* |
| **28/04/2015** | Tercera reunión del *sprint planning* e inicio del cuarto *sprint* |
| **14/05/2015** | Cuarta reunión del *sprint planning* e inicio del quinto *sprint* |
| **01/06/2015** | Reunión con la directora para acordar los puntos de la memoria e inicio de la redacción de ésta |
| **03/06/2015** | Terminar el quinto y último *sprint* |
| **21/06/2015** | Terminar la memoria |
| **21/06/2015** | Cierre del proyecto |

*Tabla 5: Diagrama de hitos*

## 7.3 Gestión de costes y adquisiciones

Durante el ciclo de vida del proyecto (generalmente en las etapas más tempranas: inicio y planificación), se han ido realizando varias adquisiciones, que aunque en el caso de este proyecto hayan tenido un coste nulo, en otros casos podría no ser así. En este apartado se mostrarán las diferentes adquisiciones realizadas durante el ciclo de vida del proyecto y los costes asociados a éstas. Por un lado, se muestran las adquisiciones relacionadas con el producto software desarrollado *(Tabla 6)*:

| Adquisición | Coste real | Coste |
|---|---|---|
| **Dispositivo Leap Motion** | Entorno a los 90€ | 0€ Prestado por la directora |
| **Visual Studio 2013 Profes-sional** | Entorno a los 647€ | 0€ Gracias a la licencia gratuita ofrecida por la UPV/EHU mediante el acuerdo con Microsoft |
| **Leap Motion SDK** | 0€ | 0€ Disponible en el portal para desarrolladores de Leap Motion (Leap Motion [18]) |
| **SQLite** | 0€ | 0€ Licencia: Dominio público |
| **Diccionario Aspell** | 0€ | 0€ Open source |
| **Total** | 737€ | 0€ |

*Tabla 6: Tabla de adquisiciones de KVLeap*

Por otra parte, se muestran las adquisiciones relacionadas con el sitio web de la aplicación *(Tabla 7)*.

| Adquisición | Coste real | Coste |
|---|---|---|
| **Symfony** | 0€ | 0€ Licencia: Licencia MIT (open source) |
| **Bootstrap** | 0€ | 0€ Licencia: Licencia MIT (open source) |
| **Leapstrap** | 0€ | 0€ Licencia: Licencia MIT (open source, es una extensión de Bootstrap) |
| **Leap Motion SDK** | 0€ | 0€ Disponible en el portal para desarrolladores de Leap Motion (Leap Motion [18]) |
| **Dominio y hosting en One.com** | 0€ | 0€ Ofrecen un año de alojamiento gratuito para un dominio. |
| **Total** | 0€ | 0€ |

*Tabla 7: Tabla adquisiciones del sitio web*

Más allá del coste que supondrían éstas aplicaciones, está el tiempo invertido para el desarrollo del proyecto, es decir, el coste humano en horas, que el proyecto ha supuesto. En la *Tabla 8,* se recoge en horas la dedicación de una persona en este proyecto. Los valores de dedicación asociados al apartado del desarrollo de la aplicación KVLeap se ven incrementados por un proceso formativo constante durante todo el desarrollo de la aplicación debido a la inexperiencia tanto con la API de Leap como con el entorno WPF.

| Categoría | Tarea | Dedicación (h) |
|---|---|---|
| **Formación y entorno** | Toma de contacto con la API de Leap | 10 |
| | Preparación del entorno de desarrollo | 5 |
| **Gestión del proyecto** | Gestión de las adquisiciones | 5 |
| | Planificación | 5 |
| | Reuniones de control | 10 |
| | Seguimiento y control | 10 |
| | Gestión del alcance | 10 |
| **KVLeap** | Funcionalidad punteros | 22 |
| | Funcionalidad teclado | 22 |
| | Interacción con el *Software de Leap* | 14 |
| | Predicción de texto | 23 |
| | Funcionalidad base de datos | 15 |
| | Configuración | 24 |
| | Ventana principal KVLeap | 11 |
| | Pruebas y solución de errores | 14 |
| | Diseño y refactorización | 12 |
| | Optimización de la interacción con el usuario y el rendimiento de la aplicación | 12 |
| | Preparar el producto del *sprint* | 7 |
| | Ejecutable KVLeap | 7 |
| **Sitio web** | Diseño del sitio web | 2 |
| | Adquisición e integración de las tecnologías del sitio web | 4 |
| | Desarrollo del sitio web | 16 |
| | Contenido del sitio web | 10 |
| **Vídeo** | Realizar un vídeo de la aplicación | 6 |
| **Informes** | Documento presentado a JISBDXX | 4 |
| | Instrucciones KVLeap | 3 |
| | Memoria | 49 |
| **Total** | | 332 |

*Tabla 8: Tabla de dedicación*

# Capítulo 8

# Conclusiones

Actualmente están apareciendo nuevas interfaces de usuario que buscan romper con el patrón clásico del uso del teclado y el ratón y mejorar la experiencia del usuario mediante los sistemas de reconocimiento de gestos. Estas interfaces de usuario, permiten a los usuarios interactuar con los ordenadores por medio de los movimientos y gestos de su cuerpo.

Leap Motion y Kinect son dos claros ejemplos de este tipo de interfaces, dos sistemas de reconocimiento gestuales que mediante cámaras y sensores captan los movimientos y gestos de los usuarios. Estos sistemas son asombrosamente precisos en el rastreo de los gestos y movimientos del cuerpo, pero todavía no son lo suficientemente precisos como para sustituir los actuales dispositivos de entrada que permiten a los humanos comunicarse con los ordenadores.

Aun así, las empresas que están detrás de estos sistemas están trabajando para mejorar la precisión de éstos, tanto en el aspecto del hardware que se encarga de rastrear la información, como en del software que se encarga de procesarla. Además, cada vez hay más desarrolladores que apuestan por este tipo de tecnologías y se dedican a realizar aplicaciones para este tipo de sistemas, como es el caso de KVLeap.

KVLeap cumple con el objetivo para el que fue desarrollado, permitiendo a los usuarios utilizar un ordenador únicamente mediante los gestos y movimientos de sus manos. Sin embargo, esta forma de interactuar con un ordenador no es tan rápida y precisa como la forma en la que se interactúa con el ordenador mediante el uso de un teclado y un ratón. Esto imposibilita su uso en escenarios donde no se pueden cometer errores y donde el tiempo es un factor clave. A este último inconveniente, hay que sumarle lo que se conoce como el problema del *"brazo de gorila"*, que establece que los brazos de los seres humanos no están preparados para mantenerse levantados realizando pequeños gestos y movimientos durante un tiempo demasiado prolongado, por lo que tampoco podría utilizarse para interactuar con un ordenador durante largos periodos de tiempo.

No obstante, KVLeap sí podría ser utilizada en los escenarios que enmarcan el proyecto. Tanto en el quirófano, donde los médicos no necesitarían desinfectarse las manos tras tocar el ordenador, como en el taller, donde los mecánicos no necesitarían lavarse las manos para usar el ordenador.

En última instancia, se puede ver esta aplicación como una introducción a los usuarios de los ordenadores a los sistemas de reconocimiento gestuales, para que prueben una forma alternativa de interactuar con su ordenador y vean que la evolución de las interfaces de usuario no acaba en el teclado y el ratón.

Desde un punto de vista técnico de la aplicación, cabe destacar los siguientes aspectos:

- Desarrollo de una interfaz de usuario natural (NUI) que mediante un sistema de reconocimiento gestual (Leap Motion) permite a los usuarios interactuar con un ordenador, en tiempo real, utilizando únicamente los movimientos y gestos de sus manos.

- Desarrollo de un sistema eficiente que *mapea* las posiciones de los dedos del usuario en el área de interacción del controlador Leap Motion a puntos del sistema de coordenadas de la pantalla del usuario.

- Implementación de dos punteros y un teclado virtual que responden a los movimientos y gestos del usuario, y que son capaces de generar los eventos equivalentes a realizar diferentes tipos de clic y a pulsar teclas.

- Desarrollo de un algoritmo para la predicción de texto basado en el algoritmo T9 y en el número de veces que el usuario utiliza una palabra.

- Uso de una base de datos completamente integrada en la aplicación que no requiere ningún tipo de instalación adicional.

- Acceso y búsqueda eficiente de palabras en una base de datos con más de 800.000 palabras, mediante el uso de un flujo de ejecución alternativo al de la aplicación que se encarga de interactuar con la base de datos, de forma transparente al usuario.

- Optimización en el almacenamiento de palabras en la base de datos, almacenando únicamente la palabra en un formato (minúsculas) y aplicándole una máscara que permite escribir variantes de la misma palabra en cuanto al formato de sus letras (mayúsculas y minúsculas).

- Optimización del uso de memoria RAM (Random Access Memory) al limitar el número de palabras que se cargan en memoria por cada consulta a la base de datos.

- Desarrollo de un panel de configuración que permite al usuario configurar a su gusto diferentes opciones y parámetros.

### Comentarios sobre aspectos técnicos

Durante las pruebas realizadas con KVLeap se ha detectado que la aplicación en combinación con el software de Leap Motion consume una cantidad de recursos notable que puede incluso llegar a repercutir al sistema y a las demás aplicaciones.

Entendiendo por recursos el uso de memoria RAM (medido en MB (Megabytes)) y CPU (en % de CPU), a continuación se presentan el uso de recursos de la aplicación KVLeap y el software de Leap Motion (en un ordenador con 4GB (Gigabytes) de memoria RAM y un procesador Intel core i3 de 2.1GHz (Giga Hercios)) :

- KVLeap comienza utilizando alrededor de 20MB de memoria RAM y durante su uso con la predicción de texto desactivada, el consumo de memoria RAM oscila entre los 40 y 45 MB.

- Con la predicción de texto activada el uso de memoria RAM se incrementa un poco y generalmente oscila entre los 50 y 55 MB.

- El uso de memoria por parte del software de Leap Motion en ambos casos oscila entre 50 y 55 MB.

- Como indican los datos mostrados en los puntos anteriores, el uso de memoria RAM no es muy elevado, es en el uso de CPU donde más se nota el consumo de estas dos aplicaciones. KVLeap comienza con un uso de CPU del 0% y se mantiene así en los periodos en los que no interactúa con el usuario. Cuando interactúa con el usuario el uso de CPU incrementa y se mantiene entre el 3 y el 5% cuando no realiza peticiones a la base de datos.

- Si la predicción de texto está activada y el usuario está pulsando teclas, se están realizando consultas a la base de datos constantemente, y el uso de CPU se incrementa y oscila entre el 15 y el 30%, lo que puede llegar a ser un consumo excesivo si se combina con el consumo del software de Leap Motion.

- En cuanto al software de Leap Motion en los periodos en los que no interactúa con el usuario, el uso de CPU oscila entre el 1 y el 6% y en los periodos en los que sí está interactuando con el usuario entre el 15 y el 30%.

- En el caso peor con las dos aplicaciones a máximo rendimiento, al ejecutarse KVLeap en combinación con el software de Leap Motion podrían llegar a hacer un uso del 60% de CPU pudiendo llegar a ser un consumo excesivo.

- No obstante, como la predicción de palabras se ha considerado como un añadido al alcance y no como parte del objetivo principal del proyecto, se ha decidido no darle mucha importancia a este valor de uso de CPU, obtenido debido al número de búsquedas en la base de datos y al volumen de ésta.

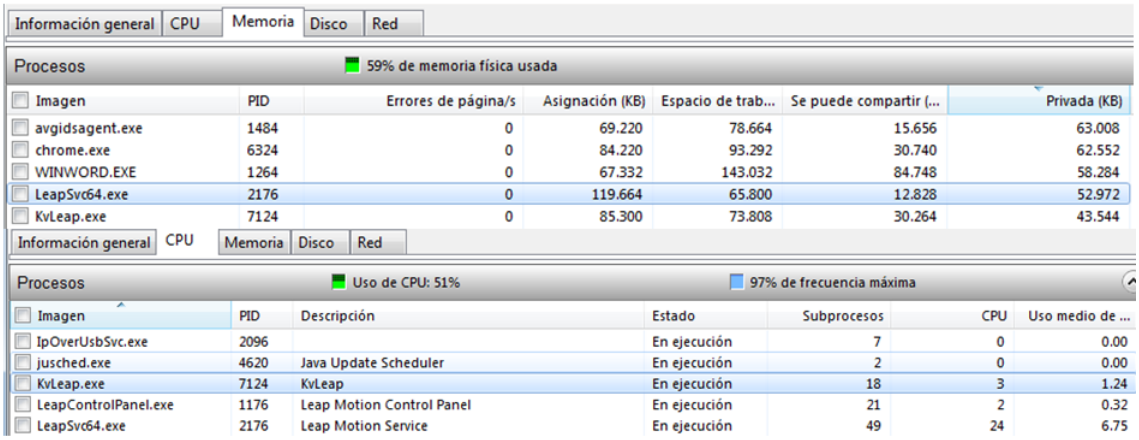


*Figura 16: KVLeap rendimiento sin predicción de texto*

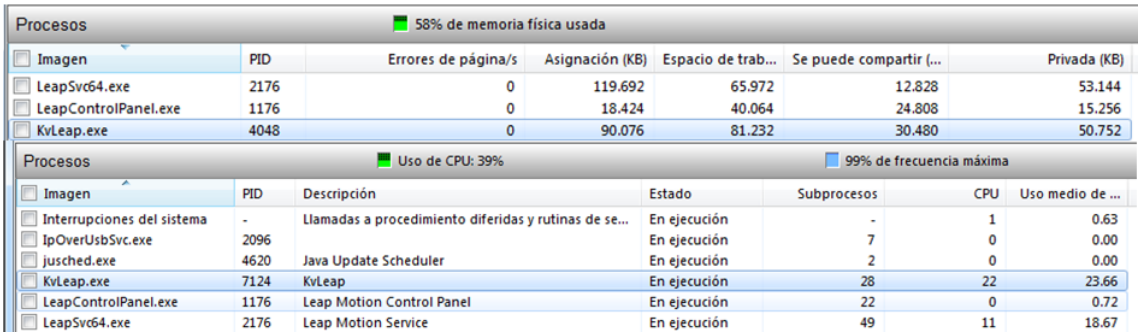


*Figura 17: KVLeap rendimiento con predicción de texto*

Por último, cabe señalar que el artículo enviado a las JISBD 2015 que se celebrarán en Santander, del 15 al 17 de septiembre, junto con la aplicación KVLeap han sido aceptados como contribución a las Jornadas, y para su presentación en éstas.

# Capítulo 9
# Líneas futuras

Este proyecto se ha ido desarrollando de forma incremental, partiendo desde un mínimo producto viable al que se le han ido aplicando extensiones y mejoras durante todo el ciclo de vida del proyecto. No obstante, enmarcado en un contexto académico y por unos recursos finitos, el proyecto ha tenido que llegar a su fase de cierre y al fin del mismo. Pero no es aquí donde termina el ciclo de vida del proyecto, ya que como producto software precisa de un mantenimiento y un proceso de evolución y mejora continua, marcado por los avances del contexto tecnológico del proyecto.

En este apartado se presentan tres diferentes líneas que marcarán el futuro de este proyecto: por un lado, están las mejoras y los puntos de extensión del propio proyecto; y por otro lado, la evolución de las interfaces de usuario y de los sistemas de reconocimiento de gestos.

## 9.1 Puntos de extensión y mejoras de KVLeap

KVLeap ha ido evolucionando y mejorando durante las iteraciones de la metodología SCRUM que se ha seguido en este proyecto. El proyecto partió de un mínimo producto viable y se fue mejorando y extendiendo hasta el cierre del proyecto. Sin embargo, todo proyecto se puede extender y mejorar, y KVLeap no iba a ser la excepción.

Las mejoras que se le podrían aplicar a proyecto están orientadas principalmente a mejorar la experiencia del usuario y la interacción con el mismo. Durante los diferentes *sprints* del proyecto se ha ido mejorando la interacción con el usuario, desde un primer producto en el que la interacción con el usuario no consiguió agradar al cliente, hasta el producto final en el que el cliente se mostró satisfecho con los resultados. Sin embargo, como ya se ha comentado, KVLeap todavía no es tan fiable ni tan rápido como los dispositivos de entrada actuales. Las mejoras a aplicar irían enfocadas a lograr una

comunicación con el usuario que se acerque más a la que se consigue mediante el teclado y el ratón.

Además de aplicarle esta mejora, a la aplicación también se le podrían aplicar puntos de extensión con el mismo objetivo. Para ello, se propone una lista de extensiones y mejoras que permitirían dar continuidad al trabajo realizado en este proyecto:

- Añadir nuevos gestos a los que la API de Leap tiene por defecto, a los que se les asociaría nuevas funcionalidades, tales como capturas de pantalla, cerrar ventanas, abrir determinadas aplicaciones...
- Extender la funcionalidad del ratón y del teclado para que permitan realizar más acciones que las que permiten un ratón y un teclado normales, como por ejemplo, añadir nuevos tipos de clic (clic con la rueda del ratón), o nuevas teclas y comandos (la tecla escape, el comando ctrl + alt + supr...).
- Mejorar el algoritmo de predicción de palabras para que no solo prediga las palabras en función de las teclas pulsadas, sino que prediga palabras también en función del contexto de estas. Además, una mejora en este aspecto permitiría acelerar el proceso de escritura del usuario.
- Mejoras en el rendimiento de la aplicación.
- Hacer que algunos de los parámetros de los gestos, así como la funcionalidad de los gestos que Leap Motion detecta, sean configurables, ya que no todos los usuarios son capaces de realizar los mismos gestos con la misma facilidad.

## 9.2 Líneas futuras en los sistemas de reconocimiento de gestos

Los sistemas de reconocimiento gestuales se encuentran en pleno desarrollo. No solo están apareciendo cada vez más, como es el caso de Nimble VR, sino que además, las organizaciones que hay detrás de los actuales están haciendo enormes esfuerzos para mejorar sus sistemas, para que finalmente puedan acabar imponiéndose a las actuales interfaces gráficas. Estos avances y la aparición de nuevas tecnologías son los que enmarcan el proyecto y los que permitirán a grandes rasgos realizar mejoras en el.

El sistema utilizado en el proyecto ha sido Leap Motion, uno de los pocos dispositivos de reconocimiento gestuales que realizan un completo seguimiento de las manos. Aun así, puede que en un tiempo aparezcan nuevos, o incluso que la propia organización lance nuevas versiones del mismo con mejores resultados, ya que los creadores están trabajando en mejorar su producto. Una prueba de ello son las diferentes versiones del SDK que han ido lanzando en las que se solucionan errores y se mejoran los algoritmos que procesan la información de rastreo que reciben del dispositivo.

Con el fin de poder adaptarse a las nuevas tecnologías que puedan aparecer en este contexto cambiante, KVLeap ha separado en un módulo la parte que se encarga de la integración y la interacción con Leap, tal y como se puede ver en la arquitectura de la aplicación, para poder sustituir este módulo en caso de que se quisiera adaptar la aplicación a otra tecnología.

## 9.3 Líneas futuras en las interfaces de interacción con los usuarios

Sin lugar a dudas, es en esta línea donde mayores avances puede haber en un futuro. Las interfaces de usuario naturales son ya una realidad y no pasará mucho tiempo para que se acaben imponiendo sobre las actuales interfaces gráficas.

No son pocas las películas de ciencia ficción, que transcurren en un futuro no muy lejano, en las que se muestran paneles de mando holográficos, o dispositivos asombrosos que se controlan mediante la voz y gestos de los usuarios. Como se suele decir, la realidad supera la ficción, y no pasará mucho tiempo hasta que esos dispositivos tan sorprendentes formen parte de la vida cotidiana. Y es que ya existen iniciativas para lograr este tipo de cosas, como por ejemplo la que ha tomado Microsoft con sus Hololens (Microsoft [21]), un proyecto de Microsoft que mediante unas gafas permite visualizar objetos digitales en el mundo real.

No se sabe exactamente cuándo se llegarán a implantar completamente estos sistemas holográficos, si es que finalmente se llegan a implantar. Lo que sí que es cierto es que la generación que sigue a las interfaces gráficas es ya una realidad, y que poco a poco éstas están evolucionando, y es aquí, donde por pequeño que sea, se espera que

este proyecto aporte su granito de arena, ya sea como aplicación, como idea o incluso como referencia de que es lo que no hay que hacer.

# Bibliografía y referencias


[1] Autodesk. (s.f.). *Leap Motion App Store*. URL: https://apps.leapmotion.com/apps/plug-in-for-autodesk-maya-2014/weblink. Visitado el 05/06/2015.

[2] Colgan, A. (9 de 8 de 2014). *Sitio web de Leap Motion*. Recuperado el 05 de 06 de 2015, del Blog oficial de Leap Motion: http://blog.leapmotion.com/hardware-to-software-how-does-the-leap-motion-controller-work/

[3] Colgan, A. (27 de 4 de 2015). *Sitio web de Leap Motion*. Recuperado el 05 de 06 de 2015, del Blog oficial de Leap Motion: http://blog.leapmotion.com/5-medical-and-assistive-technologies-being-transformed-with-leap-motion-tech/

[4] Design&Systems. (s.f.). *Design&Systems*. URL: http://designandsystems.de/en/project/holographic-interface/. Visitado el 05/06/2015.

[5] DexType. (s.f.). *Sitio web de DexType*. URL: http://dextype.com/. Visitado el 05/06/2015.

[6] Dr. Kari Heikkinen, Lappeenranta University of Technology, Finland Prof. Jari Porras, Chair WGB, WWRF (12 de 2013). UIs – past, present and future. *Wireles World Research Forum*. Recuperado el 02 de 06 de 2015, de http://www.wwrf.ch/files/wwrf/content/files/publications/outlook/WWRF_outlook_10.pdf.

[7] Ematras. (s.f.). *Leap Motion App Store*. URL: https://apps.leapmotion.com/apps/frog-dissection/windows. Visitado el 05 de 06 de 2015.

[8] Etxeberria, G. M. (2014). *Faborez App social de petición de favores instantáneos.* Proyecto de fin de grado UPV/EHU. Recuperado de https://addi.ehu.es/handle/10810/13330 el 09/05/2015.

[9] Dale L. Grover, Martin T. King, Clifford A. Kushler (1998). *Patente nº US5818437 A.* United States. Patente del algoritmo de predicción de texto T9.

[10] Guna, J., Greka, J., Matevz, P., Saso, T., & Jaka, S. (s.f.). An analysis of the precision and reliability of the leap motion sensor and its suitability for static and dynamic tracking. Recuperado de http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3958287/ el 06/06/2015.

[11] InMoov. (s.f.). *InMoov*. URLs: https://www.youtube.com/watch?v=6wULTIRCfx0 http://www.inmoov.fr/. Visitados el 05/06/2015.

[12] JISBD. (2015). *JISBD 2015*. URL: http://www.istr.unican.es/sistedes2015/jisbd.html

[13] Kinect. (s.f). *Kinect for Windows. Sitio web de Microsoft.* URL: https://www.microsoft.com/en-us/kinectforwindows/meetkinect/default.aspx

[14] Krols, D. (20 de 10 de 2010). *Blog U2U*. Recuperado el 05 de 2015, de http://blogs.u2u.be/diederik/post/2010/10/26/T9-encoding-decoding-and-prediction-in-C.aspx

[15] Leap Motion. (s.f.). *API de Leap Motion*. Obtenido del Portal de desarrolladores de Leap Motion: https://developer.leapmotion.com/documentation/cpp/api/Leap_Classes.html?proglang=current

[16] Leap Motion. (s.f.). *Leap Motion App Store*. Recuperado el 05 de 06 de 2015, de https://apps.leapmotion.com/

[17] Leap Motion. (s.f.). Problemas conocidos del SDK de Leap. *Leap Motion developer portal*. Recuperado el 03 de 06 de 2015, de https://developer.leapmotion.com/documentation/cpp/supplements/SDK_Release_Notes.html#known-issues

[18] Leap Motion. (s.f.). *Portal de desarrolladores de Leap Motion*. Recuperado el 2014, de https://developer.leapmotion.com/

[19] Leap Motion. (s.f.). *Sitio web de Leap Motion*. Recuperado el 05 de 06 de 2015, de Soluciones de Leap Motion para la industria: https://www.leapmotion.com/solutions/industrial

[20] Microsoft. (s.f.). *Documentación online de WPF*. Obtenido de https://msdn.microsoft.com/es-es/library/ms754130(v=vs.110).aspx

[21] Microsoft. (s.f.). *Microsoft HoloLens*. URL: https://www.microsoft.com/microsoft-hololens/en-us. Visitado el 10/06/2015.

[22] Minuun. (s.f.). *Sitio web de Minuun*. URL: http://minuum.com/future-of-wearable-typing/. Visitado el 06/05/2015.

[23] Myo. (s.f.). *Sitio web de Myo*. https://www.thalmic.com/myo/. Visitado el 05/06/2015.

[24] Nielsen, J. (1993). *Usability Engineering.*

[25] Nimble VR. (s.f.). *Cuenta de Youtube de Nimble*. URLs: https://www.youtube.com/watch?v=6JM9oFvqJ0o http://nimblevr.com/index.html Visitados el 05/06/2015.

[26] Nuance. (s.f.). *Sitio web de Nuance*. URL: http://www.nuance.com/for-business/by-product/t9/index.htm. Visitado el 12/06/2015.

[27] Palacio, J. (2015). Gestión de proyectos Scrum Manager. Scrum Manager®. Obtenido de http://www.scrummanager.net/blog/2015/04/libro-gestion-de-proyectos-con-scrum-manager/.

[28] Pointable. (s.f.). *Pointable*. URL: http://pointable.net/. Visitado el 06/05/2015.

[29] Polley, B. (s.f.). Form and Function 3D. Brendan Polley Biomedical Comunicator. URL: http://brendanpolley.com/. Leap Motion App Store. https://apps.leapmotion.com/apps/form-and-function-3d/windows. Visitados el 03/06/2015.

[30] Smith, C. (01 de 07 de 2014). *Sitio web oficial de Leap Motion*. Recuperado el 05 de 06 de 2015, de Blog oficial de Leap Motion: http://blog.leapmotion.com/tracking-hand-tremors-leap-motion/.

[31] SQLite. (s.f.). *Documentación online de SQLite*. Obtenido de https://www.sqlite.org/docs.html

[32] TedCas. (s.f.). *TedCas*. URL: http://www.tedcas.com/es. Visitado el 06/05/2015.

[33] Ten Toy Ray Gun. (s.f.). *Ten Tyo Ray Gun*. URL: http://tentonraygun.com/visual-touch-therapy/. Visitado el 05/06/2015.

[34] Weichert, F., Bachmann, D., Rudak, B., & Fisseler, D. (s.f.). Analysis of the Accuracy and Robustness of the Leap Motion Controller. Recuperado de http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3690061/ el 06/6/2015.

# Glosario

**Assets** En el contexto de esta memoria, se refiere a los elementos (modelos 3D, clases y funciones) que proporciona el SKD de Leap para el motor de desarrollo de videojuegos Unity. 20

**Feedback** El feedback se refiere a la retroalimentación que ofrecen los dos entornos de desarrollo que se están comparando en cuanto a los errores en el código del programa. 23

**Frames** En el contexto de esta memoría, se entiende por frame el conjunto de datos que el software de Leap envía en forma de paquete a otras aplicaciones. 10, 12, 12, 18, 37

**Framework** Un framework es un cojunto de herramientas, conceptos, ideas y prácticas para el desarrollo software. 19, 22, 38

**HandControler** El hand controler es uno de los assets que el SDK de Leap ofrece para el motor de videojuegos Unity. 20

**Java** Un lenguaje de programación orientado a objetos. 20, 20, 23

**Javascript** Un legunaje de programación orientado a objetos, debilmente tipado y dinámico. 39

**Las acciones básicas que puede realizar con un ratón y un teclado normal** Por las acciones básicas de un ratón se entiende: mover el cursor, mover la rueda del ratón y realizar clic (ya sea clic derecho, clic izquierdo, doble clic izquierdo o clic y arrastrar), y como las acciones básicas de un teclado se refiere a escribir los caracteres del teclado. 5

**Mapping** En este contexto hace referencia a la traducción entre dos puntos de dos sistemas de coordenadas distintos. 26, 54, 61

**Open source** Software de código abierto que se peude usar, modificar y distribuir libremente. 23, 57

**Output** Salida, en el contexto del *sprint* de KVLeap es lo que se obtiene como salida o como resultado, es decir el producto potencialmente entregable de ese *sprint* 53

**Plug-in** Un complemento o un añadido de una aplicación informática 14, 15

**Responsive** Una característica de los sitios web que ajustan su contenido a diferentes tipos de pantallas. 38

**Sprint** En este proyecto un sprint se refiere a un periodo de trabajo de unas 4 semanas tras el que se obtiene una versión mejorada y ampliada del producto obtenido en el sprint anterior. 52, 53, 54, 55, 56, 58

**Sprint planning** Sprint planning se refiere a las reuniones realizadas con el director y con el cliente para mostar el producto del sprint, dar por finalizado el sprint actual y comenzar con el siguiente. 53

**Startup** Empresa que está emergiendo. 14

**Suavizar** Suavizar, traducido del ingles *smooth,* se refiere a la aplicación de un algoritmo para mejorar el movimiento de las manos en función de los datos que recibe el software de Leap de las posiciones de las manos. De modo que se elimen datos incongruentes. 11, 36

**Testers** Un usuario que realiza pruebas de un producto software, en este caso de la aplicación KVLeap. i

**Thread** Un hilo de ejecución dentro de un proceso. 34

**Touchless** Que no requiere contacto físico para la interacción. 5

**Una conexión de socket local** Un proceso o hilo que permite la comunicación entre dos aplicaciones, en este caso dos aplicaciones que se ejecutan de forma local en la misma máquina. 12

**User friendly** Que es agradable para el usuario, fácil de usar y que hace que se sienta cómodo. 54, 55

**WebSocket** Similar al socket local, solo que en esta ocasión una de las aplicaciones se ejecuta en un servidor remoto. 12

**Windows Forms** Tecnología de Windows muy similar a WPF considerada como alterntiva. 20

# Apéndice A

# Algoritmo para la predicción de texto

En la memoria se hacen referencias al algoritmo utilizado para la funcionalidad de predicción de palabras de KVLeap. En esta aplicación se ha desarrollado un algoritmo tomando como referencias la idea del algoritmo T-9 para la predicción de texto desarrollado inicialmente por Tegic Communications y que ahora es propiedad de Nuance Communications (Nuance [26]), y una implementación simplificada del mismo obtenida de una entrada de un blog (Krols, 2010 [14]).

El algoritmo T-9 es el que utilizaban algunos teléfonos móviles de grandes compañías como Nokia, Samsung, Sony Ericsson... para la predicción de texto. El nombre del algoritmo se debe a que utiliza las 9 teclas numéricas del teléfono para realizar predicciones de la palabra que el usuario quiere utilizar.

Este algoritmo lo que hace es transformar las palabras almacenadas en el diccionario del dispositivo a códigos numéricos y luego a medida que el usuario va pulsando teclas que tienen un código numérico asociado, se le van mostrando las palabras que comienzan por ese código.

En este proyecto se ha adaptado esta idea al teclado de KVLeap realizando una nueva versión del algoritmo, que se describe a continuación.

Primero se deben obtener los códigos asociados a las palabras que componen el diccionario y guardar éstos en la base de datos junto a la palabra a la que están asociados. Para obtener los códigos primero se realiza una "limpieza" de la palabra para que caracteres como "á" y "a" tengan el mismo código asociado. En esta primera fase simplemente se transforman algunos caracteres menos comunes a otros más comunes. En KVLeap concretamente se hace de la siguiente forma *(Figura 18)*:

```
public string clearWord(string word)  {
    word = Regex.Replace(word, "[áàäâ]", "a");
    word = Regex.Replace(word, "[éèëê]", "e");
    word = Regex.Replace(word, "[íìïî]", "i");
    word = Regex.Replace(word, "[óòöô]", "o");
    word = Regex.Replace(word, "[úùüû]", "u");
    word = word.ToLower();
    return word;
}
```

*Figura 18: Limpieza de palabras del algoritmo de predicción de texto*

Una vez realizada la "limpieza" de la palabra, se procede a transformar cada uno de los caracteres que la componen a los códigos asociados a estos caracteres que son los que se pueden ver en el siguiente extracto de código *(Figura 19)*:

```
public string codify(string word) {
    KeyBoard keyboard = KeyBoard.getInstance();
    string code = "";
    code = clearWord(word);

    code = Regex.Replace(code, "[012]", "A");
    code = Regex.Replace(code, "[345]", "B");
    code = Regex.Replace(code, "[678]", "C");
    code = Regex.Replace(code, "[9¿?]", "D");
    code = Regex.Replace(code, "[!¡.]", "E");
    code = Regex.Replace(code, "[,;:]", "F");
    code = Regex.Replace(code, "[()'\"]", "G");
    code = Regex.Replace(code, "[#@%]", "H");
    code = Regex.Replace(code, "[&€$]", "I");
    code = Regex.Replace(code, "[-_]", "K");
    code = Regex.Replace(code, "[¬/]", "J");
    code = Regex.Replace(code, "[+=<]", "L");
    code = Regex.Replace(code, "[>~´]", "M");
    code = Regex.Replace(code, "[¨{}^]", "N");
    code = Regex.Replace(code, "[|ç`]", "O");

    //The order is important
    code = Regex.Replace(code, "[qaz]", "1");
    code = Regex.Replace(code, "[wsx]", "2");
    code = Regex.Replace(code, "[edc]", "3");
    code = Regex.Replace(code, "[rfv]", "4");
    code = Regex.Replace(code, "[tgb]", "5");
    code = Regex.Replace(code, "[yhn]", "6");
    code = Regex.Replace(code, "[ujm]", "7");
    code = Regex.Replace(code, "[iko]", "8");
    code = Regex.Replace(code, "[lpñ]", "9");

    //Special character on regular expresions
    code = code.Replace('*', 'K');
    code = code.Replace('[', 'J');
    code = code.Replace(']', 'J');
    code = code.Replace('\\', 'J');
    return code;
}
```

*Figura 19: Código que codifica las palabras*

Siguiendo el extracto de código fácilmente se puede deducir que reemplaza las letras de la palabra que aparecen en la parte de la izquierda por los códigos de la derecha, de esta forma las siguientes palabras quedarían así:

Balón → 51986

Talón → 51986

Galón → 51986

Salón → 21986

1993 → ADDB

Cuando ya se han insertado todas las palabras codificadas a la base de datos, solo queda ir codificando la palabra que el usuario está escribiendo y mostrarle las que coinciden con ésta.

Para codificar la palabra que el usuario está escribiendo no se utiliza la función anterior, sino que al crear las teclas del teclado se le asocia a esa tecla el código que le corresponde para evitar tener que ir codificando la palabra cada vez que el usuario pulsa una tecla. Lo que se va haciendo es ir guardando en una cadena todos los códigos de que ha ido pulsando y luego se buscan en la base de datos, las palabras que comiencen con esa combinación de códigos.

Por ejemplo si el usuario pulsa la tecla “s” que tiene asociado el código 2, en la base de datos se buscarán las palabras cuyo código comienza por “2”, tales como “silla”, “sofá”, “salón”... y a medida que el usuario va pulsado más teclas “sal” se van añadiendo códigos a la cadena a buscar “219” y las palabras sugeridas van cambiando, “sal”, “salita”, “salón”, “salmón”, “sapo”...

El algoritmo ha sido desarrollado para acelerar el proceso de escritura del usuario, de modo que si quiere escribir, por ejemplo, “balón” el usuario pueda pulsar independientemente cualquiera de las teclas que tienen el mismo código, que en este caso serían la “t”, la “g” y la “b”. Para facilitarle al usuario el trabajo de recordar las teclas que tienen el mismo código, se han distribuido los códigos en orden secuencial a las teclas adyacentes que son las que más probabilidades tienen de ser pulsadas al escribir una de estas teclas. En la *Figura 19* se puede ver cómo están asociados los códigos con las teclas.

# Apéndice B

# Tabla de los caracteres del teclado virtual

Como ya se ha mencionado en la memoria el teclado de KVLeap cuenta con tres juegos de palabras (alfabético, numérico y uno con caracteres especiales) y además ofrece la posibilidad de enviar algunos comandos. En la *Tabla 9* se recogen todos los caracteres y comandos que se pueden enviar con el teclado de KVLeap:

| Tecla | Acción |
|---|---|
| Caracteres del alfabeto | Escribe el carácter correspondiente a la tecla pulsada |
| Números | Escribe el número correspondiente a la tecla pulsada |
| Caracteres especiales | Escribe el carácter especial correspondiente a la tecla pulsada |
| .123 | Cambia el juego de caracteres al juego de caracteres numérico |
| #@* | Cambia el juego de caracteres al juego de caracteres especiales |
| ABC | Cambia el juego de caracteres al juego de caracteres del alfabeto |
| ␣ | Inserta un espacio y vacía la lista de palabras predicas en caso de que la predicción de texto esté activada. |
| ↲ | Inserta un salto de línea (equivalente a pulsar la tecla *ENTER*) |
| ↹ | Inserta un tabulador (equivalente a pulsar la tecla del tabulador) |
| ← | Borra un carácter y si la predicción de texto está activada, vacía la lista de palabras |
| ⇧ | Alterna entre mayúsculas y minúsculas (solo para el juego de caracteres alfabético). Es equivalente a pulsar la tecla *SHIFT.* Esta tecla se queda pulsada a esperas de que se pulse otra con la que se puede formar un comando, como por ejemplo, *SHIFT* + 5 (%). |
| supr | Equivalente a pulsar la tecla suprimir |
| ctrl | Equivalente a pulsar la tecla ctrl. Esta tecla se queda pulsada a esperas de que se pulse otra con la que puede formar un comando como por ejemplo ctrl + z (deshacer), ctr + c (copiar)... |

*Tabla 9: Tabla de los caracteres del teclado virtual*

# Apéndice C

# Artículo enviado a JISBD 2015

Las JISBD (JISBD, 2015 [12]) son un foro de referencia en la investigación de la Ingeniería del Software y las Bases de Datos en el ámbito iberoamericano. Este año las Jornadas celebran su XX edición y en ellas se presentarán diversas contribuciones en diferentes líneas de investigación, entre las que se encontrará este proyecto.

A continuación se presenta el artículo redactado para las Jornadas, en el formato establecido por las Jornadas, para el tipo de contribución “Demostración de herramienta”:

# KVLEAP: Interacción sin contacto (touchless) con ordenadores

K. Villalobos, D. Antón, A. Goñi, A. Illarramendi

Facultad de Informática UPV/EHU, Pº Manuel Lardizabal, 1 - 20018 Donostia-San Sebastián
kvillalobos001@ikasle.ehu.es

**Abstract.** Hoy en día existen diferentes alternativas para interactuar con los ordenadores. Las más extendidas y utilizadas son el teclado, el ratón o la pantalla táctil, aunque en los tres casos resulta necesario que las manos del usuario entren en contacto con algún dispositivo. Sin embargo, en determinadas circunstancias en las que la higiene de las manos es un factor importante, puede suponer un inconveniente. Mostraremos una aplicación, KVLEAP, que usando el controlador Leap Motion, un dispositivo que detecta y rastrea la posición y los movimientos de las manos en el aire, permite interactuar con un ordenador sin que las manos del usuario tengan que entrar en contacto con ningún dispositivo.

**Keywords:** Captura de movimiento, Leap Motion, Interacción touchless

## 1 Introducción

Actualmente están apareciendo dispositivos que permiten interactuar con los ordenadores sin seguir el patrón clásico de uso de ratón, teclado físico o pantalla táctil. Entre estos dispositivos podemos mencionar Kinect [1] y Leap Motion [2]. La tecnología subyacente en estos dispositivos se basa en el uso de infrarrojos para calcular la distancia a la que se encuentra el usuario y mediante un proceso de análisis interpretar sus acciones ya sean movimientos, gestos o posturas del cuerpo. Concretamente el dispositivo Leap Motion hace un seguimiento exclusivo de las manos de un usuario.

Guna et al. [3] y Weichert at al. [4] han analizado Leap Motion desde el punto de vista del hardware con el fin de comprobar la precisión de los datos que ofrece. Existen por otro lado, varios trabajos que aplican Leap Motion en áreas como el control industrial, la rehabilitación o la interacción con grandes cantidades de datos. En [5, 6] se usa Leap para crear dos sistemas de telerehabilitación para pacientes que han sufrido infarto. En [7] Bassily el at. adaptan Leap para controlar un brazo robot, destacando que mediante Leap consiguen una interacción más natural. Potter et al. [8] presentan un estudio sobre el uso de Leap como dispositivo para la interpretación de lenguaje de signos usando técnicas de inteligencia artificial. Por último Burgess et al. [9] han desarrollado un sistema de visualización de datos 3D controlado con Leap y destacan que permite una interacción más natural que otros sistemas de interacción en 2D.

En esta demostración presentamos la aplicación KVLeap que permite interactuar con el ordenador sin que sea necesario tocar físicamente ningún dispositivo.

adfa, p. 86, 2011.

## 2 KVLeap

KVLeap tiene como objetivo el convertir al dispositivo Leap Motion en un dispositivo *touchless* de entrada de datos, de tal manera que los gestos y movimientos realizados sobre Leap Motion sean capturados por KVLeap e interpretados como acciones equivalentes a las realizadas mediante el ratón y el teclado. En esta sección describimos la funcionalidad implementada en KVLeap: los eventos del ratón, el teclado virtual y la posibilidad de realizar entrada predictiva de texto. A continuación se muestra una demo de uso de KVLeap y por último escenarios de utilidad real para KVLeap.

### 2.1 Eventos del ratón

La funcionalidad de KVLeap equivalente a la ofrecida por un ratón es:

- Detectar y mover el puntero: El dispositivo Leap Motion permite detectar en tiempo real las posiciones de las manos y dedos del usuario y representarlos en pantalla. El usuario sitúa sus manos a una cierta distancia por encima del controlador Leap Motion y la aplicación KVLeap muestra un puntero asociado a cada dedo índice que el controlador reconoce. Estos punteros se mueven siguiendo los movimientos del dedo que tienen asociado. El puntero ligado al dedo que primero reconoce el controlador es el puntero principal y el otro el puntero secundario.
- Clic en el botón izquierdo o en el derecho: Cuando el usuario mueve el dedo asociado al puntero secundario como si fuese a tocar la pantalla, la aplicación detecta el evento *Screen tap gesture* (**Fig. 1**c). Si cuando el controlador detecta este gesto, únicamente el dedo índice se encuentra extendido, se genera el evento equivalente a hacer clic con el botón izquierdo en la posición del puntero principal. Si el dedo índice y el dedo medio se encuentran extendidos se genera el evento equivalente a hacer clic con el botón derecho.

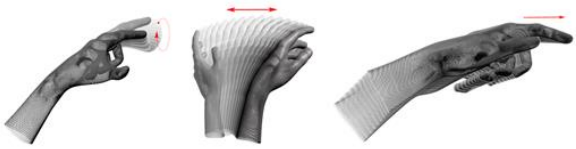

**Fig. 1 a) Circle gesture; b) Swipe gesture; c) Screen tap gesture**

- Doble clic izquierdo: si el dedo índice, el dedo medio y el dedo anular se encuentran extendidos genera el evento correspondiente a hacer doble clic con el botón izquierdo del ratón.
- Girar la rueda de un ratón: Cuando el usuario realiza un círculo con alguno de los dedos asociados a los punteros, la aplicación detecta el gesto *Circle gesture* (**Fig. 1**a). Si el círculo realizado es en el sentido de las agujas del reloj, la aplicación produce el evento correspondiente a girar la rueda del ratón hacia adelante. Si por el contrario el círculo es en el sentido contrario a las agujas del reloj, la aplicación genera el evento correspondiente a girar la rueda del ratón hacia atrás.

## 2.2 Teclado virtual

El teclado virtual aparecerá en pantalla si el usuario desliza la palma de la mano con la mano abierta desde abajo hacia arriba, *Swipe gesture* (**Fig. 1**b). Una vez que el teclado virtual esté visible, para poder escribir el usuario tendrá que deslizar el puntero principal sobre las teclas del teclado virtual. Al terminar, con un movimiento de la mano de arriba hacia abajo, la aplicación ocultará el teclado. Como la escritura con teclado virtual es excesivamente lenta se ha añadido en la aplicación la funcionalidad de predicción de texto. El usuario puede activar/desactivar la predicción de texto cuando lo considere oportuno. Cuando está activada, aparece una barra en la parte superior de la pantalla sobre la que se van mostrando las palabras sugeridas por la aplicación en función de las teclas que el usuario ha pulsado (excepto la última palabra que muestra los caracteres que el usuario ha pulsado). Si el usuario mueve la mano de izquierda a derecha, la aplicación detecta el gesto *Swipe gesture* (**Fig. 1**b) y actualiza la lista de sugerencias para mostrar nuevas palabras. Si el gesto es en dirección contraria, la aplicación vacía la lista de sugerencias.

## 2.3 KVLeap en acción

Una vez instalada la aplicación KVLeap y conectado el dispositivo Leap Motion al puerto USB del ordenador (ver Fig. 2 arriba) se puede realizar una búsqueda de información en Google, por ejemplo acerca de "Leap".

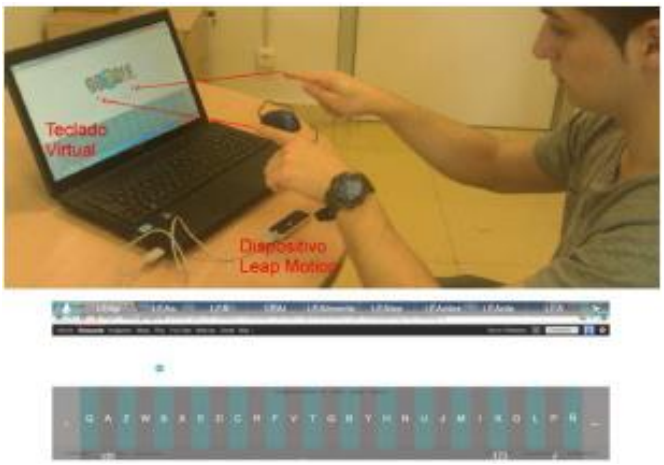


**Fig. 2 Interacción con Leap Motion utilizando KVLeap**

Para ello hay que seguir los siguientes pasos usando los eventos de ratón y teclado anteriormente descritos: 1)Abrir el navegador haciendo doble clic en su icono; 2)Seleccionar el favorito correspondiente a www.google.com; 3)Una vez cargado Google se puede hacer aparecer el teclado virtual con un swipe gesture de abajo a arriba; 4)Hacer clic en la caja de introducción de texto de Google; 5)Escribir las letras "L", "E" y "A", seleccionándolas sobre el teclado virtual; 6)Seleccionar la entrada

"LEAp" propuesta por la funcionalidad de predicción de texto (ver Fig. 2 abajo). Por último, tras hacer clic sobre el resultado www.leapmotion.com, se puede recorrer la página en su totalidad haciendo el correspondiente scroll con el gesto circle gesture (girar rueda del ratón).

### 2.4 Escenarios de utilidad real de KVLeap

Aunque usar KVLeap no sea práctico para acceder a Google, existen al menos dos tipos de escenarios donde sí es interesante: 1) cuando la interacción del usuario con un elemento no estéril pueda resultar perjudicial, o 2) cuando la suciedad intrínseca del trabajo del usuario haga desaconsejable dicha interacción. Un ejemplo de escenario de utilidad real del tipo 1) es el de un cirujano que deba interactuar con un ordenador para hacer algún tipo de consulta o realizar alguna anotación y que no pueda tocar un ordenador por encontrarse en el quirófano. Como ejemplo de escenario del tipo 2) tenemos el de un mecánico que desee realizar una consulta mientras se encuentra reparando un vehículo y que por tener las manos sucias no pueda tocar el ordenador.

Conviene señalar que la interacción *touchless* con el ordenador no es tan rápida ni tan sencilla como la forma en la que se interactúa con el ordenador mediante el uso de un teclado y un ratón. Sin embargo, en los escenarios de utilidad real señalados la interacción tendría lugar con aplicaciones cerradas, por lo que no se necesitaría toda la funcionalidad proporcionada por KVLeap.

## Referencias


1. Kinect for Windows, *2015(03/05*. Disponible en: http://www.microsoft.com/en-us/kinectforwindows/meetkinect/features.aspx.
2. Leap Motion, *2015(04/20)*. Disponible en: https://www.leapmotion.com/.
3. J. Guna, G. Jakus, M. Pogačnik, S. Tomažič and J. Sodnik. An analysis of the precision and reliability of the leap motion sensor and its suitability for static and dynamic tracking. *Sensors 14(2)*, pp. 3702-3720. 2014.
4. F. Weichert, D. Bachmann, B. Rudak and D. Fisseler. Analysis of the accuracy and robustness of the leap motion controller. *Sensors 13(5)*, pp. 6380-6393. 2013.
5. M. Iosa, G. Morone, A. Fusco, M. Castagnoli, F. R. Fusco, L. Pratesi and S. Paolucci. Leap motion controlled videogame-based therapy for rehabilitation of elderly patients with subacute stroke: A feasibility pilot study. *Topics in Stroke Rehabilitation* 2015.
6. M. Khademi, H. Mousavi Hondori, A. McKenzie, L. Dodakian, C. V. Lopes and S. C. Cramer. Free-hand interaction with leap motion controller for stroke rehabilitation. Presented at CHI'14 Extended Abstracts on Human Factors in Computing Systems. pp. 1663-1668. 2014.
7. D. Bassily, C. Georgoulas, J. Guettler, T. Linner and T. Bock. Intuitive and adaptive robotic arm manipulation using the leap motion controller. Presented at ISR/Robotik 2014; 41st International Symposium on Robotics; Proceedings Of. pp. 1-7. 2014.
8. L. E. Potter, J. Araullo and L. Carter. The leap motion controller: A view on sign language. Presented at Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration. pp. 175-178. 2013.
9. R. Burgess, A. J. Falcão, T. Fernandes, R. A. Ribeiro, M. Gomes, A. Krone-Martins and A. M. de Almeida. "Selection of large-scale 3D point cloud data using gesture recognition," in *Technological Innovation for Cloud-Based Engineering Systems* 2015.