

Software para la comunicación y educación de personas  
con dificultad de movimiento

Sergio González Francisco  
Enrique Laguna Munuera  
Daniel Ortiz Sánchez

GRADO EN INGENIERÍA INFORMÁTICA  
FACULTAD DE INGENIERÍA INFORMÁTICA

UNIVERSIDAD COMPLUTENSE DE MADRID



Trabajo Fin de Grado en Ingeniería Informática

30 de Junio de 2017

Director/es y/o colaborador:

Luis Garmendia Salvador  
María Asunción Reyes Corrochano

# Agradecimientos

*A todo el profesorado de la Facultad de Informática de la UCM, en especial al director de este proyecto por darnos la oportunidad de realizarlo, a la familia de Moisés y a él también, por darnos feedback en la medida que han podido, y por supuesto, a mis compañeros Daniel y Enrique, que sin el trabajo y esfuerzo de todos esto no hubiera sido posible.*

Sergio

*Muchas gracias a todos los profesores que han contribuido en mi formación y que me han ayudado a acabar esta maravillosa etapa.*

*Gracias también a mi tutor Luis Garmendia por darme la posibilidad de realizar un trabajo con el que poder hacer un bien colectivo y con el que encuentro tanta satisfacción personal.*

*Gracias a mis compañeros por hacerme esta etapa más amena y llevadera.*

*Y por supuesto gracias a mi familia y amigos por todo el apoyo recibido.*

Daniel

*Antes de nada, agradecer a mi familia todo el apoyo depositado en mí, sin el cual cerrar esta etapa no hubiese sido posible.*

*Asimismo, a las amistades y compañeros de proyecto, hacen que esta etapa haya merecido la pena.*

*Por último, agradecer a todos los profesores de que una manera u otra han influenciado mis conocimientos haciendo de esta etapa la más satisfactoria de mi vida.*

Enrique

# Índice

<b>Palabras clave/Keywords</b>	IV
<b>Licencias usadas/Repositorio</b>	V
<b>Resumen</b>	VI
<b>Abstract</b>	VII
<b>Estructura de la memoria</b>	VIII
<b>1. Introducción</b>	1
<b>2. Estado del arte</b>	2
2.1. ACAT. . . . .	2
2.2. Click2Speak. . . . .	6
2.3. Accessibility Scan. . . . .	9
2.4. CrossCursor. . . . .	11
2.5. HeadMouse y VirtualKeyboard. . . . .	11
2.6. DiLO. . . . .	12
2.7. Plaphoons . . . . .	13
<b>3. Metodología de desarrollo</b>	15
<b>4. Desarrollo del sistema</b>	21
4.1. Investigación. . . . .	21
4.1.1. Hipótesis de personas. . . . .	21
4.1.2. Obtención de datos. . . . .	22
4.2. Modelado. . . . .	22
4.2.1. Identificación de usuarios. . . . .	23
4.2.2. Procesar los datos. . . . .	24
4.2.3. Identificación y creación de esqueletos. . . . .	25
4.2.4. Priorización de esqueletos. . . . .	26
4.2.5. Desarrollo de personas. . . . .	27

4.3.	<b>Requisitos.</b>	28
4.3.1.	Enunciado de problemas y visiones.	28
4.3.2.	Identificación de las expectativas.	29
4.3.3.	Identificación de requisitos.	30
<b>5.</b>	<b>Metodología de programación</b>	<b>31</b>
5.1.	<b>Lenguajes de programación.</b>	<b>31</b>
5.1.1.	Java.	32
5.1.2.	HTML.	37
5.1.3.	JavaScript.	40
5.1.4.	CSS.	42
5.2.	<b>Modelos.</b>	<b>44</b>
5.2.1.	MVC.	44
5.3.	<b>Librerías.</b>	<b>46</b>
5.3.1.	jDom.	46
5.3.2.	FreeTTS.	48
5.3.3.	Bootstrap.	49
5.3.4.	jQuery.	50
5.3.5.	Responsive Voice.	51
<b>6.</b>	<b>Tecnologías usadas</b>	<b>52</b>
6.1.	<b>Eclipse.</b>	<b>52</b>
6.2.	<b>jsFiddle.</b>	<b>53</b>
6.3.	<b>Notepad++/Google Chrome.</b>	<b>54</b>
<b>7.</b>	<b>Almacenamiento de código</b>	<b>56</b>
7.1.	<b>GitHub.</b>	<b>56</b>
7.2.	<b>Google Drive.</b>	<b>57</b>
<b>8.</b>	<b>Descripción del programa</b>	<b>59</b>
8.1.	<b>Versión escritorio.</b>	<b>59</b>
8.2.	<b>Versión web.</b>	<b>66</b>
<b>9.</b>	<b>Conclusiones</b>	<b>70</b>

<b>10. Conclusions</b>	72
<b>11. Mejoras futuras</b>	73
11.1. Aplicación. . . . .	73
11.2. Web. . . . .	74
<b>12. Referencias</b>	76
<b>13. Apéndices</b>	77
13.1. Contribuciones individuales. . . . .	77
13.1.1. Sergio González Francisco . . . . .	77
13.1.2. Enrique Laguna Munuera . . . . .	78
13.1.3. Daniel Ortiz Sánchez. . . . .	79
13.2. Guía de uso. . . . .	82

# Palabras clave / Keywords

## Búsqueda en Español

- Software
- Aplicación
- Web
- Comunicación
- Interacción
- Social
- Niños
- Movilidad
- Discapacidad
- Accesibilidad
- Parálisis cerebral

## Search in English

- Software
- Application
- Web
- Communication
- Interaction
- Social
- Children
- Mobility
- Disability
- Accessibility
- Cerebral palsy

# Licencias usadas / Repositorio

Tanto en la versión de aplicación como en la versión web de nuestro producto usamos la licencia MIT. Se puede encontrar la misma en ambos repositorios en Github.

Repositorio Java: <https://github.com/daniOrtiz11/SW-Comunicacion>

Repositorio Web: <https://github.com/daniOrtiz11/SW-ComunicacionWeb>

## ¿Qué es la licencia MIT?

Es una licencia de software libre permisiva originaria como su nombre indica del Instituto Tecnológico de Massachusetts. Posee muy pocas limitaciones a la hora de reutilizar el software tanto en software libre como en propietario y es compatible con la mayoría de las licencias incluso con licencias con copyleft. No posee copyright lo que permite su modificación e inclusión libremente.

La única condición necesaria para su utilización sería que la nota de copyright, autor y todos los derechos deben ser incluidos en todas las copias o modificaciones. Es de las licencias más utilizadas en el software libre, sobre todo en plataformas como Github. Está presente en software tan conocidos como: JQuery, Node.js o Bitcoin.

Decidimos usar esta licencia debido a su gran permisividad con el producto ya que uno de nuestros principales objetivos es llegar al máximo de usuarios posibles. Sería un gran logro si con el paso del tiempo se siguiera mejorando o modificando esta aplicación por quien sea para ayudar a más gente, incluso en software propietario, por eso es necesario que sea una licencia de software libre y no restrinja de su modificación a nadie.

# Resumen

Actualmente vivimos en una época en la que se desarrolla o se ha desarrollado software para casi cualquier utilidad que podamos imaginar. El boom de las empresas tecnológicas ligado al aumento de los estudiantes que se forman en competencias informáticas ha contribuido a que llegemos donde nos encontramos hoy día. Aunque en ocasiones todo este avance y esta mejora se focaliza a nivel corporativo y empresarial dejando a un lado el aspecto social y humanitario que debería ser igual de importante en nuestra sociedad.

En este trabajo nos hemos querido centrar en esa parcela de software social y más específicamente dedicado a personas con dificultad de movimiento y con dificultad para comunicarse. Facilitando que estas personas puedan comunicarse con la gente de su entorno y de esta forma que su día a día y sus tareas cotidianas sean más fáciles y llevaderas.

Para poder llevarlo a cabo con éxito hemos necesitado comprender las necesidades y ponernos en el lugar de cualquier persona con estas dificultades para que el programa a realizar sea realmente útil y ayude a mejorar la vida de cualquier posible usuario.

Nos hemos centrado en la funcionalidad por encima de la espectacularidad y plasticidad visual ya que entendemos que el objetivo principal de nuestro programa debe ser cubrir una necesidad. Para ello hemos realizado una aplicación java y una web capaz de funcionar tanto en ordenador como en tablet y la mayoría de dispositivos móviles. Desde la web hemos incluido una opción para descargar el ejecutable de la aplicación java, reuniendo así nuestros dos productos en uno.

Por supuesto hemos decidido desarrollar el trabajo como software libre y abierto a todos los públicos para que sea susceptible de usar por el mayor número posible de usuarios y también susceptible de mejora por quien lo deseé. Más adelante se especifica concretamente este apartado. El objetivo es llegar al mayor número de personas y demostrar que aún quedan muchas necesidades por cubrir a pesar de todo el avance en este sector que vemos todos los días.

# Abstract

Nowadays we live in an era in which software is developing in every utility that we could imagine. Companies boom is linked to the students increase in learning technologies and informatics so this has contributed to our society to become what we are nowadays. However, sometimes this progress and its improvements are focussed in a corporative and enterprise level setting the social and humanitarian aspect apart that should be more important in our society.

In this TFG we want to focus on social software piece and specifically devoted to people with some movement and communication difficulties. We want help this people in order to be able to communicate with the people around them because of this their day life and their daily tasks will be easy going.

To make it real we needed to understand the necessities and put ourselves in the place of anybody with these difficulties so as the programme to be profitable and helpful to make the life of every user better.

We focused our research on the functionality over the showiness and visual plasticity since our programme principal aim must meet the specific needs. We have developed a java application and a web. Inside the web we have included an option to download the executable of the java application, we have achieved join our two products to one.

Of course, we decided to develop on this work as free software and open to everybody in order to be suitable of use by the largest possible number of users and also suitable for the improvement of everyone that needs it. Later, we will specify precisely this part. The main object is to reach as much people as possible and demonstrate that there are still a lot of necessities to cover despite the development that we could see every day in this sector.

# Estructura de la memoria

La memoria se compone de varios apartados, a continuación se procede a resumir brevemente los contenidos a desarrollar en cada uno de ellos:

- Estado del arte: En este apartado se resume la historia de la evolución tecnológica referentes a nuestro proyecto, por ejemplo aplicaciones que realicen funcionalidades similares.
- Metodología de desarrollo: Podemos encontrar la metodología empleada a la hora de desarrollar el proyecto, reuniones, investigación, redacción de la memoria...
- Estudio de usuarios: En este apartado se valoran los requisitos propuestos por el usuario, para adaptar de la mejor manera posible el software a sus necesidades, en nuestro caso, dado que es específico para un único usuario, intentamos ajustarnos lo más posible a sus requisitos funcionales.
- Forma de programación: Aquí se desarrollan todos los lenguajes usados en el desarrollo de la aplicación, se han elegido las más óptimas para cada funcionalidad y siempre pensando en que la aplicación es de código abierto. Algunos lenguajes usados son: Java, XML, HTML... y como modelo de desarrollo software nos hemos basado en el MVC (modelo vista-controlador).
- Tecnologías usadas: Podemos encontrar todos los programas usados durante todo el proceso de desarrollo del proyecto. Por ejemplo las tecnologías usadas para desarrollar el programa, hemos usado eclipse para programar en Java, Git para el control de versiones y gestión del código, Microsoft Office para la redacción de la memoria, Google Drive para compartir documentos.
- Almacenamiento del código: Aquí se muestran las formas de almacenamiento del código que hemos usado para desarrollar el proyecto. Principalmente han sido Git y Google Drive.
- Descripción del programa: En este apartado se explica detalladamente la funcionalidad del programa, desde que el usuario inicia la aplicación, la

- interacción con las diferentes vistas de la misma hasta el cierre del programa.
- Conclusiones: Podemos encontrar las conclusiones finales tras completar el desarrollo del programa, pruebas de usuario, distribución al público, aceptación social y estado del arte después de nuestro proyecto.
- Mejoras futuras: Aquí se muestran las posibles mejoras que se incluirán a la aplicación en caso de seguir con el desarrollo del software o posibles ideas para desarrolladores futuros que continúen con el proyecto.
- Aportaciones individuales: En esta sección se encuentran las aportaciones de cada integrante del grupo al proyecto.
- Referencias: En este apartado se recopilan todas las referencias, vínculos y citas usadas en la redacción de la memoria.
- Apéndices: Aquí se muestra todo el contenido que no ha podido ser categorizado en el índice.

# 1. Introducción

El objetivo de este proyecto es el desarrollo de una aplicación que permita la comunicación entre personas con parálisis cerebral, en concreto niños, y sus progenitores.

La idea de este proyecto surgió debido a un caso real, en el que la madre de un chico con parálisis cerebral no podía comunicarse de manera eficiente con su hijo. La aplicación facilita dicha comunicación, interactuando la misma con ambas partes.

Lo que se ha buscado con esta herramienta, es que permita mantener una comunicación fluida, siendo su manejo y su estética sencilla, de tal manera, que cualquier usuario principiante pueda entenderla y utilizarla sin requerir mucho tiempo para su aprendizaje.

Este proyecto supone una motivación importante tanto a nivel técnico como personal. El poder utilizar y mejorar los conocimientos aprendidos durante los años de carrera para ayudar a personas discapacitadas en su día a día, supone un extra de motivación y de satisfacción personal.

## 2. Estado del arte

Como se ha mencionado anteriormente vivimos en una época en la que se desarrolla o se ha desarrollado software de casi todo en lo que podamos pensar. Debido a esto es relativamente fácil encontrar gran cantidad de programas o aplicaciones relacionados con la comunicación y con la ayuda a personas con dificultades especiales. No hemos conseguido encontrar ninguno con la idea exacta que nosotros estamos desarrollando pero sí bastante similitudes en bastantes. Similitudes en objetivo y en desarrollo. Hemos intentado seleccionar los más destacados teniendo en cuenta su impacto, su uso y su innovación.

### 2.1 ACAT

La primera imagen que se nos viene a la cabeza al pensar en software para personas con dificultad de movimiento es la tecnología que usa Stephen Hawking para comunicarse a través de su silla y su pantalla. La plataforma se conoce como ACAT (Assistive Context-Aware Toolkit) y ha sido desarrollado por Intel. Supuso un cambio radical ya que el sistema que estaba usando el famoso físico estaba muy anticuado. Fue estrenado por él mismo a finales de 2014 y desde agosto de 2015 ha sido liberado por Intel convirtiéndolo en Open Source.

#### **¿Qué significa Open Source?**

Se trata del software distribuido y desarrollado libremente. Difiere con el término de software libre sobre todo en las cuestiones morales y éticas aunque ese debate se sale fuera del ámbito que nos ocupa. Se basa en la idea de que si el código es compartido y puede ser editado por todo el mundo será de mayor calidad que un software propietario, técnicamente el código abierto será superior al código cerrado.

Permite el acceso gratuito al código fuente para corregirlo, modificarlo o lo que se deseé. Aunque para cumplir que un software sea Open Source aparte del libre acceso debe cumplir una serie de requisitos:

1. Libre redistribución: se debe permitir que el programa sea entregado o vendido gratuitamente.
2. El código fuente debe estar siempre incluido o en su defecto se debe proporcionar una forma de acceder a él fácilmente y sin costes. Preferiblemente mediante alguna descarga en internet.
3. Se permite redistribuir una modificación libremente bajo los mismos términos que el programa original.

4. Se debe respetar la integridad del autor original del código y éste puede elegir que su nombre o el del programa sean dejados fuera de posibles modificaciones.
5. La licencia no debe discriminar a ninguna persona ni ningún grupo.
6. La licencia no puede restringir el uso del programa para ninguna actividad ni en ningún sector. El uso comercial está permitido.
7. Los derechos asociados al programa deben aplicarse a todos a los que se redistribuye el programa, sin necesidad de pedir una licencia adicional para estas terceras partes.
8. La licencia no debe ser específica de un producto. Los derechos asociados al programa no deben depender de que sea parte de una distribución de software mayor en particular. Si se usa la licencia en una parte, todo el programa debe tener la misma licencia.
9. La licencia no debe restringir a otro software. No se puede obligar a que algún otro software que sea distribuido con el software open source debe también serlo.
10. La licencia debe ser tecnológicamente neutral. No se puede basar en ninguna tecnología ni en ningún estilo de interfaz.

ACAT, como hemos explicado anteriormente, es una plataforma de open source desarrollado por Intel para permitir que las personas con enfermedades de las neuronas motoras y otras discapacidades relacionadas con la dificultad de movimiento puedan tener acceso completo a las capacidades y aplicaciones de sus ordenadores a través de interfaces limitados adecuados para sus condiciones especiales. Más específicamente, la ACAT permite a los usuarios comunicarse fácilmente con otros a través de la simulación de teclado, la predicción de escritura y la síntesis de voz. Los usuarios pueden realizar una serie de tareas como la edición, la gestión de documentos, navegación por Internet y acceder a mensajes de correo electrónico.

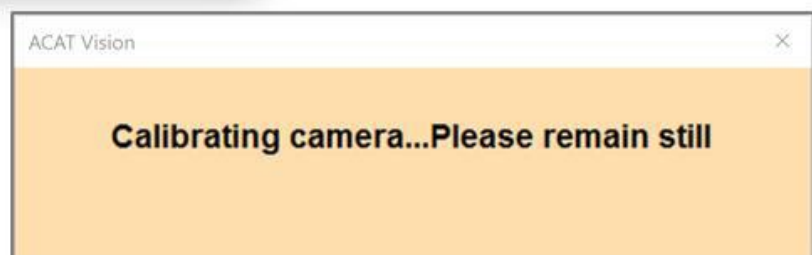
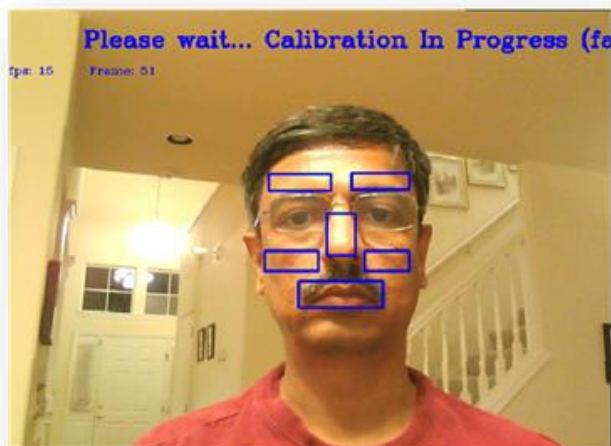
ACAT fue originalmente desarrollado por investigadores de Intel Labs para el Stephen Hawking, a través de un proceso de diseño muy iterativo y retroalimentativo en el transcurso de tres años. El propio Hawking fue fundamental para el proceso de diseño y fue uno de sus factores clave para la validación del proyecto dando su visto bueno. Después de que Intel implementara el sistema de Hawking, centraron su atención en la comunidad global y continuaron mejorando y desarrollando ACAT de un modo más configurable para apoyar un mayor número de usuarios con diferentes condiciones. La idea de Intel es que al hacer su plataforma open source desarrolladores de todo el mundo continuaran expandiendo el sistema mediante la adición de nuevas interfaces de usuario,

nuevas modalidades de detección, predicción de palabras y muchas otras características.

ACAT fue desarrollado en C# utilizando Microsoft Visual Studio 2015 y .NET 4.5. Está diseñado para funcionar con máquinas Microsoft Windows y puede conectarse a diferentes entradas de sensores tales como infrarrojos, cámara, botones y más. Está disponible en inglés, francés, español y portugués aunque actualmente se encuentra en desarrollando en nuevos idiomas. Hay que destacar que ACAT fue creado por Intel desde cero salvo por el software de predicción de palabras, impulsado por Presagio. Se basa en un motor de texto predictivo inteligente creado por Matteo Vescovi.

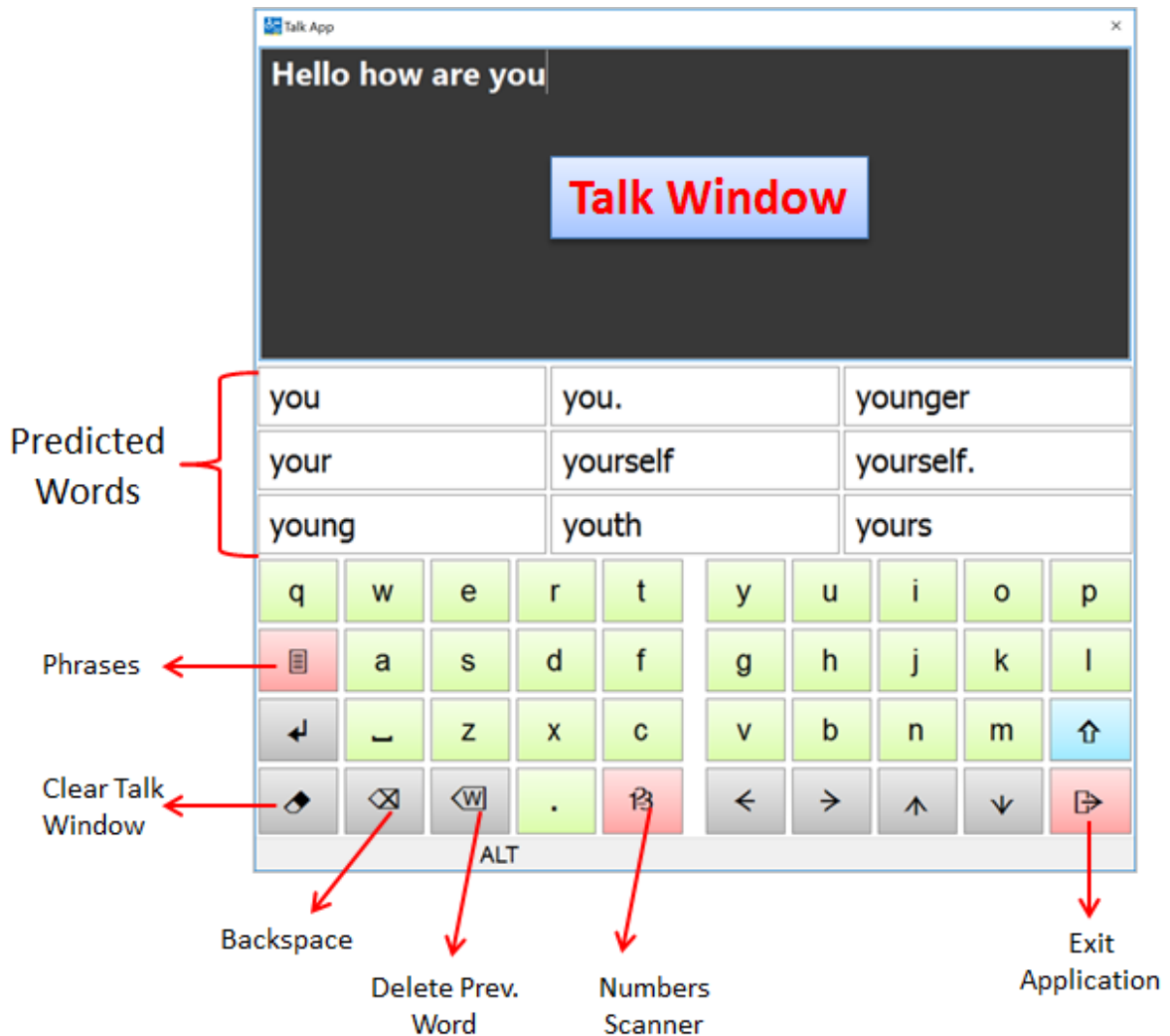
La plataforma se podría dividir en 3 partes principales para su funcionamiento:

- Sensores que miden algún movimiento que realizamos. Por ejemplo en el caso del profesor Hawking tiene un sensor de proximidad que está en sus gafas y permite que cada vez que mueve el pómulo el sensor se active y la señal se envía a una plataforma de software para ser analizada en su ordenador Windows. Pero la aplicación permite más sensores de entrada. Por ejemplo en la imagen que vemos a continuación podemos ver como usa una webcam para detectar y capturar los movimientos. Éste tipo de entrada a través de la cámara es el que ACAT usa por defecto.



- El software funciona como un conductor virtual entre el sensor y Windows. Tiene un teclado virtual y una simulación de ratón que le permite escribir

en el teclado en un interfaz temporizado. Se visualiza un cursor que va por cada letra y cuando llega a la letra deseada activamos el sensor (mover el pómulo en el caso anterior) y activa la letra. Ésta se escribe y hay un predictor de palabras. Así que cuando se ponen suficientes número de letras aparecen varias palabras directamente. Una vez que han aparecido las palabras en el predictor se puede seleccionar la deseada en vez de seguir escribiendo letra a letra. Una vez escrito todo se puede activar el sistema de lenguaje sintetizado.



- Esta es la tercera parte del sistema. Es una voz robótica personalizada para cada lenguaje. Tanto la predicción como la voz están personalizadas para cada lenguaje disponible ya que aspectos como la pronunciación o las palabras más frecuentes difieren en cada idioma.

Se puede encontrar disponible el código en Github desde [aquí](#) junto con las guías de usuario y de desarrollador.

Si únicamente se quiere a nivel de usuario se puede descargar la última versión desde [aquí](#) y sólo queda seguir los pasos que vienen en la guía de usuario a través del asistente de instalación que se proporciona.

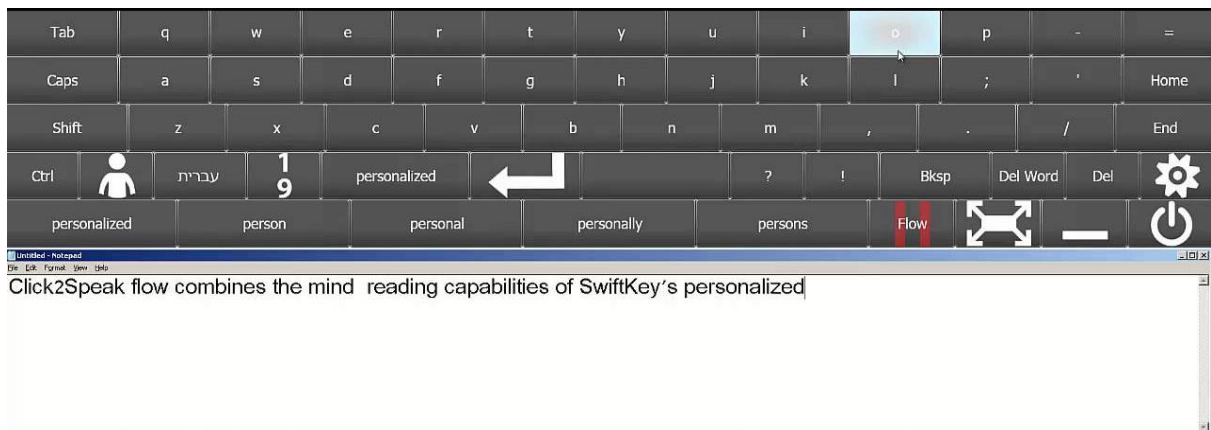
En conclusión nos encontramos con un programa muy completo. Muy avanzado en cuanto las operaciones a realizar y en el software de predicción de palabras. Sustituye casi por completo a un sistema personal a nivel de usuario, lo que hace que tareas cotidianas como escribir, navegar por internet, contestar a los correos sean mucho más llevaderos y realmente rápidos cuando se tiene experiencia usándolo. El único punto negativo que hemos encontrado es el peso de la aplicación y, a pesar de hacerlo open source para encontrar añadir más entradas, la dependencia de las formas de entrada aunque hay que decir que se han conseguido grandes avances. Posiblemente sea la más completa actualmente en su ámbito.

## 2.2 Click2Speak

Click2Speak es una aplicación que permite manejar un teclado especial y un ratón en pantalla para personas con problemas de movilidad por medio de los ojos. Permite diversos medios de entrada aparte de los tradicionales, como rastreador de ojos, ratón de pie o joystick. Es producido por la empresa Swiftkey.

Originariamente fue desarrollado por Gal Sont (enfermo de ELA) para intentar lograr alguna forma de comunicarse cuando su enfermedad estuviera muy avanzada. Pero, después de indagar, descubrió con pesar que, o bien no había aplicaciones mediante las que las personas con el tipo de discapacidad que él sufría pudieran comunicarse, o bien eran muy costosas, de manera que no eran accesibles. Por ello, recurrió a su compañero Dan Russ, también desarrollador de software, para que le ayudara a desarrollar alguna solución. De este modo, los dos programadores, con la colaboración de la compañía SwiftKey, concibieron una aplicación que permitiese a Gal controlar un teclado virtual con el movimiento de sus ojos. La aplicación del teclado la proporcionó SwiftKey, pero Gal tuvo que ingeniárselas para conseguir la cámara. La mayoría de los eye-tracking (rastreadores de ojo) del mercado tienen un coste prohibitivo (más de 1.000 dólares), por lo que intentó buscar otra solución más asequible. Compró las piezas que necesitaba a través de Internet (cámara, lente, infrarrojos iluminadores y un mini trípode) y utilizó software de open source para construir su propio sistema de eye-tracking con el que controlar el puntero de su ordenador: todo ello por menos de 500 dólares.

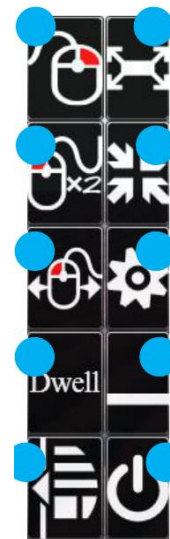
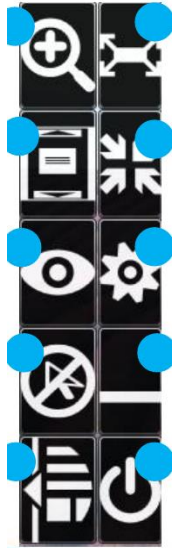
El sistema ofrece una predicción de palabras en más de 81 idiomas. La predicción se ajusta al idioma en el que nos encontremos y dentro del menú de configuración del teclado podemos editar alguna cosa más referida a ello. Se encuentra aún en fase beta por lo que sólo opera en Windows, recomendable a partir de Windows 7 y sólo en entorno de pc, pero se espera que pronto esté disponible la versión para Mac. A estos requisitos habría que añadir los requisitos del dispositivo de entrada específico. Por ejemplo el Tobii EyeX Controller (uno de los rastreadores de ojo más usados en esta plataforma) sólo opera con puertos USB 3.0 debido a la gran cantidad de datos que se reciben. La aplicación del teclado es gratuita y fácilmente instalable desde su [web](#). Una vez instalado reconocerá automáticamente tu idioma obteniéndolo de tu sistema y recomienda realizar los ajustes del teclado antes de conectar el dispositivo con el que vamos a interactuar normalmente. Una vez configurado y conectado el nuevo dispositivo debería funcionar y deberíamos interactuar por todo el sistema.



Desde el teclado aparte de las letras normales hay más funcionalidades:

- abrir la ventana de configuración.
- acceder al inicio de windows.
- text to speech: leer el texto resaltado.
- botón de ayuda: reproduce un archivo de sonido de alarma.
- panel inferior donde se muestra las palabras de la predicción.
- teclado de ayuda: acciones como copiar y pegar o impresión de pantalla.

Aparte dispone de varios teclados adicionales para editar o realizar las acciones que realizarán los dispositivos entrantes: panel eye-tracking (izq) y panel mouse (dcha)

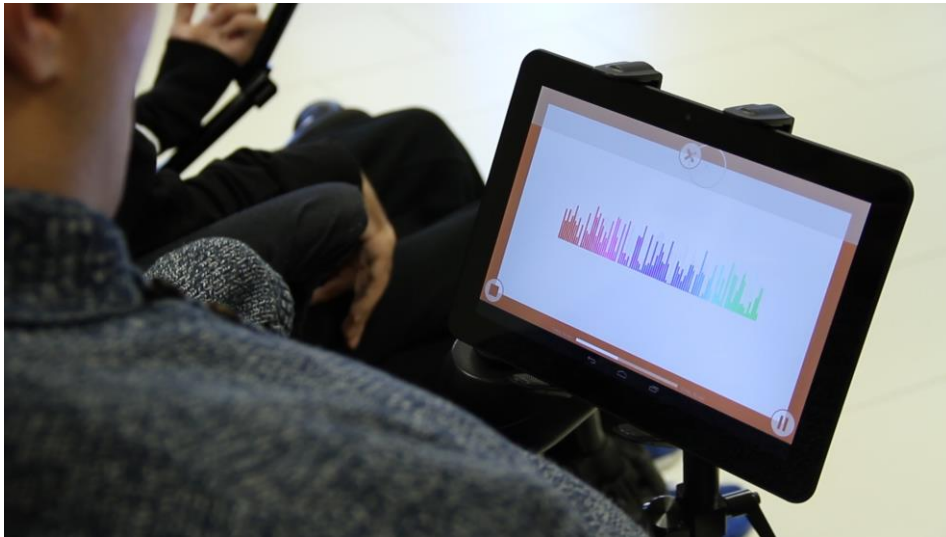


El último panel de la aplicación sería el panel de configuración desde donde se decide qué dispositivo usar y se configura el método para usar el ratón, moverse por la pantalla, los clics etc. Por ejemplo, en el caso de que se use un eye-tracking si se hacen los clics al parpadear los ojos, la velocidad a la que se captan los clics o como entrar en el modo silencioso sin reconocimiento de clics (por ejemplo mientras se ve una película).

La conclusión a la que llegamos es que Click2Speak es una aplicación útil. Es muy configurable en todos sus ámbitos, dando la opción de configurar hasta el más pequeño detalle para que el usuario tenga una interacción cómoda y a su gusto. El problema principal que encontramos es que al venir el dispositivo totalmente independiente del programa puede surgir algún problema si el programa no lo reconoce inmediatamente y un usuario que no sea avanzado en informática podría experimentar algún problema. Aunque cumple su función con una gran precisión y contar con el respaldo de una gran empresa detrás favorece a ello.

## 2.3 Accessibility Scan

Es una solución nativa de Android que permite a las personas con dificultad de movimientos interactuar con dispositivos móviles o tablets Android mediante un periférico desde donde se van dando toques y se va interactuando. Desarrollado por BQ y Limbika (empresa dedicada al desarrollo de tecnologías de apoyo).



La aplicación funciona mediante un sistema de barrido temporizado. Usa el cruce de dos ejes para situar el punto de la pantalla con el que se quiere interactuar. Los pasos serían los siguientes:

- un primer eje horizontal se desplaza hasta que el usuario pulsa el periférico.
- Un segundo eje barre la pantalla hasta que el usuario vuelve a pulsar.

También se ha adaptado el teclado con un sistema de barrido horizontal y vertical que permite la escritura mediante pulsaciones similar al anterior.

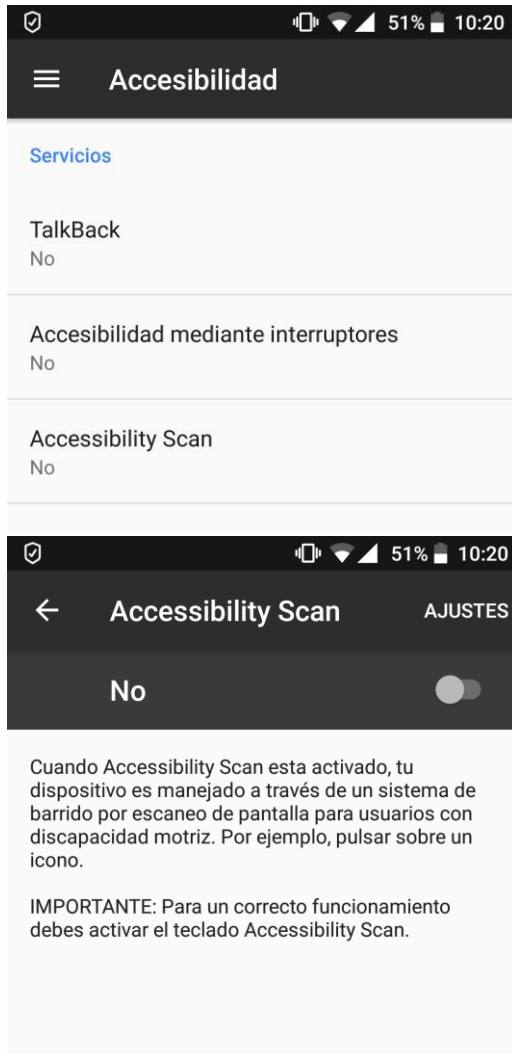
Dispone de otras funciones avanzadas como acceder a un menú de navegación abreviado, arrastrar y soltar, hacer doble clic, mantener pulsada un área de la pantalla e incluso hacer scroll.

Además de esto y complementando a lo anterior las tablets de BQ incluyen un sistema de comunicación aumentativa para que el usuario pueda llevar una vida mucho más funcional y social.

Dispone de una gran cantidad de ajustes para personalizar mucho el uso como la velocidad a la que se mueven los ejes, el sentido de los mismos, tipos de pulsaciones, colores o una ayuda de síntesis por voz en el menú en caso de que haya algún problema. Se recomienda seleccionar antes de empezar su teclado correspondiente para hacer la experiencia más completa.

Pasos para activar Accessibility Scan una vez hechos los ajustes basta con seguir los siguientes pasos en cualquier dispositivo BQ con Android:

1. Menú
2. Ajustes
3. Accesibilidad
4. Accessibility Scan



En conclusión que haya una alternativa directamente nativa en android para personas con dificultad de movimiento siempre es positivo. El único gasto que se debe hacer es comprar el dispositivo y en caso de que ya se tenga no hay que hacer nada más. Tendríamos que tener en cuenta también el periférico que en muchos casos debe ser personalizado para cada usuario. Permite gran cantidad de ajustes y reconoce el periférico desde el mismo menú de la aplicación lo que hace que sea muy cómodo. El único contrapunto que encontramos es que después de probarlo el sistema de barridos se nos hace algo lento, ya que tiene recorrer toda la pantalla. Aunque sí es muy fácil navegar por los menús y realizar tareas sencillas como sacar fotografías, usar la calculadora etc.

## 2.4 CrossCursor

CrossCursor permite controlar tanto el desplazamiento del cursor como las acciones de clic del ratón mediante un menú secuencial temporizado que se controla con una única acción de clic externa. Pertenece al portal Cátedras de Tecnologías Accesibles de la empresa Indra. Indra colabora con varias instituciones, universidades en su mayoría, para desarrollar soluciones para personas con discapacidad. Engloba los ámbitos de la educación accesible, empleo accesible y vida independiente. Las colaboraciones de Indra se centran tanto en el plano de desarrollo como en el apoyo económico a los centros donde se realizan. Los proyectos suelen ser llevados a cabo por varias universidades y coordinados por Indra y por la Fundación Adecco.

La aplicación está dirigida a personas con dificultad de movimiento que no puedan utilizar un ratón o un teclado convencional de la forma habitual y sólo puedan realizar una acción de clic mediante algún pulsador o dispositivo de accesibilidad.

Destaca por la rapidez con la que podemos empezar su uso. Basta con descargarlo desde [aquí](#) y extraer el fichero CrossCursor.exe para empezar a utilizarlo.

## 2.5 HeadMouse y VirtualKeyboard

Ambas aplicaciones también pertenecen al portal anterior Tecnologías Accesibles de Android. Se incluyen juntas debido a que se han desarrollado juntas y en múltiples ocasiones se suele hablar de ambas a la vez. Juntas superan las 400.000 descargas en multitud de países.

HeadMouse tiene como objetivo proporcionar un mecanismo de interacción alternativo y de bajo coste para personas con movilidad reducida que no puedan utilizar un ratón común. Se trata de un ratón virtual que permite el control del ratón del ordenador mediante una webcam y los movimientos de la cara y la cabeza. La aplicación fija la cabeza del usuario y a partir de ahí irá captando los movimientos que aparezcan delante de la pantalla. Las personas con dificultad de movimiento acceden al control del ratón a través de los movimientos de la cabeza con todas las funciones de arrastrar y mover del ratón incorporadas, mientras que las acciones faciales (abrir la boca o guiñar los ojos) se convierten en diversas modalidades de clic. La aplicación se puede descargar desde [aquí](#). A diferencia de la anterior ésta si requiere una instalación previa pero muy fácil de configurar.

VirtualKeyboard es un teclado virtual en pantalla que incorpora un sistema de predicción para facilitar la escritura.

Permite la escritura de textos mediante cualquier dispositivo de entrada capaz de controlar el cursor de la pantalla. El uso de la aplicación es totalmente intuitivo aunque no se tenga formación previa y además complementa y da continuidad al ratón virtual HeadMouse explicado anteriormente.

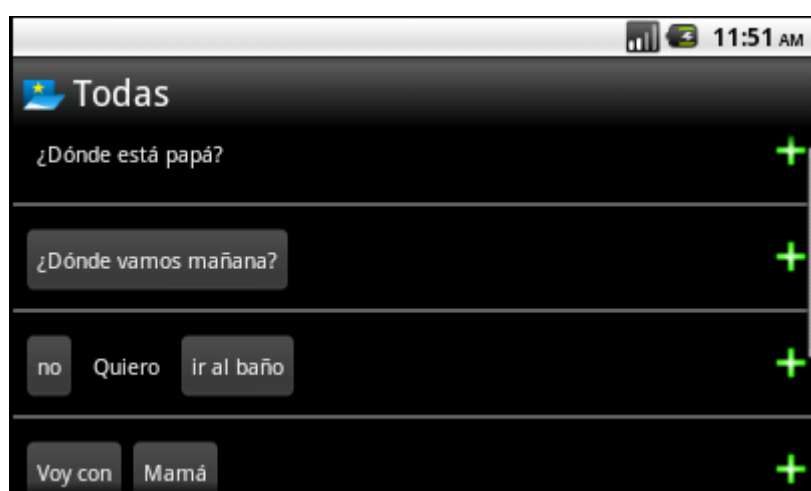
Funciona mediante un teclado que aparece en la pantalla del ordenador y que permite la escritura haciendo clics en símbolos que representan letras, símbolos y funciones con un único clic, pensado especialmente en personas que solo puedan tener éste sistema de entrada. También incorpora funciones de predicción de palabras cuyos algoritmos aprenden del modo de escribir del usuario y mejoran sus tasas de acierto con el paso del tiempo. Las pruebas realizadas escribiendo textos literarios de entre 15.000 y 20.000 palabras demuestran un ahorro de hasta un 40% en las pulsaciones de teclas necesarias para escribirlo, alrededor de 7.000 palabras menos.

Ofrece 5 diccionarios que incorporan las palabras más usuales del castellano, catalán, inglés, francés e italiano. El estilo de escritura de los usuarios se actualiza automáticamente y permite a la aplicación utilizar varios diccionarios en un mismo idioma y adaptarlos a usos específicos dependiendo de la situación: emails, textos literarios, notas...

Se puede descargar desde [aquí](#) y al igual que el anterior lleva un pequeño proceso de instalación.

## 2.6 DiLO

DiLO es una aplicación que permite a usuarios que tengan dificultad para interactuar con un teclado tradicional móvil comunicarse a través de frases ajustables dependiendo de las circunstancias del momento y de sus rutinas particulares. Está desarrollada por el Instituto Tecnológico y de Energías Renovables (ITER) de Tenerife.



Las frases, que son editables, pueden ser reproducidas desde el propio terminal por el altavoz e incluso enviarlas mediante SMS o email facilitando todo el proceso. Las frases se agrupan como máximo en cuatro categorías para facilitar su clasificación y su búsqueda. Todas las categorías y su contenido pueden ser editables por el usuario administrador, que será quien configure el contenido de la aplicación teniendo en cuenta las necesidades del usuario y así este podrá disfrutar y utilizar el comunicador de manera ágil y sencilla. Se recomienda que el administrador sea alguien que conozca al usuario en cuestión ya que será más fácil editar la aplicación de un modo más personalizado si se conoce los gustos o hábitos del usuario.

Para acceder al modo administrador sólo hay que mantener pulsada la esquina superior derecha de la pantalla durante dos segundos.

Se puede descargar para cualquier dispositivo Android desde la playstore [aquí](#). Desde nuestro punto de vista sería muy útil utilizar esta aplicación complementando alguna de las anteriores, como por ejemplo Accessibility Scan, ya que aunque está pensado para usuarios que no puedan usar un teclado convencional se supone que sí pueden interactuar con la pantalla de una forma normal y puede ser que eso suponga un pequeño problema en personas con bastante dificultad motora.

## 2.7 Plaphoons

Por último vamos a ver Plaphoons, que a primera vista no puede parecer una aplicación muy espectacular ni por su diseño puede llamar la atención pero debido a su buena acogida por los usuarios y a su largo recorrido hemos decidido incluirla. Plaphoons es una aplicación, tanto ordenador como dispositivo móvil o tablet (Android), que nos da la opción de crear tableros de comunicación y está pensada facilitar la autonomía de las personas que presentan una dificultad de movimiento y que no pueden comunicarse mediante el habla.



Nos permite crear tableros de comunicación (conocidos como plafones dentro de la aplicación) e ir introduciendo símbolos o pequeños dibujos para crear nuestros mensajes. Permite ir acumulando frases una tras otra para producir mensajes más elaborados y no sólo frases “hechas”. El mensaje final puede verse en pantalla, imprimirse o ser escuchado mediante voz sintetizada o grabada. Aparte de la opción de usarlo sólo como comunicador de mensajes a través del ratón también incorpora la opción de barrido temporizado, lo que hace que pueda ser usado por personas con dificultad de movimiento para usar los métodos de entrada habituales añadiendo valor al producto. A parte de usar los plafones disponemos de un teclado con letras y números que nos puede ayudar a complementar lo anterior.

La instalación de Android por defecto no necesita ninguna configuración previa para uso. Mientras que si se quiere usar la de ordenador, se quiere incluir alguna voz personalizada o crear plafones personalizados sí que hay que realizar una pequeña configuración y colocar los archivos en el lugar exacto aunque el propio autor, Jordi Lagares Roset, lo explica en sus tutoriales.

Una vez probada a veces sientes que necesitas ayuda en el modo usuario porque no es del todo intuitiva y definitivamente necesita un nuevo diseño, ya que la aplicación de Android es casi una copia exacta, incluido iconos paneles etc, de la aplicación de ordenador. Pero como ya hemos mencionado aunque a primera vista no lo parece es una aplicación muy completa y con muchas posibilidades.

### 3. Metodología de desarrollo

El trabajo se ha realizado de forma descentralizada y por consenso entre las partes involucradas siendo así repartido el trabajo entre los miembros del grupo para la realización de tareas concretas.

La toma de decisiones importantes el tratamiento de los problemas surgidos y la resolución de los mismos se han llevado a cabo entre todos los miembros del grupo

Como metodología de trabajo hemos aplicado la metodología ágil, que para proyectos software puede llegar a ser muy beneficiosa.

La interacción, flexibilidad y rapidez se ha vuelto una necesidad en la gestión de proyectos informáticos.

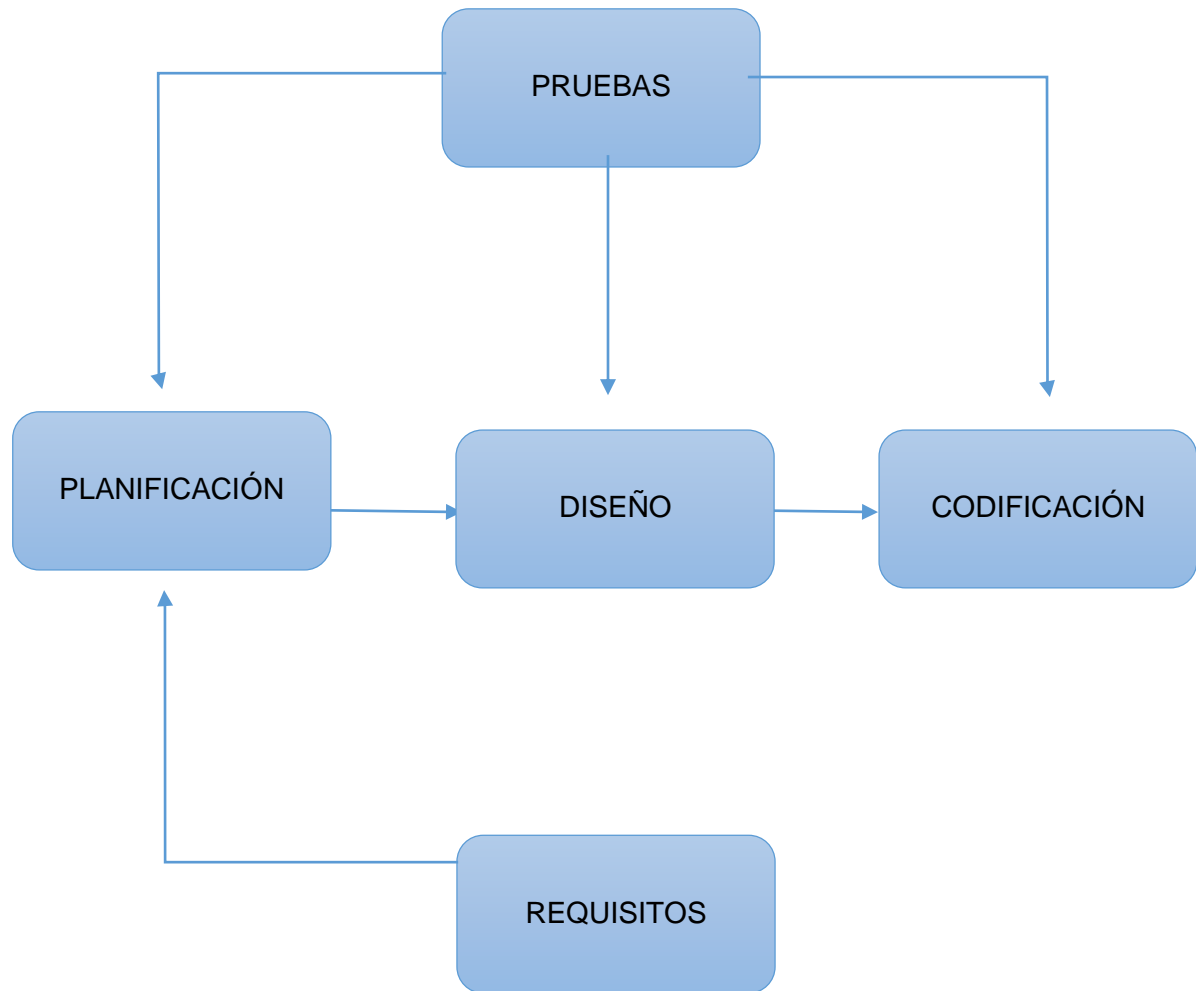
Hoy en día la tecnología y comunicación avanzan a una velocidad considerable, lo que ha provocado que la gestión de proyectos informáticos deba alcanzar la velocidad de los cambios ocasionados por esta aceleración. Esto por cierto, ha traído consigo nuevas necesidades en la metodología de trabajo y en la entrega del producto.

Es así, como la calidad, eficiencia, flexibilidad y rapidez en la entrega de un determinado producto se ha vuelto prioridad, lo que ha provocado la necesidad de implementar una nueva metodología de trabajo, la que se conoce como “Metodología Ágil”.

La Metodología Ágil es un marco metodológico de trabajo que plantea permitir mejorar la eficiencia en la producción y la calidad de los productos finales, tener la capacidad de respuesta al cambio en los productos y sus definiciones, y brindar la mayor satisfacción posible al cliente, a través de la entrega temprana y la retroalimentación continua durante la construcción del producto.

Esta metodología trae consigo diversos beneficios, pues permite una mayor flexibilidad que las metodologías tradicionales (en cascada e interactivas), debido a que éstas son menos capaces a ajustarse a las cambiantes necesidades de los clientes, del mercado, y de los nuevos desafíos que plantea la tecnología.

Al estar usando procesos ágiles para la gestión de proyectos, se perciben distintos beneficios para el cliente:



- **Flexibilidad en el proceso y las definiciones de los productos:** Permite que el equipo de desarrollo se adapte a los cambios y se beneficie de ellos en favor del cliente. Según íbamos desarrollando el software, nos han surgido nuevos requisitos por parte del cliente, que hemos ido integrando en iteraciones futuras.
- **Realimentación continua con el cliente:** De forma temprana el cliente recibe entregables de valor, lo que permite ver los constantes avances, logrando así, aportar en lo necesario para que el equipo vaya construyendo en la dirección correcta lo anterior, inmediatamente reduce de forma drástica los errores y la posibilidad de costosas correcciones, respondiendo a los cambios en requisitos de forma rápida y eficaz. Cuando teníamos un

versión nueva del software, esta era probada inmediatamente por el cliente, para poder obtener el mayor feedback pòsible.

- **Interacción constante:** Importante a la hora de dar tranquilidad al cliente sobre los avances del producto que recibirá (debido a que el producto se va analizando a medida que avanza), lo que sirve de gran valor cuando se ve en la necesidad de reportar a instancias superiores.
- **Calidad mejorada:** Esto significa que las prácticas de desarrollo ágil y sus constantes interacciones, proporcionan la funcionalidad suficiente como para satisfacer las expectativas del cliente con una alta calidad. La clave se encuentra en la continuidad de la calidad, es decir, la calidad es integral al proceso, y no añadida (la vista clásica). Por cada nueva iteración del software, se aumentaba la funcionalidad del producto manteniendo siempre su calidad.
- **Cuando los proyectos no están claramente definidos:** Esto apunta a que los requisitos del cliente se van clarificando a medida que el proyecto va avanzando, lo que permite la fácil adaptación del desarrollo para cumplir los nuevos desafíos. Tras las primeras pruebas, se detectaron enfoques erróneos en los requisitos principales de la aplicación, que posteriormente se subsanaron tras las pruebas.
- **Interacción y Comunicación:** La interacción entre los diferentes diseñadores y participantes es clave, es especialmente propicia para entornos orientados al trabajo en equipo.

### **Participantes:**

El grupo se compone de cuatro miembros, tres alumnos que han desarrollado el proyecto: Daniel Ortiz Sánchez, Enrique Laguna Munuera y Sergio González Francisco. Más un profesor-director, que ha guiado a los alumnos: Luis Garmendia Salvador.

Las pruebas se han realizado con María Asunción Reyes Corrochano y su hijo Moisés.

### **Jefe de equipo:**

Al tratarse de un grupo de tres personas, no ha habido división jerárquica entre los miembros del grupo.

### **Equipo de desarrollo:**

El trabajo se ha realizado de forma descentralizada y por consenso entre las partes involucradas, repartido el trabajo entre los miembros del grupo para la realización de tareas concretas. La comunicación entre los miembros del equipo es horizontal.

No tiene un jefe permanente, se nombra un jefe en función de cada tarea. La toma de decisiones importantes el tratamiento de los problemas surgidos y la resolución de los mismos se han llevado a cabo entre todos los miembros del grupo

### **Seguimiento y reuniones:**

El seguimiento y las reuniones entre los alumnos y los directores profesores se han llevado de manera esporádica pero seguida en el despacho 435 de la cuarta planta de la facultad de informática correspondiente al profesor Luis Garmendia Salvador.

En cada reunión con el profesor, se plantean nuevos requisitos y se itera partir de los mismos:

- **Iteración 0 (Junio 2016) :**

Reunión inicial con el tutor, descripción del proyecto, requisitos iniciales, documentación necesaria y planificación temporal del proyecto.

- **Iteración 1 (Octubre 2016):**

Nuevos requisitos funcionales, añadir teclado interactivo a la aplicación y apartado de ocio, que gestiona una base de datos XML con contenido multimedia.

Desarrollo por parte del equipo, dividiendo el trabajo según las funcionalidades de la aplicación.

- **Iteración 2 (Noviembre 2016):**

Primera POC del software, con funcionalidad básica, para realizar las primeras pruebas con el usuario.

Tras conocer la evaluación del usuario, se proponen nuevos requisitos, para mejorar esta primera versión en función de sus capacidades motrices.

- **Iteración 3 (Febrero 2017):**

Tras la implementación de los requisitos propuestos en la última reunión, se elabora la primera versión ejecutable de la aplicación.

Se vuelve a probar con el usuario para comprobar si se cumple todos los requisitos propuestos, así como el correcto funcionamiento y que el alcance de la aplicación sea el correcto.

Tras las pruebas, se sugiere que la instalación del software, sea más sencilla e intuitiva, por lo que se cambia la generación de la base de datos de forma automática. La velocidad se decide reducir, para que el usuario pueda interactuar de mejor manera con la aplicación y en cuanto al texto se decide aumentar de tamaño para su correcta visualización.

- **Iteración 4 (Abril 2017):**

Se completan los requisitos de la reunión anterior, y se comienza el desarrollo de la aplicación en formato web, para que sea más accesible desde cualquier dispositivo, con la posibilidad de usarse desde móvil o Tablet.

- **Iteración 5 (Mayo 2017):**

Tras un mes de desarrollo, el portal web es completado con éxito, siendo accesible desde cualquier dispositivo. Se entrega al tutor la URL para realizar las pruebas pertinentes, y se programa una reunión futura para realizar la entrega final del proyecto, incluyendo la memoria y software completo.

- **Iteración 6 (Junio 2017):**

El tutor, después de haber realizado las pruebas de usuario, confirma que la calidad del software es óptima ya que cumple todos los requisitos especificados al inicio del desarrollo. Puntualiza pequeños detalles de estilo que serán depurados en la entrega final, por ejemplo: El color en la opción 2, al acceder en el modo madre o administrador, no quedaba muy claro, en que modo nos encontrábamos, por tanto decidimos, que cuando nos encontrásemos en dicho modo, colorear el teclado en gris.

### **Comunicación entre el grupo:**

La comunicación entre los tres alumnos del grupo y el profesor se ha llevado a cabo mediante correos electrónicos. Por otro lado la comunicación entre los tres alumnos se ha llevado a cabo no sólo a través de correos electrónicos sino también a través de Whatsapp y google drive.

## **4. Desarrollo del Sistema**

### **4.1 Investigación**

#### **4.1.1 Hipótesis de personas**

En este apartado tratamos de realizar una definición inicial de los tipos de usuarios que usarán nuestra aplicación o web.

Durante toda la fase 3 nos referiremos a producto como el conjunto de la aplicación y web.

Nuestra hipótesis de persona establece que los usuarios serán:

- Personas con dificultad de movimiento que usen la plataforma para comunicarse.
- Personas “administradoras” que configurarán los detalles y preparen el producto para que los usuarios anteriores puedan hacer un correcto uso, de las cuales podemos distinguir dos tipos:
  - Personas que día a día usan la aplicación y están muy cercanas al usuario con dificultad de movimiento. Por ejemplo un familiar.
  - Personas que usan la aplicación de forma esporádica para una comunicación puntual con el usuario por alguna necesidad.

A continuación realizamos una hipótesis acerca de los distintos tipos de usuarios, teniendo en cuenta diversas dimensiones (objetivos, actividades, perfil), que podremos refinar al concluir la fase de investigación.

Los usuarios con dificultad de movimiento tienen como objetivo principal poder comunicarse con la gente de su alrededor de una forma rápida y sencilla. Asumimos que es posible que no tengan ningún conocimiento de informática y que sus funciones cognitivas estén algo mermadas en algún caso. Las actividades que suponemos que van a realizar son las de usar el producto en el modo usuario, todas las opciones, durante tiempos no excesivamente largos pero sí repetidas veces a lo largo del día y en ningún caso llegar a tener acceso al modo administrador.

Los usuarios con perfil de administrador tienen 2 objetivos principales:

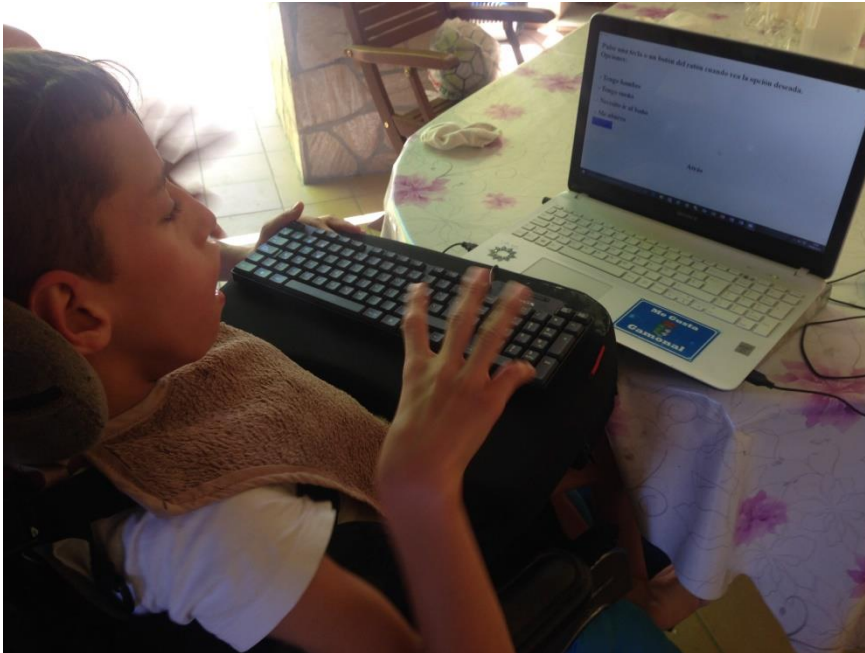
1. Ayudar a los usuarios anteriores a que se sientan cómodos usando el producto a través de la configuración.
2. Conseguir interactuar de la forma más sencilla posible con los usuarios a través de cualquiera de las 3 opciones disponibles.

Asumimos que es posible que sólo tengan un conocimiento de informática a nivel de usuario o en el mejor de los casos es posible que sepan algo de ofimática. Por tanto hemos de tener en cuenta que la configuración e interacción que tengan que realizar debe ser sencilla y fácil de comprender.

## 4.1.2 Obtención de datos

Para la recolección de datos inicial necesaria para las siguientes fases y la anterior *Hipótesis de Personas* hemos usado las iteraciones iniciales y reuniones con nuestro tutor quien nos proporcionaba “feedback” sobre casos de personas reales

y pruebas, tal como hemos reflejado en el apartado 2 de ésta memoria. Más concretamente hemos obtenido gran ayuda por parte de Moisés y su familia para la obtención de éstos datos, aprovechamos nuevamente para darles las gracias. Aquí podemos ver a Moisés en una de las pruebas iniciales:



## 4.2 Modelado

Nuestro objetivo es definir las personas de nuestro producto (tanto aplicación como web). Una persona es un modelo, una descripción detallada de un individuo imaginario que representa y encapsula información sobre un grupo de usuarios objetivo de un producto.

Hemos decidido que el diseño **top-down** se adecua más a nosotros porque nuestro producto se rige por el tipo de usuario que la utiliza. Varía en función de si es un usuario administrador o un usuario con dificultad de movimiento. Además, habiendo realizado la hipótesis de personas y usando la información obtenida en la fase de investigación podemos encajar las conclusiones obtenidas dentro de las categorías de usuario.

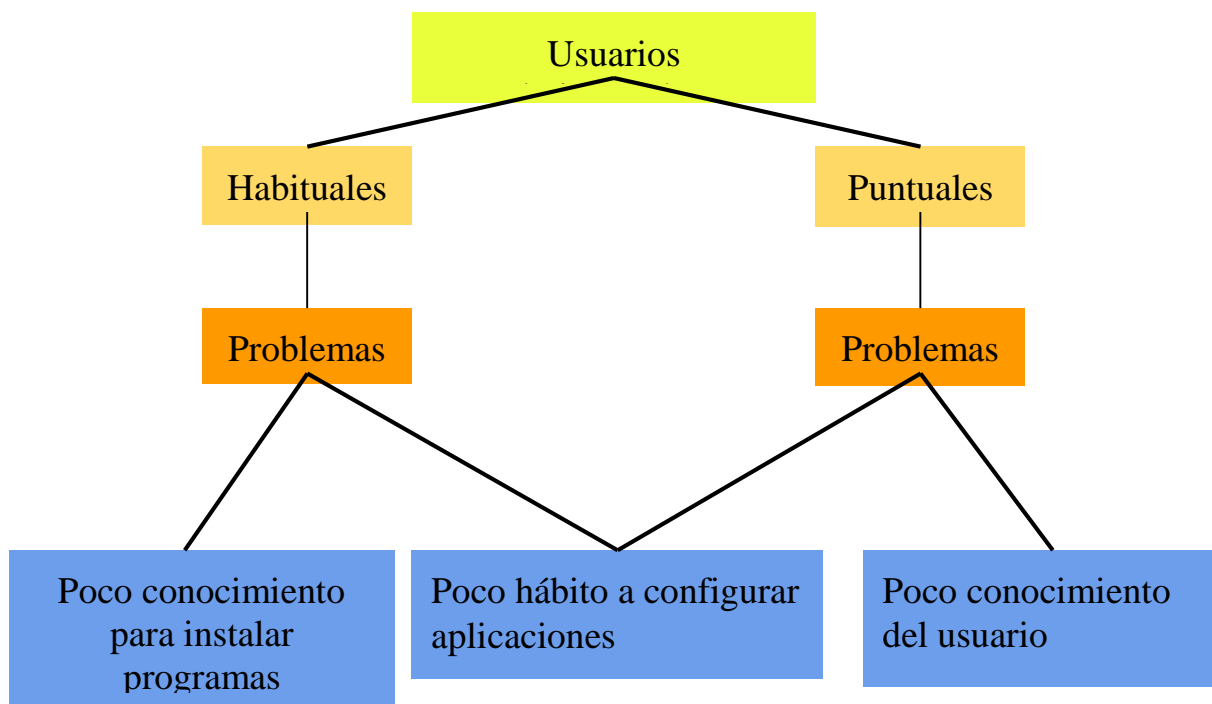
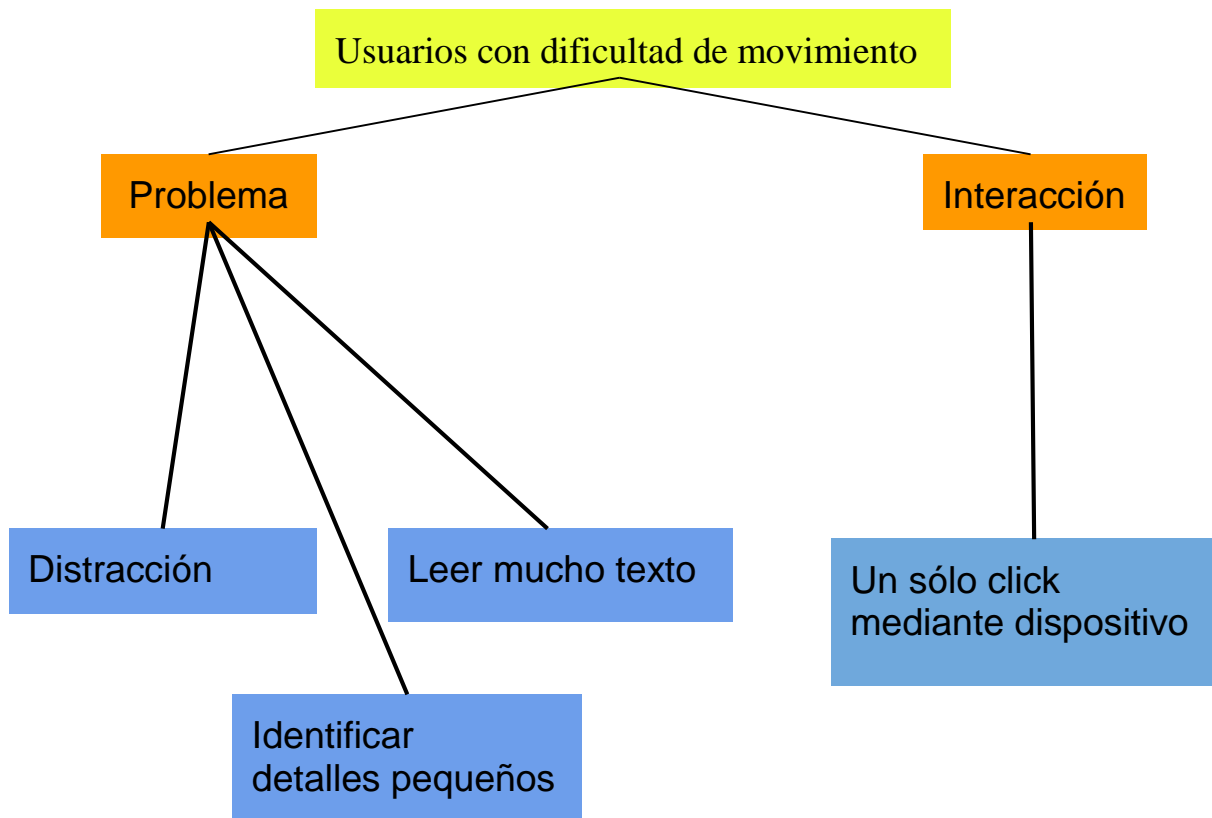
### 4.2.1 Identificación de las categorías de usuarios

Para la identificación vamos a seguir la estrategia de identificación por roles de usuario. En base a esto podemos definir 2 tipos de usuario:

1. Usuarios con dificultad de movimiento que serán los que usarán la aplicación para comunicarse con el exterior. A pesar de que puedan tener alguna limitación cognitiva, se espera que con el tiempo puedan conseguir un manejo más o menos ágil de la aplicación y la integren como una herramienta para su día a día. Las tareas que realizarán serán las de interactuar con el producto en modo usuario y nunca llegar a configurar nada en el modo administrador. Su interacción debe ser muy sencilla y simple.
2. Usuarios administradores que se encargan de configurar la aplicación. Suelen ser personas muy cercanas al usuario con dificultad de movimiento que deberán adaptar el producto a los gustos y necesidades del usuario y también interactuar con el producto para lograr una comunicación completa entre los dos usuarios. Aquí podemos distinguir dos tipos:
  - a. Usuarios esporádicos: usan la aplicación para una comunicación puntual con el usuario y sólo se encargan de interactuar con la aplicación cuando sea necesario.
  - b. Usuarios permanentes: usuarios muy cercanos al otro usuario, como pueden ser familiares, que permanecen día a día con él ajustando la aplicación a sus gustos según se vayan necesitando e interactúan con la aplicación más a conciencia que los usuarios esporádicos. Se espera que alcancen un grado alto de familiaridad con la aplicación con el paso del tiempo.

## 4.2.2 Procesar los datos

Para procesar los datos hemos organizado los datos obtenidos alrededor de las categorías de usuario.



### 4.2.3 Identificación y creación de esqueletos

Tras analizar los diagramas desarrollados en el apartado anterior, podemos verificar las categorías de usuarios indicadas anteriormente.

En el caso del diagrama de afinidad del usuario administrador, podemos verificar que existen usuarios que usan la aplicación habitualmente y están más cerca del usuario y otros que sólo interactúan puntualmente, pudiendo dividirlos en dos subcategorías:

Usuarios administradores que están en el día a día con el usuario y la aplicación.

Esqueleto:

<b>Usuario que trata todos los días con el usuario de movilidad reducida</b> Edad: 20-60 años
<b>Formación:</b> <ul style="list-style-type: none"><li>- Conocimiento de las limitaciones y problemas de los usuarios con movilidad reducida.</li><li>- Ligera experiencia en sistemas anteriores</li></ul>
<b>Nivel tecnológico:</b> <ul style="list-style-type: none"><li>- Manejo nivel usuario de ordenador, móvil o tablet.</li></ul>
<b>Información adicional:</b> <ul style="list-style-type: none"><li>- Larga experiencia en el trato con personas con movilidad reducida.</li><li>- Alto conocimiento del otro usuario, mediante un grado de parentesco o de amistad.</li><li>- Ganas de aprender y de mejorar la vida del otro usuario.</li></ul>

Usuarios administradores que usan la aplicación para comunicación puntual.

Esqueleto:

<b>Usuario ocasional para comunicación puntual</b> Edad: 20-60 años
<b>Formación:</b> <ul style="list-style-type: none"><li>- Ligera idea de las limitaciones y problemas de los usuarios con movilidad reducida.</li></ul>
<b>Nivel tecnológico:</b> <ul style="list-style-type: none"><li>- Manejo nivel usuario de ordenador, móvil o tablet.</li></ul>
<b>Información adicional:</b> <ul style="list-style-type: none"><li>- Es probable que no tenga ningún lazo emocional con el usuario a interactuar.</li></ul>

Usuarios de movilidad reducida que usará el producto para comunicarse.

Esqueleto:

<p><b>Usuario que trata todos los días con el usuario de movilidad reducida</b> Edad: 12-40 años</p>
<p><b>Formación:</b></p> <ul style="list-style-type: none"> <li>- Capacidad rápida de reacción ante gustos y decisiones.</li> <li>- Problemas para leer letra pequeña y textos demasiado grandes.</li> </ul>
<p><b>Nivel tecnológico:</b></p> <ul style="list-style-type: none"> <li>- Capacidad de interacción con el sistema mediante un clic.</li> </ul>
<p><b>Información adicional:</b></p> <ul style="list-style-type: none"> <li>- Posible rápida frustración ante fallo en el producto.</li> <li>- Incapacidad para comunicarse mediante otro sistema que no sea el producto a desarrollar.</li> </ul>

## 4.2.4 Priorización de esqueletos

En este apartado vamos a priorizar los esqueletos anteriormente para decidir qué esqueletos de personas van a convertirse realmente en personas.

Tanto en función de la importancia subjetiva que tendrán en el sistema como en la frecuencia de uso, los esqueletos principales serían los del usuario con dificultad de movimiento y usuarios administradores de uso a diario. Es importante que el diseño de la aplicación para estos dos usuarios vaya de la mano ya que van a interactuar juntos casi todo el tiempo y el futuro de la misma depende de que se entiendan y puedan adaptarse rápidamente. En un segundo plano quedaría el usuario administrador de uso puntual.

Teniendo en cuenta todos los factores anteriormente mencionados, concluimos que las personas primarias son:

- usuarios con dificultad de movimiento.
- usuarios administradores de uso diario.

Las personas secundarias son:

- usuarios administradores de uso puntual.

## 4.2.5 Desarrollo de personas

A continuación mostramos las personas desarrolladas a partir de los esqueletos antes definidos:

### **Teresa López**

Edad: 45 años

#### **Breve descripción**

Madre de Oscar, trabaja en una gestoría pero desde que nació trabaja desde casa para así poder estar más pendiente de él. Es una madre muy atenta y comprensiva, acompaña a su hijo siempre a todos sitios junto a su marido. Han sufrido un cambio de vida radical y aunque ya se han acostumbrado aún les supone mucho esfuerzo entender que quiere su hijo, incluso en cosas muy sencillas como qué película quiere ver.

#### **Objetivos, deseos, motivaciones**

Teresa siempre disfruta de la compañía de su hijo, pero desearía que se pudieran comunicar mejor ya que ahorraría mucho tiempo y él podría ser algo más autónomo. Cualquier mejora en la vida de su hijo sería bienvenida.

#### **Conocimientos y habilidades**

Manejo de ordenadores debido a su trabajo, sobre todo programas de ofimática. Después de 13 años amplia conocedora de las limitaciones que sufren las personas con dificultad de movimiento.

#### **Contexto/entorno de la persona**

Junto a su marido se pasan el día pendientes de su hijo, están familiarizados con sus gustos, comportamientos o estados de ánimo.

### **Oscar Ramirez**

Edad: 13 años

#### **Breve descripción**

Oscar es un adolescente que nació con dificultad de movimiento lo que le impide comunicarse con la gente que le rodea. También tiene problemas para leer grandes textos con dificultad y sobre todo con letras pequeñas. Comprende casi todo lo que pasa a su alrededor. Le encanta pasar las tardes viendo películas junto a sus padres, aunque sabe que a veces tiene que ir a rehabilitación.

#### **Objetivos, deseos, motivaciones**

Poder comunicarse y decir lo que quiere en cada momento. Poder decir cuando tiene hambre, a donde le apetece salir o poder elegir la siguiente película a ver ya que es un apasionado.

#### **Conocimientos y habilidades**

Dispone de un joystick con el que puede simular un click al ordenador. Tiene buenos reflejos.

### **Contexto/entorno de la persona**

Se pasa el día en casa con sus padres donde le enseñan como en la escuela. Algunos días va al centro de rehabilitación donde le espera Fernando.

### **Fernando Llull**

Edad: 30 años

### **Breve descripción**

Trabajador del centro de rehabilitación donde acude Oscar todos los martes y jueves. Estudió fisioterapia en la Universidad y le encanta su trabajo. Lleva poco tiempo tratando a Oscar ya que es nuevo en el puesto aunque siempre disfruta de lo que se esfuerza en los ejercicios.

### **Objetivos, deseos, motivaciones**

Poder comunicarse con Oscar para que le ayude en la realización de los ejercicios y para poder hacer más amena la estancia. Cree que la comunicación entre ellos tiene que ser fluida para que Oscar le vaya indicando si van por el buen camino.

### **Conocimientos y habilidades**

Manejo de dispositivos tecnológicos a nivel usuario.

Conocimiento de fisioterapia para tratar a Oscar.

### **Contexto/entorno de la persona**

Se pasa el día en el centro atendiendo a distintos niños, la parte mala es que cada uno tiene unas condiciones especiales y le resulta difícil acordarse de todas.

## **4.3 Requisitos**

Aquí se trata el proceso de definición de requisitos a partir de las personas creadas anteriormente.

### **4.3.1 Enunciado de problemas y visiones**

**Problema 1:** Oscar sólo dispone de un joystick para poder usar tecnologías. Pero ésto no es suficiente con los sistemas actuales y siente frustración al no poder expresar lo que siente.

Enunciado: nuestro sistema (web o aplicación) debe ser capaz de funcionar sólo mediante un clic.

**Problema 2:** Llega la hora de la comida y Teresa dispone de varias opciones pero le gustaría saber la opinión de su hijo, aunque no consiguen entenderse y se vuelve una tarea muy difícil.

Enunciado: con nuestro sistema Teresa será capaz de poder hacer preguntas cortas ofreciendo varias posibles respuestas a Oscar y que éste indique cual prefiere.

**Problema 3:** Llega la tarde al día de Oscar y Teresa y hoy es día de ver película. Como una de las pasiones de Oscar disponen de una gran cantidad de ellas, pero Teresa no logra averiguar cuál quiere Oscar y tiene que ponerlas una a una hasta acertar.

Enunciado: con nuestro Sistema Óscar podrá elegir qué película ver de todas las que disponen y además Teresa podrá ir añadiendo las películas nuevas que vayan comprando.

**Problema 4:** Oscar tiene ganas de expresar algo, por ejemplo que ya quiere cenar, pero no tiene forma de hacerlo. Lo único que consigue es alarmar a sus padres y conseguir aún más frustración.

Enunciado: nuestro sistema dispondrá de un teclado interactivo manejado por un clic para que Oscar pueda comunicar frases sin necesidad de que le den opciones para ello.

**Problema 5:** Teresa y Oscar ya han probado algún sistema para poder comunicarse entre ellos pero no con muy buenos resultados. No consiguen acostumbrarse al sistema y sobre todo a su velocidad.

Enunciado: nuestro sistema será configurable por parte del administrador (Teresa en este caso) y podrá adaptar los parámetros a Oscar.

**Problema 6:** Fernando recibe a Oscar en el centro encantado todos los días, pero le gustaría poder recibir un “feedback” por parte del propio Oscar para conseguir saber si los ejercicios que están siguiendo son de su agrado o simplemente le molesta alguno y poder cambiarlo.

Enunciado: con nuestro sistema incluso alguien que no sea usuario habitual como Fernando será capaz de acostumbrarse rápidamente y así poder comunicarse con Oscar.

## 4.3.2 Identificación de las expectativas

### Expectativas de personas primarias

- Teresa desearía una forma para mejorar la vida de su hijo. Pero también para poder mejorar algo la suya propia y hacer la convivencia más llevadera entre todos. Es muy frustrante no poder entender qué necesidades requiere Oscar en cada momento debido a que no pueden comunicarse. Tareas tan sencillas como elegir una película pueden convertirse en una odisea. Pierden una cantidad de tiempo muy grande en entenderse y no ayuda a una forma fácil de vivir. Con otros sistemas que han probado o no terminaban de acostumbrarse o Oscar no prestaba atención debido a que la interfaz no le llamaba la atención.
- Oscar desearía poder comunicarse con la gente que le rodea. Poder contestar a las preguntas que le hacen, poder dar su opinión o expresar algo cuando

le apetezca. Su madre se encarga de que no le falten películas pero le gustaría decir cada día cual le apetece o qué comida prefiere para hoy.

#### **Expectativas de personas secundarias**

- Fernando desearía poder comunicarse con Oscar cuando están juntos. Atiende a varias personas a lo largo del día, cada una con unas necesidades distintas, y le resulta muy difícil conseguir alguna respuesta por parte de Oscar para saber cómo le están afectando los ejercicios que están haciendo. Desearía algún sistema fácil de usar para comunicaciones puntuales con el propio Oscar y que ambos puedan empatizar más el uno con el otro ya que pasan bastantes tiempos juntos.

### **4.3.3 Identificación de requisitos**

#### **Requisitos de usuarios que usan el sistema en modo usuario**

- Sistema capaz de funcionar mediante un único clic
- Disponer de una categoría con preguntas y respuestas cortas.
- Disponer de un teclado interactivo para uso puntual.
- Disponer de un apartado multimedia (películas normalmente) para poder seleccionar distintas opciones.
- Interfaz clara y sencilla, dando toda la importancia a las posibles opciones a elegir.
- Letras grandes y mensajes cortos para evitar aburrimiento.
- Colores vivos y con gran contraste entre unos y otros para evitar perder la concentración.
- Posibilidad de incluir elementos que llamen la atención del usuario como imágenes, sonidos etc.

#### **Requisitos de usuarios que usan el sistema en modo administrador**

- Disponible un modo administrador para configurar en el modo usuario parámetros como la velocidad o el número de respuestas.
- Capacidad de guardar elementos en éste modo administrador, por ejemplo guardar películas en el apartado multimedia.
- Éste modo administrador debe ser fácil de entender y usar ya que no se tienen porque tener altos conocimientos informáticos por parte del usuario.
- El sistema debe ser intuitivo y rápido de aprender para usuarios que sólo lo usen para comunicaciones puntuales.

## **5. Metodología de programación**

El pilar que hemos utilizado como base para realizar este proyecto ha sido el lenguaje de programación Java. La mayor parte de los elementos de programación de los que hemos hecho uso han sido impartidos durante la carrera.

A continuación, se explican los distintos lenguajes, modelos y librerías que hemos utilizado:

## 5.1 Lenguajes de programación

Un lenguaje de programación es un lenguaje formal diseñado para realizar procesos que pueden ser llevados a cabo por máquinas, en este caso, ordenadores.

Los lenguajes de programación que se han utilizado son:

- Java
- HTML
- JavaScript
- CSS

### 5.1.1 Java



## **Introducción**

**Java** es un lenguaje de programación orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo, lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra.

## **Entornos de funcionamiento**

**Java** es uno de los lenguajes con un mayor crecimiento y amplitud de uso en distintos ámbitos de la industria de la informática. Se utiliza en:

- Dispositivos móviles: desde que se creó la especificación J2ME (Java 2 Platform, Micro Edition).
- Navegador web: desde los inicios, existe la opción de desarrollar pequeñas aplicaciones (Applets) que luego pueden ser incrustadas en una página HTML.
- Servidores: popular con la utilización de Servlets y JSP (Java Server Pages).
- Aplicaciones de escritorio: el entorno de ejecución Java (JRE) se ha convertido en un componente habitual en los sistemas operativos más usados del mundo.

## **Ventajas**

- Lenguaje multi-plataforma
- Manejo automático de la memoria
- Gran cantidad de recursos disponibles
- Reutilización de código

## Desventajas

- Su principal objetivo no es el rendimiento
- Difícil de aprender sin unas nociones programación orientada a objetos

## Aplicación en nuestro proyecto

Prácticamente todo el proyecto está realizado en Java.

Se ha utilizado el modelo ‘Modelo-Vista-Controlador’, el cual se explicará en un punto posterior, queriendo tener bien estructuradas y diferenciadas las distintas partes del proyecto.

Empezando por la parte del ‘Modelo’, a continuación se muestra una breve parte del mismo:

```

public void notifyCambioOpcion(int op) {
    // TODO Auto-generated method stub
    for (Observer o : observadores){
        o.onCambioOpcion(op);
    }
}

public void notifyCambioModo(boolean m) {
    // TODO Auto-generated method stub
    for (Observer o : observadores){
        o.onCambioModo(m);
    }
}

public void notifyCambioFrecuencia(int f) {
    // TODO Auto-generated method stub
    for (Observer o : observadores){
        o.onCambioFrecuencia(f);
    }
}

public boolean getModo() {
    // TODO Auto-generated method stub
    return this.modo;
}

public void setModo(boolean b) {
    // TODO Auto-generated method stub
    this.modo=b;
}

public int getFrecuencia(){
    return this.frecuencia;
}

```

El modelo se encarga de gestionar todos los accesos a la información con la que el sistema opera, y esta información, es enviada a través de los observadores a la vista, para que aparezca en la parte visible de la aplicación.

Como se puede ver, se han utilizado observadores, para que cuando uno de los objetos cambia su estado, notifique ese cambio a todas las clases dependientes del mismo (clases que están en la vista).

Al igual, existen métodos como puede ser 'getFrecuencia', que devuelve el valor de la frecuencia del temporizador en ese momento, para que la vista pueda utilizar ese valor que anteriormente tiene que haberle pedido.

En cuanto a la parte del controlador:

```
public class Controller {  
  
    private Data d;  
    private TratarXML xml;  
    private int velocidad = 1;  
    public Controller(Data d2) {  
        // TODO Auto-generated constructor stub  
        d = d2;  
    }  
  
    public boolean addObserver(Observer po) {  
        return d.addObserver(po);  
    }  
  
    public boolean removeObserver(Observer po){  
        return d.removeObserver(po);  
    }  
  
    public void onCambioOpcion(int op){  
        d.notifyCambioOpcion(op);  
    }  
  
    public void onCambioModo(boolean m) {  
        // TODO Auto-generated method stub  
        d.notifyCambioModo(m);  
    }  
}
```

El controlador responde a eventos e invoca peticiones al modelo cuando se hace alguna solicitud para conocer algún tipo de información.

Como se puede observar en la imagen, se tiene un atributo Data que corresponde al 'Modelo', al cual llama a sus métodos desde el controlador (peticiones), cuando desde la vista se requiere información o ha cambiado algo que hay que notificar.

Por último, en cuanto a la vista:

```

public ToolbarSup(Controller controlador, final int op, JFrame padre){
    this.c = controlador;
    //this.c.addPadre(padre);
    this.setBackground(new Color(211, 211, 211));
    this.setLayout(new GridLayout(1,4));
    frec = new JTextField("Velocidad: "+c.getVelocidad()+"x");
    frec.setForeground(new Color(0, 0, 0));
    frec.setEditable(false);
    frec.setEnabled(false);
    frec.setFont(new Font("Roboto", Font.PLAIN, 20));
    frec.setDisabledTextColor(Color.BLACK);
    frec.setHorizontalAlignment(SwingConstants.CENTER);
    modo = new JButton("Cambiar de modo");
    modo.setFont(new Font("Roboto", Font.PLAIN, 20));
}

```

La ‘Vista’ se encarga de presentar el ‘Modelo’ en un formato adecuado para interactuar.

Como se puede apreciar en la imagen, se han utilizado muchas de las opciones que nos ofrece Java para poder ‘maquetar’ nuestra aplicación, desde la utilización de distintos Layouts hasta el uso de diferentes fuentes en nuestros textos.

Ha sido muy importante el uso de MouseListeners y KeyListeners para poder ir recogiendo todos los eventos que se producen. La aplicación está en continua interacción con el usuario, por lo que el uso de estos Listeners nos ha facilitado bastante el trabajo.

```

ml3 = new MouseAdapter() {

    public void mouseClicked(MouseEvent e) {
        // TODO Auto-generated method stub
        timer.stop();
        cerrarVentana();
        c.onCambioOpcion(0);
        @SuppressWarnings("unused")
        Pelis v = new Pelis(c);
    }
};

```

Viendo cómo se utiliza MVC, se puede observar la variable c, que corresponde al ‘Controlador’, con la cual se notifica el cambio de estado que se está produciendo en ese momento, y con el cual el ‘Controlador’ avisará al ‘Modelo’ de dicho cambio.

Por último, un elemento muy importante ha sido ‘Timer’. Con la ayuda de este objeto, se ha podido conseguir que la opción que se pueda seleccionar vaya cambiando cada ‘x’ frecuencia.

```

private void temporizador(){
    timer = new Timer (c.getFrecuencia(), new ActionListener ()
    {
        public void actionPerformed(ActionEvent e)
        {
            if(vs1 == true){
                btnOpcion_2.requestFocus();
                btnOpcion_1.setBackground(Color.RED);
                btnOpcion_2.setBackground(Color.GREEN);
                vs1 = false;
                vs2 = true;
            }
            else if(vs2 == true){
                btnOpcion_3.requestFocus();
                btnOpcion_2.setBackground(Color.RED);
                btnOpcion_3.setBackground(Color.GREEN);
                vs2 = false;
                vs3 = true;
            }
            else if(vs3 == true){
                btnOpcion_1.requestFocus();
                btnOpcion_3.setBackground(Color.RED);
                btnOpcion_1.setBackground(Color.GREEN);
                vs3 = false;
                vs1 = true;
            }
        }
    });
}

```

## 5.1.2 HTML



## **Introducción**

**HTML** (Hypertext Markup Language) es un lenguaje de marcado que define una estructura básica y un código para la definición de contenido de una página web. Este lenguaje basa su filosofía de desarrollo en la diferenciación. Para añadir un elemento externo a la página, se hace una referencia a la ubicación de dicho elemento mediante texto. De este modo, la página web contiene solamente texto mientras que recae en el navegador web la tarea de unir todos los elementos y visualizar la página final.

## **Ventajas**

- Lenguaje de fácil aprendizaje
- Archivos pequeños y despliegue rápido
- Texto presentado de forma estructurada
- Admitido en todos los exploradores

## **Desventajas**

- Lenguaje estático
- La interpretación de cada navegador puede ser diferente
- El diseño es muy lento
- Las etiquetas son muy limitadas

## **Aplicación en nuestro proyecto**

Nuestra aplicación web está desarrollada en HTML combinado con JavaScript y CSS.

Se ha utilizado HTML para que la página tenga una estructura bien definida.

El código está estructurado en 3 partes diferenciadas:

-Header: cabecera que se mantiene constante en todas las páginas, desde la cual se puede cambiar el modo, aumentar o disminuir la velocidad, activar o desactivar el sonido y acceder a una pequeña ayuda.

```

<nav class="navbar navbar-default" role="navigation">
  <!-- El logotipo y el icono que despliega el menú se agrupan
  para mostrarlos mejor en los dispositivos móviles -->
  <div class="navbar-header">
    <button type="button" class="navbar-toggle" data-toggle="collapse"
    data-target=".navbar-ex1-collapse">
      <span class="sr-only">Desplegar navegación</span>
      <span class="icon-bar"></span>
      <span class="icon-bar"></span>
      <span class="icon-bar"></span>
    </button>
    <a class="navbar-brand" href="javascript:goHome()"><span class="glyphicon glyphicon-home"></span> SW-Communication</a>
  </div>

  <!-- Agrupar los enlaces de navegación, los formularios y cualquier
  otro elemento que se pueda ocultar al minimizar la barra -->
  <div class="collapse navbar-collapse navbar-ex1-collapse">
    <ul class="nav navbar-nav">
      <li class="active" id="btnModo"><button type="button" class="btn btn-default navbar-btn" onclick="cambiarModo()" id="btnModo">
      <p class="navbar-text" id="speedText" value=2>Velocidad: 2s</p>
      <li><button type="button" class="btn btn-default navbar-btn" onclick="aumentarVelocidad()"></button></li>
      <li><button type="button" class="btn btn-default navbar-btn" onclick="disminuirVelocidad()"></button></li>
      <p class="navbar-text">Voz</p>
      <button type="button" class="btn btn-default navbar-btn" onclick="mute()" id="btnMute"><span class="glyphicon glyphicon-glyph">
    </ul>
    <ul class="nav navbar-nav navbar-right">
      <li id="liAyuda"><button type="button" class="btn btn-default navbar-btn" onclick="ayuda()" id="btnAyuda">Ayuda <span class="glyphicon glyphicon-user">
      <p class="navbar-text" id="user">Niño <span class="glyphicon glyphicon-user"></span></p>
    </ul>
  </div>
</nav>

```

-Body: ‘cuerpo’ del código. En este apartado, se encuentra todo el grosso de la aplicación, donde realmente se interactúa con la misma, y donde aparecen las distintas opciones con las que poder jugar.



```

<div class="jumbotron">
  <div class="container">
    <h2>Página Principal</h2>
    <p id="textoOp">Preguntas rápidas: menú de opciones configurables con el fin de solucionar preguntas con un interlocutor</p>
  </div>
</div>
<div class="container-fluid">
  <div class="row">
    <div class="col-md-1"></div>
    <div class="col-md-10">
      <div class="btn-group" id="opciones">
        <button type="button" class="btn btn-default" id="b1"><h2>Preguntas Rápidas <span class="glyphicon glyphicon-check"></span></button>
        <button type="button" class="btn btn-default" id="b2"><h2>Teclado Interactivo <span class="glyphicon glyphicon-console"></span>
      </div>
    </div>
  </div>
  <div class="row">
    <div class="col-md-3"></div>
    <div class="col-md-1 btn btn-default btn-lg" id="btnArrow1"><span class="glyphicon glyphicon-arrow-up"></span></div>
    <div class="col-md-5"></div>
    <div class="col-md-1 btn btn-default btn-lg" id="btnArrow2"><span class="glyphicon glyphicon-arrow-up"></span></div>
  </div>
  <div id="textoAbajo" style="display:none">Para descargar nuestra aplicación en su ordenador: <a href="" download="Sw-Communication">Pulse
</div>
</div>

```

-Footer: pie de página. Al igual que la cabecera, se mantiene común en todas las páginas. En esta sección, se tiene la opción de descargar el .jar para poder usar la aplicación (más completa) en su ordenador.

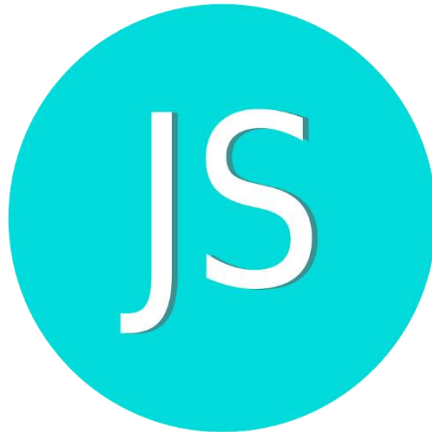
Para descargar nuestra aplicación en su ordenador: [Pulse aquí](#)

```

<div class="modal-footer">
  <div id="textoAbajo" style="display:none">Para descargar nuestra aplicación en su ordenador: <a href="" download="Sw-Communication">Pu
  <button type="button" class="btn btn-default" data-dismiss="modal">Cerrar</button>
</div>

```

## 5.1.3 JavaScript



### **Introducción**

**JavaScript** es un lenguaje de programación interpretado. Se define como orientado a objetos basado en prototipos. Se utiliza principalmente en el lado del cliente, permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, aunque existe una forma de JavaScript del lado del servidor.

### **Ventajas**

- Velocidad muy rápida al actuar en el lado del cliente
- Simplicidad y fácil aprendizaje
- Versatilidad encajando perfectamente con otros lenguajes
- Reduce la carga en el servidor

### **Desventajas**

- Baja seguridad al ejecutarse en el lado del cliente
- Distintas interpretaciones dependiendo del navegador

### **Aplicación en nuestro proyecto**

JavaScript se ha utilizado para que nuestra página web fuera una página web dinámica. Mediante JavaScript, hemos llevado a cabo el control de los distintos eventos que ocurren.

En todas las páginas, se dispone de un timer para ir cambiando la selección según pasa el tiempo.

Todos los eventos del Toolbar, como aumentar o disminuir la velocidad, cambio de modo, habilitar o deshabilitar el sonido... están controlados por JavaScript.

```
function cambiarModo(){
//primero va el click del boton y luego la función del documento
  if(mode == true){
    ini = false;
    document.getElementById("user").innerHTML = "Madre <span class='glyphicon glyphicon glyphicon-user'></span>";
    document.getElementById("btnAyuda").style = "background-color:#eee; border-color:#eee;";
    document.getElementById("liAyuda").style = "background-color:#eee; border-color:#eee;";
    document.getElementById("textoAbajo").style = "display:block";
    mode = false;
  }
  else{
    document.getElementById("textoAbajo").style = "display:none";
    document.getElementById("user").innerHTML = "Niño <span class='glyphicon glyphicon glyphicon-user'></span>";
    document.getElementById("btnAyuda").style = "background-color:#f8f8f8; border-color:#f8f8f8;";
    document.getElementById("liAyuda").style = "background-color:#f8f8f8; border-color:#f8f8f8;";
    mode = true;
  }
}
```

Tanto en la opción de preguntas rápidas como en la del teclado interactivo, la funcionalidad y los cambios de estado de cada una de ellas, se verifican con JavaScript.

```
$(document).click(function(){
  //alert("has pulsado en botn");
  if(mode == true && ini == true){
    if(respondiendo == false){
      if(op1 == true){
        var bla = $('#r1').val();
        respondiendo = true;
        document.getElementById("modalText1").innerHTML = "Has seleccionado: "+bla;
        $('#modal1').modal('show');
        if(responsiveVoice.voiceSupport()) {
          if(voz == true){
            var aux = "Has seleccionado ";
            bla = aux.concat(bla);
            responsiveVoice.speak(bla);
          }
        }
      }
    }
  }
}
```

## 5.1.4 CSS



### **Introducción**

CSS (Cascading Style Sheet) es un lenguaje de diseño gráfico usado en este caso, para establecer el diseño visual de la página web. Está diseñado principalmente para marcar la separación del contenido del documento y la forma de presentación de este.

### **Ventajas**

- Mayor control de la presentación del sitio
- Código HTML mucho más legible
- Aumento de la accesibilidad

### **Desventajas**

- Limitaciones en cuanto a distintos recursos
- Diferencias dependiendo del navegador

### **Aplicación en nuestro proyecto**

Para separar la parte del contenido web (HTML) y su presentación, hemos creado una hoja de estilos, además de utilizar Bootstrap, el cual se explica más adelante. En esta hoja de estilos se han definido distintas identidades y clases, las cuales se llaman desde la parte HTML, para poder dar formato y diseño de manera más estructurada.

Para poder hacer uso de ello, hemos tenido que incluir una referencia al archivo .css desde el archivo .html.

```
<link rel="stylesheet" type="text/css" href="includes/css/style.css">
```

Un pequeño ejemplo de uso, en la opción 1, desde el modo madre se puede definir la pregunta que se quiere hacer. Para que esta pregunta sea bien visible, se le ha dado unos formatos de letra específicos.

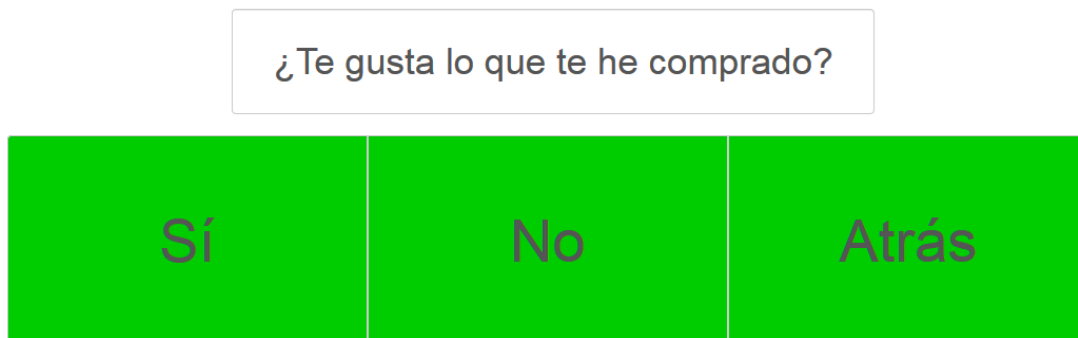
Desde el .html, se llama al 'id' pregunta que se encuentra en la hoja de estilos.

```
<div class="col-md-6">  
  <input type="text" class="form-control" id="pregunta" placeholder="Titulo de la pregunta (vacío)" value="">  
</div>
```

Desde el .css, definimos el identificador pregunta.

```
#pregunta{  
text-align: center;  
font-size: 250%;  
min-height:100px;  
max-height:100px;  
}
```

Y el resultado es el siguiente:

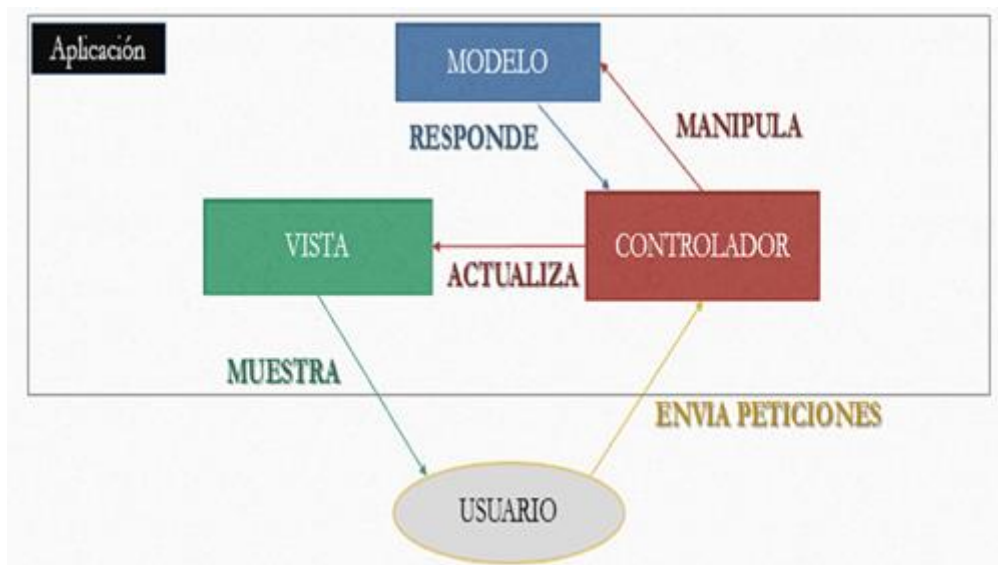


¿Te gusta lo que te he comprado?

Sí No Atrás

## 5.2 Modelos

### 5.2.1. Modelo-Vista-Controlador (MVC)



#### Introducción

El Modelo-Vista-Controlador (MVC) es un patrón de arquitectura software que separa los datos y la lógica de negocio de una aplicación, de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones.

Para ello se utilizan 3 componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado se definen las componentes para la representación de la información, y por otro lado las componentes para la interacción con el usuario.

El 'Modelo' es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso.

El 'Controlador' responde a eventos, normalmente acciones del usuario, e invoca peticiones al 'Modelo' cuando se hace alguna solicitud sobre la información.

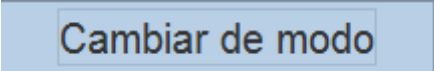
La 'Vista' presenta el 'Modelo' en un formato adecuado para interactuar con el usuario.

## Aplicación en nuestro proyecto

Como se ha explicado brevemente antes, en el apartado de Java, para llevar a cabo nuestra aplicación se ha utilizado este modelo.

A continuación, se explica la interacción de los componentes, ilustrándolo con un ejemplo real de la aplicación:

1. El usuario interactúa con la interfaz de alguna forma, por ejemplo, pulsando el botón de cambiar modo



Cambiar de modo

2. En dicho botón, tenemos asociado un ActionListener, para saber cuándo el usuario le ha pulsado, e informar al controlador de tal evento.

```
modo.addActionListener(new ActionListener() {  
  
    public void actionPerformed(ActionEvent arg0) {  
        c.onCambioModo(c.getModo());  
    }  
});
```

3. El controlador recibe la notificación de la acción solicitada por el usuario, y accede al modelo (variable d), actualizándolo.

```
public void onCambioModo(boolean m) {  
    // TODO Auto-generated method stub  
    d.notifyCambioModo(m);  
}
```

4. El modelo, a través de los observadores, informa de este cambio a todas las clases de las vistas que implementen dichos observadores.

```
public void notifyCambioModo(boolean m) {  
    // TODO Auto-generated method stub  
    for (Observer o : observadores) {  
        o.onCambioModo(m);  
    }  
}
```

5. Por último, la vista a través de ese observador (onCambioModo), actualiza, si fuera necesario, la interfaz correspondiente.

```

public void onCambioModo(boolean m) {
    // TODO Auto-generated method stub
    if(m == false){
        modo.setText("Cambiar a Modo Niño");
        this.frec.setEnabled(true);
    } else {
        modo.setText("Cambiar a Modo Madre");
        this.frec.setEnabled(false);
    }
}
}

```

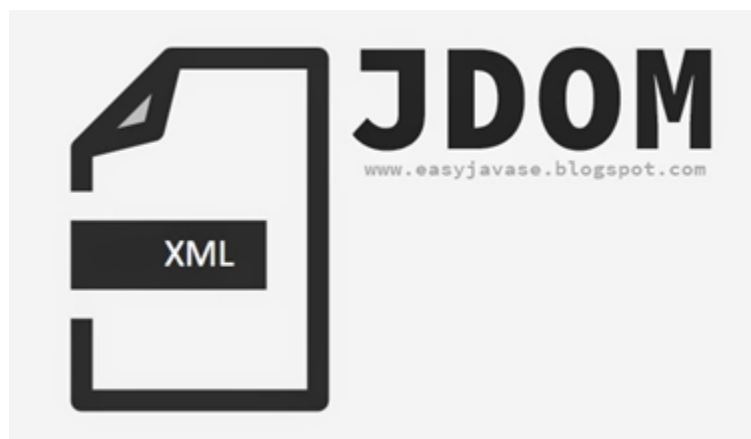
## 5.3 Librerías

Una **librería** se puede entender como un conjunto de clases, que poseen una serie de métodos y atributos. El principal objetivo de estas librerías es el de facilitar muchas operaciones.

Las librerías destacadas que hemos utilizado son:

- JDom
- FreeTTS
- Bootstrap
- jQuery
- Responsive Voice
- Font Awesome

### 4.3.1 jDom



### Introducción

JDom es una librería de código abierto para manipulaciones de datos XML optimizados para Java.

Es una biblioteca específica para usarse con Java, de manera que se beneficia de sus características.

### **Aplicación en nuestro proyecto**

Esta librería se ha utilizado para poder implementar y mantener una base de datos XML en la opción de multimedia.

En dicha opción, se almacena el título y una imagen de películas, libros etc... de tal manera que se muestran en un principio los 7 elementos que se han añadido más recientemente. Existe una opción para poder listar los elementos, en la que aparecen todos los títulos de los mismos, y se pueden elegir hasta un máximo de 7 para que sean mostrados.

```
public static HashMap<Integer,ArrayList<String>> cargarXml()
{
    //Se crea un SAXBuilder para poder parsear el archivo
    results = new HashMap<Integer,ArrayList<String>>();
    SAXBuilder builder = new SAXBuilder();
    File xmlFile = new File( "tv.xml" );
    try
    {
        if(xmlFile.exists()){

            //Se crea el documento a traves del archivo
            Document document = (Document) builder.build( xmlFile );

            //Se obtiene la raiz 'peliculas'
            Element rootNode = document.getRootElement();

            //Se obtiene la lista de hijos de la raiz 'peliculas'
            List<Element> list = rootNode.getChildren( "peli" );

            //Se recorre la lista de hijos de 'tables'
            for ( int i = 0; i < list.size(); i++ )
            {
                //Se obtiene el elemento 'tabla'
                Element peli = (Element) list.get(i);

                //Se obtiene el atributo 'nombre' que esta en el tag 'tabla'
                String nombreTabla = peli.getAttributeValue("id");
                System.out.println( "Tabla: " + nombreTabla );
            }
        }
    }
}
```

Para utilizar esta librería, el código de programación es algo más farragoso que lo que conocemos por Java 'básico'. Se ha adjuntado un ejemplo, donde aparece la función cargarXml, que recupera los datos del archivo XML para poder utilizar la información a la hora de cargar los elementos multimedia. Si este archivo no existe, se crea uno nuevo automáticamente.

Al igual que para cargar el XML, existe otra función con la que se puede añadir nuevos elementos multimedia a dicho archivo.

Estas funciones han sido programadas por los componentes del grupo, pero gracias a esta librería, ha sido menos costoso, ya que ofrece numerosas alternativas para trabajar con documentos XML.

## 5.3.2 FreeTTS

### Introducción

**FreeTTS** es un sintetizador de voz digital open source escrito en Java.

Es una librería que permite convertir textos a voz dinámicamente, actualmente solo disponible la voz americana.

### Aplicación en nuestro proyecto

Se ha realizado una extensa investigación sobre como reproducir textos de forma dinámica en castellano, pero para Java no se ha logrado encontrar nada.

Esta librería, FreeTTS, es una librería de software libre con una licencia MIT. Consiste en voces ya grabadas anteriormente, y la única que estaba disponible para utilizarla dinámicamente es la americana, las demás actualmente están en proceso como bien indican en su página web.

Para poder usarla en nuestro proyecto, solo es necesario llamar a la librería y crear un objeto de ella, pasándole el string que se quiere reproducir por parámetro.

Una vez creado el objeto, con utilizar el método Reproducir() de la propia librería, ya reproduce el sonido.

```
voz = new Voz("Welcome to software communication app");  
voz.Reproducir();
```

## 5.3.3 Bootstrap



## Introducción

**Bootstrap** es un framework de código abierto para diseño de páginas web. Contiene plantillas de diseño con tipografía, formularios, menús de navegación y otros elementos de diseño basados en HTML y CSS.

Su técnica más importante es la de ‘responsive design’ o diseño adaptativo.

## Aplicación en nuestro proyecto

El uso de Bootstrap es bastante importante, ya que adapta automáticamente la interfaz del sitio web al tamaño del dispositivo desde donde se visualice.

Ofrece diseños simples, limpios e intuitivos, lo que da agilidad a la hora de cargar y adaptarse a otros dispositivos, y trae varios elementos con estilos predefinidos fáciles de configurar, los cuales han sido usados en el proyecto.

Además de todo esto, es compatible con la mayoría de navegadores web.

```
<link href="includes/css/bootstrap.min.css" rel="stylesheet" media="screen">
```

Algún ejemplo de los elementos que se han usado:

```
<div class="collapse navbar-collapse navbar-ex1-collapse">
```

```
<div class="container-fluid">
```

## 5.3.4 jQuery



### Introducción

**jQuery** es una biblioteca multiplataforma de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, manejar eventos, desarrollar animaciones...

### Aplicación en nuestro proyecto

jQuery se ha utilizado para facilitar el código y para simplificarlo, ya que se podría decir que es una extensión de JavaScript mejorada.

Ofrece métodos más simplificados, y lo más importante, JavaScript tiene problemas con funcionalidades en algunos navegadores, mientras que jQuery consigue resolver esos problemas automáticamente.

```
<script type="text/javascript" src="includes/jquery/jquery-3.1.1.js"></script>
```

## 5.3.5 Responsive Voice



### Introducción

**Responsive Voice** es una biblioteca texto-voz de HTML5 diseñada para agregar funciones de voz a sitios web. Soporta hasta 50 idiomas con un total de 168 voces entre las que elegir.

### Aplicación en nuestro proyecto

En comparación con Java, en JavaScript sí que se han encontrado distintas librerías que se pueden utilizar para reproducir voces. Se ha elegido esta librería, ya que dispone de muchos idiomas y muchas voces entre las que elegir.

Para incluirla, solo se necesita hacer referencia al archivo externo.

```
<script src="http://code.responsivevoice.org/responsivevoice.js"></script>
```

Para poder utilizarla en el proyecto, es tan fácil como establecer que voz se quiere utilizar, y pasar al método `.speak()` la cadena de texto que se quiere reproducir.

```
if(responsiveVoice.voiceSupport()) {  
    responsiveVoice.setDefaultVoice("Spanish Female");  
}  
if(responsiveVoice.voiceSupport()) {  
    if(voz == true)  
        responsiveVoice.speak("Preguntas rápidas");  
}
```

## 6. Tecnologías usadas

## 6.1 Eclipse

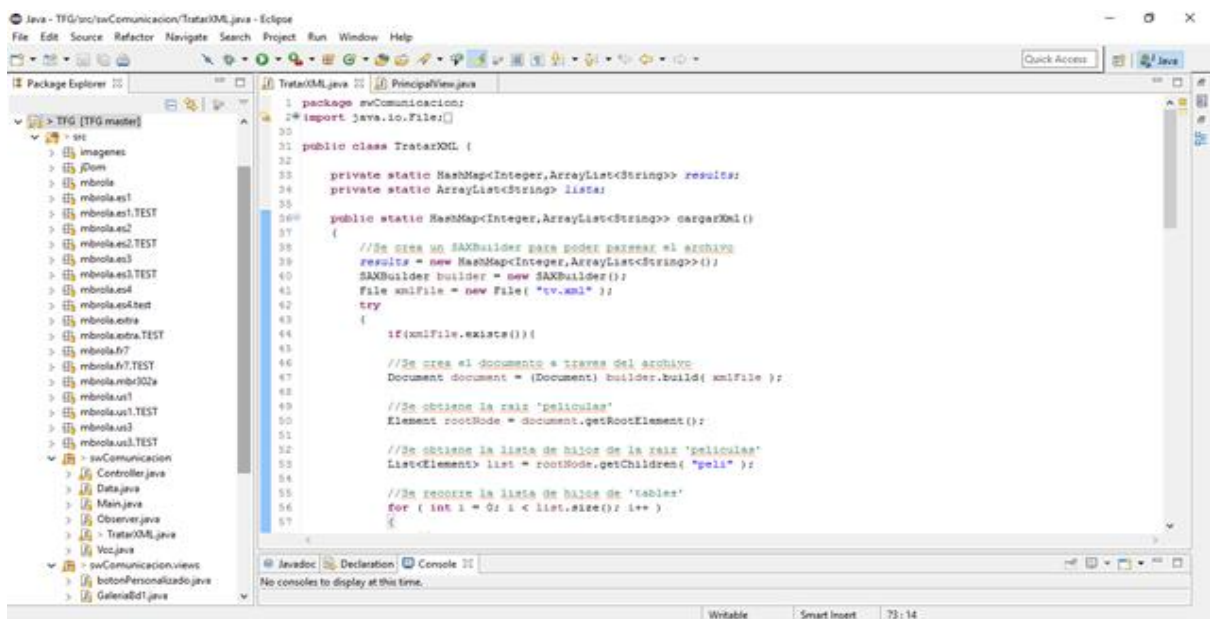


**Eclipse** es una plataforma de software compuesto por un conjunto de herramientas de programación de código abierto multiplataforma.

Eclipse dispone de un editor de texto con un analizador sintáctico y la compilación es en tiempo real.

Eclipse se ha utilizado para llevar a cabo la programación en Java. Además, ha sido muy útil su Windows Builder, para que la interfaz gráfica que se presenta al usuario esté mejor estructurada.

En la siguiente imagen, aparece parte de nuestro proyecto en Eclipse:



## 6.2 jsFiddle



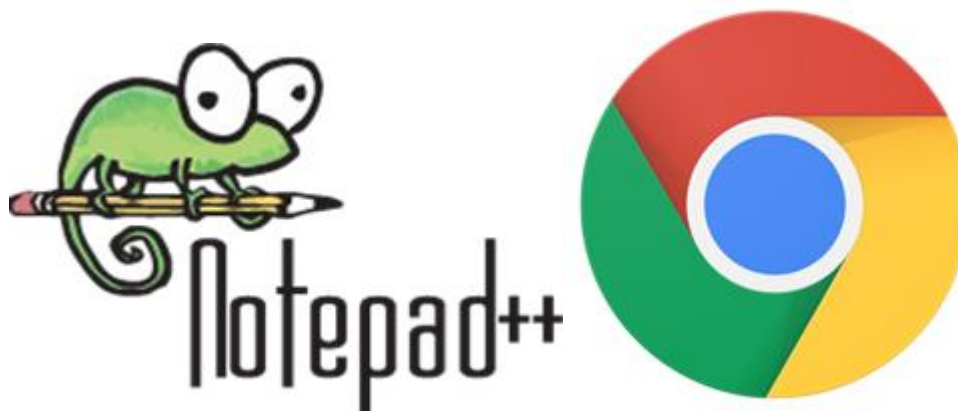
**jsFiddle** proviene de un entorno personalizado, basado en las selecciones de los usuarios, para probar código JavaScript, HTML y CSS dentro del navegador.

jsFiddle se ha utilizado para probar el código que se iba desarrollando en HTML, JavaScript y CSS de la web, como si estuviera ya corriendo en un navegador. Este código ha sido generado, o bien con Notepad++, o bien con Spring Tool Suite.

The screenshot shows the JSFiddle web interface in a browser. The address bar displays "https://jsfiddle.net". The interface includes a toolbar with "Run", "Save", "Tidy", and "Collaborate" buttons. On the left, there are sections for "Fiddle Meta", "External Resources" (with "jQuery.min.js" listed), "AJAX Requests", and "Legal, Credits and Links". The main area is a code editor with the following code:

```
1
2
3 <body onLoad="setInterval('seleccion()',3000);">
4 <script>
5     function seleccion() {
6         /*
7         Con esta funcion podremos ir cambiando la opción que se selecciona en verde
8         */
9         alert("Esta función se ejecuta cada 3 segundos");
10    }
11    $(document).click(function(){
12        /*
13        Desde esta función veremos que opción está seleccionada
14        */
15        alert("has pulsado en cualquier sitio del documento");
16    })
17 </script>
18 </body>
19
20
21
22
23
24
25
26
27
28
```

## 6.3 Notepad++ / Google Chrome



**Notepad++** es un editor de texto y de código fuente libre con soporte para varios lenguajes de programación. Incluye opciones más avanzadas que por ejemplo, el Bloc de notas, y soporta muchos lenguajes de programación.

**Google Chrome** es un navegador web gratuito desarrollado por Google y compilado con base en varios componentes e infraestructuras de desarrollo de aplicaciones de código abierto.

Notepad++ se ha utilizado para llevar a cabo el desarrollo web, complementándolo con el uso de Spring Tool Suite.

Para probar este código, después se ha probado en el navegador web Google Chrome, que es de lo más utilizados a día de hoy entre los navegadores.

```
C:\Users\Sergio\workspace\Act1\TFGWeb\home.html - Notepad++
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Macro  Ejecutar  Plugins  Ventana  ?
home.html  spoon1.html  style.css
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  <meta charset="UTF-8">
5  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
6  <meta http-equiv="X-UA-Compatible" content="IE=edge">
7  <meta name="viewport" content="width=device-width, initial-scale=1">
8  <link rel="shortcut icon" href="/includes/img/communication.png" />
9
10 <title>SW-Communication</title>
11
12 <!-- CSS de Bootstrap -->
13 <!--<link href="/includes/css/bootstrap.min.css" rel="stylesheet" media="screen">
14 <link rel="stylesheet" type="text/css" href="/includes/css/style.css">
15 <link rel="stylesheet" href="http://netdna.bootstrapcdn.com/font-awesome/3.1.1/css/font-awesome.css">
16 <script type="text/javascript" src="/includes/jquery/jquery-3.1.1.js"></script>-->
17 <!-- CSS de Bootstrap -->
18 <link href="/includes/css/bootstrap.min.css" rel="stylesheet" media="screen">
19 <link rel="stylesheet" type="text/css" href="/includes/css/style.css">
20 <script type="text/javascript" src="/includes/jquery/jquery-3.1.1.js"></script>
21 <script src="/includes/js/bootstrap.min.js"></script>
22 <script src="http://code.responsivevoice.org/responsivevoice.js"></script>
23 </head>
24 <body onload="setInterval('seleccioin()',1000);">
25 <nav class="navbar navbar-default" role="navigation">
26 <!-- El logotipo y el icono que despliega el menú se agrupan
27 para mostrarlos mejor en los dispositivos móviles -->
28 <div class="navbar-header">
29 <button type="button" class="navbar-toggle" data-toggle="collapse"
30 data-target=".navbar-ex1-collapse">
31 <span class="sr-only">Desplegar navegación</span>
32 <span class="icon-bar"></span>
33 <span class="icon-bar"></span>
34 <span class="icon-bar"></span>
35 </div>
36 </body>
37 </html>
Hyper Text Markup Language file  length: 11.142  lines: 269  Ln: 45  Col: 121  Sel: 0 | 0  Windows (CR LF)  UTF-8  INS
```



## 7. Almacenamiento de código

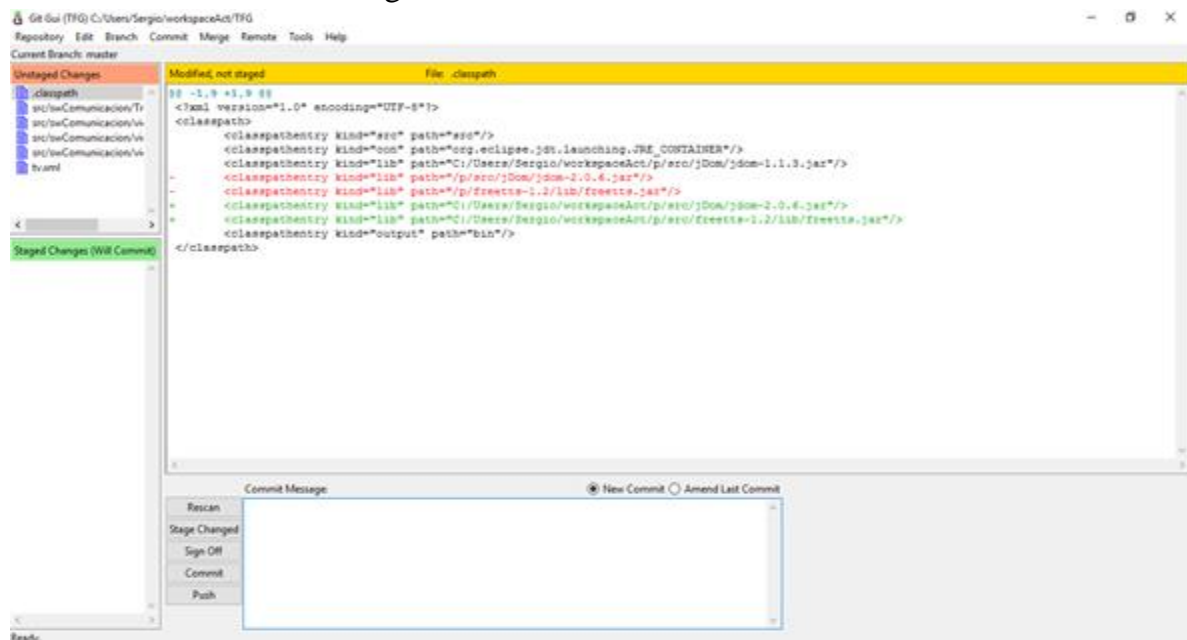
### 7.1 GitHub



**GitHub** es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git.

Se ha utilizado esta plataforma para almacenar el código Java de nuestra aplicación, por varias razones:

- A día de hoy, GitHub o LinkedIn suponen un segundo curriculum a la hora de buscar trabajo, por lo que tener proyectos en esta plataforma suponen un plus a la hora de ir a una entrevista.
- El control de versiones que ofrece GitHub es de los más buenos, eficientemente hablando.
- Se puede descargar una aplicación, que permite subir los archivos al GitHub sin tener que estar entrando a su página web, y sin que el proceso se haga excesivamente farragoso.



## 7.2 Google Drive

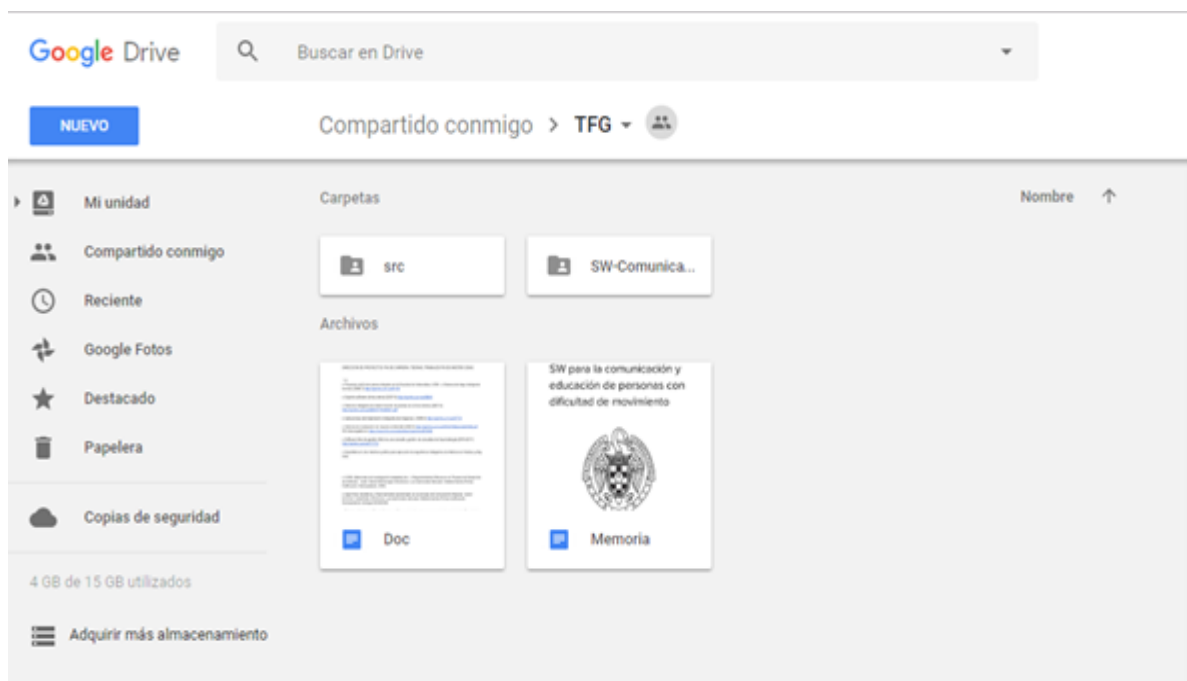


**Google Drive** es un servicio de alojamiento de archivos ofrecido por Google.

A parte de utilizar el servicio que ofrecía GitHub, se necesitaba almacenar otro tipo de información acerca del proyecto, como por ejemplo:

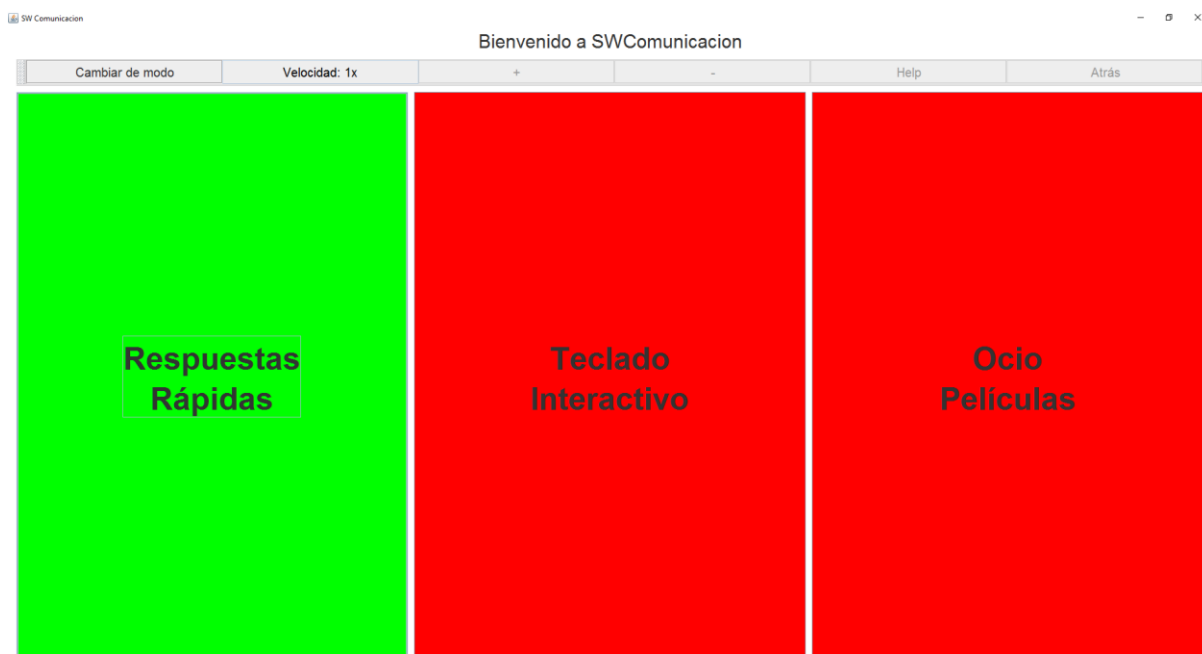
- Documentación
- Reparto de tareas
- Memoria
- Copia de seguridad de la aplicación

Para ello, se ha utilizado esta plataforma, la cual ofrece hasta 15GB de espacio gratuito por usuario, y en la que incluso se pueden realizar reuniones entre los miembros del grupo y modificaciones en línea de documentos.



## 8. Descripción del programa

### 8.1 Versión escritorio



A continuación se explica las diferentes funcionalidades y vistas de la aplicación, se compone de un .jar ejecutable, sin ninguna instalación. Al abrir la aplicación, nos aparecerá una pantalla como la siguiente:

Aparecen tres opciones, con fondo en colores fuertes, para que en caso de dificultad visual, se pueden distinguir perfectamente. Vemos en color verde la opción que en este momento está seleccionada, tras unos segundos esta selección cambiará a la siguiente opción. Cuando el usuario quiera acceder a alguna opción,

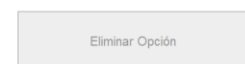
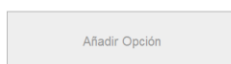
deberá esperar hasta que esta se encuentre seleccionada (en verde) y con cualquier acción del teclado o ratón, podrá acceder a ella. (Cualquier pulsación de tecla o click).

Las tres opciones principales son:

**Respuestas rápidas**, en esta opción el usuario puede encontrar, respuestas del tipo “SI” o “No” para contestar a preguntas que le planteen del exterior. En todo momento observamos una barra superior, esta contiene varias opciones que solo se desbloquean si nos encontramos en el modo administrador. Este modo se activa, pulsando sobre el primer botón superior “Cambiar de modo”, se activaran los botones superiores, ahora podemos cambiar la velocidad de transición entre opciones, ver una pequeña ayuda y volver atrás entre ventanas.



Respuesta rápida



Podemos observar tres botones, con el mismo funcionamiento que la vista anterior, con respuestas rápidas para preguntas y una opción de volver atrás entre ventanas, si seleccionamos esta opción iríamos de nuevo a la ventana principal de la aplicación.

Vemos que sigue apareciendo la barra superior, con la misma funcionalidad que en anteriores ocasiones, cambiar la velocidad, ayuda... etc.

Aquí en el modo administrador, se añaden tres botones inferiores adicionales a la ventana principal que son; Añadir opción, Editar opción y Eliminar opción,

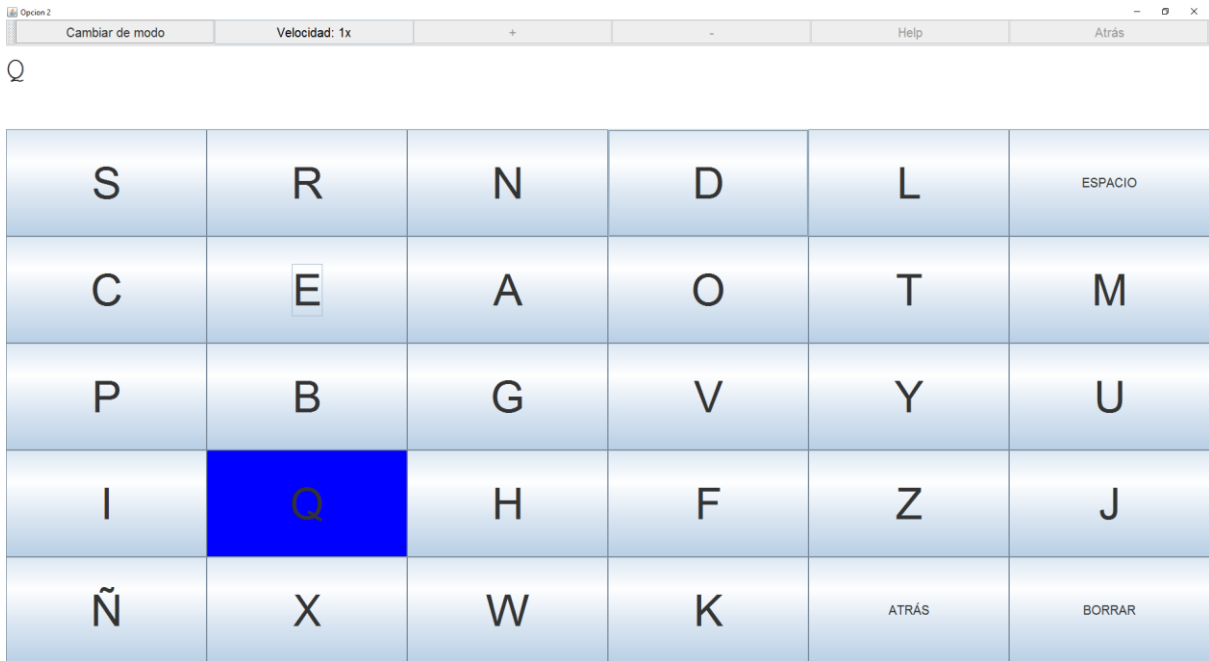
mediante estos botones, podremos añadir más respuestas al sistema, en el caso de que se quiere aumentar la complejidad de estas, o alguna mu concurrente por parte del usuario. De igual modo podremos editar una respuesta o eliminarla.

**Teclado interactivo**, esta opción consta de un teclado mediante el usuario puede escribir palabras y formular frases. Con la misma funcionalidad de las ventanas anteriores, el cursor se va desplazando entre las letras para permitir seleccionar todas ellas.

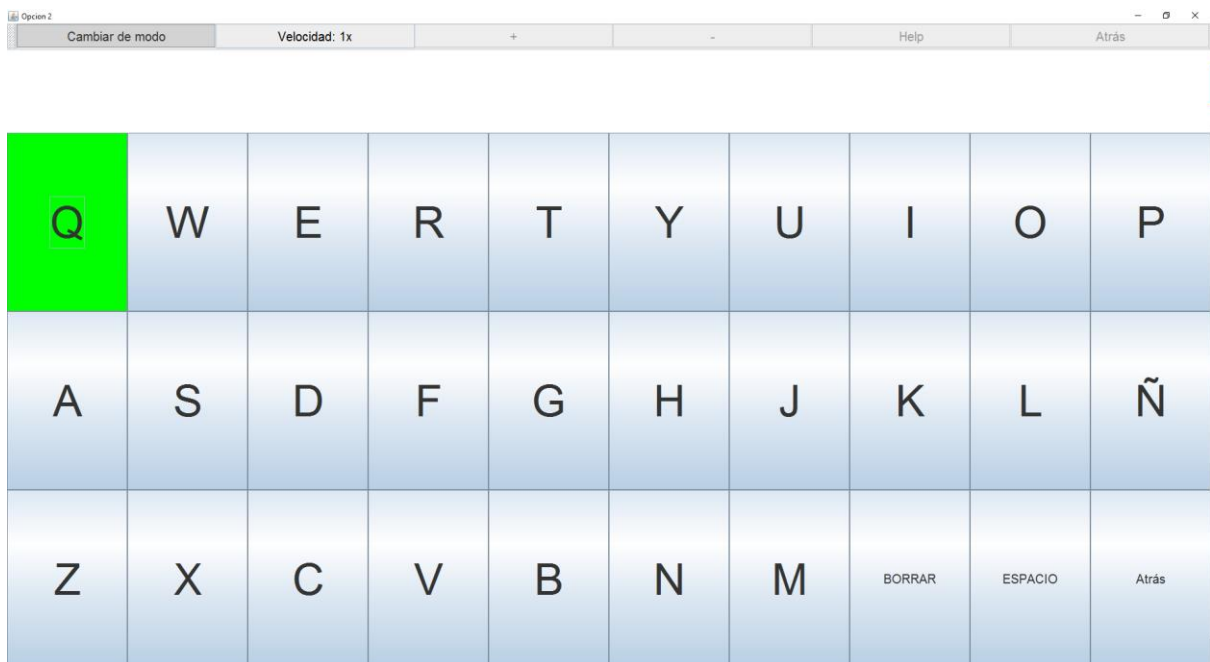
En el modo administrador, se añade un desplegable en el cual se puede elegir el tipo de teclado del sistema, podemos encontrar:

Teclado inteligente: Es el teclado por defecto del sistema, el cursor recorre las letras por columnas, cuando el usuario quiere escribir una letra de las seleccionadas, solo debe pulsar el teclado o ratón, para acceder a esa columna, una vez seleccionada la columna, empieza a recorrer la columna de arriba a abajo, cuando llegue a la última letra de la columna, se des-seleccionará la columna actual y pasará a la siguiente.





Qwerty: Este teclado es similar al teclado convencional, en formato “qwerty”, las teclas están colocadas como el teclado del ordenador, el cursor va avanzando letra a letra permitiendo escribir de forma secuencial.



Letras mayor uso: En teclado las letras se ordenan de forma que se da prioridad a las letras mas usadas en castellano, por frecuencia de aparición en las palabras. El desplazamiento es secuencial, igual que el teclado anterior.

Abecedario: En este caso las letras se colocan de forma consecutiva de aparición en el abecedario, con desplazamiento secuencial.

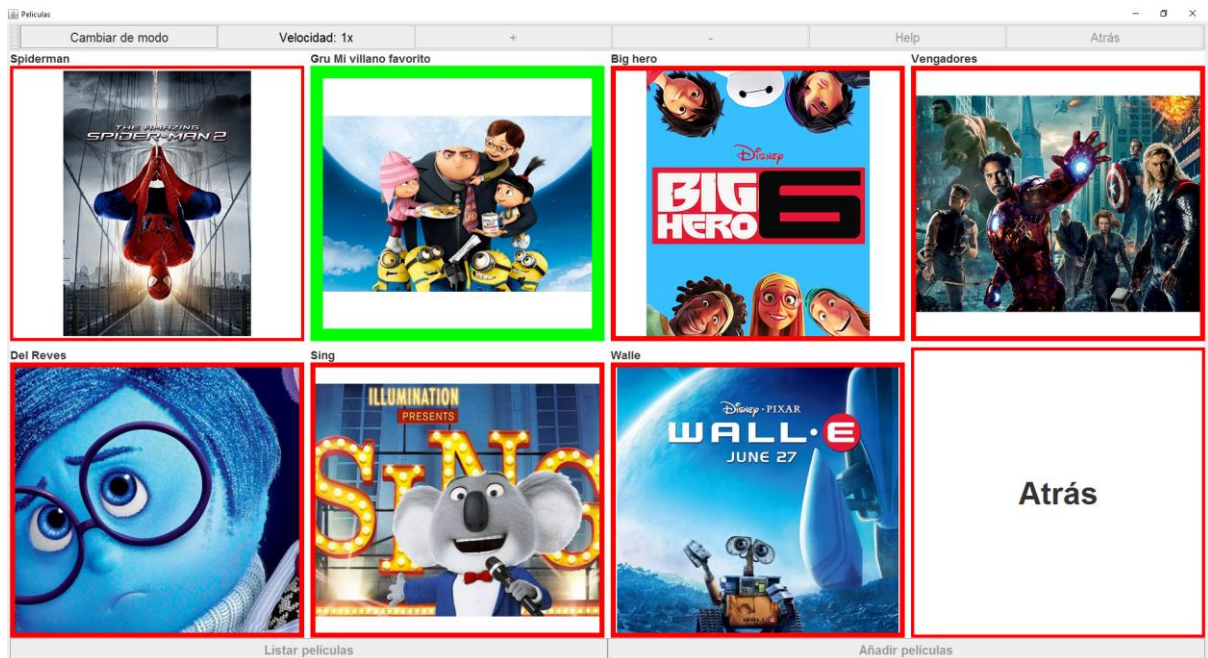
Numérico: Este teclado consta únicamente de números, con desplazamiento secuencial igual que en anteriores casos.



1	2	3	BORRAR
4	5	6	ESPACIO
7	8	9	ATRÁS

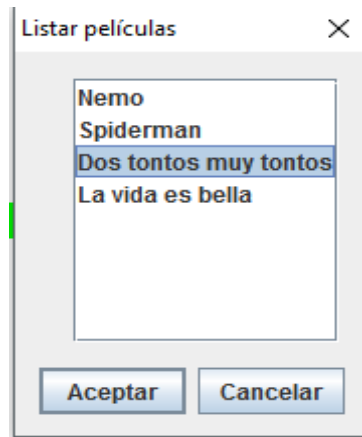
A priori, parece que el teclado más eficiente y rápido es el teclado inteligente, ya que puedes recorrer mayor cantidad de letras y es menos costoso a la hora de elegir la letra deseada, pero hemos querido añadir alguna opción de teclado más, por si en algún caso es necesaria su utilización por parte del usuario.

**Ocio/Peliculas (Multimedia)**, esta última opción se compone de varios botones, siguiendo la temática del resto de ventanas. Aquí el usuario puede seleccionar varias opciones de contenido multimedia.

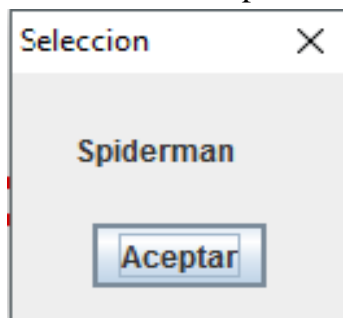


De forma secuencial el cursor va recorriendo toda la serie de contenidos cargados desde la base de datos XML del sistema, al comienzo esta base de datos estará vacía. Es responsabilidad del administrador, tras entrar en dicho modo, igual que en opciones anteriores, mediante el botón “Cambio de modo” añadir este contenido para que pueda ser seleccionado por el usuario, abajo se desbloquearán en el modo administrador dos botones, Listar películas y Añadir películas;

Mediante el primero, mostraremos la lista completa de las películas almacenadas en la base de datos, que posteriormente se mostrarán las siete primeras, el segundo botón nos permitirá añadir contenido a la base de datos.

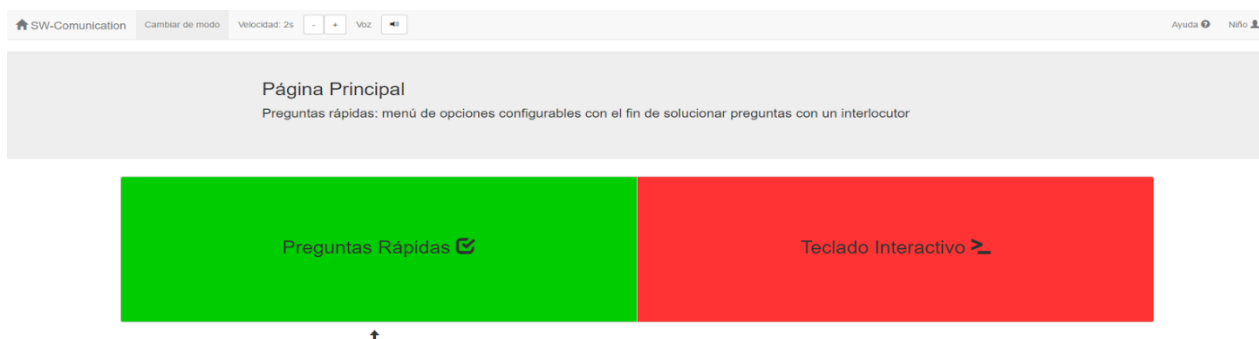


Al seleccionar con el cursor una de estas películas, y tras realizar una acción con el teclado o ratón, aparecerá un banner con en nombre de la película seleccionada. Tenemos en todo momento un botón de atrás, para volver al inicio de la aplicación.



## 8.2 Versión web

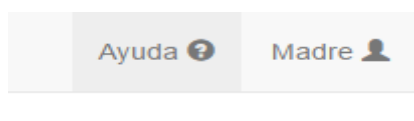
A continuación se procede a explicar las diferentes funcionalidades y vistas de la aplicación web, se accede mediante la url (<http://www.swcommunication.hol.es/>) sin ninguna instalación. Al abrir la página web, nos aparecerá una pantalla como la siguiente:



Al contrario que la versión de escritorio, encontramos únicamente dos opciones. Para seleccionar alguna de estas opciones, esperaremos que el cursor se encuentre encima de la opción en color verde, y hacemos click con el ratón en cualquier parte de la ventana.

En la barra de herramientas encontraremos en la parte superior derecha, un botón de ayuda, que muestra una breve guía de uso del sistema y el modo del sistema:

- Madre o Administrador: puede acceder a todas las vistas, editar la velocidad del barrido, añadir opciones y editar preguntas.
- Usuario: Puede acceder a las vistas que no sean de administración y seleccionar cualquier opción de respuesta.



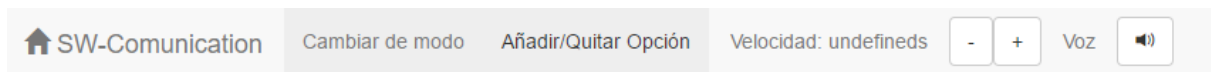
Al entrar en modo Administrador, aparecerá al pie de la web un enlace, que nos permite descargarnos la versión de escritorio de la aplicación:

Para descargar nuestra aplicación en su ordenador: [Pulse aquí](#)

- Preguntas rápidas: Al acceder a esta opción encontraremos una ventana como la que vemos a continuación:

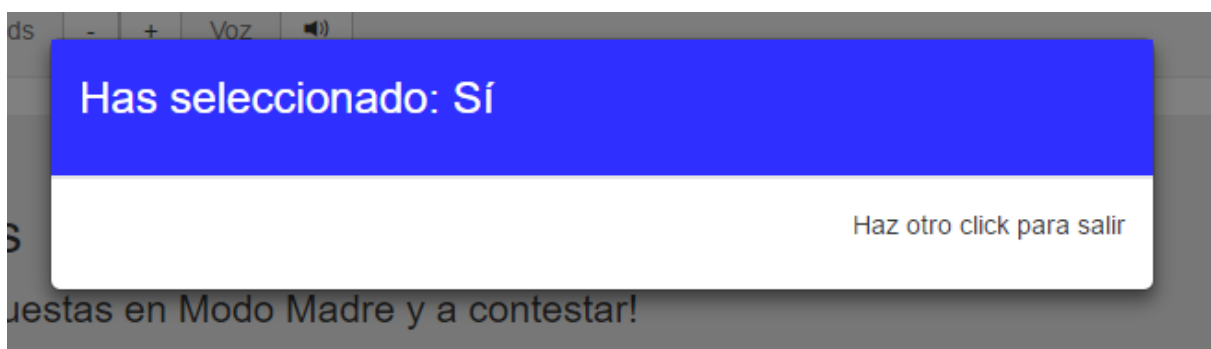


Encontramos un cuadro de texto, y tres opciones de selección. En la barra de herramientas superior:

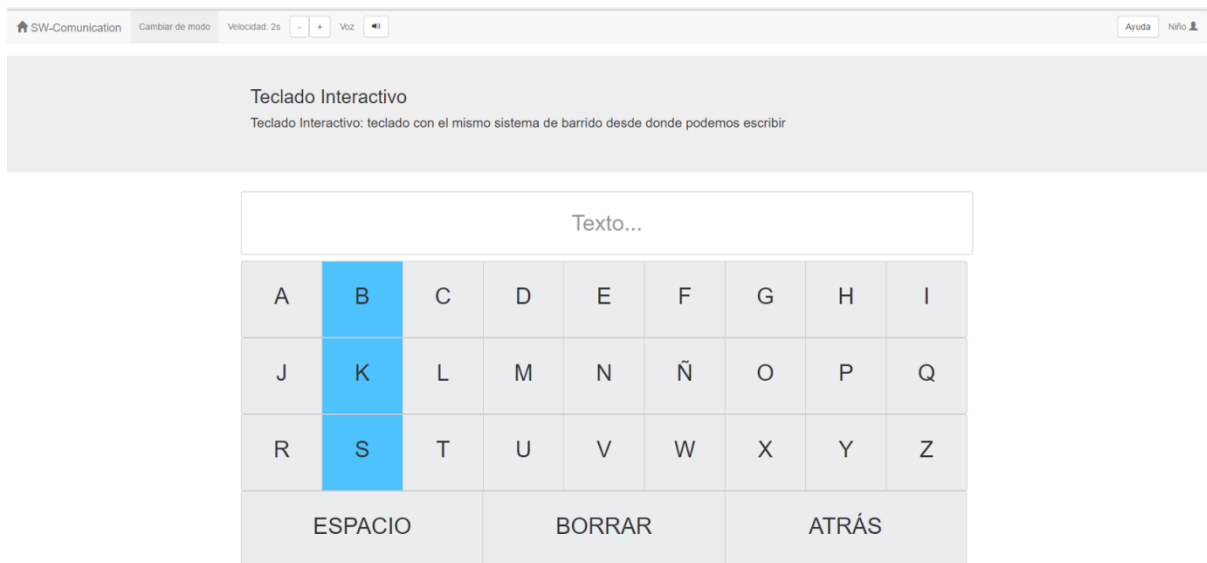


Encontramos las opciones de cambiar de modo, para acceder como administrador y poder cambiar la velocidad del cursor, quitar el sonido a la aplicación, añadir una opción más a las respuestas y introducir una pregunta en el cuadro de texto.

Al introducir una pregunta, el sistema la leerá en voz alta, para que pueda ser contestada por el usuario. Al seleccionar una respuesta para la pregunta, aparecerá un cuadro de texto y una señal de voz con la respuesta:



- Teclado interactivo: Al acceder a esta opción encontraremos una ventana como la que vemos a continuación:



El funcionamiento de esta opción es similar a la aplicación de escritorio, el sistema realiza un barrido de forma horizontal, es decir, recorre las letras en grupos de 3, por columnas (en color azul), cuando hacemos click en una de las columnas, se accede a esa columna y se recorre verticalmente las letras, una vez que se pasa por esas letras, se accede a la siguiente columna.



SW-Communication Cambiar de modo Velocidad: 2s - + Voz Ayuda Niño

Teclado Interactivo  
Teclado Interactivo: teclado con el mismo sistema de barrido desde donde podemos escribir

Texto...

A	B	C	D	E	F	G	H	I
J	K	L	M	N	Ñ	O	P	Q
R	S	T	U	V	W	X	Y	Z
ESPACIO			BORRAR			ATRÁS		

Cuando tenemos seleccionada la letra que se desea escribir en color verde, es el momento de hacer clic de nuevo para escribirla en el cuadro de texto:

SW-Communication Cambiar de modo Velocidad: 2s - + Voz Ayuda Niño

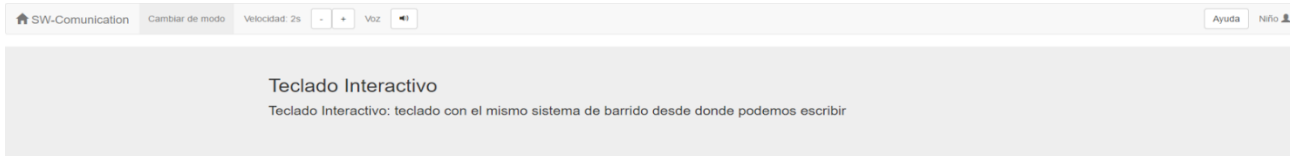
Teclado Interactivo  
Teclado Interactivo: teclado con el mismo sistema de barrido desde donde podemos escribir

E

A	B	C	D	E	F	G	H	I
J	K	L	M	N	Ñ	O	P	Q
R	S	T	U	V	W	X	Y	Z
ESPACIO			BORRAR			ATRÁS		

Para escribir palabras iremos seleccionando letras, que se irán concatenando en el cuadro de texto superior.

En la parte inferior del teclado encontramos 3 opciones, para introducir espacios en el texto, borrar el texto y volver hacia atrás en el sistema y regresar a la vista principal de la web.



## 9. Conclusiones

Después de todo el año desarrollando la aplicación, nos hemos dado cuenta que puede llegar a ser un programa muy útil y necesario para la comunicación de personas con dificultades de movimiento, ya que puede suponer una gran diferencia en cuanto a calidad de vida para estos usuarios. Es cierto que hace falta mucha más investigación en este campo para la ayuda de personas discapacitadas, ya que excluyendo la aplicación ACAT ([Assistive Context-Aware Toolkit](#)) del científico Stephen Hawking, parece que se dejan de lado aplicaciones de este tipo, porque una vez desarrolladas no llegan a ser muy populares, pero la demanda por parte de este colectivo es bastante alta.

Hay muchas alternativas a pesar de que no se explotan, ya que realizando una búsqueda no demasiado exhaustiva, podemos encontrar propuestas similares a la nuestra, con funcionalidades ligeramente diferentes aunque todas se centran en aplicación y nada web. Es justo ahí donde hemos querido diferenciarnos del resto de soluciones en el mercado. Lo hemos conseguido incluyendo una solución web dando una respuesta mucha más rápida evitando cualquier tipo de instalación y consiguiendo que sea una web “responsive” capaz de funcionar en ordenador, tableta y la mayoría de dispositivos móviles. Además incluyendo dentro de la web una opción para descargar un ejecutable de nuestra aplicación java y así conseguir más visibilidad de la misma.

Hemos podido comprobar que el problema principal del escaso éxito de éste tipo de software sobre todo consiste en la voluntad y el esfuerzo por integrar y mejorar estas soluciones en los usuarios, ya que diseñar una aplicación genérica que cumpla los requisitos de miles de usuarios a la vez es algo complicado, y desarrollar un software específico para cada persona sería algo muy costoso.

Nuestra labor ha sido recopilar todas estas funcionalidades y requisitos propuestos y elaborar un software de calidad y útil para este cometido. Intentando abarcar varias opciones de comunicación y realizar un trabajo completo. También hemos podido comprobar que éste tipo de software debe ser código libre y de fácil acceso para todos, ya que o bien porque los usuarios finales no dispongan de recursos necesarios para costearse este software, que en ocasiones puede llegar a ser muy caro, o bien por la continuación de proyectos como el nuestro, por más desarrolladores, y que juntos podamos conseguir un mejor resultado. Al no contar, por norma general, con el respaldo de grandes entidades es muy importante que una vez acabada la labor (altruista en muchos casos) se siga mejorando y revisando el producto, por eso es tan importante que el código sea libre.

Para acabar, mencionar que ha sido muy enriquecedor trabajar en un proyecto de este tipo, y más sabiendo al público que va dirigido, para intentar evitar la exclusión social y poder mejorar su calidad de vida en todo lo posible. Es cierto que nos ha faltado en ciertas partes del desarrollo una mayor comprensión del problema, nos hubiese gustado involucrarnos más en las pruebas de usuario, para pulir en cierto modo la implementación final del software, pero no dudamos en que este proyecto será continuado, para que llegue a mucha más cantidad de usuarios y pueda satisfacer todas sus necesidades.

# 10. Conclusions

After all year developing this application, we could appreciate that it could be an utile and needed programme for the communication between people with movement difficulties, since it may cause a difference regarding to living conditions of the users. Although it is true that is necessary to investigate much more in this field to help disabled people, excluding the ACAT (Assistive Context-Aware Toolkit) application that has Stephen Hawking, seems to be put aside applications like these, because once they are developed fall short of popularity, but the demand by this group is quite high.

There are a lot of options despite of they are not exploited, since searching not so exhaustive, we could find similar software as ours, with some similarities to our idea although every item is focused on application and not on the web. That's the moment where we wanted to differentiate our solutions from the others in the market. We have achieved this with a responsive web, capable of operating on a computer, tablet and the most phones. Even we included in the web an option to download an executable of our java application.

We could verify that the main problem of the little success of this kind of software specially consist on willingness and the effort to integrate and improve these solutions for users, since to design a generic application that fulfil all the requirements of millions of users at once is complicated, and develop an specific software for each user will entail high economic costs.

Our work had been to collect all this proposed features and requirements to elaborate a quality software in order to be utile to its mission. Trying to consider some communicational options and carry out a complete work. Also, we could prove that this kind of software should be free software and needs to have free access to everybody, because if final users does not have financial resources in order to fund this software (that sometimes can become very expensive) either the continuity of projects as ours, more developed, together we will achieve a better outcome. Not having large institutions support is so important that after being completed the work (altruist in the most of the time) it must continue improving and checking the product, that is why the free software or free code is so important.

To complete, it is worth mentioning that it has been quite rewarding to working on a project like this, and furthermore knowing the target audience, to avoid the social exclusion and improving the quality of life as much as possible. It is true that in some parts a great amplify in order to understand the problem have been absent,

we would love to get more involved in users' texts, but we have no doubts about this project and its function continuity in order to reach to more users and that it could satisfy all their necessities.

## 11. Mejoras Futuras

Podemos dividir las mejoras futuras y los próximos pasos a seguir en dos divisiones: las mejoras en la aplicación y las mejoras en la web.

### 11.1 Aplicación

Respecto a las mejoras en la aplicación creemos que las funcionalidades principales ya están definidas y deberían seguir siendo las mismas: las 3 opciones deberían seguir siendo las mismas con la misma esencia. Sobre esa base deberíamos ampliar las funciones que ofrecen cada una de las opciones, construir una aplicación más robusta y conseguir que la aplicación sea amigable para el usuario, cómoda y fácil de usar.

Partiendo de ahí procedemos a explicar todas las mejoras que creemos que deberían hacerse en un futuro.

#### **Mejoras:**

1. Mejorar el aspecto visual de la aplicación en su conjunto haciéndola más atractiva y aplicando los cambios obtenidos de los resultados de las pruebas de usuarios. Necesitamos conseguir que aparte de que la aplicación sea funcional también sea amigable con el usuario.
2. Inclusión de un tutorial interactivo o mensajes de ayuda que ayuden al usuario en ambos modos para no llegar nunca a producir frustración. Entendemos que según vaya pasando el tiempo el tutorial iría desapareciendo hasta llegar un punto en que los usuarios sean expertos y no sea necesario.
3. Proporcionar voz personalizada a todo el conjunto de la aplicación. A pesar de nuestra investigación acerca de ello no conseguimos encontrar ninguna librería de voces en castellano para Java y sería muy interesante que las selecciones por parte del usuario en cualquiera de las opciones fueran reproducidas por el altavoz hacia el exterior.
4. Capacidad para guardar preguntas ya hechas en la opción 1, con respuestas incluidas lo que haría más ágil el proceso si se repite por ejemplo todos los

- días la misma pregunta (tipo: ¿Qué te apetece desayunar? Más las respuestas).
5. Modificación de la opción 3 para elegir tu el número de películas a mostrar, ya que actualmente es estático.
  6. Extensión de la opción 3 a cualquier categoría de ocio que se quiera. Actualmente está pensado para películas, pero en un futuro la aplicación debería poderse comportar igual para cualquier categoría que quisiéramos crear añadiendo un submenú si fuera necesario para elegir a que categoría entrar.
  7. Traslado de toda la lógica de las películas a una pequeña base de datos o a otros lenguajes que nos permitan más flexibilidad. Aunque XML nos permite rapidez y ocupa poco espacio, cualquier modificación a posteriori resulta costosa.
  8. Predictor de palabras en opción 2. A veces resulta muy tedioso tener que escribir las palabras y frases enteras y pueda derivar en un proceso largo y aburrido para el usuario. Con un predictor de texto se arreglaría el problema. Dicho predictor, según se van escribiendo letras, ofrecería la posibilidad de elegir posibles palabras e incluso posibles frases ya utilizadas anteriormente. Sería necesario un estudio de las palabras más usadas frecuentemente en castellano y un seguimiento de los usuarios en particular para ver qué palabras o frases repiten con más frecuencia.
  9. Adaptación de la aplicación a otros idiomas, sobre todo inglés, ya que buscamos que llegue a todo el público posible.

## 11.2 Web

La idea de la realización de la web, como se explica en apartados anteriores, es proporcionar una herramienta ágil y que el usuario no tenga que descargarse ni configurar nada previo a usarla. También que sea posible de usar en todos los dispositivos así que habría que mantener que la web fuera responsive. Al realizar primero la aplicación creemos que la web se ha quedado un poco descolgada así que el paso previo sería conseguir que las 3 opciones de la aplicación funcionaran correctamente en la web. Por supuesto también debería sufrir todas las mejoras anteriores enunciadas para la aplicación, exceptuando el traslado del XML que al ser web funcionaría de otra forma que explicaremos más adelante. Partiendo de esta base enunciamos las mejoras destinadas específicamente a la plataforma web:

## Mejoras:

1. Realización de un sistema de usuarios con login para que cada uno pudiera acceder a sus categorías de ocio personalizadas y, por ejemplo, almacenar sus películas favoritas. Para ello sería necesario construir una aplicación web. Con a priori, los únicos datos referentes a usuarios (usuario y contraseña) y referentes a categorías de ocio pertenecientes a cada usuario. Se debería decidir el modelo de base de datos (sql o nosql), el servidor etc. Al ser la parte de categorías muy “abstracta” sería inteligente montar la web sobre un sistema con nosql, por ejemplo usando MongoDB. También habría que decidir si la aplicación empieza en una pantalla de login o se guarda siempre la contraseña automáticamente, ya que no creemos que fuera a tener más de un usuario distinto desde el mismo dispositivo (mediante una cookie por ejemplo). Y en caso de que empezara en una pantalla de login si ya incluso para la entrada de datos (usuario y contraseña) sería necesario crear un teclado virtual con barrido temporizado para que el propio usuario con dificultad de movimiento fuera quien introdujera sus propios datos.
2. Posibilidad de acceder a la web sin realizar ningún login mediante un modo anónimo. Al entrar a la web mediante este modo sólo se tendría acceso a las 2 primeras opciones (preguntas rápidas y teclado) y sólo con algunas funcionalidades. En el caso de la opción 1 no guardaría ningún dato sobre preguntas y respuestas hechas anteriormente y en el caso de la opción 2 no se guardaría ningún historial de palabras que pudiera usar el predictor de texto.

# 12.Referencias

- 1) <https://opensource.org/docs/osd>
- 2) <https://01.org/acad>
- 3) <http://presage.sourceforge.net/>
- 4) <https://github.com/01org/acad>
- 5) <http://www.click2speak.net/>
- 6) <https://swiftkey.com/es>
- 7) <http://www.tecnologiasaccesibles.com/es/>
- 8) <http://www.indracompany.com/es/>
- 9) <https://www.bq.com/es/limbika>
- 10) <http://dilo.iter.es/>
- 11) <https://play.google.com/store/>
- 12) <http://www.xtec.cat/~jlagares/>
- 13) [www.wikipedia.org](http://www.wikipedia.org)
- 14) [www.genbetadev.com](http://www.genbetadev.com)
- 15) [www.i2btech.com](http://www.i2btech.com)
- 16) [www.zentec.cl](http://www.zentec.cl)
- 17) <https://opensource.org/licenses/MIT>
- 18) <https://eclipse.org/>
- 19) <https://jsfiddle.net/>
- 20) <https://notepad-plus-plus.org/>
- 21) <https://www.google.es/chrome>
- 22) <https://github.com/>
- 23) [www.jdom.org/](http://www.jdom.org/)
- 24) <https://freetts.sourceforge.io/>
- 25) [www.getbootstrap.com/](http://www.getbootstrap.com/)
- 26) <https://jquery.com/>
- 27) <https://www.java.com/es>
- 28) <https://www.javascript.com/>

# 13.Apéndices

## Apéndice I: Contribuciones individuales

### 13.1 Sergio González Francisco

#### 13.1.1 Respecto a la aplicación

1. Investigación de como tener una base de datos en Java de la manera más sencilla.
2. Investigación sobre el uso específico de observadores para distintas funcionalidades.
3. Investigación sobre cómo implementar y controlar los diferentes tipos de modo de la aplicación (modo madre y modo niño).
4. Creación de la cabecera o ‘toolbar’ común para todas las vistas.
5. Creación de la opción 3 (ocio, multimedia), dentro de la cual existen partes importantes:
  - a. Investigación y uso de la librería jDom, mediante la cual se mantiene una base de datos (archivo .xml) donde se albergan los archivos multimedia y con la que se crea, modifica o rescata información del mismo.  
Esto se utiliza al mostrar los archivos multimedia iniciales, añadir nuevos (Añadir) o listar los que se encuentran en la base de datos (Listar).
  - b. Investigación e implementación de como guardar imágenes que se añaden a través de la aplicación en una ruta específica. Esto ocurre al añadir un archivo multimedia nuevo.
  - c. Control de la vista en cuanto a los distintos tipos de modo, diferente velocidad etc...
6. Modificación de distintos elementos en la página principal y en las distintas opciones de la aplicación.
7. Adecuación de la aplicación a los cambios que van surgiendo según van pasando las iteraciones.

## 13.1.2 Respecto a la web

1. Investigación e implementación de como pasar variables por URL a través de JavaScript.
2. Investigación e implementación de como descargar archivos.
3. Investigación de como subir una página web a un servidor.
4. Configuración de la página web en el servidor y su correspondiente puesta a punto.

## 13.1.3 Respecto a la memoria

1. Introducción
2. Formas de programación
  - a. Los distintos lenguajes de programación que se han utilizado para llevar las implementaciones.
  - b. El MVC (Modelo-Vista-Controlador) utilizado en Java.
  - c. Las diferentes librerías que se han usado para facilitar y mejorar las distintas implementaciones llevadas a cabo.
3. Tecnologías usadas
  - a. Programas y herramientas utilizados para la versión Java.
  - b. Programas y herramientas utilizados para la versión Web.
4. Formas de almacenamiento de código
  - a. Tipo de almacenamiento y control de versiones usado para el código.
  - b. Almacenamiento de información y documentación.
5. Descripción de distintos apartados más secundarios como: 'Referencias', 'Repositorio'...
6. Diseño de la portada, índice y paginación.
7. Maquetación de la memoria.

## 13.2 Enrique Laguna Munuera

### 13.2.1 Respecto a la aplicación

1. Investigación sobre aplicaciones con funcionalidad similar, presentes en el mercado actual y testeo de dichas aplicaciones.
2. Investigación de la posibilidad de incluir voz en aplicaciones Java.
3. Creación y modificación posterior de la opción 2 e implementación de los cambios posteriormente requeridos.
4. Investigación sobre algoritmos inteligentes de recorrido en teclado.
5. Creación de varios teclados referentes a la opción 2, para una mayor versatilidad de escritura.
  1. Teclado QWERTY
  2. Teclado Inteligente: recorriendo por filas y columnas las letras.
  3. Teclado alfabético.
  4. Teclado numérico.
  5. Teclado letras mayor uso: colocación de letras en función de porcentaje de uso en castellano.
6. Realización de ejecutable con el fin de poder realizar pruebas.
7. Adecuación de la aplicación a los cambios que van surgiendo según van pasando las iteraciones.

### 13.2.2 Respecto a la web

1. Creación de la estructura del teclado de la web para dar a la web un estilo “responsive” que permite usar la misma en varios dispositivos.
2. Investigación sobre los uso de elementos de tiempo en web.
3. Investigación sobre acciones al hacer clic en cualquier elemento de la página.
4. Investigación sobre la posible inclusión de voz en la web.
5. Investigación sobre librerías responsive para la web.
6. Creación de la opción 2 “Teclado interactivo”. Se continua con el mismo funcionamiento que en la aplicación de escritorio, se recorren las columnas de letras y cuando se selecciona alguna de estas se recorren las filas.

## 13.2.3 Respecto a la memoria

1. Estructura de la memoria
2. Metodología de desarrollo:
  - a. Explicación de la metodología usada para el desarrollo de proyecto software.
3. Descripción del programa:
  - a. Aplicación de escritorio.
  - b. Aplicación web.
4. Conclusiones:
  - a. Balance final del proyecto, conclusiones tras el desarrollo y futuro del proyecto.

## 13.3 Daniel Ortiz Sánchez

### 13.3.1 Respecto a la aplicación

1. Creación y administración del repositorio donde se alberga el software en la plataforma Github.
2. Selección de la licencia del producto.
3. Investigación sobre la viabilidad de realizar una comunicación y una aplicación dinámica en lenguaje Java sólo mediante un clic.
4. Investigación de la posibilidad de incluir voz en aplicaciones Java.
5. Creación de la idea principal de la aplicación y fundamentos básicos. Decisión de basar el funcionamiento de la misma en 2 ideas principales adaptando el modo de funcionamiento de la aplicación a ésta plataforma:
  - a. Mostrar las opciones en la pantalla y por medio de un temporizador ir cambiando el estilo y la activación de las opciones de modo que cada x tiempo cambie la opción seleccionada.
  - b. Con un evento disparador cada vez que se haga clic sobre el documento se entra a una función que nos indica qué opción está seleccionada en el momento del clic.
6. Creación de la página principal y funcionamiento de la misma, tanto del modo niño como del modo madre con las opciones de la misma en su versión inicial.
7. Creación y modificación posterior de la opción 1 e implementación de los cambios posteriormente requeridos.

8. Modificación de la parte visual de la opción 3 con objetivo a mejorar el estilo y adecuación a requisitos concretos como mostrar x películas o remarcar de una forma más intensa la película seleccionada en cada caso.
9. Realización de ejecutable con el fin de poder realizar pruebas.
10. Adecuación de la aplicación a los cambios que van surgiendo según van pasando las iteraciones.

## 11.3.2 Respecto a la web

1. Creación y administración del repositorio donde se alberga el software en la plataforma Github.
2. Selección de la licencia del producto.
3. Creación de la estructura principal de la web y decisión de bibliotecas a usar para dar a la web un estilo “responsive” que permita usar la misma en varios dispositivos.
4. Creación de la idea principal de la web y fundamentos básicos. Decisión de basar el funcionamiento de la web en las 2 mismas ideas principales que usamos en la aplicación (temporizador y clic en página) incluyendo modificaciones para adaptarlo al formato y lenguaje web. Decisión de no modificar la frecuencia del temporizador directamente si no simularlo mediante variables y adaptación de clic en todos los elementos a clic en toda la página.
5. Investigación sobre los uso de elementos de tiempo en web.
6. Investigación sobre acciones al hacer clic en cualquier elemento de la página.
7. Investigación sobre la posible inclusión de voz en la web.
8. Creación de la página principal de la web: movimiento entre opciones (incluida voz y textos), barra de herramientas superior, e implementación de las opciones de ésta, y modo madre. Creación del evento de clic en el documento para direccionar a la opción seleccionada.
9. Creación de la opción 1 “Preguntas Rápidas”. Modificación del funcionamiento de la edición de opciones con respecto a la aplicación: se editará directamente su contenido y no se dispondrá de un cuadro de texto adicional. Creación del funcionamiento de la misma basada en la opción anterior, añadiendo la opción del título en caso de estar rellenado o no. Creación del funcionamiento de la web al seleccionar una opción. Creación de las nuevas opciones en la barra con respecto a la página principal y adaptación de las demás a ésta nueva vista, nueva opción y editar opciones existentes.

### 11.3.3 Respecto a la memoria

1. Resumen.
2. Estado del arte:
  - a. Obtención de características concretas de nuestras aplicaciones para lograr el nicho de mercado donde estamos ubicados.
  - b. Listado de softwares similares ordenados por importancia (usuarios, relevancia, tecnologías usadas).
  - c. Investigación de softwares seleccionados y explicación de cada uno, en orden de relevancia.
3. Desarrollo del sistema:
  - a. Investigación sobre el desarrollo de sistemas interactivos en otros casos y pasos a seguir.
  - b. Realización del propio desarrollo.
4. Mejoras futuras:
  - a. Análisis de las necesidades de nuestra aplicación y web al concluir el desarrollo.
  - b. Explicación de la continuación del desarrollo de nuestros productos.

# Apéndice 2: Guía de uso

En la siguiente guía se detalla cómo realizar las diferentes funciones que ofrece la aplicación/web.

## 1. Instalación

Para utilizar la aplicación en su ordenador, lo único que tiene que hacer es acceder a nuestra página web (<http://www.swcommunication.hol.es>), cambiar el modo, ya que el inicial es el modo niño, y bajarse la aplicación (archivo .jar) desde el enlace que aparece en la página principal. Una vez descargado este archivo, haciendo doble click en el mismo, la aplicación comenzará a ejecutarse.

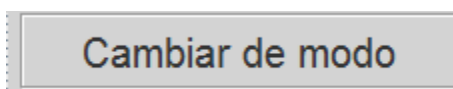
Si por el contrario, no quiere descargar la aplicación, puede hacer uso de ella en la misma página web, aunque no podrá disfrutar de la funcionalidad de ocio/multimedia.

## 2. Menú/Toolbar

En la parte superior, existe un menú-toolbar, con el cual se pueden realizar distintos cambios en el modo de la aplicación, en la velocidad etc...

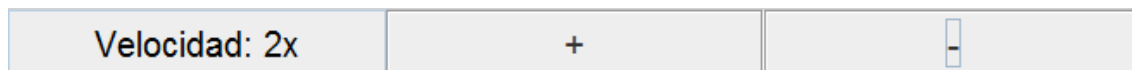
### -¿Cómo se cambia de modo?

Para cambiar de modo, se tiene que pulsar el botón 'Cambiar de modo' ubicado en la parte superior izquierda. El modo inicial es el modo niño, y solo hay 2 modos (niño, madre).



### -¿Cómo cambiar la velocidad?

Para cambiar la velocidad del temporizador, primero se debe estar en el modo madre, ya que para el modo niño está deshabilitado.



Una vez estando en el modo madre, se dispone de los botones (+) y (-), para aumentar o disminuir la velocidad. Existen únicamente 4 grados de velocidad. Habiendo escogido el deseado, se vuelve a pulsar el botón 'Cambiar de modo' y el temporizador estará a la velocidad deseada para el modo niño. Esta velocidad se mantiene al entrar al resto de opciones.

### 3. ¿Cómo acceder a una de las opciones?

Una vez arrancada la aplicación, ya sea en versión escritorio o en versión web, la primera pantalla que aparece es la siguiente:



Para poder acceder a una de las posibles opciones, en el modo niño, se tiene que esperar hasta que se marque en verde la deseada, y hacer click o pulsar una tecla. En el caso del modo madre, aparecerán todas las opciones en verde y con el ratón hará click en la que desee entrar.

### 4. Respuestas rápidas



- **¿Cómo editar las distintas opciones?**

Para editar las opciones, se debe estar en ‘modo madre’, el botón de editar opciones que aparece abajo se habilitará, y se podrán editar los cuadros.

- **¿Cómo añadir o eliminar opciones?**

Al igual que en el caso anterior, se debe estar en el ‘modo madre’, y los botones de añadir y eliminar opción se habilitarán. Se pueden tener hasta 4 opciones, y como mínimo 2, sin tener en cuenta el botón de ‘Atrás’.

## 5. Teclado interactivo

- **¿Cómo se selecciona la letra deseada?**

Primeramente, la aplicación recorre todas las columnas. Cuando la columna donde esté la letra deseada esté en verde, se hace click o se pulsa una tecla.



S	R	N	D	L	ESPACIO
C	E	A	O	T	M
P	B	G	V	Y	U
I	Q	H	F	Z	J
Ñ	X	W	K	ATRAS	BORRAR

Después, comenzará a recorrer todas las letras de dicha columna (por filas). Cuando la letra deseada esté en azul, se hace click o se pulsa una tecla, y la letra aparecerá escrita encima del teclado.

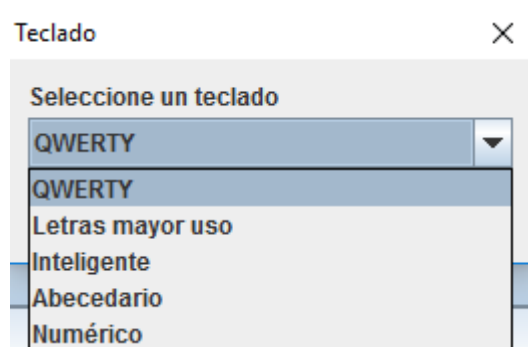


E

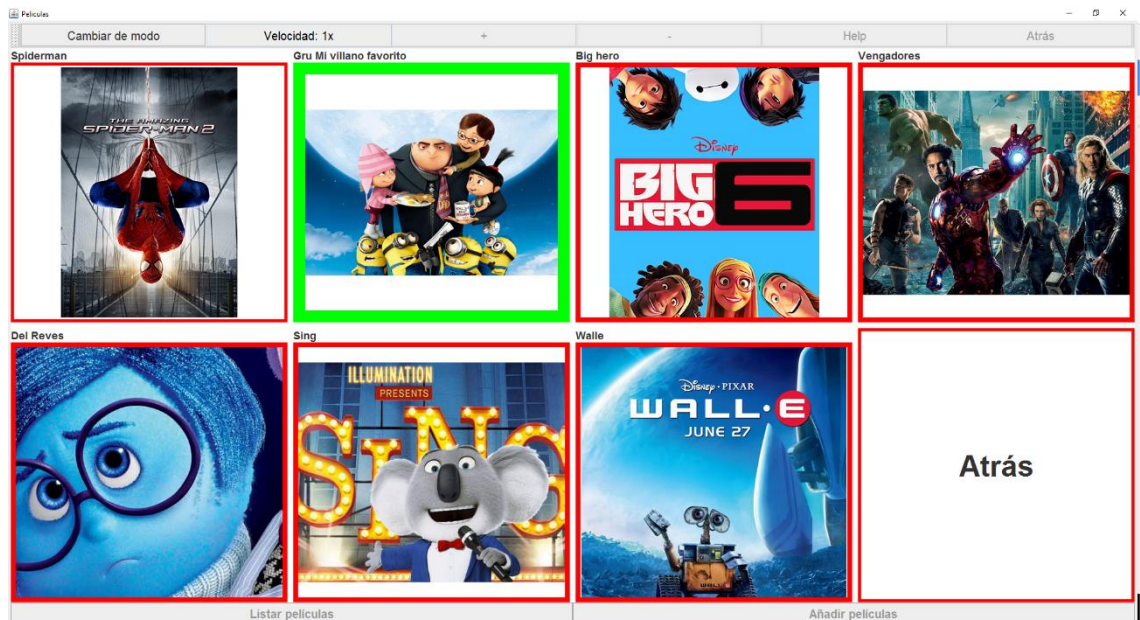
S	R	N	D	L	ESPACIO
C	E	A	O	T	M
P	B	G	V	Y	U
I	Q	H	F	Z	J
Ñ	X	W	K	ATRAS	BORRAR

- **¿Cómo se cambia de tipo de teclado?**

Para cambiar el tipo de teclado, lo único que hay que hacer es pulsar el botón de 'cambio de modo' (cambiar de 'modo niño' a 'modo madre'), y aparecerá automáticamente un desplegable con los distintos tipos de teclado que se pueden usar.

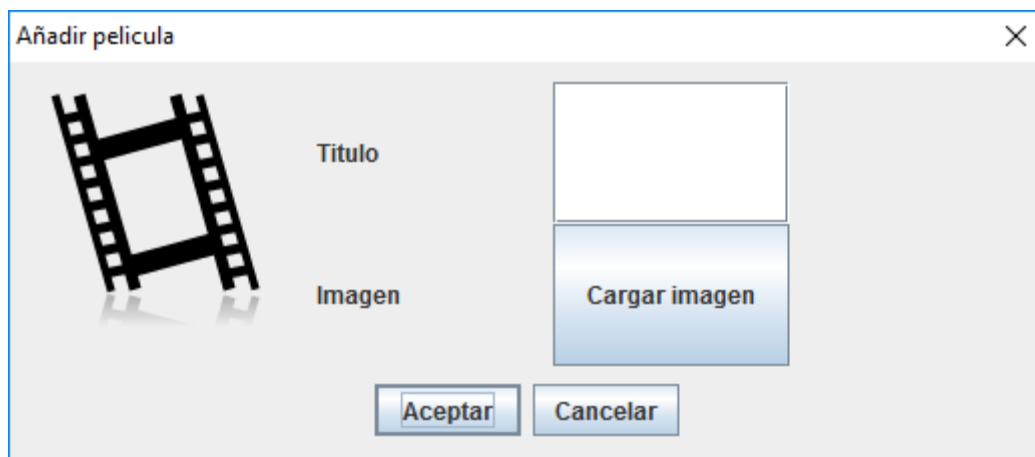


## 6. Ocio/Películas



### - ¿Cómo se añaden nuevas películas/archivos?

Para añadir nuevos archivos multimedia, hay que cambiar de modo al ‘modo madre’ para que el botón de ‘Añadir películas’ se habilite. Pulsando sobre él, se abre una nueva ventana donde se puede introducir el título e insertar una imagen que esté en su ordenador. No es obligatorio insertar una imagen.

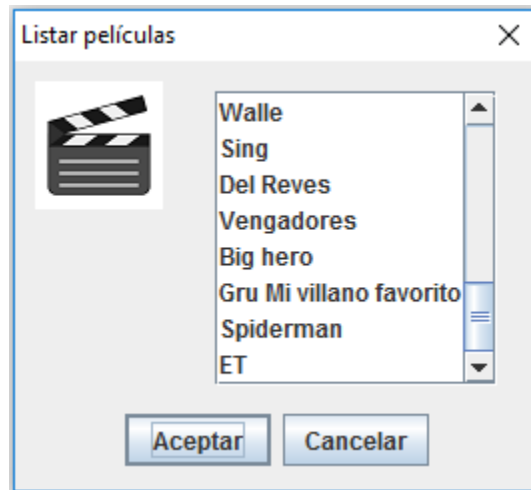


Pulsando el botón aceptar, queda guardada y aparecerá la primera en el listado de películas que se muestran.

### - ¿Cómo se muestran las películas/archivos que se quiere?

Para listar archivos multimedia, hay que cambiar de modo al ‘modo madre’ para que el botón de ‘Listar películas’ se habilite. Pulsando sobre él, se abre una nueva

ventana donde aparecen todas las películas registradas en la base de datos. Pinchando sobre ellas, se puede ir escogiendo las que quieren que se muestren. Se deben de elegir desde una película hasta un máximo de 7.



Pulsando el botón aceptar, queda guardada la selección y aparecerán las películas seleccionadas junto con su título e imagen.