



ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

Titulación:

INGENIERÍA DE TELECOMUNICACIÓN

Título del proyecto:

INTEGRACIÓN DE APLICACIONES Y GESTIÓN DE LA
BASE DE DATOS DE UN INTERFAZ TÁCTIL QUE
FACILITA LA COMUNICACIÓN DE PERSONAS CON
DISCAPACIDAD

Ainhoa Esténoz Abendaño

Dr. Carlos Fernández Valdivielso

Pamplona, 29 de julio de 2011

Agradecimientos

En primer lugar, quiero agradecerles a todos los colaboradores del proyecto (Isterria, Anfas, Laura, Elena, Rosa, Javier, Marta, Angélica), por sus consejos, trabajo y dedicación desinteresada, sin todos ellos este proyecto distaría muchísimo de todo lo que es.

En segundo lugar, muchísimas gracias a mi tutor Carlos, por darme la oportunidad de formar parte de este proyecto, por la confianza depositada en mí y por descubrirme otras apasionantes facetas de la ingeniería.

En tercer lugar, quiero dar las gracias a Eli y a Pepe, por su esfuerzo, el gran equipo que hemos formado y por todas las vivencias que dejamos por el camino. Ha sido un placer trabajar y divertirme con vosotros.

A mis compañeros de clase, por compartir tantas alegrías y penas en este viaje, pero sobre todo por no ser sólo compañeros, sino también amigos.

A Sara, sólo tú y yo sabemos qué ha significado la una para la otra durante esta etapa en nuestras vidas. Del mismo modo que lo seguirá siendo en etapas posteriores. Sé que siempre contaré contigo. Muchísimas gracias.

A mis amigas, por vuestra sincera amistad y por ayudarme a seguir creyendo en los valores más importantes de la vida y en percibir mi carrera como uno más.

A mi familia, por el apoyo incondicional todos estos años y por creer siempre en mí. Gracias a vosotros ya veo la luz al final del túnel, esa que tantas veces he dudado de su existencia.

Y en especial a ti, Jesús, por estar siempre allí. Gracias por tu paciencia inagotable, en especial durante todas las épocas de exámenes, tus infinitos consejos y por ayudarme a descubrir realmente quién soy. Gracias por todos estos años de felicidad. Tu apoyo ha sido crucial para lograr mis sueños.

*Sinceramente, muchísimas gracias a todos.
Ainhoa*

Índice

Introducción	Página 7
Motivación	Página 9
Objetivos	Página 10
Capítulo 1: Antecedentes	Página 11
Capítulo 2: Estudio y desarrollo código S.P.C.	Página 14
2.1 Estudio código S.P.C.	Página 15
2.2 Desarrollo código S.P.C.	Página 18
Capítulo 3: Estudio y desarrollo lenguaje programación	Página 21
3.1 Estudio lenguaje programación	Página 22
3.1.1 El lenguaje: Objective C	Página 22
3.1.2. Primer paso: Desarrollador	Página 23
3.1.3. Segundo paso: Descarga del kit de	Página 23
3.1.4. Xcode	Página 24
3.1.5. Interface Builder	Página 26
3.2 Desarrollo lenguaje programación	Página 29
Capítulo 4: Estudio y desarrollo BBDD	Página 36
4.1 Estudio base de datos	Página 37
4.2 Desarrollo base de datos	Página 42
4.2.1 Decisión base de datos	Página 42
4.2.2 Creación base de datos	Página 43
4.2.3 Desarrollo base de datos	Página 44

Capítulo 5: Estudio y desarrollo CoreData	Página 56
5.1 Estudio CoreData	Página 57
5.2 Desarrollo CoreData	Página 59
Capítulo 6: Estudio y desarrollo menú de configuración de la aplicación	Página 63
Capítulo 7: Conclusiones	Página 73
Capítulo 8: Líneas futuras	Página 75
8.1 Agenda	Página 76
8.2 Diario	Página 77
8.3 Otras mejoras	Página 77
Capítulo 9: Presupuesto	Página 79
Bibliografía	Página 81
Anexos	Página 85
Plan de viabilidad	Página 86
Anexo A: COMUNICADORAppDelegate.h	Página 111
Anexo B: COMUNICADORAppDelegate.m	Página 113
Anexo C: COMUNICADORViewController.h	Página 121
Anexo D: COMUNICADORViewController.m	Página 130
Anexo E: Código fuente "Clases"	Página 183

Introducción

Las nuevas tecnologías de la información y la comunicación lo han cambiado todo; la forma de vivir, de trabajar, de producir, de comunicarnos, de comprar, de vender, de enseñar, de aprender... Todo nuestro entorno es distinto, por lo que se han convertido en una herramienta básica en la vida diaria. No obstante, las TIC evolucionan notablemente para todos, menos para los que más necesitan de ellas.

Se denominan Tecnologías de la Información y las Comunicaciones al conjunto de tecnologías que permiten la adquisición, producción, almacenamiento, tratamiento, comunicación, registro y presentación de información, en forma de voz, imágenes y datos contenidos en señales de naturaleza acústica, óptica o electromagnética. Las TICs incluyen la electrónica como tecnología base que soporta el desarrollo de las telecomunicaciones, la informática y el audiovisual.

Según el diccionario de la lengua española D.R.A.E., discapacidad es la cualidad de discapacitado. Dicho de una persona, se refiere a aquella que tiene impedida o entorpecida alguna de las actividades cotidianas consideradas normales, debido a la alteración de sus funciones intelectuales o físicas.

Nuevas tecnologías y discapacidad, dos términos muy diferentes en su definición, pero que este proyecto tiene como objetivo aunar. En una Sociedad de la Información y del Conocimiento realmente desarrollada, es fundamental que todas las personas que la integran, independientemente de su edad o capacidades físicas o psíquicas, tengan acceso a las mismas oportunidades. Más allá de generar nuevas barreras, las TIC han de servir como herramienta de integración para las personas con discapacidad.

La cantidad de tipos de discapacidad y grados de la misma es muy amplia. Este proyecto se centra en la discapacidad para la comunicación. En la actualidad, sólo en España, existen alrededor de 4 millones de personas con algún tipo de discapacidad¹ y, dentro de este grupo, casi 1.600.000 poseen algún tipo de deficiencia en la comunicación¹. Existiendo esta necesidad en la sociedad y teniendo los medios adecuados que ofrecen los avances tecnológicos, este proyecto pretende ofrecer una solución, útil y totalmente adaptada, para una parte de esta población. De esta manera, se intenta mejorar la calidad de vida, tanto de los afectados por alguna discapacidad, como del resto de personas de su entorno.

1 Último boletín publicado por el INE, diciembre de 2008.

El proyecto se basa en el diseño y desarrollo de una aplicación para un interfaz táctil que favorezca la comunicación de personas con diversidad funcional, o con algún tipo de problema motor, con el resto de la gente.

Mediante el uso de este interfaz, el usuario tendrá acceso a menús sencillos e intuitivos que le permitirán dar a conocer de manera rápida y eficaz en cada momento sus necesidades e inquietudes.

Se persigue dar una solución útil, adecuada y adaptada a un problema real, por lo que, desde un principio, se ha contado con el trabajo y asesoramiento de dos instituciones especializadas en discapacidad, Anfas e Isterria. Las siglas de la primera se refieren a "Asociación navarra en favor de las personas con discapacidad intelectual". Se constituyó el 28 de mayo de 1961, siendo distinguida en 1999 por el Gobierno de Navarra con la Medalla de Oro de Navarra por su trayectoria durante sus casi cuarenta años de existencia. Por otro lado, el Centro Isterria es un Centro Integrado de Educación Especial, creado en 1966 y dirigido por Fundación Caja Navarra, en el que se imparten los niveles educativos correspondientes a Educación Básica Obligatoria (6 a 16 años) y Programas de Garantía Social (16 a 21 años), que se concretan en los programas de Iniciación Profesional (ramas agraria y electricidad) y Programas de Tránsito a la vida adulta.

Además, se ha contado con la colaboración de familiares de niños, que poseen algún tipo de discapacidad, que han aportado sus conocimientos, tiempo, ayuda y vivencia personal al proyecto.

Motivación

Las personas con dificultades comunicativas tiene muchos problemas para hacerse entender. Tradicionalmente, los métodos existentes para la comunicación de personas con discapacidad en el habla se centran en el uso de carpetas, con una gran cantidad de imágenes correspondientes, en su mayoría, al código SPC, mediante las cuales el sujeto señala qué es lo que quiere decir. Este proceso está muy limitado y hace complicada la comunicación. Ésto genera en el individuo situaciones de frustración, incluso de agresividad, por saber qué decir y no poder hacerlo.

El objetivo del proyecto es informatizar este proceso a la vez que aprovechar las nuevas tecnologías para crear nuevas formas de comunicación mucho más sencillas para el interlocutor. De este modo, las TIC pasan a ser un método de integración y grandes avances para la discapacidad.

Este tipo de herramientas tienen como propósito facilitar la comunicación con las personas del entorno y así poder integrarse mejor socialmente. Los medios de comunicación son instrumentos en constante evolución y este proyecto permite proporcionar un medio eficaz mediante el cual personas con discapacidad para la comunicación puedan convertirse en comunicadores activos, se les posibilita la capacidad del habla y aumentar la fluidez comunicativa.

Este proyecto posibilita que personas con discapacidad posean una calidad de vida superior, desarrollar su autonomía personal, mejorar su socialización y favorecer su autoestima. Del mismo modo, acerca un nuevo, útil y ventajoso mundo a la discapacidad: las nuevas tecnologías.

En paralelo a la evolución de las nuevas tecnologías, son numerosas las empresas dedicadas a desarrollar, diseñar y vender diferentes componentes software o hardware para mejorar la accesibilidad de las personas con diversidad funcional. Sin embargo, en muchas ocasiones resulta complejo buscar y encontrar una solución adecuada a las necesidades reales del colectivo de la discapacidad.

Por ello, el valor añadido de este proyecto es el asesoramiento y trabajo de profesionales especializados en discapacidad, gracias a los cuales este proyecto contribuye totalmente al paradigma de accesibilidad universal y diseño para todos.

Objetivos

Entre los objetivos que se pretenden alcanzar mediante la realización de este proyecto, cabría destacar:

- Creación de un comunicador sencillo de manejar, intuitivo y funcional para personas con algún tipo de discapacidad en la comunicación. En definitiva, se quiere conseguir una herramienta útil que resuelva los problemas de comunicación y mejore la calidad de vida de los usuarios.
- Integración de las TIC en el mundo de la discapacidad. Con esta idea, se pretende unir los sectores de las nuevas tecnologías y la discapacidad, inicialmente muy separados y así poder ofrecer soluciones tecnológicas que resuelvan problemas concretos.
- Apoyar a la discapacidad no sólo en cuanto a comunicarse se refiere, sino también como una ayuda para romper muchas barreras más (posibilitar introducción en diferentes empleos de personas discapacitadas, nuevo método educativo y de aprendizaje, etc.).

Capítulo 1: Antecedentes

El origen de este proyecto está inmerso en otro denominado "*DISEÑO Y DESARROLLO DE UN INTERFAZ TÁCTIL PARA DISCAPACITADOS CON PROBLEMAS DE COMUNICACIÓN*" desarrollado en la UPNA, Departamento de Ingeniería Eléctrica y Electrónica. El presente proyecto complementa al inicial en diferentes aspectos, como el diseño y la gestión de la base de datos, desarrollo código SPC, así como de funcionalidades extras añadidas al mismo.

En el anteriormente citado proyecto, se decidió tanto el sistema operativo con el que trabajar, como el dispositivo táctil para el cuál se iba a realizar el proyecto. Atendiendo a innovación, disponibilidad, características técnicas, calidad y precio, se tomó la decisión de seleccionar el iPad como herramienta de trabajo. Ello conllevaba un estudio del lenguaje de programación orientado a objetos Objective C e inclusión en el "mundo" Apple.



Figura 1. iPad de Apple.

De este modo, este proyecto partía de un entorno de trabajo ya seleccionado y una herramienta en vía de desarrollo a la que completar. Por ello, en primer lugar se llevó a cabo un estudio del proyecto inicial para determinar los aspectos en los que este proyecto iba a profundizar.

El primer objetivo era conseguir el sistema SPC en base al cual la aplicación para el interfaz táctil iPad funciona. Para ello, se tenían que estudiar las diferentes alternativas ya existentes y decidir cuál iba a ser la propia (*Capítulo 2*).

Posteriormente, era necesaria la creación de una base de datos para la organización, gestión y funcionamiento de la aplicación en cuestión (*Capítulo 4*). La creación de la misma conllevaba su gestión no sólo en cuanto a bases de datos se refiere, sino también en cuanto a Xcode y su correspondiente lenguaje de programación.

Como consecuencia directa del apartado anterior, era necesario un estudio previo del entorno Xcode, su lenguaje Objective C y funcionamiento (*Capítulo 3*). Además, no bastaba sólo con ello, sino que también había que indagar en el campo común entre Xcode y las bases de datos, CoreData (*Capítulo 5*).

Finalmente, había diferentes aspectos de la aplicación aún sin desarrollar y que suponían mejoras notorias para la misma (*Capítulo 6, Estudio y desarrollo menú de configuración de la aplicación*).

Además, simultáneamente al desarrollo de este proyecto, está otro bajo el título "*DISEÑO Y DESARROLLO DE COMPLEMENTOS QUE FACILITEN EL USO DE UN INTERFAZ TÁCTIL PARA DISCAPACITADOS CON PROBLEMAS DE COMUNICACIÓN*", que también completa al anterior. Como consecuencia, todos los aspectos y funcionalidades extra que este proyecto incluye influyen del mismo modo a los otros dos (tanto al ya finalizado como al simultáneo). Por lo que, dada está interacción entre diferentes proyectos, un último aspecto había que cumplir: el trabajo en equipo para la consecución de un objetivo común.

Capítulo 2: Estudio y desarrollo código S.P.C.

2.1 Estudio código S.P.C.

El método utilizado comúnmente para facilitar la comunicación de las personas que poseen algún tipo de discapacidad en el habla es simple y tradicionalmente analógico. Se basa en la utilización de algún tipo de Símbolos Pictográficos para la Comunicación (S.P.C.).

Los Símbolos Pictográficos para la Comunicación representan de una forma bastante clara las palabras y conceptos más habituales en la comunicación cotidiana. Están especialmente indicados para personas que, debido a su discapacidad (Autismo, PCI, Daño Cerebral Sobrevenido,...), tienen limitadas sus competencias comunicativas: manifiestan un nivel de lenguaje simple, un vocabulario limitado y elaboran estructuras de frases muy cortas. Estos símbolos se representan mediante dibujos muy sencillos, representativos e icónicos.



Figura 2. Calendario realizado con pictogramas.

Características principales del S.P.C. :

- Representan las palabras y conceptos más habituales en la comunicación cotidiana del sujeto.
- Está indicado para diferentes grupos de edad y de deficiencias, por su sencillez y fácil representación práctica.
- Se utilizan generalmente dos tamaños: 2,5 cm y 5 cm.
- Son de fácil utilización, ya que están preparados para ser fotocopiados y recortado para elaborar el plafón de comunicación.
- Se agrupan por categorías.
- Las categorías se suelen identificar por colores.
- Es un sistema lógico de disposición de los símbolos en el tablero.
- Son de fácil discriminación entre sí.

La comunicación de las personas con discapacidad se lleva a cabo gracias a estos pictogramas. La manera de utilización de éstos es muy variada, pero principalmente son almacenados de manera rudimentaria en bolsas de plástico y posteriormente colocados sobre tiras de velcro formando frases.

Actualmente, hay diferentes tipos de S.P.C disponibles, pero no hay ningún sistema común estandarizado. Entre los códigos S.P.C más comúnmente utilizados cabría destacar:

ARASAAC

ARASAAC (Portal Aragonés de la Comunicación Aumentativa y Alternativa). Este portal ofrece recursos gráficos y materiales para facilitar la comunicación de aquellas personas con algún tipo de dificultad en este área. Posee una inmensa cantidad de símbolos pictóricos de uso gratuito, accesibles a través de su página corporativa (www.catedu.es/arasaac) y ordenados según diferentes criterios de clasificación y categorías (acciones, acontecimientos, adjetivos, adverbios, alimentación, animales, casa, colegio, colores, cuerpo, herramientas, instrumentos...).



Figura 3. Pictogramas ARASAAC.



Mayer- Johnson (www.mayer-johnson.com/) ofrece una aplicación denominada Boardmaker para la educación, terapia e intervención, entrenamiento del lenguaje, adquisición de vocabulario, evaluación y realización de ejercicios como terapia/aprendizaje.

Este software para PC es básicamente un editor de tableros para facilitar la comunicación alternativa y el aprendizaje de la misma. Permite utilizar diferentes tamaños de símbolos pictográficos. Mediante su librería de símbolos, alrededor de los 3000 (agrupados siguiendo las categorías de SPC) es posible configurar tableros de comunicación, muchos de ellos ya definidos. Se muestra aplicable a todo tipo de discapacidades que presenten trastornos asociados a la comunicación, el lenguaje y el aprendizaje en general.



Figura 4. Pictogramas BOARDMAKER.

2.2 Desarrollo código S.P.C.

Ante la ausencia de un código S.P.C estandarizado, se debía llevar a cabo la decisión del sistema pictórico a utilizar. Dada la multitud de los mismos, en primer lugar, se determinó la elección de alguno de uso gratuito. No obstante, se barajaba la posibilidad de comercializar la aplicación y los sistemas pictográficos dejan de ser de uso gratuito si se emplean con finalidad comercial. Como consecuencia, se decidió crear un nuevo código S.P.C.

Inicialmente, y partiendo como base de colecciones pictográficas ya existentes, se enumeraron una cantidad de 5600 palabras para posibilitar la comunicación.

Éstas se presentaron a los profesionales especializados en discapacidad de Anfas e Isterría y a familiares de niños con problemas en la comunicación, quienes valoraron como una cantidad excesiva de vocabulario, parte del cual, quizás nunca se llegaría ni a entender por el usuario, ni a utilizar.

Como consecuencia, se llevó a cabo un primer filtrado de las 5600 palabras, manteniendo 1800 como las más importantes dado su uso básico en la vida cotidiana.

Un último filtrado fue realizado por los profesionales en discapacidad y familiares, determinando una colección final de 1300 pictogramas. No obstante, dentro de las 5600 palabras iniciales, faltaban algunas de gran relevancia, por lo que elaboraron una lista de los pictogramas que no estaban en la colección de los 1300, pero que querían que estuvieran.

De este modo, la colección final de palabras asciende a 1500 y están agrupadas en tres categorías: personas, verbos y complementos. Clasificación necesaria para el funcionamiento de la aplicación para la comunicación.



Figura 5. Clasificación de los pictogramas.

Para el diseño y creación de los pictogramas que representan a estas palabras, se eligió la utilización del software *Adobe Flash Professional CS5*. Además, esta tarea fue dividida entre los tres proyectandos de los proyectos que completan el diseño y desarrollo de la aplicación general.

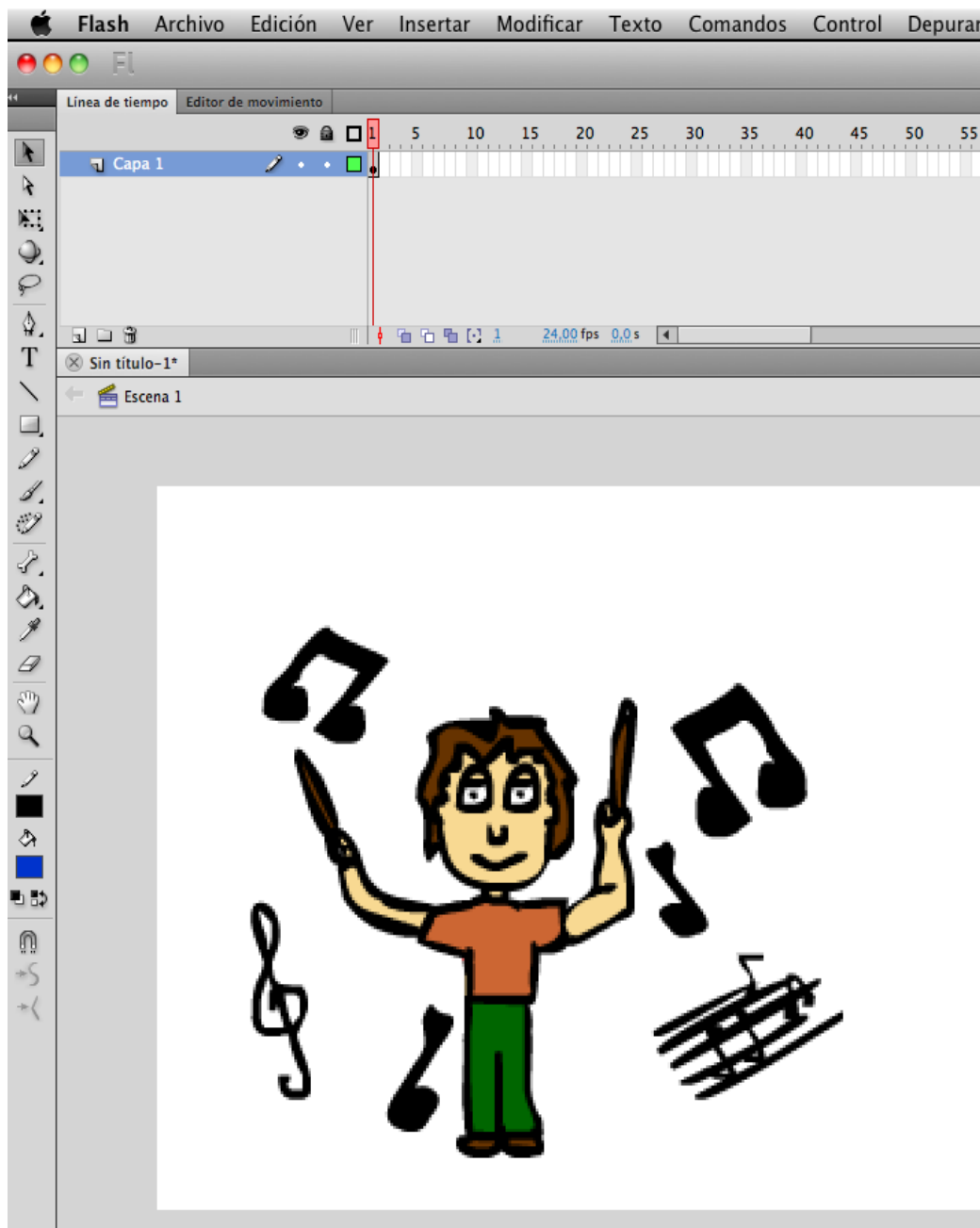


Figura 6. Adobe Flash Professional CS5.

Se trata de una aplicación de creación y manipulación de gráficos vectoriales con posibilidades de manejo de código mediante el lenguaje ActionScript en forma de estudio de animación que trabaja sobre fotogramas y está destinado a la producción y entrega de contenido interactivo para las diferentes audiencias alrededor del mundo sin importar la plataforma. Es actualmente desarrollado y comercializado por Adobe Systems Incorporated y forma parte de la familia Adobe Creative Suite, su distribución viene en diferentes presentaciones, que van desde su forma individual hasta como parte de un paquete.

El código S.P.C creado consta de un total de 1500 pictogramas, que forman la base de datos de la aplicación.

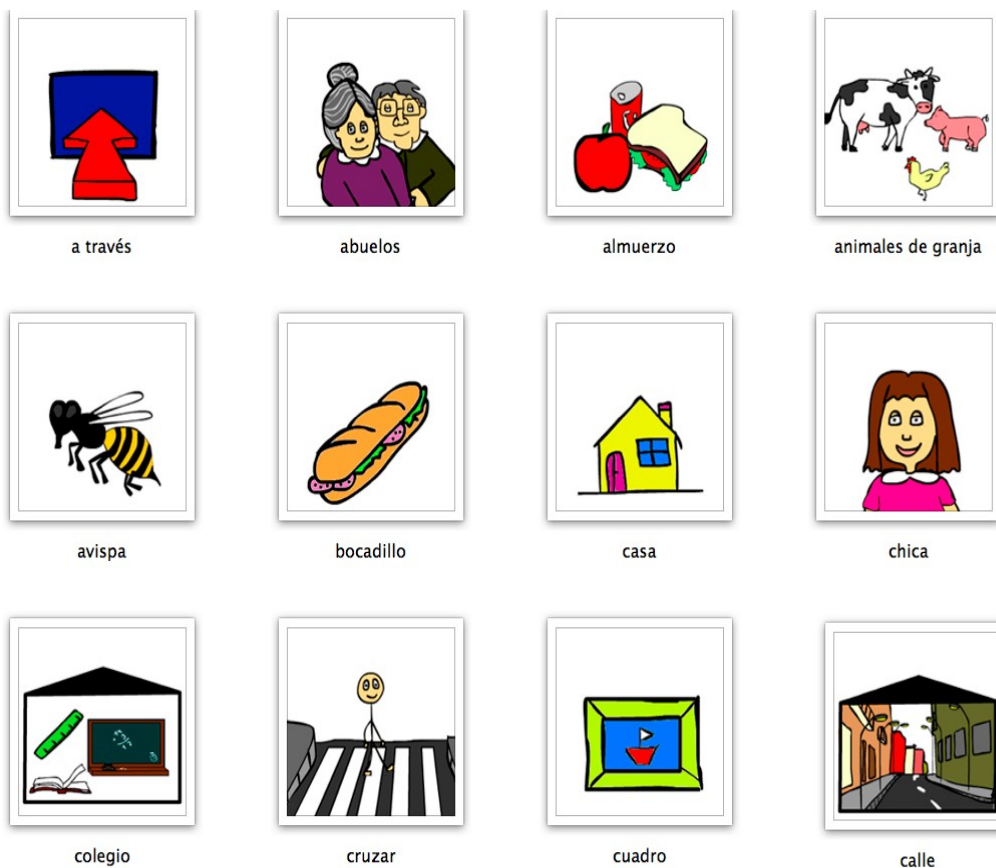


Figura 7. Ejemplos símbolos pictóricos creados.

Capítulo 3: Estudio y desarrollo lenguaje programación

Al contrario que otros sistemas operativos con lenguajes de programación libres, Apple trabaja con Objective C, un lenguaje orientado a objetos que se utiliza, casi únicamente, para los productos de esta marca.

Además de la dificultad inherente a este lenguaje de programación, estaba la complicación añadida de que Apple tiene un sistema de desarrollo de aplicaciones, por el que hay que pagar, y del cual, no hay mucha información.

3.1 Estudio lenguaje programación

3.1.1 El lenguaje: Objective C.

Objective-C es un lenguaje de programación orientado a objetos, creado como un superconjunto de C, pero que implementa un modelo de objetos parecido al de Smalltalk. Originalmente fue creado por Brad Cox y la corporación StepStone en 1980.

En 1988 fue adoptado como lenguaje de programación de NEXTSTEP y en 1992 fue liberado bajo licencia GPL para el compilador GCC. Actualmente, se usa como lenguaje principal de programación en Mac OS X y GNUstep.

Objective-C consiste en una capa muy fina situada por encima de C, y además es un estricto superconjunto de C. Esto es, es posible compilar cualquier programa escrito en C con un compilador de Objective-C, y también puede incluir libremente código en C dentro de una clase de Objective-C.



Figura 8. Logotipo representativo de la programación para iPhone e iPad.

Objective-C requiere que la interfaz e implementación de una clase estén en bloques de código separados. Por convención, la interfaz es puesta en un archivo cabecera y la implementación en un archivo de código; los archivos cabecera, que normalmente poseen el sufijo `.h`, son similares a los archivos cabeceras de C; los archivos de implementación (método), que normalmente poseen el sufijo `.m`, pueden ser muy similares a los archivos de código de C.

3.1.2. Primer paso: Desarrollador.

Al contrario que en otros entornos de programación, Apple no permite realizar una programación libre y una simulación sin problemas en sus dispositivos. Para desarrollar aplicaciones (apps.) es necesario estar registrado en su programa de desarrolladores.

Este programa consiste en registrarse como desarrollador, para lo que hay que pagar 99 dólares, obteniendo de esta forma el paquete de herramientas necesarias para crear las denominadas apps. La duración de este sistema tan novedoso es de un año y es la única manera legal de poder simular las apps. en dispositivos reales.

Este sistema de trabajo tiene una gran ventaja y es que Apple, al saber que eres desarrollador, acepta tus aplicaciones creadas y las comercializa a través de su AppStore, disponible en todos los productos de esta marca. Esta plataforma de comercialización facilita en gran medida la difusión de las apps. creadas.

Cabe destacar también que Apple ofrece un sistema de ayuda al desarrollador, denominado development center, desde donde se puede acceder a manuales de programación, ayuda en línea e, incluso, un foro de desarrolladores donde éstos comparten sus ideas y resuelven dudas.

3.1.3. Segundo paso: Descarga del kit de desarrollador.

Una vez registrado en el programa de desarrolladores es necesario descargarse el SDK (Software Development Kit) de Apple, que consiste en el conjunto de herramientas que pone a disposición de sus desarrolladores para diseñar y programar aplicaciones para sus dispositivos. Las herramientas más utilizadas y necesarias son: *Xcode e Interface Builder*.



Figura 9. Logotipo representativo del SDK de Apple.

Además de esos dos programas, este kit proporciona herramientas de control de la aplicación (app.). Se pueden realizar búsquedas de errores,

revisión de fallos, análisis de eficiencia, creación de gráficas, medición de rendimiento, etc.

Por último, un elemento vital del software de desarrollo de Apple es el simulador. Sin necesidad de poseer un dispositivo móvil real, se pueden probar las aplicaciones creadas en él y comprobar su funcionamiento.

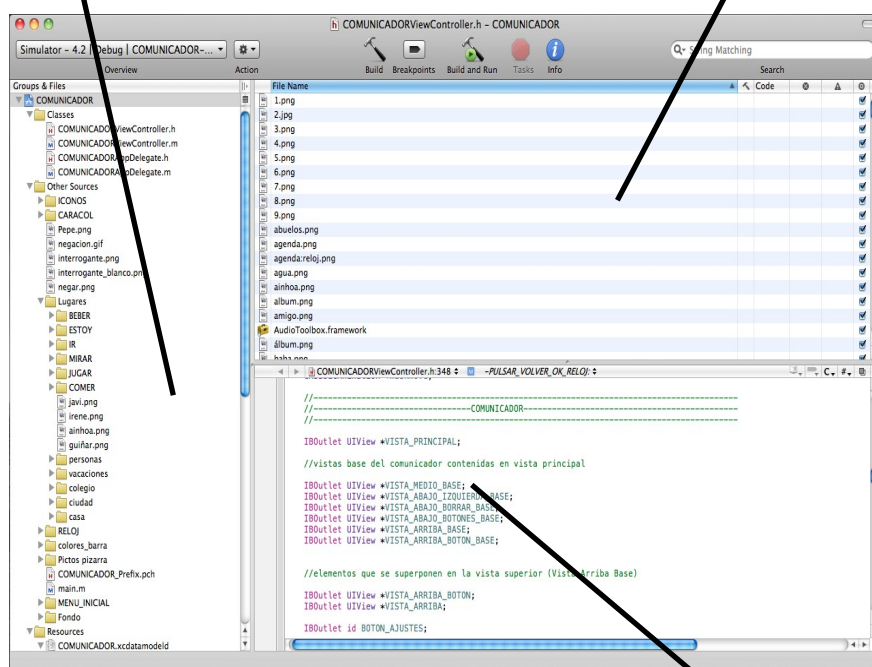
3.1.4. Xcode.

Es el motor de las aplicaciones. Dentro de Xcode es donde se escribe el código fuente que hace que el programa funcione y desde donde se accede al Interface Builder de la app. También se gestionan todos los archivos, librerías y ejecutables que posee la aplicación.

Como se puede ver en la imagen: en la parte izquierda se encuentran todos los archivos relativos a la aplicación; en la parte superior se muestra el contenido de cada carpeta y en la parte inferior se observa una previsualización del elemento seleccionado.

Archivos relacionados con el proyecto(imágenes, ejecutables, etc.)

Contenido de la carpeta COMUNICADOR



Previsualización

Figura 10. Pantalla inicial de xcode correspondiente al proyecto.

En la parte superior se encuentran los botones de compilación (Build), Compilación y simulación (Build and Run), establecimiento de puntos de interrupción (breakpoints) y el botón de ayuda.



Figura 11. Botones de XCode.

Además, se puede elegir si ejecutar el programa en el simulador o en el dispositivo. Para ello, se pueden crear diferentes ejecutables. Hay que tener en cuenta que para simular para iPhone o para iPad, hay que tener ejecutables diferentes creados en el proyecto.



Figura 12. Selección de soporte y archivo ejecutable para la simulación.

En cuanto a programación, Xcode se estructura en dos tipos de archivos: archivo.h y archivo.m. El primero de ellos recoge la declaración de las variables, de las acciones del programa y la de los elementos que van a existir en el Interface Builder.

Los archivos .m incluyen las descripciones de las acciones declaradas anteriormente y la inicialización de las variables que lo necesitan. Además dentro del entorno de Xcode se encuentra el archivo de la base de datos, llamado COMUNICADOR.xcdatamodel.

Cabe resaltar que, en el programa del objeto de este proyecto, hay dos bloques de archivos, cada uno con su elemento .h y .m, se llaman ComunicadorAppDelegate y ComunicadorViewController. El primero es el núcleo de la aplicación y el segundo es el que controla el funcionamiento de ésta.

3.1.5. Interface Builder (IB).

Esta herramienta de programación aporta toda la parte visual de la aplicación. En ella se define cómo va a ser físicamente el diseño, es decir, qué va a ver el usuario final.

Es un editor gráfico que une la parte de código puro con los elementos visuales que van a realizar las acciones descritas. Para que todos estos elementos visuales sepan qué es lo que tienen que hacer y cómo tienen que funcionar, se enlazan con el archivo .m mediante líneas de color azul que unen, por ejemplo, las acciones creadas con el botón encargado de realizarlas.

Físicamente, los archivos que contienen la programación del Interface Builder son los que tienen extensión .xib. En este caso hay dos, MainWindow.xib y COMUNICADORViewController.xib. El primero corresponde al bloque de archivos .h y .m, del COMUNICADORAppDelegate y el segundo al bloque de archivos del COMUNICADORViewController. Uno controla la ventana principal, la orientación y el tipo de interfaz; el segundo controla las vistas creadas.

Dentro del IB se encuentra la ventana de archivos del sistema, donde se encuentran los elementos de la app. (en el gráfico, la ventana de la izquierda); la ventana correspondiente al archivo en el que se está trabajando en el momento (ventana central) y el Inspector, donde se modifican las características de cada elemento (ventana derecha). La ventana negra muestra las acciones relacionadas de cada elemento.

Se puede observar a continuación cómo la línea azul une las acciones y sus respectivos botones.

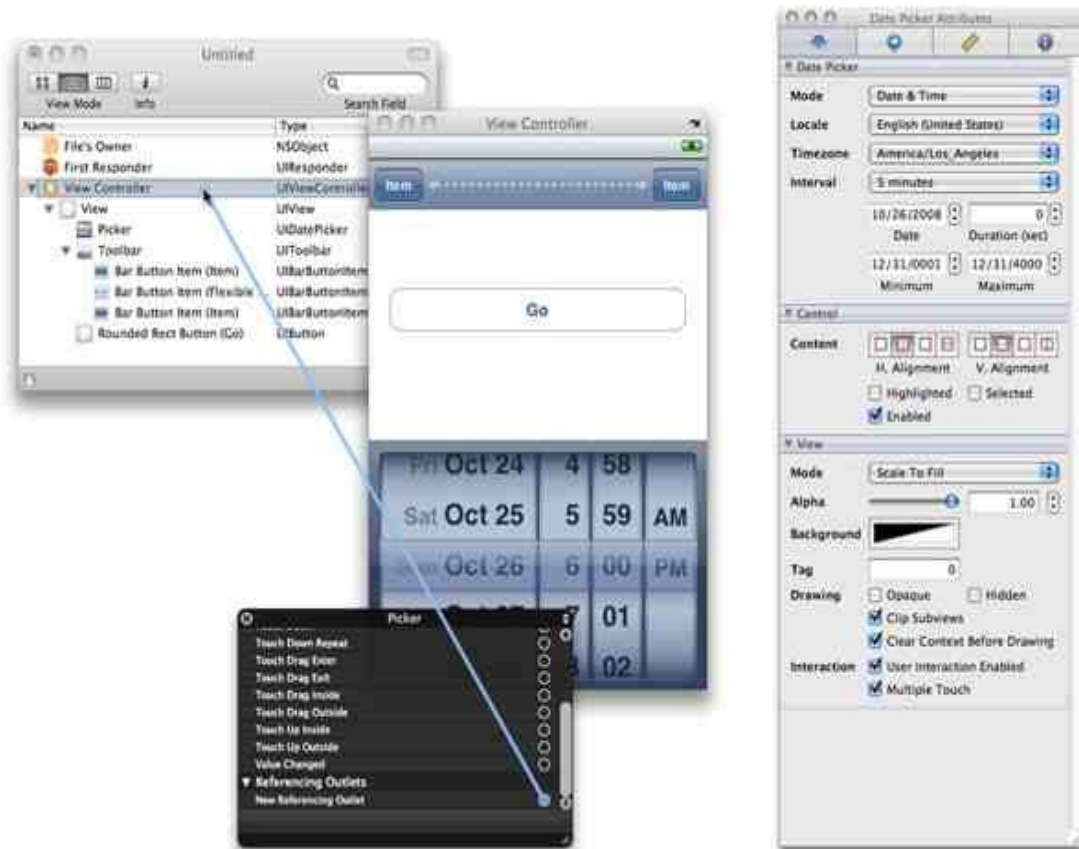


Figura 13. Esquema obtenido de www.projectomega.org.

Por último, un gráfico que define perfectamente el esquema de funcionamiento del SDK de Apple es el siguiente. Se parte de una idea de proyecto, mediante Xcode e Interface Builder se crea el código que genera la app. y se prueba en el simulador del ordenador.

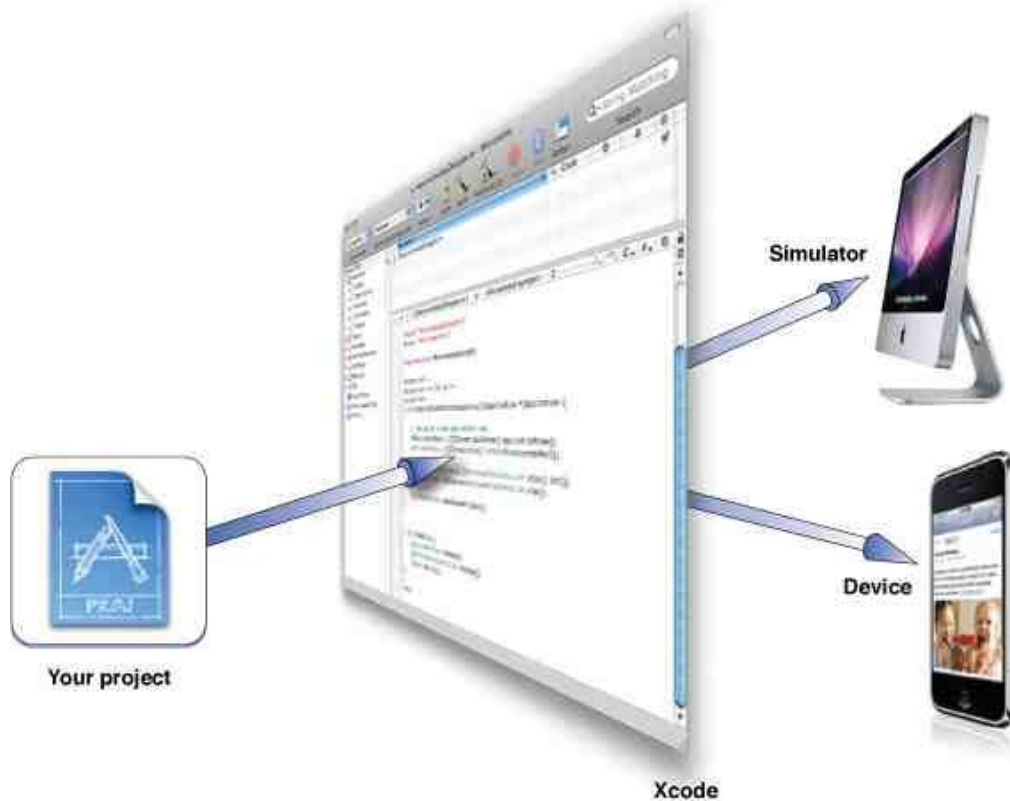


Figura 14. Gráfico obtenido de www.projectomega.org.

Se revisa, se buscan errores y se solucionan. Por último, se simula el programa en el dispositivo y se repite la búsqueda de errores. Una vez terminado todo este proceso, la app. está disponible para ser comercializada.



Figura 15. Gráfico obtenido de www.projectomega.org.

3.2 Desarrollo lenguaje programación

Uno de los objetivos primordiales del proyecto es completar la aplicación para el interfaz táctil iPad ya desarrollada en su primera versión. Esta aplicación consta de un conjunto de herramientas: Comunicador, QuieroEstoy, Pizarra, Cuenta atrás y Álbum.

- Comunicador: herramienta que constituye el ente central de la aplicación. Se basa en el sistema pictórico desarrollado para la formación de frases completas. Con esta aplicación se cubren todas las necesidades comunicativas del usuario.

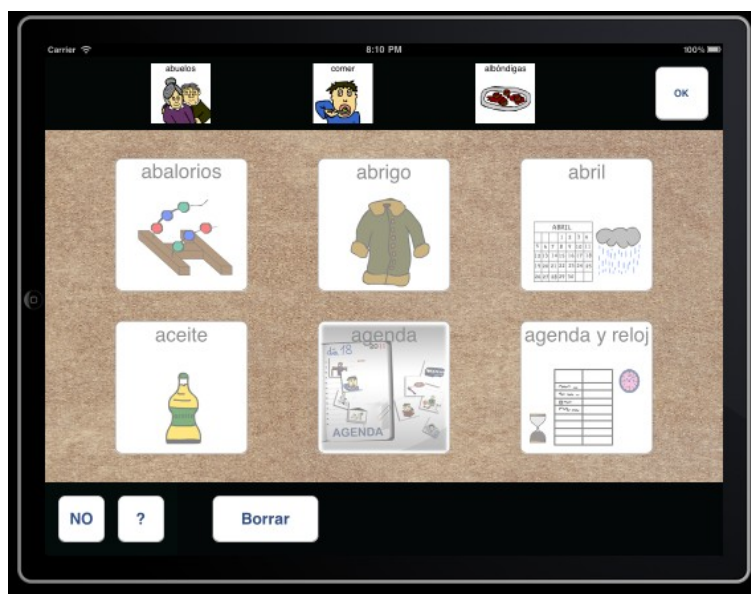


Figura 16. Herramienta Comunicador.

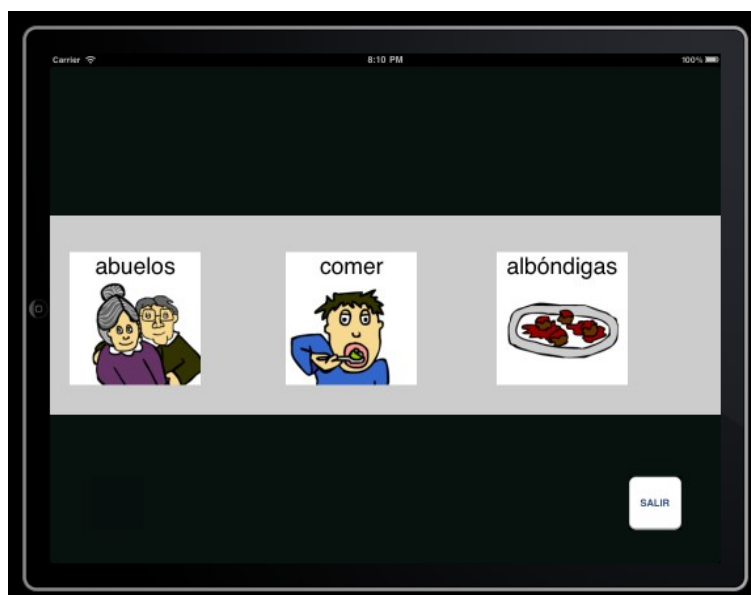


Figura 17. Herramienta Comunicador.

- QuieroEstoy: herramienta de comunicación directa y sencilla. Únicamente consta de los verbos "querer" y "estar" y un vocabulario limitado. Así, el usuario puede formar frases de primera necesidad rápidamente y sin distracciones.

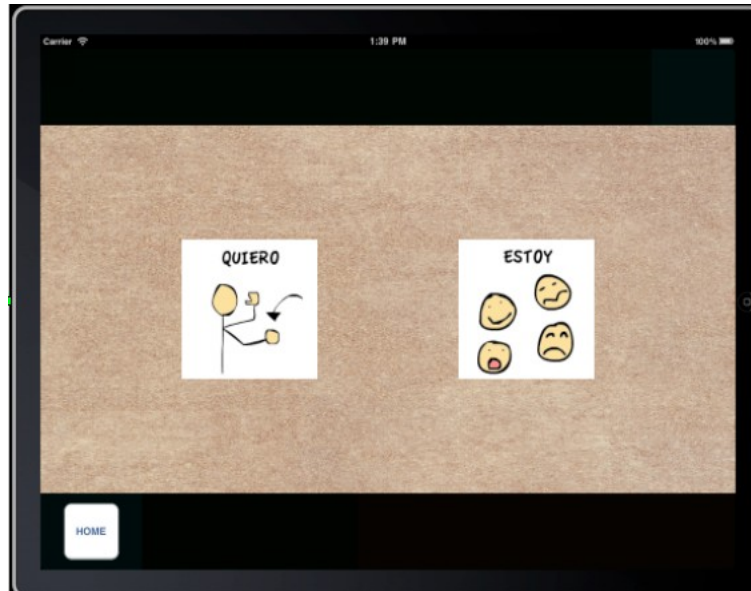


Figura 18. Herramienta QuieroEstoy.

- Pizarra: pizarra sencilla que, además de para pintar con los dedos, se puede utilizar en la generación de plantillas para que el usuario las coloree y así mejore sus habilidades motoras.

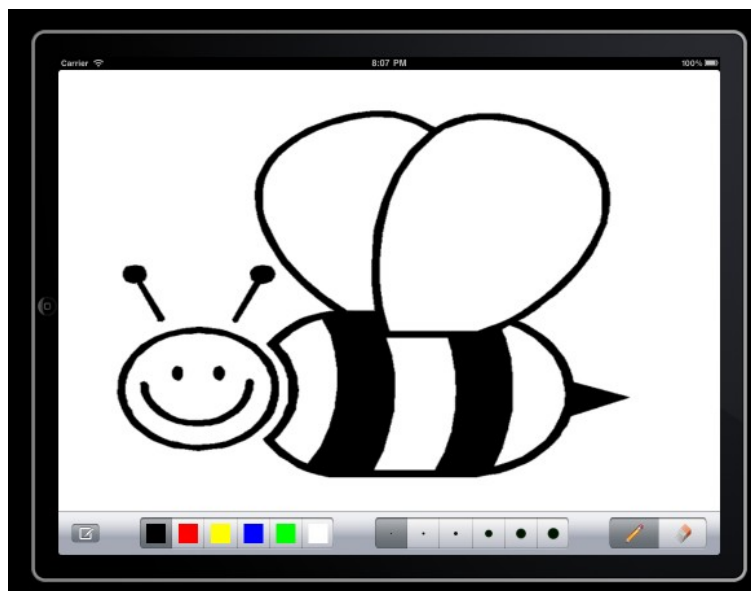


Figura 19. Herramienta Pizarra.

- Cuenta atrás: herramienta para que el usuario tome conciencia del tiempo. Para las personas con algún tipo de discapacidad, el concepto de horas, minutos y segundos es muy abstracto y por tanto esta funcionalidad les ayuda en su percepción temporal.

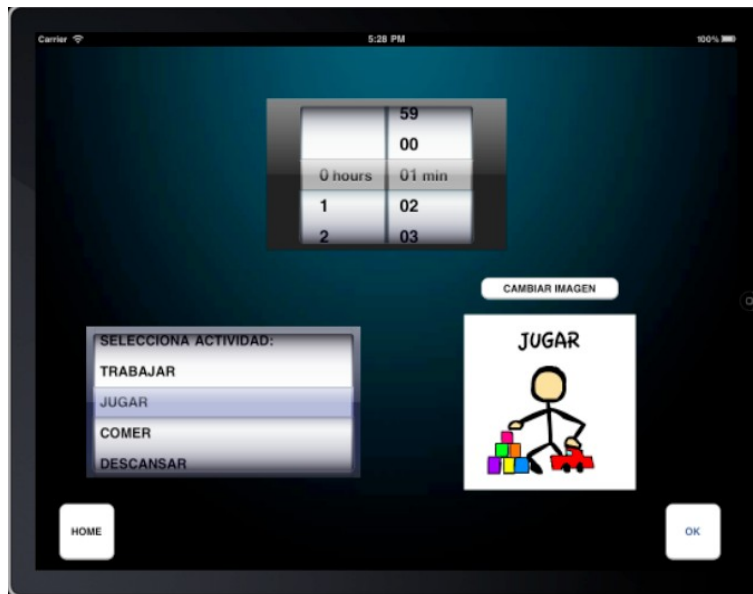


Figura 20. Herramienta Cuenta atrás.



Figura 21. Herramienta Cuenta atrás.

- Álbum: herramienta que sigue la misma línea que el resto, es decir, sencilla e intuitiva. Consta de tres carpetas iniciales, que permiten que los pictogramas e imágenes aparezcan de forma ordenada. La primera de ellas contiene la totalidad de los pictogramas de la base de datos, la segunda las imágenes guardadas en el propio iPad, y la tercera los pictogramas que se eligen como favoritos desde la herramienta Ajustes, posteriormente explicada. Gracias a esta aplicación, el usuario se familiariza con el uso de pictogramas e imágenes, mejorando así su aprendizaje.

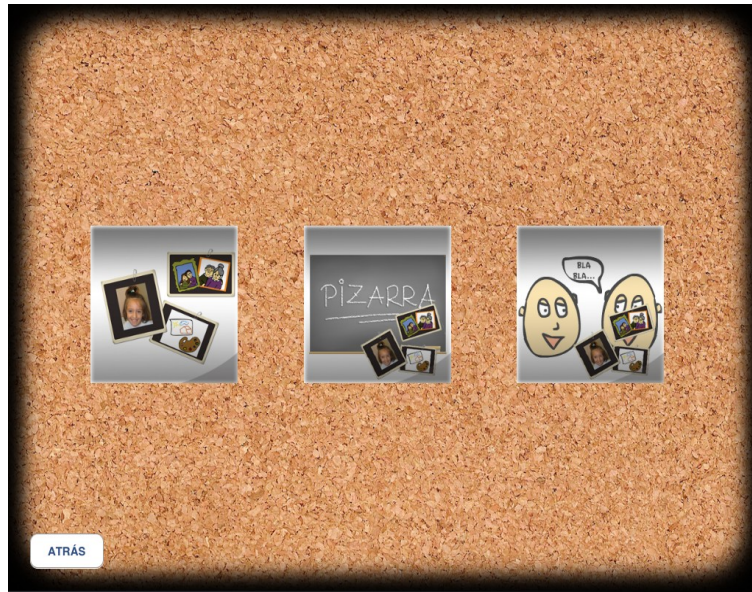


Figura 22. Herramienta Álbum.

El código perteneciente a cada una de estas herramientas forma un único bloque. Puesto que la base de datos es común a todas ellas y las funcionalidades extra que este proyecto incluye afectan a todas por igual, era necesaria una reestructuración y organización del código de programación, haciendo que fuera más sencillo de manejar y eficiente.

Por ello, se realizó un estudio de todo el código ya desarrollado con el fin de prepararlo para su organización y facilitar su manejo. La solución más adecuada era reestructurarlo constituyendo diferentes clases.

El concepto de clase en Objective-C es el mismo que en otros lenguajes de programación que están orientados a objetos. Una clase encapsula parte del estado (propiedades) y el comportamiento (métodos) de un objeto, y forma un bloque básico dentro de una aplicación.

En Objective-C las clases se descomponen en dos partes, que son interfaz e implementación. Esto quiere decir que para crear una clase se tienen que crear dos ficheros con el nombre de la clase pero con diferente extensión, uno con extensión ".h" (para la interfaz, puramente declarativo) y otro con extensión ".m" (para la implementación).

En el fichero ".h" se define toda la estructura de la clase. Se declaran las variables de instancia y los métodos que usará la clase. En este fichero también se definen las propiedades de las variables.


```
// Pizarra.h

#import <UIKit/UIKit.h>
#import <QuartzCore/QuartzCore.h>

@interface Pizarra : UIViewController <UINavigationControllerDelegate, UIImagePickerControllerDelegate> {

    IBOutlet UIView *VISTA_PRIMERA_PIZARRA;
    IBOutlet UIView *MENU_PIZARRA;
    CGPoint ultimoPunto;
    UIImageView *drawImage;
    BOOL userSwiped;
    int userMoved;
    int selectedUnit;
    UISegmentedControl *segmentedControl;
    int selectedUnit1;
    UISegmentedControl *segmentedControl_grosor;
    int selectedUnit2;
    UISegmentedControl *segmentedControl_color;
    float colorRed;
    float colorGreen;
    float colorBlue;
    float lineWidth;
    int menupizarra;
    IBOutlet UIToolbar *barra_pizarra;

    UIPopoverController *popover;
}

@property (nonatomic, retain) IBOutlet UISegmentedControl *segmentedControl;
@property (nonatomic, retain) IBOutlet UISegmentedControl *segmentedControl_grosor;
@property (nonatomic, retain) IBOutlet UISegmentedControl *segmentedControl_color;
@property (nonatomic, retain) UIPopoverController *popover;
@property (nonatomic, retain) UIImage *copyImage;

- (IBAction)saveDrawing;
- (IBAction)cancelDrawing: (id) sender;
- (IBAction)segmentedControlValueChanged;
- (void)cambioHerramienta;
- (IBAction)segmentedControlValueChanged1;
- (void)cambioGrosor;
- (IBAction)segmentedControlValueChanged2;
- (void)cambioColor;

- (IBAction)PULSAR_DESHACER: (id) sender;
- (IBAction)PULSAR_MENU_PIZARRA: (id) sender;
//- (IBAction)PULSAR_VOLVER_MENU_PIZARRA: (id) sender;
- (IBAction)choosePic: (id) sender;

@end
```

Fichero que define la estructura de la clase pizarra.

Variables

Métodos

Figura 23. Código fuente de Pizarra.h

El fichero “.m” contiene la implementación de los métodos declarados en la interfaz. Este fichero debe importar el correspondiente fichero “.h”, y también debe sintetizar las propiedades declaradas en el fichero de interfaz.

```
// Pizarra.m

#import <QuartzCore/QuartzCore.h>
#import "Pizarra.h" → Importación de Pizarra.h

@implementation Pizarra

@synthesize segmentedControl;
@synthesize segmentedControl_grosor;
@synthesize segmentedControl_color;
@synthesize popover;
@synthesize copyImage;

- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil
{
    self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil];
    if (self) {
        // Custom initialization
    }
    return self;
}

- (void)didReceiveMemoryWarning
{
    // Releases the view if it doesn't have a superview.
    [super didReceiveMemoryWarning];

    // Release any cached data, images, etc that aren't in use.
}

#pragma mark - View lifecycle

- (void)viewDidLoad
{
    drawImage = [[UIImageView alloc] initWithImage:nil];
    drawImage.frame = VISTA_PRIMERA_PIZARRA.frame;
    [VISTA_PRIMERA_PIZARRA addSubview:drawImage];
    VISTA_PRIMERA_PIZARRA.backgroundColor = [UIColor whiteColor];
    userMoved = 0;
    menupizarra=0;

    [self.view addSubview:VISTA_PRIMERA_PIZARRA];
    [VISTA_PRIMERA_PIZARRA setBackgroundColor:[UIColor whiteColor]];
    [barra_pizarra addSubview:segmentedControl];
    [barra_pizarra addSubview:segmentedControl_color];
    [barra_pizarra addSubview:segmentedControl_grosor];
    drawImage.image = nil;
    [VISTA_PRIMERA_PIZARRA setNeedsDisplay];
}

```

Implementación de
 los métodos de
 la herramienta pizarra.

Figura 24. Parte del código fuente de Pizarra.m

Atendiendo a este modo de programación, la aplicación completa se ha organizado en diferentes clases (comunicador, pizarra, reloj, ajustes, álbum y personalización). Cada clase consta de sus declaraciones propias y la implementación de sus funciones correspondientes.

Además, puesto que muchas funciones que éstas utilizan son comunes, no hace falta implementarlas una y otra vez, sino que son comunes a todas.

Como consecuencia, se reduce el código notablemente y aumenta la eficiencia en su utilización (es más funcional, mayor organización mejorando el trabajo de programar en equipo, ante error o modificación en una de las clases no afecta al resto, eliminación de redundancia ante mismas funciones...).

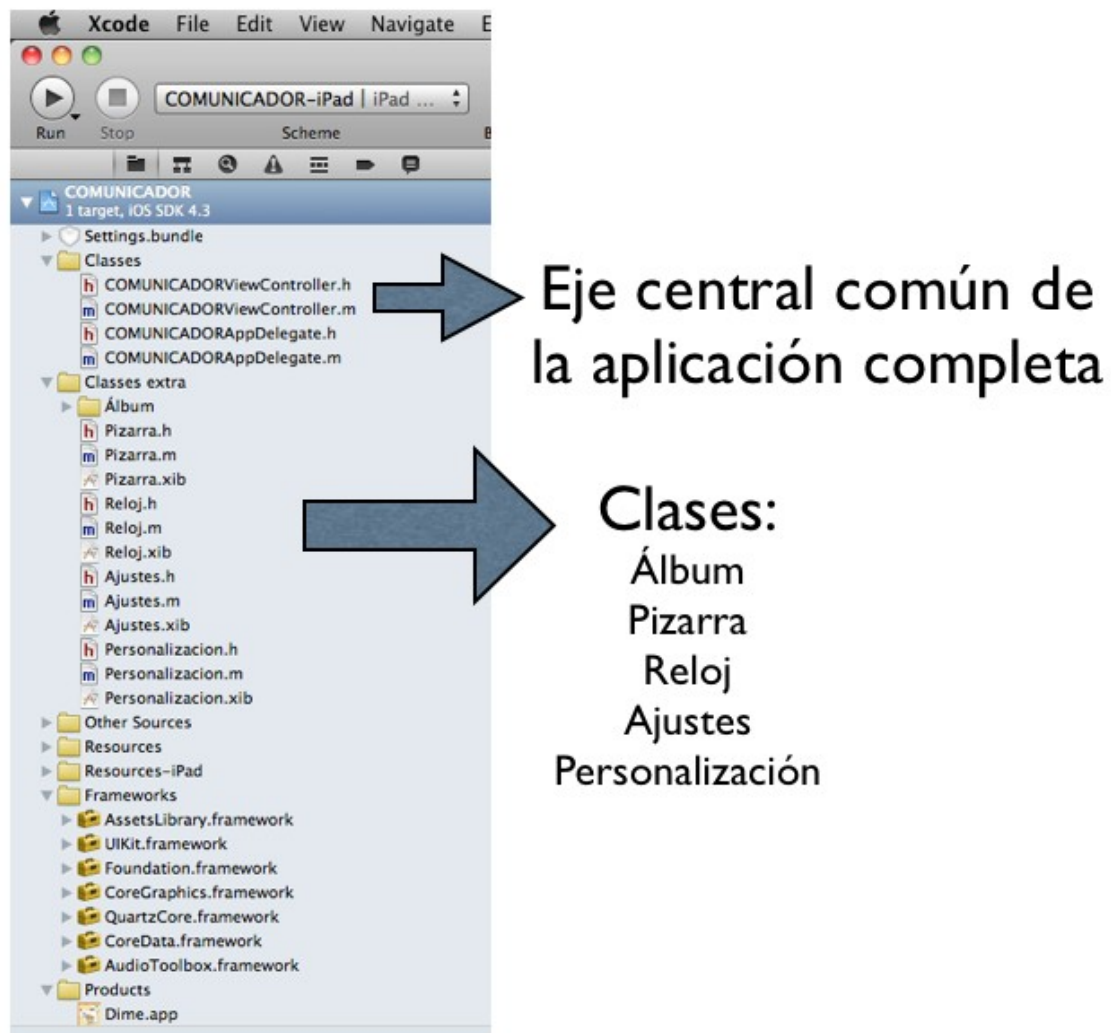


Figura 25. Organización de la aplicación basada en clases.

Capítulo 4: Estudio y desarrollo B.B.D.D.

4.1 Estudio base de datos

Todo programador que acumule datos para su desarrollo software sabe lo complicado que es organizarlos y explotarlos eficientemente. De hecho, manejar bien la información es tan importante o más que la propia información.

En la sociedad actual la información crece a un ritmo vertiginoso. Gracias a las nuevas tecnologías se dispone de más canales para su transmisión y los nuevos soportes facilitan su registro, almacenamiento y recuperación. Sin embargo, resulta difícil dar con el método adecuado y la gestión de la información suele estar muy lejos de ser óptima.

Consecuentemente, surge la necesidad de contar con un sistema de administración para controlar la información. El concepto de base de datos generalmente está relacionado con el de red. "Sistema de información" es el término general utilizado para la estructura global que incluye todos los mecanismos para compartir y gestionar datos.

En el caso concreto de este proyecto, la necesidad de un sistema para la gestión de información surge al pasar de utilizar un número reducido de pictogramas, en la versión inicial de la aplicación para la comunicación, a gestionar más de mil en la versión final. Esto implica un diseño más eficiente de la estructura interna del comunicador.

En la versión inicial, cada bloque de pictogramas estaba contenido en una vista diferente, y cada botón asociado a cada uno de ellos tenía su definición propia en el ViewController. Si se realizaba la misma estructura de código, el resultado final iba a ser un programa totalmente redundante y el código iba a alcanzar dimensiones excesivas. Por ello, era necesaria la creación de una base de datos, gracias a la cual el programa fuera más eficiente, rápido y completo.

```
//Botones Comer
- (IBAction) PULSAR_PICT07: (id) sender;
- (IBAction) PULSAR_PICT08: (id) sender;
- (IBAction) PULSAR_PICT09: (id) sender;
- (IBAction) PULSAR_PICT010: (id) sender;
- (IBAction) PULSAR_PICT011: (id) sender;
- (IBAction) PULSAR_PICT012: (id) sender;
- (IBAction) PULSAR_PICT017: (id) sender;
- (IBAction) PULSAR_PICT018: (id) sender;
- (IBAction) PULSAR_PICT019: (id) sender;
- (IBAction) PULSAR_PICT020: (id) sender;
- (IBAction) PULSAR_PICT021: (id) sender;
- (IBAction) PULSAR_PICT022: (id) sender;
|
//Botones Beber
- (IBAction) PULSAR_PICT023: (id) sender;
- (IBAction) PULSAR_PICT024: (id) sender;
- (IBAction) PULSAR_PICT025: (id) sender;
- (IBAction) PULSAR_PICT026: (id) sender;
- (IBAction) PULSAR_PICT027: (id) sender;
- (IBAction) PULSAR_PICT028: (id) sender;
|
//Botones Ver
- (IBAction) PULSAR_PICT029: (id) sender;
- (IBAction) PULSAR_PICT030: (id) sender;
- (IBAction) PULSAR_PICT031: (id) sender;
```

Figura 26. Muestra de la declaración de las acciones utilizadas en el primer diseño.

En lo concerniente a las bases de datos y su gestión, son numerosos los avances y tecnologías desarrolladas para la creación y manejo de cómputo y de esta forma ahorrar trabajo y tiempo en la producción de software.

Una base de datos (BD) es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. En la actualidad, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital.

Existen programas denominados sistemas gestores de bases de datos (SGBD), que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Los SGBD son un conjunto de programas que se encargan de manejar la creación y todos los accesos a las bases de datos. El objetivo principal del sistema gestor de base de datos es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera que sea tanto práctica como eficiente. Su utilización se ha incrementado en los últimos años de forma drástica.

Las aplicaciones de las bases de datos más usuales son para la gestión de empresas e instituciones públicas. También son ampliamente utilizadas en entornos científicos con el objeto de almacenar la información experimental.

Las bases de datos pueden clasificarse de varias maneras, de acuerdo al contexto que se esté manejando, la utilidad de las mismas o las necesidades que satisfagan, de entre otras muchas formas.

1- Según la variabilidad de los datos almacenados:

Bases de datos estáticas: bases de datos de sólo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones y tomar decisiones.

Bases de datos dinámicas: bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización, borrado y adición de datos, además de las operaciones fundamentales de consulta.

2- Según el contenido:

Bases de datos bibliográficas: contienen un subrogante (representante) de la fuente primaria, que permite localizarla. Un registro típico de una base de datos bibliográfica contiene información sobre el autor, fecha de publicación, editorial, título, edición, de una determinada publicación,

etc. Puede contener un resumen o extracto de la publicación original, pero nunca el texto completo.

Bases de datos de texto completo: almacenan las fuentes primarias, como por ejemplo, todo el contenido de todas las ediciones de una colección de revistas científicas.

Directorios: por ejemplo son las guías telefónicas en formato electrónico.

Bases de datos de información químico-biológica: bases de datos que almacenan diferentes tipos de información proveniente de la química, las ciencias de la vida o médicas.

Además de la clasificación según la función de las bases de datos, éstas también se pueden clasificar de acuerdo a su modelo de administración de datos.

Un modelo de datos es básicamente una descripción de algo conocido como contenedor de datos (algo en donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no son físicos, sino abstracciones que permiten la implementación de un sistema eficiente de base de datos. Por lo general, se refieren a algoritmos y conceptos matemáticos. Algunos modelos con frecuencia utilizados en las bases de datos son:

Bases de datos jerárquicas: bases de datos que, como su nombre indica, almacenan su información en una estructura jerárquica. En este modelo los datos se organizan en una forma similar a un árbol, en donde un nodo padre de información puede tener varios hijos. El nodo que no tiene padres es llamado raíz, y a los nodos que no tienen hijos se los conoce como hojas.

Las bases de datos jerárquicas son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos permitiendo crear estructuras estables y de gran rendimiento.

Base de datos de red: modelo ligeramente distinto al jerárquico. Su diferencia fundamental es la modificación del concepto de nodo: se permite que un mismo nodo tenga varios padres (posibilidad no permitida en el modelo jerárquico).

Es una gran mejora con respecto al modelo jerárquico, ya que ofrece una solución eficiente al problema de redundancia de datos. Aún así, la dificultad que significa administrar la información en una base de datos de red ha significado que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales.

Bases de datos transaccionales: bases de datos cuyo único fin es el envío y recepción de datos a grandes velocidades, son muy poco comunes y están dirigidas por lo general al entorno de análisis de calidad, datos de producción e industrial. Su fin único es recolectar y recuperar los datos a la mayor velocidad posible, por lo tanto la redundancia y duplicación de información no es un problema como con las demás bases de datos.

Bases de datos relacionales: modelo utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Tras ser postulados sus fundamentos en 1970 por Edgar Frank Codd, de los laboratorios IBM en San José (California), no tardó en consolidarse como un nuevo paradigma en los modelos de base de datos. Su idea fundamental es el uso de relaciones. Estas relaciones pueden considerarse en forma lógica como conjuntos de datos llamados tuplas. Pese a que ésta es la teoría de las bases de datos relacionales creadas por Codd, la mayoría de las veces se conceptualiza de una manera más fácil de imaginar. Esto es pensando en cada relación como si fuese una tabla que está compuesta por registros (las filas de una tabla), que representarían las tuplas, y campos (las columnas de una tabla).

En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario esporádico de la base de datos. La información puede ser recuperada o almacenada mediante consultas que ofrecen una amplia flexibilidad y poder para administrar la información.

El lenguaje más habitual para construir las consultas a bases de datos relacionales es SQL, Structured Query Language o Lenguaje Estructurado de Consultas, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales.

Bases de datos multidimensionales: bases de datos ideadas para desarrollar aplicaciones muy concretas, como creación de Cubos OLAP. Básicamente no se diferencian demasiado de las bases de datos relacionales. La diferencia está más bien a nivel conceptual, ya que en las bases de datos multidimensionales los campos o atributos de una tabla pueden ser de dos tipos, o bien representan dimensiones de la tabla, o bien representan métricas que se desean estudiar.

Bases de datos orientadas a objetos: este modelo, bastante reciente, y propio de los modelos informáticos orientados a objetos, trata de almacenar en la base de datos los objetos completos (estado y comportamiento).

Una base de datos orientada a objetos es una base de datos que incorpora todos los conceptos importantes del paradigma de objetos. En bases de datos orientadas a objetos, los usuarios pueden definir operaciones sobre los datos como parte de la definición de la base de datos. Una operación (llamada función) se especifica en dos partes. La

interfaz (o signatura) de una operación incluye el nombre de la operación y los tipos de datos de sus argumentos (o parámetros). La implementación (o método) de la operación se especifica separadamente y puede modificarse sin afectar la interfaz. Los programas de aplicación de los usuarios pueden operar sobre los datos invocando a dichas operaciones a través de sus nombres y argumentos, sea cual sea la forma en la que se han implementado. Esto podría denominarse independencia entre programas y operaciones.

Bases de datos documentales: permiten la indexación a texto completo, y en líneas generales realizar búsquedas más potentes. Taurus es un sistema de índices optimizado para este tipo de bases de datos.

Bases de datos deductivas: es un sistema de base de datos pero con la diferencia de que permite hacer deducciones a través de inferencias. Se basa principalmente en reglas y hechos que son almacenados en la base de datos. Las bases de datos deductivas son también llamadas bases de datos lógicas, a raíz de que se basa en lógica matemática. Este tipo de base de datos surge debido a las limitaciones de la Base de Datos Relacional de responder a consultas recursivas y de deducir relaciones indirectas de los datos almacenados en la base de datos.

Actualmente, de entre la gran cantidad de aplicaciones para la creación y gestión de bases de datos existentes se pueden destacar las de código libre (MySQL, PostgreSQL, Firebird, SQLite, Apache Derby...), las que requieren una licencia comercial (Dbase, FileMaker, Visual FoxPro, Informix, InterBase, EDeveloper, Microsoft Access, Microsoft SQL Server, NexusDB, Oracle...), así como las que se pueden usar en forma de software de instalación, u otras que se utilizan desde un navegador predeterminado.

4.2 Desarrollo base de datos

Una vez estudiados los diferentes tipos de bases de datos y sus gestores, había que llevar a cabo la elección de la base de datos más adecuada para la aplicación del proyecto.

4.2.1 Decisión base de datos

Puesto que el entorno de trabajo es Apple, se hizo un estudio de las diferentes aplicaciones existentes basadas en bases de datos desarrolladas en este ámbito. La mayoría de ellas utilizan SQLite, incluyendo Apple Mail y el gestor de RSS que se distribuye con OS X. El software Aperture de Apple también guarda la información de las imágenes en una base de datos SQLite, utilizando la API Core Data.

Consecuentemente, y teniendo también en cuenta que se trata de una aplicación nativa sin servidor intermediario y que el tipo de datos en base al cual trabaja es una colección de imágenes y atributos asociados a las mismas, se eligió SQLite como tipología de base de datos.



SQLite es un sistema de bases de datos relacional, contenida en una biblioteca en C. SQLite es un proyecto de dominio público creado por D. Richard Hipp. A diferencia de los sistemas de gestión de bases de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo.

El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos. El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos) son guardados como un sólo fichero estándar en la máquina host. Este diseño simple se logra bloqueando todo el fichero de base de datos al principio de cada transacción. En su versión 3, SQLite permite bases de datos de hasta 2 Terabytes de tamaño y también permite la inclusión de campos tipo BLOB.

Se trata de un autónomo, cero configuración y sistema de base de datos de transacciones de SQL que hace que la gestión de la base de datos sea simple y eficaz. Es muy rápido y la ventaja fundamental es que permite utilizar un amplio subconjunto del lenguaje estándar SQL. SQLite destaca también por su versatilidad. Combina el motor y el interfaz de la base de datos en una única

biblioteca, y almacena los datos en un único archivo de texto plano. Esto hace que cada usuario pueda crear tantas bases de datos como desee sin la necesidad de la intervención de un administrador de bases de datos que gestione los espacios de trabajo, usuarios y permisos de acceso. El hecho de almacenar toda la base de datos en un único archivo, facilita la portabilidad de los datos, y solamente tiene la restricción del espacio de disco asignado al usuario.

Su potencia se basa fundamentalmente en la simplicidad, lo que hace que no sea una buena solución en entornos de tráfico muy elevado y/o alto acceso concurrente a datos.

4.2.2 Creación base de datos

Existen varias formas para la creación y gestión de bases de datos SQLite, entre las que destacan:

- Aplicaciones web.
- Desde la consola de usuario.

Buscando simplicidad y rapidez en el proceso, se determinó la utilización de una aplicación web, concretamente SQLite Manager.



SQLite Manager es un SQLite gestor de bases de datos gratuito como complemento para Firefox. Al ofrecer el software como una extensión de Firefox, SQLite Manager está disponible en diferentes plataformas y se caracteriza por no necesitar ningún servidor y su acceso directo al fichero de la base de datos. Utiliza la librería SQLite integrada en su aplicación de hospedaje (Firefox), y por lo tanto compatible con las versiones de los archivos de base de datos con el apoyo de la biblioteca de Firefox.

Permite al usuario ver el contenido de una o más bases de datos SQLite, gestionar tablas, índices, vistas, ver y administrar los registros, construir una consulta SQL y ejecutarla, exportar datos a CSV o XML, visualizar el contenido de tablas existentes, modificar dicho contenido...

Para la instalación de este software, en primer lugar, hay que descargar e instalar Firefox y, a continuación, en la pestaña Herramientas, seleccionar Complementos. Posteriormente, se busca, de entre los complementos existentes para Firefox, el complemento SQLite Manager y se instala.



Figura 27. Sqlite Manager.

4.2.3 Desarrollo base de datos

El primer diseño de la base de datos arrojaba una única tabla excel donde las filas correspondían a los pictogramas y las columnas a sus atributos. Más o menos, se tenían unas 1500 filas y unas 60 columnas, lo que conllevaba una tabla enorme e impracticable.

Las primeras pruebas, aunque fueron exitosas, dejaban constancia de que esta única tabla no era más que un parche, un elemento ajeno a la programación de Apple, y que era necesario buscar algo directamente relacionado con Xcode.

ATRIBUTOS			
NOMBRE			Querer
CARACTERISTICAS			
IDENTIFICADOR			1'2'3'4
IMAGEN			quere.jpg
COLOR			verde,azul
PERSONA			primerapersona, segunda...
VERBO			si/no
SUSTANTIVO			si/no
ADJETIVO (descriptivo tras verbos ser/estar)			si/no
	COLORES (colores + rubio/moreno color pelo)		si/no
	SABORES		si/no
	TEMPERATURA (CALOR/FRIO)		si/no
	ESTADOS DE ÁNIMO/SENTIMIENTOS		si/no
	CONDUCTA (BIEN/MAL)		si/no
	ADJETIVO/ADVERBIO (ALTO/GORDO/GRANDE/MUCHO)		si/no
CONTADOR			Para ver el numero de veces que se utiliza
AUDIO			si/no
TAMAÑO X			
TAMAÑO Y			
MAYUSCULA			si/no
MINUSCULA			si/no
TIPO LETRA			
GENERO			
OBJETO			si/no
LUGAR	LUGAR		si/no
	CIUDAD (PAMPLONA/MADRID)		si/no
	MAR		si/no
	ZOO		si/no
	GRANJA		si/no
	CAMPO		si/no
	PUEBLO	ZUBIRI	si/no
		PERALTA/CAMPILLO	si/no
	CASA	CASA	si/no
		COCINA	si/no
		DORMITORIO	si/no
		BAÑO	si/no
		SALON	si/no
		COMEDOR	si/no
	COLEGIO	COLEGIO	si/no
		ISTERRIA	si/no
		CLASE	si/no
		COMEDOR	si/no
		PISCINA	si/no
		GIMNASIO	si/no
		SALÓN	si/no
		ENFERMERÍA	si/no
		PARQUE	si/no
		PANTALLA DIGITAL	si/no
		JESUITAS	si/no
		ANFAS	si/no
		ASPACE	si/no
	PARQUE		si/no
	TIENDA	TIENDA	si/no
		FRUTERIA	si/no
		PANADERIA	si/no
		CARNICERIA	si/no
		SUPERMERCADO	si/no
		PESCADERIA	si/no
		ROPA/ZAPATOS	si/no
	HOTEL		si/no
	BAR		si/no
	RESTAURANTE		si/no
	CINE/TEATRO		si/no
	PISCINA		si/no
	POLIDEPORTIVO		si/no
	CENTRO SALUD		si/no
	AEROPUERTO		si/no
	ESTACION TREN		si/no
	IGLESIA		si/no
MEDIO TRANSPORTE			si/no
PERSONA (SER HUM/PERSONAS (OTROS)			si/no
	FAMILIA		si/no
	AMIGOS		si/no
	PROFESORES		si/no
	PROFESIONES (POLICIA, BOMBERO;CARNICERO...)		si/no
ANIMAL			si/no
OCIO/JUEGOS			si/no
COMIDA			si/no
	DESAYUNO		si/no
	COMIDA		si/no
	MERIENDA		si/no
	CENA		si/no
	GOLOSINAS	AQUÍ VAN CHUCHES, HELADO...EL REÍ	si/no
BEBIDA			si/no
ROPA (VESTIDO + CALZADO)			si/no
CUERPO			si/no
TIEMPO	TIEMPO (HOY,MAÑANA ,AYER,VACACIONES,NAVIDAD,SEMANA SAsi/no		si/no
	DIAS SEMANA		si/no
	MESES AÑO		si/no
	ESTACIONES AÑO		si/no
	CLIMA (SOL,LLUVIA,NUBES, VIENTO,NIEVE)		si/no
NUMEROS			si/no

Figura 28. Listado inicial atributos.

Para la redefinición de la base de datos, se atendió a la forma en que la aplicación desarrollada hace uso de los pictogramas.

El primer filtrado a la hora de crear una frase es siempre el posicional. Primero hay que seleccionar uno de los cuatro lugares principales: Casa, Ciudad, Colegio, Vacaciones.

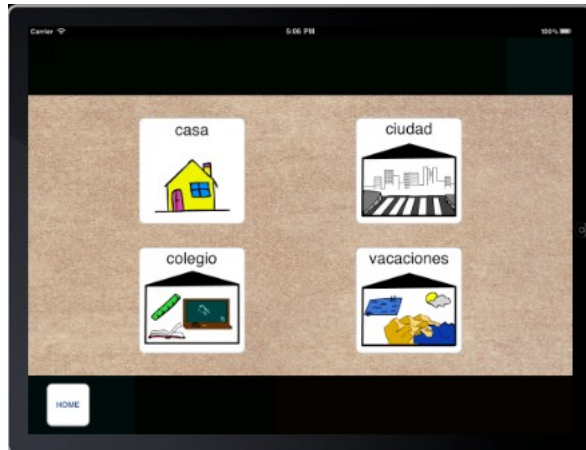
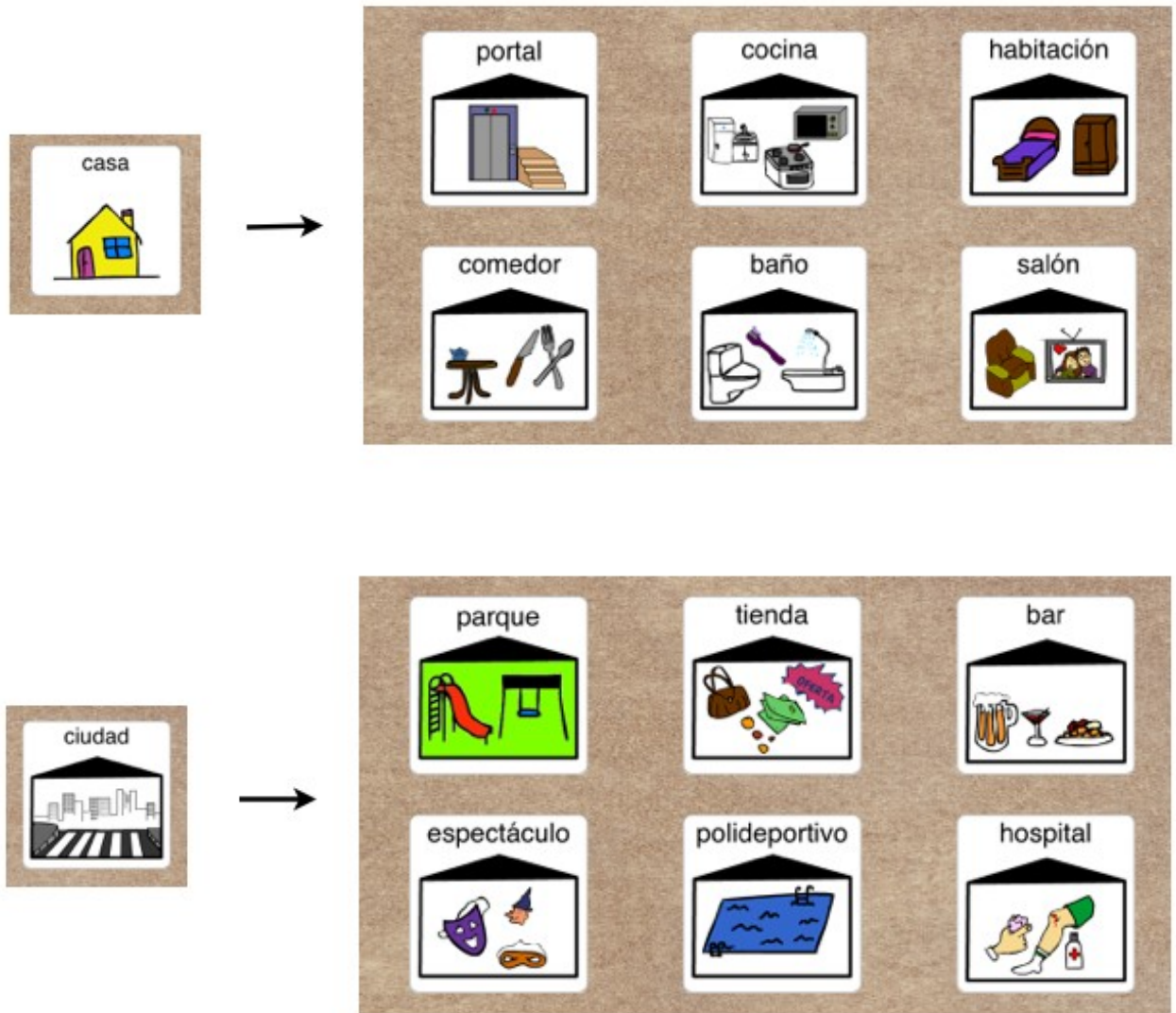


Figura 29. Imagen principal del comunicador, con los 4 lugares iniciales.

Una vez se ha escogido el lugar en cuestión, aparecen seis sublugares asociados al anterior. La relación de lugares y sublugares es la siguiente:

Casa	Ciudad	Colegio	Vacaciones
Portal	Parque	Parque	Piscina
Cocina	Tienda	Isterria	Playa
Habitación	Bar	Jesuitas	Pueblo
Comedor	Espectáculo	Anfas	Granja
Baño	Polideportivo	Piscina	Zoo
Salón	Hospital	Gimnasio	Campo

Figura 30. Tabla que relaciona lugares y sublugares.



Figuras 31 y 32. Relación gráfica de "casa" y "ciudad" con sus respectivos sublugares.

Una vez seleccionado el sublugar el que se encuentra el usuario, la frase que se va a formar se estructura de la siguiente forma: persona, verbo y complemento. Tras la selección de la persona, aparecen los verbos y posteriormente los complementos.

Para conseguir el funcionamiento anterior, la base de datos establece relaciones entre los cuatro lugares principales y sus respectivos sublugares. También relaciona cada uno de los pictogramas con los sublugares en los que tiene que aparecer, además de determinar si se trata de una persona, verbo o complemento. Para ello, se establece una base de datos basada en el modelo entidad-relación.

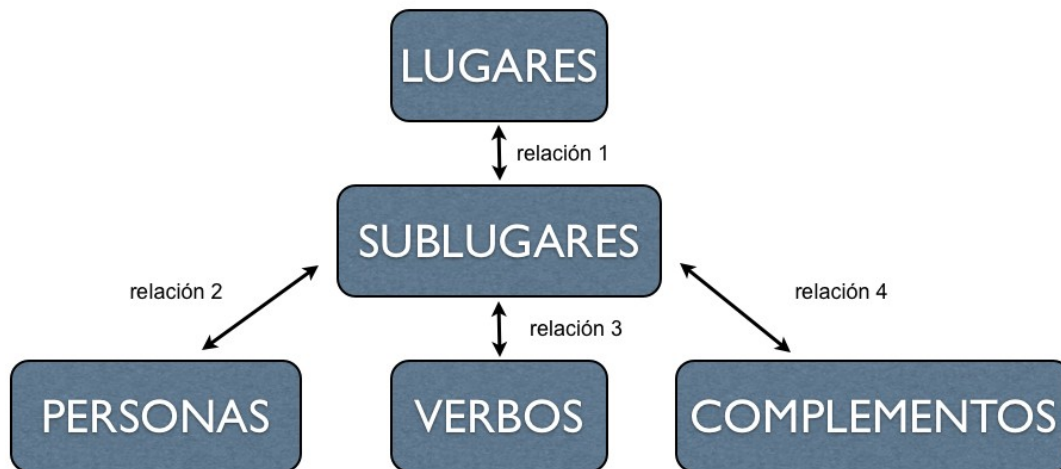


Figura 33. Tabla entidades y relaciones de la base de datos.

Se crean cinco entidades:

- 1- Lugar
- 2- Sublugar
- 3- Personas
- 4- Verbos
- 5- Complementos

Cada una de ellas forma una tabla que contiene como filas los pictogramas asociados a su entidad y todas tienen la misma estructura interna, es decir, idénticos atributos como columnas: primary-key para su identificación, nombre minúsculas castellano, nombre mayúsculas castellano, nombre minúsculas inglés, nombre mayúsculas inglés, contador frecuencia de uso para el filtrado, atributos activo y nuevo para la personalización de las aplicaciones y el atributo personalizar álbum para la selección de los pictogramas favoritos.

TABLE		ZSUBLUGAR	Buscar		Mostrar Todo		Añadir		Duplicar		Editar	
ZNOMBRE	ZNOMBRE_MAY	ZNOMBRE_ING	ZNOMBRE_MAY_ING	ZCONTADOR	ZPK	ZNUEVO	ZACTIVO	ZALBUM_PERSONALIZADO				
anfas	ANFAS	Anfas	ANFAS	0	1	0	1	0				
baño	BAÑO	bathroom	BATHROOM	0	2	0	1	0				
bar	BAR	bar	BAR	0	3	0	1	0				
calle	CALLE	street	STREET	0	4	0	1	0				
campo	CAMPO	field	FIELD	0	5	0	1	0				
cocina	COCINA	kitchen	KITCHEN	0	6	0	1	0				
comedor	COMEDOR	dining room	DINING ROOM	0	7	0	1	0				
espectáculo	ESPECTÁCULO	show	SHOW	0	8	0	1	0				
gimnasio	GIMNASIO	gym	GYM	0	9	0	1	0				
granja	GRANJA	farm	FARM	0	10	0	1	0				
habitación	HABITACIÓN	room	ROOM	0	11	0	1	0				
hospital	HOSPITAL	hospital	HOSPITAL	0	12	0	1	0				
isterria	ISTERRIA	Isterria	ISTERRIA	0	13	0	1	0				
jesuitas	JESUITAS	Jesuitas	JESUITAS	0	14	0	1	0				
parque	PARQUE	park	PARK	0	15	0	1	0				
piscina	PISCINA	pool	POOL	0	16	0	1	0				
playa	PLAYA	beach	BEACH	0	17	0	1	0				
polidepor...	POLIDEPORTIVO	sports center	SPORTS CENTER	0	18	0	1	0				
portal	PORTAL	portal	PORTAL	0	19	0	1	0				
pueblo	PUEBLO	town	TOWN	0	20	0	1	0				
salón	SALÓN	living room	LIVING ROOM	0	21	0	1	0				
tienda	TIENDA	shop	SHOP	0	22	0	1	0				
zoo	ZOO	zoo	ZOO	0	23	0	1	0				

Figura 34. Tabla entidad sublugares.

La agrupación de los pictogramas en entidades hace que ya estén preparados y ordenados para su gestión desde la aplicación. No obstante, también tienen que estar relacionados con los sublugares en los que tienen que aparecer.

Para ello, se crean tablas de relaciones entre cada uno de los 1500 pictogramas y sus sublugares.

Estas relaciones se implementan mediante la utilización de la primary-key que representa a cada pictograma y la primary-key de cada sublugar. De este modo, se crean y procesan hasta 24 relaciones (correspondientes a las sublugares) por cada pictograma (de entre un total de 1500).

TABLE Z_2LUGARSUBLUGAR		Buscar	Mostrar Todo	Añadir
Z_2SUBLUGARLUGAR	Z_7LUGARSUBLUGAR			
1	2			
1	6			
1	7			
1	11			
1	19			
1	21			
2	3			
2	4			
2	18			
2	8			
2	12			
2	22			
3	1			
3	14			
3	13			
3	16			
3	15			
3	9			
4	5			
4	10			
4	17			
4	20			
4	23			
4	16			

LUGARES

1. casa
2. ciudad
3. colegio
4. vacaciones

SUBLUGARES

1. Anfas
2. baño
3. bar
4. calle
5. campo
6. cocina
7. comedor
8. espectáculo
9. gimnasio
10. granja
11. habitación
12. hospital
- ...

Figura 35. Fragmento tabla relación lugares con sublugares.

TABLE Z_1FRASESUBLUGAR		Buscar	Mostrar Todo	Añadir
Z_1SUBLUGARCOMPLEMENTOS	Z_7FRASESUBLUGAR			
4	4			
4	5			
4	6			
4	7			
4	8			
4	9			
4	10			
4	11			
4	12			
4	13			
4	14			
4	15			
4	16			
4	17			
4	18			
4	19			
4	20			
4	21			
4	22			
4	23			
5	6			
5	7			
5	22			
6	1			
6	4			
6	8			
6	11			
6	13			
6	14			
6	22			

COMPLEMENTOS

1. a través
2. abalorios
3. abrigo
4. abril
5. aceite
6. acordeón
- ...

SUBLUGARES

1. Anfas
2. baño
3. bar
4. calle
5. campo
6. cocina
7. comedor
8. espectáculo
9. gimnasio
10. granja
11. habitación
12. hospital
- ...

Figura 36. Fragmento tabla relación complementos con sublugares.

Computacionalmente, el modelo entidad-relación facilita mucho la labor de programación ya que, una vez seleccionado el sublugar, las sentencias de gestión de la base de datos sólo tienen que determinar en qué entidad se va a realizar la búsqueda, ya que el predicado de búsqueda (tipo de filtrado que es requerido) viene dado una vez elegido el sublugar.

Por ejemplo, si se selecciona el sublugar "portal" filtrado desde el lugar "casa", conforme vaya avanzando la formación de la frase, las órdenes internas serán del tipo: busca en la entidad "personas" los elementos relacionados con el sublugar "portal" y muéstralos en pantalla. En las entidades "verbos" y "complementos" se utiliza el mismo predicado.

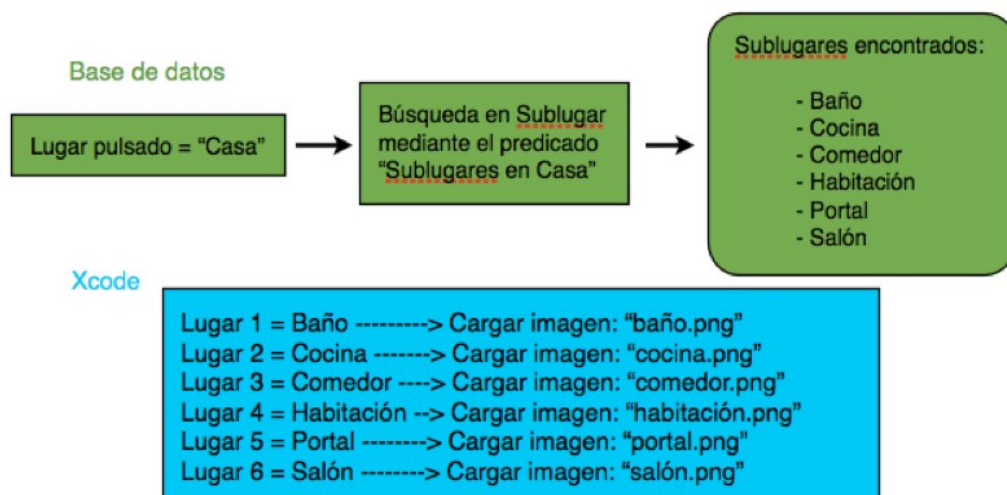


Figura 37. Explicación gráfica relación lugar-sublugar.

Gracias a la creación de este tipo de base de datos, la aplicación reduce y simplifica mucho su código fuente, ya que aporta organización e implementación de las relaciones que hacen posible su funcionamiento.

Por otro lado, como ya se ha explicado anteriormente, la aplicación para el interfaz táctil iPad no sólo consta de la herramienta comunicador. En el caso de la herramienta QuieroEstoy, su funcionamiento básico es idéntico al del comunicador, salvo por los pictogramas utilizados y el filtrado de los mismos.

Es decir, en este caso, no hay filtrado posicional en función del lugar y sublugar elegido. El filtrado se basa en el verbo principal escogido : "quiero" o "estoy". Además, no se utilizan la totalidad de los pictogramas existentes en la base de datos, sino una cantidad limitada de los mismos.

Si se escoge "estoy" aparecen los pictogramas relacionados con éste (bien, cansado, contento, triste, mal y enfadado). Un vez se escoge el estado en cuestión, la frase ya se ha formado.

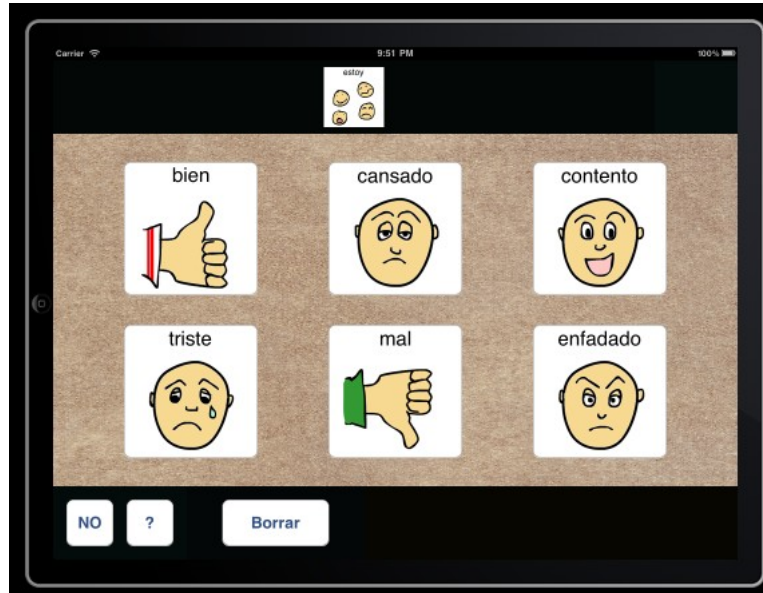


Figura 38. Estados asociados a "estoy".

Si se escoge "quiero" aparecen los verbos relacionados con éste (comer, beber, mirar, trabajar, jugar e ir). Una vez se selecciona el verbo deseado, aparecen los complementos asociados al mismo. Por ejemplo, si escoges comer aparecen albóndigas, almendras, alubias blancas, arroz, etc. Una vez se selecciona el complemento en cuestión, la frase ya se ha formado.

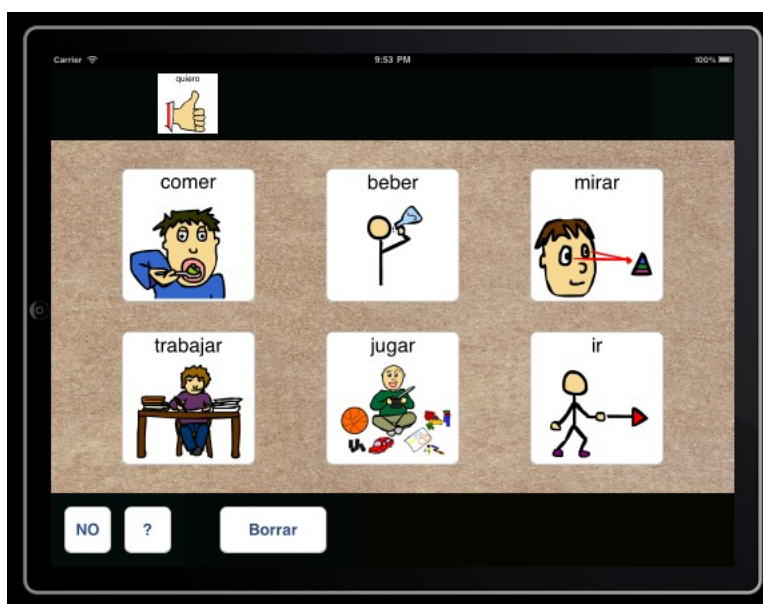


Figura 39. Verbos asociados a "quiero".

Implementar este funcionamiento desde xcode haciendo uso de la base de datos creada conllevaba insertar una cantidad de código exagerado y muy redundante. Por ello, se decidió crear dentro de la propia base general tres entidades en función de las cuales la herramienta QuieroEstoy funciona.

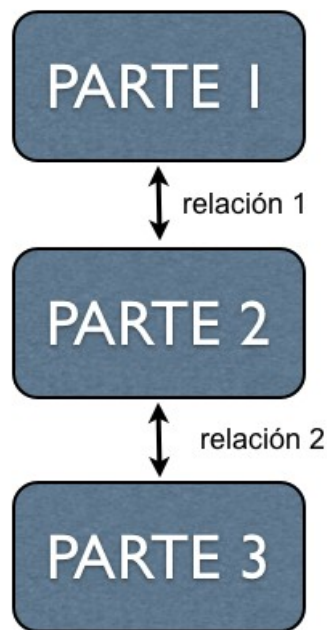


Figura 40. Tabla entidades y relaciones del QuieroEstoy.

Estas tres entidades son:

- 1- Entidad Parte 1 : pictogramas "quiero" y "estoy".
- 2- Entidad Parte 2: verbos asociados a "quiero" (comer, beber, mirar, trabajar, jugar e ir) y los pictogramas asociados a "estoy" (bien, cansado, contento, triste, mal y enfadado).
- 3- Entidad Parte 3: complementos asociados a los verbos de "quiero"

Z_PK	ZCONTADOR	ZNOMBRE_MAY	ZNOMBRE_MAY_ING	ZNOMBRE	ZNOMBRE_ING
1	0	COMER	EAT	comer	eat
2	0	BEBER	DRINK	beber	drink
3	0	MIRAR	LOOK	mirar	look
4	0	TRABAJAR	WORK	trabajar	work
5	0	JUGAR	PLAY	jugar	play
6	0	IR	GO	ir	go
7	0	BIEN	WELL	bien	well
8	0	CANSADO	TIRED	cansado	tired
9	0	CONTENTO	HAPPY	contento	happy
10	0	TRISTE	SAD	triste	sad
11	0	MAL	BAD	mal	bad
12	0	ENFADADO	ANGRY	enfadado	angry

Figura 41. Tabla entidad Parte 2.

Además, no sólo se han creado estas tres entidades, sino las relaciones necesarias entre sus elementos para filtrar el orden de aparición de éstos. Es decir, para que si por ejemplo se pulsa "quiero" aparezcan sólo los verbos asociados a "quiero" y no los estados asociados a "estoy".

Z_3PARTE2CON1	Z_4PARTE1CON2
1	1
1	2
1	3
1	4
1	5
1	6
2	7
2	8
2	9
2	10
2	11
2	12

PARTE 1
1. quiero
2. estoy

PARTE 2
1. comer
2. beber
3. mirar
4. trabajar
5. jugar
6. ir
7. bien
8. cansado
9. contento
10. mal
11. triste
12. enfadado

Figura 42. Tabla relación Parte1 con Parte2.

Como consecuencia, dada la particularidad de la herramienta QuieroEstoy, ésta basa su funcionamiento en la base de datos general del programa, pero haciendo uso de sus propias entidades y relaciones.

Finalmente, la base de datos creada consta de un total de ocho entidades, en las cuales se encuentran los 1500 pictogramas y sus atributos correspondientes, y de diferentes relaciones entre ellas que hacen posible el funcionamiento de la aplicación de la forma deseada. Se trata de una base de datos Sqlite de tipo dinámica y basada en el modelo entidad-relación.



Figura 43. Tablas entidades y relaciones de la base de datos.

Capítulo 5: Estudio y desarrollo CoreData

Una vez la base de datos está creada y el código de xcode preparado para la utilización de la misma, hay que implementar la gestión de la base desde el entorno Xcode. Es decir, hay que buscar el nexo de unión entre ambas tecnologías: CoreData.

5.1 Estudio CoreData

Core Data es un framework facilitado por Apple que cuenta con una serie de herramientas que facilitan tanto la creación del modelo de base de datos como su gestión posterior desde código.

El sistema operativo de los ordenadores de Apple (Mac OS) utiliza un sistema, denominado CoreData, que gestiona y estructura las bases de datos utilizadas en sus aplicaciones. El problema es que no estaba disponible para Xcode. Justo cuando se estaban realizando las primeras pruebas con SQLite en Xcode, Apple liberó esta herramienta para los dispositivos móviles.

Coredata se rige por el sistema denominado entidad-relación. Este nombre se refiere a un tipo de modelo de datos conceptual de alto nivel que se emplea en el diseño de las base de datos relacionales. El modelo entidad-relación muestra la estructura de la base de datos empleando todo tipo de herramientas conceptuales. Está formado por dos tipos de elementos: entidades y relaciones.

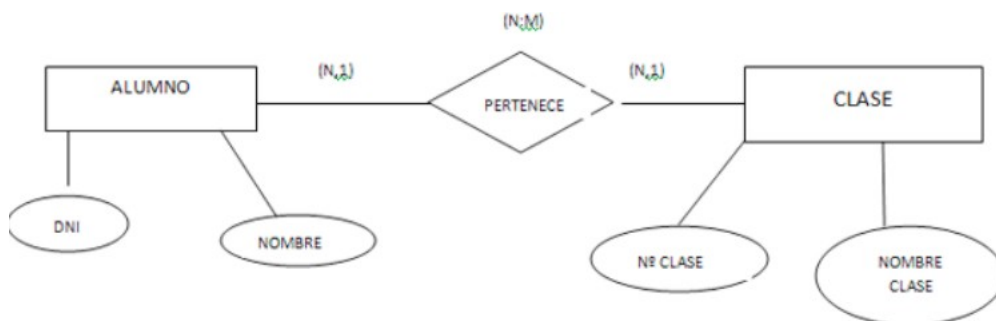


Figura 44. Esquema de una base de datos según el modelo entidad-relación.

Una entidad es la representación de un objeto o concepto del mundo real que se describe en una base de datos. Se describe en la estructura empleando un modelo de datos. Cada entidad está constituida por uno o más atributos.

Una relación o vínculo entre dos o más entidades describe alguna interacción entre las mismas. Por ejemplo, una relación entre una entidad "Alumno" y una entidad "Clase" podría ser "pertenece", porque el alumno pertenece a una clase determinada. Las relaciones se describen en la estructura de la base de datos empleando modelos de datos.

Para la gestión de una base de datos Sqlite desde Xcode hay que tener en cuenta algunos componentes básicos de CoreData:

1. Managed Objects : puede ser una instancia de `NSManagedObject` o una subclase de `NSManagedObject`. Básicamente representa un registro en una base de datos. Si tenemos una tabla de ventas, cada venta sería representada con una instancia de un Managed Object. Se pueden usar estructuras fuertemente tipeadas creando subclases de `NSManagedObject` o acceder a los atributos a través de KVO (Key-Value Observing) instanciando `NSManagedObject` .

2. Managed Object Context: se define como una instancia de `NSManagedObjectContext`. Administra colecciones de Managed Objects. Todas las operaciones de alta, baja y modificación se realizan a través del Managed Object Context. Implementa otras funcionalidades como validaciones, undo/redo, etc.

3. Managed Object Model: es una instancia de `NSManagedObjectModel`. Represente al modelo de datos, básicamente es una colección de entidades que definen los Managed Objects, cada entidad está compuesta principalmente por atributos. La descripción de la entidad esta definida en la clase `NSEntityDescription`. Cada Managed Object tiene una referencia a una instancia de `NSEntityDescription`. Es posible definir relaciones entre las entidades (uno a uno, uno a muchos, muchos a muchos) y consultas (Fetched properties).

4. Persistent Store Coordinator: por lo general no se interactua en forma directa con el Persistent Store Coordinator. Es una instancia de `NSPersistentStoreCoordinator`. Básicamente maneja una colección de Persistent Object Stores.

5. Persistent Object Store: representa un repositorio externo (por lo general SQLite). Hay distintas clases de Persistent Object Store que representan las distintas formas de almacenar la información.

6. Consultas:

Fetch Request: es una instancia de `NSFetchRequest`, básicamente representa un búsqueda de Managed Objects sobre un Managed Object Context. Se pueden cargar todos los elementos simplemente definiendo el nombre de la entidad, también se pueden especificar filtros (`NSPredicate`), el orden de los datos (`NSSortDescriptor`), etc. El resultado de la ejecución de un Fetch Request es un array de Managed Objects.

Fetched Results Controller: si se está trabajando con un `UITableView` con intención de popular la tabla con los valores obtenidos de un Fetch Request, se debería usar la clase `Fetched Results Controller`

(NSFetchedResultsController). Ésta optimiza el uso de memoria. Con este controller permite además definir el criterio de agrupación (útil para mostrar datos agrupados en UITableView), paginación, etc.

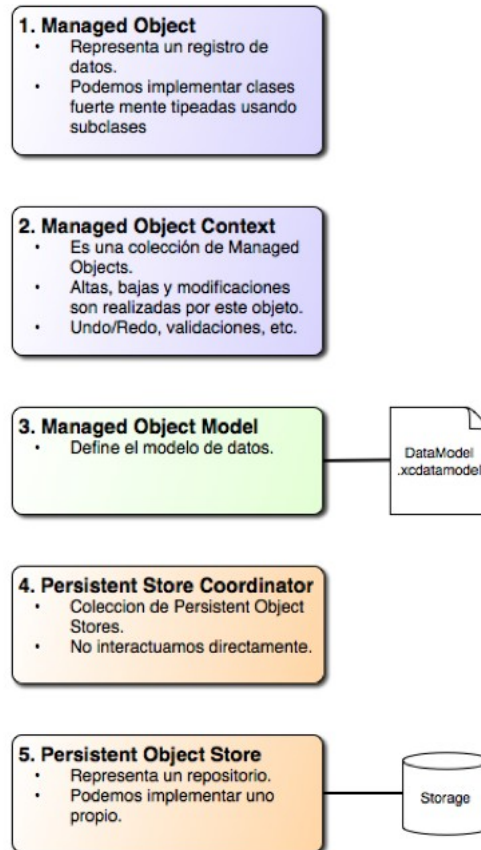


Figura 45. CoreData Stack.

5.2 Desarrollo CoreData

En el caso concreto de este proyecto, la base de datos SQLite ya está creada con una estructura definida y el programa implementado en Xcode. Lo necesario es que Xcode reconozca la base SQLite y así poder trabajar con ella y no estáticamente.

Es decir, hay que definir en CoreData la estructura que posee la base de datos para que Xcode la reconozca.

Para gestionar desde CoreData la base de datos ya creada, se definen tanto las entidades con sus atributos, como las relaciones existentes entre ellas.

Las entidades son: "Lugar", "Sublugar", "Personas", "Complementos", "Verbos", "Parte1", "Parte2" y "Parte3".

Las relaciones son: "LugarSublugar", "PersonasSublugar", "VerbosSublugar", "ComplementosSublugar", "Parte1con2" y "Parte2con3".

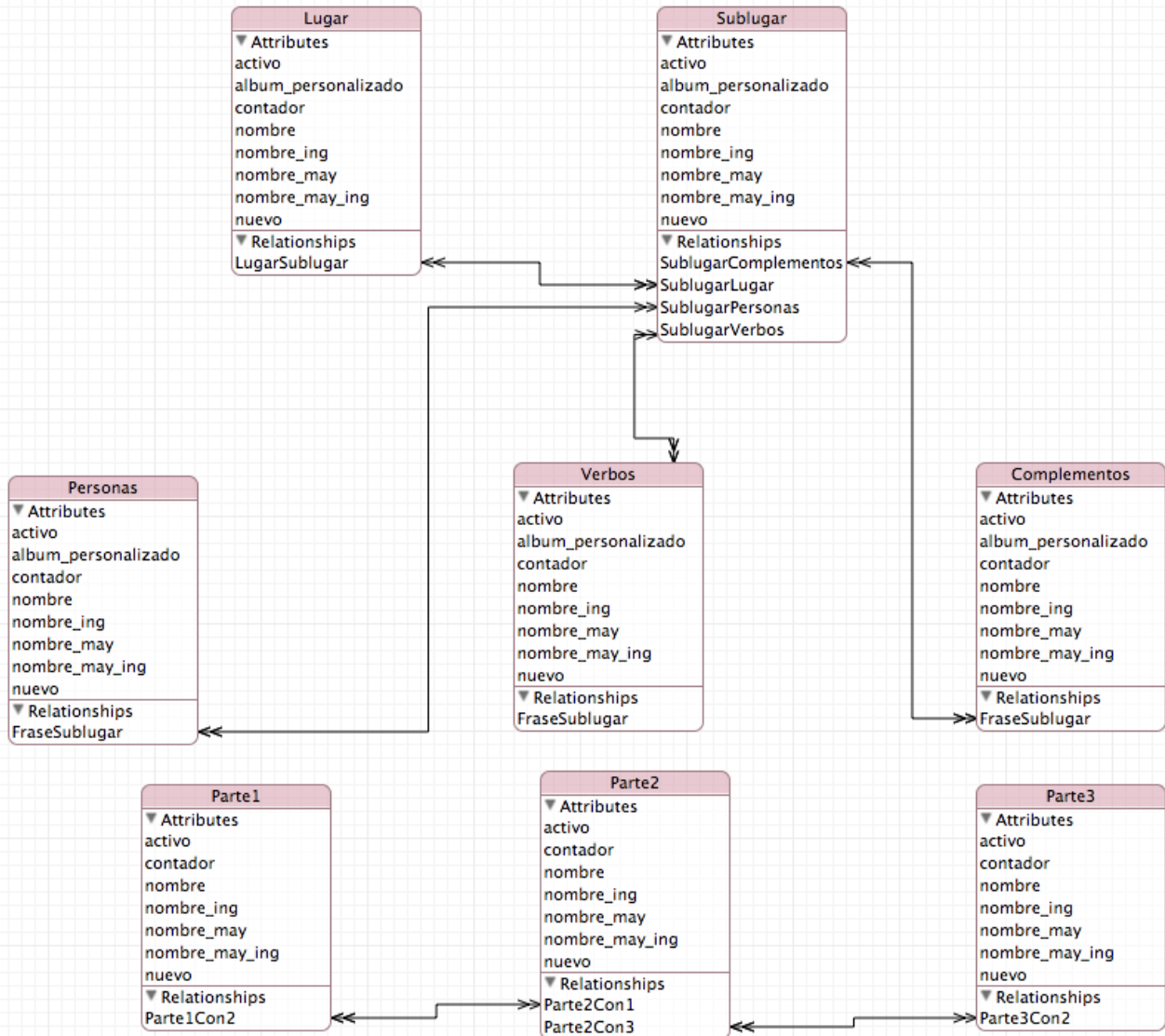


Figura 46. Definición base de datos desde CoreData.

Cada recuadro representa una entidad de la base de datos, junto con sus atributos, y las flechas las relaciones existentes entre ellas.

Una vez definido el modelo, el siguiente paso es adjuntar al programa la base de datos Sqlite ya creada.

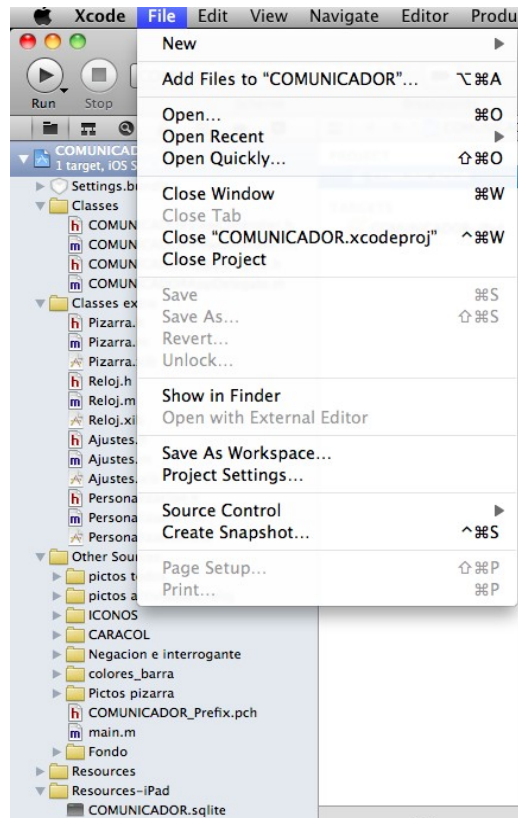


Figura 47. Adjuntar base Comunicador.sqlite a Xcode.

De esta forma, el programa ya reconoce la base de datos implementada. Para que base su funcionamiento en ella, se redefine todo el código de la aplicación y se adapta a la estructura de la base.

La nueva programación consta principalmente de llamadas a los pictogramas de la base y sus atributos correspondientes.

```
//Búsqueda de los sublugares en función del lugar pulsado.
(void) BusquedaLugares:(NSString *)entidad_búsqueda
    COMUNICADORAppDelegate *appDelegate = [[UIApplication sharedApplication]
    delegate];

    NSManagedObjectContext *context = [appDelegate managedObjectContext];

    NSEntityDescription *entityDesc = [NSEntityDescription
    entityForName:entidad_búsqueda inManagedObjectContext:context];

    NSFetchRequest *request = [[[NSFetchRequest alloc] init]autorelease];

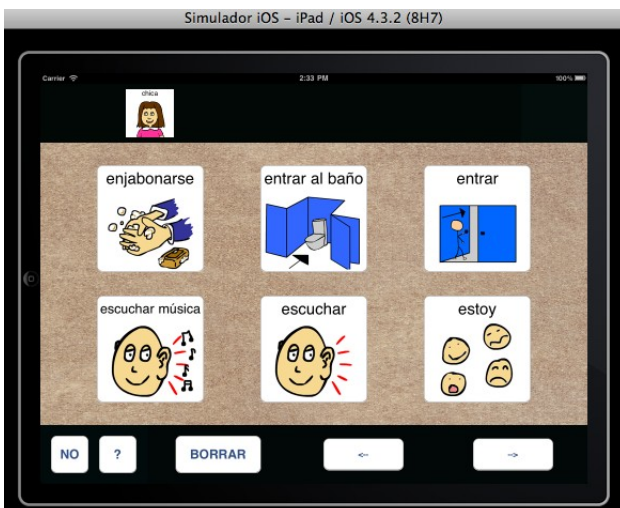
    [request setEntity:entityDesc];
    NSError *error;
    NSArray *objects = [context executeFetchRequest:request error:&error];
    NSManagedObject *matches = nil;
    [self PonerLugares:objects Objeto:matches Numero:0];
```

Figura 48. Ejemplo búsqueda y acceso elementos base de datos.

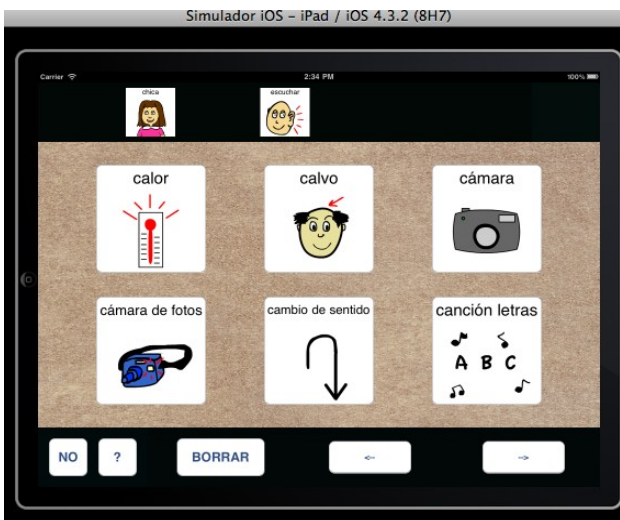
Como en la base de datos ya están todos los pictogramas clasificados en entidades y relacionados con sus respectivos sublugares, con simples llamadas éstos aparecen ya organizados para la formación de la frase:



PERSONAS



VERBOS



COMPLEMENTOS

Figura 49. Ejemplo orden aparición de pictogramas.

Capítulo 6: Estudio y desarrollo menú de configuración de la aplicación

Una de las finalidades de este proyecto es diseñar y desarrollar el mecanismo de configuración de la aplicación para el iPad.

Este mecanismo consiste en realizar una herramienta adicional, denominada Ajustes, a través de la cual se seleccionan y determinan diferentes funcionalidades del conjunto de todas las herramientas de las que consta el programa completo.

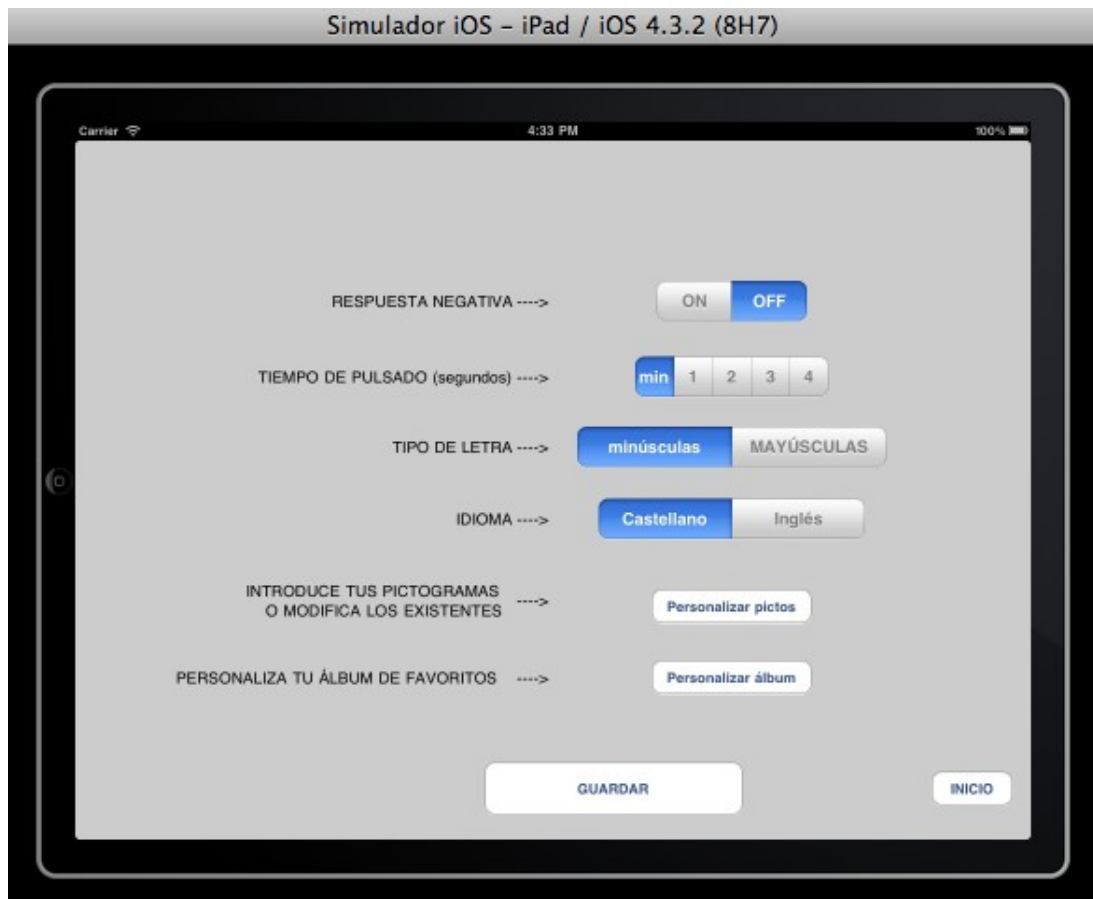


Figura 50. Pantalla de ajustes.

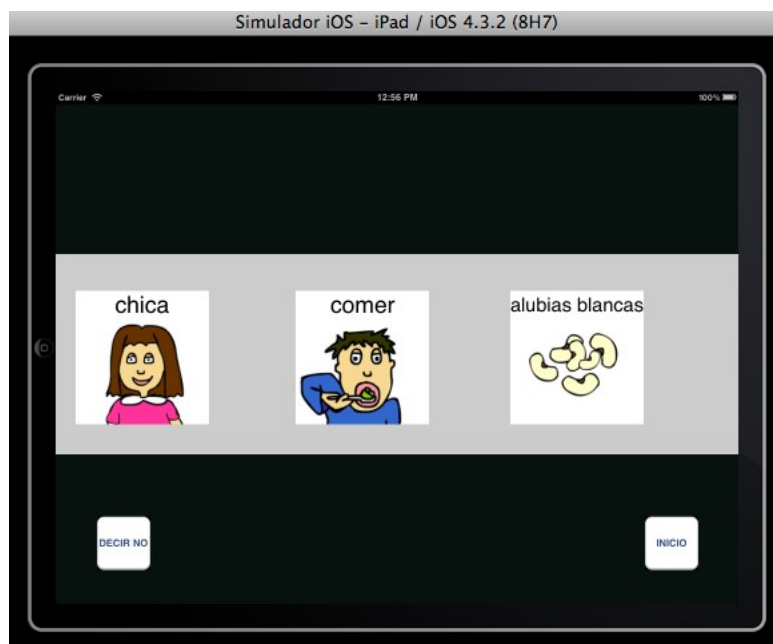
El acceso desde la aplicación al menú de ajustes se realiza pulsando en la esquina superior de la pantalla, en la cual se encuentra un botón oculto. Además, toda configuración seleccionada desde este menú influye al programa completo, a todas sus herramientas por igual.



Figura 51. Acceso al menú de configuración.

Las funcionalidades con posibilidad de ajuste son:

Respuesta negativa: botón que ofrece al educador o familiar, que se encuentre con el usuario del comunicador, la posibilidad de negarle lo que diga. Si se pulsa el botón, que por defecto está en OFF, aparece en la pantalla final, debajo de la frase definitiva formada, un botón que otorga la posibilidad de "DECIR NO".





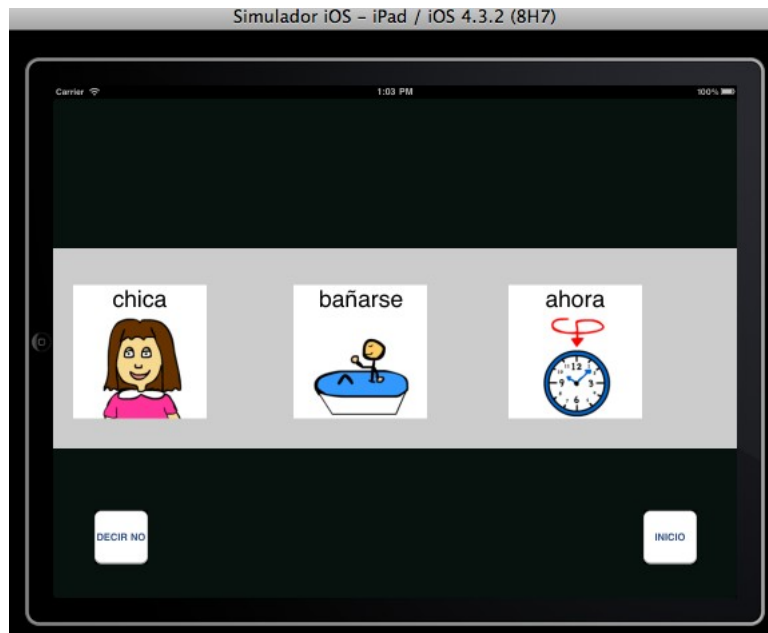
Figuras 52 y 53. Ajuste Respuesta negativa.

Tiempo de pulsado: se puede seleccionar el tiempo que ha de estar pulsado un botón para que éste ejecute la acción para la que está programado. Debido a que el usuario potencial de este comunicador presenta algún tipo de discapacidad de comunicación y es probable que tenga asociada alguna discapacidad en la motricidad, se establece esta opción para facilitarle su uso.



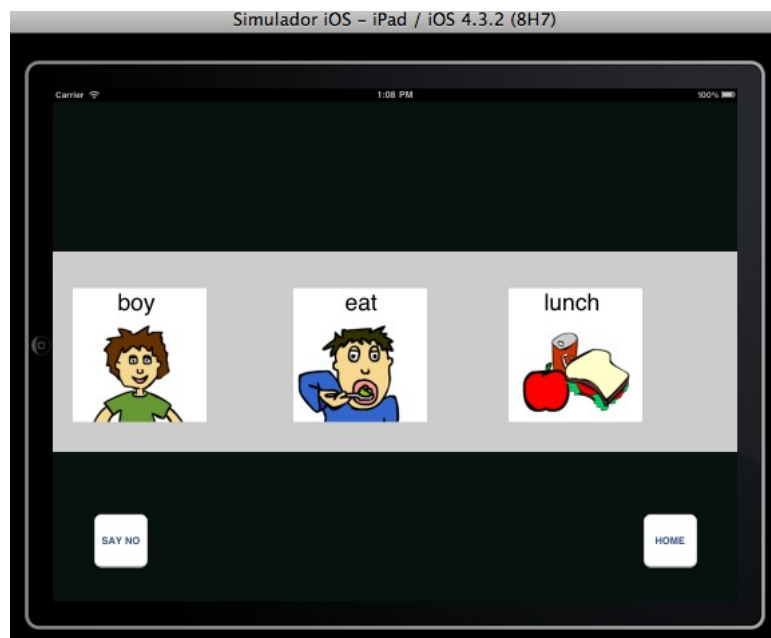
Figura 54. Ajuste Tiempo de pulsado.

Tipo de letra: dictamina si los nombres de los pictogramas aparecen en minúsculas o en mayúsculas. El motivo de este botón es la existencia de usuarios que únicamente son capaces de distinguir las letras si están de una forma o de la otra.



Figuras 55 y 56. Ajuste Tipo de letra.

Idioma: con vistas a internacionalizar el comunicador y facilitar el aprendizaje de otros idiomas, se puede seleccionar que el texto vaya en castellano o en inglés. Además, no sólo el texto es traducido al inglés o castellano, sino todos los botones, avisos, frases, etc. que aparecen en la aplicación completa.



Figuras 57 y 58. Ajuste Idioma.

Personalización: ofrece la posibilidad de actualizar pictogramas ya existentes por otros pictogramas o imágenes, a los que se esté más acostumbrado a su uso, e incluso la posibilidad de insertar nuevas palabras y elegir las imágenes asociadas a las mismas. De esta manera, las posibilidades de uso del comunicador aumentan considerablemente, gracias a su escalabilidad, presentando únicamente como limitación el espacio de disco asignado al dispositivo físico.

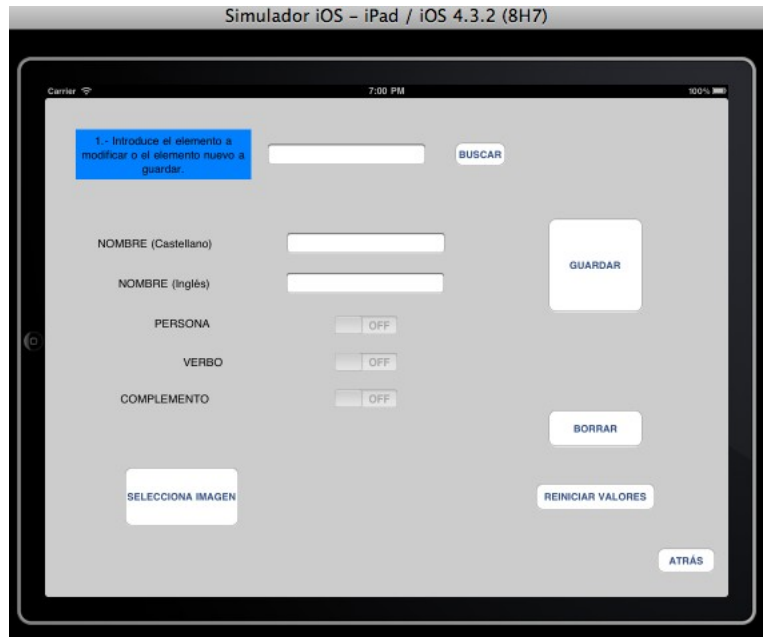


Figura 59. Ajuste Personalización.

En primer lugar, se introduce la palabra del pictograma que se va a modificar o a introducir y el programa señala si existe o no. Por ejemplo, se introduce la palabra "vocales" y el programa señala que ese valor no existe.



Figura 60. Ajuste Personalización búsqueda palabra.

Posteriormente, se rellenan los campos nombre en castellano, nombre en inglés y se selecciona si se trata de una persona, verbo o complemento.



Figura 61. Ajuste Personalización rellenar campos.

También se tiene que seleccionar la imagen que representa a esa palabra. Esto se realiza pulsando el botón "Selecciona Imagen", que redirecciona a la galería de imágenes del propio iPad.



Figura 62. Ajuste Personalización seleccionar imagen.

Una vez guardado el nuevo pictograma, éste pasa a formar parte de la base de datos de la aplicación y ya puede ser utilizado.



Figura 63. Ajuste Personalización utilización pictograma nuevo.

También cabe la posibilidad de modificar pictogramas existentes, borrarlos o incluso volver a los valores iniciales de la aplicación.

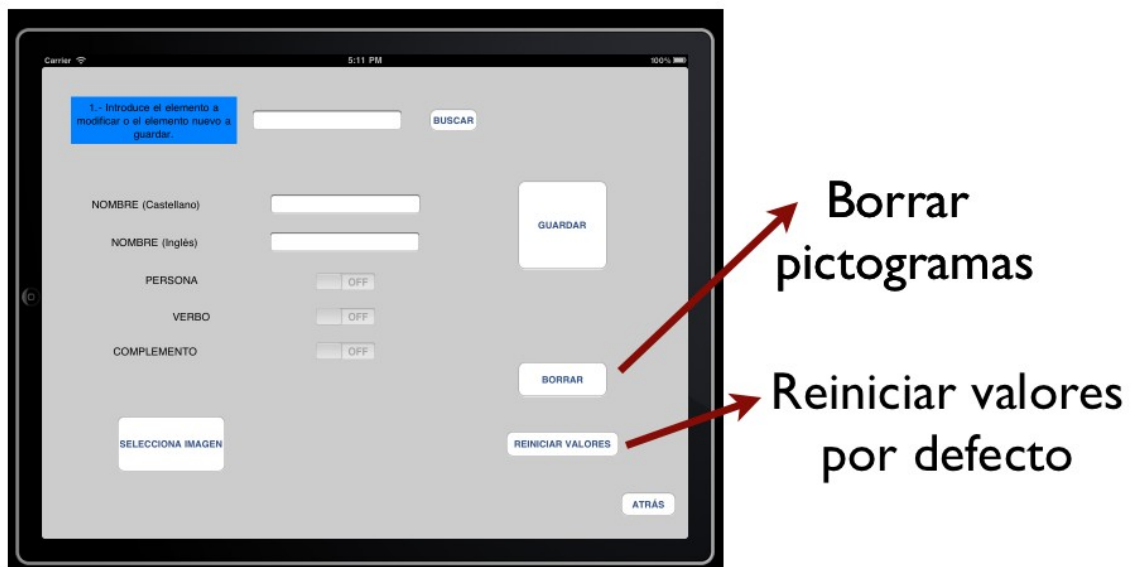


Figura 64. Ajuste Personalización borrar o restaurar pictogramas.

Favoritos: ajuste gracias al cual se pueden seleccionar, de entre todos los pictogramas que forman la base de datos de la aplicación, cuáles se prefieren y pasan a formar parte de la carpeta preferidos en la herramienta Álbum. De este modo, se puede comenzar a trabajar en el reconocimiento de pictogramas con una pequeña cantidad de los mismos, e ir aumentando poco a poco, facilitando así su aprendizaje. Esta herramienta ha sido diseñada y desarrollada por el proyecto que se ha realizado simultáneamente a éste.

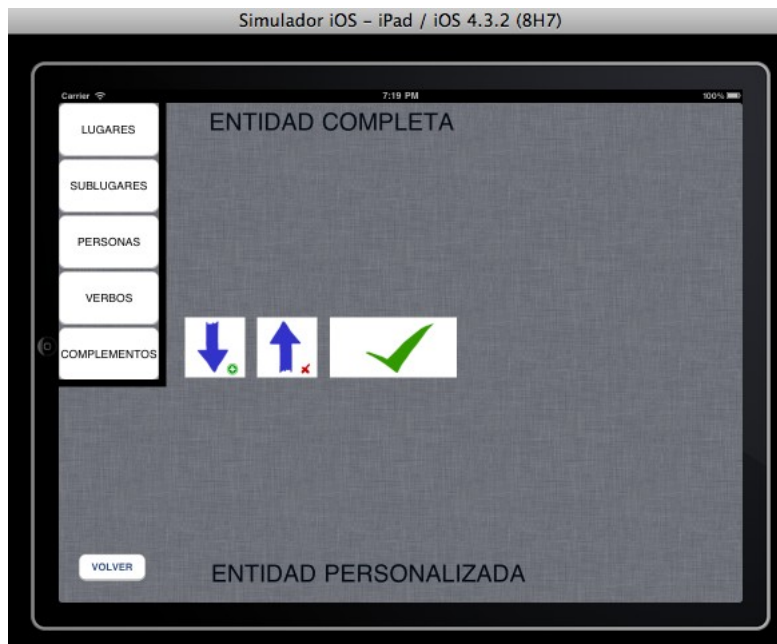


Figura 65. Ajuste Favoritos.

Por último, una nueva prestación es añadida al funcionamiento de la aplicación, pero que no queda patente en el menú Ajustes, sino que ha sido implementada por defecto. Se trata del filtrado de los pictogramas según su frecuencia de uso. Gracias a esto, los pictogramas más utilizados aparecen siempre en primer lugar, mejorando, personalizando y simplificando el uso del comunicador.



Figura 66. Pantalla del lugar cocina con los pictogramas ordenados por frecuencia de uso.

Capítulo 7: Conclusiones

Una vez terminado el proyecto, la primera conclusión que se obtiene es que, en una época en la que las nuevas tecnologías están avanzando a velocidades increíbles, una pequeña parte de la sociedad no se está beneficiando casi nada de ellas.

Después de haber asistido a reuniones, conferencias, congresos, etc. se ha constatado que hay muchas necesidades, que son solucionables con la tecnología actual y que, con esfuerzo y dinero, se pueden realizar a corto plazo.

En cuanto a la discapacidad que se afronta en este proyecto, se ha comprobado que existen, casi un millón de personas en España que son usuarios potenciales de esta aplicación y por lo tanto su calidad de vida puede aumentar con este trabajo.

Paralelamente, se ha constatado la validez de esta aplicación mediante sesiones de prueba, en este caso con niños. En pocas horas de trabajo, la familiarización con la aplicación fue plena y se consiguieron expresar a través de ella con facilidad.

Se ha conseguido realizar un equipo multidisciplinar muy completo, donde todo el mundo ha aportado sus conocimientos, enriqueciendo así al resto del grupo. Tanto el personal de Isterría, como los trabajadores de Anfas, como los familiares de estudiantes de estos centros, han sido de gran ayuda, han participado plenamente y han terminado muy satisfechos con este proyecto.

A su vez, el grupo de trabajo creado en la universidad, formado por tres proyectandos (dos estudiantes de Ingeniería de Telecomunicación y uno de Ingeniería técnica de telecomunicación, especialidad en imagen y sonido) y un tutor, ha obtenido un gran rendimiento.

Al igual que con el grupo multidisciplinar, ha habido enriquecimiento mutuo y se ha aprendido a trabajar en equipo, con todo lo que ello conlleva. Aún siendo estudiantes de Ingeniería, cada uno ha aportado conocimientos variados y diferentes dentro del campo de las telecomunicaciones.

En resumen, la valoración del proyecto es enormemente positiva. Se ha profundizado en una materia ajena al mundo de la ingeniería, como es el entorno de la discapacidad. Se ha aprendido mucho acerca de trabajar en equipo gracias a la colaboración con otras disciplinas y, ya solo la satisfacción que proporciona mejorar la calidad de vida de una persona, es suficiente como para justificar la realización del mismo.

Capítulo 8 :Líneas futuras

Gracias al desarrollo de este proyecto, se ha evolucionado y mejorado notablemente la aplicación para el dispositivo iPad. No obstante, y formando parte del valor añadido de la misma, las posibilidades de evolución y avances son enormes. Entre estas posibilidades, se pueden señalar como líneas futuras:

8.1. Agenda

En la actualidad, los familiares o cuidadores establecen el horario que va a seguir el usuario del Comunicador, en una pizarra mediante recortes, velcro y dibujos. Se puede observar un ejemplo en la imagen inferior.

	LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES
			Pepe		
		1 4 2 5 7 6 3 8		Pepe	
	Pepe		Religión Católica		

Figura 67. Imagen tomada de <http://mamideglorichi.blogspot.com>.

Este proceso es costoso, poco útil y muy limitado en imágenes y recursos. Si se consigue informatizar esta rutina y mostrarla en la pantalla del iPad, se mejorará mucho la calidad de vida de los usuarios.

Por ejemplo, a primera hora de la mañana, se podrá establecer, conjuntamente entre el familiar y el usuario, la agenda que se va a realizar a lo largo del día. Al ser un trabajo compartido, el usuario ya tiene consciencia de qué es lo que va a hacer, cuándo lo va a realizar y, si tiene dudas, lo puede consultar en cualquier momento.



Figura 68. Imagen tomada de <http://lrosilloc.blogspot.com>.

8.2. Diario

Así como la agenda será programada por los familiares o cuidadores, el diario será un elemento de uso único por parte del propietario del iPad. Deberá tener un manejo sencillo que dé la oportunidad de crear eventos, guardarlos y posteriormente poder enseñarlos.

Gracias a su uso, el usuario podrá contar qué ha hecho en el colegio en el mismo día o relatar cómo ha sido la excursión de fin de semana a sus compañeros de clase.

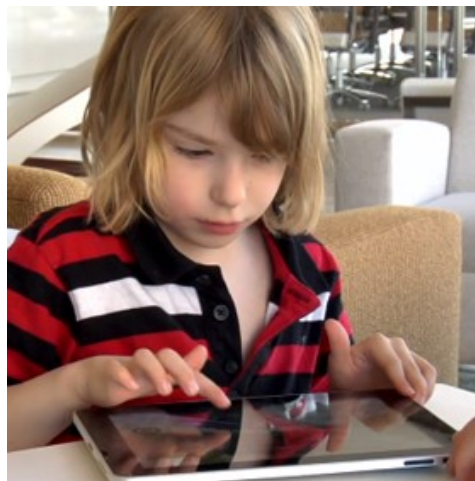


Figura 69. Imagen tomada de www.worlipad.es.

8.3. Otras mejoras

Además de las futuras herramientas extra que puede poseer el comunicador, otro aspecto bastante interesante es la adaptación del mismo a otros sistemas operativos. Desde que apareció el iPad en el mercado, han sido muchas las tabletas que han salido con diferentes sistemas operativos y equiparándose al inicial en cuanto a calidad-precio se refiere.

El análisis realizado durante la primera fase del proyecto se ha podido quedar desactualizado con las nuevas apariciones, así que otra línea a tratar en el futuro será la de realizar un nuevo estudio de mercado, analizar los nuevos productos y, si es necesario, adaptar el comunicador para ellos.

Del mismo que se ofrece la posibilidad de utilización de la aplicación en castellano o inglés, exportarla a otros idiomas ampliará la cantidad de usuarios a los que va dirigida, aumentando así su mercado y clientes.

Dada la necesidad existente de un dispositivo móvil adaptado a las necesidades reales de las personas con discapacidad en la comunicación, se va a llevar a cabo la comercialización de la aplicación e inclusión de las mejoras anteriormente explicadas. Para ello, se constituirá una empresa dedicada al diseño y desarrollo de aplicaciones móviles para la discapacidad, cuyo plan de viabilidad se encuentra detallado en el apartado *Anexos*.

De este modo, se asegura una continuidad a la aplicación (mejoras, actualizaciones, determinación de puntos débiles y su posterior corrección,...) y una oportunidad real de acceso a la misma por parte de las personas con discapacidad.

Capítulo 9 :Presupuesto

Descripción detallada:

Ordenador MacBook Air 13''

Configuración:

Core i7 de Intel de doble núcleo a 1,8 GHz
 4 GB de memoria DDR3 a 1.333 MHz
 Procesador HD Graphics 3000 de Intel con 384MB de SDRAM DDR3
 compartida con la memoria principal
 256 GB de almacenamiento flash
 SuperDrive
 Teclado retroiluminado (Español) & Guía del usuario (En Español)
 Adaptador Ethernet
 AppleCare Protection Plan
 iWork preinstalado (Versión española)
 Mini DisplayPort to VGA Adapter

Precio:.....2.040,01 €

Licencia Apple Developer.....99\$
 (Tasa=0,691).....68,37 €

iPad con Wi-Fi, 16GB negro.....479 €

Salario ingeniero de telecomunicación.....1500 € x 14 pagas = 21.000 €

Total: 23.587,38 € (IVA incl)

Bibliografía

Desarrollo de software:

<http://developer.apple.com>

<http://www.apple.com>

<http://developer.apple.com/library/mac/#documentation/Cocoa/Conceptual/CoreData/cdProgrammingGuide.html>

<http://www.iphonesoftware.com>

<http://www.iphonedevsdk.com>

<http://stackoverflow.com>

<http://www.cocoalab.com/?q=BecomeAnXcoder-Español>

<http://www.iphonesdkarticles.com>

<http://www.iphoneipad.es/index.phpacc=contenido&idcont=57&busca=&tag=&filtro=&tipo=&pagina=>

<http://www.cristalab.com/tutoriales/fundamentos-de-programacion-para-iphone-c260/>

<http://www.cristalab.com/tutoriales/tutorial-de-programacion-para-iphone-c259/>

Instituciones:

<http://www.anfasnavarra.org>

<http://comunidad.cajanavarra.es/isterria-centro-educacion-para-ninos-con-discapacidad/>

Antecedentes:

<http://www.handholdadaptive.com>

<http://voice4uaac.com>

<http://www.proloquo2go.com>

<http://www.converseapp.com>

<http://www.ablah.org>

<http://www.iautism.info>

<http://www.projectomega.org>

Bases de datos:

<http://www.sqlite.org/>

<http://www.sqlitemanager.org/>

<https://addons.mozilla.org/es-es/firefox/addon/sqlite-manager/>

<http://ablogontech.wordpress.com/2009/07/13/using-a-pre-populated-sqlite-database-with-core-data-on-iphone-os-3-0/#comments>

<http://www.tinymission.com/blog/blogengine.web/?tag=/iOS>

<http://www.iphonedevsdk.com/forum/iphone-sdk-development/25641-load-data-into-core-data.html>

<http://stackoverflow.com/questions/926536/core-data-nspredicate-for-relationships>

<http://www.slideshare.net/microeditionbiz/eva-2010-introduccion-a-core-data-en-iphone>

http://developer.apple.com/library/ios/#documentation/DataManagement/Conceptual/iPhoneCoreData01/Articles/05_Fetching.html%23//apple_ref/doc/uid/TP40008305-CH107-SW1

<http://www.microedition.biz/blog/?p=549>

<http://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/CoreData/cdProgrammingGuide.html>

http://www.techotopia.com/index.php/An_iPhone_OS_Core_Data_Tutorial

<http://www.migueldiazrubio.com/2010/10/27/desarrollo-ios-primeros-pasos-con-core-data/>

<http://www.zenbrains.com/blog/2010/04/core-data-en-iphone/>

<http://mobile.tutsplus.com/tutorials/iphone/iphone-core-data/>

<http://cocoadevcentral.com/articles/000085.php>

<http://www.raywenderlich.com/934/core-data-tutorial-getting-started>

<http://www.xprogress.com/post-30-uiswitch-tutorial-example-how-to-use-the-switch-in-xcode-for-iphone-sample-included/>

<http://cocoadevcentral.com/articles/000085.php>

<http://wiresareobsolete.com/wordpress/2009/12/adding-core-data-existing-iphone-projects/>

<http://www.zend.com/php5/articles/php5-sqlite.php>

<http://necudeco.blogspot.com/2005/08/31/sqlite-con-csharp/>

Anexos

Plan de Viabilidad

Empresa para la integración de las TIC en personas con diversidad funcional

ÍNDICE

DESCRIPCIÓN DEL PROYECTO

- Idea de Negocio
- Características básicas del proyecto
- Presentación de los promotores

ANÁLISIS DEL MERCADO

- Análisis del entorno
- Proveedores críticos
- Análisis de la demanda
- Análisis de la oferta

PLAN DE MARKETING

- Marketing Mix
- Previsión de ventas

RECURSOS

- Recursos Humanos
- Recursos Materiales

PLAN ECONÓMICO FINANCIERO

- Ingresos
- Gastos
- Inversiones

DAFO

Debilidades

Amenazas

Fortalezas

Oportunidades

CONCLUSIONES

DESCRIPCIÓN DEL PROYECTO

Idea de Negocio

¿Qué es?

Nuestra empresa posibilita que personas con diversidad funcional en el habla puedan mejorar su autonomía personal, gracias al uso de un interfaz táctil que hace posible la comunicación.

El principal objetivo de la empresa es crear un interfaz táctil que favorezca la comunicación de personas discapacitadas, o con algún tipo de problema motor, con el resto de la gente. Mediante el uso de este interfaz, el usuario tendrá acceso a menús sencillos e intuitivos que le permitirán dar a conocer en cada momento sus necesidades o inquietudes.

Gracias a nuestra empresa, las TIC pasan a ser un método de integración y grandes avances para la discapacidad. Estas grandes mejoras normalmente van dirigidas hacia discapacidades de tipo auditiva o visual, pero hay un amplio número de personas con dificultades a la hora de comunicarse que no ven en las nuevas tecnologías, de momento, ninguna mejora en su calidad de vida.

¿Por qué es necesario?

Las nuevas tecnologías de la información y la comunicación lo han cambiado todo; la forma de vivir, de trabajar, de producir, de comunicarnos, de comprar, de vender, de enseñar, de aprender. Todo nuestro entorno es distinto, por lo que se han convertido en una herramienta básica en la vida diaria. Como consecuencia, pueden llegar a considerarse un elemento más de aislamiento social para personas con diversidad funcional.

Las personas con dificultades comunicativas tienen muchos problemas para hacerse entender. Los únicos métodos que existen se centran en el uso de carpetas, con una gran cantidad de imágenes correspondientes, en su mayoría, al código SPC, mediante las cuales el sujeto señala qué es lo que quiere decir. El objetivo de nuestra empresa es informatizar este proceso a la vez que aprovechar las nuevas tecnologías para crear nuevas formas de comunicación mucho más sencillas para el interlocutor.

Este tipo de herramientas tienen como propósito facilitar la comunicación con las personas del entorno y así poder integrarse mejor socialmente. Los medios de comunicación son instrumentos en constante evolución y este proyecto permite proporcionar un medio eficaz mediante el cual personas con discapacidad para la comunicación puedan convertirse en comunicadores activos, se les posibilita la capacidad del habla y aumentar la fluidez comunicativa.

El uso de este interfaz táctil otorga una calidad de vida superior que permite desarrollar la autonomía personal, mejorar la socialización y favorecer la autoestima.

Del mismo modo, nuestra empresa acerca un nuevo, útil y ventajoso mundo a la discapacidad: las nuevas tecnologías.

Población

Para atender a las necesidades del colectivo de personas con discapacidad, se hace indispensable conocer sus circunstancias: cuántos son, qué limitaciones tienen, con qué severidad, y si disponen de ayudas y cuidados. Además, es importante comprender la realidad social y la calidad de vida de estas personas, como su acceso al empleo, el apoyo familiar, la discriminación o las barreras a las que se enfrentan.

En 2008 había 3'85 millones de personas¹ residentes en hogares con discapacidad o limitación. Esto supone una tasa de 85,5 por mil habitantes. El 67,2% de estas personas presentan limitaciones para moverse o trasladar objetos, el 55,3% tienen problemas relacionados con las tareas domésticas y el 48,4% con las tareas del cuidado e higiene personal. En un total de 3'3 millones de hogares españoles reside al menos una persona con discapacidad, lo que supone un 20,0% de los hogares españoles. 608.000 personas viven solas.

Nuestra empresa va dirigida a quienes presentan una deficiencia física o mental que les impide, total o parcialmente, comunicarse con su entorno. Nos referimos, desde personas autistas hasta, en un futuro, enfermos de alzheimer.

Tipos de discapacidades

(Personas de 6 y más años con discapacidad)

	Tasas por mil hab.	
	Varones	Mujeres
Total	72,6	106,3
Movilidad	42,6	77,5
Vida doméstica	29,5	69,2
Autocuidado	31,3	55,3
Audición	21,9	28,4
Visión	17,8	28,4
Comunicación	16,3	18,6
Aprendizaje y aplicación de conocimientos y desarrollo de tareas	12,7	17,1
Interacciones y relaciones personales	14,0	15,4

Comunicación	16,3	18,6
---------------------	-------------	-------------

Figura 70. Tabla tipos y tasa discapacidades en España.

Si en España viven 45.957.671, el número de hombres con algún tipo de discapacidad que afecte a la comunicación sería más o menos de 749.099. El número de mujeres sería aproximadamente de 854.800. Estaríamos hablando de un número de consumidores potencial de 1.603.899 personas en España.

¹ Último boletín publicado por el INE, diciembre de 2008.

Dependencia

Los cambios demográficos experimentados en las últimas décadas en España han traído consigo profundas transformaciones en la pirámide poblacional, entre ellas un proceso de envejecimiento notable. La esperanza de vida ha pasado a 83 años en hombres y 87 en mujeres.

Un efecto es el aumento de las personas con discapacidad, ya que la edad es un factor determinante en la aparición de este fenómeno. El aumento de la población anciana conlleva el incremento de posibles clientes para nuestra empresa.

A continuación, se representan un gráfico del porcentaje actual de personas con discapacidad por edades y la pirámide poblacional prevista para 2050. En ésta última, se aprecia cómo en un futuro próximo se producirá un envejecimiento de la población mayor que requerirá más ayudas o atenciones.

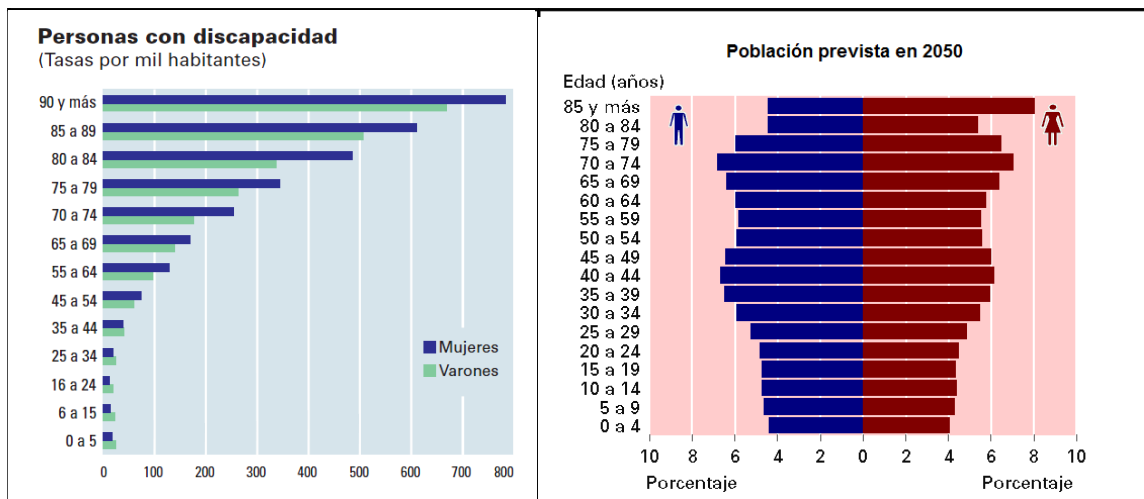


Figura 71. Gráficos personas con discapacidad y previsión 2050.

¿Por qué nuestra empresa?

En paralelo a la evolución de las nuevas tecnologías, son numerosas las empresas dedicadas a desarrollar, diseñar y vender diferentes componentes software o hardware para mejorar la accesibilidad de las personas con diversidad funcional.

Sin embargo, en muchas ocasiones resulta complejo buscar y encontrar una solución adecuada a las necesidades de cada persona. Por ello, nuestra empresa no se dedica únicamente al desarrollo software, sino a un desarrollo que satisfaga necesidades reales y esté totalmente adaptado para su uso por parte de las personas con discapacidad, es decir, un trabajo diseñado y desarrollo con el apoyo y asesoramiento de profesionales especializados en discapacidad.

Características básicas del proyecto

Requerimientos básicos

- Ubicación

En principio no sería necesario disponer de un local físico para llevar a cabo nuestras labores como empresa. No obstante, sí es recomendable de cara a la realización del trabajo, imagen de nuestra empresa, lugar de encuentro con socios y clientes...

- Recursos Humanos y materiales

Detallado en apartados posteriores (Recursos).

- Proveedores

Ya que nuestro producto es software, el único proveedor que necesitaríamos, si así se puede llamar, sería aquel que nos proporcionara los ordenadores de trabajo o los dispositivos móviles para hacer pruebas. Esto sería muy sencillo ya que pueden encontrarse en cualquier tienda especializada.

- Plan de marketing

La base primordial para que nuestra empresa funcione es dar a conocer nuestra aplicación y para ello sería necesario un plan de marketing muy elaborado, no por nosotros, sino por una persona especializada en el entorno de la discapacidad. El conocimiento de esa persona es básico para nuestro desarrollo.

- Clientes

Nuestros clientes se pueden clasificar en dos tipos: individuales o entidades. Gracias a la estructura de la tienda de Apple, nuestro producto puede llegar al usuario final directamente sin ningún tipo de intermediario pero también es interesante para nosotros la comunicación con las entidades relacionadas con la discapacidad así como: colegios especializados, fundaciones, etc... El añadir nuestro producto a su oferta educacional hará que su valor añadido aumente y por lo tanto sea beneficioso para ambos.

Etapas del proyecto

1) Desarrollo idea

El desarrollo de la aplicación, producto inicial de la empresa, está prácticamente terminado. Durante un año y apoyados por entidades especializadas en discapacidad, como Anfas e Isterria, hemos desarrollado un prototipo que se ajusta a las necesidades encontradas.

2) Formación de la empresa

En nuestro caso, antes quizás de empezar a dar a conocer la aplicación, deberíamos tener ya conformada la empresa para, en el siguiente punto, dar una imagen seria y respetable.

3) Plan de comunicación

El dar a conocer nuestro producto es la base del buen funcionamiento de éste. Una vez terminado el prototipo, es necesario elaborar un calendario de presentaciones en sociedad, charlas demostrativas, prácticas tuteladas en centros para que se dé a conocer.

4) Prototipo gratuito

Casi a la par que el plan de comunicación, es necesario elaborar una "demo" y subirla a la AppStore para que los posibles futuros clientes vayan observando el producto y a la vez, podamos ir teniendo una estimación de la aceptación que va a tener.

5) Venta

Seguido de la presentación del prototipo y, en función de la evolución del plan de comunicación, sería necesario subir a la AppStore la aplicación definitiva para que todo aquel que esté interesado pueda adquirirla.

6) Evolución como empresa

Una vez desarrollado el prototipo y puesto en marcha el proyecto, hay que hacer que la empresa evolucione y no sea empresa de un único producto. Para ello, habrá que realizar otro plan de viabilidad que analice las necesidades de la sociedad en cuanto a aplicaciones para discapacitados, tanto software como hardware y decidir un camino a seguir.

Presentación de los promotores

Ainhoa Esténoz Abendaño

- Formación: Ingeniera de Telecomunicación por la Universidad Pública de Navarra.
- Responsabilidades: Será uno de los dos socios fundadores de la empresa.

José Cruz Pérez Pi

- Formación: Ingeniero de Telecomunicación por la Universidad Pública de Navarra.
- Responsabilidades: Será uno de los dos socios fundadores de la empresa.

Una buena posibilidad sería añadir a este reparto un tercer socio especializado en el mundo de la discapacidad, ya que la base tecnológica la tenemos pero la base cognitiva nos es necesaria. No obstante, inicialmente contaremos con este perfil como un trabajador contratado por la empresa.

ANÁLISIS DEL MERCADO

Análisis del entorno

Descripción del sector

La localización física de nuestra empresa será en Navarra, Comunidad Foral en la inicialmente desarrollaremos nuestra labor y en la que no existe ninguna otra empresa dedicada exactamente a nuestro objetivo.

Aspectos normativos

En a Comunidad Autónoma Navarra no existe ninguna Norma u Ordenanza que pueda ser una limitación para el correcto desarrollo de nuestra empresa.

De hecho, dado el servicio que prestamos para las personas con discapacidad, fomentamos el cumplimiento de la normativa actual sobre discapacidad y son numerosas las ayudas a las que podemos optar. Nuestra empresa contribuye al desarrollo de medidas de acción positiva para la participación integral de las personas con discapacidad en la sociedad.

La accesibilidad al entorno por parte de las personas con diversidad funcional vendría a configurarse como un principio que afectaría a los siguientes campos:

- Empleo y ocupación
- Espacios urbanos y vivienda
- Transporte
- Bienes y Servicios a disposición del público.
- Participación ciudadana y relaciones con las Administraciones Públicas
- Sociedad de la Información: Telecomunicaciones; acceso y uso de la red; tecnologías, productos y servicios.

Desde el campo al que pertenecemos, Sociedad de la Información, posibilitamos el acceso y utilización por parte de los afectados por cualquier minusvalía motriz de los bienes y servicios para mejorar su calidad de vida, basándonos en la "accesibilidad universal" y el "diseño para todos".

- LEY FORAL 22/2003, de 25 de marzo, de modificación de la Ley Foral 4/1988, de 11 de julio, sobre barreras físicas y sensoriales.
 - La posibilidad de acceso y utilización por parte de los afectados por cualquier minusvalía orgánica o circunstancial de los bienes y servicios aparece como una condición para mejorar la calidad de vida del conjunto de los ciudadanos.

- LEY 51/2003, de 2 de diciembre, de igualdad de oportunidades, no discriminación y accesibilidad universal de las personas con discapacidad.

Accesibilidad universal: la condición que deben cumplir los entornos, procesos, bienes, productos y servicios, así como los objetos o instrumentos, herramientas y dispositivos, para ser comprensibles, utilizables y practicables por todas las personas en condiciones de seguridad y comodidad y de la forma más autónoma y natural posible. Presupone la estrategia de "diseño para todos " y se entiende sin perjuicio de los ajustes razonables que deban adoptarse.

Diseño para todos: la actividad por la que se concibe o proyecta, desde el origen, y siempre que ello sea posible, entornos, procesos, bienes, productos, servicios, objetos, instrumentos, dispositivos o herramientas, de tal forma que puedan ser utilizados por todas las personas, en la mayor extensión posible.

Aspectos sociológicos

Debido a la labor que realizamos, no esperamos ningún rechazo por parte de la sociedad; al revés, un apoyo en búsqueda de la igualdad.

Nuestra empresa tendrá un gran apoyo institucional y será bien acogida por la sociedad debido en gran medida a los recientes cambios legales en cuanto a dependencia o contratación de personas con diversidad funcional.

Proveedores críticos

- Ninguno

Análisis de la demanda

Una primera idea sobre la demanda aparece en el apartado de ¿Por qué es necesario? (Descripción del Proyecto, Idea de Negocio).

Actualmente, residen en España aproximadamente 1.600.000 personas con problemas de comunicación en cerca del 20% de los hogares.

Analizando todos estos factores, consideramos que existe una alta demanda de nuestros servicios que irá creciendo en un futuro próximo.

Análisis de la oferta

Informáticamente hay muy pocos programas, o aplicaciones, dedicados a este tipo de tareas. Una muestra de lo que existe son las aplicaciones explicadas a continuación.

VOICE4U (<http://voice4uaac.com/>)

Aplicación para iPhone que contiene una galería de imágenes de todo tipo: comida, emociones, etc.. Todas estas imágenes van con sonido. La única opción de formar frases es la de guardar una imagen y añadirle un texto y un sonido.

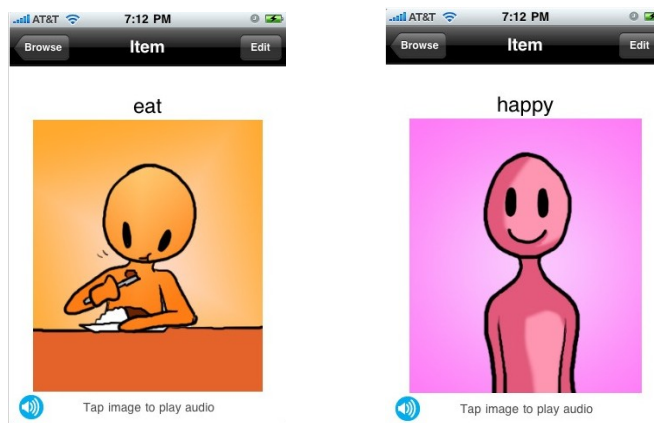


Figura 72. Capturas de la aplicación VOICE4U.

Precio: 29'99\$

IPROMPTS (<http://www.handholdadaptive.com/>)

Se trata de una aplicación para iPhone que contiene herramientas como:

- Calendarios.
- Organización de tareas.
- Cuenta atrás.
- Selección de imágenes.

Caben destacar las herramientas de cuenta atrás y la de agenda, ya que pueden servir de base para varias ideas del proyecto.



Figura 73. Capturas de la aplicación IPrompts.

Precio: 49'99 \$

La aplicación está basada en una galería de imágenes ya creada, no utiliza pictogramas de ningún tipo y, si se descarga para Iphone, da la opción de añadir fotografías obtenidas con la cámara pero no da la oportunidad de generar frases de ningún tipo.

PROLOQUO2GO (<http://www.proloquo2go.com/>)

Aplicación para iPhone, iPod Touch y iPad. Es quizás la que más se asemeja a este proyecto. Posee unos 8000 iconos y da la opción de formar frases. Los iconos aparecen en la pantalla inferior, clasificados por categorías, y la frase se forma encima.

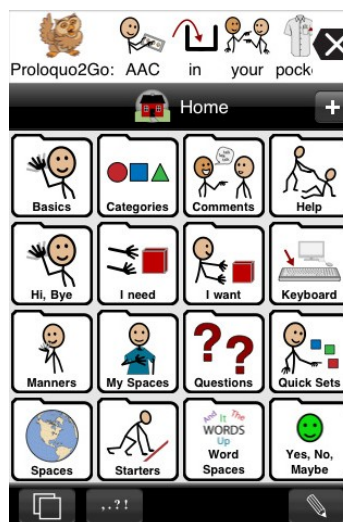


Figura 74. Captura de la aplicación PROLOQUO2GO.

Precio: 189 \$

La aplicación se comercializa únicamente en inglés y la penetración de mercado de ésta en el panorama español ha sido prácticamente nula. La distribución de lopectogramas resulta un poco caótica y el usuario tiende a perderse. En cuanto a precio, es la aplicación más cara de todas.

ICONVERSE (<http://www.converseapp.com/>)

Aplicación para iPhone que posee un álbum de imágenes a las que se les puede añadir un texto. Tiene una clasificación inicial de 6 iconos (comer, beber, enfermedad, baño, dormir y ayuda). Al ser para iPhone, se pueden añadir fotos tomadas con la cámara.

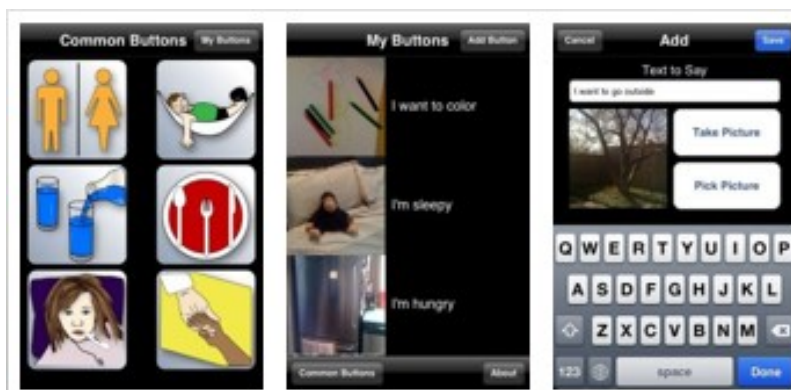


Figura 75. Capturas de la aplicación ICONVERSE.

Precio: 9'99 \$

ABLAH (<http://www.ablah.org/>)

Ablah es la última aplicación creada en esta línea de investigación y, casualmente, ha sido diseñada por un español. Fue presentada hace poco más de dos meses y ha ganado el premio a la mejor aplicación española del año.

No es tan completa como PROLOQUO2GO pero también es interesante. Es de las pocas que se puede descargar para iPad pero, al igual que muchas de ellas, es básicamente un álbum de pictogramas.

La mayor ventaja de esta aplicación es que es la única en castellano. El problema es que, al igual que la mayoría de las anteriores, no sirve para formar frases.



Figura 76. Capturas de la aplicación ABLAH.

Precio: 29.99 euros.

En el mercado existen otras aplicaciones pero las resumidas anteriormente son las más completas. A continuación se puede observar un cuadro resumen, indicando e idioma, el precio y si son, o no, compatibles con el Ipad.

Logotipo	Nombre	Disponible para Ipad	Disponible para Iphone	Idioma	Precio
	Voice4you	NO	SI	Inglés	29,99 \$
	Iprompts	SI	SI	Inglés	49,99 \$
	Proloquo2go	SI	SI	Inglés	189 \$
	Iconverse	NO	SI	Inglés	9,99 \$
	Ablah	SI	SI	Español Inglés	29,99 €

Figura 77. Cuadro resumen de los antecedentes.

En conclusión, se puede decir que existen aplicaciones que se acercan a lo que se pide por parte del entorno de las personas discapacitadas, pero son todavía limitadas, tanto en comunicación como en idioma.

El proyecto que se ha realizado, intenta resolver las carencias de esta aplicaciones y mejorar, sobre todo, la capacidad comunicativa de éstas.

PLAN DE MARKETING

Marketing Mix

Política de producto

Nuestro producto final será una aplicación software completa y adaptada al cliente y sus necesidades, que incluirá las fases de diseño, desarrollo, elaboración y actualizaciones. Además, no se trata de una única aplicación, sino que el ente central de la misma es el comunicador, pero que posee una cantidad adicional de herramientas que completan a la anterior y mejoran el aprendizaje y fomento de la comunicación en personas con discapacidad en el habla.

Política de precio

El precio de la aplicación será de 60 €, incluyendo en el mismo las futuras actualizaciones de la misma.

Política de distribución

Inicialmente, llegaremos a nuestros clientes a través de un plan publicitario para darnos a conocer en diferentes asociaciones u organizaciones que trabajen con la diversidad funcional y la accesibilidad. De este modo, daremos a conocer nuestro producto y la forma de acceder al mismo a través del AppleStore. Posteriormente, intentaremos captar clientes en agrupaciones que traten la dependencia de las personas mayores. Conforme pase el tiempo, nos seguiremos anunciando periódicamente en revistas, páginas web, foros sobre discapacidad, asociaciones...

Política de promoción

Abogamos por la accesibilidad universal y el diseño para todos, mejorando de este modo la calidad de vida de las personas con diversidad funcional. Conocemos a las personas y diseñamos para ellas. Aunamos nuestros conocimientos en las nuevas tecnologías con los relacionados con discapacidad y accesibilidad.

Previsión de ventas

Ofrecemos un producto diseñado por y para las personas con discapacidad, totalmente adaptado y que satisface un problema real existente. Nuestros clientes reciben diversas ayudas que les facilitan acceder a nuestro servicio más fácilmente. Por todo ello, estimamos conseguir, solo en España, una penetración del 5% de un total de 1.600.000 personas con discapacidad en la comunicación existentes, lo que conlleva un total de 80.000 descargas.

RECURSOS

Recursos Humanos

Medios necesarios

- Dos ingenieros de telecomunicación que se encargarán de las tareas de diseño y desarrollo de proyectos, compra y supervisión de los equipos y de la labor comercial con los clientes.
- Un terapeuta ocupacional, especializado en discapacidad, se dedicará a evaluar y estudiar la situación concreta de los clientes, sus necesidades y posibles soluciones. Además, supervisará, asesorará y trabajará en el diseño de las aplicaciones, al igual que en su usabilidad y accesibilidad. También se encargará de los contactos y relaciones con las diferentes asociaciones, colegios y entidades especializadas en discapacidad.

Cualificación necesaria

- 1 Los ingenieros deberán ser Ingenieros de Telecomunicación, con ciertos conocimientos en Ingeniería Biomédica.
- 2 El terapeuta poseerá gran experiencia en sector de la discapacidad y conocimiento de este colectivo y asociaciones relacionadas con el mismo. Habrá realizado numerosos cursos sobre accesibilidad y discapacidad.

Otros requerimientos

Debido a la posible futura expansión de nuestra empresa, puede ser necesario contratar a otros profesionales de los sectores anteriormente citados, además de algún administrativo encargado de tramitaciones.

Recursos Materiales

Instalaciones necesarias

Se necesita un local de aproximadamente 40 m² que albergue tres puestos de trabajo. Además, el local poseerá un pequeño cuarto de baño adaptado.

La empresa contratará a tiempo parcial al terapeuta y los dos ingenieros de Telecomunicación trabajarán a tiempo completo.

Equipamiento

Serán necesarios tres puestos de trabajo completos. Contaremos con alguna silla y mesa auxiliar, y estanterías.

Además, habrá una impresora multifunción para la oficina. Dispondremos de una red de datos con su correspondiente equipamiento de red que emplearemos para compartir información en la oficina y tener acceso a Internet.

Vida Útil

Debido a que el único material que tendremos en las oficinas serán equipos informáticos, la vida útil de los mismos será de aproximadamente 2 años. Pasado ese tiempo, se sustituirán por equipos más modernos.

El router wifi que da acceso a Internet no requiere una renovación tan frecuente, por lo que podremos disponer de él durante un tiempo estimado de 10 años (salvo cambios en el diseño o en la tecnología de red utilizada por una posible expansión futura).

PLAN ECONÓMICO FINANCIERO

Ingresos

Con la comercialización de la aplicación "Dime", se estima una penetración mínima de mercado del 5%. Al existir, solo en España, 1.600.000 personas con una discapacidad en la comunicación, una penetración del 5% conlleva un total de 80.000 descargas.

A un precio de descarga de 60 €, al cual se le debe restar el 30% (18 €) que se queda Apple, se obtendrían unos ingresos de: 42 € x 80.000 descargas → 3.360.000 €.

Gastos

Directos

CONCEPTO	DESCRIPCIÓN		€/unidad	TOTAL
Equipamiento	Ordenadores portátiles	2 x MacBook Pro 15"		
		Procesador Core i7 de Intel de cuatro núcleos a 2,3 GHz		
		- 8GB 1333MHz DDR3 SDRAM - 2x4GB		
		- Disco duro Serial ATA de 750 GB a 5.400 rpm		
		- SuperDrive 8x (DVD±R DL/DVD±RW/CD-RW)		
		- Pantalla panorámica brillante para el MacBook Pro de 15 pulgadas		
		- Teclado retroiluminado (Español) & Guía del usuario (En Español)		
	Licencias	2 x licencias de desarrollador	79 €	158 €
	Impresora + Fax	1x Impresora multifuncional HP Officejet serie J3600	99,00 €	99,00 €
	Ipad	Ipad 2 WiFi + 3G, 16 GB	507,63 €	1.015,26 €
	Router Wifi	1x Router Linksys WRT120N	50,00 €	50,00 €
	Teléfonos	1x Pack Teléfonos Inalámbricos 3 terminales PhilipsCD1503B	59,99 €	59,99 €
Mobiliario	Mesas	4x Mesa escritorio	275,00 €	1.100,00 €
	Sillas	4x Silla escritorio	210,00 €	840,00 €
	Sillas espera	3x Silla visita	129,00 €	387,00 €
	Estanterias	1x Estanteria metal	140,00 €	140,00 €
Publicidad	Lámparas	4x Lámpara de mesa	40,00 €	160,00 €
		Lanzamiento al mercado		300,00 €
TOTAL inversión inicial				10.705,25 €

Estructura

CONCEPTO	DESCRIPCIÓN	TOTAL mes
Alquiler Local	Local de 40 m2	650,00 €
Publicidad	Publicidad (revistas, asociaciones, online,...)	150,00 €
Traslados	Gasto en gasolina visitas (terapeuta,...)	110,00 €
Teléfono e Internet	Coste de la línea telefónica y la línea de ADSL	63,00 €
Energía	Gasto eléctrico	35,00 €
Agua		15,00 €
TOTAL (€ mensuales)		1.023,00 €

Personal

- Ingeniero de Telecomunicación → Jornada completa → 2500 €/mes
- Terapeuta Ocupacional → Media Jornada → 1000 €/mes

PUESTO	SUBTOTAL	TOTAL
Ingenieros	2.500,00 €	5.000,00 €
Terapeuta	1.000,00 €	1.000,00 €
TOTAL		6.000,00 €

Inversiones

Viendo las estimaciones de cálculos realizadas y dado que no existe un histórico sobre empresas que se dediquen al diseño y desarrollo de aplicaciones móviles para la discapacidad en este tipo de soporte, en el peor de los casos, convendrá que los socios aporten capital suficiente para poder subsistir cinco meses sin recibir ningún ingreso.

	€/mes	€/5meses
Gastos directos	10.705,25 €	10.705,25 €
Estructura	1.023,00 €	5.115,00 €
Sueldos	6.000,00 €	30.000,00 €
TOTAL		40.710,365 €

Para cubrir la inversión inicial durante los cinco primeros meses de existencia, serían necesarias: $40.710,365 \text{ €} / 42\text{€} = 970$ descargas.

Población susceptible de comprar el producto: 1.600.000 discapacitados en España con problemas de comunicación.

Penetración de mercado necesaria para cubrir gastos: $970/1.600.000 = 0,0006 \%$

Caso peor: 5 % de la población discapacitada en España realiza una descarga:

5% de 1.600.000 = 80.000 usuarios x 42 euros de beneficio por descarga = 3.360.000 euros.

Caso mejor: 10 % de la población discapacitada en España realiza una descarga:

10% de 1.600.000 = 160.000 usuarios x 42 euros de beneficio por descarga = 6.720.000 euros.

Los cálculos se han realizado teniendo únicamente en cuenta una penetración de mercado en España. Sin embargo, la aplicación ya está también disponible en inglés, por lo que el número de usuarios a los que va dirigida aumenta, del mismo modo que lo hacen las posibles descargas de la misma.

DAFO

Debilidades

- Novedad del servicio: las personas con discapacidad funcional y las empresas que quieren adecuarse a la normativa pueden mostrarse reticentes a contratarlo.
- El hecho de ser una empresa nueva en el sector puede dificultar la tarea de encontrar una cartera de clientes adecuada.
- Dependiendo de las necesidades del cliente, el coste de los componentes (y por consiguiente, del proyecto) puede ser muy elevado.

Amenazas

- Existen empresas que venden los dispositivos y los instalan sin estudiar las necesidades concretas del cliente.
- El hecho de ser una de las primeras empresas abiertas de este tipo en Navarra, puede hacer que otras con más peso y experiencia aprovechen la idea e intenten dominar este mercado ü Aparición de empresas con el mismo objetivo.
- Momento económico actual.

Fortalezas

- Actualmente hay pocas empresas que ofrezcan este servicio.
- Comodidad y ahorro de tiempo buscando dispositivos.
- Determinación de qué se necesita realmente (avalado por un terapeuta especializado).
- Ofrece gran fiabilidad (revisión y mantenimiento del diseño).
- Trato y atención completamente personalizados.
- Contar con proveedores de gran reconocimiento internacional y de gran experiencia.
- Posibilidad de adaptación de algún componente ya existente en el mercado.

Oportunidades

- Recientes mejoras a las leyes de igualdad, dependencia y accesibilidad.
 - Las empresas poseen ayudas para la inmersión laboral de personas con discapacidad y para adecuar sus instalaciones.
 - Ayudas para las personas con dependencia o discapacidad (para que mejoren su calidad de vida).
- Ayudas creación nuevas empresas. Jóvenes emprendedores.

CONCLUSIONES

- Mercado todavía virgen y sin explotar.
- Gran apoyo por parte de la sociedad y de las instituciones (componente social).
- Posible expansión debida al incipiente aumento de demanda de servicios para las personas con discapacidad y/o dependencia (marco legal).
- Baja competencia (pocas empresas de este tipo en Navarra).
- No supone una inversión inicial excesiva (al poseer el primer prototipo ya diseñado y desarrollado, preparado para la venta).
- Observando el plan económico-financiero, el balance de nuestra empresa es positivo.
- Los primeros meses de existencia supondrán el mayor reto para la empresa.
- Debido a la posible aparición de empresas competidoras será necesario invertir un esfuerzo extra en los primeros meses de existencia y una constante inversión en I+D+I.

Anexo A: COMUNICADORAppDelegate.h

// **COMUNICADORAppDelegate.h**

#import <UIKit/UIKit.h>

#import <CoreData/CoreData.h>

@class COMUNICADORViewController;

@interface COMUNICADORAppDelegate : NSObject <UIApplicationDelegate> {

IBOutlet UIWindow *window;

IBOutlet COMUNICADORViewController *viewController;

@private

NSManagedObjectContext *managedObjectContext_;

NSManagedObjectModel *managedObjectModel_;

NSPersistentStoreCoordinator *persistentStoreCoordinator_;

}

@property (nonatomic, retain) IBOutlet UIWindow *window;

@property (nonatomic, retain) IBOutlet COMUNICADORViewController
*viewController;@property (nonatomic, retain, readonly) NSManagedObjectContext
*managedObjectContext;@property (nonatomic, retain, readonly) NSManagedObjectModel
*managedObjectModel;@property (nonatomic, retain, readonly) NSPersistentStoreCoordinator
*persistentStoreCoordinator;

- (NSString *)applicationDocumentsDirectory;

@end

Anexo B: COMUNICADORAppDelegate.m

// **COMUNICADORAppDelegate.m**

#import "COMUNICADORAppDelegate.h"

#import "COMUNICADORViewController.h"

@implementation COMUNICADORAppDelegate

@synthesize window;

@synthesize viewController;

#pragma mark -

#pragma mark Application lifecycle

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:
(NSDictionary *)launchOptions {

// Override point for customization after app launch

[window addSubview:viewController.view];

[window makeKeyAndVisible];

return YES;

}

- (void)applicationWillResignActive:(UIApplication *)application {

/*

Sent when the application is about to move from active to inactive state. This can occur for certain types of temporary interruptions (such as an incoming phone call or SMS message) or when the user quits the application and it begins the transition to the background state.

Use this method to pause ongoing tasks, disable timers, and throttle down OpenGL ES frame rates. Games should use this method to pause the game.

*/

}

- (void)applicationDidBecomeActive:(UIApplication *)application {

/*

Restart any tasks that were paused (or not yet started) while the application was inactive. If the application was previously in the background, optionally refresh the user interface.

*/

}

/**

applicationWillTerminate: saves changes in the application's managed object context before the application terminates.

```
*/
```

```
- (void)applicationWillTerminate:(UIApplication *)application {
```

```
    NSError *error = nil;
    NSManagedObjectContext *managedObjectContext =
self.managedObjectContext;
    if (managedObjectContext != nil) {
        if ([managedObjectContext hasChanges] && ![managedObjectContext
save:&error]) {
```

```
            /*
```

Replace this implementation with code to handle the error appropriately.

abort() causes the application to generate a crash log and terminate. You should not use this function in a shipping application, although it may be useful during development. If it is not possible to recover from the error, display an alert panel that instructs the user to quit the application by pressing the Home button.

```
            */
```

```
            NSLog(@"Unresolved error %@, %@", error, [error userInfo]);
            abort();
```

```
        }
```

```
    }
```

```
}
```

```
#pragma mark -
```

```
#pragma mark Core Data stack
```

```
/**
```

Returns the managed object context for the application.

If the context doesn't already exist, it is created and bound to the persistent store coordinator for the application.

```
*/
```

```
- (NSManagedObjectContext *)managedObjectContext {
```

```
    if (managedObjectContext_ != nil) {
        return managedObjectContext_;
    }
```

```
    NSPersistentStoreCoordinator *coordinator = [self persistentStoreCoordinator];
```

```
    if (coordinator != nil) {
        managedObjectContext_ = [[NSManagedObjectContext alloc] init];
        [managedObjectContext_ setPersistentStoreCoordinator:coordinator];
    }
}
```

```
    return managedObjectContext_;
}

/**
Returns the managed object model for the application.
If the model doesn't already exist, it is created from the application's model.
*/

- (NSManagedObjectContext *)managedObjectContext {

    if (managedObjectContext_ != nil) {
        return managedObjectContext_;
    }
    //NSURL *modelURL = [[NSBundle mainBundle]
URLForResource:@"COMUNICADOR" withExtension:@"momd"];
    //managedObjectContext_ = [[NSManagedObjectContext alloc]
initWithContentsOfURL:modelURL];
    managedObjectContext_ = [[NSManagedObjectContext
mergedModelFromBundles:nil] retain];
    return managedObjectContext_;
}

/**
Returns the persistent store coordinator for the application.
If the coordinator doesn't already exist, it is created and the application's store added
to it.
*/

/*
- (NSPersistentStoreCoordinator *)persistentStoreCoordinator {

    if (persistentStoreCoordinator_ != nil) {
        return persistentStoreCoordinator_;
    }

    NSString *storePath = [[self applicationDocumentsDirectory]
stringByAppendingPathComponent:@"COMUNICADOR.sqlite"];

    Set up the store.
    For the sake of illustration, provide a pre-populated default store.
    */

/*
    NSFileManager *fileManager = [NSFileManager defaultManager];

```

```
// If the expected store doesn't exist, copy the default store.
if (![fileManager fileExistsAtPath:storePath]) {
    NSString *defaultStorePath = [[NSBundle mainBundle]
pathForResource:@"COMUNICADOR" ofType:@"sqlite"];
    if (defaultStorePath) {
        [fileManager copyItemAtPath:defaultStorePath toPath:storePath
error:NULL];
    }
}

NSURL *storeUrl = [NSURL URLWithString:storePath];

NSError *error;
persistentStoreCoordinator_ = [[NSPersistentStoreCoordinator alloc]
initWithManagedObjectModel: [self managedObjectModel]];
if (![persistentStoreCoordinator_ addPersistentStoreWithType:NSSQLiteStoreType
configuration:nil URL:storeUrl options:nil error:&error]) {
    // Handle error
    NSLog(@"Unresolved error %@, %@", error, [error userInfo]);
    exit(-1); // Fail
}

return persistentStoreCoordinator_;
}

*/

/**
Returns the persistent store coordinator for the application.
If the coordinator doesn't already exist, it is created and the application's store added
to it.
*/

- (NSPersistentStoreCoordinator *)persistentStoreCoordinator {
    if (persistentStoreCoordinator_ != nil) {
        return persistentStoreCoordinator_;
    }
    NSString *storePath = [[self applicationDocumentsDirectory]
stringByAppendingPathComponent: @"COMUNICADOR.sqlite"];
    /*
Set up the store.
For the sake of illustration, provide a pre-populated default store.
*/

    NSFileManager *fileManager = [NSFileManager defaultManager];
    // If the expected store doesn't exist, copy the default store.
```

```
if (![fileManager fileExistsAtPath:storePath]) {
    NSString *defaultStorePath = [[NSBundle mainBundle]
pathForResource:@"COMUNICADOR" ofType:@"sqlite"];
    if (defaultStorePath) {
        [fileManager copyItemAtPath:defaultStorePath toPath:storePath
error:NULL];
    }
}

NSURL *storeUrl = [NSURL fileURLWithPath:storePath];
//NSDictionary *options = [NSDictionary dictionaryWithObjectsAndKeys:
[NSNumber numberWithInt:YES], NSMigratePersistentStoresAutomaticallyOption,
[NSNumber numberWithInt:YES], NSInferMappingModelAutomaticallyOption, nil];
//persistentStoreCoordinator_ = [[NSPersistentStoreCoordinator alloc]
initWithManagedObjectModel: [self managedObjectModel]];

NSError *error;
persistentStoreCoordinator_ = [[NSPersistentStoreCoordinator alloc]
initWithManagedObjectModel: [self managedObjectModel]];
if (![persistentStoreCoordinator_ addPersistentStoreWithType:NSSQLiteStoreType
configuration:nil URL:storeUrl options:nil error:&error]) {
    // Handle error
    NSLog(@"Unresolved error %@, %@", error, [error userInfo]);
    exit(-1); // Fail
}

return persistentStoreCoordinator_;
/*
NSError *error;
if (![persistentStoreCoordinator_
addPersistentStoreWithType:NSSQLiteStoreType configuration:nil URL:storeUrl
options:options error:&error]) {
    // Update to handle the error appropriately.
    NSLog(@"Unresolved error %@, %@", error, [error userInfo]);
    exit(-1); // Fail
}

return persistentStoreCoordinator_;
*/
/*
if (persistentStoreCoordinator_ != nil) {
    return persistentStoreCoordinator_;
}
*/
```

```
NSURL *storeURL = [[self applicationDocumentsDirectory]
URLByAppendingPathComponent:@"COMUNICADOR.sqlite"];
```

```
NSError *error = nil;
persistentStoreCoordinator_ = [[NSPersistentStoreCoordinator alloc]
initWithManagedObjectModel:[self managedObjectModel]];
if (![persistentStoreCoordinator_ addPersistentStoreWithType:NSSQLiteStoreType
configuration:nil URL:storeURL options:nil error:&error]) {
```

Replace this implementation with code to handle the error appropriately.

`abort()` causes the application to generate a crash log and terminate. You should not use this function in a shipping application, although it may be useful during development. If it is not possible to recover from the error, display an alert panel that instructs the user to quit the application by pressing the Home button.

Typical reasons for an error here include:

- * The persistent store is not accessible;
- * The schema for the persistent store is incompatible with current managed object model.

Check the error message to determine what the actual problem was.

If the persistent store is not accessible, there is typically something wrong with the file path. Often, a file URL is pointing into the application's resources directory instead of a writeable directory.

If you encounter schema incompatibility errors during development, you can reduce their frequency by:

- * Simply deleting the existing store:

```
[[NSFileManager defaultManager] removeItemAtURL:storeURL error:nil]
```

* Performing automatic lightweight migration by passing the following dictionary as the options parameter:

```
[NSDictionary dictionaryWithObjectsAndKeys:[NSNumber
numberWithBool:YES],NSMigratePersistentStoresAutomaticallyOption, [NSNumber
numberWithBool:YES], NSInferMappingModelAutomaticallyOption, nil];
```

Lightweight migration will only work for a limited set of schema changes; consult "Core Data Model Versioning and Data Migration Programming Guide" for details.

```
*/
```

```
/*
```

```
NSLog(@"Unresolved error %@, %@", error, [error userInfo]);
abort();
```

```
    }

    return persistentStoreCoordinator_;
*/
}

#pragma mark -
#pragma mark Application's Documents directory

/**
Returns the URL to the application's Documents directory.
*/
- (NSString *)applicationDocumentsDirectory {

    return [NSSearchPathForDirectoriesInDomains(NSDocumentDirectory,
NSUserDomainMask, YES) lastObject];

    //return [[[NSFileManager defaultManager]
URLsForDirectory:NSDocumentDirectory inDomains:NSUserDomainMask] lastObject];
}

#pragma mark -

#pragma mark Memory management

- (void)dealloc {

    [managedObjectContext_ release];
    [managedObjectModel_ release];
    [persistentStoreCoordinator_ release];

    [viewController release];
    [window release];
    [super dealloc];
}

@end
```


Anexo C: COMUNICADORViewController.h

// COMUNICADORViewController.h

```
#import <UIKit/UIKit.h>
#import <QuartzCore/QuartzCore.h>
#import <CoreData/CoreData.h>

@class Pizarra;
@class Reloj;
@class Ajustes;
@class Personalizacion;

@class seleccionAlbum;
@class controlAgenda;

@interface COMUNICADORViewController : UIViewController
{

    //PANTALLA INICIAL
    IBOutlet UIView *VISTA_PRIMERA;
    NSTimer *pulsacionTimer;

    //-----AGENDA-----

    controlAgenda *controladorAgenda; //Instancia de la clase "control Agenda"

    //-----ALBUM-----

    seleccionAlbum *controladorAlbum; //Instancia de la clase "selección Album"

    //-----PIZARRA-----

    Pizarra *controladorPizarra;

    //-----RELOJ-----

    Reloj *controladorReloj;

    //-----AJUSTES-----

    Ajustes *controladorAjustes;

    //-----PERSONALIZACION-----

    Personalizacion *controladorPersonalizacion;
```

```
//-----
```

```
//-----COMUNICADOR-----
```

```
IBOutlet UIView *VISTA_PRINCIPAL;
```

```
//vistas base del comunicador contenidas en vista principal
```

```
IBOutlet UIView *VISTA_MEDIO_BASE;  
IBOutlet UIView *VISTA_ABAJO_IZQUIERDA_BASE;  
IBOutlet UIView *VISTA_ABAJO_BORRAR_BASE;  
IBOutlet UIView *VISTA_ABAJO_BOTONES_BASE;  
IBOutlet UIView *VISTA_ARRIBA_BASE;  
IBOutlet UIView *VISTA_ARRIBA_BOTON_BASE;
```

```
//AJUSTAR
```

```
int ajuste_resp_negativa;  
int ajuste_tiempo_pulsacion;  
int ajuste_tipo_letra;
```

```
//elementos que se superponen en la vista superior (Vista Arriba Base)
```

```
IBOutlet UIView *VISTA_ARRIBA_BOTON;  
IBOutlet UIView *VISTA_ARRIBA;
```

```
//elementos que se superponen en la vista inferior (Vista Abajo Base)
```

```
IBOutlet UIView *VISTA_ABAJO_IZQUIERDA;  
IBOutlet UIView *VISTA_ABAJO_BORRAR;  
IBOutlet UIView *VISTA_ABAJO_BOTONES;  
IBOutlet UIView *VISTA_ABAJO_IZQUIERDA_NO_PREGUNTA;
```

```
//AJUSTAR
```

```
IBOutlet UIButton * BOTON_BORRAR;  
IBOutlet UIButton * ajuste_bot_inicio;  
IBOutlet UIButton * ajuste_bot_decirno;  
IBOutlet UIButton * ajuste_bot_final;
```

```
//Botones izquierda y derecha correspondientes a Vista Abajo Botones
```

```
IBOutlet id BOTON_PAGINA_DERECHA;  
IBOutlet id BOTON_PAGINA_IZQUIERDA;
```

```
//elementos que se superponen a la vista del medio (Vista Medio Base)
```

```
IBOutlet UIView *VISTA_MEDIO;  
IBOutlet UIView *VISTA_MEDIO_SUBLUGARES;
```

```
IBOutlet UIView *VISTA_MEDIO QUIERO_ESTOY;
```

```
//Pantalla final donde sale la frase
```

```
IBOutlet UIView *VISTA_SECUNDARIA;
```

```
//Vista base y vista que muestra o no el boton que niega dentro de Vista  
Secundaria
```

```
IBOutlet UIView *VISTA_BASE_BOTON_NO;
```

```
IBOutlet UIView *VISTA_BOTON_NO;
```

```
//Vista que muestra el no
```

```
IBOutlet UIView *VISTA_NO;
```

```
//Vista para salir de la vista secundaria
```

```
IBOutlet id BOTON_SALIR;
```

```
//Diseño Core Data
```

```
UIButton *Imagen1;  
UIButton *Imagen2;  
UIButton *Imagen3;  
UIButton *Imagen4;  
UIButton *Imagen5;  
UIButton *Imagen6;  
UILabel *Label1;  
UILabel *Label2;  
UILabel *Label3;  
UILabel *Label4;  
UILabel *Label5;  
UILabel *Label6;
```

```
UIButton *Boton_quiero;  
UIButton *Boton_estoy;  
UILabel *Label_quiero;  
UILabel *Label_estoy;
```

```
UIButton *Imagen_lugar1;  
UIButton *Imagen_lugar2;  
UIButton *Imagen_lugar3;  
UIButton *Imagen_lugar4;  
UILabel *Label_lugar1;  
UILabel *Label_lugar2;  
UILabel *Label_lugar3;
```

```
UILabel *Label_lugar4;
```

//Predicados para sublugares

```
NSString *entidad;  
NSString *entidad1;  
NSString *entidad_a_modificar;  
NSString *predicado_base;  
NSString *predicado_sublugar;  
NSString *predicado1;  
NSString *predicado2;  
NSString *predicado3;  
NSString *predicado4;  
NSString *predicado5;  
NSString *predicado6;
```

//Imágenes de la frase

```
UIImageView *Imagen_frase1;  
UIImageView *Imagen_frase2;  
UIImageView *Imagen_frase3;  
UILabel *Label_frase1;  
UILabel *Label_frase2;  
UILabel *Label_frase3;  
UIImageView *Imagen_frase_interrogante;
```

//Imágenes de la Vista secundaria

```
UIImageView *Imagen_secundaria1;  
UIImageView *Imagen_secundaria2;  
UIImageView *Imagen_secundaria3;  
UIImageView *Imagen_secundaria_interrogante;  
UILabel *Label_secundaria1;  
UILabel *Label_secundaria2;  
UILabel *Label_secundaria3;
```

```
NSUInteger INDICE;  
int INDICE_PAGINAS;
```

```
NSInteger DERECHA;  
NSInteger MENU;  
NSInteger NEGAR;  
NSInteger INTERROGANTE;  
NSInteger QUIEROESTOY;  
NSInteger QE;
```

```
IBOutlet UIButton *BOTON_NEGAR;  
CALayer *negacion;  
CALayer *negacion_secundaria;
```

```
//Strings para asignar el idioma a las imágenes del comunicador
```

```
NSString *Etiqueta1;  
NSString *Etiqueta2;  
NSString *Etiqueta3;  
NSString *Etiqueta4;  
NSString *Etiqueta5;  
NSString *Etiqueta6;
```

```
//En Idioma irá el string correspondiente a la columna de la base de datos que  
quiero
```

```
consultar
```

```
NSString *Idioma;
```

```
UIImage *btnImage1;  
UIImage *btnImage2;  
UIImage *btnImage3;  
UIImage *btnImage4;  
UIImage *btnImage5;  
UIImage *btnImage6;
```

```
NSInteger CONTADOR_ANTERIOR;  
NSInteger FILA_A_CAMBIAR;
```

```
NSInteger PK_1;  
NSInteger PK_2;  
NSInteger PK_3;  
NSInteger PK_4;  
NSInteger PK_5;  
NSInteger PK_6;
```

```
NSInteger EsNuevo_1;  
NSInteger EsNuevo_2;
```

```
NSInteger EsNuevo_3;  
NSInteger EsNuevo_4;  
NSInteger EsNuevo_5;  
NSInteger EsNuevo_6;
```

```
NSInteger VALOR_ANTIGUO_CONTADOR_1;  
NSInteger VALOR_ANTIGUO_CONTADOR_2;  
NSInteger VALOR_ANTIGUO_CONTADOR_3;  
NSInteger VALOR_ANTIGUO_CONTADOR_4;  
NSInteger VALOR_ANTIGUO_CONTADOR_5;  
NSInteger VALOR_ANTIGUO_CONTADOR_6;
```

```
NSInteger recorrido;
```

```
}
```

```
@property (nonatomic, retain) NSTimer *pulsacionTimer;  
@property (nonatomic, retain) IBOutlet UIButton *Imagen1;  
@property (nonatomic, retain) IBOutlet UIButton *Imagen2;  
@property (nonatomic, retain) IBOutlet UIButton *Imagen3;  
@property (nonatomic, retain) IBOutlet UIButton *Imagen4;  
@property (nonatomic, retain) IBOutlet UIButton *Imagen5;  
@property (nonatomic, retain) IBOutlet UIButton *Imagen6;  
@property (nonatomic, retain) IBOutlet UILabel *Label1;  
@property (nonatomic, retain) IBOutlet UILabel *Label2;  
@property (nonatomic, retain) IBOutlet UILabel *Label3;  
@property (nonatomic, retain) IBOutlet UILabel *Label4;  
@property (nonatomic, retain) IBOutlet UILabel *Label5;  
@property (nonatomic, retain) IBOutlet UILabel *Label6;  
@property (nonatomic, retain) IBOutlet UILabel *Label_quiero;  
@property (nonatomic, retain) IBOutlet UILabel *Label_estoy;  
@property (nonatomic, retain) IBOutlet UIButton *Boton_quiero;  
@property (nonatomic, retain) IBOutlet UIButton *Boton_estoy;  
@property (nonatomic, retain) IBOutlet UIButton *Imagen_lugar1;  
@property (nonatomic, retain) IBOutlet UIButton *Imagen_lugar2;  
@property (nonatomic, retain) IBOutlet UIButton *Imagen_lugar3;  
@property (nonatomic, retain) IBOutlet UIButton *Imagen_lugar4;  
@property (nonatomic, retain) IBOutlet UILabel *Label_lugar1;  
@property (nonatomic, retain) IBOutlet UILabel *Label_lugar2;  
@property (nonatomic, retain) IBOutlet UILabel *Label_lugar3;  
@property (nonatomic, retain) IBOutlet UILabel *Label_lugar4;  
@property (nonatomic, retain) IBOutlet UIImageView *Imagen_frase1;  
@property (nonatomic, retain) IBOutlet UIImageView *Imagen_frase2;  
@property (nonatomic, retain) IBOutlet UIImageView *Imagen_frase3;  
@property (nonatomic, retain) IBOutlet UIImageView *Imagen_frase_interrogante;  
@property (nonatomic, retain) IBOutlet UILabel *Label_frase1;
```

```
@property (nonatomic, retain) IBOutlet UILabel *Label_frase2;
@property (nonatomic, retain) IBOutlet UILabel *Label_frase3;
@property (nonatomic, retain) IBOutlet UIImageView *Imagen_secundaria1;
@property (nonatomic, retain) IBOutlet UIImageView *Imagen_secundaria2;
@property (nonatomic, retain) IBOutlet UIImageView *Imagen_secundaria3;
@property (nonatomic, retain) IBOutlet UIImageView
*Imagen_secundaria_interrogante;
@property (nonatomic, retain) IBOutlet UILabel *Label_secundaria1;
@property (nonatomic, retain) IBOutlet UILabel *Label_secundaria2;
@property (nonatomic, retain) IBOutlet UILabel *Label_secundaria3;

@property (nonatomic, retain) IBOutlet UIView *VISTA_BASE_BOTON_NO;
@property (nonatomic, retain) IBOutlet UIView *VISTA_BOTON_NO;
```

```
@property (nonatomic, retain) Pizarra *controladorPizarra;
@property (nonatomic, retain) Reloj *controladorReloj;
@property (nonatomic, retain) Ajustes *controladorAjustes;
@property (nonatomic, retain) Personalizacion *controladorPersonalizacion;
@property (nonatomic, retain) seleccionAlbum *controladorAlbum;
@property (nonatomic, retain) controlAgenda *controladorAgenda;
```

```
//Ajustes
```

```
@property (nonatomic, retain) IBOutlet UIButton * BOTON_BORRAR;
@property (nonatomic, retain) IBOutlet UIButton * ajuste_bot_inicio;
@property (nonatomic, retain) IBOutlet UIButton * ajuste_bot_decirno;
@property (nonatomic, retain) IBOutlet UIButton * ajuste_bot_final;
```

```
- (IBAction) PULSAR_VOLVER_COMUNICADOR: (id) sender;
```

```
- (void)showVista;
- (IBAction)touchDown:(id)sender;
- (IBAction)touchUp:(id)sender;
```

```
//-----COMUNICADOR-----
```

```
- (IBAction) PULSAR_BOTON_PAGINA_DERECHA: (id) sender;
- (IBAction) PULSAR_BOTON_PAGINA_IZQUIERDA: (id) sender;
- (IBAction) PULSAR_BOTON_BORRAR: (id) sender;
- (IBAction) PULSAR_BOTON_SALIR: (id) sender;
```

```
- (IBAction) PULSAR_BOTON_ARRIBA_OK: (id) sender;
```

```
- (IBAction) PULSAR_BOTON_AJUSTES: (id) sender;
```


- (IBAction) PULSAR_BOTON_NO: (id) sender;
- (IBAction) PULSAR_BOTON_SALIR_NO: (id) sender;
- (IBAction) PULSAR_BOTON_NEGAR: (id) sender;
- (IBAction) PULSAR_BOTON_INTERROGANTE: (id) sender;

- (void)Ajustar;

- (void) BusquedaLugares:(NSString *)entidad_busqueda;
- (void) BusquedaYSituacion:(NSString *) entidad_busqueda Predicado:
(NSString*)predicado_busqueda;
- (void) BusquedaYSituacionFrase:(NSString *)entidad_busqueda Predicado:
(NSString*)predicado_busqueda;
- (void) BusquedaYSituacionBorrar:(NSString *)entidad_busqueda Predicado:
(NSString*)predicado_busqueda;
- (void) PonerLugares:(NSArray *)objects Objeto:(NSManagedObject *)matches
Numero:(int)valor;
 - (void) PonerImágenes:(NSArray *)objects Objeto:(NSManagedObject *)matches
Numero:(int) valor;

- (void) IniciarImágenes;
- (void) DeshabilitarImágenesYEtiquetas;
- (void) HabilitarImágenes;
- (void) HabilitarEtiquetas;
- (void) BusquedaBotonesPaginas;
- (void) ModificarContadores: (int) fila_a_cambiar ValorAnterior: (int)
valor_contador_anterior Entidad: (NSString *) entidad_modificar;

@end

Anexo D: COMUNICADORViewController.m

// **COMUNICADORViewController.m**

```
#import "COMUNICADORAppDelegate.h"  
#import "COMUNICADORViewController.h"  
#import <QuartzCore/QuartzCore.h>  
#import <AudioToolbox/AudioServices.h>  
#import "Pizarra.h"  
#import "Reloj.h"  
#import "Ajustes.h"  
#import "Personalizacion.h"  
#import "seleccionAlbum.h"  
#import "controlAgenda.h"
```

```
@implementation COMUNICADORViewController
```

```
@synthesize BOTON_BORRAR;  
@synthesize ajuste_bot_inicio;  
@synthesize ajuste_bot_decirno;  
@synthesize ajuste_bot_final;
```

```
@synthesize controladorAlbum;  
@synthesize controladorAgenda;  
@synthesize controladorPizarra;  
@synthesize controladorReloj;  
@synthesize controladorAjustes;  
@synthesize controladorPersonalizacion;
```

```
@synthesize pulsacionTimer;  
@synthesize VISTA_BASE_BOTON_NO, VISTA_BOTON_NO;  
@synthesize Imagen_lugar1, Imagen_lugar2, Imagen_lugar3, Imagen_lugar4,  
Label_lugar1, Label_lugar2, Label_lugar3, Label_lugar4, Boton_estoy, Boton_quiero,  
Label_estoy, Label_quiero;
```

```
@synthesize Imagen1, Imagen2, Imagen3, Imagen4, Imagen5, Imagen6, Label1,  
Label2, Label3, Label4, Label5, Label6;  
@synthesize Imagen_frase1, Imagen_frase2, Imagen_frase3,  
Imagen_frase_interrogante;  
@synthesize Label_frase1, Label_frase2, Label_frase3;  
@synthesize Imagen_secundaria1, Imagen_secundaria2, Imagen_secundaria3,  
Imagen_secundaria_interrogante;  
@synthesize Label_secundaria1, Label_secundaria2, Label_secundaria3;
```

```
//Inicializo el picker, la etiqueta label y el arrayDatos
```

```
- (void)viewDidLoad {

    QUIEROESTOY=0;
    QE=0;

    [self Ajustar];

    controladorAlbum=[[seleccionAlbum alloc] initWithNibName:@"seleccionAlbum"
bundle:nil]; //Eli. Iniciamos el archivo xib de la siguiente pantalla.
    controladorAgenda=[[controlAgenda alloc] initWithNibName:@"controlAgenda"
bundle:nil]; //Eli. Iniciamos el archivo xib de la siguiente pantalla.

    //controladorAjustes.pulsacion=NO;
    controladorPizarra=[[Pizarra alloc] initWithNibName:@"Pizarra" bundle:nil];
    controladorReloj=[[Reloj alloc] initWithNibName:@"Reloj" bundle:nil];
    controladorAjustes=[[Ajustes alloc] initWithNibName:@"Ajustes" bundle:nil];
    controladorPersonalizacion=[[Personalizacion alloc]
initWithNibName:@"Personalizacion" bundle:nil];
    //controladorAjustes.tipo_letra=20;
    if (controladorAjustes.tipo_letra<40)
    {
        [BOTON_BORRAR setTitle:@"BORRAR" forState:normal];
        [ajuste_bot_decirno setTitle:@"DECIR NO" forState:normal];
        [ajuste_bot_inicio setTitle:@"INICIO" forState:normal];
        [ajuste_bot_final setTitle:@"INICIO" forState:normal];

    }
    else
    {
        [BOTON_BORRAR setTitle:@"DELETE" forState:normal];
        [ajuste_bot_decirno setTitle:@"SAY NO" forState:normal];
        [ajuste_bot_inicio setTitle:@"HOME" forState:normal];
        [ajuste_bot_final setTitle:@"HOME" forState:normal];

    }

    [super viewDidLoad];
}

- (void)Ajustar
{
    [controladorAjustes viewDidLoad];

    ajuste_resp_negativa=controladorAjustes.resp_negativa;
    ajuste_tiempo_pulsacion=controladorAjustes.tiempo_pulsacion;
    ajuste_tipo_letra=controladorAjustes.tipo_letra;
```

```
controladorAjustes.resp_negativa=ajuste_resp_negativa;
controladorAjustes.tipo_letra=ajuste_tipo_letra;
controladorAjustes.tiempo_pulsacion=ajuste_tiempo_pulsacion;

    NSLog(@"respuesta negativa %i, tiempo pulsacion %i, tipo letra
%i",ajuste_resp_negativa,ajuste_tiempo_pulsacion,ajuste_tipo_letra);
}

// Override to allow orientations other than the default portrait orientation.
- (BOOL)shouldAutorotateToInterfaceOrientation:
(UIInterfaceOrientation)interfaceOrientation {
    if((interfaceOrientation==UIInterfaceOrientationLandscapeLeft)||
(interfaceOrientation==UIInterfaceOrientationLandscapeRight))
        return YES;
    return NO;
}

- (void)didReceiveMemoryWarning {
    // Releases the view if it doesn't have a superview.
    [super didReceiveMemoryWarning];

    // Release any cached data, images, etc that aren't in use.
}

//Configuración pulsación pantalla inicial

-(IBAction)touchDown:(id)sender{
    self.pulsacionTimer = [NSTimer
scheduledTimerWithTimeInterval:controladorAjustes.tiempo_pulsacion target:self
selector:@selector(showVista) userInfo:sender repeats:NO];
}

-(IBAction)touchUp:(id)sender{
    [pulsacionTimer invalidate];
}

-(void)showVista{

    [self Ajustar];

    switch([pulsacionTimer.userInfo tag]){

        /*
        Caso 0: Comunicador
```

Caso 1: Reloj
Caso 2:
Caso 3: QuieroEstoy
Caso 4: Pizarra
Caso 5: Álbum
*/

```
if (controladorAjustes.tipo_letra<40)
{

    [BOTON_BORRAR setTitle:@"BORRAR" forState:normal];
    [ajuste_bot_decirno setTitle:@"DECIR NO" forState:normal];
    [ajuste_bot_inicio setTitle:@"INICIO" forState:normal];
    [ajuste_bot_final setTitle:@"INICIO" forState:normal];

}
else
{
    [BOTON_BORRAR setTitle:@"DELETE" forState:normal];
    [ajuste_bot_decirno setTitle:@"SAY NO" forState:normal];
    [ajuste_bot_inicio setTitle:@"HOME" forState:normal];
    [ajuste_bot_final setTitle:@"HOME" forState:normal];

}

case 0:

if (controladorAjustes.tipo_letra<40)
{
    [BOTON_BORRAR setTitle:@"BORRAR" forState:normal];
    [ajuste_bot_decirno setTitle:@"DECIR NO" forState:normal];
    [ajuste_bot_inicio setTitle:@"INICIO" forState:normal];
    [ajuste_bot_final setTitle:@"INICIO" forState:normal];

}
else
{
    [BOTON_BORRAR setTitle:@"DELETE" forState:normal];
    [ajuste_bot_decirno setTitle:@"SAY NO" forState:normal];
    [ajuste_bot_inicio setTitle:@"HOME" forState:normal];
    [ajuste_bot_final setTitle:@"HOME" forState:normal];

}

}
```

```
[self.view addSubview:VISTA_PRINCIPAL];
[VISTA_PRINCIPAL addSubview:VISTA_ABAJO_IZQUIERDA_BASE];
[VISTA_PRINCIPAL addSubview:VISTA_MEDIO_BASE];
[VISTA_PRINCIPAL addSubview:VISTA_ABAJO_BORRAR_BASE];
[VISTA_PRINCIPAL addSubview:VISTA_ABAJO_BOTONES_BASE];
[VISTA_PRINCIPAL addSubview:VISTA_ARRIBA_BASE];
[VISTA_PRINCIPAL addSubview:VISTA_ARRIBA_BOTON_BASE];

[VISTA_ARRIBA_BASE addSubview:VISTA_ARRIBA];

[VISTA_MEDIO_BASE addSubview:VISTA_MEDIO];

[BOTON_PAGINA_IZQUIERDA removeFromSuperview];

[VISTA_ABAJO_IZQUIERDA_BASE
addSubview:VISTA_ABAJO_IZQUIERDA];

[VISTA_SECUNDARIA addSubview:VISTA_BASE_BOTON_NO];
Label_frase1.hidden=YES;
Label_frase2.hidden=YES;
Label_frase3.hidden=YES;

[self HabilitarImagenes];
[self HabilitarEtiquetas];
[self BusquedaLugares:@"Lugar"];

INDICE=0;
DERECHA=0;
MENU=0;
NEGAR=0;
INTERROGANTE=0;
QUIEROESTOY=0;
INDICE_PAGINAS=6;
CONTADOR_ANTERIOR=0;
FILA_A_CAMBIAR=5;

break;

case 1:
[controladorReloj viewDidLoad];
[VISTA_PRIMERA addSubview:controladorReloj.view];

break;

case 2:
```

```
[self.view addSubview:controladorAgenda.view];//añadimos la vista del controlador de la agenda. Eli
```

```
        NSLog(@"Estoy pulsando la agenda");  
        break;  
  
    case 3:  
  
        if (controladorAjustes.tipo_letra<40)  
        {  
            [BOTON_BORRAR setTitle:@"BORRAR" forState:normal];  
            [ajuste_bot_decirno setTitle:@"DECIR NO" forState:normal];  
            [ajuste_bot_inicio setTitle:@"INICIO" forState:normal];  
            [ajuste_bot_final setTitle:@"INICIO" forState:normal];  
  
        }  
        else  
        {  
            [BOTON_BORRAR setTitle:@"DELETE" forState:normal];  
            [ajuste_bot_decirno setTitle:@"SAY NO" forState:normal];  
            [ajuste_bot_inicio setTitle:@"HOME" forState:normal];  
            [ajuste_bot_final setTitle:@"HOME" forState:normal];  
  
        }  
  
        [self.view addSubview:VISTA_PRINCIPAL];  
        [VISTA_PRINCIPAL addSubview:VISTA_ABAJO_IZQUIERDA_BASE];  
        [VISTA_PRINCIPAL addSubview:VISTA_MEDIO_BASE];  
        [VISTA_PRINCIPAL addSubview:VISTA_ABAJO_BORRAR_BASE];  
        [VISTA_PRINCIPAL addSubview:VISTA_ABAJO_BOTONES_BASE];  
        [VISTA_PRINCIPAL addSubview:VISTA_ARRIBA_BASE];  
        [VISTA_PRINCIPAL addSubview:VISTA_ARRIBA_BOTON_BASE];  
  
        [VISTA_ARRIBA_BASE addSubview:VISTA_ARRIBA];  
  
        [VISTA_MEDIO_BASE addSubview:VISTA_MEDIO QUIERO_ESTOY];  
  
        [BOTON_PAGINA_IZQUIERDA removeFromSuperview];  
  
        [VISTA_ABAJO_IZQUIERDA_BASE  
addSubview:VISTA_ABAJO_IZQUIERDA];  
  
        [VISTA_SECUNDARIA addSubview:VISTA_BASE_BOTON_NO];
```



```
QE=1;
    [self HabilitarImágenes];
[self HabilitarEtiquetas];
[self BusquedaLugares:@"Parte1"];
Label_frase1.hidden=YES;
Label_frase2.hidden=YES;
Label_frase3.hidden=YES;

    INDICE=0;
    DERECHA=0;
    MENU=0;
    NEGAR=0;
    INTERROGANTE=0;
    QUIEROESTOY=0;
    INDICE_PAGINAS=6;
    CONTADOR_ANTERIOR=0;

FILA_A_CAMBIAR=0;
    break;

case 4:

    [VISTA_PRIMERA addSubview:controladorPizarra.view];

break;
case 5:
    //[VISTA_PRIMERA addSubview:VISTA_PRIMERA_ALBUM];
    [self.view addSubview:controladorAlbum.view];//añadimos la vista del
controlador del álbum. Eli

    NSLog(@"Estoy pulsando el álbum");
    break;

case 6:

    //[VISTA_PRIMERA addSubview:controladorPersonalizacion.view];

break;

/*
Caso 11: Casa
Caso 12: Colegio
```

Caso 13: Ciudad
Caso 14: Vacaciones
Caso 15: Quiero
Caso 16: Estoy
*/

case 11:

```
predicado_base=@"ANY SublugarLugar like '1' AND activo=1";  
predicado1=@"ANY FraseSublugar like '2' AND activo=1";  
predicado2=@"ANY FraseSublugar like '6' AND activo=1";  
predicado3=@"ANY FraseSublugar like '7' AND activo=1";  
predicado4=@"ANY FraseSublugar like '11' AND activo=1";  
predicado5=@"ANY FraseSublugar like '19' AND activo=1";  
predicado6=@"ANY FraseSublugar like '21' AND activo=1";  
INDICE=1;
```

//Hago Búsqueda de Sublugares de Lugar=1 (Casa). Funcion
BusquedaYSituacion

```
[self BusquedaYSituacion:@"Sublugar" Predicado:@"ANY SublugarLugar like '1'  
AND activo=1"];
```

break;

case 12:

```
predicado_base=@"ANY SublugarLugar like '2' AND activo=1";  
predicado1=@"ANY FraseSublugar like '3' AND activo=1";  
predicado2=@"ANY FraseSublugar like '4' AND activo=1";  
predicado3=@"ANY FraseSublugar like '8' AND activo=1";  
predicado4=@"ANY FraseSublugar like '12' AND activo=1";  
predicado5=@"ANY FraseSublugar like '18' AND activo=1";  
predicado6=@"ANY FraseSublugar like '22' AND activo=1";  
INDICE=1;
```

//Hago Búsqueda de Sublugares de Lugar=2 (Colegio). Funcion
BusquedaYSituacion

```
[self BusquedaYSituacion:@"Sublugar" Predicado:@"ANY SublugarLugar  
like '2' AND activo=1"];
```

break;

case 13:

```
predicado_base=@"ANY SublugarLugar like '3' AND activo=1";  
predicado1=@"ANY FraseSublugar like '1' AND activo=1";  
predicado2=@"ANY FraseSublugar like '9' AND activo=1";  
predicado3=@"ANY FraseSublugar like '13' AND activo=1";  
predicado4=@"ANY FraseSublugar like '14' AND activo=1";  
predicado5=@"ANY FraseSublugar like '15' AND activo=1";  
predicado6=@"ANY FraseSublugar like '16' AND activo=1";
```

```
INDICE=1;
```

//Hago Búsqueda de Sublugares de Lugar=3 (Ciudad). Funcion
BusquedaYSituacion

```
[self BusquedaYSituacion:@"Sublugar" Predicado:@"ANY SublugarLugar  
like '3' AND activo=1"];
```

```
break;
```

```
case 14:
```

```
predicado_base=@"ANY SublugarLugar like '4' AND activo=1";  
predicado1=@"ANY FraseSublugar like '5' AND activo=1";  
predicado2=@"ANY FraseSublugar like '10' AND activo=1";  
predicado3=@"ANY FraseSublugar like '16' AND activo=1";  
predicado4=@"ANY FraseSublugar
```

```
like '17' AND activo=1";
```

```
predicado5=@"ANY FraseSublugar like '20' AND activo=1";  
predicado6=@"ANY FraseSublugar like '23' AND activo=1";
```

```
INDICE=1;
```

//Hago Búsqueda de Sublugares de Lugar=4 (Vacaciones). Funcion
BusquedaYSituacion

```
[self BusquedaYSituacion:@"Sublugar" Predicado:@"ANY SublugarLugar  
like '4' AND activo=1"];
```

```
break;
```

```
case 15:
```

```
Label_frase1.hidden=NO;  
Imagen_frase1.image=[UIImage imageNamed:Label_quiero.text];  
//Imagen_frase1.image=[UIImage imageNamed:Label_quiero.text];  
Label_frase1.text=Label_quiero.text;
```

```
predicado1=@"ANY Parte3Con2 like '1' AND activo=1";  
predicado2=@"ANY Parte3Con2 like '2' AND activo=1";  
predicado3=@"ANY Parte3Con2 like '3' AND activo=1";  
predicado4=@"ANY Parte3Con2 like '4' AND activo=1";  
predicado5=@"ANY Parte3Con2 like '5' AND activo=1";  
predicado6=@"ANY Parte3Con2 like '6' AND activo=1";
```

```
DERECHA=0;  
QUIEROESTOY=1;
```

```
[VISTA_ABAJO_IZQUIERDA_BASE  
addSubview:VISTA_ABAJO_IZQUIERDA_NO_PREGUNTA];
```

```
[BOTON_PAGINA_IZQUIERDA removeFromSuperview];  
[BOTON_PAGINA_DERECHA removeFromSuperview];
```

```
[self IniciarImagenes];
```

```
entidad=@"Parte2";  
predicado_sublugar=@"ANY Parte2Con1 like '1' AND activo=1";
```

```
INDICE=3;
```

```
//Hago Búsqueda de Sublugares de Lugar=5 (Quiero). Funcion  
BúsquedaYSituacion
```

```
[self BúsquedaYSituacion:entidad Predicado:predicado_sublugar];
```

```
break;
```

```
case 16:
```

```
Label_frase2.hidden=NO;  
Imagen_frase2.image=[UIImage imageNamed:Label_estoy.text];  
//Imagen_frase2.image=[UIImage imageNamed:Label_estoy.text];  
Label_frase2.text=Label_estoy.text;
```

```
QUIEROESTOY=2;
```

```
[VISTA_ABAJO_IZQUIERDA_BASE  
addSubview:VISTA_ABAJO_IZQUIERDA_NO_PREGUNTA];
```

```
DERECHA=0;
```

```
[BOTON_PAGINA_IZQUIERDA removeFromSuperview];
[BOTON_PAGINA_DERECHA removeFromSuperview];

[self IniciarImágenes];

entidad=@"Parte2";
predicado_sublugar=@"ANY Parte2Con1 like '2' AND activo=1";

INDICE=4;

//Hago Búsqueda de Sublugares de Lugar=6 (Estoy). Funcion
BúsquedaYSituacion

[self BúsquedaYSituacion:entidad Predicado:predicado_sublugar];

break;

/*
Caso 21: Primer botón Frase
Caso 22: Segundo botón Frase
Caso 23: Tercer botón Frase
Caso 24: Cuarto botón Frase
Caso 25: Quinto botón Frase

Caso 26: Sexto botón Frase
*/
case 21:

DERECHA=0;

[BOTON_PAGINA_IZQUIERDA
removeFromSuperview];
[BOTON_PAGINA_DERECHA removeFromSuperview];
FILA_A_CAMBIAR=PK_1;
CONTADOR_ANTERIOR=VALOR_ANTIGUO_CONTADOR_1;

[self IniciarImágenes];

if (INDICE==1){

predicado_sublugar=predicado1;
entidad=@"Personas";
[VISTA_ABAJO_IZQUIERDA_BASE
addSubview:VISTA_ABAJO_IZQUIERDA_NO_PREGUNTA];
```

```
    }  
    else if (INDICE==2){  
        entidad=@"Verbos";  
Label_frase1.hidden=NO;  
        Imagen_frase1.image=btnImage1;  
Label_frase1.text=Label1.text;  
  
[self ModificarContadores: FILA_A_CAMBIAR ValorAnterior:  
CONTADOR_ANTERIOR Entidad: @"Personas"];  
  
    }  
    else if (INDICE==3){  
        entidad=@"Complementos";  
entidad_a_modificar=@"Verbos";  
Label_frase2.hidden=NO;  
        Imagen_frase2.image=btnImage1;  
Label_frase2.text=Label1.text;  
        if (QUIEROESTOY==1) {  
entidad=@"Parte3";  
            predicado_sublugar=predicado1;  
        }  
    else{  
  
[self ModificarContadores: FILA_A_CAMBIAR ValorAnterior:  
CONTADOR_ANTERIOR Entidad: entidad_a_modificar];  
    }  
  
    }  
    else if (INDICE==4){  
  
entidad_a_modificar=@"Complementos";  
if (QUIEROESTOY==1){  
    entidad_a_modificar=@"Parte3";  
}  
Label_frase3.hidden=NO;  
        Imagen_frase3.image=btnImage1;  
Label_frase3.text=Label1.text;  
[VISTA_ARRIBA_BOTON_BASE  
addSubview:VISTA_ARRIBA_BOTON];
```

```
[self ModificarContadores: FILA_A_CAMBIAR
ValorAnterior: CONTADOR_ANTERIOR Entidad: entidad_a_modificar];
[self DeshabilitarImágenesYEtiquetas];
}

[self BusquedaYSituacionFrase:entidad Predicado:predicado_sublugar];

break;

case 22:

    DERECHA=0;
    [BOTON_PAGINA_IZQUIERDA removeFromSuperview];
    [BOTON_PAGINA_DERECHA removeFromSuperview];
    FILA_A_CAMBIAR=PK_2;
    CONTADOR_ANTERIOR=VALOR_ANTIGUO_CONTADOR_2;

    [self IniciarImágenes];

    if (INDICE==1){

        predicado_sublugar=predicado2;
        entidad=@"Personas";
        [VISTA_ABAJO_IZQUIERDA_BASE
addSubview:VISTA_ABAJO_IZQUIERDA_NO_PREGUNTA];

    }
    else if (INDICE==2){

        entidad=@"Verbos";
        Label_frase1.hidden=NO;
        Imagen_frase1.image=btnImage2;
        Label_frase1.text=Label2.text;

        [self ModificarContadores: FILA_A_CAMBIAR ValorAnterior:
CONTADOR_ANTERIOR Entidad: @"Personas"];

    }
    else if (INDICE==3){

        entidad=@"Complementos";
        entidad_a_modificar=@"Verbos";
        Label_frase2.hidden=NO;
        Imagen_frase2.image=btnImage2;
```

```
Label_frase2.text=Label2.text;
if (QUIEROESTOY==1) {

    entidad=@"Parte3";
        predicado_sublugar=predicado2;
    }
else{
    [self ModificarContadores: FILA_A_CAMBIAR ValorAnterior:
CONTADOR_ANTERIOR Entidad: entidad_a_modificar];
    }
}
else if (INDICE==4){

entidad_a_modificar=@"Complementos";
if (QUIEROESTOY==1){
    entidad_a_modificar=@"Parte3";
}
Label_frase3.hidden=NO;
    Imagen_frase3.image=btnImage2;
    Label_frase3.text=Label2.text;
    [VISTA_ARRIBA_BOTON_BASE
addSubview:VISTA_ARRIBA_BOTON];
    [self ModificarContadores: FILA_A_CAMBIAR
ValorAnterior: CONTADOR_ANTERIOR Entidad: entidad_a_modificar];
    [self DeshabilitarImágenesYEtiquetas];
    }

[self BusquedaYSituacionFrase:entidad Predicado:predicado_sublugar];

    break;
case 23:

    DERECHA=0;
    [BOTON_PAGINA_IZQUIERDA removeFromSuperview];
    [BOTON_PAGINA_DERECHA removeFromSuperview];
    FILA_A_CAMBIAR=PK_3;
    CONTADOR_ANTERIOR=VALOR_ANTIGUO_CONTADOR_3;

[self IniciarImágenes];

    if (INDICE==1){

        predicado_sublugar=predicado3;
        entidad=@"Personas";
        [VISTA_ABAJO_IZQUIERDA_BASE
addSubview:VISTA_ABAJO_IZQUIERDA_NO_PREGUNTA];
```



```
    }  
    else if (INDICE==2){  
  
        entidad=@"Verbos";  
Label_frase1.hidden=NO;  
Imagen_frase1.image=btnImage3;  
        Label_frase1.text=Label3.text;  
[self ModificarContadores: FILA_A_CAMBIAR ValorAnterior:  
CONTADOR_ANTERIOR Entidad: @"Personas"];  
  
    }  
    else if (INDICE==3){  
  
        entidad=@"Complementos";  
entidad_a_modificar=@"Verbos";  
Label_frase2.hidden=NO;  
        Imagen_frase2.image=btnImage3;  
        Label_frase2.text=Label3.text;  
        if (QUIEROESTOY==1) {  
  
            entidad=@"Parte3";  
                predicado_sublugar=predicado3;  
            }  
        else{  
[self ModificarContadores: FILA_A_CAMBIAR ValorAnterior:  
CONTADOR_ANTERIOR Entidad: entidad_a_modificar];  
            }  
        }  
    else if (INDICE==4){  
  
        entidad_a_modificar=@"Complementos";  
        if (QUIEROESTOY==1){  
  
entidad_a_modificar=@"Parte3";  
  
        }  
        Label_frase3.hidden=NO;  
            Imagen_frase3.image=btnImage3;  
            Label_frase3.text=Label3.text;  
            [VISTA_ARRIBA_BOTON_BASE  
addSubview:VISTA_ARRIBA_BOTON];  
            [self ModificarContadores: FILA_A_CAMBIAR  
ValorAnterior: CONTADOR_ANTERIOR Entidad: entidad_a_modificar];  
            [self DeshabilitarImágenesYEtiquetas];
```

```
    }  
  
    [self BusquedaYSituacionFrase:entidad Predicado:predicado_sublugar];  
  
    break;  
  
case 24:  
    DERECHA=0;  
    [BOTON_PAGINA_IZQUIERDA removeFromSuperview];  
    [BOTON_PAGINA_DERECHA removeFromSuperview];  
    FILA_A_CAMBIAR=PK_4;  
    CONTADOR_ANTERIOR=VALOR_ANTIGUO_CONTADOR_4;  
  
    [self IniciarImágenes];  
  
    if (INDICE==1){  
  
        predicado_sublugar=predicado4;  
        entidad=@"Personas";  
        [VISTA_ABAJO_IZQUIERDA_BASE  
addSubview:VISTA_ABAJO_IZQUIERDA_NO_PREGUNTA];  
  
    }  
    else if (INDICE==2){  
  
        entidad=@"Verbos";  
        Label_frase1.hidden=NO;  
        Imagen_frase1.image=btnImage4;  
        Label_frase1.text=Label4.text;  
        [self ModificarContadores: FILA_A_CAMBIAR ValorAnterior:  
CONTADOR_ANTERIOR Entidad: @"Personas"];  
  
    }  
    else if (INDICE==3){  
  
        entidad=@"Complementos";  
        entidad_a_modificar=@"Verbos";  
        Label_frase2.hidden=NO;  
        Imagen_frase2.image=btnImage4;  
        Label_frase2.text=Label4.text;  
        if (QUIEROESTOY==1) {  
  
            entidad=@"Parte3";  
            predicado_sublugar=predicado4;  
        }  
    }  
}
```

```
else{
    [self ModificarContadores: FILA_A_CAMBIAR ValorAnterior:
CONTADOR_ANTERIOR Entidad: entidad_a_modificar];
    }
}
else if (INDICE==4){

entidad_a_modificar=@"Complementos";
if (QUIEROESTOY==1){
    entidad_a_modificar=@"Parte3";
}
Label_frase3.hidden=NO;
    Imagen_frase3.image=btnImage4;
    Label_frase3.text=Label4.text;
    [VISTA_ARRIBA_BOTON_BASE
addSubview:VISTA_ARRIBA_BOTON];
    [self ModificarContadores: FILA_A_CAMBIAR
ValorAnterior: CONTADOR_ANTERIOR Entidad: entidad_a_modificar];
    [self DeshabilitarImagenesYEtiquetas];

}

[self BusquedaYSituacionFrase:entidad Predicado:predicado_sublugar];

break;
case 25:

    DERECHA=0;
    [BOTON_PAGINA_IZQUIERDA removeFromSuperview];
    [BOTON_PAGINA_DERECHA removeFromSuperview];
    FILA_A_CAMBIAR=PK_5;
    CONTADOR_ANTERIOR=VALOR_ANTIGUO_CONTADOR_5;

[self IniciarImagenes];

    if (INDICE==1){

        predicado_sublugar=predicado5;
        entidad=@"Personas";
        [VISTA_ABAJO_IZQUIERDA_BASE
addSubview:VISTA_ABAJO_IZQUIERDA_NO_PREGUNTA];

    }
    else if (INDICE==2){

        entidad=@"Verbos";
```

```
Label_frase1.hidden=NO;
Imagen_frase1.image=btnImage5;
    Label_frase1.text=Label5.text;
[self ModificarContadores: FILA_A_CAMBIAR ValorAnterior:
CONTADOR_ANTERIOR Entidad: @"Personas"];

    }
    else if (INDICE==3){

        entidad=@"Complementos";
entidad_a_modificar=@"Verbos";
Label_frase2.hidden=NO;
    Imagen_frase2.image=btnImage5;
    Label_frase2.text=Label5.text;
    if (QUIEROESTOY==1) {

        entidad=@"Parte3";
            predicado_sublugar=predicado5;
        }
    else{
        [self ModificarContadores: FILA_A_CAMBIAR ValorAnterior:
CONTADOR_ANTERIOR Entidad: entidad_a_modificar];
        }
    }
    else if (INDICE==4){

entidad_a_modificar=@"Complementos";
if (QUIEROESTOY==1){
    entidad_a_modificar=@"Parte3";
}
Label_frase3.hidden=NO;
    Imagen_frase3.image=btnImage5;
    Label_frase3.text=Label5.text;
    [VISTA_ARRIBA_BOTON_BASE
addSubview:VISTA_ARRIBA_BOTON];
        [self ModificarContadores: FILA_A_CAMBIAR
ValorAnterior: CONTADOR_ANTERIOR Entidad: entidad_a_modificar];
        [self DeshabilitarImagenesYEtiquetas];

    }

    [self BusquedaYSituacionFrase:entidad Predicado:predicado_sublugar];

break;
case 26:
```

```
DERECHA=0;
[BOTON_PAGINA_IZQUIERDA removeFromSuperview];
[BOTON_PAGINA_DERECHA removeFromSuperview];
FILA_A_CAMBIAR=PK_6;
CONTADOR_ANTERIOR=VALOR_ANTIGUO_CONTADOR_6;

[self IniciarImagenes];

if (INDICE==1){

    predicado_sublugar=predicado6;
    entidad=@"Personas";
    [VISTA_ABAJO_IZQUIERDA_BASE
addSubview:VISTA_ABAJO_IZQUIERDA_NO_PREGUNTA];

}
else if (INDICE==2){

    entidad=@"Verbos";
    Label_frase1.hidden=NO;
    Imagen_frase1.image=btnImage6;
    Label_frase1.text=Label6.text;
    [self ModificarContadores: FILA_A_CAMBIAR ValorAnterior:
CONTADOR_ANTERIOR Entidad: @"Personas"];

}
else if (INDICE==3){

    entidad=@"Complementos";
    entidad_a_modificar=@"Verbos";
    Label_frase2.hidden=NO;
    Imagen_frase2.image=btnImage6;
    Label_frase2.text=Label6.text;
    if (QUIEROESTOY==1) {

        entidad=@"Parte3";
        predicado_sublugar=predicado6;

    }
else{

    [self ModificarContadores: FILA_A_CAMBIAR ValorAnterior: CONTADOR_ANTERIOR
Entidad: entidad_a_modificar];

}
}
```

```
    }  
        else if (INDICE==4){  
  
            entidad_a_modificar=@"Complementos";  
            if (QUIEROESTOY==1){  
                entidad_a_modificar=@"Parte3";  
            }  
            Label_frase3.hidden=NO;  
                Imagen_frase3.image=btnImage6;  
                Label_frase3.text=Label6.text;  
                [VISTA_ARRIBA_BOTON_BASE  
addSubview:VISTA_ARRIBA_BOTON];  
                [self ModificarContadores: FILA_A_CAMBIAR  
ValorAnterior: CONTADOR_ANTERIOR Entidad: entidad_a_modificar];  
                [self DeshabilitarImagenesYEtiquetas];  
  
        }  
  
        [self BusquedaYSituacionFrase:entidad Predicado:predicado_sublugar];  
  
        break;  
  
        default:  
            break;  
    }  
}  
  
//-----  
//-----COMUNICADOR-----  
//-----  
  
//Hacemos la búsqueda de los sublugares en función del lugar pulsado. Solo sirve para  
//la primera búsqueda.  
- (void) BusquedaLugares:(NSString *)entidad_búsqueda {  
  
    COMUNICADORAppDelegate *appDelegate = [[UIApplication sharedApplication]  
delegate];  
  
    NSManagedObjectContext *context = [appDelegate managedObjectContext];
```

```
NSEntityDescription *entityDesc = [NSEntityDescription  
entityForName:entidad_busqueda inManagedObjectContext:context];  
  
NSFetchRequest *request = [[[NSFetchRequest alloc] init]autorelease];  
  
[request setEntity:entityDesc];  
  
NSError *error;  
  
NSArray *objects = [context executeFetchRequest:request error:&error];  
  
NSManagedObject *matches = nil;  
  
[self PonerLugares:objects Objeto:matches Numero:0];  
  
}  
  
- (void) BusquedaYSituacion:(NSString *)entidad_busqueda Predicado:  
(NSString*)predicado_busqueda{  
  
[VISTA_MEDIO removeFromSuperview];  
[VISTA_ABAJO_IZQUIERDA removeFromSuperview];  
[VISTA_ABAJO_BORRAR_BASE addSubview:VISTA_ABAJO_BORRAR];  
[VISTA_MEDIO_BASE addSubview:VISTA_MEDIO_SUBLUGARES];  
  
COMUNICADORAppDelegate *appDelegate = [[UIApplication sharedApplication]  
delegate];  
  
NSManagedObjectContext *context = [appDelegate managedObjectContext];  
  
NSEntityDescription *entityDesc = [NSEntityDescription  
entityForName:entidad_busqueda inManagedObjectContext:context];  
  
NSFetchRequest *request = [[NSFetchRequest alloc] init];  
  
[request setEntity:entityDesc];  
  
NSPredicate *pred = [NSPredicate predicateWithFormat:predicado_busqueda];  
  
[request setPredicate:pred];
```

```
    NSSortDescriptor *sortDescriptor = [[[NSSortDescriptor alloc]
initWithKey:@"contador" ascending:NO]autorelease];

    [request setSortDescriptors:[NSArray arrayWithObject:sortDescriptor]];

    NSError *error;

    NSArray *objects = [context executeFetchRequest:request error:&error];

    NSManagedObject *matches = nil;

    [self PonerImágenes:objects Objeto:matches Numero:0];

    [request release];

}

//Hacemos la búsqueda de los elementos de la frase en función del sublugar pulsado.
Segunda búsqueda.

- (void) BusquedaYSituacionFrase:(NSString *)entidad_busqueda Predicado:
(NSString*)predicado_busqueda{

    COMUNICADORAppDelegate *appDelegate = [[UIApplication sharedApplication]
delegate];

    NSManagedObjectContext *context = [appDelegate managedObjectContext];

    NSEntityDescription *entityDesc = [NSEntityDescription
entityForName:entidad_busqueda inManagedObjectContext:context];

    NSFetchRequest *request = [[NSFetchRequest alloc] init];

    [request setEntity:entityDesc];

    NSPredicate *pred = [NSPredicate predicateWithFormat:predicado_busqueda];

    NSError *error;

    [request setPredicate:pred];
```



```
NSSortDescriptor *sortDescriptor = [[[NSSortDescriptor alloc]
initWithKey:@"contador" ascending:NO]autorelease];

[request setSortDescriptors:[NSArray arrayWithObject:sortDescriptor]];

NSArray *objects = [context executeFetchRequest:request error:&error];

NSManagedObject *matches = nil;

[self PonerImágenes:objects Objeto:matches Numero:0];

if ([objects count] > 6) {

    if (INDICE<4) {

        DERECHA=1;
        [VISTA_ABAJO_BOTONES_BASE addSubview:VISTA_ABAJO_BOTONES];
        [VISTA_ABAJO_BOTONES addSubview:BOTON_PAGINA_DERECHA];
        [BOTON_PAGINA_IZQUIERDA removeFromSuperview];
        }

    else if (INDICE==4){
        DERECHA=1;
    }
}

INDICE=INDICE+1;
INDICE_PAGINAS=6;
[request release];
NSLog(@"Hay %i elementos",[objects count]);

}

//Búsqueda hacia atrás, al pulsar Borrar. Parece igual que la anterior pero no inicializa
DERECHA
- (void) BusquedaYSituacionBorrar:(NSString *)entidad_busqueda Predicado:
(NSString*)predicado_busqueda{

    COMUNICADORAppDelegate *appDelegate = [[UIApplication sharedApplication]
delegate];
```

```
NSManagedObjectContext *context = [appDelegate managedObjectContext];

NSEntityDescription *entityDesc = [NSEntityDescription
entityForName:entidad_busqueda inManagedObjectContext:context];

NSFetchRequest *request = [[[NSFetchRequest alloc] init];

[request setEntity:entityDesc];

NSPredicate *pred = [NSPredicate predicateWithFormat:predicado_busqueda];

[request setPredicate:pred];

NSSortDescriptor *sortDescriptor = [[[NSSortDescriptor alloc]
initWithKey:@"contador" ascending:NO]autorelease];

[request setSortDescriptors:[NSArray arrayWithObject:sortDescriptor]];

NSError *error;

NSArray *objects = [context executeFetchRequest:request error:&error];

NSManagedObject *matches = nil;

[self PonerImagenes:objects Objeto:matches Numero:0];

if ([objects count] > 6) {

    if (INDICE<4) {

        DERECHA=1;
        [VISTA_ABAJO_BOTONES_BASE addSubview:VISTA_ABAJO_BOTONES];
        [VISTA_ABAJO_BOTONES addSubview:BOTON_PAGINA_DERECHA];
        [BOTON_PAGINA_IZQUIERDA removeFromSuperview];
    }
}

[request release];

}
```

```
- (void) PonerLugares:(NSArray *)objects Objeto:(NSObject *)matches  
Numero:(int)valor{
```

```
    Idioma=@"nombre";  
    if (controladorAjustes.tipo_letra==20){  
        Idioma=@"nombre";  
    }  
    else if (controladorAjustes.tipo_letra==30){  
        Idioma=@"nombre_may";  
    }  
    else if (controladorAjustes.tipo_letra==50){  
        Idioma=@"nombre_ing";  
    }  
    else if (controladorAjustes.tipo_letra==60){
```

```
        Idioma=@"nombre_may_ing";
```

```
    }
```

```
    if (QE==1) {
```

```
        if ([objects count] == 0) {
```

```
            //Label4.text = @"No hay coincidencias";
```

```
        }
```

```
        else if ([objects count] == 2){
```

```
            matches = [objects objectAtIndex:valor];  
            Etiqueta1=[matches valueForKey:@"nombre"];  
            PK_1=[[matches valueForKey:@"pk"]intValue];  
            VALOR_ANTIGUO_CONTADOR_1=[[matches  
valueForKey:@"contador"]intValue];  
            Label_quiero.text=[matches valueForKey:Idioma];  
            //UIImage *btnImage1;  
            btnImage1 = [UIImage imageNamed:Etiqueta1];  
            [Boton_quiero setImage:btnImage1 forState:UIControlStateNormal];
```

```
            matches = [objects objectAtIndex:valor+1];  
            Etiqueta2=[matches valueForKey:@"nombre"];  
            PK_2=[[matches valueForKey:@"pk"]intValue];  
            VALOR_ANTIGUO_CONTADOR_2=[[matches  
valueForKey:@"contador"]intValue];  
            Label_estoy.text=[matches valueForKey:Idioma];  
            //UIImage *btnImage2;  
            btnImage2 = [UIImage imageNamed:Etiqueta2];
```

```
[Boton_estoy setImage:btnImage2 forState:UIControlStateNormal];  
  
        }  
    }  
else {  
  
    if ([objects count] == 0) {  
  
        //Label4.text = @"No hay coincidencias";  
        }  
  
    else if ([objects count] == 4){  
  
        matches = [objects objectAtIndex:valor];  
        Etiqueta1=[matches valueForKey:@"nombre"];  
        PK_1=[[matches valueForKey:@"pk"]intValue];  
        VALOR_ANTIGUO_CONTADOR_1=[[matches  
valueForKey:@"contador"]intValue];  
        Label_lugar1.text=[matches valueForKey:Idioma];  
        //UIImage *btnImage1;  
  
        btnImage1 = [UIImage imageNamed:Etiqueta1];  
        [Imagen_lugar1 setImage:btnImage1 forState:UIControlStateNormal];  
  
        matches = [objects objectAtIndex:valor+1];  
        Etiqueta2=[matches valueForKey:@"nombre"];  
        PK_2=[[matches valueForKey:@"pk"]intValue];  
        VALOR_ANTIGUO_CONTADOR_2=[[matches  
valueForKey:@"contador"]intValue];  
        Label_lugar2.text=[matches valueForKey:Idioma];  
        //UIImage *btnImage2;  
        btnImage2 = [UIImage imageNamed:Etiqueta2];  
        [Imagen_lugar2 setImage:btnImage2 forState:UIControlStateNormal];  
  
        matches = [objects objectAtIndex:valor+2];  
        Etiqueta3=[matches valueForKey:@"nombre"];  
        PK_3=[[matches valueForKey:@"pk"]intValue];  
        VALOR_ANTIGUO_CONTADOR_3=[[matches  
valueForKey:@"contador"]intValue];  
        Label_lugar3.text=[matches valueForKey:Idioma];  
        //UIImage *btnImage3;  
        btnImage3 = [UIImage imageNamed:Etiqueta3];  
        [Imagen_lugar3 setImage:btnImage3 forState:UIControlStateNormal];  
  
        matches = [objects objectAtIndex:valor+3];
```

```
Etiqueta4=[matches valueForKey:@"nombre"];
PK_4=[[matches valueForKey:@"pk"]intValue];
VALOR_ANTIGUO_CONTADOR_4=[[matches
valueForKey:@"contador"]intValue];
Label_lugar4.text=[matches valueForKey:Idioma];
//UIImage *btnImage4;
btnImage4 = [UIImage imageNamed:Etiqueta4];
[Imagen_lugar4 setImage:btnImage4 forState:UIControlStateNormal];

    }
}
}
```

//Después de una búsqueda, coloca las imágenes en su sitio comprobando si hay más de 6 o no

//Si hay menos, deshabilita el botón. La variable valor, la utilizo para los botones página izquierda

//y página derecha, para que haya un control de los elementos del array que vamos sacando en las diferentes pantallas.

- (void) PonerImágenes:(NSArray *)objects Objeto:(NSObject *)matches
Numero:(int)valor{

```
Label1.hidden=NO;
Label2.hidden=NO;
Label3.hidden=NO;
Label4.hidden=NO;
Label5.hidden=NO;
Label6.hidden=NO;
```

```
if (controladorAjustes.tipo_letra==20){
    Idioma=@"nombre";
}
else if (controladorAjustes.tipo_letra==30){
    Idioma=@"nombre_may";
}
else if (controladorAjustes.tipo_letra==50){
    Idioma=@"nombre_ing";
}
else if (controladorAjustes.tipo_letra==60){
    Idioma=@"nombre_may_ing";
}
```

```
if ([objects count] == 0) {
    //Label4.text = @"No hay coincidencias";
}
```

```
else if ([objects count] == (valor+1)){

    matches = [objects objectAtIndex:valor];
    Etiqueta1=[matches valueForKey:@"nombre"];
    PK_1=[[matches valueForKey:@"pk"]intValue];
    EsNuevo_1=[[matches valueForKey:@"nuevo"]intValue];
    VALOR_ANTIGUO_CONTADOR_1=[[matches
valueForKey:@"contador"]intValue];
    Label1.text=[matches valueForKey:Idioma];
    //UIImage *btnImage1;
    if (EsNuevo_1==1){
        btnImage1 = [controladorPersonalizacion loadImage:Etiqueta1];    }
    else{
        btnImage1 = [UIImage imageNamed:Etiqueta1];

    }
    [Imagen1 setImage:btnImage1 forState:UIControlStateNormal];

    Imagen2.hidden=YES;
    Imagen2.enabled=NO;
    Label2.hidden=YES;
    [Imagen2 setImage:nil forState:UIControlStateNormal];
    Label2.text=nil;

    Imagen3.hidden=YES;
    Imagen3.enabled=NO;
    Label3.hidden=YES;
    [Imagen3 setImage:nil forState:UIControlStateNormal];
    Label3.text=nil;

    Imagen4.hidden=YES;

    Imagen4.enabled=NO;
    [Imagen4 setImage:nil forState:UIControlStateNormal];
    Label4.hidden=YES;
    Label4.text=nil;

    Imagen5.hidden=YES;
    Imagen5.enabled=NO;
    Label5.hidden=YES;
    [Imagen5 setImage:nil forState:UIControlStateNormal];
    Label5.text=nil;

    Imagen6.hidden=YES;
    Imagen6.enabled=NO;
    Label6.hidden=YES;
```

```
[Imagen6 setImage:nil forState:UIControlStateNormal];
Label6.text=nil;

}
else if ([objects count] == (valor+2)){

    matches = [objects objectAtIndex:valor];
    Etiqueta1=[[matches valueForKey:@"nombre"];
    PK_1=[[matches valueForKey:@"pk"]intValue];
    EsNuevo_1=[[matches valueForKey:@"nuevo"]intValue];
    VALOR_ANTIGUO_CONTADOR_1=[[matches
valueForKey:@"contador"]intValue];
    Label1.text=[[matches valueForKey:Idioma];
    //UIImage *btnImage1;
    if (EsNuevo_1==1){
        btnImage1 = [controladorPersonalizacion loadImage:Etiqueta1];    }
    else{
        btnImage1 = [UIImage imageNamed:Etiqueta1];

    }
    [Imagen1 setImage:btnImage1 forState:UIControlStateNormal];

    matches = [objects objectAtIndex:valor+1];
    Etiqueta2=[[matches valueForKey:@"nombre"];
    PK_2=[[matches valueForKey:@"pk"]intValue];
    EsNuevo_2=[[matches valueForKey:@"nuevo"]intValue];
    VALOR_ANTIGUO_CONTADOR_2=[[matches
valueForKey:@"contador"]intValue];
    Label2.text=[[matches valueForKey:Idioma];
    //UIImage *btnImage2;
    if (EsNuevo_2==1){
        btnImage2 = [controladorPersonalizacion loadImage:Etiqueta2];
    }
    else{
        btnImage2 = [UIImage imageNamed:Etiqueta2];

    }
}

[Imagen2 setImage:btnImage2 forState:UIControlStateNormal];

Imagen3.hidden=YES;
Imagen3.enabled=NO;
Label3.hidden=YES;
[Imagen3 setImage:nil forState:UIControlStateNormal];
Label3.text=nil;

Imagen4.hidden=YES;
```

```
Imagen4.enabled=NO;
Label4.hidden=YES;
[Imagen4 setImage:nil forState:UIControlStateNormal];
Label4.text=nil;

Imagen5.hidden=YES;
Imagen5.enabled=NO;
Label5.hidden=YES;
[Imagen5 setImage:nil forState:UIControlStateNormal];
Label5.text=nil;

Imagen6.hidden=YES;
Imagen6.enabled=NO;
Label6.hidden=YES;
[Imagen6 setImage:nil forState:UIControlStateNormal];
Label6.text=nil;

}
else if ([objects count] == (valor+3)){

    matches = [objects objectAtIndex:valor];
    Etiqueta1=[matches valueForKey:@"nombre"];
    PK_1=[[matches valueForKey:@"pk"]intValue];
    EsNuevo_1=[[matches valueForKey:@"nuevo"]intValue];
    VALOR_ANTIGUO_CONTADOR_1=[[matches
valueForKey:@"contador"]intValue];
    Label1.text=[matches valueForKey:Idioma];
    //UIImage *btnImage1;
    if (EsNuevo_1==1){
        btnImage1 = [controladorPersonalizacion loadImage:Etiqueta1];    }
    else{
        btnImage1 = [UIImage imageNamed:Etiqueta1];

    }
    [Imagen1 setImage:btnImage1 forState:UIControlStateNormal];

    matches = [objects objectAtIndex:valor+1];
    Etiqueta2=[matches valueForKey:@"nombre"];
    PK_2=[[matches valueForKey:@"pk"]intValue];

    EsNuevo_2=[[matches valueForKey:@"nuevo"]intValue];
    VALOR_ANTIGUO_CONTADOR_2=[[matches
valueForKey:@"contador"]intValue];
    Label2.text=[matches valueForKey:Idioma];
    //UIImage *btnImage2;
    if (EsNuevo_2==1){
        btnImage2 = [controladorPersonalizacion loadImage:Etiqueta2];
```



```
}  
else{  
    btnImage2 = [UIImage imageNamed:Etiqueta2];  
  
}  
[Imagen2 setImage:btnImage2 forState:UIControlStateNormal];  
  
matches = [objects objectAtIndex:valor+2];  
Etiqueta3=[matches valueForKey:@"nombre"];  
PK_3=[[matches valueForKey:@"pk"]intValue];  
EsNuevo_3=[[matches valueForKey:@"nuevo"]intValue];  
VALOR_ANTIGUO_CONTADOR_3=[[matches  
valueForKey:@"contador"]intValue];  
Label3.text=[matches valueForKey:Idioma];  
//UIImage *btnImage3;  
if (EsNuevo_3==1){  
    btnImage3 = [controladorPersonalizacion loadImage:Etiqueta3];  
}  
else{  
    btnImage3 = [UIImage imageNamed:Etiqueta3];  
  
}  
[Imagen3 setImage:btnImage3 forState:UIControlStateNormal];  
  
Imagen4.hidden=YES;  
Imagen4.enabled=NO;  
Label4.hidden=YES;  
[Imagen4 setImage:nil forState:UIControlStateNormal];  
Label4.text=nil;  
  
Imagen5.hidden=YES;  
Imagen5.enabled=NO;  
Label5.hidden=YES;  
[Imagen5 setImage:nil forState:UIControlStateNormal];  
Label5.text=nil;  
  
Imagen6.hidden=YES;  
Imagen6.enabled=NO;  
Label6.hidden=YES;  
[Imagen6 setImage:nil forState:UIControlStateNormal];  
Label6.text=nil;  
  
}  
else if ([objects count] == (valor+4)){
```

```
matches = [objects objectAtIndex:valor];
Etiqueta1=[matches valueForKey:@"nombre"];
PK_1=[[matches valueForKey:@"pk"]intValue];
EsNuevo_1=[[matches valueForKey:@"nuevo"]intValue];
VALOR_ANTIGUO_CONTADOR_1=[[matches
valueForKey:@"contador"]intValue];
Label1.text=[matches valueForKey:Idioma];
//UIImage *btnImage1;
if (EsNuevo_1==1){
    btnImage1 = [controladorPersonalizacion loadImage:Etiqueta1];    }
else{
    btnImage1 = [UIImage imageNamed:Etiqueta1];
}
[Imagen1 setImage:btnImage1 forState:UIControlStateNormal];

matches = [objects objectAtIndex:valor+1];
Etiqueta2=[matches valueForKey:@"nombre"];
PK_2=[[matches valueForKey:@"pk"]intValue];
EsNuevo_2=[[matches valueForKey:@"nuevo"]intValue];
VALOR_ANTIGUO_CONTADOR_2=[[matches
valueForKey:@"contador"]intValue];
Label2.text=[matches valueForKey:Idioma];
//UIImage *btnImage2;
if (EsNuevo_2==1){
    btnImage2 = [controladorPersonalizacion loadImage:Etiqueta2];
}
else{
    btnImage2 = [UIImage imageNamed:Etiqueta2];
}
[Imagen2 setImage:btnImage2 forState:UIControlStateNormal];

matches = [objects objectAtIndex:valor+2];
Etiqueta3=[matches valueForKey:@"nombre"];
PK_3=[[matches valueForKey:@"pk"]intValue];
EsNuevo_3=[[matches valueForKey:@"nuevo"]intValue];
VALOR_ANTIGUO_CONTADOR_3=[[matches
valueForKey:@"contador"]intValue];
Label3.text=[matches valueForKey:Idioma];
//UIImage *btnImage3;
if (EsNuevo_3==1){
    btnImage3 = [controladorPersonalizacion loadImage:Etiqueta3];
}
else{
    btnImage3 = [UIImage imageNamed:Etiqueta3];
}
```

```
}  
[Imagen3 setImage:btnImage3 forState:UIControlStateNormal];  
  
matches = [objects objectAtIndex:valor+3];  
Etiqueta4=[matches valueForKey:@"nombre"];  
PK_4=[[matches valueForKey:@"pk"]intValue];  
EsNuevo_4=[[matches valueForKey:@"nuevo"]intValue];  
VALOR_ANTIGUO_CONTADOR_4=[[matches  
valueForKey:@"contador"]intValue];  
Label4.text=[matches valueForKey:Idioma];  
//UIImage *btnImage4;  
if (EsNuevo_4==1){  
    btnImage4 = [controladorPersonalizacion loadImage:Etiqueta4];  
}  
else{  
    btnImage4 = [UIImage imageNamed:Etiqueta4];  
}  
[Imagen4 setImage:btnImage4 forState:UIControlStateNormal];  
  
Imagen5.hidden=YES;  
Imagen5.enabled=NO;  
Label5.hidden=YES;  
[Imagen5 setImage:nil forState:UIControlStateNormal];  
Label5.text=nil;  
  
Imagen6.hidden=YES;  
Imagen6.enabled=NO;  
Label6.hidden=YES;  
[Imagen6 setImage:nil forState:UIControlStateNormal];  
Label6.text=nil;  
  
}  
else if ([objects count] == (valor+5)){  
  
    matches = [objects objectAtIndex:valor];  
    Etiqueta1=[matches valueForKey:@"nombre"];  
    PK_1=[[matches valueForKey:@"pk"]intValue];  
    EsNuevo_1=[[matches valueForKey:@"nuevo"]intValue];  
    VALOR_ANTIGUO_CONTADOR_1=[[matches  
valueForKey:@"contador"]intValue];  
    Label1.text=[matches valueForKey:Idioma];  
    //UIImage *btnImage1;  
    if (EsNuevo_1==1){
```

```
        btnImage1 = [controladorPersonalizacion loadImage:Etiqueta1];    }
    else{
        btnImage1 = [UIImage imageNamed:Etiqueta1];
    }

    [Imagen1 setImage:btnImage1 forState:UIControlStateNormal];

    matches = [objects objectAtIndex:valor+1];
    Etiqueta2=[matches valueForKey:@"nombre"];
    PK_2=[[matches valueForKey:@"pk"]intValue];
    EsNuevo_2=[[matches valueForKey:@"nuevo"]intValue];
    VALOR_ANTIGUO_CONTADOR_2=[[matches
valueForKey:@"contador"]intValue];
    Label2.text=[matches valueForKey:Idioma];
    //UIImage *btnImage2;
    if (EsNuevo_2==1){
        btnImage2 = [controladorPersonalizacion loadImage:Etiqueta2];
    }
    else{
        btnImage2 = [UIImage imageNamed:Etiqueta2];
    }
    [Imagen2 setImage:btnImage2 forState:UIControlStateNormal];

    matches = [objects objectAtIndex:valor+2];
    Etiqueta3=[matches valueForKey:@"nombre"];
    PK_3=[[matches valueForKey:@"pk"]intValue];
    EsNuevo_3=[[matches valueForKey:@"nuevo"]intValue];
    VALOR_ANTIGUO_CONTADOR_3=[[matches
valueForKey:@"contador"]intValue];
    Label3.text=[matches valueForKey:Idioma];
    //UIImage *btnImage3;
    if (EsNuevo_3==1){
        btnImage3 = [controladorPersonalizacion loadImage:Etiqueta3];
    }
    else{
        btnImage3 = [UIImage imageNamed:Etiqueta3];
    }
    [Imagen3 setImage:btnImage3 forState:UIControlStateNormal];

    matches = [objects objectAtIndex:valor+3];
    Etiqueta4=[matches valueForKey:@"nombre"];
    PK_4=[[matches valueForKey:@"pk"]intValue];
    EsNuevo_4=[[matches valueForKey:@"nuevo"]intValue];
```

```

    VALOR_ANTIGUO_CONTADOR_4=[[matches
valueForKey:@"contador"]intValue];
    Label4.text=[matches valueForKey:Idioma];
    //UIImage *btnImage4;
    if (EsNuevo_4==1){
        btnImage4 = [controladorPersonalizacion loadImage:Etiqueta4];
    }
    else{
        btnImage4 = [UIImage imageNamed:Etiqueta4];
    }
    [Imagen4 setImage:btnImage4 forState:UIControlStateNormal];

    matches = [objects objectAtIndex:valor+4];
    Etiqueta5=[matches valueForKey:@"nombre"];
    PK_5=[[matches valueForKey:@"pk"]intValue];
    EsNuevo_5=[[matches valueForKey:@"nuevo"]intValue];
    VALOR_ANTIGUO_CONTADOR_5=[[matches
valueForKey:@"contador"]intValue];
    Label5.text=[matches valueForKey:Idioma];
    //UIImage *btnImage5;
    if (EsNuevo_5==1){
        btnImage5 = [controladorPersonalizacion loadImage:Etiqueta5];
    }
    else{
        btnImage5 = [UIImage imageNamed:Etiqueta5];
    }
    [Imagen5 setImage:btnImage5 forState:UIControlStateNormal];

    Imagen6.hidden=YES;
    Imagen6.enabled=NO;
    Label6.hidden=YES;
    [Imagen6 setImage:nil forState:UIControlStateNormal];
    Label6.text=nil;
}
else if ([objects count] >= (valor+6)){

    matches = [objects objectAtIndex:valor];
    Etiqueta1=[matches valueForKey:@"nombre"];
    PK_1=[[matches valueForKey:@"pk"]intValue];
    EsNuevo_1=[[matches valueForKey:@"nuevo"]intValue];
    VALOR_ANTIGUO_CONTADOR_1=[[matches
valueForKey:@"contador"]intValue];
    Label1.text=[matches valueForKey:Idioma];
    //UIImage *btnImage1;

```

```
if (EsNuevo_1==1){
    btnImage1 = [controladorPersonalizacion loadImage:Etiqueta1];    }
else{
    btnImage1 = [UIImage imageNamed:Etiqueta1];
}
[Imagen1 setImage:btnImage1 forState:UIControlStateNormal];

matches = [objects objectAtIndex:valor+1];
Etiqueta2=[matches valueForKey:@"nombre"];
PK_2=[[matches valueForKey:@"pk"]intValue];
EsNuevo_2=[[matches valueForKey:@"nuevo"]intValue];
VALOR_ANTIGUO_CONTADOR_2=[[matches
valueForKey:@"contador"]intValue];
Label2.text=[matches valueForKey:Idioma];

//UIImage *btnImage2;
if (EsNuevo_2==1){
    btnImage2 = [controladorPersonalizacion loadImage:Etiqueta2];
}
else{
    btnImage2 = [UIImage imageNamed:Etiqueta2];
}
[Imagen2 setImage:btnImage2 forState:UIControlStateNormal];

matches = [objects objectAtIndex:valor+2];
Etiqueta3=[matches valueForKey:@"nombre"];
PK_3=[[matches valueForKey:@"pk"]intValue];
EsNuevo_3=[[matches valueForKey:@"nuevo"]intValue];
VALOR_ANTIGUO_CONTADOR_3=[[matches
valueForKey:@"contador"]intValue];
Label3.text=[matches valueForKey:Idioma];
//UIImage *btnImage3;
if (EsNuevo_3==1){
    btnImage3 = [controladorPersonalizacion loadImage:Etiqueta3];
}
else{
    btnImage3 = [UIImage imageNamed:Etiqueta3];
}
[Imagen3 setImage:btnImage3 forState:UIControlStateNormal];

matches = [objects objectAtIndex:valor+3];
Etiqueta4=[matches valueForKey:@"nombre"];
PK_4=[[matches valueForKey:@"pk"]intValue];
EsNuevo_4=[[matches valueForKey:@"nuevo"]intValue];
```

```
VALOR_ANTIGUO_CONTADOR_4=[[matches
valueForKey:@"contador"]intValue];
Label4.text=[matches valueForKey:Idioma];
//UIImage *btnImage4;
if (EsNuevo_4==1){
    btnImage4 = [controladorPersonalizacion loadImage:Etiqueta4];
}
else{
    btnImage4 = [UIImage imageNamed:Etiqueta4];
}
[Imagen4 setImage:btnImage4 forState:UIControlStateNormal];

matches = [objects objectAtIndex:valor+4];
Etiqueta5=[matches valueForKey:@"nombre"];
PK_5=[[matches valueForKey:@"pk"]intValue];
EsNuevo_5=[[matches valueForKey:@"nuevo"]intValue];
VALOR_ANTIGUO_CONTADOR_5=[[matches
valueForKey:@"contador"]intValue];
Label5.text=[matches valueForKey:Idioma];

//UIImage *btnImage5;
if (EsNuevo_5==1){
    btnImage5 = [controladorPersonalizacion loadImage:Etiqueta5];
}
else{
    btnImage5 = [UIImage imageNamed:Etiqueta5];
}
[Imagen5 setImage:btnImage5 forState:UIControlStateNormal];

matches = [objects objectAtIndex:valor+5];
Etiqueta6=[matches valueForKey:@"nombre"];
PK_6=[[matches valueForKey:@"pk"]intValue];
EsNuevo_6=[[matches valueForKey:@"nuevo"]intValue];
VALOR_ANTIGUO_CONTADOR_6=[[matches
valueForKey:@"contador"]intValue];
Label6.text=[matches valueForKey:Idioma];
//UIImage *btnImage6;
if (EsNuevo_6==1){
    btnImage6 = [controladorPersonalizacion loadImage:Etiqueta6];
}
else{
    btnImage6 = [UIImage imageNamed:Etiqueta6];
}
[Imagen6 setImage:btnImage6 forState:UIControlStateNormal];
```

```
    }  
}  
  
//Selecciona la fila que quiere cambiar, que será la primary key del objeto pulsado  
anteriormente, y  
//aumenta en una unidad el valor del contador de ese elemento pulsado para que  
luego aparezca en  
//orden de elementos más utilizados.  
- (void) ModificarContadores: (int) fila_a_cambiar ValorAnterior: (int)  
valor_contador_anterior Entidad: (NSString *) entidad_modificar{  
  
    COMUNICADORAppDelegate *appDelegate = [[UIApplication sharedApplication]  
delegate];  
    NSManagedObjectContext *context = [appDelegate managedObjectContext];  
  
    NSEntityDescription *entityDesc = [NSEntityDescription  
entityForName:entidad_modificar inManagedObjectContext:context];  
    NSFetchRequest *request = [[NSFetchRequest alloc] init];  
  
    [request setEntity:entityDesc];  
  
    NSError *error;  
  
    NSArray *items=[context executeFetchRequest:request error:&error];  
    [request release];  
  
    NSManagedObject *micontador= [items objectAtIndex:(fila_a_cambiar-1)];  
  
    [micontador setValue:[NSNumber numberWithInt: (valor_contador_anterior+1)]  
forKey:@"contador"];  
  
    if (![context save:&error]) {  
        NSLog(@"error de coredata %@, %@", error, [error userInfo]);  
        exit(-1);  
    }  
  
    NSLog(@"LLego");  
  
}
```


- (void) IniciarImagenes{

```
Imagen1.enabled=YES;  
Imagen2.enabled=YES;  
Imagen3.enabled=YES;  
Imagen4.enabled=YES;  
Imagen5.enabled=YES;  
Imagen6.enabled=YES;  
Imagen1.hidden=NO;  
Imagen2.hidden=NO;  
Imagen3.hidden=NO;  
Imagen4.hidden=NO;  
Imagen5.hidden=NO;  
Imagen6.hidden=NO;
```

}

- (void) DeshabilitarImagenesYEtiquetas{

```
Imagen1.enabled=NO;  
Imagen2.enabled=NO;  
Imagen3.enabled=NO;  
Imagen4.enabled=NO;  
Imagen5.enabled=NO;  
Imagen6.enabled=NO;  
Label1.enabled=NO;  
Label2.enabled=NO;
```

```
Label3.enabled=NO;  
Label4.enabled=NO;  
Label5.enabled=NO;  
Label6.enabled=NO;
```

}

- (void) HabilitarImagenes {

```
Imagen1.enabled=YES;  
Imagen2.enabled=YES;  
Imagen3.enabled=YES;  
Imagen4.enabled=YES;  
Imagen5.enabled=YES;  
Imagen6.enabled=YES;
```

}

- (void) HabilitarEtiquetas {

```
Label1.enabled=YES;  
Label2.enabled=YES;
```

```
Label3.enabled=YES;
Label4.enabled=YES;
Label5.enabled=YES;
Label6.enabled=YES;

}

-(IBAction) PULSAR_VOLVER_COMUNICADOR: (id) sender
{
    [VISTA_PRINCIPAL removeFromSuperview];
    QE=0;
}

- (IBAction) PULSAR_BOTON_ARRIBA_OK: (id) sender{

    [self.view addSubview:VISTA_SECUNDARIA];

    if (controladorAjustes.resp_negativa==0){
        [VISTA_BASE_BOTON_NO addSubview: VISTA_BOTON_NO];
    }
    else if (controladorAjustes.resp_negativa==1){
        [VISTA_BOTON_NO removeFromSuperview];
    }

    Imagen_secundaria1.image=Imagen_frase1.image;
    Imagen_secundaria2.image=Imagen_frase2.image;
    Imagen_secundaria3.image=Imagen_frase3.image;
    Label_secundaria1.text=Label_frase1.text;
    Label_secundaria2.text=Label_frase2.text;
    Label_secundaria3.text=Label_frase3.text;
    Imagen_frase1.image=nil;
    Imagen_frase2.image=nil;
    Imagen_frase3.image=nil;
    Label_frase1.text=nil;
    Label_frase2.text=nil;
    Label_frase3.text=nil;
    if (QUIEROESTOY==2){
        Label_secundaria1.hidden=YES;
    }
    else{
        Label_secundaria1.hidden=NO;
    }
}

}
```

//Cuando hay posibilidad de dos paginas, este es el boton de pasar pagina a la derecha

```
-(IBAction) PULSAR_BOTON_PAGINA_DERECHA: (id) sender
{
    /* PASAR A MAYUSCULAS
    NSString *caca;
    caca=[Label4.text uppercaseString];
    Label4.text=caca;
    */
    if (DERECHA>0) {

        [BOTON_PAGINA_DERECHA removeFromSuperview];
        [VISTA_ABAJO_BOTONES addSubview:BOTON_PAGINA_IZQUIERDA];
    }

    [self BusquedaBotonesPaginas];

    COMUNICADORAppDelegate *appDelegate = [[UIApplication sharedApplication]
delegate];

    NSManagedObjectContext *context = [appDelegate managedObjectContext];

    NSEntityDescription *entityDesc = [NSEntityDescription entityForName:entidad
inManagedObjectContext:context];

    NSFetchRequest *request = [[NSFetchRequest alloc] init];

    [request setEntity:entityDesc];

    NSPredicate *pred = [NSPredicate predicateWithFormat:predicado_sublugar];

    [request setPredicate:pred];

    NSSortDescriptor *sortDescriptor = [[[NSSortDescriptor alloc]
initWithKey:@"contador" ascending:NO]autorelease];

    [request setSortDescriptors:[NSArray arrayWithObject:sortDescriptor]];

    NSError *error;

    NSArray *objects = [context executeFetchRequest:request error:&error];
}
```

```
    NSMutableArray *matches = nil;

    [self PonerImágenes:objects Objeto:matches Numero:INDICE_PAGINAS];

    if ([objects count] > INDICE_PAGINAS+6) {

        [VISTA_ABAJO_BOTONES addSubview:BOTON_PAGINA_DERECHA];

    }

    INDICE_PAGINAS=INDICE_PAGINAS+6;

}

//Cuando hay posibilidad de dos paginas, este es el boton de pasar pagina a la
izquierda

-(IBAction) PULSAR_BOTON_PAGINA_IZQUIERDA: (id) sender
{
    INDICE_PAGINAS=INDICE_PAGINAS-6;

    if (DERECHA>0) {

        [BOTON_PAGINA_IZQUIERDA removeFromSuperview];
        [VISTA_ABAJO_BOTONES addSubview:BOTON_PAGINA_DERECHA];
    }

    [self BusquedaBotonesPaginas];

    COMUNICADORAppDelegate *appDelegate = [[UIApplication sharedApplication]
delegate];

    NSMutableArray *context = [appDelegate managedObjectContext];

    NSEntityDescription *entityDesc = [NSEntityDescription entityForName:entidad
inManagedObjectContext:context];

    NSFetchRequest *request = [[NSFetchRequest alloc] init];

    [request setEntity:entityDesc];

    NSPredicate *pred = [NSPredicate predicateWithFormat:predicado_sublugar];
```

```
[request setPredicate:pred];

NSSortDescriptor *sortDescriptor = [[[NSSortDescriptor alloc]
initWithKey:@"contador" ascending:NO]autorelease];

[request setSortDescriptors:[NSArray arrayWithObject:sortDescriptor]];

NSError *error;

NSArray *objects = [context executeFetchRequest:request error:&error];

NSManagedObject *matches = nil;

[self PonerImágenes:objects Objeto:matches Numero:(INDICE_PAGINAS-6)];

    if (INDICE_PAGINAS > 6) {

        [VISTA_ABAJO_BOTONES
addSubview:BOTON_PAGINA_IZQUIERDA];

    }

}

-(void) BusquedaBotonesPaginas{

    [self IniciarImágenes];

    if (INDICE==2) {
        entidad=@"Personas";
    }

    else if (INDICE==3){

        if (QUIEROESTOY>0){

            entidad=@"Parte2";

        }
        else{
            entidad=@"Verbos";
        }

    }

    else if (INDICE==4){
```

```
if (QUIEROESTOY>0){
    entidad=@"Parte3";
}
else{
    entidad=@"Complementos";
}

}

//Botones negar e interrogante

- (IBAction) PULSAR_BOTON_NEGAR: (id) sender{
    if (NEGAR==0) {

        negacion=[CALayer layer];
        //negacion=Imagen_frase2.layer;
        negacion.bounds=CGRectMake(405.0f, 10.0f, 90.0f, 90.0f);
        negacion.position=CGPointMake(0.0f, 0.0f);
        negacion.anchorPoint=CGPointMake(0.0f, 0.0f);
        negacion.contents = (id)[[UIImage imageNamed:@"negacion.gif"] CGImage];

        if (QUIEROESTOY==1) {
            [Imagen_frase1.layer addSublayer:negacion];
        }
        else {
            [Imagen_frase2.layer addSublayer:negacion];
        }

        negacion_secundaria=[CALayer layer];
        //negacion=Imagen_frase2.layer;
        negacion_secundaria.bounds=CGRectMake(360.0f, 279.0f, 200.0f, 200.0f);
        negacion_secundaria.position=CGPointMake(0.0f, 0.0f);
        negacion_secundaria.anchorPoint=CGPointMake(0.0f, 0.0f);
        negacion_secundaria.contents = (id)[[UIImage imageNamed:@"negacion.gif"]
CGImage];

        if (QUIEROESTOY==1) {
            [Imagen_secundaria1.layer addSublayer:negacion_secundaria];
        }

        else
        {
            [Imagen_secundaria2.layer addSublayer:negacion_secundaria];
        }
    }
}
```

```
[BOTON_NEGAR setTitle:@"SI" forState:0];
NEGAR=1;
}
else if (NEGAR==1){

    [negacion removeFromSuperlayer];
    [negacion_secundaria removeFromSuperlayer];
    [BOTON_NEGAR setTitle:@"NO" forState:0];
    NEGAR=0;
}
}

- (IBAction) PULSAR_BOTON_INTERROGANTE: (id) sender{
    if (INTERROGANTE==0) {

        Imagen_frase_interrogante.image=[UIImage imageNamed:
@"interrogante_blanco.png"];
        Imagen_secundaria_interrogante.image=[UIImage imageNamed:
@"interrogante.png"];
        INTERROGANTE=1;
    }
    else if (INTERROGANTE==1){
        Imagen_frase_interrogante.image=nil;
        Imagen_secundaria_interrogante.image=nil;
        INTERROGANTE=0;
    }
}

//Boton oculto esquina superior izquierda de ajustes

- (IBAction) PULSAR_BOTON_AJUSTES: (id) sender
{
    [VISTA_PRIMERA addSubview:controladorAjustes.view];
}

//Niega la frase del usuario

- (IBAction) PULSAR_BOTON_NO: (id) sender
{
    [self.view addSubview:VISTA_NO];
}
```

//Sale de la pantalla NO

```
-(IBAction) PULSAR_BOTON_SALIR_NO: (id) sender
{
    [VISTA_NO removeFromSuperview];
}
```

//Boton salir de la pantalla de la frase final

```
-(IBAction) PULSAR_BOTON_SALIR: (id) sender
{
    if (NEGAR==1){
        [negacion removeFromSuperlayer];
        [negacion_secundaria removeFromSuperlayer];
        [BOTON_NEGAR setTitle:@"NO" forState:0];
        NEGAR=0;
    }

    [VISTA_SECUNDARIA removeFromSuperview];
    [VISTA_ARRIBA_BOTON removeFromSuperview];
    [VISTA_MEDIO_SUBLUGARES removeFromSuperview];
    if (QUIEROESTOY>0) {
        [VISTA_MEDIO_BASE addSubview:VISTA_MEDIO QUIERO_ESTOY];
        [self BusquedaLugares:@"Parte1"];
    }
    else{
        [self BusquedaLugares:@"Lugar"];
    }

    [VISTA_MEDIO_BASE addSubview:VISTA_MEDIO];
    [VISTA_ABAJO_BORRAR removeFromSuperview];
    [VISTA_ABAJO_IZQUIERDA_BASE addSubview:VISTA_ABAJO_IZQUIERDA];
    [VISTA_ABAJO_IZQUIERDA_NO_PREGUNTA removeFromSuperview];

    QUIEROESTOY=0;
    INDICE=0;
    DERECHA=0;
    INTERROGANTE=0;
    INDICE_PAGINAS=6;

    [self IniciarImagenes];

    [self HabilitarEtiquetas];
}
```



```
    Imagen_secundaria1.image=nil;  
    Imagen_secundaria2.image=nil;  
    Imagen_secundaria3.image=nil;  
    Label_secundaria1.text=nil;  
    Label_secundaria2.text=nil;  
    Label_secundaria3.text=nil;  
  
    Imagen_frase_interrogante.image=nil;  
    Imagen_secundaria_interrogante.image=nil;  
    Label_frase1.hidden=YES;  
    Label_frase2.hidden=YES;  
    Label_frase3.hidden=YES;  
}
```

//Pulsar los botones de los pictos

//Pulsar BORRAR para deshacer frases

```
-(IBAction) PULSAR_BOTON_BORRAR: (id) sender  
{  
    [self IniciarImagenes];  
  
    if (INDICE==1) {  
        [VISTA_MEDIO_SUBLUGARES removeFromSuperview];  
        [self BusquedaLugares:@"Lugar"];  
        [VISTA_MEDIO_BASE addSubview:VISTA_MEDIO];  
        INDICE=INDICE-1;  
        [VISTA_ABAJO_BORRAR removeFromSuperview];  
        [VISTA_ABAJO_IZQUIERDA_BASE addSubview:VISTA_ABAJO_IZQUIERDA];  
    }  
    else if (INDICE==2){  
  
        INDICE=INDICE-1;  
        if (DERECHA==1) {  
  
            [BOTON_PAGINA_IZQUIERDA removeFromSuperview];  
            [BOTON_PAGINA_DERECHA removeFromSuperview];  
            DERECHA=0;  
            INDICE_PAGINAS=6;  
  
        }  
        INTERROGANTE=0;  
        if (NEGAR==1){  
  
            [negacion removeFromSuperlayer];  

```

```
[negacion_secundaria removeFromSuperlayer];
[BOTON_NEGAR setTitle:@"NO" forState:0];

NEGAR=0;
}
Imagen_frase_interrogante.image=nil;
Imagen_secundaria_interrogante.image=nil;
[VISTA_ABAJO_IZQUIERDA_NO_PREGUNTA removeFromSuperview];

[self BusquedaYSituacionBorrar:@"Sublugar" Predicado:predicado_base];
}
else if (INDICE==3){

Label_frase1.hidden=YES;
Imagen_frase1.image=nil;
Label_frase1.text=nil;
INDICE=INDICE-1;
if (QUIEROESTOY==1) {
entidad=@"Parte1";
[VISTA_MEDIO_SUBLUGARES removeFromSuperview];
[VISTA_ABAJO_BORRAR removeFromSuperview];
[VISTA_ABAJO_IZQUIERDA_BASE
addSubview:VISTA_ABAJO_IZQUIERDA];
if (NEGAR==1){

[negacion removeFromSuperlayer];
[negacion_secundaria removeFromSuperlayer];
[BOTON_NEGAR setTitle:@"NO" forState:0];
NEGAR=0;

}
Imagen_frase_interrogante.image=nil;
Imagen_secundaria_interrogante.image=nil;
INTERROGANTE=0;

}
else {
if (DERECHA==1) {

[BOTON_PAGINA_IZQUIERDA removeFromSuperview];
[BOTON_PAGINA_DERECHA removeFromSuperview];
DERECHA=0;
INDICE_PAGINAS=6;

}
entidad=@"Personas";
[self BusquedaYSituacionBorrar:entidad Predicado:predicado_sublugar];

}
```

```
}  
else if (INDICE==4){  
  
    Label_frase2.hidden=YES;  
    Imagen_frase2.image=nil;  
    Label_frase2.text=nil;  
    INDICE=INDICE-1;  
    if (QUIEROESTOY==2) {  
        [VISTA_MEDIO_SUBLUGARES removeFromSuperview];  
        [VISTA_ABAJO_BORRAR removeFromSuperview];  
        [VISTA_ABAJO_IZQUIERDA_BASE  
addSubview:VISTA_ABAJO_IZQUIERDA];  
        if (NEGAR==1){  
  
            [negacion removeFromSuperlayer];  
            [negacion_secundaria removeFromSuperlayer];  
            [BOTON_NEGAR setTitle:@"NO" forState:0];  
            NEGAR=0;  
        }  
        Imagen_frase_interrogante.image=nil;  
        Imagen_secundaria_interrogante.image=nil;  
        INTERROGANTE=0;  
        entidad=@"Parte2";  
        predicado_sublugar=@"ANY Parte2Con1 like '2' AND activo=1";  
    }  
    else {  
  
        if (DERECHA==1) {  
  
            [BOTON_PAGINA_IZQUIERDA removeFromSuperview];  
            [BOTON_PAGINA_DERECHA removeFromSuperview];  
            DERECHA=0;  
            INDICE_PAGINAS=6;  
        }  
  
        if (QUIEROESTOY==1) {  
  
            entidad=@"Parte2";  
            predicado_sublugar=@"ANY Parte2Con1 like '1' AND activo=1";  
        }  
        else if (QUIEROESTOY==0){  
  
            entidad=@"Verbos";  
        }  
    }  
}
```

```
[self BusquedaYSituacionBorrar:entidad Predicado:predicado_sublugar];  
  
}  
}  
  
else if (INDICE==5){  
  
[self IniciarImagenes];  
[self HabilitarEtiquetas];  
  
Label_frase3.hidden=YES;  
Imagen_frase3.image=nil;  
Label_frase3.text=nil;  
INDICE=INDICE-1;  
INDICE_PAGINAS=6;  
[VISTA_ARRIBA_BOTON removeFromSuperview];  
if (DERECHA==1){  
    [VISTA_ABAJO_BOTONES addSubview:BOTON_PAGINA_DERECHA];  
}  
}  
}
```

```
- (void)viewDidUnload {  
  
    Imagen1.imageView.image = nil;  
    Imagen2.imageView.image = nil;  
    Imagen3.imageView.image = nil;  
    Imagen4.imageView.image = nil;  
    Imagen5.imageView.image = nil;  
    Imagen6.imageView.image = nil;  
  
    Imagen_lugar1.imageView.image = nil;  
    Imagen_lugar2.imageView.image = nil;  
    Imagen_lugar3.imageView.image = nil;  
    Imagen_lugar4.imageView.image = nil;  
  
    Boton_estoy.imageView.image = nil;  
    Boton_quiero.imageView.image = nil;  
  
    Label_lugar1.text = nil;  
    Label_lugar2.text = nil;
```

```
Label_lugar3.text = nil;  
Label_lugar4.text = nil;
```

```
Label1.text = nil;  
Label2.text = nil;  
Label3.text = nil;  
Label4.text = nil;  
Label5.text = nil;  
Label6.text = nil;
```

```
Label_quiero.text = nil;  
Label_estoy.text = nil;  
Label_frase1.text = nil;  
Label_frase2.text = nil;  
Label_frase3.text = nil;  
Imagen_frase1.image=nil;  
Imagen_frase2.image=nil;  
Imagen_frase3.image=nil;  
Label_secundaria1.text = nil;  
Label_secundaria2.text = nil;  
Label_secundaria3.text = nil;  
Imagen_secundaria1.image=nil;  
Imagen_secundaria2.image=nil;  
Imagen_secundaria3.image=nil;
```

```
[super viewDidUnload];  
// Release any retained subviews of the main view.  
// e.g. self.myOutlet = nil;  
}
```

```
- (void)dealloc {
```

```
[Imagen1 release];  
[Imagen2 release];  
[Imagen3 release];  
[Imagen4 release];
```

```
[Imagen5 release];  
[Imagen6 release];  
[Imagen_lugar1 release];  
[Imagen_lugar2 release];  
[Imagen_lugar3 release];  
[Imagen_lugar4 release];
```

```
[Label_lugar1 release];  
[Label_lugar2 release];
```

```
[Label_lugar3 release];  
[Label_lugar4 release];
```

```
[Label1 release];  
[Label2 release];  
[Label3 release];  
[Label4 release];  
[Label5 release];  
[Label6 release];  
[Label_quiero release];  
[Label_estoy release];
```

```
[Boton_quiero release];  
[Boton_estoy release];  
[Label_frase1 release];  
[Label_frase2 release];  
[Label_frase3 release];  
[Imagen_frase1 release];  
[Imagen_frase2 release];  
[Imagen_frase3 release];  
[Label_secundaria1 release];  
[Label_secundaria2 release];  
[Label_secundaria3 release];  
[Imagen_secundaria1 release];  
[Imagen_secundaria2 release];  
[Imagen_secundaria3 release];
```

```
[super dealloc];  
}
```

@end

Anexo E: Código fuente "Clases"

```
// Pizarra.h
#import <UIKit/UIKit.h>
#import <QuartzCore/QuartzCore.h>

@class Ajustes;

@interface Pizarra : UIViewController <UINavigationControllerDelegate,
UIImagePickerControllerDelegate> {

    Ajustes * ControladorAjustes;

    int ajuste_resp_negativa;
    int ajuste_tiempo_pulsacion;
    int ajuste_tipo_letra;

    IBOutlet UIView *VISTA_PRIMERA_PIZARRA;
    IBOutlet UIView *MENU_PIZARRA;

    UIAlertView *alertView;

    IBOutlet UIButton * but_menu_cargar_imagen;
    IBOutlet UIButton * but_menu_deshacer;
    IBOutlet UIButton * but_menu_borrar_todo;
    IBOutlet UIButton * but_menu_guardar_imagen;
    IBOutlet UIButton * but_menu_inicio;

    CGPoint ultimoPunto;
    UIImageView *drawImage;
    BOOL userSwiped;
    int userMoved;
    int selectedUnit;
    UISegmentedControl *segmentedControl;
    int selectedUnit1;
    UISegmentedControl *segmentedControl_grosor;
    int selectedUnit2;
    UISegmentedControl *segmentedControl_color;

    float colorRed;
    float colorGreen;
    float colorBlue;
    float lineWidth;

    int menupizarra;

    IBOutlet UIToolbar *barra_pizarra;
```



```
UIPopoverController *popover;
}

@property(nonatomic,retain)Ajustes * ControladorAjustes;
@property (nonatomic, retain) IBOutlet UISegmentedControl *segmentedControl;
@property (nonatomic, retain) IBOutlet UISegmentedControl
*segmentedControl_grosor;
@property (nonatomic, retain) IBOutlet UISegmentedControl *segmentedControl_color;

@property (nonatomic, retain)IBOutlet UIButton * but_menu_cargar_imagen;
@property (nonatomic, retain)IBOutlet UIButton * but_menu_deshacer;
@property (nonatomic, retain)IBOutlet UIButton * but_menu_borrar_todo;
@property (nonatomic, retain)IBOutlet UIButton * but_menu_guardar_imagen;
@property (nonatomic, retain)IBOutlet UIButton * but_menu_inicio;

@property (nonatomic, retain) UIPopoverController *popover;
@property (nonatomic, retain) UIImage *copyImage;

- (void)Ajustar;

- (IBAction)saveDrawing;
- (IBAction)cancelDrawing: (id) sender;
- (IBAction)segmentedControlValueChanged;
- (void)cambioHerramienta;
- (IBAction)segmentedControlValueChanged1;
- (void)cambioGrosor;
- (IBAction)segmentedControlValueChanged2;
- (void)cambioColor;

- (IBAction)PULSAR_DESHACER: (id) sender;
- (IBAction)PULSAR_MENU_PIZARRA: (id) sender;
//- (IBAction)PULSAR_VOLVER_MENU_PIZARRA: (id) sender;
- (IBAction)choosePic: (id) sender;

@end
```

```
// Pizarra.m
#import <QuartzCore/QuartzCore.h>
#import "Pizarra.h"
#import "Ajustes.h"

@implementation Pizarra

@synthesize ControladorAjustes;

@synthesize segmentedControl;
@synthesize segmentedControl_grosor;
@synthesize segmentedControl_color;

@synthesize but_menu_cargar_imagen;
@synthesize but_menu_deshacer;
@synthesize but_menu_borrar_todo;
@synthesize but_menu_guardar_imagen;
@synthesize but_menu_inicio;

@synthesize popover;
@synthesize copyImage;

- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil
{
    self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil];
    if (self) {
        // Custom initialization
    }
    return self;
}

- (void)didReceiveMemoryWarning
{
    // Releases the view if it doesn't have a superview.
    [super didReceiveMemoryWarning];

    // Release any cached data, images, etc that aren't in use.
}

#pragma mark - View lifecycle
```

```
- (void)viewDidLoad  
{
```

```
ControladorAjustes =[[Ajustes alloc]init];
```

```
drawImage = [[UIImageView alloc] initWithImage:nil];  
drawImage.frame = VISTA_PRIMERA_PIZARRA.frame;  
[VISTA_PRIMERA_PIZARRA addSubview:drawImage];  
VISTA_PRIMERA_PIZARRA.backgroundColor = [UIColor whiteColor];  
userMoved = 0;  
menupizarra=0;
```

```
[self.view addSubview:VISTA_PRIMERA_PIZARRA];  
[VISTA_PRIMERA_PIZARRA setBackgroundColor:[UIColor whiteColor]];  
[barra_pizarra addSubview:segmentedControl];  
[barra_pizarra addSubview:segmentedControl_color];  
[barra_pizarra addSubview:segmentedControl_grosor];  
drawImage.image = nil;  
[VISTA_PRIMERA_PIZARRA setNeedsDisplay];  
colorRed = 0.0f;  
colorGreen = 0.0f;  
colorBlue = 0.0f;  
lineWidth = 1;  
segmentedControl.selectedSegmentIndex=0;  
segmentedControl_grosor.selectedSegmentIndex=0;  
segmentedControl_color.selectedSegmentIndex=0;  
[barra_pizarra setFrame:CGRectMake(0, 680, 1024, 68)];  
[segmentedControl setFrame:CGRectMake(614, 684, 150, 50)];  
[segmentedControl_grosor setFrame:CGRectMake(390, 684, 300, 50)];  
[segmentedControl_color setFrame:CGRectMake(70, 684, 300, 50)];
```

```
[self Ajustar];
```

```
[super viewDidLoad];  
// Do any additional setup after loading the view from its nib.
```

```
}
```

```
- (void)Ajustar  
{
```

```
[ControladorAjustes viewDidLoad];  
ajuste_resp_negativa=ControladorAjustes.resp_negativa;  
ajuste_tiempo_pulsacion=ControladorAjustes.tiempo_pulsacion;  
ajuste_tipo_letra=ControladorAjustes.tipo_letra;
```

```
NSLog(@"respuesta negativa %i, tiempo pulsacion %i, tipo letra  
%i",ajuste_resp_negativa,ajuste_tiempo_pulsacion,ajuste_tipo_letra);
```

```
}

- (void)viewDidUnload
{

    [super viewDidUnload];
    // Release any retained subviews of the main view.
    // e.g. self.myOutlet = nil;
}

- (BOOL)shouldAutorotateToInterfaceOrientation:
(UIInterfaceOrientation)interfaceOrientation
{
    // Return YES for supported orientations
    return YES;
}

- (IBAction) PULSAR_VOLVER_PIZARRA: (id) sender{

    [MENU_PIZARRA removeFromSuperview];
    menupizarra=0;
    [self.view removeFromSuperview];
    colorRed = 0.0;
    colorGreen = 0.0;
    colorBlue = 0.0;
    segmentedControl_color.selectedSegmentIndex=0;
    segmentedControl_grosor.selectedSegmentIndex=0;
    segmentedControl.selectedSegmentIndex=0;

}

- (void)touchesBegan:(NSSet *)touches withEvent:(UIEvent *)event {

    self.copyImage = drawImage.image;

    if (menupizarra==1){
        [MENU_PIZARRA removeFromSuperview];
        menupizarra=0;
    }

    userSwiped = NO;
    UITouch *touch = [touches anyObject];
    ultimoPunto = [touch locationInView:VISTA_PRIMERA_PIZARRA];
    ultimoPunto.y -= 1;//por la statusBar
}
```

```
}  
  
- (void)touchesMoved:(NSSet *)touches withEvent:(UIEvent *)event {  
  
    userSwiped = YES;  
  
    UITouch *touch = [touches anyObject];  
    CGPoint currentPoint = [touch locationInView:VISTA_PRIMERA_PIZARRA];  
  
    currentPoint.y -= 1;  
  
    UIGraphicsBeginImageContext(VISTA_PRIMERA_PIZARRA.frame.size);  
    [drawImage.image drawInRect: CGRectMake(0, 0,  
VISTA_PRIMERA_PIZARRA.frame.size.width,  
VISTA_PRIMERA_PIZARRA.frame.size.height)];  
    CGContextSetLineCap(UIGraphicsGetCurrentContext(), kCGLineCapRound);  
    CGContextSetLineWidth(UIGraphicsGetCurrentContext(), lineWidth);  
    CGContextSetRGBStrokeColor(UIGraphicsGetCurrentContext(),  
colorRed,colorGreen, colorBlue, 1.0);  
    CGContextBeginPath(UIGraphicsGetCurrentContext());  
    CGContextMoveToPoint(UIGraphicsGetCurrentContext(), ultimoPunto.x,  
ultimoPunto.y);  
    CGContextAddLineToPoint(UIGraphicsGetCurrentContext(), currentPoint.x,  
currentPoint.y);  
    CGContextStrokePath(UIGraphicsGetCurrentContext());  
    drawImage.image = UIGraphicsGetImageFromCurrentImageContext();  
    UIGraphicsEndImageContext();  
  
    ultimoPunto = currentPoint;  
  
    userMoved++;  
  
    if (userMoved == 10) {  
        userMoved = 0;  
    }  
  
}  
  
- (void)touchesEnded:(NSSet *)touches withEvent:(UIEvent *)event {
```

```
        if(!userSwiped) {
            UIGraphicsBeginImageContext(VISTA_PRIMERA_PIZARRA.frame.size);
            [drawImage.image drawInRect:CGRectMake(0, 0,
VISTA_PRIMERA_PIZARRA.frame.size.width,
VISTA_PRIMERA_PIZARRA.frame.size.height)];
            CGContextSetLineCap(UIGraphicsGetCurrentContext(), kCGLineCapRound);
            CGContextSetLineWidth(UIGraphicsGetCurrentContext(), lineWidth);
            CGContextSetRGBStrokeColor(UIGraphicsGetCurrentContext(),
colorRed,colorGreen, colorBlue, 1.0);
            CGContextMoveToPoint(UIGraphicsGetCurrentContext(), ultimoPunto.x,
ultimoPunto.y);
            CGContextAddLineToPoint(UIGraphicsGetCurrentContext(), ultimoPunto.x,
ultimoPunto.y);
            CGContextStrokePath(UIGraphicsGetCurrentContext());
            CGContextFlush(UIGraphicsGetCurrentContext());
            drawImage.image = UIGraphicsGetImageFromCurrentImageContext();
            UIGraphicsEndImageContext();

        }

    }

- (IBAction) PULSAR_MENU_PIZARRA: (id) sender
{
    [self Ajustar];

    if (ajuste_tipo_letra<40) // IDIOMA CASTELLANO
    {
        [but_menu_cargar_imagen setTitle:@"CARGAR IMAGEN" forState:normal];
        [but_menu_deshacer setTitle:@"DESHACER" forState:normal];
        [but_menu_borrar_todo setTitle:@"BORRAR TODO" forState:normal];
        [but_menu_guardar_imagen setTitle:@"GUARDAR IMAGEN" forState:normal];
        [but_menu_inicio setTitle:@"INICIO" forState:normal];

    }
    else //IDIOMA INGLÉS
    {

        [but_menu_cargar_imagen setTitle:@"LOAD PICTURE" forState:normal];
        [but_menu_deshacer setTitle:@"UNDO" forState:normal];
        [but_menu_borrar_todo setTitle:@"DELETE ALL" forState:normal];
        [but_menu_guardar_imagen setTitle:@"SAVE PICTURE" forState:normal];
        [but_menu_inicio setTitle:@"HOME" forState:normal];

    }

}
```

```
}

    if (menupizarra==0){
[VISTA_PRIMERA_PIZARRA addSubview: MENU_PIZARRA];
[MENU_PIZARRA setFrame:CGRectMake(0, 371, 300, 309)];
menupizarra=1;
    }
    else if (menupizarra==1){
[MENU_PIZARRA removeFromSuperview];
menupizarra=0;
    }

}

/*

- (IBAction) PULSAR_MENU_PIZARRA: (id) sender{

    if (menupizarra==0){
[VISTA_PRIMERA_PIZARRA addSubview: MENU_PIZARRA];
[MENU_PIZARRA setFrame:CGRectMake(0, 371, 300, 309)];
menupizarra=1;
    }
    else if (menupizarra==1){
[MENU_PIZARRA removeFromSuperview];
menupizarra=0;
    }

}

*/

- (void)PULSAR_VOLVER_MENU_PIZARRA: (id) sender{

[MENU_PIZARRA removeFromSuperview];
menupizarra=0;

}

*/

- (void)PULSAR_DESHACER: (id) sender{

    if (menupizarra==1){
[MENU_PIZARRA removeFromSuperview];
```

```
        menupizarra=0;
    }

    drawImage.image = copyImage;

}

- (void)viewDidAppear:(BOOL)animated {
    [super viewDidAppear:animated];
    selectedUnit = segmentedControl.selectedSegmentIndex;
    selectedUnit1 = segmentedControl_grosor.selectedSegmentIndex;
    selectedUnit2 = segmentedControl_color.selectedSegmentIndex;
    [self cambioHerramienta];
    [self cambioGrosor];
    [self cambioColor];
}

- (IBAction)segmentedControlValueChanged {
    selectedUnit = segmentedControl.selectedSegmentIndex;
    [self cambioHerramienta];
    [VISTA_PRIMERA_PIZARRA setNeedsDisplay];
}

- (IBAction)segmentedControlValueChanged1 {
    selectedUnit1 = segmentedControl_grosor.selectedSegmentIndex;
    [self cambioGrosor];
    [VISTA_PRIMERA_PIZARRA setNeedsDisplay];
}

- (IBAction)segmentedControlValueChanged2 {
    selectedUnit2 = segmentedControl_color.selectedSegmentIndex;
    [self cambioColor];
    [VISTA_PRIMERA_PIZARRA setNeedsDisplay];
}

- (void)cambioHerramienta {

    if (menupizarra==1){
        [MENU_PIZARRA removeFromSuperview];
        menupizarra=0;
    }
}
```



```
if (selectedUnit == 0) {
    colorRed = 0.0;
    colorGreen = 0.0;
    colorBlue = 0.0;
    segmentedControl_color.selectedSegmentIndex=0;
    [VISTA_PRIMERA_PIZARRA setNeedsDisplay];
}
if (selectedUnit == 1) {
    colorRed = 1.0;
    colorGreen = 1.0;
    colorBlue = 1.0;

    [VISTA_PRIMERA_PIZARRA setNeedsDisplay];
}

}

/*
Negro = 0 0 0
Azul = 0 0 255

Verde = 0 255 0
Cíán = 0 255 255
Rojo = 255 0 0
Magenta = 255 0 255
Amarillo = 255 255 0
Blanco = 255 255 255
*/

- (void)cambioColor {

    if (segmentedControl.selectedSegmentIndex==1){
        segmentedControl.selectedSegmentIndex=0;
    }

    if (menupizarra==1){
        [MENU_PIZARRA removeFromSuperview];
        menupizarra=0;
    }

    if (selectedUnit2 == 0) {
        colorRed = 0.0;
        colorGreen = 0.0;
        colorBlue = 0.0;
        [VISTA_PRIMERA_PIZARRA setNeedsDisplay];
    }
    if (selectedUnit2 == 1) {
```

```
    colorRed = 1.0;
    colorGreen = 0.0;
    colorBlue = 0.0;
    [VISTA_PRIMERA_PIZARRA setNeedsDisplay];
}
if (selectedUnit2 == 2) {
    colorRed = 1.0;
    colorGreen = 1.0;
    colorBlue = 0.0;
    [VISTA_PRIMERA_PIZARRA setNeedsDisplay];
}
if (selectedUnit2 == 3) {
    colorRed = 0.0;
    colorGreen = 0.0;
    colorBlue = 1.0;
    [VISTA_PRIMERA_PIZARRA setNeedsDisplay];
}
if (selectedUnit2 == 4) {
    colorRed = 0.0;
    colorGreen = 1.0;
    colorBlue = 0.0;
    [VISTA_PRIMERA_PIZARRA setNeedsDisplay];
}
if (selectedUnit2 == 5) {
    colorRed = 1.0;
    colorGreen = 1.0;
    colorBlue = 1.0;
    [VISTA_PRIMERA_PIZARRA setNeedsDisplay];
}
}

- (void)cambioGrosor {

    if (menupizarra==1){
        [MENU_PIZARRA removeFromSuperview];
        menupizarra=0;
    }

    if (selectedUnit1 == 0) {
        lineWidth = 1;
        [VISTA_PRIMERA_PIZARRA setNeedsDisplay];}
    if (selectedUnit1 == 1) {
        lineWidth = 5;
        [VISTA_PRIMERA_PIZARRA setNeedsDisplay];}
```

```
if (selectedUnit1 == 2) {
    lineWidth = 10;
    [VISTA_PRIMERA_PIZARRA setNeedsDisplay];}
if (selectedUnit1 == 3) {
    lineWidth = 20;
    [VISTA_PRIMERA_PIZARRA setNeedsDisplay];}
if (selectedUnit1 == 4) {
    lineWidth = 30;
    [VISTA_PRIMERA_PIZARRA setNeedsDisplay];}
if (selectedUnit1 == 5) {
    lineWidth = 35;
    [VISTA_PRIMERA_PIZARRA setNeedsDisplay];}

}

- (IBAction)saveDrawing {

    [MENU_PIZARRA removeFromSuperview];
    menupizarra=0;
    UIGraphicsBeginImageContext(self.view.bounds.size);
    [VISTA_PRIMERA_PIZARRA.layer
renderInContext:UIGraphicsGetCurrentContext()];
    UIImage *finishedPic =UIGraphicsGetImageFromCurrentImageContext();
    UIGraphicsEndImageContext();

    UIImageWriteToSavedPhotosAlbum(finishedPic, self,
    @selector(exitProg:didFinishSavingWithError:contextInfo:), nil);
}

- (IBAction)cancelDrawing: (id) sender {

    [MENU_PIZARRA removeFromSuperview];
    menupizarra=0;
    drawImage.image = nil;
    [VISTA_PRIMERA_PIZARRA setNeedsDisplay];

}

- (void)exitProg:(UIImage *)image didFinishSavingWithError:(NSError *)error
contextInfo:(void *)contextInfo {

    [self Ajustar];

    if (ajuste_tipo_letra<40) // IDIOMA CASTELLANO
    {
```

```
alertView = [[UIAlertView alloc] initWithTitle:@"ENHORABUENA" message:@"Su
imagen ha sido guardada" delegate:self cancelButtonTitle:nil
otherButtonTitles:@"Aceptar", nil];

}
else //IDIOMA INGLÉS
{
    alertView = [[UIAlertView alloc] initWithTitle:@"CONGRATULATIONS"
message:@"Your picture has been saved" delegate:self cancelButtonTitle:nil
otherButtonTitles:@"Ok", nil];

}

[alertView show];
[alertView release];

}

- (IBAction)choosePic: (id) sender {
    UIImagePickerController *imagePicker = [[UIImagePickerController alloc] init];
    imagePicker.delegate = self;
    imagePicker.allowsEditing = NO;
    imagePicker.sourceType = UIImagePickerControllerSourceTypePhotoLibrary;

    popover = [[UIPopoverController alloc]
initWithContentViewController:imagePicker];
    [imagePicker release];
    UIButton *btnSender = sender;

    CGRect r = btnSender.frame;
    [popover presentPopoverFromRect:r inView:self.view
permittedArrowDirections:UIPopoverArrowDirectionAny animated:YES];
    [MENU_PIZARRA removeFromSuperview];
}

- (void)imagePickerController:(UIImagePickerController *)picker_pizarra
didFinishPickingImage:(UIImage *)image editingInfo:(NSDictionary *)editingInfo {

    drawImage.image=image;
    [picker_pizarra release];
    [self.popover dismissPopoverAnimated:YES];
}
```

```
}
```

```
- (void)imagePickerControllerDidCancel:(UIImagePickerController *)picker_pizarra {
```

```
    [picker_pizarra release];
```

```
    [self.popover dismissPopoverAnimated:YES];
```

```
}
```

```
- (void)dealloc
```

```
{
```

```
    [segmentedControl release];
```

```
    [segmentedControl_grosor release];
```

```
    [copyImage release];
```

```
    [super dealloc];
```

```
}
```

```
@end
```

// ReLoj.h

```
#import <UIKit/UIKit.h>
#import <QuartzCore/QuartzCore.h>

@class Ajustes;

@interface ReLoj : UIViewController <UIPickerViewDelegate,
UIPickerViewDataSource>{

//Elementos usados para efectuar ajustes:
Ajustes * ControladorAjustes;

int ajuste_resp_negativa;
int ajuste_tiempo_pulsacion;
int ajuste_tipo_letra;

//Array del picker con las actividades y etiquetas de los pickers.

    NSArray *arrayDatos;

    IBOutlet UITextField *txt;
    IBOutlet UILabel *label;
    IBOutlet UILabel *label_crono;

    IBOutlet UIPickerView *picker; //picker actividades
    IBOutlet UIDatePicker *picker_crono;
    IBOutlet NSTimer *timer;
    int mainInt;
    int cuenta_atras;

    IBOutlet UIImageView *reloj_caracol;
    IBOutlet NSString *prueba;

    IBOutlet UIView *mascara;
    IBOutlet UIView *animalillo;

    IBOutlet UIView *VISTA_PRINCIPAL_RELOJ;
    IBOutlet UIView *VISTA_SECUNDARIA_RELOJ;
    IBOutlet UIView *VISTA_SECUNDARIA_IMAGEN;
    IBOutlet UIView *VISTA_IMAGEN_RELOJ;

    IBOutlet UIView *VISTA_BOTON_RELOJ_OK_BASE;
    IBOutlet UIView *VISTA_BOTON_RELOJ_OK;
    IBOutlet UIView *VISTA_ACCION_1;
```

```
IBOutlet UIView *VISTA_ACCION_2;  
IBOutlet UIView *VISTA_ACCION_3;  
  
IBOutlet UIView *VISTA_ACCION_4;  
IBOutlet UIView *VISTA_ACCION_OTRO;  
IBOutlet UIView *VISTA_ACCION_SELECCIONA;  
  
IBOutlet UIView *VISTA_SECUNDARIA_BASE_ATRAS;  
IBOutlet UIView *VISTA_SECUNDARIA_ATRAS;  
IBOutlet UIView *VISTA_SECUNDARIA_BASE_START;  
IBOutlet UIView *VISTA_SECUNDARIA_START;  
IBOutlet UIView *VISTA_SECUNDARIA_BASE_HOME;  
IBOutlet UIView *VISTA_SECUNDARIA_HOME;  
IBOutlet UIView *VISTA_SECUNDARIA_STOP;  
IBOutlet UIView *VISTA_BOTON_OCULTO;  
IBOutlet UIView *VISTA_BOTON_OCULTO_BASE;
```

//Ajustar

```
IBOutlet UIButton * but_reloj_inicio;  
IBOutlet UIButton * but_reloj_volver;  
IBOutlet UIButton * but_reloj_empezar;  
IBOutlet UIButton * but_reloj_parar;  
IBOutlet UIButton * but_reloj_inicioprimerero;
```

//Creo las capas del caracol

```
CALayer *hoja;  
CALayer *babamask;  
CALayer *hierba;  
CALayer *caracol;  
CALayer *cabeza;  
CALayer *baba;
```

//animaciones del caracol

```
CABasicAnimation *cabezamove;  
CABasicAnimation *caracolmove;  
CABasicAnimation *maskmove;
```

```
BOOL pulsacion_reloj;
```

```
}
```

```
@property (nonatomic, retain)IBOutlet UIButton * but_reloj_inicio;  
@property (nonatomic, retain)IBOutlet UIButton * but_reloj_volver;  
@property (nonatomic, retain)IBOutlet UIButton * but_reloj_empezar;
```

```
@property (nonatomic, retain)IBOutlet UIButton * but_reloj_parar;  
@property (nonatomic, retain)IBOutlet UIButton * but_reloj_inicioprimeros;
```

```
@property(nonatomic,retain)Ajustes * ControladorAjustes;
```

```
@property (nonatomic,retain) IBOutlet UIPickerView *picker;  
@property (nonatomic,retain) IBOutlet UILabel *label;  
@property (nonatomic, retain) IBOutlet NSTimer *timer;
```

```
- (void)Ajustar;
```

```
- (IBAction) PULSAR_VOLVER_RELOJ: (id) sender;
```

```
- (IBAction) PULSAR_VOLVER_OK_RELOJ: (id) sender;
```

```
- (IBAction) PULSAR_OK_RELOJ: (id) sender;
```

```
- (IBAction) PULSAR_VOLVER_OK_A_HOME: (id) sender;
```

```
- (IBAction) PULSAR_START: (id) sender;
```

```
- (IBAction) PULSAR_STOP: (id) sender;
```

```
- (IBAction) PULSAR_BOTON_OCULTO_PARAR: (id) sender;
```

```
@end
```



```
// ReLoj.m
```

```
#import "ReLoj.h"  
#import <AudioToolbox/AudioServices.h>  
#import "Ajustes.h"
```

```
@implementation ReLoj
```

```
@synthesize ControladorAjustes;
```

```
@synthesize picker;  
@synthesize label;  
@synthesize timer;
```

```
@synthesize but_reloj_inicio;  
@synthesize but_reloj_volver;  
@synthesize but_reloj_empezar;  
@synthesize but_reloj_parar;  
@synthesize but_reloj_inicioprimerero;
```

```
- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil  
{  
    self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil];  
    if (self) {  
        // Custom initialization  
    }  
    return self;  
}
```

```
- (void)dealloc  
{  
    [super dealloc];  
}
```

```
- (void)didReceiveMemoryWarning  
{  
    // Releases the view if it doesn't have a superview.  
    [super didReceiveMemoryWarning];  
  
    // Release any cached data, images, etc that aren't in use.  
}
```

```
#pragma mark - View lifecycle
```

```
- (void)viewDidLoad
{

    ControladorAjustes =[[Ajustes alloc]init];
    ControladorAjustes =[[Ajustes alloc]init];
    [self Ajustar];

    [self Ajustar];

    if (ajuste_tipo_letra<40)
    {
        [but_reloj_inicio setTitle:@"INICIO" forState:normal];
        [but_reloj_volver setTitle:@"VOLVER" forState:normal];
        [but_reloj_empezar setTitle:@"EMPEZAR" forState:normal];
        [but_reloj_parar setTitle:@"PARAR" forState:normal];
        [but_reloj_inicioprimeros setTitle:@"INICIO" forState:normal];
    }
    else
    {
        [but_reloj_inicio setTitle:@"HOME" forState:normal];
        [but_reloj_volver setTitle:@"BACK" forState:normal];
        [but_reloj_empezar setTitle:@"START" forState:normal];
        [but_reloj_parar setTitle:@"STOP" forState:normal];
        [but_reloj_inicioprimeros setTitle:@"HOME" forState:normal];
    }

    if (ajuste_tipo_letra<40) // IDIOMA CASTELLANO
    {
        arrayDatos = [[NSArray alloc] initWithObjects:@"SELECCIONA
ACTIVIDAD:",@"TRABAJAR", @"JUGAR", @"COMER", @"BAILAR", @"OTRO", nil];
        label.text=@"SELECCIONA ACTIVIDAD:";
    }
    else //IDIOMA INGLÉS
    {
        arrayDatos = [[NSArray alloc] initWithObjects:@"CHOOSE
ACTIVITY:",@"WORK", @"PLAY", @"EAT", @"DANCE", @"OTHER", nil];
        label.text=@"CHOOSE ACTIVITY:";
    }

    [picker reloadData];

    self.view.backgroundColor = [UIColor groupTableViewBackgroundColor];
}
```

```
[picker_crono addTarget:self
    action:@selector(changeDateInLabel:)
    forControlEvents:UIControlEventValueChanged];
pulsacion_reloj=NO;

[self.view addSubview: VISTA_PRINCIPAL_RELOJ];

[VISTA_PRINCIPAL_RELOJ addSubview:VISTA_BOTON_RELOJ_OK_BASE];
[VISTA_PRINCIPAL_RELOJ addSubview:VISTA_IMAGEN_RELOJ];
[VISTA_SECUNDARIA_RELOJ addSubview:VISTA_SECUNDARIA_BASE_ATRAS];
[VISTA_SECUNDARIA_RELOJ addSubview:VISTA_SECUNDARIA_BASE_START];
[VISTA_SECUNDARIA_RELOJ addSubview:VISTA_SECUNDARIA_BASE_HOME];
[VISTA_IMAGEN_RELOJ addSubview:VISTA_ACCION_SELECCIONA];

[super viewDidLoad];
// Do any additional setup after loading the view from its nib.
}

- (void)Ajustar
{
    [ControladorAjustes viewDidLoad];
    ajuste_resp_negativa=ControladorAjustes.resp_negativa;
    ajuste_tiempo_pulsacion=ControladorAjustes.tiempo_pulsacion;
    ajuste_tipo_letra=ControladorAjustes.tipo_letra;

    NSLog(@"respuesta negativa %i, tiempo pulsacion %i, tipo letra
%i",ajuste_resp_negativa,ajuste_tiempo_pulsacion,ajuste_tipo_letra);
}

- (void)viewDidUnload
{
    [super viewDidUnload];
    // Release any retained subviews of the main view.
    // e.g. self.myOutlet = nil;
}

- (BOOL)shouldAutorotateToInterfaceOrientation:
(UIInterfaceOrientation)interfaceOrientation
{
    // Return YES for supported orientations
    return YES;
}

- (NSInteger)pickerView:(UIPickerView *)pickerView numberOfRowsInComponent:
(NSInteger)component
```

```
{  
  
    return [arrayDatos count];  
  
}  
  
- (NSInteger)numberOfComponentsInPickerView:(UIPickerView *)pickerView  
{  
    return 1;  
}  
  
- (NSString *)pickerView:(UIPickerView *)pickerView titleForRow:(NSInteger)row  
forComponent:(NSInteger)component  
{  
    NSString *returnStr;  
    if (component == 0 )  
    {  
        returnStr = [arrayDatos objectAtIndex:row];  
    }  
  
    return returnStr;  
  
}  
  
//picker de actividades.  
  
- (void)pickerView:(UIPickerView *)pickerView didSelectRow:(NSInteger)row  
inComponent:(NSInteger)component{  
    //[[[UIApplication sharedApplication] delegate] cargarVista:row];  
    label.text = [arrayDatos objectAtIndex:row];  
    prueba=@"00:00:00";  
    if ((label.text=="@OTRO")||((label.text=="@OTHER"))){  
        txt.hidden=NO;  
        [VISTA_IMAGEN_RELOJ addSubview:VISTA_ACCION_OTRO];  
        [txt becomeFirstResponder];  
  
    }  
    else {  
        txt.hidden=YES;  
        [txt resignFirstResponder];  
    }  
  
    if (((label.text=="@SELECCIONA ACTIVIDAD:")||((label.text=="@CHOOSE  
ACTIVITY:"))| ([label_crono.text isEqualToString:prueba])){
```

```
[VISTA_BOTON_RELOJ_OK removeFromSuperview];
[VISTA_IMAGEN_RELOJ addSubview:VISTA_ACCION_SELECCIONA];

}
else {
    [VISTA_BOTON_RELOJ_OK_BASE addSubview:VISTA_BOTON_RELOJ_OK];
}

//En función de lo seleccionado en el picker, saco una imagen u otra
if ((label.text=="@TRABAJAR")||((label.text=="@WORK")))
{
    [VISTA_IMAGEN_RELOJ addSubview:VISTA_ACCION_1];
}
else if ((label.text=="@JUGAR")||((label.text=="@PLAY")))
{
    [VISTA_IMAGEN_RELOJ addSubview:VISTA_ACCION_2];
}
else if ((label.text=="@COMER")||((label.text=="@EAT")))
{
    [VISTA_IMAGEN_RELOJ addSubview:VISTA_ACCION_3];
}
else if ((label.text=="@BAILAR")||((label.text=="@DANCE")))
{
    [VISTA_IMAGEN_RELOJ addSubview:VISTA_ACCION_4];
}
else if ((label.text=="@OTRO")||((label.text=="@OTRO")))
{
    [VISTA_IMAGEN_RELOJ addSubview:VISTA_ACCION_OTRO];
}
}

-(IBAction) PULSAR_VOLVER_RELOJ: (id) sender
{
    [self Ajustar];

    if (ajuste_tipo_letra<40)
    {
        [but_reloj_volver setTitle:@"VOLVER" forState:normal];
        [but_reloj_inicioprimer setTitulo:@"INICIO" forState:normal];
    }
}
```

```
}
else
{
    [but_reloj_volver setTitle:@"BACK" forState:normal];
    [but_reloj_inicioprimerero setTitle:@"HOME" forState:normal];
}

if (ajuste_tipo_letra<40) // IDIOMA CASTELLANO
{
    label.text=@"SELECCIONA ACTIVIDAD:";
}
else //IDIOMA INGLÉS
{
    label.text=@"SELECT ONE ACTIVITY:";
}

[self.view removeFromSuperview];
txt.hidden=YES;
[VISTA_BOTON_RELOJ_OK removeFromSuperview];
[VISTA_IMAGEN_RELOJ addSubview:VISTA_ACCION_SELECCIONA];
txt.text=@"";
[picker selectRow:0 inComponent:0 animated:NO];
picker_crono.date=[NSDate distantPast];

NSDateFormatter *format = [[NSDateFormatter alloc] init];
[format setDateFormat:@"hh:mm:ss"];
NSString *prod=@"00:00:00";
NSDate *myDate = [format dateFromString: prod];
picker_crono.date=myDate;

NSDateFormatter *df = [[NSDateFormatter alloc] init];
df.dateFormat = NSDateFormatterMediumStyle;
[df setDateFormat:@"HH:mm:ss"];
label_crono.text = [NSString stringWithFormat:@"%@",
                    [df stringFromDate:picker_crono.date]];
[df release];
}

//Actualizar etiqueta label_crono en funcion del picker_crono
```

```
- (void)changeDateInLabel:(id)sender{
    //Use NSDateFormatter to write out the date in a friendly format
    NSDateFormatter *df = [[NSDateFormatter alloc] init];
    df.dateStyle = NSDateFormatterMediumStyle;
    [df setDateFormat:@"HH:mm:ss"];
    label_crono.text = [NSString stringWithFormat:@"%@",
        [df stringFromDate:picker_crono.date]];
    [df release];

    if (((label.text=="SELECCIONA ACTIVIDAD:")||(label.text=="CHOOSE
ACTIVITY:")) | ([label_crono.text isEqualToString:prueba])){

        [VISTA_BOTON_RELOJ_OK removeFromSuperview];
        [VISTA_IMAGEN_RELOJ addSubview:VISTA_ACCION_SELECCIONA];

    }
    else {
        [VISTA_BOTON_RELOJ_OK_BASE addSubview:VISTA_BOTON_RELOJ_OK];
    }

}

-(IBAction) PULSAR_OK_RELOJ: (id) sender
{
    [VISTA_PRINCIPAL_RELOJ removeFromSuperview];
    [self.view addSubview:VISTA_SECUNDARIA_RELOJ];
    [VISTA_SECUNDARIA_RELOJ addSubview:VISTA_SECUNDARIA_IMAGEN];
    [VISTA_SECUNDARIA_BASE_ATRAS
addSubview:VISTA_SECUNDARIA_ATRAS];
    [VISTA_SECUNDARIA_BASE_START
addSubview:VISTA_SECUNDARIA_START];
    [VISTA_SECUNDARIA_BASE_HOME
addSubview:VISTA_SECUNDARIA_HOME];
    //[VISTA_SECUNDARIA_RELOJ addSubview:reloj];
    [VISTA_SECUNDARIA_RELOJ addSubview:reloj_caracol];
    [VISTA_SECUNDARIA_RELOJ addSubview:mascara];
    [VISTA_SECUNDARIA_RELOJ addSubview:animalillo];
    //[reloj setImage:[UIImage imageNamed:@"reloj_0.gif"]];

    if ((label.text=="TRABAJAR")||(label.text=="WORK"))
    {
        [VISTA_SECUNDARIA_IMAGEN addSubview:VISTA_ACCION_1];
    }
}
```

```
}  
else if ((label.text=="@JUGAR")||((label.text=="@PLAY"))  
{  
    [VISTA_SECUNDARIA_IMAGEN addSubview:VISTA_ACCION_2];  
}  
else if ((label.text=="@COMER")||((label.text=="@EAT"))  
{  
    [VISTA_SECUNDARIA_IMAGEN addSubview:VISTA_ACCION_3];  
}  
else if ((label.text=="@BAILAR")||((label.text=="@DANCE"))  
{  
    [VISTA_SECUNDARIA_IMAGEN addSubview:VISTA_ACCION_4];  
}  
  
if (txt.hidden == NO) {  
    label.text=txt.text;  
    [VISTA_SECUNDARIA_IMAGEN addSubview:VISTA_ACCION_OTRO];  
}
```

```
//Coloco las capas del caracol  
hierba=[CALayer layer];  
hierba=reloj_caracol.layer; // "hierba" es una CALayer que va insertada dentro  
del UIView "reloj".  
hierba.bounds=CGRectMake(0.0f, 0.0f, 900.0f, 95.0f); // los límites de la imagen  
que contenga, en el origen 0,0 con respecto al total de pantalla.  
hierba.anchorPoint=CGPointMake(0.5f, 0.0f); // el punto de referencia de la capa  
se sitúa en el 0,5; 0,0 de la capa en la que está contenida.  
hierba.contents= (id)[[UIImage imageNamed:@"hierba.png"] CGImage]; // Esta  
CALayer contiene la imagen "hierba.png"  
  
hoja=[CALayer layer];  
hoja.bounds=CGRectMake(0.0f, 0.0f, 134.0f, 146.0f);  
hoja.position=CGPointMake(820.0f, 30.0f); // Se señala la posición con respecto  
a la capa que va a ocupar. (sublayer de "hierba")  
hoja.contents= (id)[[UIImage imageNamed:@"hoja_caracol.png"] CGImage];  
  
[hierba addSubview:hoja];  
  
baba=[CALayer layer]; // "baba" es una CALayer que va insertada dentro del  
UIView "mascara",  
baba=mascara.layer; // puesto que va a tener un efecto así, y no queremos que  
el efecto se aplique a otros UIViews.(como "hierba")  
baba.bounds=CGRectMake(0.0f, 0.0f, 900.0f, 95.0f);  
baba.anchorPoint=CGPointMake(0.5f, 0.4f);  
baba.contents= (id)[[UIImage imageNamed:@"baba.png"] CGImage]; // Contiene  
la imagen propia de la baba.
```



```
babamask=[CALayer layer]; // "babamask" es una CALayer que va a hacer el
efecto de máscara sobre la CALayer que la coloquemos.
// La imagen que contenga, será la forma del hueco que dejará ver.
babamask.bounds=CGRectMake(0.0f, 0.0f, 150.0f, 95.0f);
babamask.anchorPoint=CGPointMake(0.0f, 0.0f);
babamask.contents=(id)[[UIImage imageNamed:@"mascara.png"] CGImage]; //
contiene la imagen "mascara.png" consistente en un rectángulo.
```

```
[baba setMask:babamask]; // Se aplica la máscara solamente a la capa deseada.
```

```
caracol=[CALayer layer];
caracol=animalillo.layer;
caracol.bounds=CGRectMake(0.0f, 0.0f, 200.0f, 174.0f);
caracol.anchorPoint=CGPointMake(2.3f, 0.25f);
```

```
caracol.contents= (id)[[UIImage imageNamed:@"caracolillo.png"] CGImage];
```

```
cabeza=[CALayer layer];
cabeza.bounds=CGRectMake(0.0f, 0.0f, 200.0f, 174.0f);
cabeza.position=CGPointMake(0.0f, 174.0f);
cabeza.anchorPoint=CGPointMake(0.0f, 1.0f);
cabeza.contents= (id)[[UIImage imageNamed:@"cabeza_caracol.png"]
```

```
CGImage];
```

```
[caracol addSublayer:cabeza];
```

```
}
```

```
- (IBAction) PULSAR_START: (id) sender;
```

```
{
```

```
mainInt=[picker_crono countdownDuration];
```

```
cuenta_atras=[picker_crono countdownDuration];
```

```
//cuenta_atras=cuenta_atras/10;
```

```
timer=[NSTimer scheduledTimerWithTimeInterval:(1.0/1.0) target:self
```

```
selector:@selector(updateLabel) userInfo:nil repeats:YES];
```

```
[self Ajustar];
```

```
if (ajuste_tipo_letra<40)
```

```
{
```

```
[but_reloj_empezar setTitle:@"EMPEZAR" forState:normal];
```

```
}
```

```
else
{
    [but_reloj_empezar setTitle:@"START" forState:normal];

}

[VISTA_SECUNDARIA_START removeFromSuperview];
[VISTA_SECUNDARIA_ATRAS removeFromSuperview];
[VISTA_SECUNDARIA_HOME removeFromSuperview];
[VISTA_SECUNDARIA_RELOJ addSubview:VISTA_BOTON_OCULTO_BASE];
[VISTA_BOTON_OCULTO_BASE addSubview:VISTA_BOTON_OCULTO];

//Creamos la animacion del caracol
caracolmove=[CABasicAnimation animationWithKeyPath:@"anchorPoint"];
caracolmove.fromValue=[NSValue valueWithCGPoint: CGPointMake(2.3, 0.25)];
caracolmove.toValue=[NSValue valueWithCGPoint: CGPointMake(-0.8, 0.25)];
caracolmove.duration=cuenta_atras;

caracolmove.fillMode = kCAFillModeForwards; //estas dos líneas hacen que la
animación se mantenga en su posición final.
caracolmove.removedOnCompletion = NO;

maskmove=[CABasicAnimation animationWithKeyPath:@"bounds"];
maskmove.fromValue=[NSNumber valueWithCGRect: CGRectMake(0.0f, 0.0f,
150.0f, 95.0f)];
maskmove.toValue=[NSNumber valueWithCGRect: CGRectMake(0.0f, 0.0f,
700.0f, 95.0f)];
maskmove.duration=cuenta_atras;
maskmove.fillMode = kCAFillModeForwards;
maskmove.removedOnCompletion=NO;

// Añadimos la animación a la capa correspondiente y así la inicializamos:

babamask.speed=1.0; //Activamos la velocidad en caso de que haya sido
desactivada.
caracol.speed=1.0;

[caracol addAnimation:caracolmove forKey:@"caracolmove"];
[babamask addAnimation:maskmove forKey:@"maskmove"];

}
```

```
- (IBAction) PULSAR_STOP: (id) sender;
{
    [timer invalidate];

    [animalillo removeFromSuperview];
    [mascara removeFromSuperview];

    [self Ajustar];

    if (ajuste_tipo_letra < 40)
    {
        [but_reloj_parar setTitle:@"PARAR" forState:normal];
    }
    else
    {
        [but_reloj_parar setTitle:@"STOP" forState:normal];
    }

    [VISTA_SECUNDARIA_STOP removeFromSuperview];
    [VISTA_BOTON_OCULTO removeFromSuperview];
    [VISTA_SECUNDARIA_BASE_HOME
addSubview:VISTA_SECUNDARIA_HOME];
}

//Actualizar label_crono y funcionamiento de reloj.

-(void) updateLabel{

    mainInt = mainInt - 1;
    if (mainInt == 0.0) {

        [timer invalidate];

        [VISTA_BOTON_OCULTO removeFromSuperview];
        [VISTA_SECUNDARIA_STOP removeFromSuperview];
        [VISTA_SECUNDARIA_BASE_HOME
addSubview:VISTA_SECUNDARIA_HOME];
    }
}
```

```
cabezamove=[CABasicAnimation
animationWithKeyPath:@"transform.rotation.z"];
cabezamove.fromValue = [NSNumber numberWithFloat:0];
cabezamove.toValue = [NSNumber numberWithFloat:0.05];
cabezamove.duration=1;
cabezamove.autoreverses=YES;
cabezamove.repeatCount=2;

[cabeza addAnimation:cabezamove forKey:@"cabezamove"];
hoja.contents= (id)[[UIImage imageNamed:@"hoja_mordida.png"] CGImage];

//Hacemos que suene el mordisco
SystemSoundID mordisco;
AudioServicesCreateSystemSoundID((CFURLRef)[NSURL fileURLWithPath:
[[NSBundle mainBundle] pathForResource:@"mordisco" ofType:@"wav"]], &mordisco);
AudioServicesPlaySystemSound (mordisco);
}

NSDate *nuevaFecha = [NSDate
dateWithTimeIntervalSinceReferenceDate:mainInt-3600];
NSDateFormatter *df = [[NSDateFormatter alloc] init];
[df setDateFormat:@"HH:mm:ss"];
label_crono.text = [df stringFromDate:nuevaFecha];
[df release];
}

-(IBAction) PULSAR_VOLVER_OK_RELOJ: (id) sender
{
[VISTA_SECUNDARIA_STOP removeFromSuperview];
[VISTA_SECUNDARIA_RELOJ removeFromSuperview];
[self.view addSubview:VISTA_PRINCIPAL_RELOJ];
[VISTA_PRINCIPAL_RELOJ addSubview:VISTA_IMAGEN_RELOJ];

if ((label.text==@"TRABAJAR")||((label.text==@"WORK")))
{
[VISTA_IMAGEN_RELOJ addSubview:VISTA_ACCION_1];
}
else if ((label.text==@"JUGAR")||((label.text==@"PLAY")))
{
[VISTA_IMAGEN_RELOJ addSubview:VISTA_ACCION_2];
}
else if ((label.text==@"COMER")||((label.text==@"EAT")))
```

```
{
    [VISTA_IMAGEN_RELOJ addSubview:VISTA_ACCION_3];
}
else if ((label.text==@"BAILAR")||((label.text==@"DANCE")))
{
    [VISTA_IMAGEN_RELOJ addSubview:VISTA_ACCION_4];
}
else if ((label.text==@"OTRO")||((label.text==@"OTRO")))
{
    [VISTA_IMAGEN_RELOJ addSubview:VISTA_ACCION_OTRO];
}
}

-(IBAction) PULSAR_VOLVER_OK_A_HOME: (id) sender
{
    [self Ajustar];

    if (ajuste_tipo_letra<40) // IDIOMA CASTELLANO
    {
        label.text=@"SELECCIONA ACTIVIDAD:";
    }
    else //IDIOMA INGLÉS
    {
        label.text=@"SELECT ONE ACTIVITY:";
    }

    if (ajuste_tipo_letra<40)

    {
        [but_reloj_inicio setTitle:@"INICIO" forState:normal];
    }
    else
    {
        [but_reloj_inicio setTitle:@"HOME" forState:normal];
    }

    [VISTA_SECUNDARIA_RELOJ removeFromSuperview];
    [self.view addSubview: VISTA_PRINCIPAL_RELOJ];
    [self.view removeFromSuperview];
}
```

```
babamask.speed=0.0; //Inhabilitamos la velocidad
caracol.speed=0.0;
[cabeza removeFromSuperlayer];
[hoja removeFromSuperlayer];

txt.text=@"";
txt.hidden=YES;
[VISTA_BOTON_RELOJ_OK removeFromSuperview];
[picker selectRow:0 inComponent:0 animated:NO];

NSDateFormatter *format = [[NSDateFormatter alloc] init];
[format setDateFormat:@"hh:mm:ss"];
NSString *prod=@"00:00:00";
NSDate *myDate = [format dateFromString: prod];
picker_crono.date=myDate;

NSDateFormatter *df = [[NSDateFormatter alloc] init];
df.dateFormat = NSDateFormatterMediumStyle;
[df setDateFormat:@"HH:mm:ss"];
label_crono.text = [NSString stringWithFormat:@"%@",
                    [df stringFromDate:picker_crono.date]];
[df release];
}

- (IBAction) PULSAR_BOTON_OCULTO_PARAR: (id) sender
{
    if (pulsacion_reloj == NO)
    {
        pulsacion_reloj = YES;
        [VISTA_SECUNDARIA_BASE_START
addSubview:VISTA_SECUNDARIA_STOP];
    }
    else
    {
        pulsacion_reloj = NO;
        [VISTA_SECUNDARIA_STOP removeFromSuperview];
    }
}

@end
```

// Ajustes.h

```
#import <UIKit/UIKit.h>

# define KEY_TIPO_LETRA @"tipo_letra"
# define KEY_RESP_NEGATIVA @"resp_negativa"
# define KEY_TIEMPO_PULSACION @"tiempo_pulsacion"

# define ficheroPersistente @"ajustes.plist"

@class COMUNICADORViewController;
@class EditarAlbumFavorito;
@class Personalizacion;

@interface Ajustes : UIViewController {

    NSString *rutaFichero;
    NSMutableArray *ajustes;

    COMUNICADORViewController *controladorViewController;
    EditarAlbumFavorito * EditarAlbumFavoritoVista;
    Personalizacion * controladorPersonalizacion;

    int selectedUnit_resp_negativa;
    IBOutlet UISegmentedControl *segmentedControl_resp_negativa;

    int selectedUnit3;
    UISegmentedControl *segmentedControl_tiempo_pulsacion;

    int selectedUnit_minusc_mayusc;
    UISegmentedControl *segmentedControl_minusc_mayusc;

    int selectedUnit_cast_ing;
    UISegmentedControl *segmentedControl_cast_ing;

    IBOutlet UILabel * label_resp_negativa;
    IBOutlet UILabel * label_tiempo_pulsacion;
    IBOutlet UILabel * label_tamano_letra;
    IBOutlet UILabel * label_idioma;
    IBOutlet UILabel * label_pictos_personalizar;
    IBOutlet UILabel * label_album_favorito_personalizar;

    IBOutlet UIButton * but_pictos_personalizar;
    IBOutlet UIButton * but_album_favorito_personalizar;
```

```
IBOutlet UIButton * but_guardar_ajustes;

IBOutlet UIButton * but_inicio;

IBOutlet id BOTON_AJUSTES;

int resp_negativa;
int tiempo_pulsacion;
int tipo_letra;

NSString * resp_negativa_string;
NSString * tiempo_pulsacion_string;
NSString * tipo_letra_string;

}

@property (nonatomic, retain) NSString *rutaFichero;
@property (nonatomic, retain) NSMutableArray *ajustes;

@property (nonatomic, retain) COMUNICADORViewController
*controladorViewController;
@property (nonatomic, retain) EditarAlbumFavorito *EditarAlbumFavoritoVista;
@property (nonatomic, retain) Personalizacion *controladorPersonalizacion;
@property (nonatomic, retain) IBOutlet UISegmentedControl
*segmentedControl_resp_negativa;
@property (nonatomic, retain) IBOutlet UISegmentedControl
*segmentedControl_tiempo_pulsacion;
@property (nonatomic, retain) IBOutlet UISegmentedControl
*segmentedControl_minusc_mayusc;
@property (nonatomic, retain) IBOutlet UISegmentedControl
*segmentedControl_cast_ing;

@property (nonatomic, retain) IBOutlet UILabel * label_resp_negativa;
@property (nonatomic, retain) IBOutlet UILabel * label_tiempo_pulsacion;
@property (nonatomic, retain) IBOutlet UILabel * label_tamano_letra;
@property (nonatomic, retain) IBOutlet UILabel * label_idioma;
@property (nonatomic, retain) IBOutlet UILabel * label_pictos_personalizar;
@property (nonatomic, retain) IBOutlet UILabel * label_album_favorito_personalizar;

@property (nonatomic, retain) IBOutlet UIButton * but_pictos_personalizar;
@property (nonatomic, retain) IBOutlet UIButton * but_album_favorito_personalizar;
@property (nonatomic, retain) IBOutlet UIButton * but_guardar_ajustes;
@property (nonatomic, retain) IBOutlet UIButton * but_inicio;
```



```
@property (nonatomic) int resp_negativa;  
@property (nonatomic) int tiempo_pulsacion;  
@property (nonatomic) int tipo_letra;  
  
@property (nonatomic,retain) NSString *resp_negativa_string;  
@property (nonatomic,retain) NSString *tiempo_pulsacion_string;  
  
@property (nonatomic,retain) NSString *tipo_letra_string;  
  
- (IBAction)segmentedControlValueChangedRespNegativa;  
- (void)cambioRespuestaNegativa;  
  
- (IBAction)segmentedControlValueChanged3;  
- (void)cambioTiempoPulsacion;  
  
- (IBAction)segmentedControlValueChanged4;  
- (void)cambiodioma;  
  
- (IBAction)segmentedControlValueChanged6;  
- (void)cambioMayusculasMinusculas;  
  
- (IBAction)ajustes_personalizar_pictos;  
- (IBAction)ajustes_personalizar_album_favorito;  
  
- (NSString *) rutaFichero;  
- (void) mostrarAjustes;  
- (void) guardarAjustes;  
  
- (int)CargarRespuestaNegativa;  
  
//- (IBAction) PULSAR_BOTON OPCIONES: (id) sender;  
- (IBAction) PULSAR_BOTON_SALIR_AJUSTES: (id) sender;  
- (IBAction) PULSAR_BOTON_GUARDAR_AJUSTES: (id)sender;  
  
@end
```

// Ajustes.m

```
#import "Ajustes.h"
#import "COMUNICADORViewController.h"
#import "EditarAlbumFavorito.h"
#import "Personalizacion.h"

@implementation Ajustes

@synthesize rutaFichero;
@synthesize ajustes;

@synthesize controladorViewController;
@synthesize EditarAlbumFavoritoVista;
@synthesize controladorPersonalizacion;

@synthesize segmentedControl_resp_negativa;
@synthesize segmentedControl_tiempo_pulsacion;
@synthesize segmentedControl_minusc_mayusc;
@synthesize segmentedControl_cast_ing;

@synthesize label_resp_negativa;
@synthesize label_tiempo_pulsacion;
@synthesize label_tamano_letra;
@synthesize label_idioma;
@synthesize label_pictos_personalizar;
@synthesize label_album_favorito_personalizar;

@synthesize but_pictos_personalizar;
@synthesize but_album_favorito_personalizar;
@synthesize but_guardar_ajustes;
@synthesize but_inicio;

@synthesize resp_negativa;
@synthesize tiempo_pulsacion;
@synthesize tipo_letra;

@synthesize resp_negativa_string;
@synthesize tiempo_pulsacion_string;
@synthesize tipo_letra_string;

- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil
{
```

```
self = [super initWithNibName:nibNameOrNil bundle:nibNameOrNil];
if (self) {

// Custom initialization
}
return self;
}

- (void)dealloc
{
[super dealloc];
}

- (void)didReceiveMemoryWarning
{
// Releases the view if it doesn't have a superview.
[super didReceiveMemoryWarning];

// Release any cached data, images, etc that aren't in use.
}

#pragma mark - View lifecycle

- (void)viewDidLoad
{
NSString *fichero= [self rutaFichero];
if ([[NSFileManager defaultManager] fileExistsAtPath:fichero])
{
NSArray *listado =[[NSArray alloc] initWithContentsOfFile:fichero];
ajustes=[[NSMutableArray alloc] initWithArray:listado];
[self mostrarAjustes];
}
else
{
ajustes=[[NSMutableArray alloc] init];

resp_negativa=1;
tiempo_pulsacion=0.5;
tipo_letra=20;

[self guardarAjustes];
}

[super viewDidLoad];
// Do any additional setup after loading the view from its nib.
}
```

```
- (int)CargarRespuestaNegativa
{

    return resp_negativa;

}

- (void)viewDidUnload
{
    [super viewDidUnload];
    // Release any retained subviews of the main view.
    // e.g. self.myOutlet = nil;
}

- (BOOL)shouldAutorotateToInterfaceOrientation:
(UIInterfaceOrientation)interfaceOrientation
{
    // Return YES for supported orientations
    return YES;
}

- (void)viewDidAppear:(BOOL)animated {
    [super viewDidAppear:animated];
    selectedUnit_resp_negativa=segmentedControl_resp_negativa.selectedSegmentIndex;
    [self cambioRespuestaNegativa];
    selectedUnit3 = segmentedControl_tiempo_pulsacion.selectedSegmentIndex;
    [self cambioTiempoPulsacion];
    selectedUnit_cast_ing = segmentedControl_cast_ing.selectedSegmentIndex;
    [self cambioldioma];
    selectedUnit_minusc_mayusc =
segmentedControl_minusc_mayusc.selectedSegmentIndex;
    [self cambioMayusculasMinusculas];

}

- (IBAction)segmentedControlValueChangedRespNegativa
{
    selectedUnit_resp_negativa =
segmentedControl_resp_negativa.selectedSegmentIndex;
    [self cambioRespuestaNegativa];
    [self.view setNeedsDisplay];
}

- (IBAction)segmentedControlValueChanged3 {
    selectedUnit3 = segmentedControl_tiempo_pulsacion.selectedSegmentIndex;
```

```
[self cambioTiempoPulsacion];
[self.view setNeedsDisplay];
}

- (IBAction)segmentedControlValueChanged4 {
    selectedUnit_cast_ing = segmentedControl_cast_ing.selectedSegmentIndex;
    [self cambioldioma];
    [self.view setNeedsDisplay];
}

- (IBAction)segmentedControlValueChanged6 {
    selectedUnit_minusc_mayusc =
segmentedControl_minusc_mayusc.selectedSegmentIndex;
    [self cambioMayusculasMinusculas];
    [self.view setNeedsDisplay];
}

- (void)cambioRespuestaNegativa
{
    if (selectedUnit_resp_negativa == 0) {
        resp_negativa=0;
        [self.view setNeedsDisplay];
    }
    else if (selectedUnit_resp_negativa == 1) {
        resp_negativa=1;
        [self.view setNeedsDisplay];
    }
}

- (void)cambioTiempoPulsacion {
    if (selectedUnit3 == 0) {
        tiempo_pulsacion=0.5;
        [self.view setNeedsDisplay];
    }
    else if (selectedUnit3 == 1) {
        tiempo_pulsacion=1;
        [self.view setNeedsDisplay];
    }
    else if (selectedUnit3 == 2) {
        tiempo_pulsacion=2;
        [self.view setNeedsDisplay];
    }
    else if (selectedUnit3 == 3) {
        tiempo_pulsacion=3;
        [self.view setNeedsDisplay];
    }
}
```

```
    else if (selectedUnit3 == 4) {
        tiempo_pulsacion=4;
        [self.view setNeedsDisplay];
    }

    //[(seleccionAlbumAjustes setAjustes:tiempo_pulsacion);

}

- (void)cambiolIdioma
{
    if (selectedUnit_cast_ing == 0 && selectedUnit_minusc_mayusc==0)
    {
        tipo_letra=20;
        [self.view setNeedsDisplay];
    }
    else if (selectedUnit_cast_ing == 0 && selectedUnit_minusc_mayusc==1)
    {
        tipo_letra=30;
        [self.view setNeedsDisplay];
    }
    else if (selectedUnit_cast_ing == 1 && selectedUnit_minusc_mayusc==0)
    {
        tipo_letra=50;
        [self.view setNeedsDisplay];
    }
    else if (selectedUnit_cast_ing == 1 && selectedUnit_minusc_mayusc==1)
    {
        tipo_letra=60;
        [self.view setNeedsDisplay];
    }
}

- (void)cambioMayusculasMinusculas
{
    if (selectedUnit_cast_ing == 0 && selectedUnit_minusc_mayusc==0)
    {
        tipo_letra=20;
        [self.view setNeedsDisplay];
    }
    else if (selectedUnit_cast_ing == 0 && selectedUnit_minusc_mayusc==1)
    {
        tipo_letra=30;
        [self.view setNeedsDisplay];
    }
    else if (selectedUnit_cast_ing == 1 && selectedUnit_minusc_mayusc==0)
```

```
{
    tipo_letra=50;
    [self.view setNeedsDisplay];
}
else if (selectedUnit_cast_ing == 1 && selectedUnit_minusc_mayusc==1)
{
    tipo_letra=60;
    [self.view setNeedsDisplay];
}
}
/*
-(IBAction) PULSAR_BOTON OPCIONES: (id) sender
{
//NSUserDefaults *defaults = [NSUserDefaults standardUserDefaults];
//resp_negativa = [defaults objectForKey:@"Respuesta negativa"];

if (resp_negativa == NO)
{
[BOTON_AJUSTES setTitle:@"ON" forState:0];
resp_negativa = YES;

}
else
{
[BOTON_AJUSTES setTitle:@"OFF" forState:0];
resp_negativa = NO;
}
}
*/
- (IBAction)ajustes_personalizar_pictos
{
[controladorPersonalizacion viewDidLoad];
controladorPersonalizacion=[[Personalizacion alloc]
initWithNibName:@"Personalizacion" bundle:[NSBundle mainBundle]];

[self.view addSubview:controladorPersonalizacion.view];
}

- (NSString *)rutaFichero
{
    NSArray *rutas = NSSearchPathForDirectoriesInDomains(NSDocumentDirectory,
    NSUserDomainMask, YES);
    NSLog(@"Directorios: %@", rutas);

    NSString *directorio = [rutas objectAtIndex:0];
```

```
return [directorio stringByAppendingPathComponent:ficheroPersistente];
}

- (void) mostrarAjustes
{
    NSDictionary *item =[ajustes objectAtIndex:0];

    //se convierten los valores a números
    resp_negativa= [[item objectForKey:KEY_RESP_NEGATIVA] intValue];
    tiempo_pulsacion = [[item objectForKey:KEY_TIEMPO_PULSACION] intValue];
    tipo_letra = [[item objectForKey:KEY_TIPO_LETRA] intValue];

    int idioma;
    int tamaño_letra;

    if (tipo_letra<40)
    {
        idioma=0;
        label_resp_negativa.text=@"RESPUESTA NEGATIVA";
        label_tiempo_pulsacion.text=@"TIEMPO DE PULSADO (segundos)";
        label_tamaño_letra.text=@"TIPO DE LETRA";
        label_idioma.text=@"IDIOMA";
        label_pictos_personalizar.text=@"INTRODUCE TUS PICTOGRAMAS O
MODIFICA LOS EXISTENTES";
        label_album_favorito_personalizar.text=@"PERSONALIZA TU ÁLBUM DE
FAVORITOS";

        [segmentedControl_minusc_mayusc setTitle:@"minúsculas"
forSegmentAtIndex:0];
        [segmentedControl_minusc_mayusc setTitle:@"MAYÚSCULAS"
forSegmentAtIndex:1];
        [segmentedControl_cast_ing setTitle:@"Castellano" forSegmentAtIndex:0];
        [segmentedControl_cast_ing setTitle:@"Inglés" forSegmentAtIndex:1];

        [but_pictos_personalizar setTitle:@"Personalizar pictos" forState:normal];
        [but_album_favorito_personalizar setTitle:@"Personalizar álbum" forState:normal];
        [but_guardar_ajustes setTitle:@"GUARDAR" forState:normal];
        [but_inicio setTitle:@"INICIO" forState:normal];

    }
    else
    {
        idioma=1;
    }
}
```



```
label_resp_negativa.text=@"SET ACTIVE 'SAY NO";  
label_tiempo_pulsacion.text=@"TOUCHING TIME (seconds);  
label_tamano_letra.text=@"FONT TYPE";  
label_idioma.text=@"LANGUAGE";  
label_pictos_personalizar.text=@"ADD OR EDIT YOUR OWN PICTOS";  
label_album_favorito_personalizar.text=@"CHOOSE YOUR FAVOURITE ALBUM  
PICTURES";
```

```
[segmentedControl_minusc_mayusc setTitle:@"small letter"  
forSegmentAtIndex:0];  
[segmentedControl_minusc_mayusc setTitle:@"CAPITAL LETTER"  
forSegmentAtIndex:1];  
[segmentedControl_cast_ing setTitle:@"Spanish" forSegmentAtIndex:0];  
[segmentedControl_cast_ing setTitle:@"English" forSegmentAtIndex:1];  
  
[but_pictos_personalizar setTitle:@"Pictos personalize" forState:normal];  
[but_album_favorito_personalizar setTitle:@"Album personalize" forState:normal];  
[but_guardar_ajustes setTitle:@"SAVE" forState:normal];  
[but_inicio setTitle:@"HOME" forState:normal];
```

```
}  
if (tipo_letra==20 || tipo_letra==50)  
{  
    tamano_letra=0;  
}
```

```
else  
{  
    tamano_letra=1;  
}
```

```
NSLog(@"negativa %i, pulsacion %i, idioma %i, tamano letra %i", resp_negativa,  
tiempo_pulsacion, idioma, tamano_letra);
```

```
//se colocan los segmentos seleccionados correspondientes.  
[segmentedControl_resp_negativa setSelectedSegmentIndex:resp_negativa];  
[segmentedControl_tiempo_pulsacion  
setSelectedSegmentIndex:tiempo_pulsacion];  
[segmentedControl_cast_ing setSelectedSegmentIndex:idioma];  
[segmentedControl_minusc_mayusc setSelectedSegmentIndex:tamano_letra];  
}
```

```
- (void) guardarAjustes;  
{  
    if (selectedUnit_cast_ing == 0 && selectedUnit_minusc_mayusc==0)  
    {  
        tipo_letra=20;  
        [self.view setNeedsDisplay];  
    }  
}
```

```
    }
    else if (selectedUnit_cast_ing == 0 && selectedUnit_minusc_mayusc==1)
    {
        tipo_letra=30;
        [self.view setNeedsDisplay];
    }
    else if (selectedUnit_cast_ing == 1 && selectedUnit_minusc_mayusc==0)
    {
        tipo_letra=50;
        [self.view setNeedsDisplay];
    }
    else if (selectedUnit_cast_ing == 1 && selectedUnit_minusc_mayusc==1)
    {
        tipo_letra=60;
        [self.view setNeedsDisplay];
    }

    resp_negativa_string=[NSString stringWithFormat:@"%d",resp_negativa];
    tiempo_pulsacion_string=[NSString stringWithFormat:@"%d",tiempo_pulsacion];
    tipo_letra_string=[NSString stringWithFormat:@"%d",tipo_letra];

    NSLog(@"negativa %@, tipo letra %i",resp_negativa_string, tipo_letra);

    NSDictionary *item =[NSDictionary dictionaryWithObjectsAndKeys:tipo_letra_string,
    KEY_TIPO_LETRA, tiempo_pulsacion_string, KEY_TIEMPO_PULSACION,
    resp_negativa_string, KEY_RESP_NEGATIVA, nil];
    [self.ajustes removeAllObjects];

    [self.ajustes addObject:item];

    [ajustes writeToFile:[self rutaFichero] atomically:YES];

    [self mostrarAjustes];

}
- (IBAction)ajustes_personalizar_album_favorito
{
    EditarAlbumFavoritoVista=[[EditarAlbumFavorito alloc]
initWithNibName:@"EditarAlbumFavorito" bundle:[NSBundle mainBundle]];

    [self.view addSubview:EditarAlbumFavoritoVista.view];
}

- (IBAction) PULSAR_BOTON_SALIR_AJUSTES: (id) sender
```

```
{
    [self guardarAjustes];
    [self.view removeFromSuperview];
}

- (IBAction) PULSAR_BOTON_GUARDAR_AJUSTES: (id)sender
{
    [self guardarAjustes];
    if (tipo_letra<40) // IDIOMA CASTELLANO
    {

        UIAlertView *alertView = [[UIAlertView alloc] initWithTitle:@"ENHORABUENA"
        message:@"Sus ajustes han sido guardados correctamente" delegate:self
        cancelButtonTitle:nil otherButtonTitles:@"Ok", nil];
        [alertView show];
        [alertView release];
    }
    else //IDIOMA INGLÉS
    {
        UIAlertView *alertView = [[UIAlertView alloc]
        initWithTitle:@"CONGRATULATIONS" message:@"Your changes have been saved
        correctly" delegate:self cancelButtonTitle:nil otherButtonTitles:@"Ok", nil];
        [alertView show];
        [alertView release];
    }
}
@end
```

// Personalizacion.h

```
#import <CoreData/CoreData.h>
#import <UIKit/UIKit.h>

@class Ajustes;

@interface Personalizacion : UIViewController <UINavigationControllerDelegate,
UIImagePickerControllerDelegate> {

    Ajustes * controladorAjustes;

    IBOutlet UISwitch *switch_persona;
    IBOutlet UISwitch *switch_verbo;
    IBOutlet UISwitch *switch_complemento;
    IBOutlet UILabel *label_frase_guardado;
    IBOutlet UILabel *label_busqueda;
    IBOutlet UILabel *label_instrucciones;
    IBOutlet UILabel *label_borrar;

    IBOutlet UILabel *label_nombre_castellano;
    IBOutlet UILabel *label_nombre_ingles;
    IBOutlet UILabel *label_Persona;
    IBOutlet UILabel *label_Verbo;
    IBOutlet UILabel *label_Complemento;

    IBOutlet UITextField *texto_castellano;
    IBOutlet UITextField *texto_ingles;
    IBOutlet UITextField *texto_busqueda;
    IBOutlet UIImageView *VISTA_IMAGEN;

    IBOutlet UIButton *BOTON_BUSCAR;
    IBOutlet UIButton *BOTON_GUARDAR;
    IBOutlet UIButton *BOTON_BUSCAR_IMAGEN;
    IBOutlet UIButton *BOTON_BORRAR;

    IBOutlet UIButton *BOTON_REINICIAR_VALORES;
    IBOutlet UIButton *BOTON_ATRAS;

    NSString *entidad_personalizacion;

    UIPopoverController *popover;

    int existe;
    int existe_personas;
```

```
int existe_verbos;
int existe_complementos;

int fila_a_cambiar_post_busqueda_personas;
int fila_a_cambiar_post_busqueda_verbos;
int fila_a_cambiar_post_busqueda_complementos;
int Es_nuevo_personas;
int Es_nuevo_verbos;
int Es_nuevo_complementos;

int ajuste_resp_negativa;
int ajuste_tiempo_pulsacion;
int ajuste_tipo_letra;
}

@property(nonatomic,retain)Ajustes * controladorAjustes;

@property (nonatomic, retain) UISwitch *switch_persona;
@property (nonatomic, retain) UISwitch *switch_verbo;
@property (nonatomic, retain) UISwitch *switch_complemento;
@property (nonatomic,retain) IBOutlet UILabel *label_frase_guardado;
@property (nonatomic, retain) UIPopoverController *popover;
@property (nonatomic, retain) UIImage *copyImage;

@property (nonatomic, retain)IBOutlet UILabel *label_busqueda;
@property (nonatomic, retain)IBOutlet UILabel *label_instrucciones;
@property (nonatomic, retain)IBOutlet UILabel *label_borrar;

@property (nonatomic, retain)IBOutlet UILabel *label_nombre_castellano;
@property (nonatomic, retain)IBOutlet UILabel *label_nombre_ingles;
@property (nonatomic, retain)IBOutlet UILabel *label_Persona;
@property (nonatomic, retain)IBOutlet UILabel *label_Verbo;
@property (nonatomic, retain)IBOutlet UILabel *label_Complemento;

@property (nonatomic, retain)IBOutlet UIButton *BOTON_BUSCAR;
@property (nonatomic, retain)IBOutlet UIButton *BOTON_GUARDAR;
@property (nonatomic, retain)IBOutlet UIButton *BOTON_BUSCAR_IMAGEN;
@property (nonatomic, retain)IBOutlet UIButton *BOTON_BORRAR;

@property (nonatomic, retain)IBOutlet UIButton *BOTON_REINICIAR_VALORES;
@property (nonatomic, retain)IBOutlet UIButton *BOTON_ATRAS;

- (IBAction) PULSAR_VOLVER_PERSONALIZACION: (id) sender;
```

- (IBAction) PULSAR_BUSCAR: (id) sender;
- (IBAction) PULSAR_REINICIAR_VALORES: (id) sender;
- (IBAction) GUARDAR_NUEVA_ENTRADA: (id) sender;

- (void) EliminoObjetosEntidad: (NSString *) entidad;
- (void) ModificoColumnasActivo: (NSString *) entidad;
- (void) ModificoColumnasContador: (NSString *) entidad;
- (UIImage*)imageWithImage:(UIImage*)image scaledToSize:(CGSize)newSize;
- (void) Nuevo_elemento: (NSString *) entidad_a_guardar Fila_cambia: (int) fila Indice: (int) indice;
- (IBAction) EliminarObjeto: (id) sender;
- (void) funcion_de_eliminar_objeto: (NSString *) entidad fila: (int) fila_cambio Variable_nuevo: (int) es_nuevo;

- (IBAction)choosePic: (id) sender;

- (void)saveImage:(UIImage*)image:(NSString*)imageName;
- (void)removeImage:(NSString*)fileName;
- (UIImage*)loadImage:(NSString*)imageName;

- (void) DeshabilitarElementos;
- (void) HabilitarElementos;

- @end

// Personalizacion.m

```
#import "COMUNICADORAppDelegate.h"
```

```
#import "Personalizacion.h"
```

```
#import "Ajustes.h"
```

```
@implementation Personalizacion
```

```
@synthesize controladorAjustes;
```

```
@synthesize switch_persona;
```

```
@synthesize switch_verbo;
```

```
@synthesize switch_complemento;
```

```
@synthesize popover;
```

```
@synthesize copyImage;
```

```
@synthesize label_frase_guardado;
```

```
@synthesize label_busqueda;
```

```
@synthesize label_instrucciones;
```

```
@synthesize label_borrar;
```

```
@synthesize label_nombre_castellano;
```

```
@synthesize label_nombre_ingles;
```

```
@synthesize label_Persona;
```

```
@synthesize label_Verbo;
```

```
@synthesize label_Complemento;
```

```
@synthesize BOTON_BUSCAR;
```

```
@synthesize BOTON_GUARDAR;
```

```
@synthesize BOTON_BUSCAR_IMAGEN;
```

```
@synthesize BOTON_BORRAR;
```

```
@synthesize BOTON_REINICIAR_VALORES;
```

```
@synthesize BOTON_ATRAS;
```

```
- (void)Ajustar
```

```
{
```

```
    [controladorAjustes viewDidLoad];
```

```
    ajuste_resp_negativa=controladorAjustes.resp_negativa;
```

```
    ajuste_tiempo_pulsacion=controladorAjustes.tiempo_pulsacion;
```

```
    ajuste_tipo_letra=controladorAjustes.tipo_letra;
```

```
    NSLog(@"respuesta negativa %i, tiempo pulsacion %i, tipo letra %i",ajuste_resp_negativa,ajuste_tiempo_pulsacion,ajuste_tipo_letra);
```

```
}

- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil
{
    self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil];
    if (self) {
        // Custom initialization
    }
    return self;
}

- (void)didReceiveMemoryWarning
{
    // Releases the view if it doesn't have a superview.
    [super didReceiveMemoryWarning];

    // Release any cached data, images, etc that aren't in use.
}

#pragma mark - View lifecycle

- (void)viewDidLoad
{
    NSLog(@"estoy entrando en personalizacion");

    controladorAjustes = [[Ajustes alloc] init];

    [self Ajustar];

    if (ajuste_tipo_letra < 40) // IDIOMA CASTELLANO
    {
        label_instrucciones.text = @"1.- Introduce el elemento a modificar o el elemento
nuevo a guardar.";
        label_nombre_castellano.text = @"NOMBRE (Castellano)";
        label_nombre_ingles.text = @"NOMBRE (Inglés)";
        label_Persona.text = @"PERSONA";
        label_Verbo.text = @"VERBO";
        label_Complemento.text = @"COMPLEMENTO";

        [BOTON_BUSCAR_IMAGEN setTitle:@"SELECCIONA IMAGEN"
forState:normal];
        [BOTON_GUARDAR setTitle:@"GUARDAR" forState:normal];
        [BOTON_BUSCAR setTitle:@"BUSCAR" forState:normal];
        [BOTON_BORRAR setTitle:@"BORRAR" forState:normal];
    }
}
```



```
[BOTON_REINICIAR_VALORES setTitle:@"REINICIAR VALORES"
forState:normal];
[BOTON_ATRAS setTitle:@"ATRÁS" forState:normal];
}
else //IDIOMA INGLÉS
{

    label_instrucciones.text=@"1.- Introduce a new element to store or modify an
existing one.";
    label_nombre_castellano.text=@"NAME (Spanish)";
    label_nombre_ingles.text=@"NAME (English)";
    label_Persona.text=@"PERSON";
    label_Verbo.text=@"VERB";
    label_Complemento.text=@"COMPLEMENT";

    [BOTON_BUSCAR_IMAGEN setTitle:@"CHOOSE IMAGE" forState:normal];
    [BOTON_GUARDAR setTitle:@"SAVE" forState:normal];
    [BOTON_BUSCAR setTitle:@"SEARCH" forState:normal];
    [BOTON_BORRAR setTitle:@"DELETE" forState:normal];
    [BOTON_REINICIAR_VALORES setTitle:@"RESTORE VALUES"
forState:normal];
    [BOTON_ATRAS setTitle:@"BACK" forState:normal];
}

[self DeshabilitarElementos];
label_instrucciones.hidden=NO;

[super viewDidLoad];
// Do any additional setup after loading the view from its nib.
}

- (void)viewDidUnload
{
    [super viewDidUnload];
    // Release any retained subviews of the main view.
    // e.g. self.myOutlet = nil;
}

- (BOOL)shouldAutorotateToInterfaceOrientation:
(UIInterfaceOrientation)interfaceOrientation
{
    // Return YES for supported orientations
    return YES;
}
```

//Busco elementos en todas las entidades (Persona, verbo y complemento) y activo los switches

- (IBAction) PULSAR_BUSCAR: (id) sender{

```
switch_persona.on=NO;  
switch_verbo.on=NO;  
switch_complemento.on=NO;  
label_frase_guardado.text=nil;
```

```
existe=1;  
existe_personas=1;  
existe_verbos=1;  
existe_complementos=1;
```

```
Es_nuevo_personas=0;  
Es_nuevo_verbos=0;  
Es_nuevo_complementos=0;
```

```
COMUNICADORAppDelegate *appDelegate = [[UIApplication sharedApplication]  
delegate];
```

```
NSManagedObjectContext *context = [appDelegate managedObjectContext];
```

```
NSEntityDescription *entityDesc = [NSEntityDescription  
entityForName:@"Personas" inManagedObjectContext:context];
```

```
NSFetchRequest *request = [[NSFetchRequest alloc] init];
```

```
[request setEntity:entityDesc];
```

```
NSPredicate *pred = [NSPredicate predicateWithFormat:@"(nombre = %@ AND  
activo=1)", texto_busqueda.text];
```

```
NSError *error;
```

```
[request setPredicate:pred];
```

```
NSArray *objects = [context executeFetchRequest:request error:&error];
```

```
if ([objects count] == 0) {  
    existe=existe*1;  
    VISTA_IMAGEN.image=nil;  
}
```

```
else if ([objects count] == 1){
```

```
existe=existe*0;
existe_personas=0;
NSManagedObject *matches = [objects objectAtIndex:0];
texto_castellano.text=[matches valueForKey:@"nombre"];
texto_ingles.text=[matches valueForKey:@"nombre_ing"];
fila_a_cambiar_post_busqueda_personas=[[matches
valueForKey:@"pk"]intValue];
Es_nuevo_personas=[[matches valueForKey:@"nuevo"]intValue];
switch_persona.on=YES;
NSLog(@"fila %i", fila_a_cambiar_post_busqueda_personas);
}

entityDesc= [NSEntityDescription entityForName:@"Verbos"
inManagedObjectContext:context];
[request setEntity:entityDesc];
[request setPredicate:pred];

objects = [context executeFetchRequest:request error:&error];

if ([objects count] == 0) {
    existe=existe*1;
    VISTA_IMAGEN.image=nil;
}

else if ([objects count] == 1){

    existe=existe*0;
    existe_verbos=0;
    NSManagedObject *matches = [objects objectAtIndex:0];
    texto_castellano.text=[matches valueForKey:@"nombre"];
    texto_ingles.text=[matches valueForKey:@"nombre_ing"];
    fila_a_cambiar_post_busqueda_verbos=[[matches valueForKey:@"pk"]intValue];
    Es_nuevo_verbos=[[matches valueForKey:@"nuevo"]intValue];

    switch_verbo.on=YES;
}

entityDesc= [NSEntityDescription entityForName:@"Complementos"
inManagedObjectContext:context];
[request setEntity:entityDesc];
[request setPredicate:pred];
objects = [context executeFetchRequest:request error:&error];

if ([objects count] == 0) {
```

```
    existe=existe*1;
    VISTA_IMAGEN.image=nil;
}

else if ([objects count] == 1){

    existe=existe*0;
    existe_complementos=0;
    NSManagedObject *matches = [objects objectAtIndex:0];
    texto_castellano.text=[matches valueForKey:@"nombre"];
    texto_ingles.text=[matches valueForKey:@"nombre_ing"];
    fila_a_cambiar_post_busqueda_complementos=[[matches
valueForKey:@"pk"]intValue];
    Es_nuevo_complementos=[[matches valueForKey:@"nuevo"]intValue];

    switch_complemento.on=YES;
}
if (existe==1){

    if (ajuste_tipo_letra<40) // IDIOMA CASTELLANO

    {
        label_busqueda.text = @"Ese valor no existe. Rellene los parametros inferiores
para crearlo";
    }
    else //IDIOMA INGLÉS
    {
        label_busqueda.text = @"This value don't exist. Please introduce the following
values to create it";
    }

    label_instrucciones.hidden=YES;
    [self HabilitarElementos];
    texto_castellano.text=texto_busqueda.text;
    texto_busqueda.text=nil;
    texto_castellano.enabled=NO;
    texto_ingles.text=nil;
    BOTON_BORRAR.enabled=NO;

}
else if (existe==0){

    if (ajuste_tipo_letra<40) // IDIOMA CASTELLANO
    {
        label_busqueda.text = @"Ese nombre ya existe. Modifique los parametros
inferiores y pulse guardar";
    }
}
```

```
        label_borrar.text = @"Si desea eliminarlo, pulse BORRAR";
    }
    else //IDIOMA INGLÉS
    {
        label_busqueda.text = @"This element already exists. Please introduce new
values and touch 'SAVE';
        label_borrar.text = @"If you want to delete it, please touch 'DELETE';
    }

    label_instrucciones.hidden=YES;
    [self HabilitarElementos];
    texto_busqueda.text=nil;
    texto_castellano.enabled=NO;
    UIImage *btnImage;
    if (Es_nuevo_personas==1 || Es_nuevo_verbos==1 ||
Es_nuevo_complementos==1) {

        btnImage = [self loadImage:texto_castellano.text];
    }
    else{
        btnImage = [UIImage imageNamed:texto_castellano.text];
    }
    VISTA_IMAGEN.image=btnImage;
}
}
```

//Guardo los elementos en la entidad correspondiente en función del switch activado
- (IBAction) GUARDAR_NUEVA_ENTRADA: (id) sender{

```
    if (switch_persona.on==YES){

        [self Nuevo_elemento:@"Personas"
Fila_cambia:fila_a_cambiar_post_busqueda_personas Indice:1];
    }
    if (switch_verbo.on==YES){

        [self Nuevo_elemento:@"Verbos"
Fila_cambia:fila_a_cambiar_post_busqueda_verbos Indice:2];
    }
    if (switch_complemento.on==YES){
```

```
[self Nuevo_elemento:@"Complementos"  
Fila_cambia:fila_a_cambiar_post_busqueda_complementos Indice:3];  
  
}  
  
UIImage *imagen=VISTA_IMAGEN.image;  
  
/*  
//Con esto, salvo en el álbum  
  
UIImageWriteToSavedPhotosAlbum(VISTA_IMAGEN.image, nil, nil, nil);  
*/  
[self saveImage: imagen: texto_castellano.text];  
//UIImage *myUIImage = [self loadImage: @"myUIImageName"];  
  
//VISTA_IMAGEN2.image= myUIImage;  
  
if (ajuste_tipo_letra<40) // IDIOMA CASTELLANO  
{  
    label_frase_guardado.text=@"Nueva entrada guardada";  
  
    UIAlertView *alertView = [[UIAlertView alloc] initWithTitle:@"ENHORABUENA"  
message:@"Su elemento ha sido guardado" delegate:self cancelButtonTitle:nil  
otherButtonTitles:@"Ok", nil];  
  
    [alertView show];  
    [alertView release];  
}  
else //IDIOMA INGLÉS  
{  
    label_frase_guardado.text=@"ENTRY SAVED";  
  
    UIAlertView *alertView = [[UIAlertView alloc]  
initWithTitle:@"CONGRATULATIONS" message:@"Your element has been saved"  
delegate:self cancelButtonTitle:nil otherButtonTitles:@"Ok", nil];  
  
    [alertView show];  
    [alertView release];  
}  
  
[self DeshabilitarElementos];  
label_borrar.text=nil;  
label_busqueda.text=nil;
```

```
label_instrucciones.hidden=NO;

}

//Guardo un nuevo elemento con sus respectivas relaciones a todos los sublugares
- (void) Nuevo_elemento: (NSString *) entidad_a_guardar Fila_cambia: (int) fila Indice:
(int) indice{

    COMUNICADORAppDelegate *appDelegate = [[UIApplication sharedApplication]
delegate];
    NSManagedObjectContext *context = [appDelegate managedObjectContext];

    NSEntityDescription *entityDesc = [NSEntityDescription entityForName:@"Sublugar"
inManagedObjectContext:context];

    NSFetchRequest *request = [[NSFetchRequest alloc] init];

    [request setEntity:entityDesc];

    NSError *error;

    NSArray *items=[context executeFetchRequest:request error:&error];

    [request release];

    NSManagedObjectContext *context1 = [appDelegate managedObjectContext];

    NSEntityDescription *entityDesc1 = [NSEntityDescription
entityForName:entidad_a_guardar inManagedObjectContext:context];

    NSFetchRequest *request1 = [[NSFetchRequest alloc] init];

    [request1 setEntity:entityDesc1];

    NSArray *items1=[context executeFetchRequest:request1 error:&error];

    [request1 release];

    NSError *error1;
```

```

    NSManagedObject *mo3 = [NSEntityDescription
insertNewObjectForEntityForName:entidad_a_guardar
inManagedObjectContext:context1];

[mo3 setValue:texto_castellano.text forKey:@"nombre"];
[mo3 setValue:texto_ingles.text forKey:@"nombre_ing"];
NSString *cambio_mayusculas;
cambio_mayusculas=[texto_castellano.text uppercaseString];
[mo3 setValue:cambio_mayusculas forKey:@"nombre_may"];
cambio_mayusculas=[texto_ingles.text uppercaseString];
[mo3 setValue:cambio_mayusculas forKey:@"nombre_may_ing"];

[mo3 setValue:[NSNumber numberWithInt:([items1 count]+1)] forKey:@"pk"];
[mo3 setValue:[NSNumber numberWithInt: 1] forKey:@"nuevo"];
[mo3 setValue:[NSNumber numberWithInt:0] forKey:@"album_personalizado"];
[mo3 setValue:[NSNumber numberWithInt: 1] forKey:@"activo"];

if (indice==1) {

if (existe_personas==0){

    NSManagedObject *fila_que_cambia= [items1 objectAtIndex:(fila-1)];

[fila_que_cambia setValue:[NSNumber numberWithInt: 0] forKey:@"activo"];
    existe_personas=1;

    //[mo3 setValue:[NSNumber numberWithInt:([items1 count]+1)] forKey:@"activo"];
}
}

else if (indice==2) {

if (existe_verbos==0){

    NSManagedObject *fila_que_cambia= [items1 objectAtIndex:(fila-1)];

[fila_que_cambia setValue:[NSNumber numberWithInt: 0] forKey:@"activo"];
    existe_verbos=1;

    //[mo3 setValue:[NSNumber numberWithInt:([items1 count]+1)] forKey:@"activo"];
}
}

else if (indice==3){
if (existe_complementos==0){

    NSManagedObject *fila_que_cambia= [items1 objectAtIndex:(fila-1)];

```



```
[fila_que_cambia setValue:[NSNumber numberWithInt: 0] forKey:@"activo"];
existe_complementos=1;

//[mo3 setValue:[NSNumber numberWithInt:([items1 count]+1)] forKey:@"activo"];
}
}
```

```
NSMutableArray *mo= [items objectAtIndex:0];
[[mo3 mutableSetValueForKey:@"FraseSublugar"] addObject:mo];
mo= [items objectAtIndex:1];
[[mo3 mutableSetValueForKey:@"FraseSublugar"] addObject:mo];
mo= [items objectAtIndex:2];
[[mo3 mutableSetValueForKey:@"FraseSublugar"] addObject:mo];
mo= [items objectAtIndex:3];
[[mo3 mutableSetValueForKey:@"FraseSublugar"] addObject:mo];
mo= [items objectAtIndex:4];
[[mo3 mutableSetValueForKey:@"FraseSublugar"] addObject:mo];
mo= [items objectAtIndex:5];
[[mo3 mutableSetValueForKey:@"FraseSublugar"] addObject:mo];
mo= [items objectAtIndex:6];
[[mo3 mutableSetValueForKey:@"FraseSublugar"] addObject:mo];
mo= [items objectAtIndex:7];
[[mo3 mutableSetValueForKey:@"FraseSublugar"] addObject:mo];
mo= [items objectAtIndex:8];
[[mo3 mutableSetValueForKey:@"FraseSublugar"] addObject:mo];
mo= [items objectAtIndex:9];
[[mo3 mutableSetValueForKey:@"FraseSublugar"] addObject:mo];
mo= [items objectAtIndex:10];
[[mo3 mutableSetValueForKey:@"FraseSublugar"] addObject:mo];
mo= [items objectAtIndex:11];
[[mo3 mutableSetValueForKey:@"FraseSublugar"] addObject:mo];
mo= [items objectAtIndex:12];
[[mo3 mutableSetValueForKey:@"FraseSublugar"] addObject:mo];

mo= [items objectAtIndex:13];
[[mo3 mutableSetValueForKey:@"FraseSublugar"] addObject:mo];
mo= [items objectAtIndex:14];
[[mo3 mutableSetValueForKey:@"FraseSublugar"] addObject:mo];
mo= [items objectAtIndex:15];
[[mo3 mutableSetValueForKey:@"FraseSublugar"] addObject:mo];
mo= [items objectAtIndex:16];
[[mo3 mutableSetValueForKey:@"FraseSublugar"] addObject:mo];
mo= [items objectAtIndex:17];
[[mo3 mutableSetValueForKey:@"FraseSublugar"] addObject:mo];
mo= [items objectAtIndex:18];
[[mo3 mutableSetValueForKey:@"FraseSublugar"] addObject:mo];
```

```
mo= [items objectAtIndex:19];
[[mo mutableSetValueForKey:@"FraseSublugar"] addObject:mo];
mo= [items objectAtIndex:20];
[[mo mutableSetValueForKey:@"FraseSublugar"] addObject:mo];
mo= [items objectAtIndex:21];
[[mo mutableSetValueForKey:@"FraseSublugar"] addObject:mo];
mo= [items objectAtIndex:22];
[[mo mutableSetValueForKey:@"FraseSublugar"] addObject:mo];

[context refreshObject:mo mergeChanges:YES];

NSLog(@"He guardado");

if (![context save:&error]) {
    NSLog(@"error de coredata %@, %@", error, [error userInfo]);
    exit(-1);
}

if (![context1 save:&error1]) {
    NSLog(@"error de coredata %@, %@", error1, [error1 userInfo]);
    exit(-1);
}

NSLog(@"LLeego");

}

//Método del boton "reiniciar valores" que llama a los métodos
//EliminoObjetoEntidad, ModificoColumnasActivo y ModificoColumnasContador en
todas las entidades

- (IBAction) PULSAR_REINICIAR_VALORES: (id) sender{

    [self EliminoObjetosEntidad:@"Personas"];
    [self ModificoColumnasActivo:@"Personas"];
    [self ModificoColumnasContador:@"Personas"];

    [self EliminoObjetosEntidad:@"Verbos"];
    [self ModificoColumnasActivo:@"Verbos"];
    [self ModificoColumnasContador:@"Verbos"];
}
```

```
[self EliminoObjetosEntidad:@"Complementos"];
[self ModificoColumnasActivo:@"Complementos"];
[self ModificoColumnasContador:@"Complementos"];

if (ajuste_tipo_letra<40) // IDIOMA CASTELLANO
{
    label_frase_guardado.text=@"Se han restaurado los valores iniciales";
    UIAlertView *alertView = [[UIAlertView alloc] initWithTitle:@"ENHORABUENA"
message:@"Se han restaurado los valores iniciales" delegate:self cancelButtonTitle:nil
otherButtonTitles:@"Ok", nil];
    [alertView show];
    [alertView release];
}
else //IDIOMA INGLÉS
{
    label_frase_guardado.text=@"All values have been restored";

    UIAlertView *alertView = [[UIAlertView alloc]
initWithTitle:@"CONGRATULATIONS" message:@"All values have been restored"
delegate:self cancelButtonTitle:nil otherButtonTitles:@"Ok", nil];

    [alertView show];
    [alertView release];
}
}
```

//Método que elimina todos los elementos de la entidad "entidad" que tengan un 1 en la columna "nuevo"

```
- (void) EliminoObjetosEntidad: (NSString *) entidad{
```

```
    COMUNICADORAppDelegate *appDelegate = [[UIApplication sharedApplication]
delegate];
```

```
    NSManagedObjectContext *context = [appDelegate managedObjectContext];
```

```
    NSEntityDescription *entityDesc = [NSEntityDescription entityForName:entidad
inManagedObjectContext:context];
```

```
    NSFetchedRequest *request = [[NSFetchedRequest alloc] init];
```

```
[request setEntity:entityDesc];

NSPredicate *pred = [NSPredicate predicateWithFormat:@"ANY nuevo=1"];

[request setPredicate:pred];

NSError *error;

NSArray *items=[context executeFetchRequest:request error:&error];

[request release];

if ([items count]==0){
    NSLog(@"no hay elementos");
}

else{

    NSManagedObject *objeto_que_eliminar= [items objectAtIndex:0];
    [context deleteObject:objeto_que_eliminar];
    [self EliminoObjetosEntidad:entidad];

}

if (![context save:&error]) {
    NSLog(@"error de coredata %@, %@", error, [error userInfo]);
    exit(-1);
}

}

//Método que pone la columna activo igual a 1 en todos los elementos.
- (void) ModificoColumnasActivo: (NSString *) entidad{

    COMUNICADORAppDelegate *appDelegate = [[UIApplication sharedApplication]
delegate];
    NSManagedObjectContext *context = [appDelegate managedObjectContext];

    NSEntityDescription *entityDesc = [NSEntityDescription entityForName:entidad
inManagedObjectContext:context];

    NSFetchedRequest *request = [[NSFetchRequest alloc] init];
```

```
[request setEntity:entityDesc];

NSPredicate *pred = [NSPredicate predicateWithFormat:@"ANY activo=0"];

[request setPredicate:pred];

NSError *error;

NSArray *items=[context executeFetchRequest:request error:&error];

[request release];

if ([items count]==0){
    NSLog(@"no hay elementos");
    if (![context save:&error]) {
        NSLog(@"error de coredata %@, %@", error, [error userInfo]);
        exit(-1);
    }
}

else{

    NSManagedObject *objeto_que_cambiar= [items objectAtIndex:0];
    [objeto_que_cambiar setValue:[NSNumber numberWithInt: 1] forKey:@"activo"];
    if (![context save:&error]) {
        NSLog(@"error de coredata %@, %@", error, [error userInfo]);
        exit(-1);
    }
    [self ModificoColumnasActivo:entidad];

}

}

//Método que pone la columna contador igual a 0 en todos los elementos.
- (void) ModificoColumnasContador: (NSString *) entidad{

    COMUNICADORAppDelegate *appDelegate = [[UIApplication sharedApplication]
delegate];
    NSManagedObjectContext *context = [appDelegate managedObjectContext];

    NSEntityDescription *entityDesc = [NSEntityDescription entityForName:entidad
```

```
inManagedObjectContext:context];

NSFetchRequest *request = [[NSFetchRequest alloc] init];

[request setEntity:entityDesc];

NSPredicate *pred = [NSPredicate predicateWithFormat:@"ANY contador!=0"];

[request setPredicate:pred];

NSError *error;

NSArray *items=[context executeFetchRequest:request error:&error];

[request release];

if ([items count]==0){
    NSLog(@"no hay elementos");
    if (![context save:&error]) {
        NSLog(@"error de coredata %@, %@", error, [error userInfo]);
        exit(-1);
    }
}

else{

    NSManagedObject *objeto_que_cambiar= [items objectAtIndex:0];
    [objeto_que_cambiar setValue:[NSNumber numberWithInt: 0]
    forKey:@"contador"];
    if (![context save:&error]) {
        NSLog(@"error de coredata %@, %@", error, [error userInfo]);
        exit(-1);
    }
    [self ModificoColumnasContador:entidad];

}

}

//Elimino el objeto buscado
- (IBAction)EliminarObjeto: (id) sender{

    if (existe_personas==0){
```

```
[self funcion_de_eliminar_objeto:@"Personas" fila:
fila_a_cambiar_post_busqueda_personas Variable_nuevo:Es_nuevo_personas];
}

if (existe_verbos==0){
    [self funcion_de_eliminar_objeto:@"Verbos" fila:
fila_a_cambiar_post_busqueda_verbos Variable_nuevo:Es_nuevo_verbos];
}

if (existe_complementos==0){
    [self funcion_de_eliminar_objeto:@"Complementos" fila:
fila_a_cambiar_post_busqueda_complementos
Variable_nuevo:Es_nuevo_complementos];
}

texto_castellano.text=nil;
texto_ingles.text=nil;
texto_busqueda.text=nil;
VISTA_IMAGEN.image=nil;
switch_persona.on=NO;
switch_verbo.on=NO;
switch_complemento.on=NO;

if (ajuste_tipo_letra<40) // IDIOMA CASTELLANO
{
    label_frase_guardado.text=@"Elemento borrado";
    UIAlertView *alertView = [[UIAlertView alloc] initWithTitle:@"ENHORABUENA"
message:@"El elemento ha sido borrado" delegate:self cancelButtonTitle:nil
otherButtonTitles:@"Ok", nil];
    [alertView show];
    [alertView release];
}
else //IDIOMA INGLÉS
{
    label_frase_guardado.text=@"Element deleted";
    UIAlertView *alertView = [[UIAlertView alloc]
initWithTitle:@"CONGRATULATIONS" message:@"The element has been deleted"
delegate:self cancelButtonTitle:nil otherButtonTitles:@"Ok", nil];
    [alertView show];
    [alertView release];
}
```

```
}

label_borrar.text=nil;
label_busqueda.text=nil;

label_instrucciones.hidden=NO;
[self DeshabilitarElementos];

}

//funcion que elimina el objeto
- (void) funcion_de_eliminar_objeto: (NSString *) entidad fila: (int) fila_cambio
Variable_nuevo: (int) es_nuevo{

    COMUNICADORAppDelegate *appDelegate = [[UIApplication sharedApplication]
delegate];
    NSManagedObjectContext *context = [appDelegate managedObjectContext];

    NSEntityDescription *entityDesc = [NSEntityDescription entityForName:entidad
inManagedObjectContext:context];

    NSFetchRequest *request = [[NSFetchRequest alloc] init];

    [request setEntity:entityDesc];

    //NSPredicate *pred = [NSPredicate predicateWithFormat:@"ANY nuevo=1"];

    //[request setPredicate:pred];

    NSError *error;

    NSArray *items=[context executeFetchRequest:request error:&error];

    [request release];

    NSManagedObject *fila_que_eliminar= [items objectAtIndex:(fila_cambio-1)];

    if (es_nuevo==1){

        [context deleteObject:fila_que_eliminar];

        [self removeImage:texto_castellano.text];
    }
}
```



```
}
else{
    [fila_que_eliminar setValue:[NSNumber numberWithInt: 0] forKey:@"activo"];
}

if (![context save:&error]) {
    NSLog(@"error de coredata %@, %@", error, [error userInfo]);

exit(-1);
}

}

/*

save -->[self saveImage: myUIImage: @"myUIImageName"];
load --> myUIImage = [self loadImage: @"myUIImageName"];
remove --> [self removeImage: @"myUIImageName"];

*/

- (void)saveImage:(UIImage*)image:(NSString*)imageName {

    NSData *imageData = UIImagePNGRepresentation(image); //convert image into
.png format.

    NSFileManager *fileManager = [NSFileManager defaultManager];//create instance of
NSFileManager

    NSArray *paths = NSSearchPathForDirectoriesInDomains(NSDocumentDirectory,
NSUserDomainMask, YES); //create an array and store result of our search for the
documents directory in it

    NSString *documentsDirectory = [paths objectAtIndex:0]; //create NSString object,
that holds our exact path to the documents directory

    NSString *fullPath = [documentsDirectory stringByAppendingPathComponent:
[NSString stringWithFormat:@"%@.png", imageName]]; //add our image to the path

    [fileManager createFileAtPath:fullPath contents:imageData attributes:nil]; //finally
save the path (image)
```

```
    NSLog(@"image saved");
}

- (void)removeImage:(NSString*)fileName {

    NSFileManager *fileManager = [NSFileManager defaultManager];

    NSArray *paths = NSSearchPathForDirectoriesInDomains(NSDocumentDirectory,
    NSUserDomainMask, YES);

    NSString *documentsDirectory = [paths objectAtIndex:0];

    NSString *fullPath = [documentsDirectory stringByAppendingPathComponent:
    [NSString stringWithFormat:@"%s.png", fileName]];

    [fileManager removeItemAtPath: fullPath error:NULL];

    NSLog(@"image removed");
}

- (UIImage*)loadImage:(NSString*)imageName {

    NSArray *paths = NSSearchPathForDirectoriesInDomains(NSDocumentDirectory,
    NSUserDomainMask, YES);

    NSString *documentsDirectory = [paths objectAtIndex:0];

    NSString *fullPath = [documentsDirectory stringByAppendingPathComponent:
    [NSString stringWithFormat:@"%s.png", imageName]];

    return [UIImage imageWithContentsOfFile:fullPath];
}

- (UIImage*)imageWithImage:(UIImage*)image scaledToSize:(CGSize)newSize
{
    UIGraphicsBeginImageContext( newSize );
    [image drawInRect:CGRectMake(0,0,newSize.width,newSize.height)];
    UIImage* newImage = UIGraphicsGetImageFromCurrentImageContext();
    UIGraphicsEndImageContext();

    return newImage;
}
```

```
- (IBAction) PULSAR_VOLVER_PERSONALIZACION: (id) sender{

    [self.view removeFromSuperview];
    switch_persona.on=NO;
    switch_verbo.on=NO;
    switch_complemento.on=NO;
    texto_castellano.text=nil;
    texto_ingles.text=nil;
    texto_busqueda.text=nil;
    label_busqueda.text=nil;
    [self DeshabilitarElementos];
    label_instrucciones.hidden=NO;

    label_frase_guardado.text=nil;
    VISTA_IMAGEN.image=nil;
    label_borrar.text=nil;

}

-(void)exitProg:(UIImage *)image didFinishSavingWithError:(NSError *)error
contextInfo:(void *)contextInfo {

    if (ajuste_tipo_letra<40) // IDIOMA CASTELLANO
    {UIAlertView *alertView = [[UIAlertView alloc] initWithTitle:@"ENHORABUENA"
message:@"Su imagen ha sido guardada" delegate:self cancelButtonTitle:nil
otherButtonTitles:@"Ok", nil];
        [alertView show];
        [alertView release];
    }
    else //IDIOMA INGLÉS
    {
        UIAlertView *alertView = [[UIAlertView alloc]
initWithTitle:@"CONGRATULATIONS" message:@"Image has been saved"
delegate:self cancelButtonTitle:nil otherButtonTitles:@"Ok", nil];
        [alertView show];
        [alertView release];
    }
}
```

```
}
```

```
- (IBAction)choosePic: (id) sender {  
    UIImagePickerController *imagePicker = [[UIImagePickerController alloc] init];  
    imagePicker.delegate = self;  
    imagePicker.allowsEditing = NO;  
    imagePicker.sourceType = UIImagePickerControllerSourceTypePhotoLibrary;
```

```
    popover = [[UIPopoverController alloc]  
        initWithContentViewController:imagePicker];  
    [imagePicker release];  
    UIButton *btnSender = sender;  
    CGRect r = btnSender.frame;
```

```
    [popover presentPopoverFromRect:r inView:self.view  
        permittedArrowDirections:UIPopoverArrowDirectionAny animated:YES];
```

```
}
```

```
- (void)imagePickerController:(UIImagePickerController *)picker_pizarra  
didFinishPickingImage:(UIImage *)image editingInfo:(NSDictionary *)editingInfo {
```

```
    VISTA_IMAGEN.image=[self imageWithImage:image  
        scaledToSize:CGSizeMake(190,190)];
```

```
    [picker_pizarra release];  
    [self.popover dismissPopoverAnimated:YES];
```

```
}
```

```
- (void)imagePickerControllerDidCancel:(UIImagePickerController *)picker_pizarra {
```

```
    [picker_pizarra release];  
    [self.popover dismissPopoverAnimated:YES];
```

```
}
```

```
- (void) DeshabilitarElementos{
```

```
texto_castellano.enabled=NO;  
texto_ingles.enabled=NO;  
switch_persona.enabled=NO;  
switch_verbo.enabled=NO;  
switch_complemento.enabled=NO;  
BOTON_GUARDAR.enabled=NO;  
BOTON_BUSCAR_IMAGEN.enabled=NO;  
BOTON_BORRAR.enabled=NO;  
  
}
```

- (void) HabilitarElementos{

```
texto_castellano.enabled=YES;  
texto_ingles.enabled=YES;  
switch_persona.enabled=YES;  
switch_verbo.enabled=YES;  
switch_complemento.enabled=YES;  
  
BOTON_GUARDAR.enabled=YES;  
BOTON_BUSCAR_IMAGEN.enabled=YES;  
BOTON_BORRAR.enabled=YES;  
  
}
```

- (void) dealloc

```
{  
    [switch_persona release];  
    [switch_verbo release];  
    [switch_complemento release];  
    [label_frase_guardado release];  
    [copyImage release];  
    [super dealloc];  
}
```

@end

