

---

# SERVICIO ALTERNATIVO Y AUMENTATIVO DE MENSAJERIA INSTANTANEA

---

Proyecto final de carrera  
Ingeniería en Sistemas de Información

Bosquiazzo, Darío Andrés

Julio 2016

**Directora**

Dra. Gutiérrez, María de los Milagros

## Tabla de contenido

<b>Capítulo 1</b> .....	<b>3</b>
<b>Introducción</b> .....	<b>3</b>
1.1. <i>Fundamentos del proyecto</i> .....	4
1.2. <i>Objetivo General</i> .....	5
1.3. <i>Objetivos Específicos</i> .....	6
<b>Capítulo 2</b> .....	<b>7</b>
<b>Conocimientos previos</b> .....	<b>7</b>
2.1. <i>Discapacidad y limitaciones en la comunicación</i> .....	7
2.1.1. <i>Discapacidad cognitiva</i> .....	8
Autismo .....	8
Síndrome de Down.....	9
Parálisis cerebral .....	9
Daño cerebral adquirido.....	10
Afasia .....	10
2.2. <i>Definición de SAAC</i> .....	11
2.2.1. <i>Clasificación de los SAAC</i> .....	11
Sistemas de comunicación con ayuda.....	12
Sistema BLISS .....	12
SPC (Sistema pictográfico de comunicación) .....	12
2.3. <i>Productos de apoyo para la comunicación</i> .....	14
2.4. <i>Mensajería instantánea</i> .....	16
2.5. <i>Android</i> .....	17
2.6. <i>XMPP (Extensible Messaging and Presence Protocol)</i> .....	19
Streaming XML.....	20
Elemento <message> .....	21
Elemento <iq> .....	21
Elemento <presence>.....	23
2.7. <i>Librería Smack</i> .....	25
<b>Capítulo 3</b> .....	<b>26</b>
<b>Análisis y diseño del servicio</b> .....	<b>26</b>
3.1. <i>Elección de la plataforma de implementación</i> .....	26
3.2. <i>Elección del protocolo de comunicación</i> .....	27
3.3. <i>Elección del sistema de símbolos</i> .....	29
3.4. <i>Casos de uso</i> .....	30
3.5. <i>Arquitectura física del servicio</i> .....	33
3.6. <i>Arquitectura lógica del servicio</i> .....	35

3.6.1.	Componentes de software .....	35
3.6.2.	Diagrama de clases.....	38
3.6.3.	Almacenamiento de Pictogramas .....	42
3.6.4.	Extensión del protocolo XMPP .....	44
<b>Capítulo 4</b>	<b>.....</b>	<b>47</b>
<b>Implementación de la aplicación</b>	<b>.....</b>	<b>47</b>
4.1.	<i>Identificación de la aplicación</i> .....	47
4.2.	<i>Inicio de sesión</i> .....	47
4.2.1.	Creación de una cuenta de usuario.....	48
4.2.2.	Duración de una sesión .....	49
4.3.	<i>Lista de contactos</i> .....	50
4.4.	<i>Conversaciones</i> .....	52
4.4.1.	Formato de los mensajes .....	53
4.4.2.	Creación de mensajes .....	54
4.4.3.	Recepción de un mensaje .....	57
4.5.	<i>Administración</i> .....	59
4.5.1.	Perfil del usuario .....	60
4.5.2.	Contactos .....	61
	Añadir un contacto.....	63
	Responder a una invitación .....	66
	Eliminar un contacto .....	68
	Modificar un contacto .....	69
	Ocultar un contacto.....	69
	Añadir un comunicador.....	70
4.5.3.	Categorías de pictogramas.....	71
	Nueva categoría .....	72
	Editar categoría .....	73
	Ver pictogramas .....	73
	Añadir pictogramas .....	74
	Editar pictograma .....	75
	Ordenar pictogramas .....	76
	Eliminar pictogramas.....	77
	Añadir categorías predefinidas .....	78
4.5.4.	Ajustes.....	79
	Opción "General" .....	80
	Opción "Notificaciones" .....	82
	Opción "Texto a voz" .....	85
4.5.5.	Cierre de sesión.....	87
<b>Capítulo 5</b>	<b>.....</b>	<b>88</b>
<b>Conclusiones y trabajos futuros</b>	<b>.....</b>	<b>88</b>
<b>Bibliografía</b>	<b>.....</b>	<b>90</b>

# Capítulo 1

## Introducción

Existen muchas personas que, debido a alteraciones en funciones motrices, sensoriales y/o cognitivas, tienen limitaciones para ejercer actividades básicas como el habla (Beukelman & Mirenda, 1998). Dentro de este colectivo de personas existen aquellos que no han podido alcanzar un habla funcional que les permita comunicarse con su entorno, expresar sus necesidades personales y mantener los vínculos sociales necesarios para su desarrollo.

Sus necesidades de comunicación los motivan a buscar vías alternativas de expresión a través de códigos distintos de la palabra hablada. Estas necesidades especiales son atendidas por los denominados Sistemas Aumentativos y Alternativos de Comunicación (SAAC). Los SAAC son formas de expresión distintas al lenguaje hablado, que tienen como objetivo aumentar (aumentativos) y/o compensar (alternativos) las dificultades de comunicación y lenguaje de personas con discapacidad (Portal ARASAAC, 2015).

Un ejemplo típico de SAAC son los denominados sistemas pictográficos (Fernandez, 2013). En este tipo de sistemas cada imagen (pictograma) representa un concepto de uso común: comer, jugar, escuela, amigo, etc. Los sistemas pictográficos están destinados a personas que no están alfabetizadas, ya sea a causa de la edad o la discapacidad, y tiene la flexibilidad de permitir llevar adelante una comunicación desde un nivel muy básico hasta uno muy rico y avanzado.

El desarrollo de la tecnología está poniendo a disposición de las personas con discapacidad un amplio abanico de opciones para seleccionar el software más adecuado a sus demandas y adaptado a sus necesidades. Los SAAC han tenido un importante impulso a raíz de la popularización de los dispositivos como tabletas y celulares. Estos dispositivos, utilizados como soporte de apoyo para la comunicación, presentan muchas ventajas: facilitan su traslado, actúan según la lógica más simple y poseen el componente táctil que permite disminuir el tiempo entre una acción y su consecuencia alentando la atención y el interés por parte del usuario.

La mayoría de los SAAC gráficos que se han desarrollado para estos dispositivos, se centran en la comunicación directa cara a cara, es decir que tanto el emisor como el receptor del mensaje deben encontrarse en el mismo ambiente físico. Esto supone una seria

barrera en la comunicación de la persona con discapacidad al no contar con herramientas que le permitan alcanzar las mismas posibilidades de comunicación que el resto de los usuarios que cuentan, por ejemplo, con entornos estables de mensajería online. Es evidente que trasladar los sistemas pictográficos a la mensajería instantánea impulsaría la participación social de la persona con discapacidad al favorecer el intercambio de mensajes sin límite de tiempo ni espacio.

Hay que destacar que la participación social es un derecho fundamental de todas las personas, independientemente de sus capacidades. En este sentido las nuevas tecnologías se han convertido en uno de los pilares de la comunicación y de la interacción social de las personas y deben ser aprovechadas por todos independientemente de su edad, género, situación social o diversidad funcional.

## 1.1. Fundamentos del proyecto

El avance de la tecnología ha propiciado el desarrollo de múltiples y variados soportes de apoyo para los SAAC: ordenadores personales, tabletas, comunicadores del habla, tableros electrónicos, etc. Concretamente para el caso de las tabletas se diseñan cada vez más aplicaciones que intentan aprovechar las ventajas de este soporte. Su uso intuitivo y la interacción predominantemente visual y táctil proporcionan una estimulación constante para el usuario, con la ventaja adicional de que puede transportarse con facilidad.

En cuanto a las aplicaciones disponibles actualmente para personas que utilizan SAAC gráficos, existe una gran variedad, desde completos tableros de comunicación digitales hasta agendas que permiten establecer las actividades y rutinas de cada día a través de imágenes. Sin embargo existen muy pocos proyectos dedicados a la mensajería instantánea. Uno de ellos es Messenger Visual (Sitio web oficial Messenger Visual, 2015), una aplicación SAAC patrocinada por la fundación catalana El Maresme (Sitio web oficial Fundación Maresme, 2015) que permite la comunicación entre dos personas a través del intercambio de imágenes, sin embargo está disponible sólo para PC's y particularmente para Windows, además actualmente sólo es posible utilizarla sobre una red de Intranet. Otra aplicación de este estilo es Olá Mundo (Sitio web oficial Ola Mundo, 2015), una aplicación paga de mensajería que utiliza imágenes propietarias como símbolos en los mensajes; solo está disponible para su uso en iPad y iPhone.

En el mercado Argentino no existe una aplicación con las características mencionadas. Incluso las que se describieron anteriormente no tienen llegada aquí, o bien porque son pagas o porque su uso se restringe a una PC o un costoso iPad. Si tenemos en cuenta que según algunas estimaciones para el 2018 en Argentina habrá 70 millones de dispositivos móviles conectados a Internet (entre celulares y tabletas) (Ericsson, Junio 2013), es evidente que aquí existe una importante oportunidad para explotar el desarrollo de aplicaciones de comunicación alternativa y aumentativa, y específicamente haciendo uso de la comunicación online.

Con el presente trabajo se pretende realizar un aporte para aquellos usuarios con discapacidad que no encuentran en el mercado actual ninguna aplicación de mensajería instantánea que utilice sistemas pictográficos como principal forma de comunicación. Con la adecuada asistencia profesional y la colaboración de familiares y educadores, esta aplicación puede asistir en la mejora de la calidad de vida de los usuarios ya sea como parte del apoyo al aprendizaje o bien favoreciendo aspectos de la vida cotidiana como la comunicación y la integración social.

Cabe destacar que este trabajo fue presentado y aceptado para ser publicado en el simposio EST de las 45 Jornadas Argentinas de Informática (JAIIO - 5 al 9 de Septiembre de 2016 - ).

## 1.2. Objetivo General

Desarrollar un servicio de mensajería instantánea orientado a la comunicación aumentativa y alternativa que utilice símbolos pictográficos como elementos principales para la construcción de mensajes. A su vez procurar que este servicio pueda ser accedido desde dispositivos comunes de acceso masivo como tabletas y celulares y que la comunicación pueda ser establecida desde cualquier ubicación a través de Internet.

Se hará hincapié en dotar al servicio de las características necesarias para que los usuarios puedan interactuar con él con el menor entrenamiento posible. En este sentido, tanto el diseño del aspecto visual de la aplicación como su funcionalidad seguirán premisas de usabilidad y accesibilidad.

### 1.3. Objetivos Específicos

- Investigar acerca de la infraestructura tecnológica necesaria para implementar un servicio de mensajería instantánea que pueda ser accedido desde dispositivos móviles.
- Diseñar una arquitectura física y lógica que permita implementar el servicio para ser accedido desde dispositivos móviles desde cualquier ubicación a través de Internet.
- Diseñar interfaces de usuario que sigan los lineamientos de accesibilidad para soportar el uso del servicio por parte de usuarios con capacidades diferentes.
- Diseñar el lenguaje con el cual se construirán los mensajes gestionados por la aplicación.
- Desarrollar e implementar el sistema utilizando las herramientas seleccionadas y de acuerdo a la arquitectura diseñada.

## Capítulo 2

### Conocimientos previos

El objetivo de este capítulo es ofrecer una introducción a los conceptos básicos necesarios para comprender el contexto de este trabajo.

En primer lugar se describirán las principales características que presentan las personas con limitaciones en la comunicación ya que serán estas personas las que se beneficiarán con el uso del servicio y las que motivan el desarrollo de este proyecto. Luego se explicará el objetivo que persiguen los sistemas aumentativos de comunicación y se describirán brevemente los más utilizados. Finalmente se expondrán conceptos relacionados con la mensajería instantánea y su implementación en la plataforma Android.

#### 2.1. Discapacidad y limitaciones en la comunicación

Para la Organización Mundial de la Salud [4], la discapacidad es el término general que abarca las deficiencias, las limitaciones de la actividad y las restricciones de la participación. Las deficiencias son problemas que afectan a una estructura o función corporal; las limitaciones de la actividad son dificultades para ejecutar acciones o tareas, y las restricciones de la participación son problemas para participar en situaciones vitales.

Dependiendo de qué capacidad de la persona se vea afectada, la discapacidad se puede clasificar en: física, cognitiva o intelectual, sensorial y psíquica. La discapacidad física incluye cualquier impedimento en las habilidades motoras. En la discapacidad cognitiva, la persona tiene problemas para realizar tareas que requieren un proceso mental sencillo. La discapacidad sensorial afecta la capacidad de la persona para captar y percibir normalmente el mundo que lo rodea. Finalmente una persona presenta una discapacidad psíquica cuando sufre de trastornos en su comportamiento adaptativo, generalmente permanentes.

Para el estudio de este proyecto interesan principalmente aquellas limitaciones que impiden una persona el poder llevar adelante una comunicación eficaz. Generalmente los problemas graves en la comunicación se encuadran dentro de las denominadas discapacidades cognitivas.

### 2.1.1. Discapacidad cognitiva

Discapacidad cognitiva es un término genérico que describe a las personas que tienen limitaciones sustanciales en el desarrollo de tareas que dependen de procesos mentales básicos como la atención, la memoria y el lenguaje. Esta condición generalmente impide que la persona pueda comprender conceptos abstractos o hacer generalizaciones entre contextos. Las personas con discapacidad cognitiva también tienen problemas para desarrollar tareas que requieren de la interacción social. Finalmente, estas limitaciones pueden reducir drásticamente la confianza en sí mismo y el nivel de autonomía. Al igual que con los otros tipos de discapacidad, discapacidad cognitiva puede ser hereditaria, adquirida o desarrollada debido a factores naturales. Hoy en día hay muchas enfermedades que causan discapacidad cognitiva, pero cada uno tiene sus propias causas y plantea diferentes limitaciones para el individuo.

A continuación se describen algunas de las discapacidades cognitivas más comunes.

#### Autismo

Se define al autismo como la gama de trastornos complejos del neurodesarrollo, caracterizados por impedimentos sociales, dificultades en la comunicación, y patrones de conducta estereotípicos, restringidos y repetitivos. El trastorno autista, a veces llamado autismo o TEA (trastorno del espectro autista) clásico, es la forma más grave de TEA, mientras que otras afecciones a lo largo del espectro incluyen una forma más leve, por ejemplo el síndrome Asperger.

Las personas que sufren este trastorno tienen una deficiente relación social con su entorno, ya que normalmente no suelen poseer la capacidad de interpretar emociones, ideas, gestos, etc. Algunos de estos individuos suelen presentar conductas repetitivas. Suele diagnosticarse a temprana edad excepto en los casos en que la persona posea un impedimento físico el cual enmascara el trastorno. Algunos indicadores para diagnosticar el autismo son:

- No poseer habilidades de relacionarse con su entorno.
- No mostrar emociones recíprocas a las personas que interactúan con el individuo.
- Mostrar rutinas o comportamientos repetitivos.
- Comunicación deficiente o nula.

## Síndrome de Down

El Síndrome de Down (SD) es una condición genética que es causada por la presencia de una copia extra del vigésimo primer cromosoma. Se caracteriza por la presencia de un grado variable de discapacidad cognitiva y unos rasgos físicos peculiares que le dan un aspecto reconocible. Los rasgos físicos de las personas con SD incluyen una cara redonda y poco tono muscular, mientras que los deterioros cognitivos incluyen un coeficiente intelectual inferior a la media.

Las personas con SD también tienen problemas de salud adicionales, tales como un mayor riesgo de defectos congénitos del corazón y disfunciones tiroideas, que al final implica una esperanza de vida más corta. Hoy en día la incidencia del SD es de alrededor de uno por cada setecientos nacimientos. Es ampliamente reconocido que con apoyo familiar y una educación adecuada, las personas con SD pueden ser integradas plenamente en la sociedad.

## Parálisis cerebral

La parálisis cerebral se refiere a algunos trastornos neurológicos, que pueden provenir del sistema nervioso (movimiento, audición, aprendizaje, pensamiento y visión) o de disfunciones del cerebro. Las anomalías suelen aparecer durante la gestación de un bebé o en los dos primeros años de vida del niño. Algunas de las lesiones que puede tener el cerebro suelen ser debidas a la falta de oxígeno.

Para la parálisis cerebral no existe cura hasta el momento. Normalmente se suelen realizar tratamientos que le permitan al individuo ser menos dependiente y poder así desenvolverse mejor en su entorno. Algunos de los síntomas de la parálisis cerebral pueden variar en función de la gravedad de la lesión y de la parte del cuerpo afectada. Algunos de estos síntomas son:

- Articulaciones rígidas
- Debilidad muscular o pérdida de movimiento.
- Pérdida de coordinación
- Pérdida de visión o audición
- Dificultad para transmitir ideas

- Dificultad para hablar

### Daño cerebral adquirido

Es un daño cerebral (Gómez, Sancho, & Cabeza, 2009), ya sea causada por una lesión traumática, por ejemplo, un accidente de coche, o una lesión no traumática, por ejemplo, un Accidente Cerebrovascular (ACV). A pesar de que no afecta directamente a las capacidades intelectuales, por lo general presenta limitaciones físicas, sensoriales y cognitivas para el individuo. Por un lado las limitaciones físicas afectan a las capacidades motoras, incluyendo las dificultades para controlar y coordinar acciones. Por otro lado limitación sensorial puede incluir déficits visuales y de audición que dificultan la interacción con el mundo externo. Por último, las limitaciones cognitivas incluyen la atención, la memoria y trastornos del lenguaje que alteran la capacidad del individuo para expresar sus pensamientos. Al final, esas deficiencias físicas, sensoriales y cognitivas restringen el autocuidado y la interacción social, que en última instancia, tiene un impacto negativo en el comportamiento individual.

### Afasia

La afasia es un trastorno que afecta tanto a la expresión como a la comprensión del lenguaje. Esta enfermedad puede estar causada por lesiones, traumatismos o enfermedades de los sectores cerebrales implicados en el lenguaje. La afasia se puede dividir en:

*Afasia expresiva:* deficiencia en transmitir las ideas a través de la escritura o el habla.

*Afasia receptiva:* insuficiencia en la comprensión de la comunicación oral o escrita.

*Afasia anómica:* deficiencia en relacionar correctamente un objeto, personas o lugares con su nombre correspondiente.

*Afasia global:* esta tipología es la más grave, ya que el individuo pierde toda la capacidad comunicativa. Es decir, engloba las tres anteriores.

Determinados tratamientos para la afasia; normalmente se enfocan en terapias del lenguaje que se ajustan a las necesidades de cada individuo.

## 2.2. Definición de SAAC

Los **sistemas de comunicación aumentativos y alternativos (SAAC)** abarcan cualquier forma de comunicación distinta del habla y que es utilizada por una persona para expresar sus necesidades, sentimientos, ideas, etc. El uso de signos manuales y gráficos, el sistema Morse, la escritura, etc., son formas alternativas de comunicación para una persona que carece de la habilidad de hablar. La palabra **aumentativa** subraya el hecho de que la enseñanza de estas formas de comunicación tiene un doble objetivo: promover y apoyar el habla, y garantizar una forma de comunicación **alternativa** si la persona no aprende a hablar (Tetzchner & Martinsen, 1993). A menudo se habla indistintamente de “Comunicación Alternativa y Aumentativa (AAC)” o de “Comunicación Aumentativa (CA)”. Todas estas expresiones se refieren generalmente a un mismo concepto: los sistemas de comunicación aumentativa y alternativa.

### 2.2.1. Clasificación de los SAAC

Existen diferentes tipos de sistemas aumentativos y alternativos de comunicación, y por lo general están clasificados en dos categorías: **con ayuda** o **sin ayuda** (Basil & Puig, 1988).

En los **sistemas de comunicación sin ayuda** los códigos utilizados no necesitan ningún elemento físico, externo al emisor de dicho código, para realizarlo. Ejemplo de sistemas de comunicación sin ayuda: el habla, lenguajes de signos, empleo de gestos indicativos, utilización de gestos de sí y no, gestos de comprensión fácil, etc.

En los **sistemas de comunicación con ayuda** los códigos que se utilizan requieren un apoyo físico, un material, una ayuda externa, físicamente independiente del emisor que realiza la actividad comunicativa. Ejemplo de sistemas de comunicación con ayuda: un lápiz y papel (en el caso de la escritura), o un trozo de cartulina con símbolos gráficos dibujados, Braille, sistemas pictográficos, el habla a través de sintetizadores de voz, etc.

En ambas categorías la comunicación está compuesta por diversos **sistemas de símbolos**. En los sistemas de comunicación sin ayuda los símbolos son predominantemente *gestuales* (mímica, gestos o signos manuales) en tanto que en los sistemas de comunicación con ayuda los símbolos son *gráficos* (fotografías, dibujos, pictogramas, palabras o letras).

## Sistemas de comunicación con ayuda

Existen numerosos sistemas de comunicación con ayuda que van desde sistemas muy sencillos basados en dibujos o fotografías hasta sistemas progresivamente más complejos como los sistemas pictográficos o la ortografía tradicional (letras, palabras y frases). A continuación se describen dos de los sistemas de comunicación con ayuda más difundidos.

### Sistema BLISS

Los símbolos Bliss son símbolos gráficos que representan significados. Utilizan formas básicas para la transmisión de significados, a su vez los segmentos de estas formas se pueden variar para lograr nuevos significados. La combinación de los símbolos entre sí de diversas maneras también genera nuevos significados, con lo que se crea un sistema complejo capaz de expresar conceptos diferentes.

Los símbolos son fácilmente discernibles dispuestos en categorías diferenciadas por colores. El sistema BLISS posee reglas sintácticas propias aunque puede adaptarse a las necesidades de la persona que lo va a utilizar.

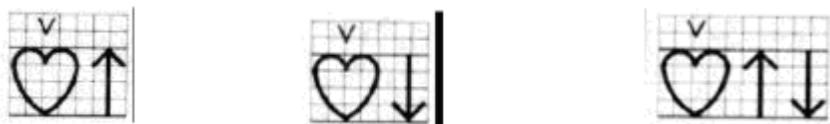


Figura 2.1 - Símbolos del sistema Bliss

A partir de símbolos simples, con significado propio se van conformando símbolos más complejos cuyo significado está en relación con dichos símbolos. Por ejemplo, la figura 2.1 muestra que cuando se añaden los símbolos de ARRIBA y/o ABAJO al símbolo SENTIMIENTO, se pueden crear tres nuevos significados que expresan emociones: FELIZ (sentimiento hacia arriba), TRISTE (sentimiento hacia abajo) y PREOCUPADO (altibajos en el sentimiento).

### SPC (Sistema pictográfico de comunicación)

Este sistema permite la comunicación mediante símbolos gráficos denominados pictogramas. Un pictograma es un dibujo que puede representar una realidad concreta (un

objeto, animal, persona, etc.), una realidad abstracta (un sentimiento, una idea), una acción, (leer, correr), e incluso un elemento gramatical (adjetivos, conjunciones, artículos, preposiciones, etc.).

El SPC fue diseñado especialmente para facilitar el reconocimiento y la asociación de significados y tiene la virtud de permitir desde un nivel de comunicación muy básico, que se adapta a personas con niveles cognitivos bajos, hasta un nivel de comunicación muy rico y avanzado, aunque nunca tan completo y flexible como el que se puede alcanzar con el uso de la lengua escrita. Está especialmente indicado para personas con un nivel de lenguaje expresivo simple, vocabulario limitado y que puede realizar frases con una estructura sencilla

El SPC no tiene una estructura gramatical propia, sigue la de la lengua de origen de la persona usuaria, esto hace que se pueda utilizar en diferentes realidades culturales y también combinar sus símbolos con símbolos de otros sistemas.

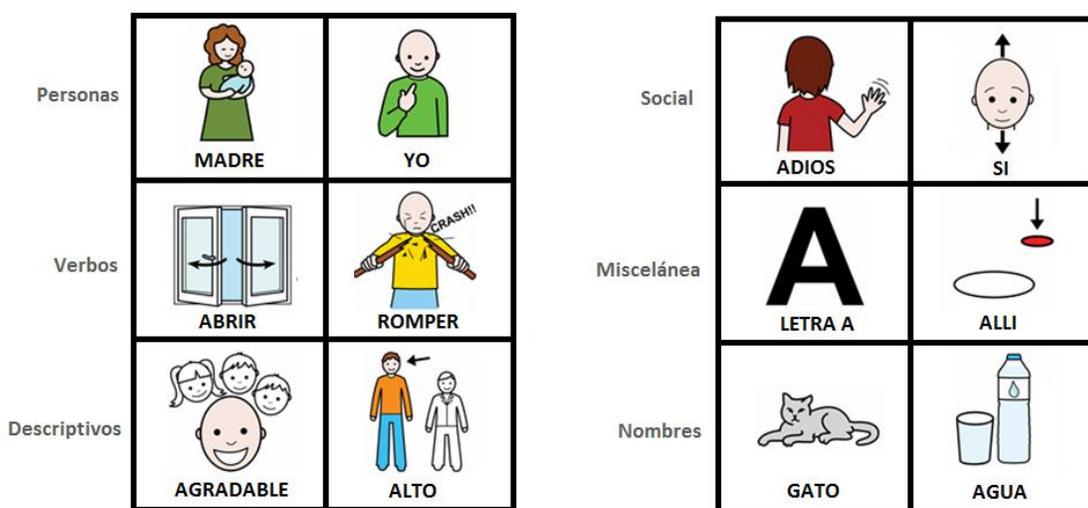


Figura 2.2 - Pictogramas y categorías del sistema SPC

Este sistema fue creado en 1981 por Roxana Mayer Jonson y originalmente contaba con 300 pictogramas. La figura 2.2 muestra algunos de estos pictogramas organizados en diferentes categorías.

Actualmente existen numerosas fundaciones y organizaciones gubernamentales que ponen a disposición en sus sitios de Internet, un amplio catálogo de pictogramas. Tal es el caso del portal ARASAAC (Portal ARASAAC, 2015) que dispone aproximadamente de

15000 pictogramas de descarga gratuita, y se amplía cada vez más de acuerdo a las necesidades y sugerencia de los usuarios.

### 2.3. Productos de apoyo para la comunicación

Al utilizar **sistemas de comunicación con ayuda** (BLISS y SPC, por ejemplo) es necesario seleccionar también un producto de apoyo que dé soporte al sistema elegido. Los productos de apoyo para la comunicación son recursos tecnológicos que sirven de soporte a los símbolos del sistema de comunicación seleccionado. Estos productos de apoyo se dividen en productos de baja y alta tecnología.

Los **productos de apoyo de baja tecnología** son instrumentos simples, generalmente contruidos con materiales comunes de bajo costo y su fabricación es muy sencilla. Como ejemplo de este tipo de productos se pueden nombran: papel y lápiz, impresiones sobre papel (figura 2.3), paneles de cartón, hojas plastificadas, imanes sobre pizarra de metal, etc. Los símbolos del sistema de comunicación elegido se disponen sobre el producto de apoyo de modo que se adapten a las características motrices y visuales de la persona usuaria.



Figura 2.3 - Tarjetas impresas utilizadas como productos de apoyo de baja tecnología

Los **productos de apoyo de alta tecnología** son equipos de tecnología compleja, generalmente electrónicos, que permiten utilizar diferentes niveles de vocabulario y adaptarse a las necesidades comunicativas de la persona usuaria. Dentro de esta

clasificación se encuentran los comunicadores electrónicos, las computadoras personales, los celulares y tabletas, etc. En la figura 2.4 se muestran dos tipos de productos de apoyo de alta tecnología: smartphones (izquierda) y comunicador electrónico (derecha).



Figura 2.4- Productos de apoyo de alta tecnología

Generalmente se denomina **comunicador** al dispositivo electrónico portátil diseñado y construido específicamente para la interacción comunicativa mediante el uso de pictogramas que representan el lenguaje. Sin embargo el desarrollo de dispositivos como teléfonos, tabletas, notebooks, etc., está permitiendo que, mediante la instalación de un software de comunicación, se conviertan en un verdadero comunicador equiparable a los denominados tradicionalmente así (Delgado Santos, 2012).

Un **software de comunicación** es un programa que, instalado en un dispositivo concreto, permite a personas con discapacidad y limitaciones en esta actividad, poder comunicarse, interactuar con las personas que le rodean y modificar su entorno.

De acuerdo a los conceptos introducidos anteriormente es posible encuadrar a la herramienta desarrollada en el presente proyecto como un **software de comunicación** que utiliza el SAAC pictográfico **SPC** como lenguaje para la construcción de mensajes y utiliza como soporte físico un **producto de apoyo a la comunicación de alta tecnología** como lo son los dispositivos smartphone y las tabletas.

## 2.4. Mensajería instantánea

Los servicios de Mensajería Instantánea (MI) están diseñados para permitir a los usuarios comunicarse a través de una red pública de datos, por lo general en Internet, mediante el intercambio de mensajes multimedia en tiempo real. Para mantener la comunicación con otros usuarios los servicios de MI permiten mantener una lista de contactos e intercambiar mensajes de presencia para saber cuándo los contactos estén disponibles. En cuanto a la comunicación, los servicios de MI admiten tanto mensajes de uno a uno como mensajes a múltiples contactos a la vez. En las conversaciones (chat) uno a uno, dos usuarios intercambian mensajes de texto de forma privada, mientras que en los chat multiusuario dos o más participantes se unen a una sala virtual para comunicarse. Además de mantener una conversación, la mayoría de los servicios ofrece otras funciones adicionales: mostrar y mantener un estado (Disponible, Ocupado, Invisible, etc.), mostrar un mensaje de estado, crear y eliminar contactos, agrupar contactos, utilizar imágenes de perfil (avatar), etc.

En general, los servicios de MI utilizan una arquitectura cliente-servidor para funcionar. Por un lado, uno o más servidores se encargan de proporcionar las características básicas de MI, como la autenticación de usuario y el enrutamiento de mensajes. Existen dos enfoques con respecto a la arquitectura de servidor: simétrica y asimétrica. En el enfoque simétrico todos los servidores realizan las mismas funciones, mientras que en el enfoque asimétrico cada servidor se dedica a una tarea particular. Por otro lado, los clientes de MI proporcionan una interfaz gráfica que permite a los usuarios iniciar sesión en el servicio, administrar la lista de contactos y comunicarse con otros usuarios.

Para coordinar la comunicación entre los servidores y los clientes, se utiliza un determinado protocolo que define cómo intercambiar mensajes. Actualmente, los protocolos que soportan este tipo de servicios de MI, así como los clientes y servidores que permiten a los usuarios comunicarse, son propietarios y sus especificaciones no están disponibles al público. Por ejemplo, el servicio Whatsapp utiliza FunXMPP, una variante del protocolo abierto XMPP y sus especificaciones no están disponibles para los desarrolladores o usuarios. Hangouts, el mensajero de Google, utiliza un protocolo de comunicación propietario, dejando de lado recientemente al XMPP utilizado por Google Talk. Finalmente Skype utiliza un protocolo cerrado y no es posible conocer sus detalles.

## 2.5. Android

Android es un sistema operativo basado en el núcleo de Linux. Fue diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes y tabletas aunque su desarrollo ha evolucionado y actualmente puede ser instalado en otros dispositivos como relojes inteligentes, televisores y automóviles. Android, a diferencia de otros sistemas operativos para dispositivos móviles como iOS o Windows Mobile, se desarrolla de forma abierta y se puede acceder tanto al código fuente como a la lista de incidencias donde se pueden ver problemas aún no resueltos y reportar nuevos problemas. La empresa que lleva adelante el desarrollo de Android es Google, quien lidera la Open Handset Alliance, un consorcio de empresas que se comprometen a seguir los estándares abiertos y libres.

Si bien fue construido para dispositivos que ciertamente disponen de menos recursos (almacenamiento, procesamiento, memoria) que una PC de escritorio, la plataforma Android exhibe características similares a las que posee un framework para PC. Google permite a los desarrolladores construir sobre este framework utilizando kit de desarrollo llamado: SDK (Software Development Kit) de Android.

El SDK de Android es compatible con la mayor parte de la Plataforma Java Standard Edition (Java SE). Debido a que las aplicaciones se construyen en Java a través del SDK, es necesario disponer de una máquina virtual de Java (JVM) para interpretar el código de Java en tiempo de ejecución. Android ofrece su propia JVM con el fin de contrarrestar las limitaciones de los dispositivos tales como la memoria, la velocidad del procesador y la disponibilidad de la batería. Esta máquina virtual se llama Dalvik VM.

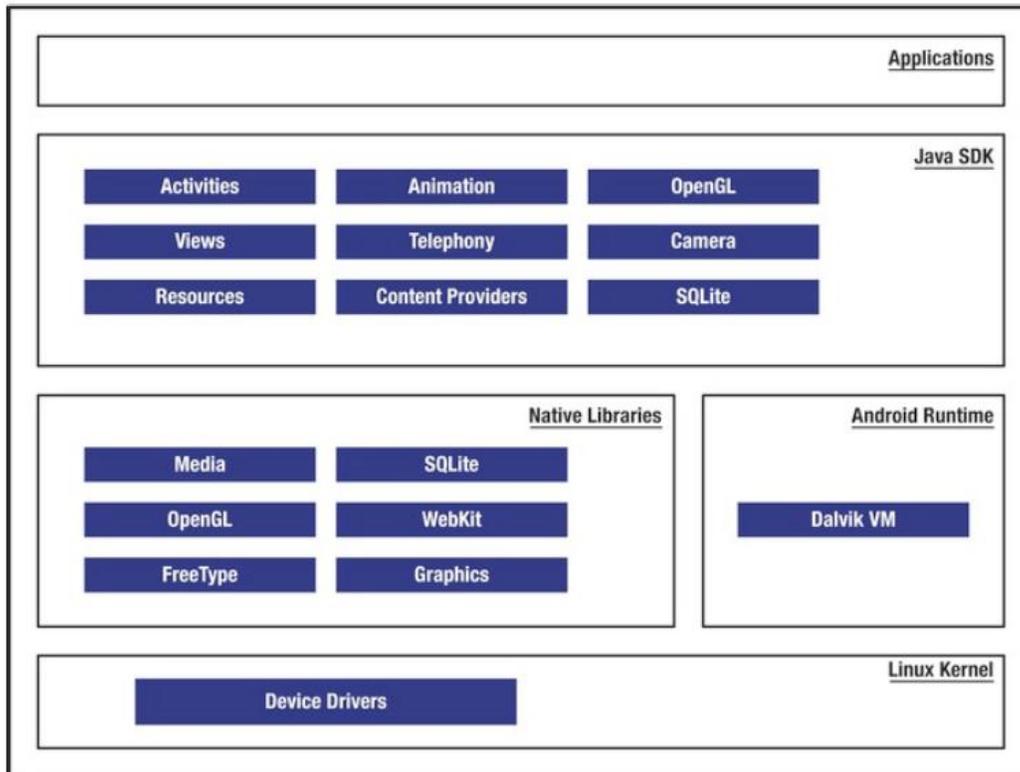


Figura 2.5 - Pila de software de Android

En la figura 2.5 se representa esquemáticamente todos los niveles de software que forman parte de la plataforma Android. El núcleo de Android es un kernel Linux responsable de los controladores de dispositivos, acceso a los recursos, administración de energía, y otras tareas del sistema operativo. Los controladores de dispositivos más comunes son la pantalla, cámara, teclado, Wi-Fi, memoria flash, audio, etc.

En el siguiente nivel y por encima del núcleo se incluyen un conjunto de bibliotecas de [C/C++](#) usadas por varios componentes del sistema. Estas características se exponen a los desarrolladores a través del SDK de Android. Algunas de estas bibliotecas son: System C library (implementación biblioteca C estándar), Media (librerías de grabación y reproducción de formatos de audio y vídeo), OpenGL (bibliotecas de gráficos 2d y 3d) y SQLite (base de datos relacional), entre otras.

La mayor parte del framework Android accede a estas bibliotecas del núcleo a través de la Dalvik VM. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalvik ha sido escrito de forma que un dispositivo puede correr múltiples máquinas virtuales de forma eficiente.

Finalmente el Java SDK de Android permite que los desarrolladores puedan construir sus aplicaciones. El SDK incluye librerías como telefonía, interfaz de usuario, bases de datos, administración de paquetes (instalación de aplicaciones), conectividad, etc.

## 2.6. XMPP (Extensible Messaging and Presence Protocol)

La mayoría de los protocolos que se utilizan para proveer servicios de MI, así como los clientes y servidores que permiten a los usuarios comunicarse, son propietarios y sus especificaciones no están disponibles al público. Para proporcionar una alternativa abierta, la Internet Engineering Task Force (IETF), responsable de la estandarización de la mayoría de las tecnologías para Internet (TCP/IP, HTTP, SMTP, POP, etc.), decidió formalizar un protocolo de mensajería al que llamó Extensible Messaging and Presence Protocol (XMPP)<sup>1</sup>. Luego de algunos años de trabajo la IETF publicó las especificaciones fundamentales de XMPP en las series de RFC (Request for comment) “RFC 3920” y “RFC 3921” (IETF, 2004) en Octubre de 2004.

En esencia el protocolo XMPP es una tecnología abierta destinada a las comunicaciones en tiempo real basada en la transmisión (streaming) de XML. Se trata de un estándar abierto debido a que no está vinculado a ninguna empresa o proyecto de software sino que las especificaciones XMPP definen protocolos que se pueden utilizar para la comunicación entre entidades a través de una red, por ejemplo, Internet.

El protocolo XMPP sigue la arquitectura cliente/servidor. El servidor XMPP participa en todas las comunicaciones XMPP. Su principal responsabilidad es la de proveer de servicios XMPP a los clientes de su dominio, como pueden ser la redirección de sus mensajes y la gestión de sus cuentas de usuario. El cliente XMPP es quien interactúa directamente con el usuario, recoge sus peticiones y las envía al servidor para que las trate. Normalmente el cliente transmitirá al servidor los mensajes que el usuario quiere que lleguen a otro usuario, luego el servidor deberá localizar al destinatario y hacerle llegar el mensaje.

Los clientes XMPP únicamente se comunican con el servidor XMPP de su dominio, que es quien les ofrece el servicio, esto se esquematiza en la figura 2.6. El servidor de ese dominio será el encargado de comunicarse con el servidor del destinatario para finalmente entregar el mensaje al cliente correspondiente. Vale aclarar que los mensajes que envían o

---

<sup>1</sup> <http://xmpp.org/>

reciben los usuarios son fragmentos de XML correctamente formados y es el mismo protocolo el que define el formato adecuado para cada situación.

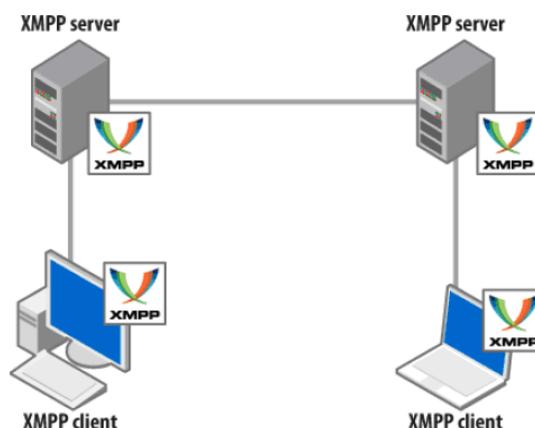


Figura 2.6 - Interacción XMPP entre clientes y servidores

Esta arquitectura permite crear clientes sencillos, ya que la mayor parte del trabajo de procesamiento y lógica del servicio es llevada a cabo por el/los servidor/es. Esto hace que la implementación de los clientes de mensajería XMPP resulte más simple. Por ejemplo, si solo se quiere crear un cliente sencillo que envíe y reciba mensajes, habrá que implementar esa pequeña parte del protocolo y la mayor parte del esfuerzo de programación consistirá en desarrollar la interfaz de usuario.

### Streaming XML

XMPP provee una forma de enviar piezas de XML desde una entidad a otra (desde el cliente al servidor y viceversa) en tiempo real. Las piezas que cada entidad envía van formando gran documento XML que se abre al principio de la comunicación y que se va construyendo a medida que estas piezas se van enviando a la otra entidad. Cuando alguna de estas entidades decide culminar la comunicación, sus respectivos documentos se cierran.

Existen dos tipos de piezas XML que se intercambian en una comunicación, éstas son: *XML stream* y *XML stanza*. Un *XML stream* es un contenedor de todos los elementos XML que se intercambian y representa el inicio y el final de una comunicación XMPP. Se denota por la etiqueta `<stream>` de inicio y la etiqueta `</stream>` de fin. Dentro de un *XML stream* una entidad puede enviar un número ilimitado de *XML stanzas*. Un *XML stanza* es una unidad de información que cuenta con una semántica determinada y que viaja

dentro de un *XML stream*. En principio el protocolo define tres tipos de *XML stanzas*: `<message>`, `<iq>` y `<presence>`. Cada una de ellas representa un bloque XML bien formado y puede tener atributos y elementos anidados que están definidos por el mismo protocolo. A continuación se explica cada uno de ellos.

### Elemento `<message>`

El *XML stanza* `<message>` es el elemento que se utiliza para trasladar información de una entidad a otra, generalmente en forma de texto. En la figura 2.7 puede verse un ejemplo típico de este elemento.

```
<message
  from='juliet@example.com'
  to='romeo@example.net'>
  <body>
    Hola, cómo estás?
  </body>
</message>
```

Figura 2.7 - Mensaje típico en XMPP

El protocolo define algunos atributos básicos para este elemento como lo son *from* y *to*, que indican el remitente y el destinatario del mensaje respectivamente utilizando el formato JID (EL JID, por Jabber ID, representa la forma de identificar a un usuario en XMPP y es similar a como lo hace el correo electrónico). En el subelemento `<body>` se almacena el texto del mensaje que se quiere enviar. Los elementos deben estar bien formados por lo que todas las etiquetas de inicio deben contener sus correspondientes etiquetas de fin, en este caso `</body>` y `</message>`.

### Elemento `<iq>`

El *XML stanza* `<iq>` (por Info/Query) se utiliza para realizar interacciones de tipo petición/respuesta, así que cuando una entidad envía un elemento `<iq>` siempre deberá recibir una respuesta del destinatario. En cada petición se debe incluir un atributo obligatorio *id* que también debe incorporarse en la respuesta para realizar la trazabilidad de cada petición. Otro atributo obligatorio es *type* que indica el tipo de petición o respuesta que se está enviando y puede contener los valores “get”, “set” o “result”. El valor “get” se utiliza en una petición que solicita algún tipo de información, el valor “set” se utiliza

en peticiones que proveen algún tipo de información y el valor “result” se utiliza en las respuestas a dichas peticiones.

La semántica de un elemento `<iq>` está dado por el subelemento `<query>`. Los elementos `<query>` que van dentro de los paquetes `<iq>` establecen un espacio de nombres XML usando el atributo `xmlns`. Cada espacio de nombres se denomina extensión `<iq>` y tiene sus propias etiquetas y atributos, y determinadas reglas para su procesamiento. La diferencia entre las extensiones `<iq>` son los campos que presenta y su significado para quien lo envía y para quien procesa la petición. El protocolo XMPP define muchas extensiones como `jabber:iq:roster` (manejo de contactos), `jabber:iq:auth` (autorización en el servidor) y `jabber:iq:register` (manejo de cuentas de usuario) pero es posible agregar extensiones propias, siempre que tanto el emisor como el receptor entiendan la extensión y la implementen para poder tratarla adecuadamente.

Cliente juliet@example.com	Servidor example.com
<pre>&lt;iq   from="juliet@example.com"   id="rr82a1z7"   type="get"&gt;   &lt;query xmlns="jabber:iq:roster"/&gt; &lt;/iq&gt;</pre>	<pre>&lt;iq   id="rr82a1z7"   to="juliet@example.com"   type="result"&gt;   &lt;query xmlns="jabber:iq:roster"&gt;     &lt;item jid="romeo@example.net"/&gt;     &lt;item jid="moran@example.net "/&gt;   &lt;/query&gt; &lt;/iq&gt;</pre>

Figura 2.8 - Petición y respuesta `<iq>`

La figura 2.8 muestra un ejemplo de petición y respuesta `<iq>` que se utiliza para obtener la lista de contactos de un usuario haciendo uso de la extensión `jabber:iq:roster` cuya especificación está definida en el protocolo. En la petición se incluye el atributo `type` con el valor “get” lo que denota que se está solicitando información. También incluye el atributo `from` que contiene el emisor de la petición, y el atributo `id` que identifica unívocamente a la petición dentro de todo el `XML stream`. Desde el servidor se obtiene un

elemento `<iq>` de tipo “result” lo que indica que es una respuesta y adicionalmente mediante el atributo `id` se puede relacionar con la petición que originó dicha respuesta. La lista de contactos se devuelve como un conjunto de elementos `<item>` dentro de `<query>`.

### Elemento `<presence>`

El último elemento que define el protocolo es `<presence>`. La presencia, en el contexto de la MI, es la posibilidad que tiene un usuario de hacer saber a sus contactos su estado de conexión (online, offline) y su disponibilidad para la comunicación (ocupado, fuera, etc.). Cuando un usuario cambia su estado, el cliente envía un elemento `<presence>` al servidor y éste será el encargado de hacerlo llegar a cada uno de los contactos del usuario.

Un elemento de presencia comienza con la etiqueta `<presence>` y posee, en su versión simplificada, dos atributos: `from` y `type`. El atributo `from` indica el emisor de la presencia y el atributo `type` indica el estado conectado o desconectado según tenga los valores “available” o “unavailable” respectivamente. Puede contener un subelemento `<show>` y un subelemento `<status>`. `<show>` se utiliza para indicar la disponibilidad del usuario y puede tener los valores: **chat** (disponible para chatear), **away** (fuera), **xa** (extended away, fuera por un periodo prolongado de tiempo) y **dnd** (do not disturb, el usuario no desea recibir mensajes). En el subelemento `<status>` se puede colocar un texto libre que mostrará al resto de los contactos. En la figura 2.9 se observa un ejemplo de este tipo de *XML stanza*.

```
<presence
  from="juliet@example.com"
  type="available">
  <show>xa</show>
  <status>down the rabbit hole!</status>
</presence>
```

Figura 2.9 - Ejemplo típico de un mensaje de presencia en XMPP

En la figura 2.10 se muestra un ejemplo sencillo de una comunicación XMPP entre un servidor y un cliente. Las piezas de XML que se muestran allí son enviadas de una entidad a otra a través de una red. El cliente comienza un *XML stream* enviando la etiqueta

<stream> al servidor, esto representa el inicio de una sesión del cliente con el servidor. Cuando un servidor acepta una conexión de un cliente responderá con <stream> para confirmar el inicio de la misma. Luego de establecer la conexión es natural que se lleven adelante diferentes tareas entre las dos entidades como la autenticación del usuario y el establecimiento del tipo de encriptación de la conexión, todo esto mediante el intercambio de elementos <iq> cuyas extensiones están definidas en el mismo protocolo. Durante toda la vida de la conexión tanto el cliente como el servidor pueden intercambiar una ilimitada cantidad de elementos <message>, <iq> y <presence>.



Figura 2.10 - Ejemplo de una comunicación XMPP

Tanto el servidor como el cliente pueden cerrar el flujo de datos en cualquier momento enviando una etiqueta de cierre `</stream>`. En ese momento, cada uno podrá cerrar la conexión y terminar la comunicación.

## 2.7. Librería Smack

Smack es una librería escrita en Java que permite desarrollar clientes que utilicen la comunicación con servidores XMPP en tiempo real. Implementa gran parte del protocolo XMPP y muchas de sus extensiones. La ventaja de utilizar una librería para trabajar con XMPP radica en el hecho que permite abstraerse la comunicación y detalles de bajo nivel y concentrarse en los esfuerzos para construir el software que hará uso de XMPP.

Smack proporciona construcciones de alto nivel que permiten programar de manera más eficiente y no requiere estar familiarizado con el formato XML o conocer en detalle el protocolo XMPP. Es una librería Open Source bajo la licencia Apache y en su versión 4.1 puede ser utilizada tanto para aplicaciones de PC como para Android.

## Capítulo 3

### Análisis y diseño del servicio

#### 3.1. Elección de la plataforma de implementación

Una de las premisas sobre las que se basó la elección de la plataforma de implementación fue la de procurar que la aplicación pueda ser utilizada por la mayor cantidad de usuarios posibles. Para cumplir con este propósito, se evaluaron 2 variables: la cantidad de dispositivos móviles disponibles que actualmente soportan la plataforma y el precio de adquisición de un dispositivo nuevo. Una plataforma que corre sobre dispositivos que tienen una gran variedad de marcas y características asegura la utilización masiva de los mismos. A su vez un precio accesible, permite que aquellas personas que no poseen actualmente un dispositivo y deseen adquirirlo, puedan hacerlo al menor costo posible.

En la actualidad existen 4 plataformas que compiten en el mercado de dispositivos móviles: iOS, Android, Windows Phone y BlackBerry OS. En cuanto a la aceptación por parte de los usuarios de las diversas plataformas, existen muchas estadísticas que demuestran la ventaja que lleva Android sobre el uso de cualquier otra plataforma, por lo menos en Argentina. Una de estas estadísticas es la que ofrece Kantar Worldpanel<sup>2</sup> en su sitio Web (figura 3.1). Esta empresa se dedica a la recolección y entrega de datos acerca de patrones de compra y uso de productos de consumo masivo por parte de los consumidores. Según los datos recogidos en Mayo de 2015, Android lidera el mercado con un 79.2% de la cuota de mercado. Lo sigue el sistema operativo Windows son un 13,7% de la cuota del mercado.

Sistemas operativos –Smartphones			
Share de ventas (%)			
Argentina	3m-May14	3m-May15	% pt. Change
Android	77,9	79,2	1,3
iOS	0,6	2,8	2,2
Windows	10,6	13,7	3,1
BlackBerry	7,9	2,3	-5,6
Other	3,0	2,0	-1,0

Figura 3.1 - Estadísticas de uso de S.O. en smartphones

<sup>2</sup> [www.kantarworldpanel.com/ar](http://www.kantarworldpanel.com/ar)

Una de las particularidades de Android (y seguramente un factor clave de su éxito) es su libre distribución y su código fuente abierto, por lo que las empresas que desarrollan dispositivos pueden instalarlos en sus productos sin ningún costo e inclusive introducir algunas modificaciones y adaptaciones si lo creen necesario. La mayoría de las empresas fabricantes de dispositivos móviles ofrece Android como software de base para sus productos, algunos ejemplos son: Sony, Samsung, LG, Alcatel, Motorola, Philips, Huawei, etc.

La gran variedad de marcas que utilizan Android en sus productos también permite contar con un amplio abanico de precios que varían dependiendo la calidad del producto, la marca, el hardware que posee, etc.

Otra de las ventajas de Android es que las aplicaciones se desarrollan en Java, un lenguaje conocido y estudiado en el ambiente académico y que además tiene una cantidad considerable de documentación online. Las aplicaciones en el iOS de Apple se programan en Objective-c en tanto que en Windows se deben programar en .Net aunque también es posible correr aplicaciones desarrolladas en Java.

### 3.2. Elección del protocolo de comunicación

Como se mencionó en el apartado de IM, el esquema típico para desarrollar un servicio de este tipo es la arquitectura cliente-servidor. Para que el cliente y servidor puedan comunicarse se debe establecer un protocolo que indique la forma en que dicha comunicación se llevará a cabo. Aquí la primera decisión que se debe tomar es si se desarrolla un protocolo ad-hoc o bien si se utiliza un protocolo existente y se adapta para dar solución al problema planteado.

Si se utiliza una solución personalizada se debe tener en cuenta que el cliente y el servidor son dos entidades con funciones completamente diferentes. El cliente se encarga de interactuar con el usuario y enviar mensajes al servidor, en tanto que el servidor se encarga de encaminar los mensajes al destinatario y almacenar información relevante para cada usuario. Esto implica que se deberán desarrollar dos aplicaciones, una para el cliente y otra para el servidor. La mayor parte del esfuerzo se deberá destinar al desarrollo del servidor debido a las múltiples funciones que debe realizar, entre las que se encuentran: administración de cuentas de usuarios, autenticación y autorización de usuarios, almacenamiento y entrega de mensajes, mantenimiento del estado del usuario y sus

contactos, etc. Además se debe poner atención en cumplir con determinados requisitos como la seguridad en la comunicación, la disponibilidad del servicio, la velocidad de entrega, etc.

Adicionalmente se debe tener en cuenta que el servicio se plantea para ser utilizado desde cualquier ubicación a través de Internet por lo que el servidor deberá estar disponible en Internet a través de un servicio de hosting.

La otra opción de comunicación entre cliente y servidor es utilizar alguna solución existente y de ser necesario adaptarla para cumplir con los objetivos de este proyecto. Ya se mencionó que la mayoría de los servicios de mensajería son propietarios y no se pueden obtener los detalles de su implementación ni tampoco tener acceso a productos de software libre (frameworks, aplicaciones, librerías) que permitan interactuar con estos servicios a menos que se utilicen sus propias aplicaciones.

Inmediatamente surge el nombre del protocolo XMPP como una alternativa libre y abierta para llevar adelante la implementación. Que éste protocolo sea abierto significa que es posible conocer todas sus especificaciones ya que son públicas. Además posee la categoría de estándar y es mantenido por la IETF. Esto último es importante ya que permite que los desarrolladores puedan crear sus propias aplicaciones o piezas de software basados en este protocolo y éstas puedan ser utilizadas por el resto de la comunidad para sus propios desarrollos. Por ejemplo, un determinado cliente de chat basado en XMPP puede ser utilizado para conectarse a cualquier servidor que también implemente XMPP.

Actualmente existen gran cantidad de software basado en XMPP: clientes, servidores y librerías para interactuar con ambos. La mayor parte de este software es de distribución libre y de código abierto. Una de las principales ventajas es que se puede tener un servidor plenamente funcional en muy poco tiempo y totalmente gratis con sólo descargar e instalar algunos del software de servidor XMPP disponibles (*ejabberd*, *Tigase*, *OpenFire*, etc). No sólo eso, sino que se puede utilizar cualquier servidor de la red federada y utilizar sus servicios para fines propios, es decir, sólo desarrollar la aplicación cliente y crear los usuarios en cualquier servidor público de la red XMPP para lograr la comunicación entre clientes.

En cuanto al desarrollo de aplicaciones clientes existen muchas alternativas y algunas de ellas también son libres, aunque para dispositivos móviles no hay tantas opciones. Algunas de ellas son: Beem, Conversations, Xabber, etc. Para el caso de la

aplicación cliente se optará por desarrollar una aplicación desde cero ya que tomar una aplicación previamente desarrollada y realizar las modificaciones necesarias para lograr lo que se pretende en este proyecto, requeriría un esfuerzo igual o mayor que el que requiere construirla totalmente desde el comienzo.

En conclusión, se optará por utilizar el protocolo XMPP para la comunicación entre el cliente y servidor. El servidor a utilizar podrá ser cualquiera de los servidores públicos de la red XMPP. La aplicación cliente se desarrollará bajo la plataforma Android utilizando el SDK provisto por el fabricante y se hará uso de la librería **Smack** para implementar la interacción XMPP con el servidor.

### 3.3. Elección del sistema de símbolos

Una rápida investigación acerca de los sistemas de símbolos disponibles, basta para afirmar que el SPC es el sistema más utilizado en el contexto de los SAAC gráficos. Y es que posee muchas ventajas que lo diferencian del resto de los sistemas.

En primer lugar su diseño es “transparente” es decir que el diseño de sus pictogramas revela su significado, sin aprendizaje previo por parte de la persona ni de los interlocutores, puesto que sus imágenes tienen un gran parecido con la realidad (<http://ares.cnice.mec.es/informes/18/contenidos/89.htm>)

Otra ventaja importante es que no tiene una sintaxis propia lo que hace que la construcción de frases sea muy simple y dependan de la destreza intelectual o motora de la persona usuaria adecuándose a cada situación particular. Incluso no tiene restricciones de símbolos, si la persona necesita utilizar una imagen determinada por fuera de las preestablecidas lo puede hacer, incluso puede utilizar símbolos de otros sistemas. Finalmente es posible encontrar infinidad de recursos que se han desarrollado utilizando este sistema; libros de cuentos, tableros de comunicación, software de comunicación, catálogos de pictogramas, etc. son algunos ejemplos de estos recursos.

Para el desarrollo de este proyecto interesa particularmente disponer de un conjunto de pictogramas para permitir que los usuarios del servicio creen sus mensajes. El portal ARASAAC posee una gran cantidad de pictogramas que se publican bajo Licencia Creative Commons lo que significa que es posible utilizarlos siempre que no se haga un uso comercial de ellos. Otra opción gratuita son los pictogramas **Sclera Picto's** que tienen la particularidad que están diseñados en color blanco sobre un fondo negro para ayudar la

lectura por parte de personas con baja visión. También es posible encontrar catálogos de pictogramas pagos, como es el caso de <https://www.mayer-johnson.com> en donde se ofrecen, por ejemplo, conjuntos de 350 pictogramas a un precio de 15 dólares.

En definitiva se optará por utilizar el sistema SPC empleando pictogramas ARASAAC debido a su disponibilidad gratuita y su amplio catálogo.

### 3.4 Casos de uso

A continuación se muestran los principales casos de uso que representan las funcionalidades básicas con las que deberá contar la aplicación cliente.

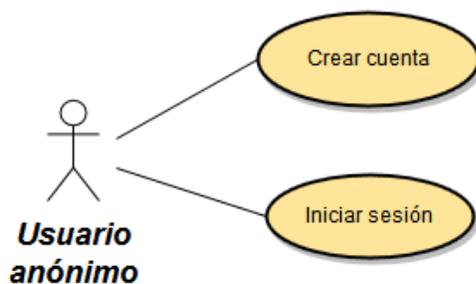
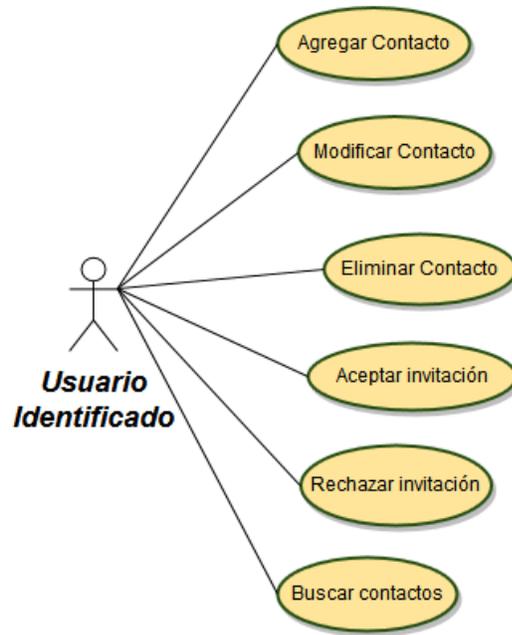


Figura 3.2 - Caso de uso para un usuario no identificado

En la figura 3.2 se muestra un usuario que aún no ha iniciado sesión. Aquí el usuario tiene dos opciones: ingresar mediante un nombre de usuario (cuenta) previamente creado o bien proceder a la creación de una nueva cuenta que le permitirá luego el acceso a la aplicación.



*Figura 3.3 - Casos de uso para administración de contactos*

La figura 3.3 muestra los casos de uso cuando el actor es el usuario identificado, es decir que ha ingresado a la aplicación a través de un inicio de sesión (CU anterior). El usuario podrá invitar a otro usuario (contacto) a formar parte de su lista de contactos, el contacto que recibe la invitación podrá aceptarla o rechazarla. Es importante señalar que ambos usuarios deberán estar de acuerdo en mantenerse en contacto uno con el otro.

También será posible colocar un nombre o alias a cada contacto para su mejor identificación. Finalmente el usuario podrá eliminar uno de sus contactos en cualquier momento.

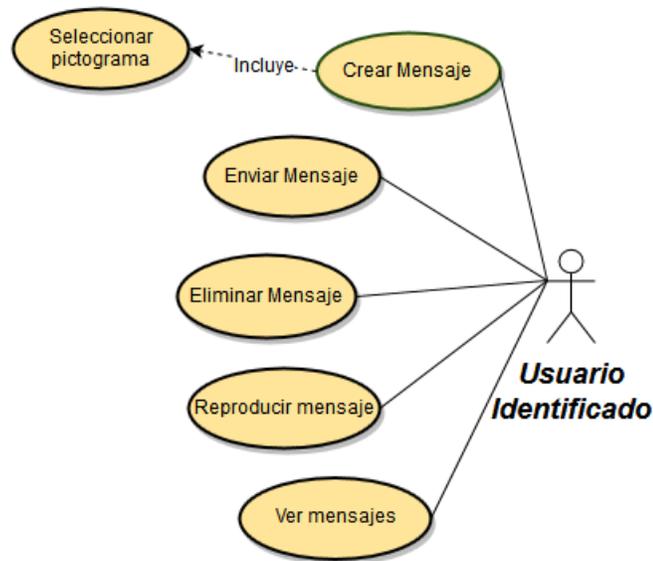


Figura 3.4 - Casos de uso para envío y recepción de mensajes

La figura 3.4 muestra los casos de uso que incluyen las funcionalidades básicas de envío y recepción de mensajes de cualquier aplicación de mensajería: crear mensajes, enviarlos y ver los mensajes recibidos. La particularidad aquí radica en el hecho de que la construcción de los mensajes implica la selección de los pictogramas que se incluirán en él. Adicionalmente será posible reproducir los mensajes a través de la síntesis de voz.

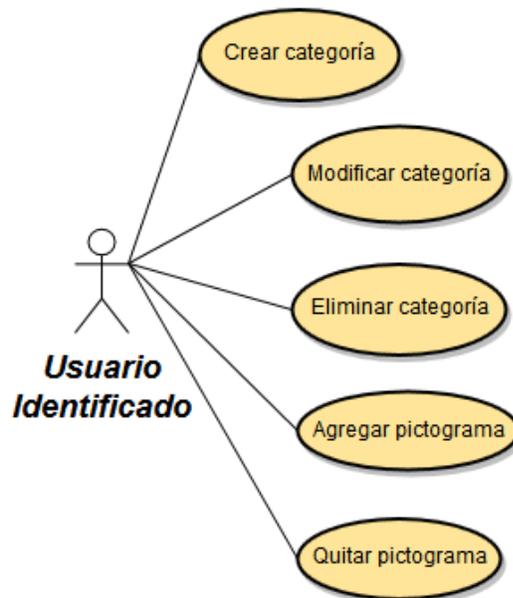


Figura 3.5 - Casos de uso para administración de categorías

Al momento de crear un mensaje, el usuario deberá seleccionar los pictogramas que formarán parte del mensaje. Para facilitar esta selección los pictogramas se mostrarán en diferentes categorías. La figura 3.5 muestra los casos de uso referidos a esta funcionalidad. El usuario deberá crear estas categorías antes de comenzar a utilizar la aplicación pero podrán modificarse en cualquier momento.

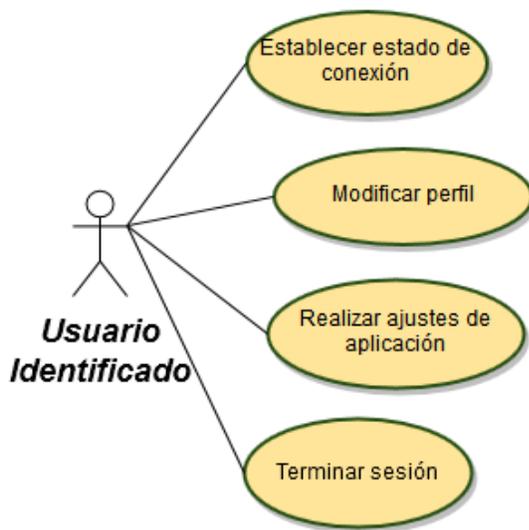


Figura 3.6 - Casos de uso de administración de estado del usuario

En la figura 3.6 se observan los casos de uso referentes a la administración del estado de conexión del usuario y su perfil y configuraciones. El usuario podrá establecer su estado de conexión: Disponible, Ocupado, etc. Podrá modificar datos de su perfil público como lo son su nombre completo y su foto. Los ajustes de aplicación se refiere a cambios en el aspecto o comportamiento de la aplicación para que se adecúe al gusto y posibilidad de cada usuario; por ejemplo: el tamaño de los pictogramas, el volumen del sonido, etc. Finalmente al terminar una sesión se culminará la sesión del usuario actual y se cerrará la aplicación. Para volver a utilizar la aplicación se deberá iniciar sesión nuevamente.

### 3.5 Arquitectura física del servicio

La figura 3.7 presenta un esquema de la arquitectura física del servicio, luego se dará un detalle de cada una de las partes que lo componen.

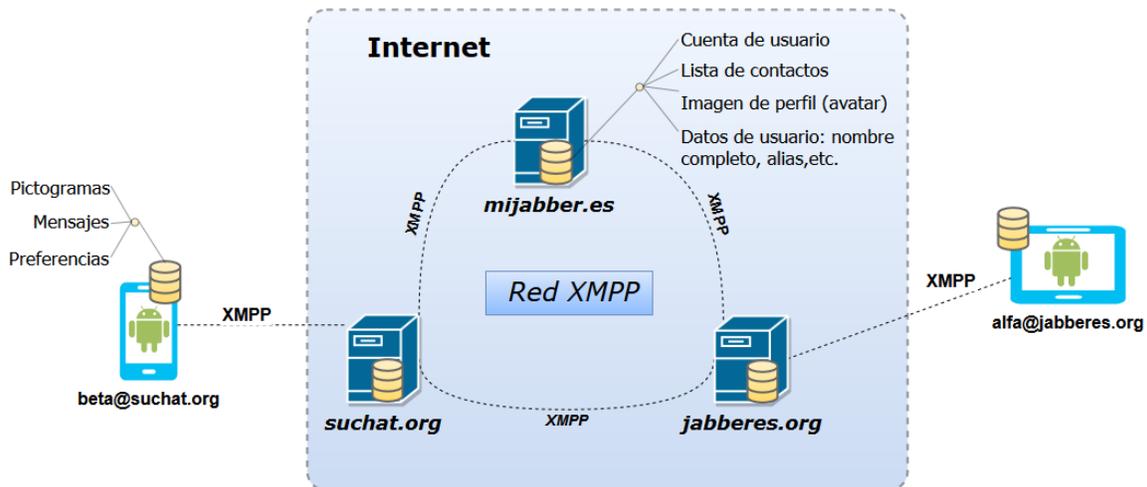


Figura 3.7 - Esquema de la arquitectura física del servicio

Los dispositivos móviles actúan como **clientes** del servicio siendo los encargados de interactuar con el usuario. Para esto se desarrolló una aplicación en *Android* que contiene la interfaz de usuario y ejecuta la lógica necesaria para la comunicación con el servidor mediante *XMPP*. La interfaz de usuario permite crear y recibir mensajes utilizando pictogramas del sistema *SPC*. Cada aplicación posee un almacenamiento local en donde guarda: un conjunto de pictogramas preestablecido, los mensajes que se han enviado o recibido y las configuraciones que el usuario haya realizado para modificar el comportamiento de la aplicación.

Para el caso de los **servidores** no se realizó ningún tipo de desarrollo. Debido a que se utilizó un estándar abierto para la comunicación, se tiene la certeza de que cualquier servidor que implemente *XMPP* se puede comunicar con los clientes que se han desarrollado. Entonces la aplicación puede utilizar cualquiera de los servidores públicos *XMPP* (como lo esquematiza la figura 3.7) o bien cualquier servidor *XMPP* propio que sea montado en internet o en una intranet. Para que un usuario pueda comenzar a utilizar la aplicación debe poseer una cuenta en el servidor al que desea conectarse. En caso de no disponer de una cuenta en el servidor, la misma puede crearse desde la aplicación (excepto en el caso de algunos servidores públicos en los que sólo se permite crear cuentas desde sus sitios web).

El hecho de utilizar el protocolo *XMPP* y respetar el estándar, tiene ventajas y desventajas. La principal ventaja es que se puede hacer uso de los servidores públicos disponibles (o incluso instalar uno propio) sin necesidad de desarrollos específicos en el

servidor por lo que el mayor esfuerzo se concentra en el desarrollo de la aplicación. La contraparte de esto es que si se desea ampliar la funcionalidad de un servicio para permitir cuestiones que el protocolo originalmente no contempla se deben realizar desarrollos específicos en el servidor o bien intentar resolverlo desde la aplicación.

Concretamente, el protocolo sólo contempla el almacenamiento en el servidor de determinados datos: la cuenta de usuario, lista de contactos, imagen de perfil, etc. No es posible almacenar imágenes, por ejemplo. Teniendo en cuenta lo anterior y para reducir la complejidad del desarrollo sin dejar de cumplir con los objetivos planteados para este proyecto se decide que la aplicación será la encargada de almacenar todos aquellos datos que el estándar no define. Quiere decir que los pictogramas, los mensajes enviados y recibidos y las configuraciones realizadas por un usuario serán almacenados localmente en cada dispositivo. Mientras que la lista de contactos, el avatar, etc., serán almacenadas en el servidor como lo establece el protocolo.

La consecuencia de lo anterior es que si un usuario en un determinado dispositivo crea, por ejemplo, determinadas categorías de pictogramas y además ha recibido y enviado mensajes, cuando intente utilizar otro dispositivo no tendrá disponible las categorías ni los mensajes ya que fueron almacenados localmente en el primer dispositivo. Lo que sí se mantendrá de dispositivo a dispositivo es la lista de contactos y datos de la cuenta de usuario como la imagen de perfil.

## 3.6. Arquitectura lógica del servicio

En las siguientes secciones se presentan los diferentes elementos que componen la arquitectura lógica del servicio.

### 3.6.1. Componentes de software

En la figura 3.8 se presenta un modelo de componentes simplificado de alto nivel de la aplicación. Cada componente consiste en un paquete o conjunto de paquetes de Java que contienen las clases necesarias para llevar adelante el funcionamiento previsto.

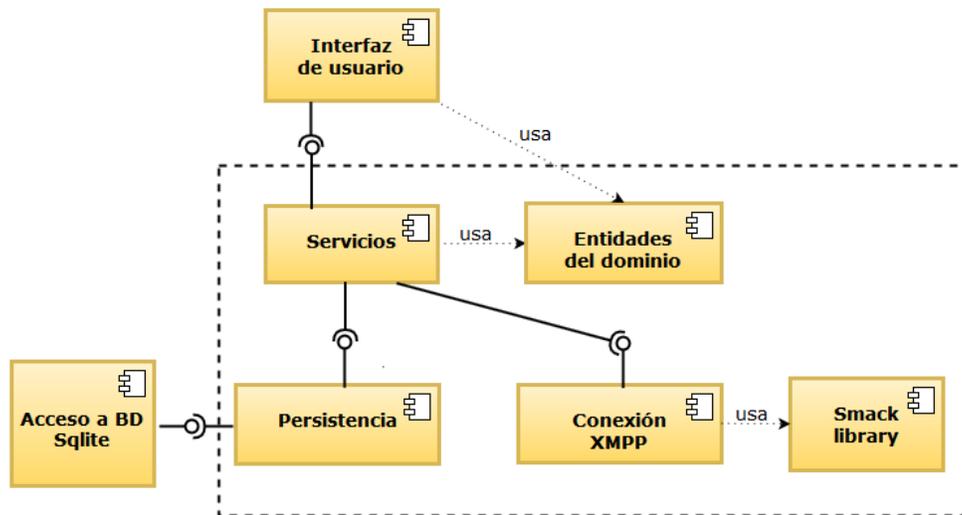


Figura 3.8 - Esquema de componentes de software

En el interior de la línea punteada se encuentran los componentes que corresponden a la lógica de negocios de la aplicación. Estos componentes fueron construidos con el objetivo de poder ser utilizados en otro proyecto (para esta plataforma) ya que no contienen dependencias a objetos de interfaz de usuario y no están ligados a ningún motor de base de datos específico. Puede verse como un gran componente que lleva adelante la lógica de la aplicación que puede ser manipulado, tanto para obtener información como para accionar determinada lógica, a través de una interfaz definida. Al desacoplar la interfaz de usuario y el acceso a datos de la lógica de negocio, es posible implementarlas de diferentes maneras según la necesidad de cada proyecto.

Para este proyecto en particular se construyó una interfaz de usuario utilizando Android SDK y se implementó la persistencia en base de datos haciendo uso de una base de datos en SQLite.

A continuación se explican brevemente cada uno de estos componentes.

- **Interfaz de usuario**

Se trata de la interfaz gráfica con la que el usuario deberá interactuar para hacer uso de la funcionalidad de la aplicación. En este proyecto se utilizó el framework Android SDK para la construcción de las interfaces de usuario.

- **Entidades del Dominio**

Es el conjunto de *entidades* que representan los conceptos utilizados para modelar el dominio del problema. Cada una de estas entidades posee determinados *atributos* y puede interactuar con otras entidades a través de sus *relaciones*.

La principal función de este componente es la de transportar información entre la interfaz de usuario y los servicios, de alguna manera las entidades pueden verse como el idioma que utilizan para comunicarse ambos componentes. Esto implica que si la interfaz de usuario desea solicitar determinada operación a algún servicio deberá esperar como resultado una o más entidades del modelo de dominio. De igual forma, si en alguna operación debe proporcionar algún parámetro, debe hacerlo a través de una entidad.

- **Servicios**

A través de los servicios se accede a la lógica de la aplicación. Los servicios se encargan de ofrecer operaciones para que se puedan llevar adelante las reglas de negocio de la aplicación. Algunos ejemplos de operaciones son: creación de una cuenta de usuario, envío de un mensaje a un determinado contacto, obtener todos los chat activos, eliminar un contacto, etc.

Cada una de estas operaciones puede hacer uso de funcionalidades que se encuentran en otros componentes como lo son la *Persistencia* y la *Conexión XMPP*. Sin embargo los detalles de la implementación de las operaciones de un servicio son transparentes para el componente que accede a ellas, públicamente sólo se exponen operaciones con una semántica clara.

- **Conexión XMPP**

Este componente se encarga de administrar la conexión con el servidor XMPP haciendo uso de la librería Smack. La principal utilidad de este componente es ocultar detalles de implementación de la comunicación XMPP al componente de servicios brindándole una interface simplificada.

- **Persistencia**

Este componente es el encargado de guardar y recuperar la información en la base de datos. Recibe las peticiones de almacenamiento o recuperación desde

el componente de servicios y devuelve los resultados; para esto ofrece una interface especializada y oculta los detalles de implementación como el motor de base de datos utilizado y la forma de acceder a los datos.

- **Acceso a BD SQLite**

Para que el componente *Persistencia* pueda llevar adelante su trabajo se le debe proveer una implementación concreta que finalmente almacene los datos en un determinado SGBD. Este componente concreto es específico de cada proyecto ya que dependerá de los SGBD de los cuales se disponga. Para este proyecto se implementó un componente que persiste los datos en SQLite.

### 3.6.2. Diagrama de clases

Debido a la cantidad de clases con las que se cuenta y con la intención de describir mejor los aspectos destacados del diseño de las clases, se mostrarán dos diagramas simplificados. El primero de ellos muestra el punto de acceso a la lógica de la aplicación, los servicios, como muestra la figura 3.9. El segundo detalla las principales entidades que se diseñaron como se muestra en la figura 3.10.

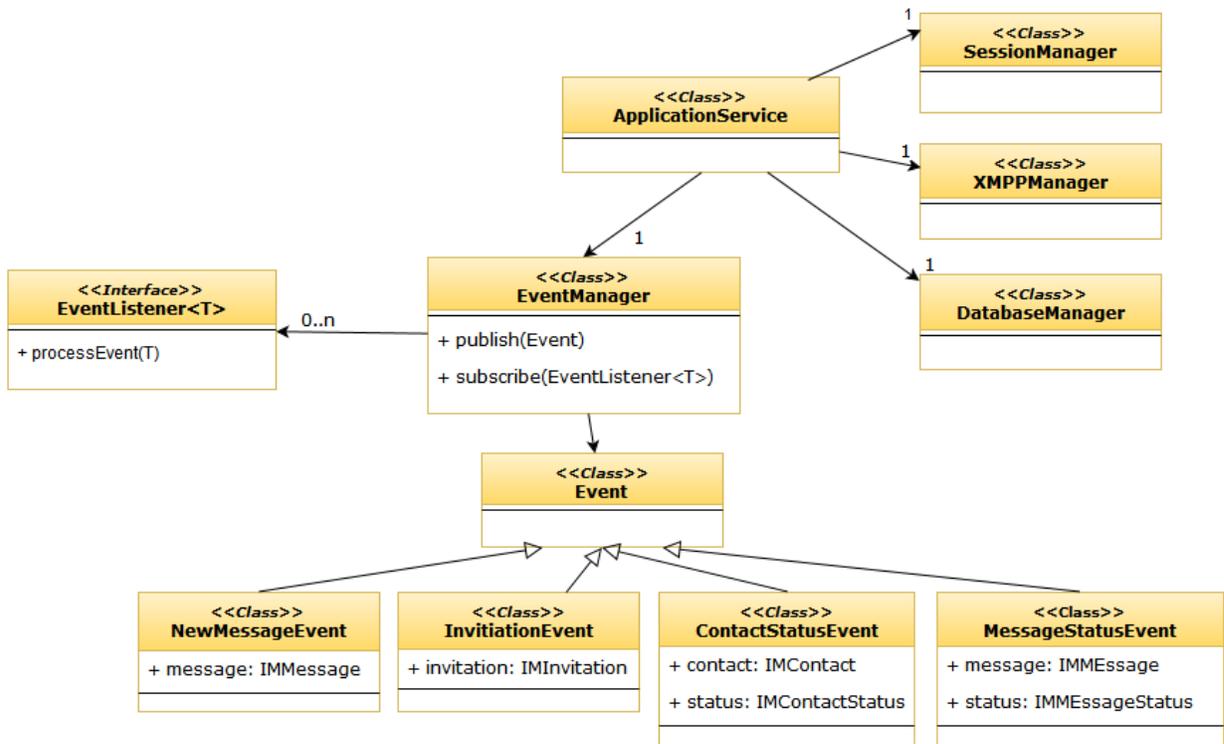


Figura 3.9 – Diagrama de clases de alto nivel de la aplicación

### ApplicationService

Esta clase forma parte del componente *Service* y es la puerta de entrada a la aplicación, contiene todas las operaciones necesarias para llevar adelante la lógica de la aplicación, desde la autenticación de un usuario hasta el envío/recepción de mensajes. Fue construido siguiendo el patrón de diseño Fachada (Gamma, y otros, 1994) utilizado para simplificar y unificar el acceso a la lógica de un sistema.

### SessionManager

Es una clase de instancia única (Singleton) que se encarga de guardar el estado de una sesión en la aplicación. Una sesión comienza cuando un usuario ingresa a la aplicación mediante un login y concluye cuando el usuario explícitamente cierra la aplicación. Algunos de los datos que administra esta clase son: el usuario que inicia sesión y la hora de inicio de sesión.

### XMPPManager

Es una clase de instancia única (Singleton) que se encarga de realizar todas las operaciones relacionadas con el protocolo XMPP, tales como: establecer la conexión con el servidor, iniciar una sesión con un determinado usuario, obtener la lista de contactos, etc. Internamente hace uso de la librería Smack para interactuar con XMPP.

### **DatabaseManager**

Es una clase de instancia única (Singleton) que realiza las operaciones de almacenamiento y recuperación de datos necesarios para que la aplicación pueda funcionar. Los usuarios de la aplicación, los contactos de cada usuario, los mensajes enviados, los símbolos pictográficos son algunos de los datos que se almacenan en la aplicación. Esta clase implementa el acceso a la base de datos SQLite.

### **EventManager**

En una aplicación de mensajería la mayoría de los eventos que suceden se dan de manera asíncrona. La llegada de un mensaje, por ejemplo, puede suceder en cualquier momento y sin que el usuario que recibe el mensaje haya realizado alguna acción deliberada para que esto ocurra.

La clase EventManager permite que cualquier objeto de la aplicación pueda suscribirse a un evento específico para ser notificado cuando ocurra. Para ello debe implementar la interfaz `EventListener<T>` y luego suscribirse al EventManager a través del método `subscribe()`. Cuando un objeto implementa la interface `EventListener<T>` debe especificar el tipo de evento a procesar (parámetro genérico T), entre los principales eventos se encuentran: `NewMessageEvent`, `InvitationEvent`, `ContactStatusEvent`, `MessageStatusEvent`. El patrón de diseño que se sigue aquí es el denominado Publish-Subscribe.

Un ejemplo típico de la utilización de este esquema es cuando se desea mostrar una notificación al arribar un nuevo mensaje. Aquí el componente que muestra la notificación deberá implementar la interface `EventListener<NewMessageEvent>` y suscribirse al evento a través de `EventManager.subscribe()`. Cuando arribe un nuevo mensaje se ejecutará el método `processEvent(event)` del objeto que se suscribió al evento y podrá mostrar una notificación o realizar cualquier otra acción.

En la figura 3.10 se muestra un diagrama de clases simplificado de las clases que componen el modelo de entidades de la aplicación.

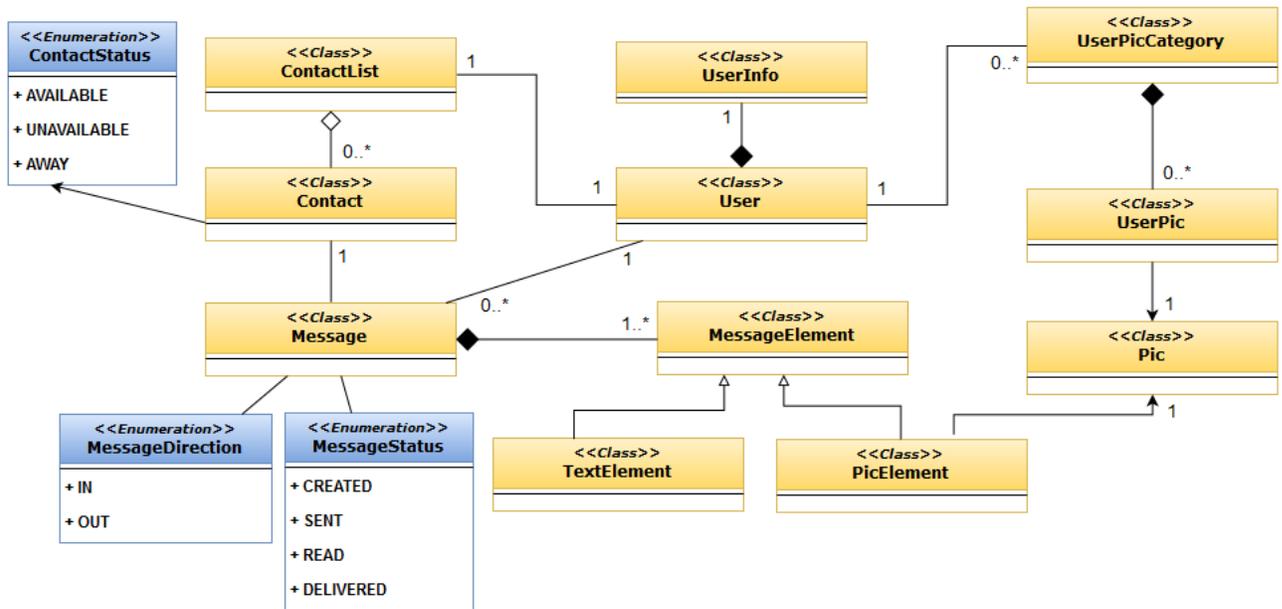


Figura 3.10 - Diagrama de clases de las entidades del dominio

La principal entidad es el mensaje representado por la clase *Message* el cual contiene las imágenes (clase *PicElement*) o el texto (clase *TextElement*) que un usuario envía o recibe de un contacto. Si es un mensaje que un usuario ha enviado, su dirección será OUT y si es un mensaje recibido la dirección será IN. Un mensaje también tiene un estado que indica cuál es su situación con respecto al envío o recepción. En la figura 3.11 se muestran los estados por los que transita un mensaje saliente.

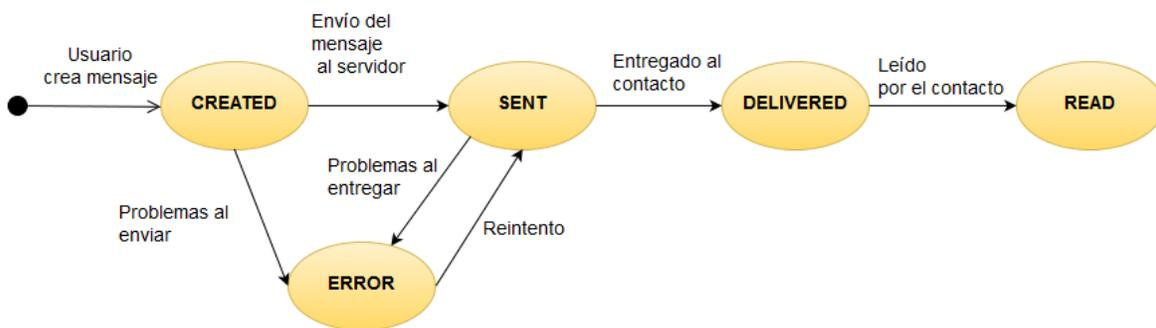


Figura 3.11- Diagrama de estados de un mensaje saliente

Un usuario (clase *User*) es aquel que inicia sesión en la aplicación y la utiliza para comunicarse con sus contactos. El atributo más relevante del usuario es su nombre de

usuario ya que lo identifica unívocamente entre todos los usuarios de la aplicación. Posee también un alias para identificarse más cómodamente pudiendo utilizar su nombre, seudónimo, apodo, etc. La clase *UserInfo* contiene datos adicionales del usuario como su imagen de perfil o sus datos personales.

Un contacto (clase *Contact*) no es más que otro usuario que forma parte de la lista de contactos (clase *ContactList*) de un usuario. Cada contacto tiene tres estados de conexión posible: Disponible (AVAILABLE), No disponible (UNAVAILABLE) y Fuera (AWAY). Los atributos más relevantes de un contacto son su nombre de usuario, su alias, su imagen de perfil y su estado de conexión. Para que dos usuarios puedan comunicarse ambos deben aceptar formar parte de sus respectivas listas de contactos, este procedimiento se explicará con más detalle en secciones posteriores.

Finalmente la clase *Pic* representa un pictograma provisto por la aplicación y posee como atributos relevantes una imagen, un texto descriptivo y un texto específico para ser sintetizado a voz. Cada usuario debe definir categorías de pictogramas (clase *UserPicCategory*) que son utilizadas para agilizar la selección de pictogramas al momento de la creación de un mensaje. Cada pictograma que el usuario agrega a una categoría se representa mediante la clase *UserPic* que si bien está relacionado con la clase *Pic*, tiene atributos adicionales como el orden que ocupa dentro de la categoría.

### 3.6.3. Almacenamiento de Pictogramas

La funcionalidad principal de la aplicación es la de poder enviar y recibir mensajes contruidos en base a pictogramas. Un pictograma se compone básicamente de una imagen, un texto descriptivo y un texto para ser sintetizado por voz, tal como lo esquematiza la figura 3.12.

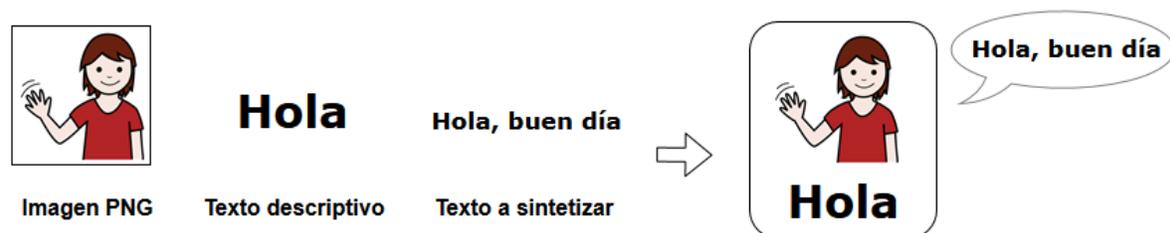


Figura 3.12 - Composición de un pictograma

Los pictogramas que utiliza la aplicación son instalados al momento de iniciarse por primera vez y en principio no se podrán agregar más pictogramas. Esto implica que todos los usuarios de la aplicación tendrán el mismo conjunto de pictogramas.

El almacenamiento de los pictogramas se concreta mediante la utilización de un directorio en el sistema de archivos y una tabla en la base de datos. Por un lado, la imagen de cada pictograma es un archivo de tipo PNG por lo que se guarda en un directorio específico de la aplicación. Por otro lado se mantiene una tabla en la base de datos que permite almacenar los demás atributos de cada pictograma y relacionarlos con la ubicación de la imagen. En la figura 3.13 se muestra la tabla **Pic** que guarda los datos de cada pictograma.

Pic	
PK	ID
	name
	text
	path
	voice
	order
	categoryID

Figura 3.13 - Estructura de la tabla que almacena un pictograma

En donde:

- **ID**: Identificador único del pictograma. Un pictograma siempre tiene el mismo ID ya que se establece en tiempo de diseño de la aplicación.
- **name**: Nombre del pictograma
- **text**: Texto que se muestra generalmente debajo de la imagen del pictograma
- **path**: Ruta del archivo en donde se encuentra la imagen del pictograma
- **voice**: Texto que se sintetiza a voz si el usuario lo requiere
- **order**: Orden que ocupa en la categoría que tiene asignada
- **categoryID**: Identificador de la categoría a la cual pertenece (las categorías son almacenadas en una tabla creada para tal fin)

### 3.6.4. Extensión del protocolo XMPP

El protocolo XMPP, como se explicó anteriormente, establece la manera en que se comunican los clientes de mensajería con sus servidores para lograr la comunicación en tiempo real. Una de las partes más importantes de ese protocolo es el envío de un mensaje de un usuario a su contacto y que se implementa utilizando el elemento `<message>` como se detalló en la sección 2.6.

El objetivo de la aplicación es utilizar XMPP para enviar y recibir mensajes utilizando pictogramas. Sin embargo, en los RFC que estandarizan el protocolo no se provee ninguna forma de enviar imágenes o archivos, ni de conformar un mensaje de otra forma que no sea a través de texto. Afortunadamente XMPP fue pensado para ser extensible y es posible definir estructuras de datos XML propias para que viajen dentro del paquete `<message>`. Estas estructuras se denominan extensiones y deben tener asociado un espacio de nombres propio (atributo `xmlns` en XML). Sus etiquetas, atributos y significado, así como las reglas para su procesamiento son definidas por quien la crea.

Con una extensión es posible ampliar la funcionalidad de XMPP siempre que tanto el emisor como el receptor entiendan la extensión para poder tratarla correctamente. En el caso de este proyecto son los clientes los encargados de generar la extensión cuando se envíe un mensaje y de procesarla adecuadamente al momento de recibir un mensaje. Como la extensión viaja dentro de un elemento `<message>`, el servidor puede entregar el mensaje al destinatario como lo hace normalmente (directamente ignora la extensión) por lo que no es necesario realizar ningún tipo de ajuste en su lógica.

A continuación se muestra el esquema XML de extensión que viaja en cada uno de los mensajes:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="pics">
    <xs:complexType>
      <xs:sequence>
        <xs:choice maxOccurs="unbounded" minOccurs="1">
          <xs:element name="pic">
            <xs:complexType>
              <xs:attribute name="id" type="xs:integer" use="required" />
              <xs:attribute name="text" type="xs:string" />
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

        <xs:attribute name="speech" type="xs:string" />
    </xs:complexType>
</xs:element>
</xs:choice>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

Básicamente se tiene un elemento raíz `<pics>` el cual puede contener muchos elementos `<pic>` y `<string>`. Un elemento `<pic>` representa un pictograma del mensaje y tiene como atributo obligatorio un `id` que identifica a la imagen que se envía. Los atributos `text` y `speech` representan el texto descriptivo y el sintetizado respectivamente. Estos últimos atributos no son obligatorios ya que si no se envían el destinatario utilizará sus datos locales del pictograma.

En la figura 3.14 se muestra un ejemplo de un mensaje que puede generarse desde la aplicación y que combina pictogramas y texto.



Figura 3.14 - Mensaje que combina pictograma y texto

Al momento de enviar el mensaje a su destinatario la estructura XML saliente tendrá la siguiente forma:

```

<message
  from='alfa@jabber.es'
  to='beta@jabber.es'>
  <body></body>
  <pics xmlns="pictogram:pics">
    <pic id="124" text="hola" speech="hola, buen día" />
    <string> bienvenido </string>
  </pics>
</message>

```

Como se observa, la extensión está empaquetada dentro de un `<message>` utilizando el espacio de nombres que se ha denominado *pictogram:pics*. El destinatario deberá decodificar el mensaje de la siguiente manera: el elemento `<pic>` representa un pictograma por lo que buscará su base de datos local el pictograma con *id* igual a 124 y así obtener la imagen del mismo, además utilizará los atributos *text* y *speech* para componer la representación visual y sonora del pictograma. Si estos atributos no tienen ningún valor, se obtendrán los valores desde la base de datos local.

Para decodificar el elemento `<string>` simplemente se leerá su contenido y se concatenará con el pictograma ya decodificado para así obtener el mensaje completo. El elemento `<body>` se ignorará.

## Capítulo 4

### Implementación de la aplicación

En este capítulo se describen las principales funcionalidades que provee la aplicación y se detalla la forma en que los usuarios deben interactuar con las diferentes interfaces gráficas.

#### 4.1. Identificación de la aplicación

Toda aplicación implementada en Android debe poseer un nombre y un ícono para distinguirla de las demás aplicaciones instaladas en el sistema. Tanto el nombre como el ícono son utilizados por Android en diferentes ocasiones, por ejemplo cuando se informa al usuario una alerta o una notificación relacionadas con la aplicación.

El nombre escogido para la aplicación desarrollada es *Pictogram*, y el ícono que la identifica es el que se muestra en la figura 4.1.



*Figura 4.1 - Icono y nombre de la aplicación*

#### 4.2. Inicio de sesión

Una vez que se ha iniciado la aplicación se mostrará la pantalla de inicio de sesión que se observa en la figura 4.2.



Figura 4.2 - Pantalla de inicio de sesión en la aplicación

Esta pantalla permite iniciar sesión en un servidor utilizando una cuenta de usuario. Este inicio de sesión es un requisito necesario para comenzar a utilizar la aplicación. Si no se posee alguna cuenta, es posible crearla desde la aplicación como se detalla en la sección '4.2.1. Creación de una cuenta de usuario'.

Para iniciar sesión se deben completar los campos "Nombre de usuario" y "Contraseña" ambos obligatorios. El primero de los campos se refiere al nombre de usuario o cuenta XMPP, que tiene la forma *usuario@servidor.com*. El campo "Contraseña" es la clave de dicha cuenta XMPP.

El botón "Ingresar" permite iniciar la sesión con los datos proporcionados. Si el usuario y la clave son válidas, la sesión se inicia y es posible comenzar a utilizar la aplicación, en caso contrario se informa el problema mediante un mensaje.

#### 4.2.1. Creación de una cuenta de usuario

Desde la aplicación es posible crear una cuenta de usuario en un servidor XMPP. Para acceder a la pantalla de creación de cuenta se debe presionar la opción "Crear una nueva cuenta" que se encuentra en la pantalla de inicio de sesión como se observa en la figura 4.3.

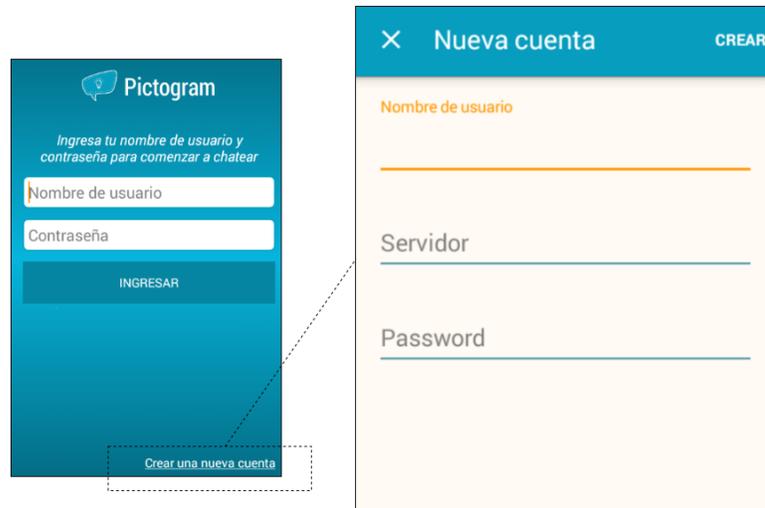


Figura 4.3 - Pantalla de creación de una nueva cuenta de usuario

Aquí se debe proporcionar obligatoriamente un nombre de usuario, un servidor en donde se creará la cuenta y la contraseña que tendrá la misma. El botón “Crear” permite confirmar la creación. Si la creación es exitosa se iniciará la sesión en la aplicación, en caso contrario se informará mediante un mensaje y permanecerá en esta pantalla. En todo momento se puede cerrar la pantalla para volver a la pantalla de inicio de sesión mediante el botón [  ].

**NOTA:** Algunos servidores públicos (ej.: *mijabber.es*, *suchat.org*) no permiten la creación de cuentas desde aplicaciones externas, en cambio ofrecen un sitio web específico desde donde es posible hacerlo. En estos casos, la creación de cuentas desde la aplicación presentará un mensaje de error.

#### 4.2.2. Duración de una sesión

Una vez que un usuario ha iniciado sesión, la misma permanecerá activa incluso si se cierra la aplicación (por ejemplo, quitándola de la lista de aplicaciones abiertas del dispositivo) o se reinicia el dispositivo. La única forma de terminar la sesión es mediante la opción “Cerrar sesión” (Ver ‘4.5.5 Cierre de sesión’). El hecho de mantener la sesión activa hasta que el usuario explícitamente decida cerrarla, evita que éste tenga que iniciar sesión cada vez que desee utilizar la aplicación.

Mientras un usuario tenga una sesión iniciada podrá continuar recibiendo mensajes e invitaciones, independientemente si la aplicación se encuentra abierta o no. Si se abre la

aplicación con una sesión iniciada, se ingresará sin pasar por la pantalla de inicio de sesión. Si se apaga el dispositivo y la sesión se encontraba iniciada, la misma volverá a iniciarse automáticamente cuando el dispositivo se encienda.

Cuando un usuario inicia una sesión aparece un icono en la barra de estado y una notificación asociada (figura 4.4). Ambos elementos indican que un usuario posee una sesión activa en la aplicación y no se irán a menos que éste la cierre explícitamente. Si se toca sobre la notificación, la aplicación se abrirá.

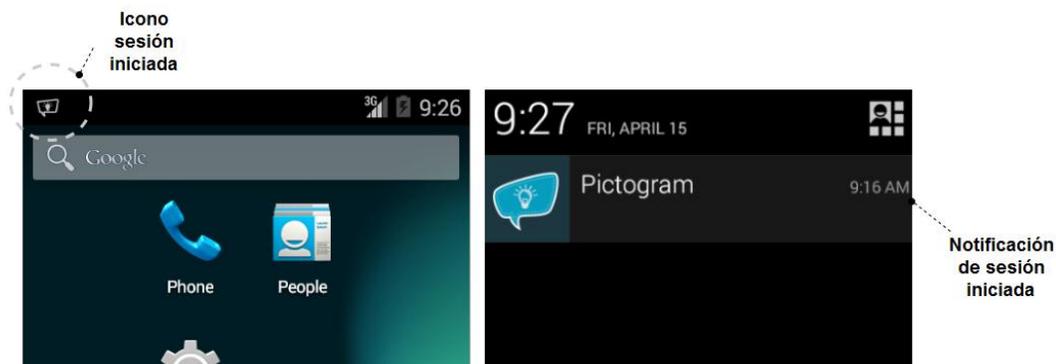


Figura 4.4 - Icono y notificación de inicio de sesión

### 4.3. Lista de contactos

La primera pantalla que se muestra al iniciar la sesión, es la lista de contactos del usuario. Esta lista muestra los contactos que el usuario ha decidido aceptar a través de la suscripción mutua (ver sección '4.5.2. Contactos'), es decir, aquellos contactos con los que explícitamente acordó mantener una comunicación.

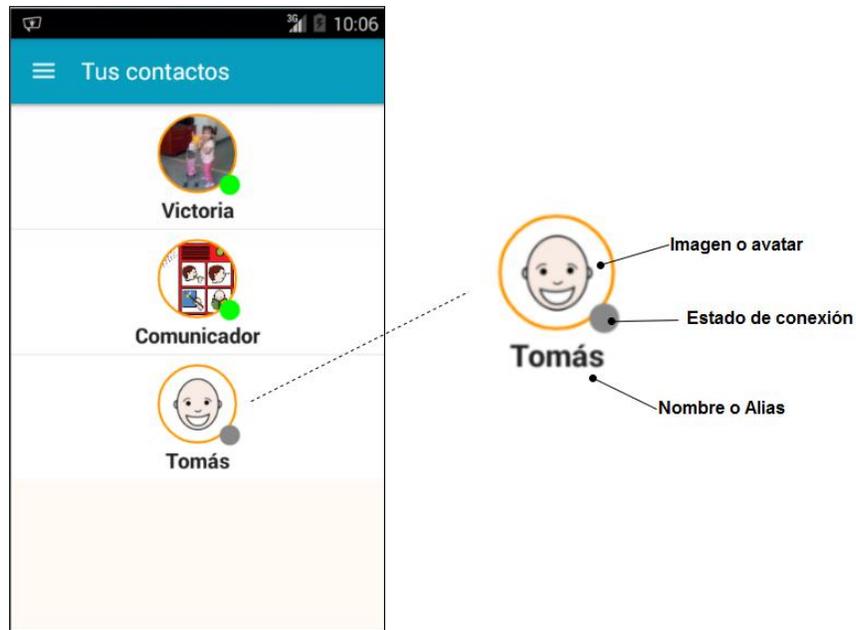


Figura 4.5 - Lista de contactos y propiedades de un contacto

Cada contacto tiene un nombre, un estado de conexión y un avatar como muestra la figura 4.5. El nombre o alias lo establece el usuario para cada contacto y es posible modificarlo en cualquier momento. El estado de un contacto indica la disponibilidad actual para establecer la comunicación. La tabla 4.1 muestra los tres diferentes estados definidos para un contacto.

Tabla 4.1 - Posibles estados de un usuario

	<b>Conectado:</b> El contacto ha iniciado sesión y se encuentra disponible para chatear
	<b>Inactivo:</b> El contacto ha iniciado sesión pero no ha utilizado la aplicación durante un tiempo prolongado
	<b>Desconectado:</b> El contacto no ha iniciado sesión.

Finalmente el avatar es una foto, un dibujo, etc., que el contacto ha establecido como su imagen de perfil.

#### 4.4. Conversaciones

Para iniciar una conversación (chat), el usuario debe seleccionar el contacto deseado desde la lista de contactos. Luego de esto se mostrará la pantalla de chat que se observa en la figura 4.6.

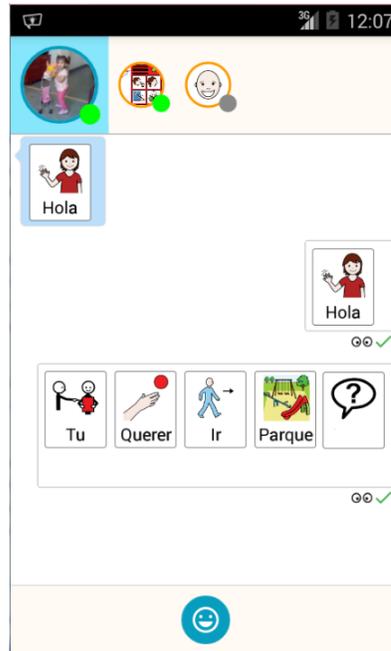


Figura 4.6 - Pantalla de chat

En la parte superior de esta pantalla se muestran los contactos en el mismo orden que se muestran en la lista de contactos. El contacto seleccionado aparecerá resaltado. Esta disposición de los contactos permite que el usuario no tenga que volver nuevamente a la lista de contactos para comenzar o continuar una conversación, simplemente lo escoge de la lista superior. Al seleccionar un contacto, su conversación se muestra en la zona de mensajes.

En el centro de la pantalla se encuentra la zona de mensajes en donde se muestran los mensajes que se han enviado y recibido de un contacto. Los mensajes más recientes se muestran en la parte inferior de la lista.

Finalmente, en la parte inferior se encuentra la zona de selección de pictogramas en donde el usuario podrá componer un mensaje mediante pictogramas y texto. La selección de pictogramas se puede realizar en pantalla completa o bien en la misma pantalla de chat; ambos tipos de selección se explican en la sección '4.4.2. Creación de mensajes'.

#### 4.4.1. Formato de los mensajes

Los mensajes, tanto recibidos como enviados, se ubican dentro de la zona de mensajes. Un mensaje recibido se alinea a la izquierda de la pantalla, mientras que un mensajes recibido se alinea a la derecha, ambos tipos de mensajes utilizan colores diferentes como muestra la figura 4.7.

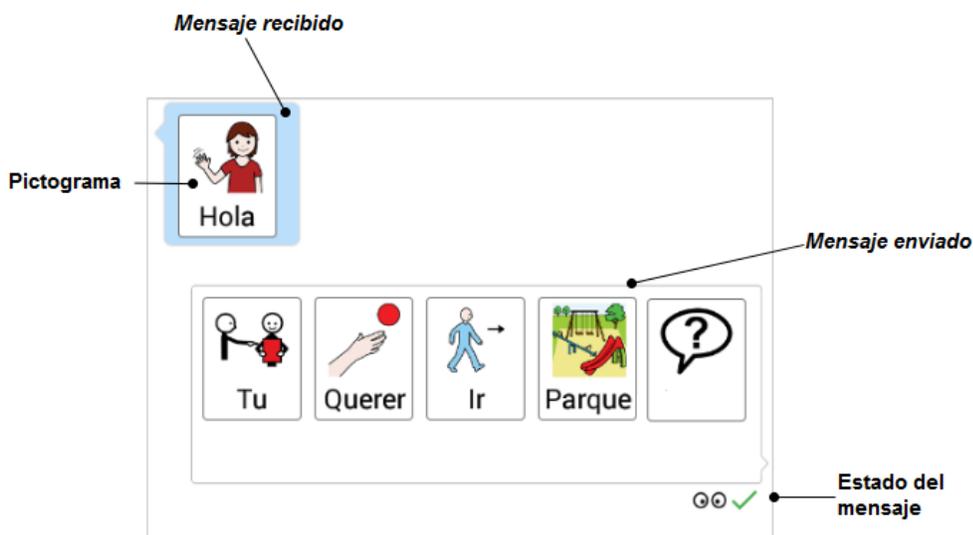


Figura 4.7 – Formato de mensajes enviados y recibidos

Un mensaje puede contener pictogramas, texto o cualquier combinación de ambos. Cada pictograma puede contener un texto descriptivo que se ubicará debajo de la imagen. Un mensaje enviado tiene la particularidad de tener un estado que indica si el mensaje ha sido enviado y si ha sido leído por el contacto. En la tabla 4.2 se detallan los posibles estados de un mensaje enviado y su correspondiente símbolo gráfico.

Tabla 4.2 – Estados de un mensaje

Símbolo	Significado
	<b>Enviando:</b> El mensaje se encuentra en proceso de envío.
	<b>Enviado:</b> El mensaje ha sido enviado exitosamente al servidor del destinatario pero éste aún no lo ha entregado al contacto.

	<p><b>Entregado:</b> El mensaje ha sido entregado al dispositivo del contacto pero aún no ha sido leído.</p>
	<p><b>Leído:</b> El mensaje ha sido leído por el contacto.</p>
	<p><b>Error:</b> Cuando el proceso de envío no puede completar la operación por algún motivo.</p>

#### 4.4.2. Creación de mensajes

Para la creación de mensajes se dispone de dos opciones: creación en pantalla completa o creación en la misma pantalla de chat; la opción a utilizar debe establecerse mediante configuración (ver sección ‘4.5.4. Ajustes’). La utilidad de ambas opciones depende del tipo de usuario que utilice la aplicación y el tamaño del dispositivo con el que se cuente.

Para un dispositivo con dimensiones generosas es conveniente crear el mensaje en la misma pantalla de chat porque eso permite tener siempre a la vista la conversación actual, pero si el usuario no tiene demasiada precisión fina en el toque de pantalla quizás prefiera crear el mensaje en una pantalla aparte en donde los pictogramas pueden tener mayores dimensiones.

Para la creación de un mensaje en pantalla completa se debe presionar el botón [  ] que se encuentra en la pantalla de chat como lo muestra la figura 4.8. Mientras que para la creación de un mensaje en la misma pantalla de chat se debe tocar el cuadro de texto o el botón [  ] que se encuentran en la pantalla de chat como lo muestra la figura 4.9.

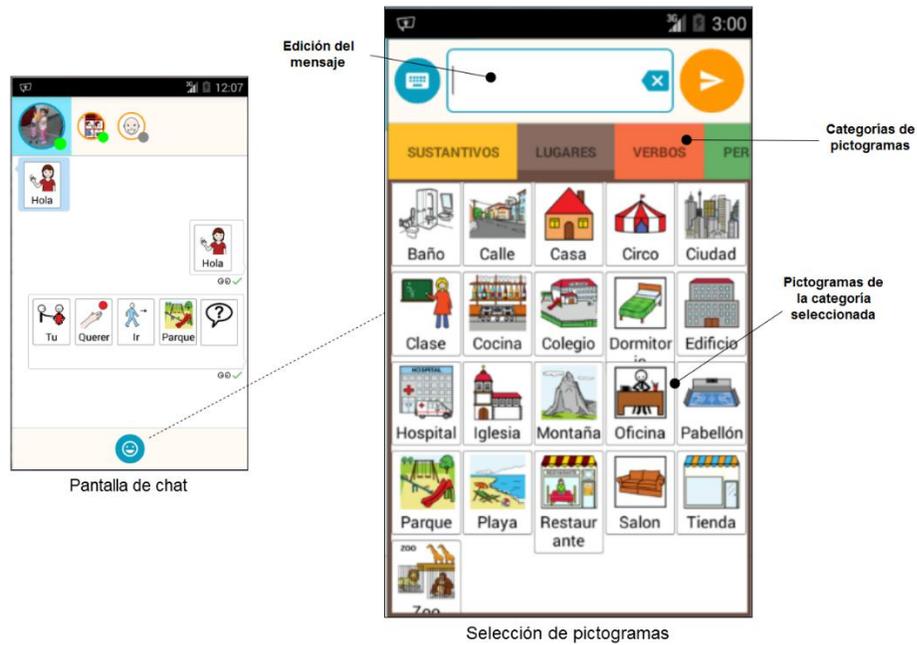


Figura 4.8 - Creación de mensaje en pantalla completa

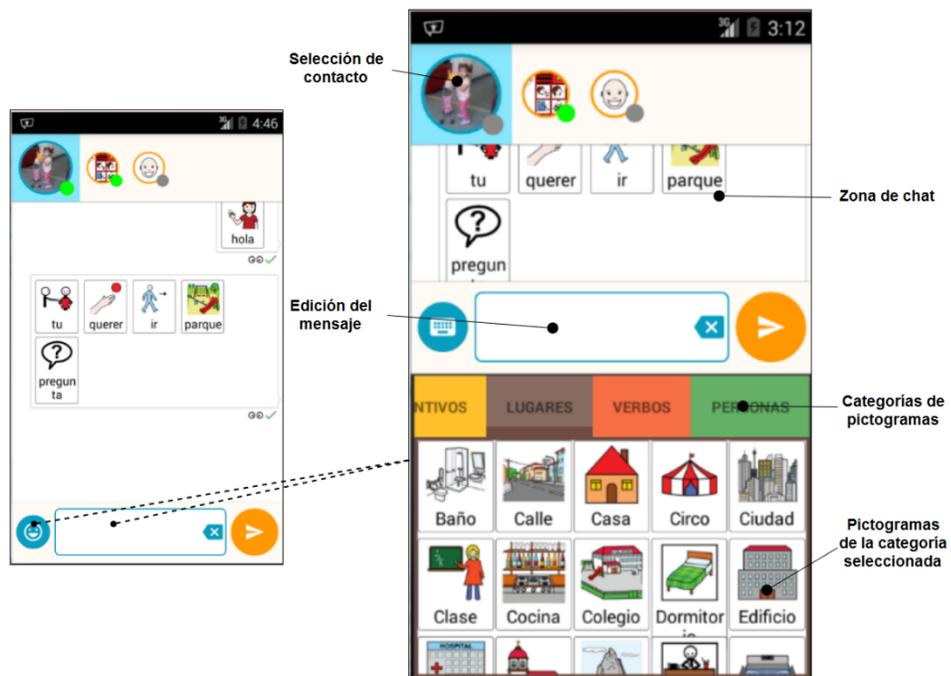


Figura 4.9 - Creación de un mensaje en la misma pantalla de chat

Para crear un nuevo mensaje el usuario puede hacer uso de los pictogramas disponibles o bien escribir mediante el teclado. Los pictogramas se organizan en categorías

(ver sección '4.5.3. Categorías de pictogramas') previamente configuradas las cuales se muestran en pestañas debajo de la zona de edición del mensaje.

Para visualizar los pictogramas pertenecientes a una categoría, el usuario debe tocar alguna de las pestañas de categorías, inmediatamente los pictogramas de esa categoría se mostrarán debajo. Para agregar un pictograma al mensaje, se debe tocar sobre él. A medida que se va construyendo el mensaje, los pictogramas se ubican en el cuadro de texto de la zona de edición como muestra en la figura 4.10.



Figura 4.10 - Creación de un mensaje

Para introducir texto se debe utilizar el teclado por defecto del sistema (figura 4.11) el cual se muestra al presionar el botón [  ]. Para ocultar el teclado se deberá presionar nuevamente el botón [  ].

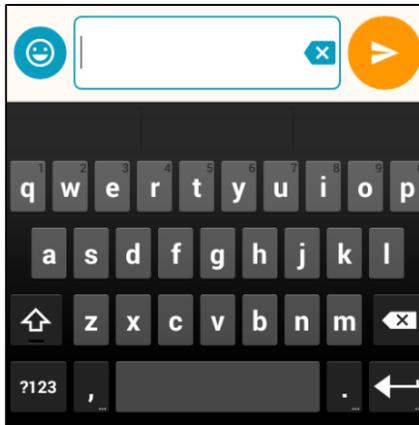


Figura 4.11 - Teclado para introducir texto en un mensaje

En cualquier momento el usuario puede borrar el contenido que haya introducido (pictograma o texto) mediante el botón [  ], que borra la letra o pictograma que se encuentra a su izquierda. El cuadro de texto de la zona de edición permite ubicar el cursor en cualquier lugar del mensaje por lo que es posible borrar un solo pictograma o una porción de texto de una ubicación arbitraria del mensaje. En la figura 4.12 se muestra un mensaje compuesto por dos pictogramas, y se observa que el usuario ha ubicado el cursor luego del primer pictograma.



Figura 4.12 - Ubicación del cursor dentro de un mensaje

Una vez que el usuario ha construido el mensaje, puede enviarlo presionando el botón [  ] lo cual colocará el mensaje al final de la conversación actual y dejará listo el cuadro de edición para crear un mensaje nuevo.

#### 4.4.3. Recepción de un mensaje

Cuando un contacto envía un mensaje, el usuario es notificado de diferentes maneras. Si el usuario se encuentra en la pantalla de chat entablando una comunicación con el usuario que envía el mensaje, dicho mensaje se muestra al final del chat, titilando unos segundos como muestra la figura 4.13. Adicionalmente se reproducirá un sonido y una vibración las cuales pueden ser modificadas mediante configuración (ver sección '4.5.4. Ajustes').



Figura 4.13 - Llegada de un mensaje nuevo

En cambio, si un mensaje es recibido cuando el usuario se encuentra en otra pantalla de la aplicación, como por ejemplo, manteniendo una conversación con otro contacto (diferente al que envía el mensaje), se mostrará una **notificación** del sistema. Una notificación es un componente de la plataforma Android que permite informar al usuario sobre eventos que ocurren en una aplicación y que se muestra fuera de la interfaz gráfica de la misma, más precisamente en la barra de estado del sistema. Cuando Android muestra una notificación aparece un pequeño ícono en la barra de estado. Para ver los detalles de la notificación se debe desplegar el área de notificaciones del sistema.

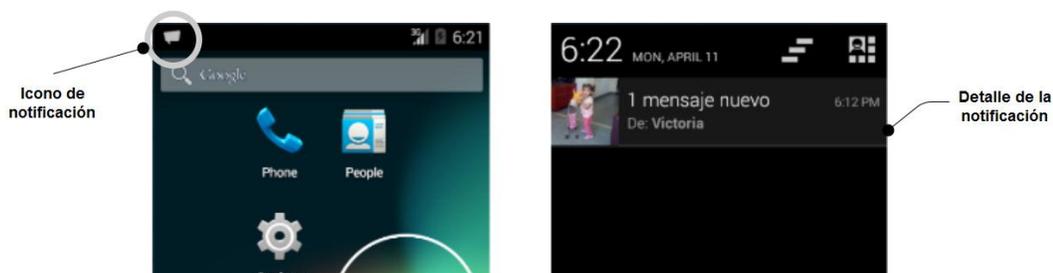


Figura 4.14 - Notificación de un mensaje nuevo

Cuando la aplicación informa la llegada de un nuevo mensaje mediante una notificación, se observa el ícono [  ] en la barra de estado (figura 4.14). Al desplegar el área de notificaciones se muestra el cuerpo de la notificación con el detalle de la cantidad de mensajes recibidos y el nombre y avatar del contacto que los envía. En caso de recibir mensajes de más de un contacto, se mostrará sólo la cantidad de mensajes recibidos y un avatar genérico. Al tocar sobre el cuerpo de la notificación se abre la pantalla de chat del contacto.

Una notificación, además de la interfaz gráfica, tiene asociado un sonido, una vibración y una indicación con luz led (si el dispositivo lo permite). Estos tres aspectos pueden modificarse por configuración de la aplicación como se indica en la sección '4.5.4. Ajustes'.

## 4.5. Administración

La administración de la aplicación permite realizar configuraciones básicas que son necesarias para comenzar a entablar conversaciones, como la creación de categorías de pictogramas y el agregado de contactos, como así también proporciona una manera de ajustar la aplicación para que se adapte al gusto y necesidades del usuario. Generalmente la administración está destinada al usuario **administrador**, es decir, aquella persona encargada de guiar al **usuario final** en el proceso de aprendizaje y que puede ser un familiar, un maestro, un psicopedagogo, etc.

Para acceder a la administración de la aplicación se debe tocar el botón [  ] disponible en la lista de contactos como se muestra en la figura 4.15. En la parte superior del menú se muestran los datos personales del usuario mientras que en la parte inferior se

muestran los accesos a los diferentes ítems de administración. A continuación se detallan cada uno de ellos.

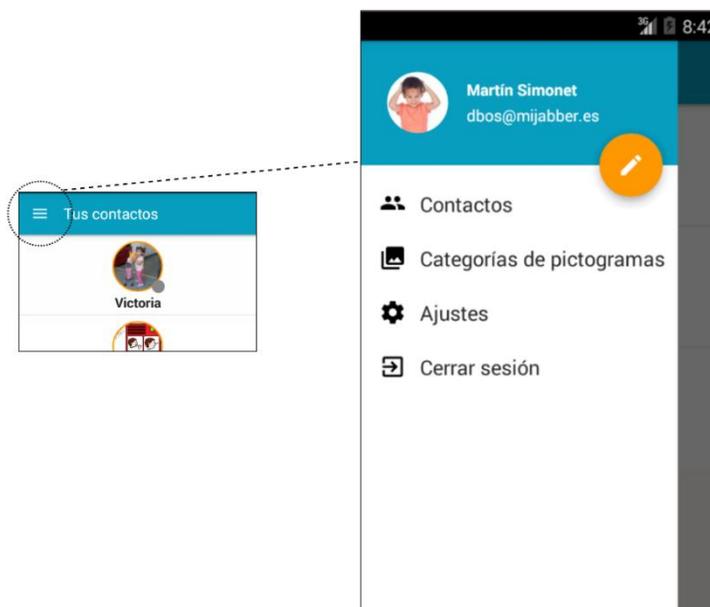


Figura 4.15 - Menú de administración de la aplicación

#### 4.5.1. Perfil del usuario

Un usuario puede configurar su nombre, apellido y avatar. Estos datos se almacenan tanto en la aplicación como en el servidor XMPP (ya que es parte del estándar XMPP) por lo que si el usuario inicia sesión, por ejemplo, en otro dispositivo, se podrá obtener la información guardada en el servidor. Estos datos se utilizan para mostrar a los demás usuarios con los cuales se tenga contacto.

Para modificar la información de perfil se debe tocar el botón [  ] que se encuentra ubicado en la parte superior del menú de administración.

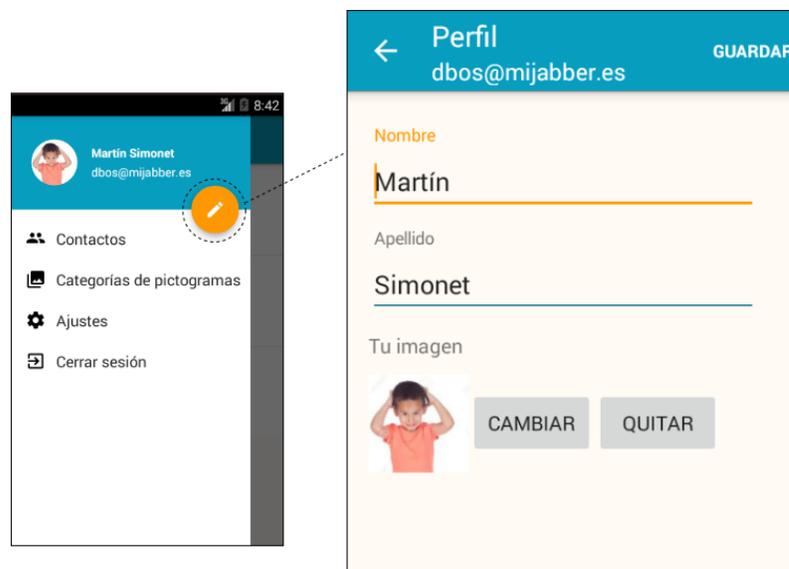


Figura 4.16 - Edición del perfil

Todos los campos (“Nombre”, “Apellido” y “Tu imagen”) son opcionales. El campo “Tu imagen” permite seleccionar cualquier imagen almacenada en el dispositivo a través del botón “Cambiar”; mediante el botón “Quitar” se elimina la imagen actual quedando por defecto un avatar genérico. Para guardar los cambios se debe presionar “Guardar”.

#### 4.5.2. Contactos

Los contactos, en el contexto de la aplicación, son otros usuarios con los cuales habitualmente se mantienen conversaciones y que han sido agregados a la **lista de contactos** mediante el procedimiento de **suscripción mutua**. La suscripción mutua se presenta cuando dos usuarios aceptan mantener explícitamente, el uno con el otro, un vínculo para la comunicación; un usuario es el que invita a suscribirse y el otro la acepta o la rechaza. La suscripción mutua perdura hasta que alguna de las partes decide terminar la suscripción. Todos los contactos con los cuales se tiene suscripción mutua se denomina **lista de contactos**.

La aplicación permite construir una lista de contactos a través del procedimiento de suscripción mutua. Para acceder a la administración de contactos se debe ingresar al ítem “Contactos” del menú de administración como muestra la figura 4.17.

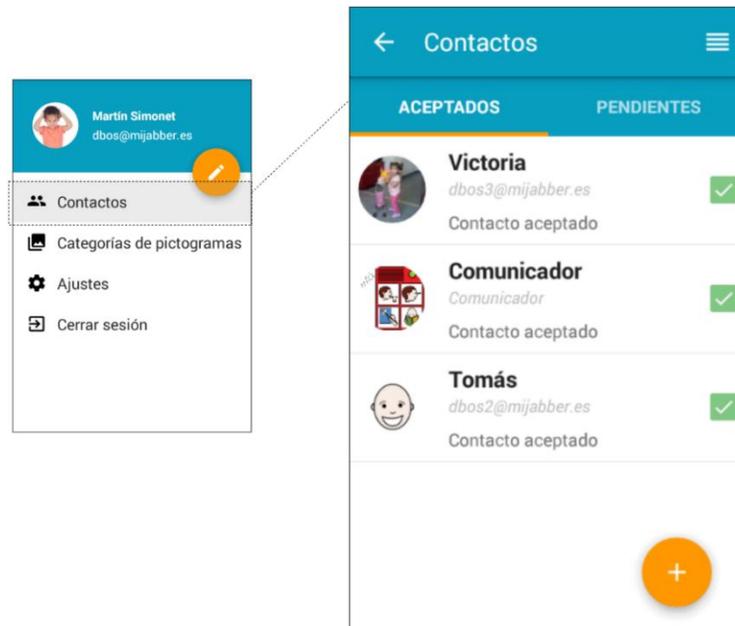


Figura 4.17 - Administración de contactos

La administración de contactos presenta una pantalla con dos pestañas. La pestaña “Aceptados” contiene los contactos que forman parte de la lista de contactos, mientras que la pestaña “Pendientes” agrupa los contactos que por algún motivo aún no han sido agregados a la lista de contactos. Sólo los contactos que se encuentran como “Aceptados” serán aquellos con los que el usuario podrá comunicarse. No se recibirán mensajes ni se podrá establecer comunicación con los contactos “Pendientes”.

El orden de los contactos en la lista de contactos puede modificarse mediante el botón [☰] y luego desplazando el ítem a la posición que se desee (figura 4.18). Finalmente para confirmar el orden se debe presionar “Aceptar”. Este orden se respetará tanto en la pantalla inicial de contactos del usuario como en la pantalla de chat (figuras 4.5 y 4.6 respectivamente).

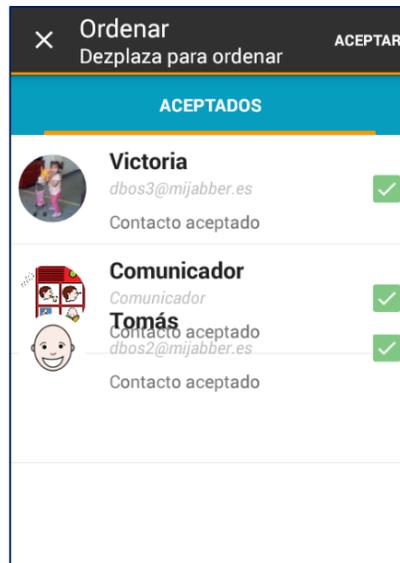


Figura 4.18 - Cambio en el orden de un contacto

### Añadir un contacto

La creación de un contacto es la primera parte del procedimiento de suscripción mutua ya que, inmediatamente después de crear el contacto, se le envía una solicitud de suscripción (o sencillamente *invitación*), invitándolo a mantener un vínculo de comunicación. La suscripción mutua se completa si el contacto acepta esta invitación.

Para añadir un contacto a la lista de contactos se debe tocar el botón [  ] que se encuentra en la administración de contactos; allí se desplegará el menú que muestra la figura 4.19.

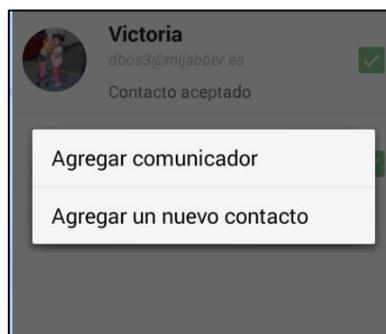


Figura 4.19 - Menú para agregar un nuevo contacto

El ítem “Agregar comunicador” permite añadir un contacto especial denominado *Comunicador* que se explicará más adelante en la sección ‘*Añadir un comunicador*’. El ítem “Agregar nuevo contacto” abre la pantalla que se muestra en la figura 4.20 que permite agregar un nuevo contacto.

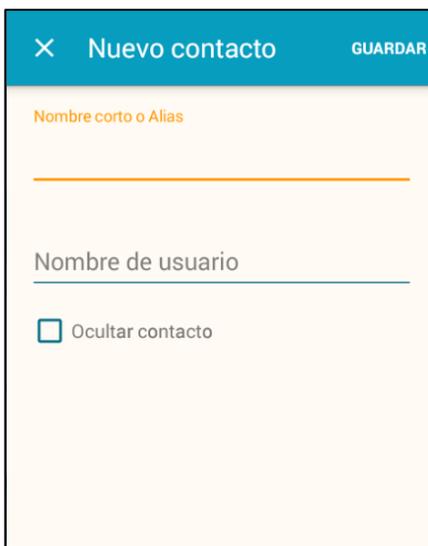


Figura 4.20 - Pantalla para la creación de un nuevo contacto

Para la creación de un contacto son necesarios dos campos obligatorios, el nombre corto o alias del contacto y su nombre de usuario. El nombre corto o alias es el texto que se muestra junto al avatar del contacto en las diferentes pantallas de la aplicación y que sirve para identificarlo rápidamente. El nombre de usuario es la denominación de la cuenta XMPP del contacto y que tiene la forma *usuario@servidor.com*.

Para confirmar la creación del contacto se debe presionar el botón “Guardar”, momento en el cual suceden dos cosas: a) automáticamente se le envía una solicitud de suscripción al contacto y b) la información del contacto se almacena y se muestra como un ítem dentro de la lista de contactos “Pendientes”.

En la figura 4.21 se muestra un contacto recientemente creado con el alias “Exequiel” y el nombre de usuario “exequiel@suchat.org”. El contacto aún no ha respondido a la invitación por lo que se deberá esperar su aceptación o rechazo. Se observa una leyenda “*Enviaste una invitación pero aún no ha sido aceptada*”.

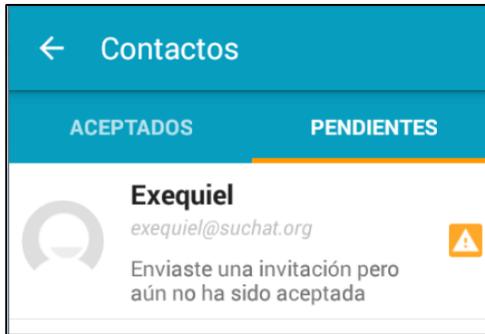


Figura 4.21 - Ejemplo del contacto 'Exequiel' recién creado

Si el contacto rechaza la invitación, la leyenda cambiará a la siguiente: “*Debes enviar una invitación para comenzar a hablar con este contacto*” lo que indica que no hay ninguna invitación en curso. Esto se muestra en la figura 4.22.



Figura 4.22 - El contacto 'Exequiel' ha rechazado la invitación

Si en cambio el contacto acepta la invitación, se ha completado la suscripción mutua por lo que lo ubicará automáticamente en la pestaña “Aceptados”. En este momento la leyenda del contacto cambiará a “Contacto aceptado” y su avatar se actualizará a la imagen que haya establecido en su perfil (figura 4.23).

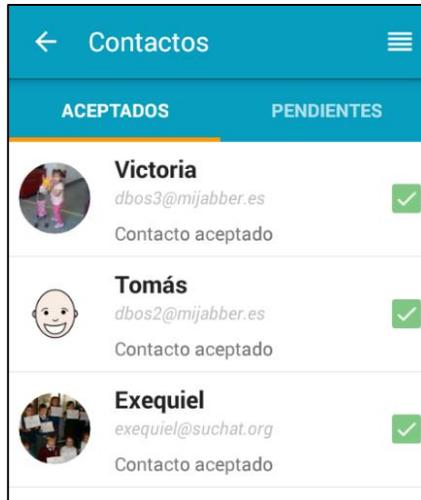


Figura 4.23 - El contacto 'Exequiel' ha aceptado la invitación

### Responder a una invitación

En esta sección se explicará el otro lado del procedimiento de suscripción, que tiene lugar cuando se recibe una invitación por parte de otro usuario y se decide aceptarla o rechazarla. Cuando arriba una invitación, la aplicación informa la situación a través de una notificación del sistema como se muestra en la figura 4.24. El sonido, vibración, y luz de led de la notificación pueden modificarse mediante configuración (ver sección '4.5.4. Ajustes').



Figura 4.24 - Notificación de una solicitud de suscripción (invitación)

El detalle de la notificación mostrará la leyenda “Nueva invitación” junto con el avatar del usuario que envía la invitación y su nombre, ambos obtenidos de su información de perfil. Al tocar sobre la notificación se abrirá la pantalla de administración de contactos para mostrar los contactos “Pendientes”. Allí se encontrarán todas las invitaciones que se han recibido etiquetadas con el texto “INVITACION” como muestra la figura 4.25.

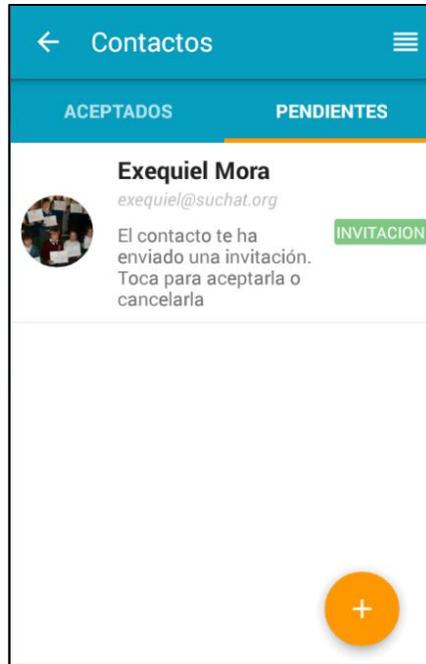


Figura 4.25 - Invitación pendiente

Al presionar sobre el ítem del contacto se accede al menú que se muestra en la figura 4.26 el cual permite aceptar o rechazar la invitación.

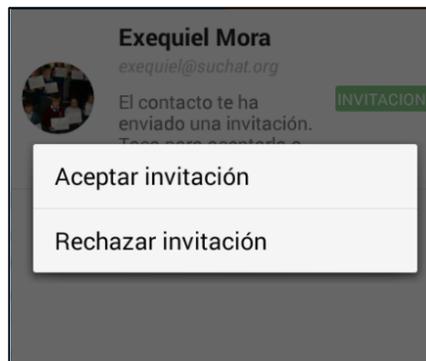


Figura 4.26 - Menú para confirmar/rechazar invitación

La opción “Aceptar invitación” permite aprobar la suscripción mutua con el contacto que ha enviado la invitación. Al presionar esta opción se presenta una pantalla que permite modificar el nombre corto o alias del contacto que se mostrará en la lista de contactos; por defecto se utiliza el nombre obtenido de sus datos de perfil (figura 4.27). Al aceptar la invitación el contacto pasará a formar parte de la lista de contactos.

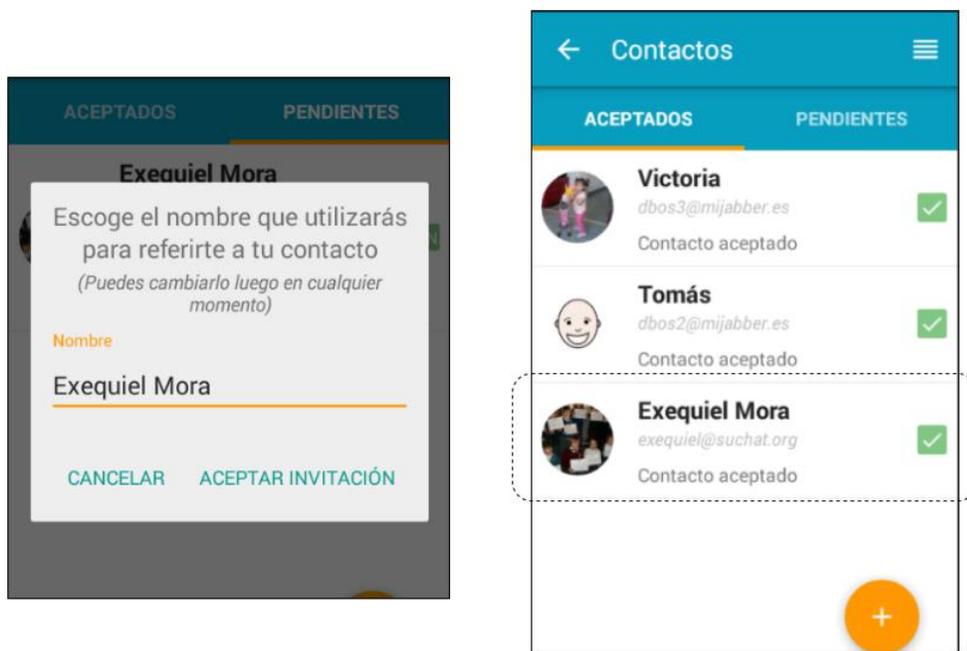


Figura 4.27 - Aceptación de una invitación

La opción “Rechazar invitación” además de informar al contacto acerca del rechazo, elimina el contacto de la lista de contactos pendientes. Previamente se mostrará el mensaje de confirmación que se muestra en la figura 4.28.

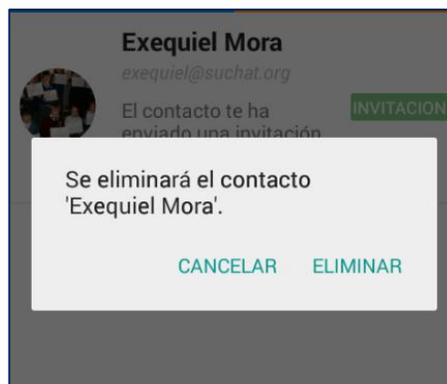


Figura 4.28 - Rechazo de una invitación

### Eliminar un contacto

En todo momento es posible eliminar un contacto de la aplicación, ya sea que se trate de un contacto “Aceptado” o “Pendiente”. Para eliminar un contacto se debe acceder al menú contextual que aparece cuando se presiona un ítem de un contacto. Este menú se muestra la figura 4.29.

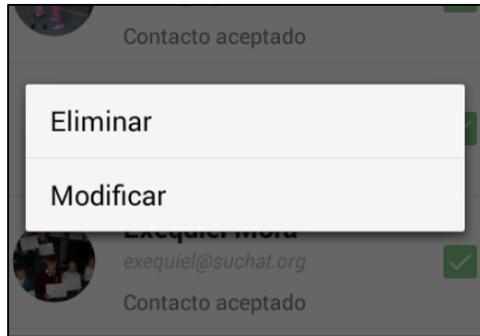


Figura 4.29 - Menú que se despliega al tocar sobre un contacto

La opción “Eliminar” borrará de la aplicación toda la información relacionada con el contacto e inmediatamente le informará que se ha rechazado la suscripción mutua.

### Modificar un contacto

Para modificar un contacto, ya sea “Aceptado” o “Pendiente”, se debe acceder al menú contextual que aparece cuando se presiona un ítem de un contacto (figura 4.29). La opción “Modificar” presenta la pantalla que se muestra en la figura 4.30. Los datos que se permiten modificar son el alias y la visibilidad del contacto (se explica en la siguiente sección). El botón “Guardar” confirmará los cambios.

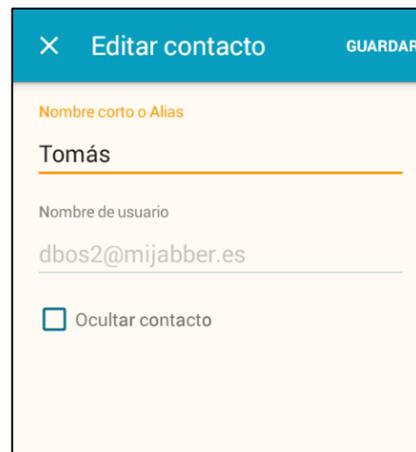


Figura 4.30 - Modificación de un contacto

### Ocultar un contacto

En la pantalla de edición de un contacto es posible modificar la visibilidad de un contacto mediante la casilla de verificación “Ocultar contacto”. La visibilidad se utiliza para

mostrar/ocultar un determinado contacto al usuario final, tanto en las pantalla inicial de contactos de la aplicación como en la pantalla de chat. Cuando un contacto se encuentra oculto tampoco se pueden enviar ni recibir mensajes del mismo. De esta forma es posible mantener la suscripción mutua con un contacto pero deshabilitar temporalmente la comunicación.

En la figura 4.31 se muestra la lista de contactos de la aplicación, en la cual uno de ellos se encuentra oculto. A la derecha se observan las pantallas de contactos y de chat, las cuales no muestran el contacto oculto.

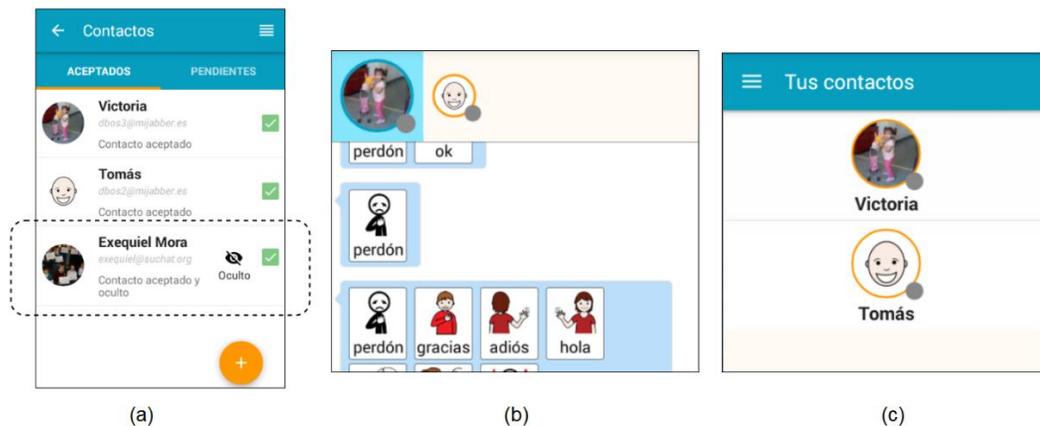


Figura 4.31 - (a) Contacto oculto en la lista de contactos. (b) y (c) Pantallas de chat y de contactos que no muestran el contacto oculto.

### Añadir un comunicador

Un *Comunicador* es un contacto especial al que se le puede enviar mensajes pero del cual nunca se recibirán respuestas. Uno de los objetivos del comunicador es brindar al usuario la posibilidad de que utilice la aplicación y su dispositivo de forma similar a como lo haría con comunicador digital, generando y reproduciendo mensajes para utilizarlo durante una comunicación cara a cara.

Cuando un mensaje es enviado al *Comunicador*, no llega a ningún destinatario ya que no se trata de una persona; sin embargo el mensaje se agrega a la zona de mensajes y su sonido es reproducido, por lo que cualquier interlocutor que se encuentre físicamente junto al usuario puede ver y oír dicho mensaje.

Para agregar un comunicador se debe presionar el botón [  ], que se encuentra en la pantalla de administración de contactos, y seleccionar la opción “Agregar comunicador”. Automáticamente se agregará el contacto *Comunicador* a la lista de contactos como muestra la figura 4.32. Si ya se ha agregado el comunicador, la opción “Agregar comunicador” no estará disponible ya que sólo es posible agregar uno.



Figura 4.32 - Agregado de un comunicador

Una vez creado, el comunicador puede modificarse, ocultarse o eliminarse como si fuese un contacto más.

#### 4.5.3. Categorías de pictogramas

Las categorías de pictogramas se utilizan para facilitar al usuario final la memorización de los pictogramas y para agilizar su búsqueda; generalmente agrupan pictogramas que pertenecen a un mismo tema (ej.: sustantivos, verbos, personas, etc.). Antes de que un usuario final pueda componer un mensaje, se deben crear y configurar las categorías de pictogramas que utilizará. Estas categorías serán las que se mostrarán en la pantalla de selección de pictogramas al momento de componer un mensaje (ver figuras 4.8 y 4.9). A cada categoría es posible establecerle un nombre, un color y asociarle los pictogramas que se consideren necesarios.

Para ingresar a la administración de categorías se debe acceder ítem de administración “Categoría de pictogramas” como se muestra en la figura 4.33. Si aún no se ha creado categorías, aparecerá una lista vacía. A medida que se vayan creando categorías,

se mostrarán en dicha lista. En cada ítem de la lista se muestra el nombre de la categoría, el color asociado y un botón para su edición.

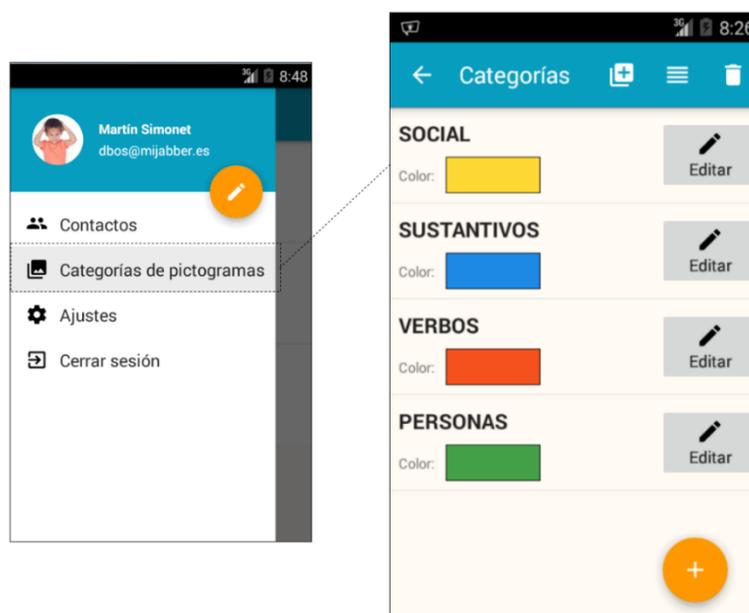


Figura 4.33 - Administración de categorías de pictogramas

### Nueva categoría

Para crear una categoría nueva se debe tocar el botón [  ] que se encuentra en la lista de categorías. En la figura 4.34 se observa la pantalla que se utiliza para crear una nueva categoría. Allí se debe proporcionar un nombre y un color, también es posible indicar si se desea ocultar esta categoría. Si una categoría está oculta no se mostrará en la pantalla de selección de pictogramas al momento de crear mensajes. Para guardar la categoría se debe tocar “Guardar”.

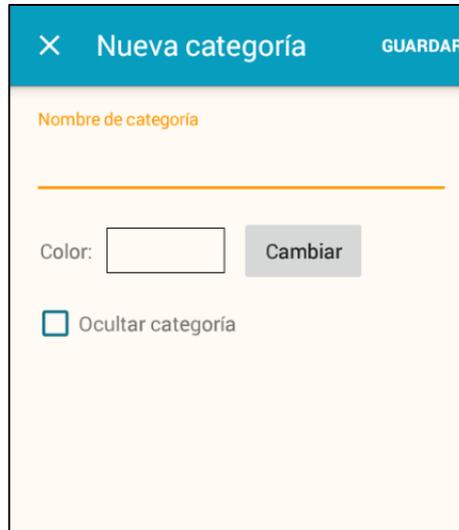


Figura 4.34 - Pantalla para crear una nueva categoría

### Editar categoría

Para editar una categoría se debe tocar el botón “Editar” que se encuentra en cada ítem de la lista de categorías. Esto abrirá la pantalla que se muestra en la figura 4.34 pero con los datos de la categoría seleccionada ya precargados. Para guardar los cambios se debe tocar el botón “Guardar”.

### Ver pictogramas

Inicialmente cuando se crea una categoría nueva, ésta no posee pictogramas asociados, por lo que se los debe ir añadiendo manualmente. Para añadir pictogramas a la categoría se debe acceder a su lista de pictogramas tocando sobre el ítem de la categoría. Allí se mostrará la pantalla que se observa en la figura 4.35 en donde se listan los pictogramas añadidos a dicha categoría (si aún no se han añadido pictogramas ésta lista aparecerá vacía). En la cabecera de la pantalla se muestra el nombre de la categoría y la cantidad de pictogramas configurados.



Figura 4.35 - Lista de pictogramas de una categoría

### Añadir pictogramas

Para añadir un nuevo pictograma a la categoría se debe tocar el botón [  ]. Esto abre la pantalla de selección de pictogramas que se muestra en la figura 4.36.



Figura 4.36 - Selección de pictogramas predefinidos para añadir a una categoría

Esta pantalla permite seleccionar uno o más pictogramas para añadirlos a la categoría que se está editando. Los pictogramas que se muestran en esta pantalla son los *pictogramas predefinidos* que vienen instalados en la aplicación. Para facilitar su búsqueda cada pictograma predefinido se encuentra asociado a una *categoría predefinida* (Animales, Aseo, etc.).

NOTA: Los *pictogramas predefinidos* son los únicos pictogramas que pueden ser escogidos para ser añadidos a una categoría. La aplicación no presenta la posibilidad de generar o incorporar nuevos pictogramas.

Para seleccionar un pictograma se debe activar su casilla de verificación realizando un toque sobre su recuadro. Para quitarlo de la selección se lo debe tocar nuevamente. En la figura 4.36 se observa que se han seleccionado los pictogramas “Animal”, “Pez” y “Cepillo”. Es posible realizar una búsqueda de un pictograma mediante el botón [  ]. Finalmente para añadir los pictogramas seleccionados a la categoría se debe tocar el botón “Aceptar”.

Un mismo pictograma puede añadirse más de una vez a una categoría. La razón de esto es que un mismo pictograma puede representar diferentes significados para el usuario. Por ejemplo, es posible agregar en una categoría el pictograma ‘Hola’ dos veces. Luego se podría editar el texto de cada pictograma (ver siguiente sección) para reflejar dos significados similares pero diferentes, colocando “Buen día” en uno y “¿Hola, cómo estás?” en el otro.

Por último vale aclarar que un pictograma puede añadirse a más de una categoría. Cuando un pictograma se añade a una categoría pasa a tener una identidad propia independiente de cualquier otro pictograma, por lo que puede modificarse, eliminarse, ocultarse, etc., sin alterar ningún otro pictograma.

### Editar pictograma

Cada pictograma dentro de una categoría tiene asociado dos textos, uno que se muestra debajo de su imagen y otro texto que se sintetiza cuando se reproduce el sonido del pictograma (por ejemplo, cuando se construye un mensaje). Ambos textos se pueden modificar mediante la edición del pictograma. Para abrir la pantalla de edición de un pictograma (figura 4.37) se debe tocar sobre él en la lista de pictogramas de la categoría.

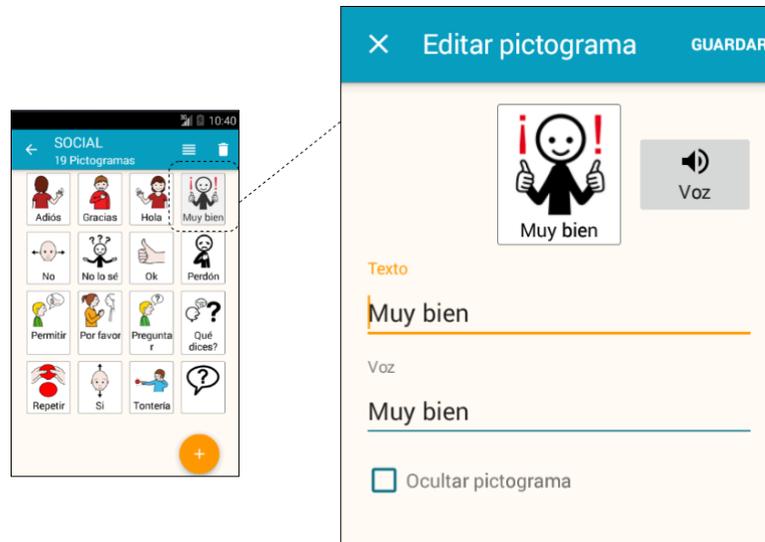


Figura 4.37 - Edición de un pictograma dentro de una categoría

En la pantalla de edición de un pictograma es posible modificar su texto y su voz (texto a reproducir o sintetizar), también es posible ocultar el pictograma para que no se muestre al usuario cuando componga un mensaje. Una pre-visualización del pictograma se muestra en la parte superior de la pantalla con la posibilidad de reproducir el campo “Voz”. Para guardar los cambios se debe tocar el botón “Guardar”.

### Ordenar pictogramas

El orden en que se muestran los pictogramas se puede modificar mediante el botón [  ] disponible en la lista de pictogramas de una categoría. Para cambiar el orden de un pictograma se lo debe desplazar a la posición deseada (figura 4.38). Para confirmar el ordenamiento se debe tocar el botón “Aceptar”.

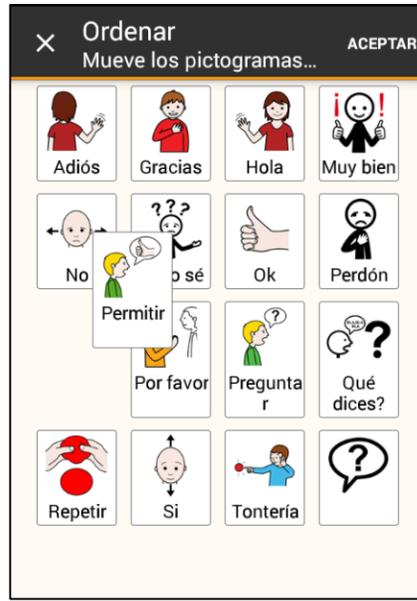


Figura 4.38 - Ordenamiento de pictogramas de una categoría

### Eliminar pictogramas

La eliminación de un pictograma quita el pictograma sólo de la categoría que se está editando. Para hacer esto se debe presionar el botón [  ] disponible en la lista de pictogramas de la categoría. Al tocar este botón aparecen casillas de verificación en cada uno de los pictogramas para que puedan ser seleccionados (figura 4.39). Los pictogramas seleccionados se eliminarán cuando se presione el botón “Eliminar”.



Figura 4.39 - Eliminación de pictogramas en una categoría

### Añadir categorías predefinidas

La aplicación cuenta con un conjunto de pictogramas denominados *pictogramas predefinidos*. Estos pictogramas se encuentran organizados en *categorías predefinidas* para facilitar su localización. Las categorías predefinidas también se pueden utilizar como base para crear nuevas categorías propias. El usuario luego podrá realizar las modificaciones que considere necesarias: agregar o quitar pictogramas, cambiar su nombre y color, etc.

Para tomar como base una categoría predefinida se debe presionar el botón [  ] que se encuentra en la lista de categorías. Este botón abre la pantalla que muestra todas las categorías predefinidas (figura 4.40).



Figura 4.40 - Selección de categorías predefinidas

Se permite la selección de una o más categorías predefinidas, para ello se debe tocar sobre el ítem correspondiente lo que activará la casilla de verificación. Para ver la lista con los pictogramas que contiene la categoría se debe tocar el botón “Ver”. Finalmente al presionar el botón “Agregar” se añadirán las categorías seleccionadas a la lista de categorías propias. Se creará una categoría propia por cada categoría predefinida que se haya seleccionado.

#### 4.5.4. Ajustes

Los ajustes permiten modificar el comportamiento o aspecto de las diferentes funcionalidades de la aplicación para que se adapte a las preferencias y necesidades del usuario. Para ingresar a los ajustes de la aplicación se debe acceder al menú de administración “Ajustes” como se muestra en la figura 4.41.



Figura 4.41 - Pantalla de ajustes de la aplicación

Los ajustes de la aplicación se encuentran agrupados en tres opciones de menú: “General”, “Notificaciones” y “Texto a Voz”. Cada una de estas opciones contiene un conjunto de ajustes a los cuales se accede al tocar sobre el ítem. A continuación se explica en detalle cada uno de ellos.

### Opción “General”

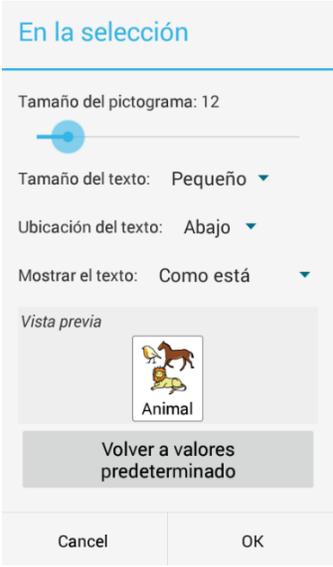
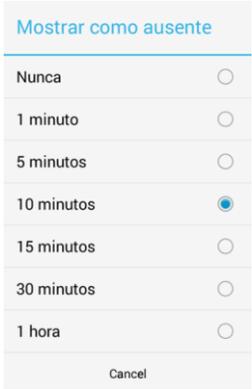
Los ajustes generales permiten modificar parámetros tales como el tipo de selección de pictogramas y su aspecto. En la figura 4.42 se muestra la pantalla que contiene los ajustes de esta opción. En la tabla 4.3 se explica el comportamiento de cada uno de ellos.



Figura 4.42 - Pantalla de ajustes de la opción 'General'

Tabla 4.3 - Descripción de ajustes de la opción 'General'

Nombre del ajuste	Descripción	Imagen
Selección de pictogramas/ Disposición de pictogramas	Indica la manera en la que los pictogramas se organizan al momento de realizar la selección en la creación de un mensaje. Los valores posibles son "Por categorías" y "Por columnas"	
Selección de pictogramas/ Selección pantalla completa	Si la casilla de verificación está activa, la selección de pictogramas al momento de crear mensajes, ocupará la pantalla completa. En caso contrario ocupará sólo una parte de la pantalla. (Ambas opciones son analizadas en la sección '4.4.2. Creación de mensajes')	

<p>Aspecto de pictogramas/En la selección</p>	<p>Permite modificar el aspecto que tendrán los pictogramas en la pantalla de selección de pictogramas. Es posible cambiar: el tamaño de la imagen (<i>escala 1 a 100</i>), el tamaño del texto (<i>Sin texto, Pequeño, Mediano, Grande</i>), la ubicación del texto (<i>Arriba, Abajo</i>) y mostrarlo en mayúscula o minúscula. La vista previa permite observar los cambios antes de guardar. Se puede volver a los valores por defecto mediante el botón “Volver a valores predeterminados”.</p>	
<p>Aspecto de pictogramas/En los mensajes</p>	<p>Permite modificar el aspecto que tendrán los pictogramas en los mensajes.</p>	<p>&lt;Ídem anterior&gt;</p>
<p>Estado/Mostrar como ausente</p>	<p>Configura el tiempo que debe transcurrir sin actividad para que la aplicación automáticamente establezca el estado de conexión en “Ausente”.</p>	

### Opción “Notificaciones”

Los ajustes de notificaciones permiten activar o desactivar las notificaciones tanto de mensajes nuevos como de invitaciones. Permiten además modificar el sonido, vibración y luz de led de cada tipo de notificación. En la figura 4.43 se muestra la pantalla que contiene los ajustes de esta opción. En la tabla 4.4 se explica el comportamiento de cada uno de ellos.

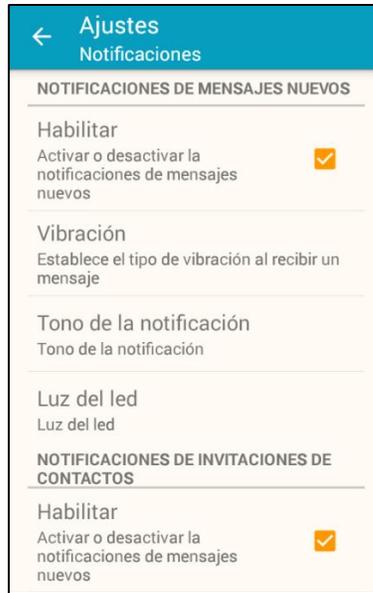
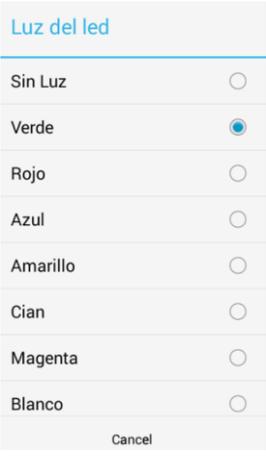
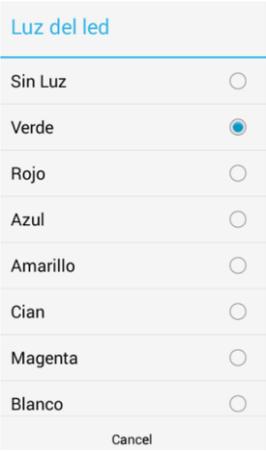


Figura 4.43 - Pantalla de ajustes de la opción 'Notificaciones'

Tabla 4.4 - Descripción de ajustes de la opción 'Notificaciones'

Nombre del ajuste	Descripción	Imagen
Notificaciones de mensajes nuevos/ Habilitar	Activa o desactiva las notificaciones que alertan sobre la llegada de un nuevo mensaje. Si la casilla está desactivada no se recibirán notificaciones de mensajes nuevos.	
Notificaciones de mensajes nuevos/ Vibración	Establece la vibración que se produce cuando se muestra una notificación de mensaje nuevo.	

<p>Notificaciones de mensajes nuevos/ Tono de la notificación</p>	<p>Establece el ringtone que se reproduce cuando se muestra una notificación de mensaje nuevo.</p>	
<p>Notificaciones de mensajes nuevos/ Luz de led</p>	<p>Establece el color del led que destellará cuando se muestre una notificación de un mensaje nuevo</p>	
<p>Notificaciones de invitaciones / Habilitar</p>	<p>Activa o desactiva las notificaciones que alertan sobre un nuevo mensaje recibido. Si la casilla está desactivada no se recibirán notificaciones de invitaciones.</p>	
<p>Notificaciones de invitaciones / Vibración</p>	<p>Establece la vibración que se produce cuando se muestra una notificación de una invitación.</p>	

<p>Notificaciones de invitaciones / Tono de la notificación</p>	<p>Establece el ringtone que se reproduce cuando se muestra una notificación de una invitación.</p>	
<p>Notificaciones de invitaciones / Luz de led</p>	<p>Establece el color del led que destellará cuando se muestre una notificación de una invitación.</p>	

### Opción “Texto a voz”

La opción “Texto a voz” permite activar o desactivar la síntesis de voz y establecer en que situaciones se debe convertir el texto a voz. En la figura 4.44 se muestra la pantalla que contiene los ajustes de esta opción. En la tabla 4.5 se explica el comportamiento de cada uno de ellos.

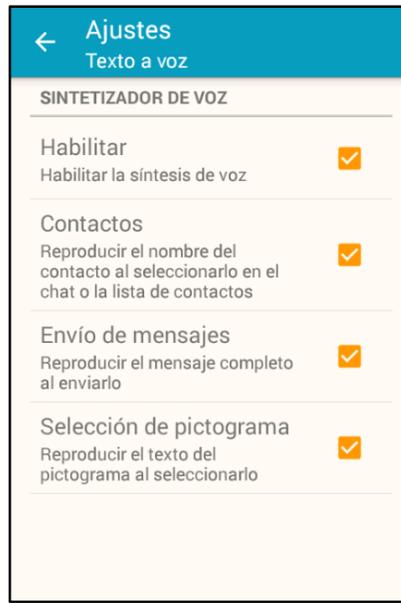


Figura 4.44 - Pantalla de ajustes de la opción 'Texto a voz'

Tabla 4.5 - Descripción de los ajustes de la opción 'Texto a voz'

Nombre del ajuste	Descripción	Imagen
Sintetizador de voz / Habilitar	Activa o desactiva la síntesis de voz en la aplicación.	
Sintetizador de voz / Contactos	Si la casilla de verificación se encuentra activa, se reproducirá el nombre del contacto al seleccionarlo en la pantalla de chat o en la pantalla de contactos.	
Sintetizador de voz / Envío de mensajes	Si la casilla de verificación se encuentra activa, se reproducirá el mensaje completo al momento de enviarlo.	
Sintetizador de voz / Selección de pictogramas	Si la casilla de verificación se encuentra activa, se reproducirá el texto del pictograma cuando se seleccione en la pantalla de selección de pictogramas.	

#### 4.5.5. Cierre de sesión

El cierre de sesión implica la desconexión del servidor al cual se está conectado y el posterior cierre de la aplicación. El usuario pasará inmediatamente al estado “Desconectado” y el servicio que permite recibir mensajes e invitaciones se detendrá.

Cuando el usuario intente iniciar la aplicación nuevamente, se presentará la pantalla de inicio de sesión para identificarse y comenzar una nueva sesión. Para proceder al cierre de sesión se debe presionar el botón “Cerrar sesión” disponible en el menú de administración (figura 4.45).

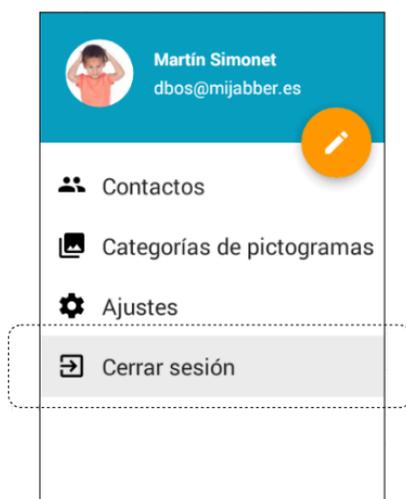


Figura 4.45 - Cierre de sesión de la aplicación

Si un contacto envía mensajes al usuario mientras éste se encuentra “Desconectado”, los mensajes se almacenarán en el servidor XMPP y serán entregados la próxima vez que el usuario inicie sesión.

## Capítulo 5

### Conclusiones y trabajos futuros

En el presente trabajo se ha desarrollado un servicio de mensajería que permite a dos o más personas mantener una comunicación en tiempo real y que se basa en la utilización de pictogramas del sistema *SPC* para la construcción de los mensajes. Este servicio se compone de una aplicación móvil que se encarga de interactuar con el usuario, y de un servidor que hace posible la comunicación entre los usuarios.

El servicio fue construido con el objetivo de constituirse en una herramienta de apoyo para la comunicación, para aquellas personas que por diferentes motivos no han podido desarrollar el habla adecuadamente o que poseen un vocabulario limitado. Al utilizar el *SPC* para componer los mensajes, el usuario dispone de un sistema sencillo que puede representar adecuadamente la realidad de su entorno. A su vez el componente táctil de los dispositivos móviles y su predominio de salidas auditivas y visuales, provocan en el usuario una estimulación muy rica que alienta la comunicación.

La característica fundamental del servicio, como en todo sistema de mensajería, es la posibilidad de establecer una comunicación en tiempo real con un interlocutor que se encuentra físicamente en otra ubicación. Para muchos usuarios esto puede representar un medio para favorecer su autonomía personal y su interacción social ya que permite incorporar la comunicación en muchas más situaciones de la vida diaria. Adicionalmente, la aplicación desarrollada también proporciona la función de *Comunicador* para los casos en los que el usuario desea mantener una comunicación cara a cara con su interlocutor.

Durante el transcurso de este proyecto han surgido muchas líneas de desarrollo e investigación que debido al alcance del mismo no han sido tratadas, pero sería muy interesante poder trabajar sobre ellas en un futuro. A continuación se listan los trabajos futuros más relevantes:

- Desarrollo de un mecanismo de predicción de pictogramas, similar a los que ya existen para mensajes de texto, que tenga en cuenta la historia de mensajes del usuario así como también el contexto de la comunicación

actual (Por ej.: pictogramas más utilizados, pictogramas previamente seleccionados en el mensaje, persona a la cual se dirige, etc.)

- Implementación de la aplicación en otras plataformas: escritorio (Windows, Linux), web y móvil (iPad).
- Incorporar el soporte para utilizar otros tipos de sistemas de símbolos basados en pictogramas (BLISS, MINSPEAK, etc.).
- Incorporar la posibilidad de que el usuario pueda añadir pictogramas personalizados seleccionando la imagen, texto y sonido que desee.
- Centralización en el servidor del almacenamiento de toda la información configurada en la aplicación, de esta manera el usuario podrá tener la misma configuración en cualquier dispositivo que utilice.

## Bibliografía

- Basil, C., & Puig, R. (1988). *Comunicación aumentativa*. Madrid: INSERSO.
- Beukelman, D., & Mirenda, P. (1998). *Augmentative and alternative communication: management of severe communication disorders in children and adults*. Paul H. Brookes Publishing Co.
- Delgado Santos, C. I. (2012). *Mi software de comunicación*. Madrid: CEAPAT-INSERSO.
- Ericsson. (Junio 2013). *Ericsson Mobility Report*. Obtenido de <http://www.ericsson.com/res/docs/2013/ericsson-mobility-report-june-2013.pdf>
- Fernandez, P. (2013). *Apoyo a la comunicación*. España: Mc Graw Hill Interamericana.
- Gamma, E., Johnson, R., Helm, R., Vlissides, J., Booch, G., & Bosquiazzo, D. (1994). *Design Pattern: element of reusable object-oriented software*. Addison-Wesley.
- Gómez, E. F., Sancho, A. R., & Cabeza, A. S. (2009). Terapia ocupacional en daño cerebral adquirido. *TOG (Revista en internet)*, 6, 410-464. Obtenido de [www.revistatog.com/suple/num4/cerebral.pdf](http://www.revistatog.com/suple/num4/cerebral.pdf)
- IETF. (2004). RFC 3920. *Extensible Messaging and Presence Protocol (XMPP): Core*.
- Portal ARASAAC. (2015). Obtenido de <http://arasaac.org/>
- Sitio web oficial Fundación Maresme. (2015). Obtenido de <http://www.fundmaresme.com/>
- Sitio web oficial Messenger Visual. (2015). Obtenido de <http://www.messengervisual.com/>
- Sitio web oficial Ola Mundo. (2015). Obtenido de <http://www.olamundo.com/>
- Tetzchner, S., & Martinsen, H. (1993). *Introducción a la enseñanza de signos y al uso de ayudas técnicas para la comunicación*. Antonio Machado.