



UNIVERSIDAD NACIONAL DE SANTIAGO DEL ESTERO
FACULTAD DE CIENCIAS EXACTAS Y TECNOLOGÍAS



LICENCIATURA EN SISTEMAS DE INFORMACIÓN

TRABAJO FINAL DE GRADUACIÓN

**PROTOTIPO DE UN SISTEMA
ALTERNATIVO DE COMUNICACIÓN
PARA NIÑOS CON PARÁLISIS
CEREBRAL INFANTIL**

Autor:

MELISA NATALY DEL VALLE CÓRDOBA

Profesor guía:

MG. LIC. PAOLA DANIELA BUDÁN

Abril de 2017

TRABAJO FINAL DE GRADUACIÓN DE LA LICENCIATURA EN SISTEMAS DE
INFORMACIÓN

**PROTOTIPO DE UN SISTEMA ALTERNATIVO DE
COMUNICACIÓN PARA NIÑOS CON PARÁLISIS
CEREBRAL INFANTIL**

Autor:

.....

Nombre Apellido

Profesor Guía:

.....

Nombre Apellido

Asesor:

.....

Nombre Apellido

Aprobado el día del mes de del año 20.....
por el Tribunal integrado por

.....

(firma)

.....

(firma)

.....

(firma)

.....

(aclaración)

.....

(aclaración)

.....

(aclaración)

A

Mi familia quienes con su amor y comprensión me acompañaron en cada paso, reforzando permanentemente la confianza y fortaleza para no desistir en mis proyectos.

Agradecimientos

A mi compañero de vida, Antonio y a nuestros hijos, Benjamín y Daniel, gracias por su amor, por sus sacrificios y por ser el motor para que pueda lograr una anhelada meta en mi vida.

A mis padres Teresa y Hugo, quienes con el ejemplo y afecto me inculcaron valores de la vida, dándome las herramientas necesarias para ir surcando mi camino, y no abandonar mis sueños.

A Pedro y Mercedes quienes afectuosamente acompañaron y confiaron en mí para culminar mis proyectos.

A mis hermanos y sobrinos por todo el cariño brindado.

A mi directora Paola D. Budan que abiertamente me brindó sus conocimientos profesionales para nutrirme; quien además de motivar de manera permanente cada etapa del proyecto ofreció sus consejos y tiempo para que logre alcanzar una meta más en mi vida.

A mi asesor Pablo J. Najar, quien dedicó sus conocimientos y su tiempo para orientar el desarrollo.

A todos los integrantes del proyecto de investigación que con predisposición brindaron sus aprendizajes para enriquecer y complementar mis conocimientos.

M.N.V.C.

Santiago del Estero, Argentina

Abril de 2017

ÍNDICE DE CONTENIDOS

RESUMEN	vi
INTRODUCCIÓN.....	vii
Capítulo 1 - Presentación del Problema y Objetivos.....	1
1.1. Planteo Del Problema	2
1.2. Objetivos Del Trabajo	3
Capítulo 2 - Marcos Referenciales	4
2.1. Marco Teórico	5
2.1.1. Parálisis Cerebral Infantil.....	5
2.1.2. Sistemas Alternativos de Comunicación.....	7
2.1.3. Aplicaciones Móviles.....	9
2.1.4. SO Android	11
2.1.5. Sensibilidad al Contexto.....	14
2.2. Marco Procedimental	15
2.2.1. eXtremining Programming.....	15
2.2.2. SCRUM.....	17
2.2.3. Feature Drive Development	18
2.2.4. Mobile-D	18
2.3. Conclusión.....	23
Capítulo 3 - Estado del Arte	24
3.1. SAC Electrónicos Manuales.....	25
3.1.1. Proyect@ PECS	25
3.1.2. Palphoons	26
3.1.3. Sistemas de Comunicación No Verbales	28
3.1.4. OTTAA Project.....	29
3.1.5. AraBoard	31

3.2.	SAC – Características Deseables.....	31
3.3.	SAC en el IPRI	32
3.4.	Conclusión.....	33
Capítulo 4 - Fase de Exploración		35
4.1.	Establecimiento De Los Interesados.....	37
4.2.	Definición De Alcances.....	39
4.2.1.	Planificación Inicial del Proyecto.....	40
4.2.2.	Colección de Requerimientos Iniciales	42
4.3.	Establecimiento Del Proyecto	45
4.3.1.	Selección del Entorno – Línea de Arquitectura.....	47
4.3.2.	Establecimiento del Proceso.....	49
4.4.	Verificación Del Plan de Proyecto	52
4.5.	Conclusión.....	53
Capítulo 5 - Fase de Inicialización		54
5.1	Configuración Del Proyecto	56
5.1.1.	Configuración del Entorno	57
5.1.2.	Entrenamiento.....	60
5.1.3.	Establecimiento de la comunicación	61
5.2	Día De Planificación Inicial O Iteración 0	62
5.2.1.	Análisis de Necesidades Iniciales.....	64
5.2.2.	Planificación de Líneas de Arquitectura.....	65
5.2.3.	Planificación de la Iteración 0	67
5.2.4.	Generación de Pruebas de Aceptación	78
5.3	Día De Prueba De La Iteración 0.....	84
5.4	Conclusión.....	87
Capítulo 6 - Fase de Producción.....		88
6.1	Día De Planificación.....	91
6.1.1.	Taller Post-Iteración	93
6.1.2.	Análisis de Requerimientos.....	93

6.1.3. Planificación de la Iteración.....	96
6.1.4. Generación de Pruebas de Aceptación.....	111
6.2 Día De Trabajo	115
6.2.1. Recapitulaciones.....	117
6.2.2. Desarrollo de Componentes Funcionales.....	117
6.2.3. Informar al Cliente	123
6.3 Día De Lanzamiento.....	124
6.3.1. Integración del Sistema y Pruebas Previas a la Liberación.....	125
6.3.2. Pruebas de Aceptación	126
6.3.3. Ceremonia de Lanzamiento.....	135
6.4 Conclusión.....	136
Capítulo 7 - Conclusiones y Trabajo Futuro	137
Bibliografía	141
Anexo.....	145
Encuesta a los terapeutas del IPRI.....	146
Código de desarrollo	150
Pruebas de Verificación de Iteración 0	168
Fichas de HU.....	170

ÍNDICE DE TABLAS

Tabla 4-1: Tareas por fase e hitos de control	41
Tabla 4-2: Resumen de la documentación prevista - proyecto de desarrollo.....	51
Tabla 4-3: Lista de verificación de la fase de exploración.....	52
Tabla 5-1: Recursos necesarios para el desarrollo del SAC.....	58
Tabla 5-2: Requerimientos priorizados por el grupo de interesados.....	65
Tabla 5-3: Formato general de la ficha de historia de usuario	68
Tabla 5-4: Historia de usuario 1 “agrupamiento de palabras”	69
Tabla 5-5: Historia de usuario 2 “navegación entre pantallas”	70
Tabla 5-6: Historia de usuario 3 “complejidad del vocabulario”	71
Tabla 5-7: Historia de usuario 4 “alertas según horarios”.....	72
Tabla 5-8: Ejemplo de jerarquía de palabras para el caso puntual de Almuerzo	74
Tabla 5-9: Validación del “agrupamiento de palabras” según terapeutas.....	79
Tabla 5-10: Validación de la “navegación entre pantallas” según terapeutas.....	80
Tabla 5-11: Validación terapéutica de la “complejidad del vocabulario”.....	81
Tabla 5-12: Funcionalidades de SAC priorizadas.....	86
Tabla 6-1: Requerimientos y productos por iteración	94
Tabla 6-2: Tareas a llevar a cabo durante la Iteración I	98
Tabla 6-3: Tareas a llevar a cabo durante la Iteración II.....	99
Tabla 6-4: Tareas a llevar a cabo durante la Iteración III.....	100
Tabla 6-5: Ejemplo de pictogramas y acciones del comunicador	101
Tabla 6-6: Pruebas de aceptación para cada requerimiento por iteración	112
Tabla 6-7: Validación de las funcionalidades del prototipo de la iteración I....	128
Tabla 6-8: Validación de las funcionalidades del prototipo en la iteración II...	130
Tabla 6-9: Validación de las funcionalidades del prototipo en la iteración III .	130
Tabla 6-10: Listado de defectos encontrados	132
Tabla 6-11: Requerimientos pos iteraciones para trabajos futuros	134

ÍNDICE DE FIGURAS

Figura 2-1: Arquitectura del SO Android	13
Figura 2-2: Fases y etapas del método Mobile-D.....	19
Figura 3-1: Interfaz principal de Project@ PECS	26
Figura 3-2: Versión configurable en computadora de Plaphoons	27
Figura 3-3: Interfaz de usuario final del sistema de comunicación no verbal.....	29
Figura 3-4: Interfaz de reproducción de OTTAA Project	30
Figura 3-5: Ejemplos de SAC manuales	33
Figura 4-1: Planificación temporal del proyecto de desarrollo del SAC	41
Figura 4-2: Tendencias en los últimos años de tecnologías móviles	48
Figura 4-3: Arquitectura cliente monolítica	49
Figura 5-1: Tarea de configuración del entorno	58
Figura 5-2: Representación gráfica de la configuración de usuarios	66
Figura 5-3 : Imágenes utilizadas en el diseño de las interfaces – Iteración 0	75
Figura 5-4: Construcción del prototipo funcional	76
Figura 5-5: Ejemplo de interfaces para iteración 0	76
Figura 5-6: Ejemplo interfaces/ navegación del prototipo, niños de 2 a 4 años..	77
Figura 5-7: Ejemplo de jerarquía de palabras para el SAC	78
Figura 5-8: Pila de productos para el desarrollo del SAC.....	87
Figura 6-1: Implementación de historia de usuario en iteraciones.....	96
Figura 6-2: Ejemplo de pictogramas iguales con nomenclatura diferente	102
Figura 6-3: Ejemplo de clasificación de acciones según su generalidad	103
Figura 6-4: Ejemplos de diferentes interfaces del sistema.....	104
Figura 6-5: Esquema de navegación del comunicador Básico (Iteración I).....	106
Figura 6-6: Navegación entre pantallas para el “Comunicador adaptable a las preferencias del usuario” (iteración II).....	108
Figura 6-7: Navegación en la iteración III del “SAC sensible al tiempo”	110

RESUMEN

Los *sistemas alternativos de comunicación* son herramientas de apoyo a la comunicación de las personas con afecciones en la comunicación oral. Dentro de los usuarios de estos sistemas se encuentran personas con parálisis cerebral infantil, quienes presentan diversas alteraciones sensoriales, en la comunicación, en la motricidad, entre otros. El uso de *sistemas alternativos de comunicación manuales* para comunicarse por parte de los niños con parálisis cerebral infantil requiere asistencia externa por parte de un tutor y el proceso - hasta poder generar un mensaje - es lento.

Este trabajo propone el desarrollo de un prototipo de aplicación móvil que permita apoyar las necesidades de comunicación de los niños de 2 a 10 años de edad con PCI leve a moderada, pero que puede ser utilizado por cualquier persona con limitaciones motrices moderadas y dificultades de comunicación oral, ya sea temporal o permanente. Dicho prototipo funcionará bajo el sistema operativo Android, para lograr un comunicador portátil de habla artificial, que hará uso de recursos gráficos como ser símbolos pictográficos, ortográficos, entre otros. Estos recursos deben poder seleccionarse en forma táctil, emitiendo una salida de voz acorde a las expresiones y modismos santiagueños para facilitar la comunicación.

Esta propuesta se ve motivada por lo siguiente: La posibilidad de proveer una *herramienta de apoyo a las necesidades comunicativas orales* de niños con PCI que asisten al Instituto Provincial de Rehabilitación Integral (IPRI); proporcionar un *recurso tecnológico móvil de bajo costo* para mejorar las habilidades comunicativas orales de estos niños con sus pares y adultos; brindar una herramienta adaptada a expresiones y sonidos en el lenguaje nativo.

Desde el punto de vista computacional la originalidad de este trabajo consiste en construir un sistema móvil sensible al contexto temporal, aplicando la metodología Mobile-D para su desarrollo.

Palabras clave: Sistema Alternativo de Comunicación; aplicación móvil; prototipo; disminución de las capacidades de comunicación: parálisis cerebral infantil.

INTRODUCCIÓN

La comunicación tiene como finalidad influir en un medio ambiente determinado, sin embargo, esta influencia puede ser invariablemente reducida si algún elemento del proceso que la lleva a cabo se ve afectado (Rodríguez Durán, Ugalde Briones, 2012). La comunicación puede llevarse a cabo de forma oral, escrita, mediante el uso de símbolos, entre otros. Los *sistemas alternativos de comunicación* (SAC) surgieron para dar soporte a la comunicación cuando el emisor no puede utilizar la voz. Los SAC pueden definirse como sistemas cuya característica principal es permitir el reemplazo/ ampliación del habla cuando ésta no existe (Baumgart, Johnson, & Helmstetter, 1996), es decir, son herramientas que permiten a las personas limitadas a nivel de lenguaje oral, comunicarse de manera más fluida con su entorno. Estos sistemas utilizan diferentes recursos simbólicos y tecnológicos que, combinados adecuadamente, permiten una comunicación efectiva de diferentes necesidades. El lenguaje de señas, los tableros y libros de comunicación, así como también los comunicadores electrónicos, son ejemplos concretos de SAC. En el caso de tres últimos se utilizan recursos simbólicos para apoyar el proceso. Los comunicadores electrónicos incorporan además la reproducción de sonido (Fernández, Sánchez, Roldán, Rodríguez-Fórtiz, Hurtado-Torres, & Medina-Medina, 2009) de la necesidad que se busca comunicar.

Esta investigación se ve motivada por la situación que se presenta diariamente en el ámbito del Instituto Provincial de Rehabilitación Integral (IPRI) de la Provincia de Santiago del Estero, donde los terapeutas y familiares de pacientes con Parálisis Cerebral Infantil (PCI) que utilizan SAC manuales, expresan la necesidad de mejorar las posibilidades de comunicación de estos niños con su entorno. La PCI conlleva trastornos motores, sensoriales y cognitivos, así como también aparece alteraciones de la comunicación, de la conducta, de la percepción y/o del intelecto, los cuales pueden o no acompañarse con epilepsia (Argüelles, 2008), por lo cual según la severidad clínica clasificada como leve, moderada o severa (Sfaello Z., 2015), se requiere asistencia eventual o permanente siendo esencial el uso de un SAC para comunicarse. Este trabajo de interés académico alcanza a niños con PCI leve a moderada pues el proceso de comunicación que implementan requiere asistencia externa de un tutor con lo cual se ralentiza. En la actualidad, para apoyar el proceso de comunicación de los niños con PCI, se utilizan cuadernos o libros de

comunicación, poco portables, que no permiten satisfacer completamente las necesidades comunicativas sin asistencia permanente por parte de terapeutas o tutores, generando una dependencia del asistente. El proceso de comunicación con estos SAC es complejo (Ortí, 2002), por ejemplo en situaciones donde el niño va a interactuar con diferentes personas y no todas conocen el sistema, lo cual provoca que el niño tienda a aislarse (Abadín, Delgado Santos & Vigara Cerrato, 2010). En estas situaciones donde se presentan barreras semánticas en la comunicación (Rodríguez Durán, Ugalde Briones, 2012) el asistente debe hacer de traductor de las necesidades permanentemente, dado que para interpretar la comunicación se necesita conocer los significados de los recursos del SAC utilizado.

En el mercado actual se encuentran disponibles diferentes tipos de SAC electrónicos que se implementan con la misma funcionalidad de los cuadernos de comunicación, que son más portables y requieren menos intervención del asistente del niño para comunicar. Además, permiten tener acceso a mayor cantidad de símbolos gráficos en menor tiempo -comparados con los cuadernos-, y disponen de una salida en forma de habla digitalizada o sintetizada para comunicar las necesidades. Este tipo de comunicadores promueve la independencia en la comunicación, necesitando menos intervención del asistente para relacionarse con el entorno social¹ (Baldassarri, Peña, Cerezo, & Marco, 2014). Pero en el caso de los niños con PCI, son difíciles de implementar, pues en su mayoría están desarrollados sin contemplar afecciones motrices o, los que la contemplan, requieren el uso de adaptadores externos como pulsadores, teclados adaptados, entre otros (Ortí, 2002) además, de haber sido desarrollados en culturas extranjeras (González, Sosa, & Martín, 2014) y tener un costo asociado. El detalle de considerar de dónde provienen no es menor, pues el desarrollo del contenido y la terminología que utiliza cada comunicador considera las necesidades propias de la comunidad de origen, es decir la diversidad de pensamientos, necesidades y costumbres propios de la cultura.

Por lo expuesto, en el presente trabajo se propone el desarrollo de SAC basado en tecnología móvil accesible en nuestro medio, para apoyar al proceso de comunicación oral de los niños con PCI leve a moderada. Esto requerirá integrar las funcionalidades de los SAC actuales del IPRI, incorporando un diseño de interfaz

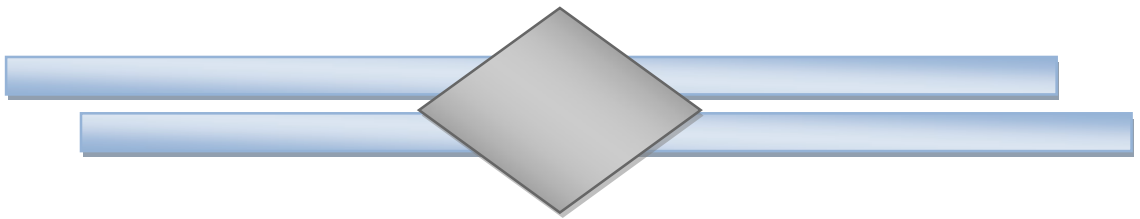
¹ <http://bj-adaptaciones.com/blog/ventajas-del-uso-de-comunicadores/> (Consultado en Marzo 2017)

pertinente y reproducción en el lenguaje nativo, a un sistema aprovechable en una plataforma móvil. Dado que los usuarios poseen una motricidad reducida, la aplicación facilitará escenarios de comunicación, incorporando la sensibilidad temporal del contexto (Valenzuela, & Terriza, 2008), adaptándose al usuario, la tarea y su contexto (Galico, Natanzon, Vega, Matalonga, Solari, 2015).

Para llevar a cabo el desarrollo de esta aplicación se decide utilizar Mobile-D, por ser un método para el desarrollo de software destinado a dispositivos móviles, centrado en los requerimientos específicos y un ciclo de desarrollo muy rápido. Además se integra la sensibilidad temporal al contexto a la metodología y a las particularidades de cada niño, quedando fuera del alcance la sensibilidad al contexto de ubicación espacial.

Este trabajo se estructura de la siguiente manera: En el *capítulo 1* se presenta el problema y se definen los objetivos del trabajo final de graduación. En el *capítulo 2* se desarrollan los marcos referenciales de base para la investigación, mientras que en el *capítulo 3* se describen y citan ejemplos de SAC electrónicos y algunos manuales disponibles en la actualidad. La aplicación del método propuesto para el desarrollo del SAC comienza a describirse en el *capítulo 4*, a partir de la explicitación de la *Fase de Exploración*. En el *capítulo 5* se presenta la *Fase de Inicialización* que permite obtener una mejor comprensión a cerca del producto general. Se destina el *capítulo 6* para la *Fase de Producción*, y es aquí en donde se implementa la funcionalidad del prototipo mediante iteraciones incrementales. Por último, se sintetizan las *conclusiones* obtenidas tras el desarrollo habiendo aplicado el método Mobile-D en el *capítulo 7*, además se mencionan también línea de investigación para trabajos futuros. El trabajo cuenta además con un *Anexo* destinado a agregar información de utilidad obtenida como parte de la investigación

Capítulo 1 - Presentación del Problema y Objetivos



1.1. PLANTEO DEL PROBLEMA

El IPRI es un organismo dedicado a la asistencia de niños, adolescentes y adultos con daños neurológicos de la provincia de Santiago del Estero. El objetivo principal del IPRI es lograr la reinserción de estos pacientes en el ámbito social, familiar y laboral. Dentro de la población que atiende, se encuentran niños con PCI que, además de alteraciones en el tono muscular, la postura y el movimiento, les ocasiona trastornos visuales, auditivos, cognitivos, emocionales y del lenguaje.

En particular, los niños que asisten al Instituto son tratados con diferentes técnicas y tratamientos médico-terapéuticos, y cuentan con el apoyo de la Escuela Especial Hellen Keller que se encarga de la tarea educativa y que se encuentra integrada dentro del IPRI.

El trabajo que se propone está enfocado especialmente a las necesidades de comunicación oral de los niños entre 2 a 10 años de edad, que poseen PCI leve a moderada, y que actualmente logran comunicarse con dificultad. En este momento, los profesionales encargados de tratar a estos niños utilizan SAC manuales, como tableros y libros de comunicación, que le permiten a cada niño expresar necesidades, pensamientos y sentimientos. El SAC está basado en el uso de símbolos, tanto gráficos como por ejemplo fotografías, dibujos, pictogramas, palabras o letras; como gestuales como la mímica, los gestos o los signos manuales. La desventaja principal de estos SAC es que son poco prácticos para utilizarse fuera del IPRI, y muchas veces difíciles de transportar o portar.

Existen SAC electrónicos disponibles, los cuales mayormente están diseñados para personas con trastorno del espectro autista (TEA). Sin embargo esta alteración es diferente a la PCI ya que las personas con TEA presentan un aspecto físico-motriz dentro de los parámetros normales, aunque suele retrasarse la adquisición del habla, y se acentúan dificultades del lenguaje y de sociabilización. A pesar de ello, los niños con este tipo de trastorno poseen un nivel de memoria elevado y la tendencia a uso de estereotipos (Calderón, 2012), por lo cual las herramientas diseñadas para el TEA no consideran la disminución motriz que se presenta en la PCI. Las aplicaciones existentes no incorporan los diferentes modismos propios de la cultura en la que se encuentran inmersos los pacientes, así como tampoco las conjugaciones del habla cotidiana y la terminología propia para referir a objetos o acciones que hagan uso del idioma de la zona.

Sintéticamente, se propone el desarrollo de un SAC portable destinado a niños de 2 a 10 años de edad con dificultades en su lenguaje oral, que funcione en plataformas tecnológicas disponibles en la actualidad y de alcance masivo, como son por ejemplo los dispositivos móviles y que facilite la emisión de necesidades y emociones de estos niños en su lenguaje nativo, además de adaptarse al contexto temporal en el cual se encuentra.

1.2. OBJETIVOS DEL TRABAJO

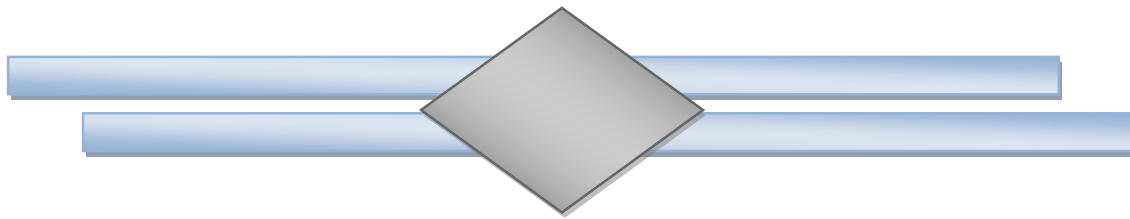
El objetivo general de este trabajo final de grado se formula de la siguiente manera:

Diseñar una herramienta basada en la computación móvil que sirva de apoyo al proceso de comunicación oral de los niños con PCI.

Mientras que los objetivos específicos que se persiguen son:

- ✓ Integrar las funcionalidades de los SAC implementados manualmente en el IPRI en un sistema aprovechable desde una plataforma móvil.
- ✓ Incorporar al diseño de la interfaz los aspectos de: agrupación de símbolos y correcta distribución espacial de los mismos; adecuado uso de símbolos; emisión por voz del significado atendiendo el género del usuario.
- ✓ Favorecer la fácil interpretación de los símbolos utilizados, y su traducción al lenguaje nativo oral.

Capítulo 2 - Marcos Referenciales



Este capítulo se divide en dos secciones principales. En primer lugar se abordan, en el *marco teórico*, los conceptos principales sobre los cuales se sustenta esta investigación. Entre ellos se encuentran conceptos específicos del área de aplicación. Por otro lado, en el *marco procedimental* se hace referencia a los métodos ágiles para el desarrollo de software, en especial al método sobre el cual se basa esta investigación, conocido como Mobile-D.

2.1. MARCO TEÓRICO

Se presentan a continuación los conceptos teóricos referidos a PCI ya que el producto que se quiere desarrollar está destinado a este dominio de aplicación; los SAC debido que se desea construir un prototipo que se encuadra dentro de esta clase de sistemas; el concepto de Aplicaciones Móviles esencial para esta investigación; el Sistema Operativo (SO) Android por ser la plataforma de desarrollo que se utilizará; y la definición del concepto de sensibilidad al contexto inherente al tipo de sistemas que se quiere desarrollar.

2.1.1. Parálisis Cerebral Infantil

La PCI es un término utilizado para hacer referencia a un grupo de trastornos del desarrollo, del movimiento y de la postura, causantes de una limitación de la actividad. No es una enfermedad, sino una discapacidad (Robaina-Castellanos, G., Riesgo-Rodríguez, S., & Robaina-Castellanos, M., 2007) producto de una lesión, permanente y no progresiva en el cerebro en desarrollo, sobrevenida durante la gestación, el parto o durante los primeros años de vida del niño. Ésta interfiere en el desarrollo del Sistema Nervioso Central (SNC) por lo que, una vez producido el daño, éste repercute en el proceso madurativo del cerebro y, por tanto, en el desarrollo del niño. Generalmente conlleva a trastornos motores, sensoriales, cognitivos, de la comunicación, de la conducta, perceptivos y/o intelectuales, que pueden o no estar acompañados de epilepsia (Argüelles, 2008).

El grado en el que la PCI afecta a cada individuo es diferente, y se determina por el momento concreto en que se produce el daño; así existen personas con una PC que resulta apenas perceptible, desarrollando una vida totalmente normal, frente a otras que necesitan del apoyo de terceras personas para realizar tareas básicas.

Las personas con este grupo de trastornos, requieren la asistencia de profesionales con experiencia en fisioterapia, fonoaudiología, apoyo educativo, terapeutas de la conducta y otros. La intervención y el abordaje desde estas áreas es un trabajo conjunto para que las personas con PCI aprendan habilidades específicas que satisfagan sus necesidades, mejorando sus movimientos, estimulando el desarrollo intelectual y/o comunicacional, así como también potenciando sus habilidades sociales (Robaina-Castellanos, G., Riesgo-Rodríguez, S., & Robaina-Castellanos, M., 2007).

La PCI puede ser clasificada desde un punto de vista funcional según la severidad clínica de la afectación como leve (L), moderada (M) o severa (S): la PCI-L se produce cuando el individuo no está limitado en las actividades de la vida diaria, aunque presenta alguna alteración física; en la PCI-M el individuo tiene dificultades para realizar las actividades diarias y necesita medios de asistencia o apoyo; y en la PCI-S el individuo requiere de apoyos para prácticamente todas las actividades de la vida diaria (Sfaello Z., 2015).

En la PCI a menudo se ve afectada la producción del habla, el lenguaje y los gestos para la comunicación. Esto es porque presentan trastornos del habla, es decir, presentan dificultades con la articulación de palabras, y también trastornos específicos del lenguaje que provocan que no puedan emitir sus ideas (Beltrán, et.al, 2013; Crespo, Narbona, 2003). La correcta pronunciación de cualquier palabra, en niños con estos trastornos, suele resultar una tarea compleja por presentar déficit en la articulación, lo que afecta al aspecto fonético del lenguaje y la funcionalidad del habla. Con el fin de maximizar la capacidad de comunicación y de ayudar a tomar un rol independiente en la interacción de los niños con PCI, los profesionales del área de lenguaje hacen uso de múltiples recursos alternativos en la terapia, como por ejemplo recortes de imágenes, cuadernos estructurados con imágenes representativas, entre otros ²

Los niños con PCI pueden presentar dificultades en la coordinación muscular al realizar movimientos voluntarios, así como también músculos tensos y rígidos y reflejos exagerados, suele involucrar asociados otros trastornos médicos como ser por ejemplo convulsiones, percepción y discriminación de la visión o audición

² <http://espanol.ninds.nih.gov/trastornos/paraliscerebral.htm> [consultado el 30 de Noviembre de 2016]

deteriorada, lo que dificulta su comunicación con el entorno. Para mejorar la capacidad de comunicarse con el mundo desde las terapias del lenguaje se enseña a percibir la información del entorno y a procesarla, para ello se utilizan diferentes recursos³ como ser imágenes, pictogramas, letras, palabras, etc que son explicadas junto a su correspondiente significado, de esta manera desde las terapias se les enseña a utilizar estos recursos para poder comunicarse.

Así, los niños con PCI utilizan sistemas de apoyo conocidos como SAC para poder comunicarse con su entorno. Es por ello que la siguiente sección se dedica a este concepto.

2.1.2. Sistemas Alternativos de Comunicación

Los SAC son medios diseñados para ampliar o reemplazar el habla cuando ésta no existe o se ve limitada por diferentes factores. Son instrumentos de intervención destinados a personas con alteraciones diversas de la comunicación y/o lenguaje. Pueden ser alternativos o aumentativos⁴. Son herramientas auxiliares para el desarrollo comunicativo de cualquier persona que posee una afección temporal o permanente en el lenguaje oral (Baumgart, Johnson, & Helmstetter, 1996). Permiten reducir las limitaciones en el proceso de comunicación con el entorno social, aumentando la autonomía de la persona que no puede comunicarse oralmente (Villa & Martínez, 2008). Los SAC permiten a cada usuario expresar necesidades, pensamientos y sentimientos, haciendo uso de símbolos, tanto gráficos como ser fotografías, dibujos, pictogramas, palabras o letras; o gestuales como por ejemplo mímica, gestos o signos manuales. Son ejemplos de SAC el lenguaje de señas, los tableros y libros de comunicación con pictogramas y los comunicadores electrónicos que incorporan el sonido como salida (Fernández, Sánchez, Roldán, Rodríguez-Fórtiz, Hurtado-Torres, & Medina-Medina, 2009).

Los SAC se pueden clasificar de acuerdo a si hacen o no uso de recursos externos para la comunicación (Fábrega, 2014; Villa & Martínez, 2008). Por un lado

³ Ibídem

⁴ <http://arasaac.org/> Portal de Aragón de la Comunicación Aumentativa y Alternativa

[consultado ultima vez 22 de Febrero del 2017]

existen SAC sin ayuda gráfica, que son aquellos basados en los movimientos de las distintas partes del cuerpo del usuario para expresarse. Éstos incluyen:

- Los gestos de uso común, como la sonrisa;
- Los códigos gestuales no lingüísticos creados especialmente o adaptados con fines educativos o terapéuticos;
- Los sistemas de signos manuales de los no oyentes que poseen reglas de estructuración y combinación de los signos;
- Los sistemas de signos manuales pedagógicos como por ejemplo, la comunicación bimodal, el sistema de comunicación total o el vocabulario Makaton, en los que se utiliza una combinación de signos manuales con lenguaje oral (Alonso, Modinos, & Sánchez, 2009).
- Lenguajes codificados gestuales, tales como el código Morse que se basa en la transmisión y recepción de mensajes utilizando sonidos o rayos de luz junto a un alfabeto alfanumérico que se compone de puntos y rayas; y la palabra completada (*Cued Speech*) basada en la lectura labiofacial complementada con configuraciones de la posición de las manos sobre el rostro.

Asimismo, existen otros SAC con ayuda gráfica, que permiten que el usuario haga uso de diferentes recursos materiales -como imágenes y objetos - para poder comunicarse. Esta categoría incluye (Fábrega, 2014):

- Sistemas basados en elementos muy representativos o símbolos iconográficos: como objetos reales, miniaturas, fotografías, o dibujos fotográficos;
- Sistemas basados en dibujos lineales o pictogramas;
- Sistemas que combinan símbolos pictográficos, ideográficos y arbitrarios;
- Sistemas basados en las experiencias de enseñanza del lenguaje a antropoides;
- Sistemas basados en la ortografía tradicional como letras, sílabas, palabras o frases;
- Lenguajes codificados o comunicadores del habla artificial.

En la actualidad se utilizan ambas categorías de SAC en forma combinada para favorecer la comunicación.

En el marco de esta investigación es importante presentar las ventajas y desventajas de los SAC, acorde a la categoría a la cual pertenecen. Por ejemplo, los SAC sin ayuda gráfica pueden iniciarse desde muy tempranas edades, se adaptan a

las posibilidades motrices del usuario lo cual potencia su accesibilidad, pero requieren que éste posea algún tipo de motricidad. El proceso comunicativo utilizando estos SAC puede ser bastante rápido, aunque requiere mucha atención por parte del interlocutor, ya que la comunicación no queda escrita ni grabada porque se representa en el espacio. Como exigencia adicional para el interlocutor, los SAC sin ayuda gráfica requieren el conocimiento de los signos comunicativos. Por otra parte, propician una comunicación limitada a pocas personas, es decir favorecen solamente las necesidades de comunicación con las personas del entorno más cercano al usuario. En forma complementaria, en los SAC con ayuda gráfica, los elementos pictográficos pueden ser señalados por cualquier parte del cuerpo que tenga movilidad, aunque para comunicarse en distintos ámbitos el usuario debe tener disponible la “ayuda gráfica” permanentemente. El proceso comunicativo en este tipo de SAC es más lento y requiere cierta atención y habilidades de discriminación visual por parte del emisor, aunque la atención requerida por el receptor es menor. El lenguaje soportado por ayudas gráficas es más fluido por ser comprensible por los receptores.

La breve descripción presentada es útil para enmarcar conceptualmente el SAC cuyo prototipo se desarrollará en la presente investigación: un *SAC portable con ayuda gráfica*. Para diseñarlo y ponerlo en funcionamiento, es conveniente introducir el concepto de aplicaciones móviles y de SO Android, a lo que se dedican las siguientes secciones.

2.1.3. Aplicaciones Móviles

Según se define en (Santiago, Trinaldo, Kamijo, & Fernández, 2015) una aplicación móvil es toda aplicación informática que ha sido diseñada para ejecutarse en dispositivos móviles (teléfonos inteligentes, tabletas y otros). Este tipo de aplicaciones son distribuidas a través de plataformas pertenecientes a las compañías propietarias de los SO móviles (Android, iOS, BlackBerry OS y Windows Phone, entre otros) para las cuales son desarrolladas.

Las aplicaciones móviles se dividen en nativas, webs e híbridas⁵. Cada tipo de aplicación tiene un conjunto único de características que condicionan el tipo de SO a utilizar (Santiago, Trbaldo, Kamijo, & Fernández, 2015). Se describe esta clasificación de aplicaciones móviles a continuación:

- *Aplicaciones nativas*: se integran o instalan en el dispositivo desde las librerías de cada uno de los SO para móviles. Este tipo de aplicaciones se desarrolla específicamente para la plataforma del dispositivo en el que operará la aplicación. Las aplicaciones nativas tienen archivos ejecutables binarios que se descargan directamente al dispositivo y se almacenan localmente. Las ventajas de las aplicaciones nativas son la buena *performance* ya que son programas instalados en el dispositivo real, por lo cual pueden acceder a todas las características del hardware del móvil, como por ejemplo cámara, GPS, agenda, dispositivos de almacenamiento (Restrepo Lenis, 2016), no requieren conexión activa a Internet para su uso, y están disponibles a través de tiendas de aplicaciones de consumo público. Sin embargo, presentan desventajas tales como su dependencia a la plataforma, deben estar instaladas en el dispositivo, y necesitan ser actualizadas cuando la aplicación cambia.
- *Aplicaciones web*: están disponibles con una conexión a Internet utilizando un navegador web instalado en el dispositivo móvil. El explorador actúa como interfaz entre el dispositivo móvil y el código de la aplicación en un servidor web. Entre las ventajas de este tipo de aplicaciones se citan: el mismo código de base se puede reutilizar en diferentes plataformas, el usuario siempre dispone de la última versión, y están desarrolladas en lenguajes muy conocidos por los programadores. Entre las desventajas principales de las aplicaciones web se mencionan el hecho de requerir constantemente una conexión a Internet, el tiempo de respuesta es menor que el de las aplicaciones nativas, y tiene un acceso restringido a las características del *hardware* (Quishpe Pila, 2013).

5

ftp://ftp.software.ibm.com/la/documents/gb/commons/27754_IBM_WP_Native_Web_or_hybrid_2846853.pdf. [consultado el 16 de Octubre del 2016]

- Aplicaciones *híbridas*: combinan una aplicación nativa con una aplicación web, conteniendo partes o funciones de ambas. La instalación es nativa pero están codificadas para ser utilizadas en múltiples plataformas. Además, este tipo de aplicaciones permiten acceder a las características del *hardware* del dispositivo. Pero la principal ventaja que poseen las aplicaciones híbridas es la posibilidad de ser distribuidas en la tienda de aplicaciones. La principal desventaja es que el diseño visual no siempre se adapta al SO (Bartz, 2015).

La aplicación que se propone en el presente trabajo corresponde al tipo nativa, y para desarrollarla se utilizará el SO Android, que se introduce brevemente a continuación.

2.1.4. SO Android

Android es un SO de código abierto, que presenta varias versiones o evoluciones, y es aplicable a dispositivos móviles. El presentar varias versiones ocasiona que se trate de un SO fragmentado en versiones intermedias a las versiones oficiales, pudiéndose presentar problemas de incompatibilidad con algunas aplicaciones (Polanco, & Taibo, 2011).

Se trata de un SO personalizable, lo que significa que permite agregar, quitar o mover en el escritorio *widgets* - como por ejemplo de tiempo - e instalar aplicaciones y complementos alternativos que pueden descargarse desde un sitio oficial conocido como *Google Play* (Cataldi, & Lage, 2012). Otra característica importante de Android es que permite gestionar varias aplicaciones al mismo tiempo evitando un consumo excesivo de batería, aunque esto agrega dificultad para configurar ciertas funciones.

La estructura de este SO se basa en Linux con una interfaz de programación Java. Contiene herramientas conocidas como Android SDK (*Software Development Kit*) que posibilitan el desarrollo de aplicaciones para Android. El SDK incluye un compilador, un depurador, un emulador de dispositivos, y su propia máquina virtual para ejecutar programas. Además, proporciona una biblioteca de interfaz de usuario y soporta gráficos 2D y 3D, mediante el uso de librerías específicas como *OpenGL*. Cuenta además con un gestor de base de datos integrado (Méndez Martín, 2015).

En cuanto a la arquitectura de este SO, es una arquitectura de capas que se muestra en la Figura 2.1, donde cada una de las capas utiliza elementos de la capa inferior para satisfacer sus funciones. Los componentes de la arquitectura pueden describirse brevemente continuación, según lo consignado en (Barranco Mesa, 2012):

- *Capa 0.* En esta se encuentra el núcleo o *Kernel* del SO, en este caso, *Linux Kernel*. Android está basado en Linux 2.6. Esta capa actúa como una capa de abstracción entre el *hardware* del dispositivo y el resto de capas. Para poder acceder a ella se utilizan librerías que se encuentran en la capa superior, evitando tener que conocer las características de cada dispositivo. Otra función del *Kernel* es gestionar los diferentes recursos del teléfono y del SO en sí.
- *Capa 1.* Es la encargada de proporcionar funcionalidades a las aplicaciones para las tareas que se utilizan con frecuencia, garantizando que se lleven a cabo con eficiencia. Se encuentra separada en dos secciones: *Librerías* y *Android Runtime*. *Librerías* contiene todas las librerías nativas de Android, escritas en C o C++ y compiladas según el *hardware* específico del dispositivo. La sección *Android Runtime*, contiene algunas de las librerías básicas del lenguaje JAVA y la máquina virtual *Dalvik* (DVM). DVM permite conseguir la portabilidad en las aplicaciones desarrolladas en Android, pues las aplicaciones se codifican en JAVA y son compiladas para ser ejecutadas en Dalvik, consiguiendo una aplicación funcional en cualquier dispositivo.
- *Capa 2.* En ésta se encuentra el *Framework* de Aplicación, conformado por todas las clases y servicios que utilizan las aplicaciones para realizar sus funciones.
- *Capa 3:* Es la capa de Aplicaciones en donde se encuentran todas las aplicaciones que hay instaladas en el SO, tanto las que vienen por defecto como las que el usuario ha instalado.

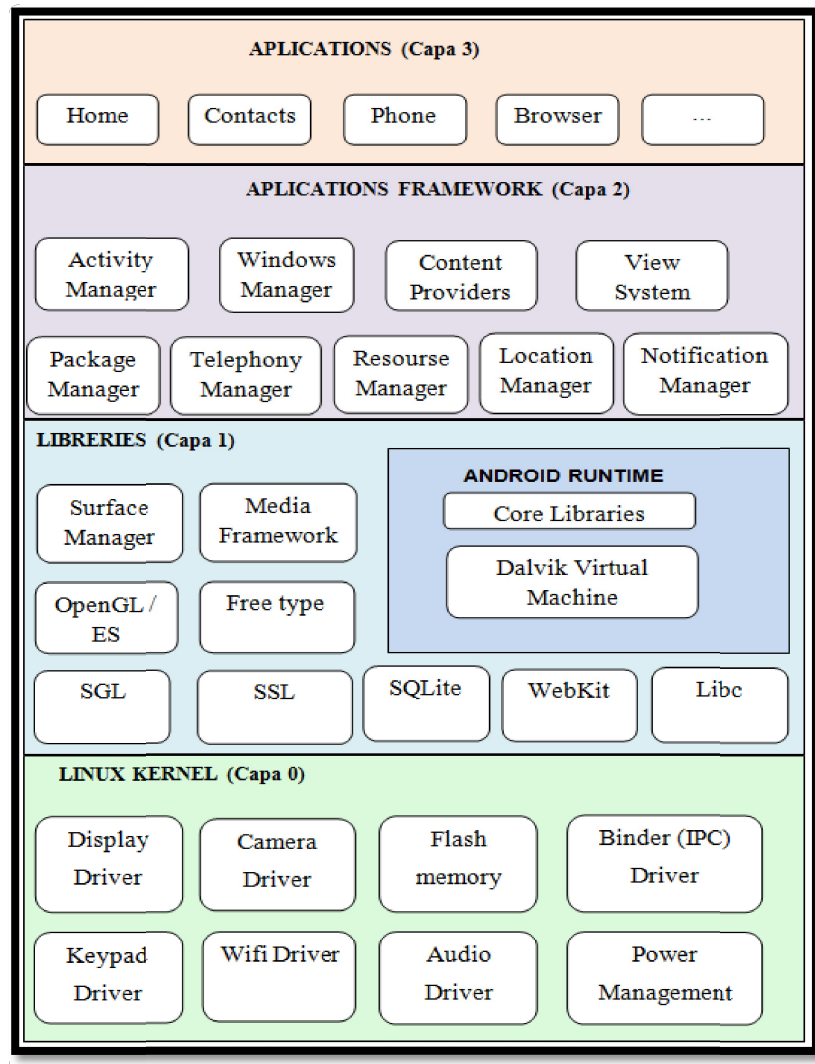


Figura 2-1: *Arquitectura del SO Android*

Otra característica importante es que al ser de código abierto, admite el desarrollo de aplicaciones de terceros y proporciona un sistema de paso de mensajes entre aplicaciones. Esto conlleva a facilitar la reutilización de los componentes reduciendo la carga de los desarrolladores y favoreciendo la colaboración entre aplicaciones. El hecho de promover el paso de mensajes para generar aplicaciones colaborativas, implica que los desarrolladores tomen los recaudos necesarios para evitar mensajes maliciosos que puedan comprometer la privacidad del usuario y las políticas de seguridad de las aplicaciones. Es por ello que existen herramientas que permiten a los desarrolladores detectar la vulnerabilidad antes del lanzamiento de las aplicaciones (Chin, Felt, Greenwood, & Wagner, 2011).

Independientemente del tipo de aplicación que se desarrolle y del SO que se emplee, uno de los aspectos preponderantes en el prototipo a desarrollar es la *sensibilidad al contexto*, concepto que se introduce a continuación.

2.1.5. Sensibilidad al Contexto

La *sensibilidad al contexto* es un concepto que define una propiedad emergente del software, e indica la capacidad de adaptación de éste al usuario, a la tarea y al contexto. Actualmente, es una noción que ha tomado una importancia creciente en investigaciones sobre computación ubicua e inteligencia ambiental, aunque aún se carece de una definición formal y útil para los desarrolladores (Galico, Natanzon, Vega, Matalonga, Solari, 2015). Implica dotar a los sistemas de una cierta consciencia respecto del entorno en el cual se ejecutan. Los servicios de los sistemas con sensibilidad al contexto se establecen dependiendo de la información del contexto o entorno que los rodea. Si bien es cierto que la información contextual puede variar en el tiempo, el funcionamiento y los servicios proporcionados al usuario por parte del sistema también varían reflejando este cambio. Los sistemas sensibles al contexto buscan poder anticipar las necesidades de los usuarios en situaciones particulares, incorporando la capacidad de reflejar ciertos cambios que ocurren en el entorno. A consecuencia de esto, son sistemas que habilitan respuestas selectivas para el usuario y la situación particular, al disparar alarmas por ejemplo (Pascoe, Ryan & Morse, 1999).

Esta característica es importante en la computación móvil dado que constantemente los desarrollos están perfeccionando el contacto directo con las personas y buscando flexibilidad o adaptabilidad en su uso. La funcionalidad adaptable se logra mediante el empleo de técnicas de razonamiento automáticas en el procesamiento de la información (Valenzuela & Terriza, 2008).

Según Loke (Loke, 2006), los sistemas sensibles al contexto deberían poseer tres elementos básicos: *sensores*, *procesamiento* y *actuadores*. Los *sensores* proveen los datos referentes a las actividades y eventos que ocurren en el mundo real y permiten obtener la información del contexto, debiendo considerarse qué tipo de información capturada es relevante para la aplicación en cuestión. El *procesamiento* implementado con *algoritmos de percepción*, provee un sentido y significado a los estímulos recibidos del entorno o contexto. Estos algoritmos son los encargados de clasificar las situaciones en diferentes contextos, para lo cual utilizan distintos modelos existentes como por ejemplo los basados en lenguajes de marcas, los modelos gráficos, orientados a objetos y basados en ontologías (Strang, & Linnhoff-Popien, 2004). Basándose en el contexto observado, el sistema ejecuta diferentes

acciones brindando sus servicios y adaptándose a los cambios percibidos del contexto. Estas acciones son llevadas a cabo por los *actuadores*. El procesamiento de la información del contexto intenta representar el conocimiento del dominio. Para ello se utilizan diferentes técnicas entre las cuales se encuentran el modelo de tripletas objeto-atributo-valor, la lógica *fuzzy* que representa la incertidumbre, los modelos basados en reglas, las redes semánticas o mapas conceptuales, entre otros (Ga, Djuric & Deved, 2006).

Entre la información de contexto relevante para este tipo de sistemas se encuentra (Korkea-Aho, 2000):

- La *identidad del usuario*: quién es, con quién se encuentra en un momento determinado, con qué recursos cuenta, sus preferencias.
- El *espacio*: ubicación, orientación, velocidad y aceleración.
- El *tiempo*: hora del día, fecha, estación del año.
- La *organización de horarios y agendas*.
- El *ambiente*: temperatura, calidad del aire, el nivel de luz o ruido, etc.
- La *situación social* o personas que están cerca, entre otros.

La sensibilidad contextual que se considera en el marco de la presente investigación se enmarca en la sensibilidad temporal, de organización y de identidad del usuario.

2.2. MARCO PROCEDIMENTAL

El método a utilizar para desarrollar el prototipo es Mobile-D, un método ágil para el desarrollo de aplicaciones móviles. A continuación se presenta una breve reseña sobre los métodos ágiles en general, y el método Mobile-D en particular.

2.2.1. eXtreming Programming

XP (Joskowicz, 2008) es una metodología que consta un ciclo de vida dinámico y plantea la realización de ciclos de desarrollo cortos o *iteraciones*, donde se obtienen funcionalidades entregables hasta lograr la funcionalidad completa del sistema solicitado. Cada iteración realiza con un ciclo completo de cuatro fases (Pressman, & Troya, 1988): la fase de *planeación*, la fase de *diseño*, la fase de *codificación* y la fase de *pruebas*.

La fase de *planeación* utiliza las historias de usuario para recolectar los requerimientos que permiten comprender el contexto, la finalidad del sistema, las características principales y funcionalidades requeridas. Luego se decide qué historias se desarrollarán, el tiempo que durará el desarrollo y los recursos y la fecha de entrega del primer incremento. Una vez que se realiza la primera entrega, el cliente puede solicitar el agregado de nuevas historias de usuario. La fase de *diseño*, persigue el objetivo de guiar la implementación de una historia conforme se escribe, sin agregar ni quitar funcionalidad. Utiliza *tarjetas* para identificar y organizar las clases orientadas a objetos que son relevantes para el incremento actual de sistema. Cuando el diseño de una historia encuentra un problema de diseño difícil, se crea un prototipo operativo de esa porción del diseño, luego se implementa y evalúa el prototipo. Se incentiva al rediseño, es decir, la mejora del diseño del código después que ha sido escrito, por lo cual se realiza de manera continua en la medida que se construye el sistema. En la fase de *codificación* se inicia la programación por parejas para crear código para una historia, es decir, cada pareja de programadores trabaja sobre una historia de usuario. Terminado el código se le aplica una prueba unitaria a cada una de las historias de la entrega, así como también se desarrollan las pruebas de aceptación. En la medida que las diferentes parejas de programadores terminan su trabajo, se integran y prueban todos los códigos desarrollados, para lograr una “integración continua” y evitar problemas de compatibilidad. En cuanto a la fase de *pruebas* se aplican pruebas de aceptación XP, que son especificadas por el cliente y se centran en las características y la funcionalidad general del sistema.

La base fundamental de esta metodología son cinco valores integrales subyacentes: comunicación, sencillez, retroalimentación, valor y respeto (Joskowicz, 2008; Pressman, & Troya, 1988; Boehm, 2007). La *comunicación* es un valor fundamental ya que debe ser permanente y debe estar presente durante todo el proyecto. La comunicación puede verse afectada cuando la cantidad de integrantes aumenta, pues los procesos comunicativos se complejizan. La *sencillez* en XP implica que se comienza desarrollando las soluciones a los requerimientos más sencillos que se han planteado. Las funcionalidades más complejas del sistema se añaden en la medida que se adquiere más información sobre los requerimientos iniciales. En esta metodología de desarrollo la *retroalimentación* de información debe suceder de forma permanente, ya que los desarrolladores deben brindar información cierta sobre las funcionalidades desarrolladas al cliente al finalizar cada

iteración, y como respuesta éste debe brindar una retroalimentación sobre las funciones esperadas. Al ser una metodología de desarrollo dinámica, requiere *valor* o *disciplina* de equipo para reconocer requerimientos cambiantes en el tiempo lo que demanda repeticiones sustanciales del diseño y del código implementado. Además, es muy importante el *respeto* mutuo sobre experiencias y opiniones, tanto entre desarrolladores como con el cliente.

En general, XP no se basa en la documentación formal escrita en sí, sino que ésta se realiza través de la comunicación oral, el código, y la transmisión de conocimientos entre los integrantes. Por ello los sistemas obtenidos con esta metodología tienen dificultades en el mantenimiento (Joskowicz, 2008).

2.2.2. SCRUM

SCRUM (Gallego, 2012) es un marco de trabajo especialmente aplicable a proyectos con requerimientos altamente cambiantes o poco definidos, o que necesitan resultados rápidos. Sus principales características son dos: por un lado el desarrollo de software se realiza mediante iteraciones que duran 30 días, denominadas *sprints*, donde el resultado de cada *sprint* es un incremento ejecutable que se muestra al cliente; por otro, el proyecto está basado en reuniones, entre las que se destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración. El desarrollo de cada *sprint* permite ir ajustando la funcionalidad ya construida en la iteración anterior, e ir añadiendo nuevas funciones priorizando aquellas que agreguen más beneficios al cliente.

En este marco se reúnen un conjunto de patrones que se centran en las prioridades del proyecto, en las unidades de trabajo agrupadas, en la comunicación y la retroalimentación permanente con el cliente. El proceso de construcción es iterativo e incremental y el trabajo es realizado por equipos auto-organizados y auto-dirigidos, para lograr la motivación, responsabilidad y compromiso necesario para el desarrollo.

SCRUM se desarrolla en forma general alrededor de tres roles, tres actividades y tres artefactos (Escobar, & Gualteros, 2013). Los *roles* son: propietario del producto, gestor o responsable de que el equipo sea funcional y productivo, y por último integrantes del equipo de desarrollo. Las *actividades* que se consideran en este marco de desarrollo son las reuniones de planificación del siguiente *sprint*, que incluye la siguiente entrega, las reuniones diarias para el seguimiento y control, y las

reuniones de revisión. Los *artefactos* son la *bitácora del producto* o lista priorizada de requerimientos, *bitácora de Sprint* o planificación de las tareas de la siguiente iteración, y el *diagrama de progreso* o marca de seguimiento de los diferentes *sprints*.

2.2.3. Feature Drive Development

Feature Drive Development (FDD) (García Rodríguez, 2015) es un modelo de proceso iterativo incremental. Propone iteraciones cortas que implementan una serie de características del sistema, en donde cada característica individual es un pequeño bloque de funcionalidad entregable. El proceso consta de 5 fases:

- Desarrollo de un modelo global que se construye en base a la visión, contexto y requerimientos del sistema a construir.
- Construcción de una lista de características o resumen de las funcionalidades del sistema, la cual es evaluada por los clientes. Las funcionalidades se subdividen para un mejor entendimiento.
- Planificación de funcionalidades o priorización de las mismas, para poder luego ser asignadas a los programadores.
- Diseño de un conjunto de funcionalidades de la lista. En este caso es importante la decisión de cuál funcionalidad se va a realizar en cada iteración.
- Construcción total del proyecto.

2.2.4. Mobile-D

Mobile-D⁶ es un método ágil para el desarrollo de software móvil. Se basa en algunas prácticas de desarrollo tomadas de XP, en la escalabilidad provista por la familia de metodologías *Crystal*, y en el ciclo de vida RUP (*Rational Unified Process*). Se basa en ciclos de desarrollo muy rápidos, y es aconsejable cuando el equipo de desarrollo es muy pequeño, es decir, equipos que cuentan hasta con 10 desarrolladores. Está destinado a lograr la implementación de proyectos cuyo objetivo principal es el lanzamiento de productos completamente funcionales aproximadamente en diez semanas.

⁶ <http://agile.vtt.fi/mobiled.html> [consultado el 2 de Agosto del 2016]

En la figura 2.2 se muestra el ciclo del proyecto, en el se pueden observar las fases y etapas del método.

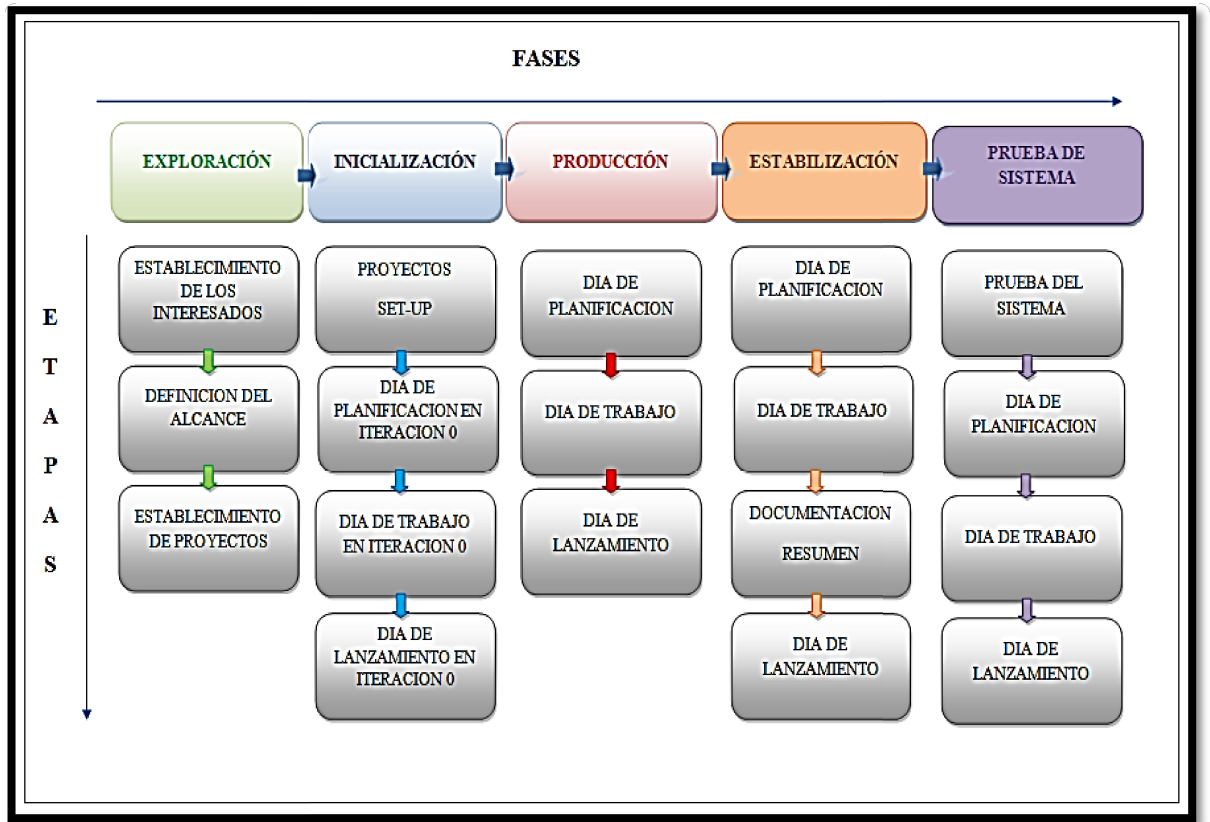


Figura 2-2: Fases y etapas del método Mobile-D

En general, todas las fases con excepción de la primera contienen tres momentos de desarrollo distintos: planificación, trabajo y liberación. Las fases se describen a continuación:

- Fase de Exploración:** Plantea el establecimiento de un plan de proyecto y los conceptos básicos inherentes al mismo. La participación de los clientes es muy importante en esta fase. Consiste en las siguientes etapas:
 - Establecer los interesados*, se definen la información relevante de los grupos de actores, los roles y los recursos pertinentes a las diferentes tareas del proyecto. Además se pueden incluir el grupo de dirección, de gestión de proyectos, el grupo de arquitectura y especialistas de proceso. Todos estos actores juegan un papel vital en las tareas posteriores de la fase exploración y en la ejecución del proyecto.

2. *Definir el Alcance*, se definen los objetivos y el alcance del proyecto de desarrollo de software, acordados con los grupos de interés. Esto incluye cuestiones tales como los requisitos para el producto y la línea de tiempo del proyecto.
 3. *Establecer el proyecto*, para acordar temas ambientales físicos y técnicos del proyecto, así como el personal necesario tanto para el desarrollo de software como personal de apoyo.
 4. *Normalmente esta fase comprende la identificación de los interesados y de las necesidades de desarrollo del sistema, una clasificación de requisitos, la planificación inicial del proyecto, y la planificación de la línea de base.*
- **Fase de Inicialización:** Los desarrolladores preparan e identifican todos los recursos necesarios para el desarrollo de la aplicación. Se preparan los planes para las siguientes fases y se establece el entorno técnico o línea arquitectónica. En esta fase, los desarrolladores analizan el conocimiento y los patrones arquitectónicos utilizados extraídos de proyectos anteriores de la empresa, y los relacionan con el proyecto actual. Se agregan observaciones, se identifican similitudes y se extraen soluciones viables para su aplicación en el proyecto. Las etapas individuales de la fase inicialización son las siguientes:
 1. *Set-Up de los proyectos*, donde se ponen en marcha los recursos físicos y técnicos para el proyecto, a fin de monitorear el medio ambiente. Se capacita al equipo de proyecto si es necesario, y se establecen las formas específicas de comunicación con el grupo de clientes.
 2. *Planificación en Iteración 0*, para obtener una buena comprensión global del producto basándose en los requisitos y en las descripciones de la arquitectura. Además se quieren perfeccionar las descripciones detalladas acerca de la arquitectura y del plan del proyecto, propiciando la creación de un plan de arquitectura. En esta etapa se puede preparar planes para comprobar el estado de preparación de temas críticos de desarrollo, como el entorno de desarrollo, los elementos arquitectónicos y otras entidades externas del software a ser desarrollado, así como la comunicación

entre los elementos y las entidades. Incluye definir el número de iteraciones estimadas para el desarrollo, así como también los puntos de control necesarios.

3. *Día de trabajo en Iteración 0*, se realizan las verificaciones y configuraciones del entorno de desarrollo técnico. Además se corrobora que todo esté listo para la implementar el producto de desarrollo de software. En este día también se comunica el progreso, se busca incrementar la funcionalidad básica del sistema o resolver algún problema pendiente de desarrollo crítico en caso de ser posible.
- **Fase de Producción:** Se repiten la planificación, el trabajo y el lanzamiento iterativamente hasta implementar todas las funcionalidades. Primero se planifica la iteración de trabajo en términos de requisitos y tareas a realizar. Se preparan las pruebas de la iteración. Las tareas se llevan a cabo durante el momento de trabajo, desarrollando e integrando el código con los repositorios existentes. Luego se produce la integración del sistema en aquellos casos en los que varios equipos estuvieran trabajando de forma independiente, seguida de las pruebas de aceptación. Consiste en las siguientes etapas:
 1. *Planificación* donde se define el contenido o tareas para la iteración.
 2. *Día de trabajo* para decidir cómo implementar la funcionalidad de una manera controlada y gestionada.
 3. *Día de lanzamiento* para verificar y validar la funcionalidad implementada. Normalmente culmina en un lanzamiento real, pero puede ser un lanzamiento “no oficial” ya que el producto sólo se evalúa con los principales actores del proyecto.
 - **Fase de Estabilización:** Se llevan a cabo las últimas acciones de integración para asegurar el correcto funcionamiento del sistema completo. Es la fase más importante en los proyecto multi-equipo, donde cada equipo desarrolla diferentes subsistemas. Todas las actividades de los desarrolladores están orientadas a la integración del sistema. Adicionalmente se puede considerar en esta fase la producción de documentación. Las etapas individuales de la fase estabilización son las siguientes:

1. *Planificación* que consiste en definir las tareas para la ejecución del resto de las características del producto y para mejorar la calidad externa e interna del mismo.
 2. *Día de trabajo* para finalizar la aplicación del producto, así como mejorar y asegurar la calidad del mismo.
 3. *Documentación resumen*, que consiste en finalizar los documentos de la arquitectura de software, diseño e interfaz de usuario.
 4. *Día de lanzamiento* para verificar y validar la funcionalidad implementada y la calidad de todo el software y su documentación. El día del lanzamiento culmina en la versión final del software.
- **Fase de Prueba y Reparación del Sistema:** tiene como meta la disponibilidad de una versión estable y plenamente funcional del sistema. Se prueba el producto terminado e integrado con los requisitos de cliente y se eliminan todos los defectos encontrados. Las etapas individuales de la fase prueba y reparación del sistema son:
 1. *Prueba del sistema* o etapa en la que se prueba el sistema como se describe en el modelo de tarea de prueba. Se documentan los defectos encontrados, con el fin de resolverlos en una iteración denominada *FIX*.
 2. *Planificación* para definir las tareas de la iteración *FIX*. Los defectos encontrados en la etapa de prueba del sistema son entradas para la descripción de las tareas.
 3. *Día de trabajo* destinado a reparar los defectos encontrados y finalizar la implementación del producto. Se pretende además documentar un resumen de la arquitectura del software, diseño e interfaz de usuario. La documentación se actualiza para corresponder con los cambios realizados durante la iteración *FIX*.
 4. *Día de lanzamiento* destinado a verificar y validar la funcionalidad implementada y la calidad de todo el software y su documentación. Culmina en la versión final de todo el software.

El método Mobile-D⁷ puede ser aplicado a diversos dominios, como por ejemplo el área de finanzas o de seguridad.

2.3. CONCLUSIÓN

En este capítulo se definieron dos secciones generales, por un lado el marco teórico que rige esta investigación y por otro el marco procedimental que orienta el desarrollo del prototipo del presente trabajo.

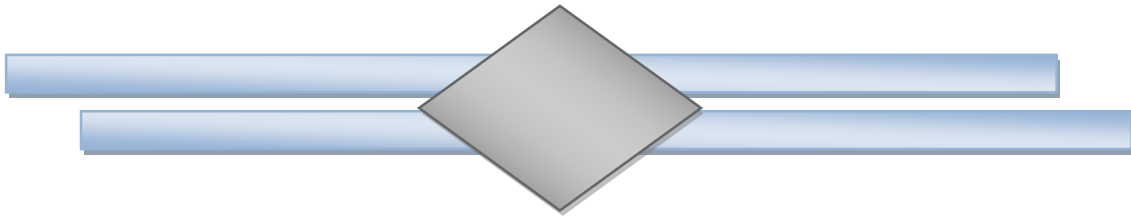
En el *marco teórico* se presentaron definiciones de PCI, así como también aspectos sobre cómo ésta influye en el funcionamiento del sistema nervioso central en forma general, afectando la comunicación de quien la presenta. Además se explicó brevemente cómo los niños encuentran una solución a este problema mediante el uso de un SAC.

Debido al alcance de esta investigación fue necesario describir aspectos inherentes a las aplicaciones móviles, como por ejemplo la definición sobre el sistema operativo que se utiliza para el trabajo, comentando sus características, sus diferentes capas e interacción entre ellas. Asimismo, el concepto de sensibilidad al contexto en este tipo de aplicaciones es muy importante, por lo que se plasmaron algunas definiciones y clasificaciones al respecto.

Por último, dentro del *marco procedimental* se procedió a introducir diferentes herramientas para el desarrollo ágil de aplicaciones, presentando en este caso XP, Scrum y FDD. Se dedicó mayor nivel de detalle al método Mobile-D por ser el que orienta el desarrollo de ésta investigación, dadas sus características. Es decir, la utilización de este método resulta apropiada por las siguientes razones: i) es adecuado para un problema de pequeña envergadura, ii) es adaptable a un desarrollador, iii) ahonda en la exploración del dominio, con una visión integrada software-necesidades específicas de contexto, como lo es el dominio de aplicación del IPRI.

⁷ <http://agile.vtt.fi/mobiled.html>, [consultado el 17 de Agosto de 2016]

Capítulo 3 - Estado del Arte



Como se ha expresado en el capítulo anterior, los SAC se utilizan como sistemas de apoyo al proceso comunicativo para que éste se desarrolle con mayor fluidez en contextos donde la comunicación oral se ve imposibilitada o dificultada. Si bien existen diferentes tipos, en este capítulo se exploran y presentan algunos ejemplos de SAC que se basan en tecnología o comunicadores electrónicos, dado el alcance de la presente investigación. Se mencionan también los SAC manuales a modo de ilustrar el origen de los comunicadores.

3.1. SAC ELECTRÓNICOS MANUALES

Se pueden identificar múltiples y variados SAC para dar soporte a pacientes con afecciones en el habla o en el lenguaje, ya sean aplicaciones tanto pagas como gratuitas. Las mismas se encuentran disponibles tanto para estaciones de trabajo como para dispositivos móviles. Debido a los alcances del presente caso de estudio, se describen a continuación algunos SAC desarrollados para el SO Android.

3.1.1. **Proyect@ PECS**

*Proyect@ PECS*⁸: Es una aplicación en español, estándar, gratuita, y disponible para Android versión 2.2 o superiores. Fue desarrollada por un estudiante para obtener su título de grado en la universidad de Valparaíso (Chile). Diseñada para proporcionar apoyo a comunicación aumentativa-alternativa, está planteada principalmente para personas con trastornos del espectro autista (TEA). La aplicación está basada en el sistema de comunicación por intercambio de imágenes. Utiliza los pictogramas disponibles en <http://www.catedu.es/arasaac/>. Permite iniciar un tablero donde se van colocando las imágenes que corresponden al mensaje que el usuario intenta transmitir. La reproducción sonora o de voz grabada que acompaña a cada imagen colocada en el tablero, se produce solo cuando el usuario selecciona una a una las mismas. Como resultado se obtiene un mensaje sonoro desde el dispositivo. En la figura 3.1 se muestra la pantalla principal de la aplicación en la que se realiza una jerarquización de conceptos, es decir, las imágenes se agrupan por categorías que representan las palabras a comunicar.

8 https://play.google.com/store/apps/details?id=air.TestPECS&hl=es_419 [consultado septiembre del 2016]

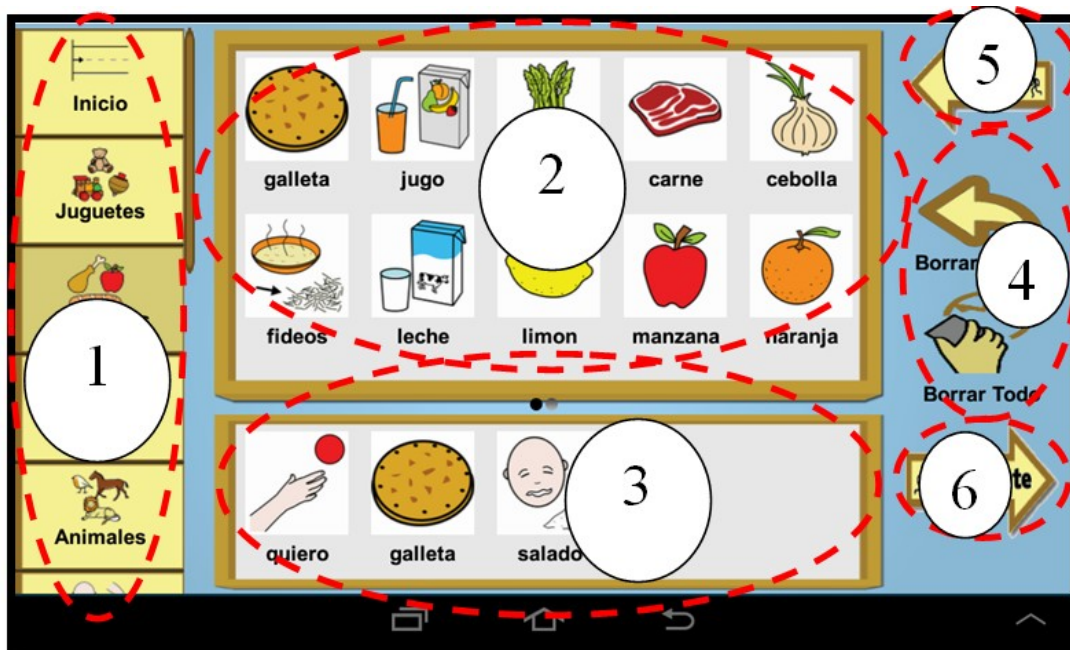


Figura 3-1: *Interfaz principal de Project@ PECS*

Los números de la figura 3.1 indican secciones de la pantalla que se describen a continuación:

1. Categoría de imágenes.
2. Imágenes seleccionables pertenecientes a una categoría específica.
3. Pizarra digital: contiene las imágenes que fueron seleccionadas de una categoría para reproducir en forma sonora.
4. Acciones que se pueden llevar sobre las imágenes de la pizarra.
5. Vuelve a atrás en la acción realizada.
6. Cambia de interfaz para reproducir las imágenes que se encuentran en la pizarra.

3.1.2. Palphoons⁹

Es un comunicador dinámico personalizable, gratuito disponible para Android y para computadoras personales. Destinado a pacientes que no pueden comunicarse mediante la voz o que poseen limitaciones motrices. Puede manejarse con la vista, con mouse común o especial, o táctil. Es una aplicación recomendada

⁹ <http://solca.aig.gob.pa/home/para-discapacitados>. [Consultado el 6 de agosto de 2016]

para personas con Autismo, Síndrome de Down y disminución significativa en la motricidad. Está disponible en versiones estándares predefinidas o personalizadas, en ambos casos se dispone de símbolos seleccionables en la pantalla, que pueden imprimirse o ser escuchados mediante voz sintetizada o grabada. Para personalizar la pantalla es necesario crear nuevos plafones de comunicación que introducen en la pantalla símbolos o pictogramas distribuidos en filas y columnas. La configuración personalizada es compleja y se realiza en computadora. Una vez configurado el plafón está listo para usarse. Además se puede crear una versión portable para usarse en tabletas. Dado que la distribución de los símbolos de la pantalla es mediante filas y columnas, si hay demasiadas filas y/o columnas estos se vuelven demasiado pequeños. Por este motivo la aplicación permite conectar varias “pantallas” mediante acciones configurables de retroceso y avance. En este caso la navegación puede hacerse dificultosa en usuarios con disminución motriz. La utilización del comunicador puede hacerse desde una computadora o desde tabletas, como se muestra en la figura 3.2.

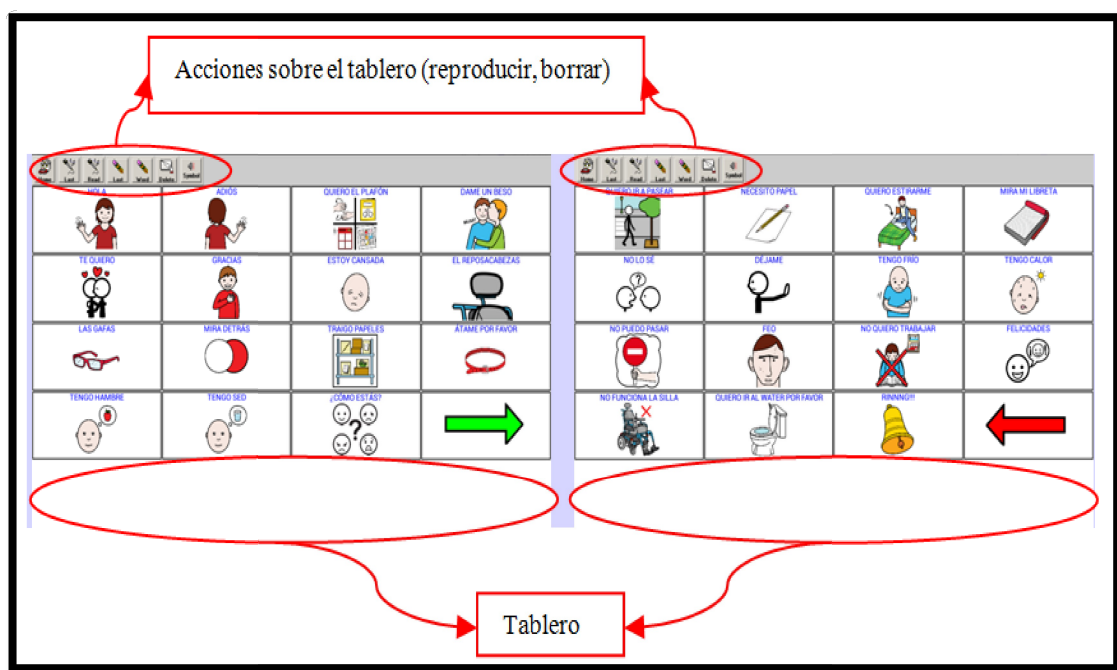


Figura 3-2: Versión configurable en computadora de Plaphoons

En las tabletas la salida de voz se produce solo después de haber colocado los símbolos que se quieren reproducir en el tablero de la parte inferior de la aplicación, y de seleccionar la opción de reproducción disponible en la parte superior izquierda de la pantalla. El contenido del tablero puede ser editado para reproducciones

diferentes, utilizando los iconos del menú de acciones que permite agregar o quitar símbolos.

3.1.3. Sistemas de Comunicación No Verbales

Es un proyecto llevado a cabo por la Universidad Nacional de la Patagonia Austral (UNPA) de Santa Cruz, Argentina en el año 2014. Se enfoca a personas con discapacidad motora y/o cognitiva para proponer el diseño de un SAC Web personalizable, accesible desde computadoras con el mouse y/o el teclado, o con el tacto en caso de tabletas y *smartphones*. El acceso al sistema está restringido al uso de un nombre de usuario y una clave de acceso. Los tipos de usuario pueden ser: administrador, posibilitador o usuario final. El administrador tiene permisos de acceso total; puede realizar la gestión de los usuarios profesionales y el mantenimiento de la base de datos. En cuanto al usuario posibilitador, normalmente inherente a profesionales o familiares, se dan de alta automáticamente y cada uno puede registrarse desde la Web. Estos usuarios pueden crear pictogramas o una nueva imagen, agregar o eliminar usuarios finales, asociar plantillas a usuarios finales, consultar pictogramas disponibles y crear plantillas de pictogramas. Por último, el usuario final o persona con algún grado de discapacidad motora y/o cognitiva, utiliza el sistema para explotar su funcionalidad. A estos usuarios la aplicación le permite realizar selecciones de una plantilla de pictogramas asociada a su sesión, y el sistema da un aviso que puede ser sonoro o de lectura del sinónimo asociado al pictograma seleccionado, como se muestra en la Figura 3.3. Esta aplicación requiere de un asistente que inicie sesión en el sistema y habilite las plantillas para su uso (González, Sosa, & Martín, 2014).

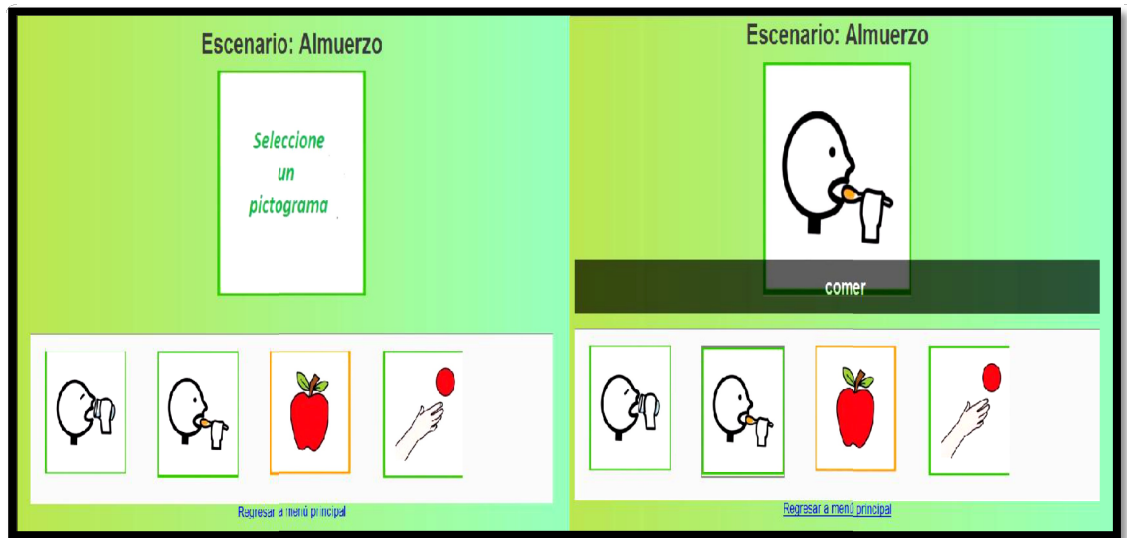


Figura 3-3: *Interfaz de usuario final del sistema de comunicación no verbal*

3.1.4. OTTAA Project¹⁰

Es una aplicación Argentina, desarrollada por cordobeses para ser ejecutada en dispositivos móviles con SO Android. Permite especificar el idioma de reproducción, generando una salida de audio sintetizada. La forma de comunicarse y efectividad de esta herramienta depende de cómo sea configurada, utilizando información del dispositivo como por ejemplo identidad, cuentas del dispositivo, tarjeta de contacto, calendario, eventos de calendario e información confidencial. Con la información recabada la aplicación va sugiriendo pictogramas adecuados para la comunicación de la situación, permitiendo el armado de oraciones con los pictogramas sugeridos, como se muestra en la Figura 3.4.

¹⁰ <http://www.ottaaproject.com/index.html> [consultado 27 Septiembre de 2016]

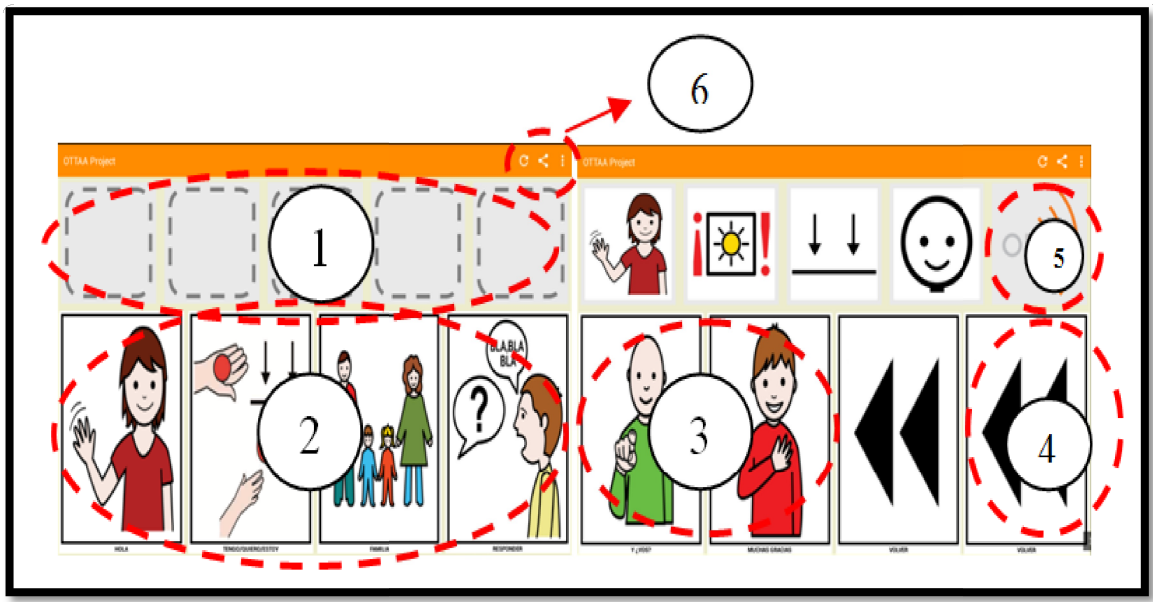


Figura 3-4: Interfaz de reproducción de OTTAA Project

Las opciones numeradas en la Figura 3.4 representan:

1. El *tablero de sección* donde se agregan pictogramas para una reproducción.
2. Categoría principal de pictogramas no modificables, o pictogramas de apertura de todas las frases que son reproducidas. Contienen agrupaciones de pictogramas relacionados al pictograma de apertura.
3. Sección de pictogramas reproducibles o subconjunto de pictogramas que pueden ser editados, es decir, pueden variar dependiendo de las configuraciones personales de cada usuario.
4. Representa la acción de regresar un nivel atrás de los pictogramas.
5. Icono seleccionable para ejecutar la salida sonora que representa la oración creada.
6. Configuración general de la aplicación.

OTTAA es considerado como un SAC predictivo ya que la agrupación de palabras a comunicar puede configurarse utilizando los pictogramas existentes en la base de datos o las imágenes adquiridas desde la cámara del dispositivo. Una versión de prueba está disponible para descarga directa.

3.1.5. AraBoard¹¹

Es un conjunto de herramientas gratuitas diseñadas para la comunicación alternativa y aumentativa. Está basada en el uso de imágenes, audios disponibles en el dispositivo y pictogramas de ARASAAC¹² (Portal Aragonés de la Comunicación Aumentativa y Alternativa), los cuales al ser seleccionados reproducen un sonido representativo de la imagen seleccionada. Permiten crear, editar y usar tableros de comunicación y pueden ser ejecutados en distintos dispositivos como computadoras, teléfonos o tabletas. El conjunto de herramientas está disponible para distintos SO. La instalación de *AraBoard* necesita permisos de acceso a información del dispositivo en el que se instale, como ser fotos, archivos multimedia, cámara y micrófono. Se compone de dos herramientas complementarias *AraBoard Constructor* y *AraBoard Player* que se utiliza para ejecutar los tableros de comunicación previamente creados en *AraBoard Constructor*. *AraBoard Player* requiere que esté cargado el tablero necesario para comunicar. Para reproducir la salida de audio basta con seleccionar el pictograma para que la salida de voz se reproduzca. Requiere Android en 3.2 y versiones superiores. Se encuentra disponible en los varios idiomas.

3.2. SAC – CARACTERÍSTICAS DESEABLES

Para este tipo de aplicaciones, un aspecto muy importante a tener en cuenta es el idioma que emplea el SAC. La mayoría de estas aplicaciones presentan la posibilidad de configurar el idioma de reproducción. En las opciones de reproducción en español, ejecutan una salida (voz sintetizada o grabada) con acentos y terminología propia de otras culturas (España, Centro América o Chile) dificultando el entendimiento en un contexto particular. Por ejemplo, lo que en Argentina se conoce como *hamaca*, en México se reproduce como *columpio*.

Por otro lado, las aplicaciones existentes no distinguen entre usuarios femeninos y masculinos, y cuentan con una salida de voz única para expresar o

¹¹ http://aratools.catedu.es/wp-content/manuales/Manual_AraBoard.pdf [consultado Septiembre del 2016]

¹² <http://arasaac.org/> Portal de Aragón de la Comunicación Aumentativa y Alternativa [consultado ultima vez 22 de Febrero del 2017]

representar ambos géneros. Algunos de estos SAC pueden ser personalizables, aunque requieren tiempo para su configuración y un adiestramiento o práctica considerable para poner en pleno funcionamiento al software.

Otro aspecto a tener en cuenta de los SAC electrónicos disponibles, es que las interfaces son simples, es decir, no están saturadas de imágenes lo cual es una ventaja en su uso. Sin embargo, existen algunos donde el vocabulario disponible es más amplio y requieren varios desplazamientos en la pantalla hasta lograr seleccionar una acción reproducible. A mayor vocabulario más complejo es el uso de la aplicación cuando el usuario presenta dificultades motrices. En otros donde la amplitud del vocabulario es bastante limitada, la selección de pictogramas es más directa. Con ello se concluye que las aplicaciones analizadas deben priorizar la amplitud del vocabulario a costa de la complejidad para la reproducción y viceversa.

Además, es deseable que la distribución de pictogramas y el tamaño de cada uno de ellos, se realice analizando las posibilidades de selección del usuario. Pues iconos demasiado juntos pueden dificultar la selección precisa, y demasiado espaciados limitarán la amplitud del vocabulario disponible. Por ello es necesario considerar la dimensión de la pantalla en la que se ejecutará la aplicación.

En cuanto a la interfaz de los SAC resulta un factor importante que el conjunto de imágenes utilizadas sean conocidas, fáciles comprender por los usuarios y acompañadas por un breve pero significativo texto que las describa. Así también, es importante que en lo posible el vocabulario que el comunicador despliegue en su pantalla sea en función de la actividad que el usuario está por realizar.

3.3. SAC EN EL IPRI

En el IPRI actualmente se utilizan los SAC manuales, como los ilustrados en la figura 3.5. Los mismos no son ampliamente utilizados a nivel mundial, en su mayoría, por la falta de portabilidad que representan. Los más importantes son:

- *Pizarras de comunicación*: Son superficies planas magnéticas o con velcro, donde se adhieren imágenes; el niño con ayuda de su asistente, busca y selecciona una imagen para luego colocarla en la pizarra, indicando de ésta manera qué es lo que necesita.
- *Libros de comunicación*: Son encuadernaciones estándares que tienen plasmadas en sus hojas imágenes de lo que un niño puede comunicar. En este

caso el niño busca y señala lo que necesita. Al ser estándar su diseño, varios niños pueden usar el mismo libro de comunicación.

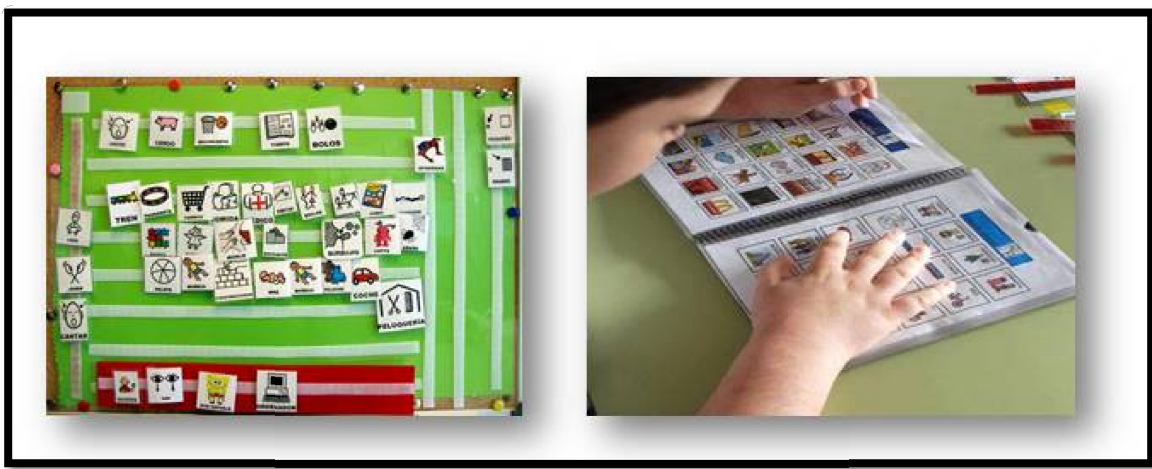


Figura 3-5: Ejemplos de SAC manuales

- *Cuadernos de comunicación*: Son encuadernaciones de hojas con imágenes personalizadas. Son diseñados armados con la familia o con los terapeutas, añadiendo imágenes con pegamento según las necesidades de comunicación de cada niño.

Es importante notar que el lenguaje utilizado en los niños con PCI es el *lenguaje de señas*, principalmente gestual, no requieren asistencia externa, y permite que el niño exprese sus necesidad con su rostro y otras partes de su cuerpo. Sucede que muchas veces debido a la afección del niño con PCI es necesario el uso de un SAC adicional.

3.4. CONCLUSIÓN

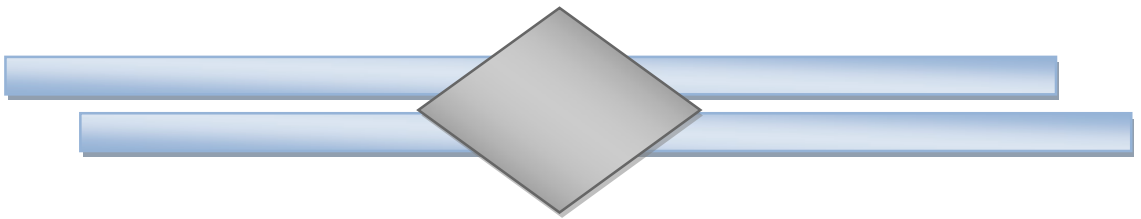
En la mayoría de las terapias de Santiago del Estero, así como también en los distintos ámbitos donde se desenvuelven los niños con PCI, se utilizan los *sistemas de comunicación descriptos como manuales*. Las imágenes que utilizan estos sistemas son universalmente conocidas como ARASAAC¹³ y varían su complejidad dependiendo de la edad cognitiva de cada niño. Por ejemplo, se pueden encontrar aquellas que representan necesidades básicas como ser “sed”, “dolor”, “sueño”, “baño.

¹³ Ibídem

El proceso de comunicación es lento, dado que en la mayoría de los casos los niños necesitan contar con un asistente para poder comunicarse, y éste debe poder interpretar la necesidad que intenta comunicar para proveerle las herramientas necesarias. Es por esto que se considera adecuado integrar funcionalidades de los SAC manuales en un dispositivo móvil mediante una aplicación, otorgando mayor fluidez y menor complejidad en el proceso de comunicación.

Con la incorporación de éstas características en una aplicación se busca fomentar la integración social, la independencia, la alfabetización en el caso de los más pequeños, entre otros beneficios derivados de la comunicación, así como también incorporar al SAC un vocabulario propio acompañado de imágenes representativas de la cultura en particular.

Capítulo 4 - Fase de Exploración



La fase de *exploración* se identifica como “*tipo: patrón de fase*” dentro del método Mobile-D y está compuesta por etapas que tienden a establecer un proyecto incipiente al finalizar la fase, que irá perfeccionándose a lo largo de la aplicación del método. Se presenta la fase con el siguiente resumen:

Metas de la fase de exploración:

- Establecer el grupo de *stakeholders o interesados*, que colaboran en la planificación y en el monitoreo del proyecto de desarrollo del software.
- Definir las metas y el alcance del proyecto de desarrollo del software.
- Elaborar el plan de proyecto inicial, considerando el ambiente, las personas involucradas, y los procesos a desarrollar.

Entrada de la fase de exploración:

- Propuesta de realización de un producto.
- Estándares necesarios para la realización del mismo, si existieran.

Salidas de la fase de exploración:

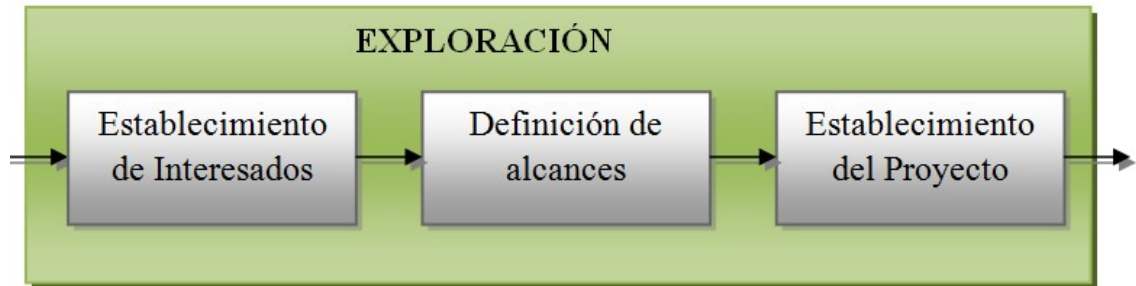
- El documento inicial de requerimientos.
- El plan de proyecto que incluye la especificación de una línea de tiempo, la definición de las partes interesadas y sus responsabilidades, de los recursos necesarios para llevar a cabo el proyecto.
- La descripción del proceso de base, la definición de un proceso de línea de base, actividades de control de calidad, documentación, puntos de integración.
- El plan de entrenamiento.
- La descripción inicial de una línea de arquitectura.

Tareas de la fase que permiten alcanzar las metas y obtener las salidas:

- Identificación de los interesados.
- Identificación de las necesidades de desarrollo del sistema o alcance del proyecto.
- Clasificación de requerimientos.
- Planificación inicial del proyecto.

- Establecimiento del proyecto y de la línea de base.

Resumen gráfico de la fase comprendida como proceso:



Esta es una fase que se podría denominar *transversal*, ya que sus actividades pueden llevarse a cabo en cualquiera de las fases siguientes de Mobile-D y que puede ayudar a controlar la implementación.

4.1. ESTABLECIMIENTO DE LOS INTERESADOS

La etapa de *establecimiento de los interesados* se identifica como “*tipo: patrón de etapa de clasificación esencial*” dentro del método Mobile-D. Podemos presentar la etapa con el siguiente resumen:

Motivación del establecimiento de los interesados:

- Definir aquellos grupos importantes que darán soporte al desarrollo del software. Se requiere la identificación de grupos en los cuales resalte la experiencia en el dominio de aplicación y que, posteriormente, puedan cooperar en planificar una implementación efectiva y controlada del producto software.

Meta del establecimiento de los interesados:

- Identificar el grupo de interesados que acompañará las diferentes etapas y fases.

Entrada a la etapa establecimiento de los interesados:

- Propuesta de realización de un producto.

Salida de la etapa del establecimiento de los interesados:

- Un plan de proyecto en el cual se definen los grupos de interesados y sus tareas específicas, roles y responsabilidades a través del proyecto.

Tareas específicas a realizar en el establecimiento de los interesados:

- Establecer interesados incluye identificar, comprometer y establecer diferentes grupos necesarios en el proceso de desarrollo del software.
- Identificación de los grupos de clientes que tienen un rol especial en el proceso Mobile-D.

Roles a definir en el establecimiento de los interesados:

- Equipo de exploración
- Grupos de interesados.

Presentados estos lineamientos generales que establece Mobile-D, se procede a identificar el *grupo de interesados*. En este proyecto estará representado por el *equipo de terapeutas del IPRI* quien se constituirá como cliente, dado que los niños que son destinatarios últimos del sistema no pueden expresar sus necesidades. Será con los integrantes de este equipo de terapeutas con quienes se mantendrá contacto directo para el asesoramiento, la delimitación del dominio y el tratamiento de cuestiones específicas que debe satisfacer el SAC.

Precisando el *establecimiento del grupo de interesados*, el IPRI cuenta con un área de Terapia Infantil, conformada por un equipo interdisciplinario de terapeutas que trabajan colaborativamente en el tratamiento y la rehabilitación de los niños con PCI que asisten a la Institución. Es importante aclarar que si bien los usuarios finales de la aplicación son los niños que asisten a la institución, no participan en el desarrollo de este trabajo ya que sus identidades e información de contexto están reservadas por ser menores de edad y por las dificultades de expresión que poseen. De esta manera, el grupo inicial de interesados estará conformado por los psicólogos, maestros de educación especial, y psicomotricistas que asisten a estos niños en el IPRI. Estos interesados tienen funciones bien definidas que impactan en el desarrollo del prototipo, como por ejemplo, orientar el desarrollo de la aplicación desde su

experiencia práctica, aportar conocimientos teóricos del dominio, realizar las pruebas de aceptación, y responder las historias de usuario que se propongan.

Por otro lado, la tarea requiere definir un *equipo de exploración* que en este caso está constituido por el desarrollador de este trabajo de grado y especialistas en el área informática con experiencia en el desarrollo de aplicaciones. Estos últimos tienen asignado el rol de asesorar en las cuestiones específicas que pudieran surgir a lo largo del desarrollo del prototipo.

4.2. DEFINICIÓN DE ALCANCES

La etapa *definición de alcances* se identifica como “*tipo: patrón de etapa de clasificación esencial*” dentro del método Mobile-D. Se presenta con el siguiente resumen:

Motivación de la definición de alcances:

- Definir las metas del proyecto incipiente que se está gestando tanto en lo que se refiere al contenido como a la línea de tiempo de mencionado proyecto.

Metas de la definición de alcances:

- Definir el cronograma del proyecto.
- Definir los requerimientos iniciales del proyecto.

Entrada a la definición de alcances:

- Propuesta de realización de un producto.
- Expresión inicial de requerimientos sin detallar.
- Estándares relevantes, si existieran.

Salida de la etapa definición de alcances:

- Un plan de proyecto actualizado.
- Un documento inicial de requerimientos que incluye tanto los requerimientos funcionales como no funcionales del sistema.

Tareas específicas a realizar en la definición de alcances:

- Planificación inicial del proyecto, implica definir una línea de tiempo inicial para el desarrollo del proyecto y el ritmo al que deberían llevarse a cabo las iteraciones.
- Colección inicial de requerimientos definidos a un nivel apropiado para una etapa inicial. Se deben identificar tanto los requerimientos funcionales como los no funcionales. Si bien se intenta estipular requerimientos fijos, en el futuro de la línea de tiempo del desarrollo pueden emerger nuevos requerimientos. Estos requerimientos constituyen la base para la iteración 0.

Roles implicados en la definición de alcances:

- Equipo de exploración
- Grupos de interesados

Para dar comienzo a esta etapa, se detecta un *requerimiento inicial* expresado por el *grupo de interesados*. El mismo plantea favorecer la comunicación diaria entre los profesionales del IPRI y el niño paciente con PCI, y entre el niño paciente con PCI y su entorno más próximo fuera del IPRI, a través de un SAC que funcione en dispositivos móviles. El alcance de esta investigación incluye el desarrollo de un prototipo de herramienta para automatizar parte del proceso comunicativo *emisor-canal-receptor*.

A partir de este requerimiento inicial y del alcance del sistema, se procede a realizar una planificación inicial de proyecto tal como propone Mobile-D.

4.2.1. Planificación Inicial del Proyecto

La planificación inicial del proyecto se identifica como “*tipo: patrón de tarea de clasificación esencial*” dentro del método Mobile-D. Las *metas* de la tarea son: i) configurar una línea de tiempo inicial del proyecto, ii) establecer el ritmo para el desarrollo del producto, y iii) estimar las inversiones necesarias para el proyecto (tanto de esfuerzo humano como financieras).

La línea de tiempo que se configura en esta tarea debe tener inicios y fines flexibles dado que aún no se ha establecido una lista de requerimientos. Por el contrario, la tarea considera el requerimiento inicial y será ajustada posteriormente por los requerimientos refinados. Dicho de otro modo, la línea de tiempo constituye un calendario general estimado para el desarrollo del proyecto.

Realizadas estas aclaraciones, la planificación temporal inicial del proyecto se expresa temporalmente en la figura 4.1.

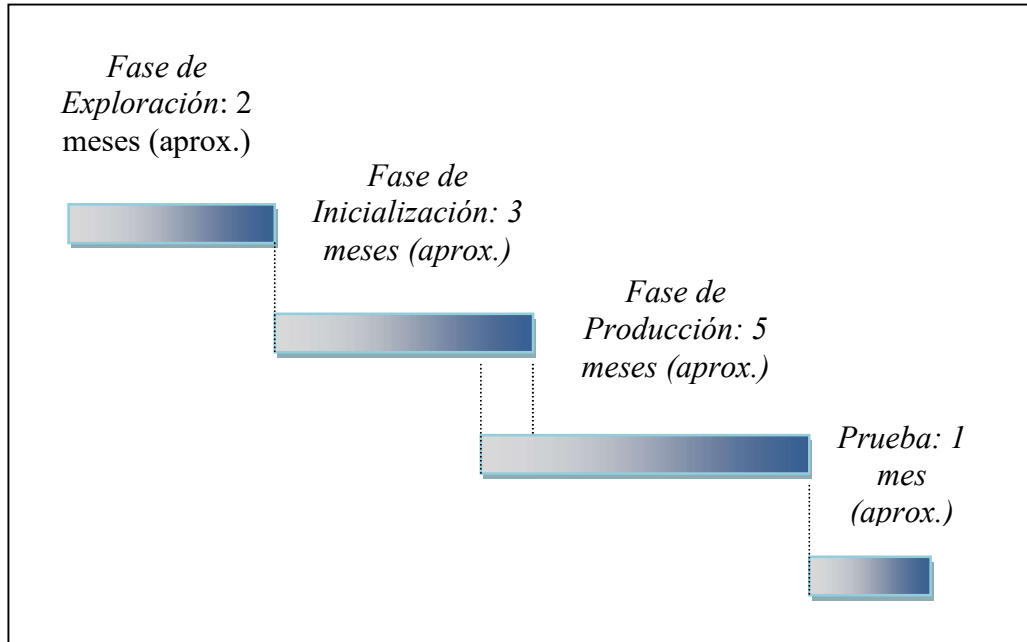


Figura 4-1: Planificación temporal del proyecto de desarrollo del SAC

A continuación, la tabla 4.1 detalla las tareas por cada una de estas fases, y se especifican aquellas que constituyen hitos de control:

	Tarea	Hito
Exploración	Establecimiento de los interesados.	
	Definición del alcance del sistema	
	Definición y clasificación de los requerimientos.	
	Aceptación de necesidades del sistema	Hito de control
Inicialización	Establecimiento de la comunicación con los interesados.	
	Configuración del proyecto.	
	Planificación de la línea arquitectónica.	
	Análisis de requerimientos iniciales.	
	Aceptación de los requerimientos	Hito de control

Tabla 4-1: Tareas por fase e hitos de control

	Tarea	Hito
Inicialización	Selección y análisis de requerimientos iniciales para el diseño de interfaces. Diseño y redacción de historia de usuario.	
	Diseño de interfaces.	
	Ejecución de pruebas y verificación con historia de usuario.	Hito de control
	Interpretación y descripción los resultados de las historias de usuario.	
	Redefinición de requerimientos.	
	Aceptación de los requerimientos	Hito de control
	Determinación de componentes del sistema.	
Producción	Definición del/los componente/s a desarrollar en el día de trabajo y pruebas de aceptación	
	Programación del/los componente/s del prototipo. Pruebas de código	
	Evaluación final de la funcionalidad del prototipo con los terapeutas.	Hito de control
	Redacción de resultados de la evaluación del prototipo.	

Tabla 4.1. *Tareas por fase e hitos de control (Continuación)*

Al ser un proyecto a realizar en el marco de un trabajo final de grado, no se considera necesario establecer el ritmo de las iteraciones ni las inversiones materiales para el desarrollo.

4.2.2. Colección de Requerimientos Iniciales

La tarea de colección de requerimientos iniciales se identifica como “*tipo: patrón de tarea de clasificación esencial*” dentro del método Mobile-D. Las metas de esta tarea son: i) Identificar y acordar el alcance, la funcionalidad básica y los

requerimientos no funcionales más importantes para el producto a desarrollar y ii) documentar los requerimientos en el formato acordado.

En esta tarea se busca producir una definición general inicial del alcance, así como también definir el propósito y funcionalidad del producto para permitir la planificación y el establecimiento del proyecto. Las tareas para alcanzar estas metas son reunir los requerimientos del grupo de interesados y de su ambiente de trabajo diario, y definir las restricciones para el desarrollo del producto. Una vez reunidos los requerimientos iniciales, se comprueba que éstos sean comprendidos de la misma manera por el equipo de exploración y el grupo de interesados, y que no haya conflictos entre los requerimientos. Seguidamente, se acuerda cuáles de los requerimientos identificados son relevantes para el producto, estableciendo el alcance del desarrollo y la cartera de productos inicial.

Para el desarrollo del SAC, se reúnen los requerimientos iniciales de los grupos de interesados a partir de la información obtenida de realizar entrevistas al equipo de terapeutas del IPRI, dichas entrevistas se consigna en el anexo. Estos requerimientos agrupan las funcionalidades del SAC y las limitaciones de hardware y software para el sistema.

De acuerdo con las entrevistas realizadas al grupo de interesados, los pacientes que asisten al área de terapia infantil del IPRI son niños de 2 a 10 años de edad con PCI. A consecuencia de este trastorno, los niños tienen numerosos conflictos en el proceso de comunicación con su entorno social más cercano. Ellos no se comunican mediante el lenguaje hablado, sino haciendo uso de un SAC manual.

Actualmente, los SAC manuales¹⁴ que se usan en el IPRI utilizan imágenes denominadas pictogramas o fotografías para representar acciones u objetos. Los niños con PCI necesitan de una persona que los asista para llevar a cabo el proceso de comunicación dada la condición motriz que presentan. Actualmente, cuando el niño desea expresar una necesidad en su rol de emisor de un mensaje, debe ser asistido por los terapeutas, familiares o docentes para que la comunicación pueda completarse. El rol del asistente en cada terapia es proveer al niño de las imágenes, pizarras o cuadernos, de manera que el niño pueda elegir e indicar lo que desea. El proceso de comunicación es lento, pues el asistente debe interpretar el contexto del

¹⁴ Descriptos en el Capítulo 3

paciente y en base a ello decidir cuáles son las imágenes más adecuadas para proveerle, dejando luego que el niño indique lo que desea comunicar. En todos los casos, la variedad de imágenes depende de la interpretación del asistente y de la disponibilidad de las mismas, pues al ser SAC manuales deben ser transportados a los diferentes lugares de tratamiento dentro del IPRI donde se encuentre el paciente.

Por otra parte, los terapeutas expresaron que para los tutores de los pacientes existen situaciones particulares donde les resulta dificultoso utilizar SAC manuales. En ese momento, los tutores intentan utilizar SAC electrónicos desarrollados en otros países, pero esto les resulta complejo. Esto es así dado que los términos utilizados en la reproducción sonora de los SAC electrónicos no siempre se corresponden directamente con los de nuestro país, porque pueden ser interpretados de diferentes maneras dependiendo de la cultura para la cual fueron desarrollados. Además, estos sistemas no contemplan en sus interfaces adaptaciones a las necesidades motrices de los niños, y en muchos casos llega a entorpecer el proceso de comunicación.

Habiendo comprendido la situación existente dentro del IPRI y en el entorno familiar del niño con PCI, se pretende sustituir el SAC manual en uso por uno que permita la realización automática de la mayoría de las actividades de asistencia. Para ello, se propone el desarrollo de un prototipo de SAC electrónico que funcione en dispositivos móviles. En el diseño de este SAC se considerarán las características propias de los SAC actuales y las restricciones inherentes a los aspectos motrices, cognitivos y fonológicos brindadas por los profesionales del IPRI o grupo de interesados.

Para alcanzar el cumplimiento de esta meta, se realizó en forma conjunta con el grupo de interesados, un listado de requerimientos preliminares del SAC electrónico. El producto de esta tarea se detalla a continuación mediante la descripción de los requerimientos iniciales que servirá como entrada para la fase de inicialización. Se desea un SAC que:

1. Se ejecute en dispositivos móviles. Requerimiento no funcional.
2. Presente en su interfaz íconos o imágenes seleccionables que representen acciones que el usuario desea llevar a cabo. Requerimiento funcional.
3. Contenga íconos, imágenes o fotos pre-obtenidas como parte de la aplicación. Requerimiento no funcional.
4. Muestre imágenes o pictogramas según el género y el rango de edad del

paciente. Requerimiento funcional.

5. Ofrezca imágenes seleccionables al usuario diferenciando si el usuario se encuentra en la casa o la terapia. Requerimiento funcional.
6. Reproduzca en forma sonora una acción (frase o palabra) cuando el usuario seleccione un ícono de la interfaz. Requerimiento funcional.
7. No requiera en ningún caso grandes esfuerzos motores del usuario (movimientos de brazos o de cuerpo entero) para ejecutar sus opciones. Requerimiento no funcional.

Una vez identificados los requerimientos iniciales, se procedió a una primera validación considerando que se haya considerado un nivel de detalle satisfactorio, que no existan conflictos entre ellos, y que sean entendidos sin ambigüedad por el grupo de interesados. Así se obtuvo el listado de requerimientos iniciales que se utilizará en la fase de inicialización posterior.

4.3. ESTABLECIMIENTO DEL PROYECTO

La etapa *establecimiento del proyecto* se identifica como “*tipo: patrón de etapa de clasificación esencial*” dentro del método Mobile-D. Se presenta con el siguiente resumen:

Motivación del establecimiento del proyecto:

- Definir y asignar: los recursos técnicos y humanos necesarios para el proyecto de desarrollo de software incipiente, el proceso de línea de base para el proyecto, de manera que el equipo de desarrollo pueda comenzar el mismo sin retrasos innecesarios producto de la falta de planificación.

Metas del establecimiento del proyecto:

- Definir el tamaño del proyecto.
- Especificar y asignar los recursos humanos necesarios para el desarrollo software. Esta meta no suele considerarse en proyectos de pequeña envergadura.
- Establecer y asignar los recursos técnicos necesarios para el desarrollo software.

- Definir los recursos relacionados con el entorno de trabajo y necesarios para el desarrollo del software.
- Precisar el proceso de línea de base para el proyecto.
- Definir las prácticas, métricas y herramientas para el monitoreo de proyectos.

Entrada al establecimiento del proyecto:

- Documento de requerimientos iniciales.
- Plan del proyecto.
- La propuesta de un producto.

Salida de la etapa de establecimiento del proyecto:

- Plan de proyecto actualizado.
- Descripción de la línea de arquitectura,
- Un plan de entrenamiento, necesario cuando se requiere capacitación en nuevas tecnologías o el aprendizaje de lenguajes para el desarrollo fluido del proyecto.
- Descripción del proceso base.

Tareas específicas a realizar en el establecimiento del proyecto:

- Selección del entorno, implica planear el proyecto en relación con el entorno técnico (incluidos, por ejemplo dispositivos específicos para los cuales se realiza el desarrollo, herramienta de desarrollo, plataformas), como así también se consideran los recursos humanos necesarios.
- Asignación de personal a las tareas específicas del desarrollo.
- Establecimiento de procesos tales como la identificación y la definición del entrenamiento del equipo de desarrollo en aspectos técnicos; y el esbozo de un proceso de base a partir del cual realizar los monitoreos del desarrollo del proyecto incipiente.

Roles implicados en el establecimiento del proyecto:

- Equipo de exploración

- Grupos de interesados

Para dar comienzo a esta etapa, se puede planear el proyecto en relación con el entorno técnico disponible, considerar las herramientas de desarrollo y la plataforma necesarias para que el proyecto pueda realizarse de manera fluida sin retrasos por imprevistos.

4.3.1. Selección del Entorno – Línea de Arquitectura

Esta tarea se identifica como “*tipo: patrón de tarea de clasificación recomendada*” dentro del método Mobile-D. La meta de esta tarea es definir una línea de arquitectura para el proyecto. Tiende a explorar las cuestiones inherentes a la arquitectura y las fuentes potenciales de riesgos antes de comenzar la producción de la aplicación, por ejemplo, el contexto del sistema, las plataformas de software, los *drivers* de la arquitectura, la integración de la arquitectura del software con el proceso de desarrollo, entre otros.

Para definir esta línea de arquitectura es necesario analizar los requerimientos iniciales junto con la propuesta, identificando las características principales del producto y los objetivos de éste. Se define el contexto del sistema y se analizan las interfaces del sistema con sus entidades externas, identificando también el tipo de arquitectura del producto definiendo la estructura y el contenido de la misma. Se define la plataforma necesaria para el desarrollo del proyecto.

Tomando como base los requerimientos iniciales de este proyecto, se propone el desarrollo de un prototipo evolutivo para una aplicación móvil. Tras un análisis de las tendencias de las tecnologías móviles más usadas en nuestro país que puede observarse en la Figura 4.2, se decide desarrollar el SAC para dispositivos que posean Sistema Operativo Android.

Se pueden precisar los siguientes detalles inherentes a la arquitectura de desarrollo:

- *Tipo de aplicación*: se descarta la posibilidad de desarrollar una aplicación web o una híbrida dado que los usuarios finales no siempre van a tener acceso a navegadores web, según consta en las entrevistas. De este modo se opta por desarrollar una aplicación móvil nativa.
- *Plataforma de desarrollo*: para programar el SAC se utilizará Android Studio, que permite la programación de aplicaciones nativas, siendo éste un entorno

de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones en el sistema operativo Android, basado en IntelliJ IDEA.

- *La arquitectura del producto:* es de tipo monolítica, se describe en un solo bloque de tipo Cliente (Ruiz Najar, 2014), es decir, toda la aplicación reside en el dispositivo y utiliza los recursos del mismo para el funcionamiento del sistema.

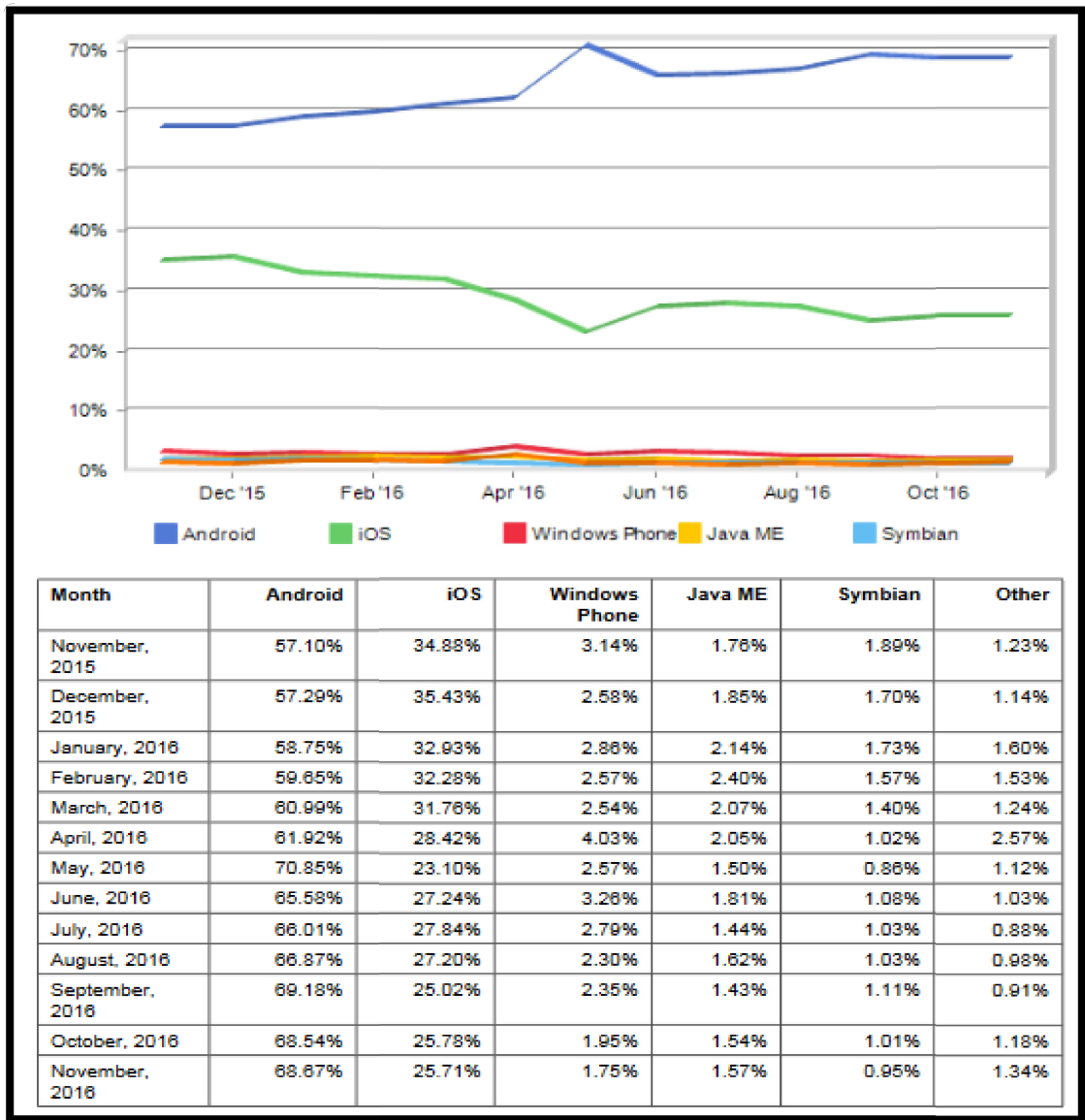


Figura 4-2: Tendencias en los últimos años de tecnologías móviles¹⁵

Dentro de la arquitectura intervienen e interactúan tres componentes básicos: interfaz de usuario, lógica de negocios y lógica de datos. La lógica de negocios

¹⁵ <http://netmarketshare.com/downloads/guest636272593499220000.pdf> [consultado ultima vez 11 de Septiembre 2016]

permite la comunicación entre los datos y la interfaz del usuario. En la figura 4.3 se representa esta comunicación entre componentes. El usuario accede al funcionamiento de la aplicación mediante una interfaz de usuario. La selección de la interfaz visible para el usuario se realiza con una lógica de procesamiento de la información del contexto y del tipo de usuario. Por ejemplo, el género y la fecha de nacimiento. La información de cada tipo de usuario se encuentra en la lógica de datos, ya que ella es la encargada del manejo y control de los datos almacenados, por lo que la interfaz interactúa con la lógica de negocio para acceder a ella. En la interacción entre la interfaz y la lógica de negocios se ejecuta el procesamiento de datos y la gestión de los recursos del dispositivo.

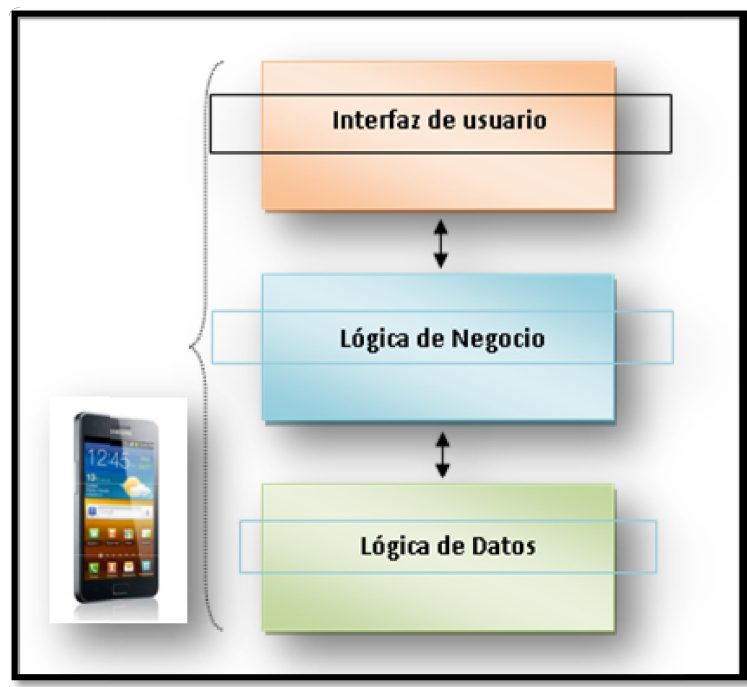


Figura 4-3: *Arquitectura cliente monolítica*

Cabe destacar que la arquitectura descrita sirve como lineamiento general para el desarrollo, y será refinada en la siguiente fase de aplicación del método que guía esta investigación.

4.3.2. Establecimiento del Proceso

El establecimiento del proceso se identifica como “*tipo: patrón de tarea de clasificación esencial*” dentro del método Mobile-D. Las metas de esta tarea son: i) adaptar el proceso de línea de base, ii) acordar la documentación necesaria del

proyecto, iii) definir las prácticas, métricas y herramientas para el monitoreo del proyecto.

La línea base para el desarrollo del SAC está constituida por el documento de requerimientos inicial aprobado por el grupo de interesados, y que se completará con mayor detalle en la fase de inicialización. En forma general se controlará que dicho documento guarde consistencia a lo largo del desarrollo, pues todo el proceso será acompañado por el grupo de interesados. No se dispone en esta instancia de algún componente de software o documentación que pueda complementar el documento de requerimientos.

El método Mobile-D provee patrones de apoyo para la organización y ejecución del proyecto. Éstos proporcionan una guía de documentación y permiten clasificar cada actividad como: esencial, recomendado, o de apoyo. Si bien se trata de un desarrollo ágil, se sabe que al reducirse el producto software a un prototipo de SAC, la documentación es esencial para proponer futuros incrementos de funcionalidades. Así, todo el documento final producido en esta investigación quedará a disposición del IPRI, para evaluar futuras extensiones. La documentación se irá efectuando a medida que se avance en el desarrollo, es decir, será de producción iterativa y se basará en los patrones de documentación mencionados.

La descripción presentada al comienzo de cada una de las fases y etapas se realiza teniendo en cuenta los patrones de documentación, adaptados a las necesidades específicas del desarrollo del SAC. Para el desarrollo del proyecto se utilizan además las plantillas propuestas por Mobile-D que se consideren apropiadas.

En lo que respecta a la definición de prácticas, métricas y herramientas para monitorear el proyecto de desarrollo del SAC, el mismo es de pequeña envergadura. Sin embargo, dada las características motrices de los pacientes con PCI el diseño del prototipo será constantemente monitoreado por el grupo de interesados o terapeutas del IPRI.

En esta instancia se realiza también una descripción de la documentación a producir durante el desarrollo del proyecto, la cual se sintetiza en la tabla 4.2.

	Etapa	Tarea	Documentación	Puntos de revisión
Exploración		Planificación del proyecto	Plan de proyecto	--
	Definición del alcance	Definición y clasificación de requerimientos	Documento de requerimientos iniciales	Aceptación de necesidades del sistema y de los requerimientos
Inicialización	Configuración del proyecto	Configuración del entorno Entrenamiento Establecimiento de la comunicación Configuración del proyecto		--
	Día de planificación: iteración 0	Análisis de necesidades iniciales Planificación de la línea arquitectónica Planificación de la Iteración	Requerimientos iniciales priorizados por el usuario Historia de usuario Prototipos funcionales	Ejecución de pruebas con historia de usuario
	Día de prueba: iteración 0	Implementar otras tareas	Pila de productos	--
Producción	Día de planificación	Taller Post iteración Análisis de requerimientos Planificación de la iteración Generación de pruebas de aceptación Revisión de las pruebas de aceptación	Definición de interfaces y Navegación de cada iteración Tareas	Verificación de lista de productos a implementar en cada iteración
	Día de trabajo	Recapitulaciones Desarrollo guiado por pruebas Programación en parejas Integración continua Refactorización Informes al cliente	Código de desarrollo de componentes funcionales	Pruebas de código
	Día de lanzamiento	Integración del sistema Pruebas previas a la liberación Pruebas de aceptación Ceremonia de lanzamientos	Lista de defectos o errores encontrados	Evaluación final de la funcionalidad del prototipo

Tabla 4-2: Resumen de la documentación prevista - proyecto de desarrollo

En la tabla de resumen sólo se consignan los documentos formales, pero se puede considerar que la documentación para el proyecto general está realizada en lenguaje coloquial. Es decir, la documentación se atiende de forma especial por tratarse de un trabajo final de grado.

4.4. VERIFICACIÓN DEL PLAN DE PROYECTO

Una vez finalizado el establecimiento del proyecto se procede a verificación del Plan del Proyecto utilizando la *lista de verificación* propuesta por Mobile-D. La misma se expresa en la Tabla 4.3 y se completa en conjunto con el grupo de interesados. Cada una de las actividades puede haberse cumplido o no, o no ser considerada porque no corresponde según la envergadura del proyecto.

LISTA DE VERIFICACIÓN DEL PLAN DEL PROYECTO		Fecha: 27/06/2017		
FASE DE EXPLORACIÓN				
Requerimientos iniciales	Si	No	N/C	
1. Todos los requerimientos funcionales iniciales se han incluido en el plan de proyecto	X			
2. Todos los requerimientos no funcionales se han incluido en el plan de proyecto	X			
Programación y ritmo	Si	No	N/C	
1. La programación general se ha incluido en el plan del proyecto	X			
2. El ritmo previsto (número y la duración de cada fase y sus iteraciones) han sido definido en el plan del proyecto	X			
Recursos	Si	No	N/C	
1. El plan del proyecto se ha actualizado con los grupos de interés identificados y sus miembros	X			
2. En el plan del proyecto se ha actualizado la información relativa a las herramientas de desarrollo de software.	X			

Tabla 4-3: *Lista de verificación de la fase de exploración*

LISTA DE VERIFICACIÓN DEL PLAN DEL PROYECTO		Fecha: 27/06/2017		
FASE DE EXPLORACIÓN				
Documentación	Si	No	N/C	
1. Los documentos que se producirán en el proyecto han sido incluidos en el plan del proyecto	X			
2. La vida útil de cada documento se ha incluido en el plan del proyecto (por ejemplo, los contenidos , actualización)			X	
Seguro de calidad	Si	No	N/C	
Los procedimientos de garantía de calidad se han definido en el plan de proyecto para cada producto (documentación , código y producto) incluyendo los actores y el cronograma			X	

Tabla 4.3. *Lista de verificación de la fase de exploración (Continuación)*

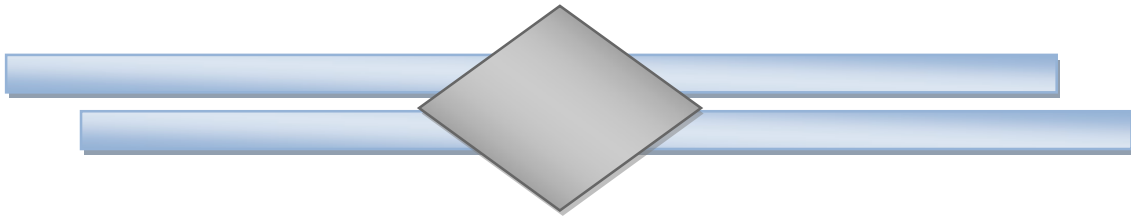
4.5. CONCLUSIÓN

En este capítulo se ha desarrollado la *Fase de Exploración* propuesta por el método Mobile-D, obteniendo como salida principal el documento de requerimientos iniciales y el plan de proyecto para el desarrollo del SAC, ambos considerando un nivel incipiente, pero que constituirá la base para la siguiente fase de inicialización. Asimismo, se comenzaron a especificar cuestiones inherentes a la arquitectura del software, que será profundizada en el resto del trabajo.

Es importante iniciar el desarrollo del proyecto identificando el grupo de interesados, los alcances del sistema, y las condiciones iniciales del entorno de trabajo, lo cual ha sido detallado en el presente capítulo.

Por último, se ha aplicado la lista de verificación del plan de proyectos a fin de realizar un control exhaustivo sobre el desarrollo de la fase, tal como lo propone el método.

Capítulo 5 - Fase de Inicialización



La fase de *inicialización* se identifica como “*tipo: patrón de fase de clasificación esencial*” dentro del método Mobile-D y está compuesta por etapas que diseñadas para permitir el éxito de las próximas fases del proyecto preparando y verificando los problemas críticos de desarrollo de los requerimientos seleccionados por los interesados. A continuación se resume esta fase según la propuesta general de Mobile-D:

Metas de la fase de inicialización:

- Obtener una buena comprensión general del producto basado en los requerimientos iniciales y en las descripciones de la línea de arquitectura.
- Preparar los recursos físicos y técnicos, así como también la comunicación con los interesados, los planes de proyectos y todos los aspectos críticos del desarrollo, de modo que se esté en condiciones de implementar los requerimientos seleccionados por los interesados durante las próximas fases del proyecto.

Entrada de la fase de iniciación:

- El documento de requerimientos iniciales.
- El plan del proyecto.
- La descripción del proceso base.
- El plan de medición.
- El plan de entrenamiento.
- La descripción de la línea de arquitectura.
- La cartera de productos.

Salidas de la fase de iniciación:

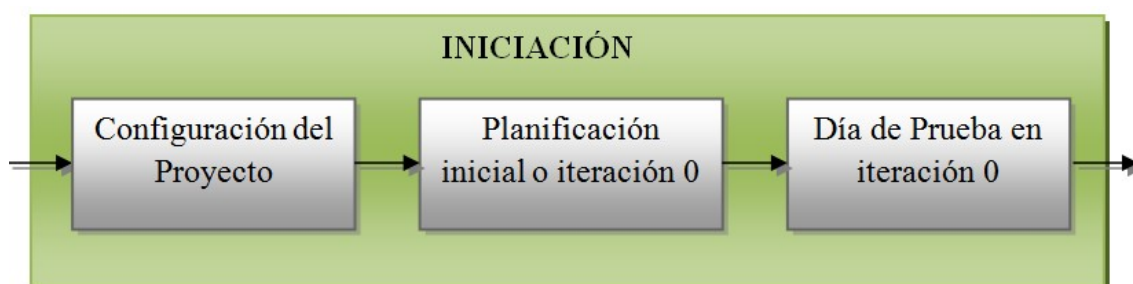
- El plan de proyecto actualizado y plan de línea de arquitectura.
- La primera versión de la arquitectura del software y el documento de descripción de diseño.
- La funcionalidad implementada.
- La primera versión de la cartera de productos.
- El documento de requerimientos iniciales actualizado.

- Las notas del desarrollador y la interfaz de usuario - ilustraciones de cada requerimiento tratado.
- Las pruebas de aceptación para cada requerimiento.

Tareas de la fase que permiten alcanzar las metas y obtener las salidas:

- Establecimiento del proyecto.
- Planificación inicial iteración 0.
- Día de Prueba en Iteración 0.

Resumen gráfico de la fase de inicialización comprendida como un proceso:



A continuación se describen cada una de las etapas, haciendo una selección y adaptación de aquellas tareas que son necesarias en un proyecto de pequeña o mediana envergadura como lo es el desarrollo de este SAC.

5.1 CONFIGURACIÓN DEL PROYECTO

La etapa de configuración del proyecto se identifica como “tipo: patrón de etapa de clasificación esencial” dentro del método Mobile-D. Se puede presentar la etapa con el siguiente resumen:

Motivación de la configuración del proyecto:

- Establecer los recursos físicos y técnicos para el proyecto, así como el ambiente para el monitoreo del mismo, capacitar al equipo del proyecto según sea necesario, y establecer las maneras específicas de comunicarse con el grupo de interesados. Todas las tareas de la configuración incluyen la participación del equipo del proyecto.
- Metas de la configuración del proyecto:
- Establecer los recursos físicos para el equipo del proyecto.

- Establecer los recursos técnicos para el proyecto.
- Lograr un acuerdo con el grupo de interesados sobre la forma en que se comunicarán a lo largo del proyecto.

Entrada a la etapa de configuración del proyecto:

- El plan de proyecto.
- El plan de entrenamiento.
- La descripción del proceso base.
- El plan de medición (si existiera).

Salida de la etapa de configuración del proyecto:

- Plan de proyecto.

Tareas específicas en la configuración del proyecto:

- Configuración del entorno técnico y físico para el proyecto.
- Entrenamiento.
- El establecimiento de la comunicación con los interesados durante todo el proyecto.

Roles a definir en la configuración del proyecto:

- Equipo de desarrollo
- Grupos de interesados.

5.1.1. Configuración del Entorno

La configuración del entorno es una tarea para establecer el entorno técnico y físico del proyecto. Es importante para los desarrolladores de software, ya que puede delimitar una necesidad de formación en lo que respecta al entorno técnico de desarrollo.

En forma sintética y gráfica, la tarea se expresa en la figura 5.1.

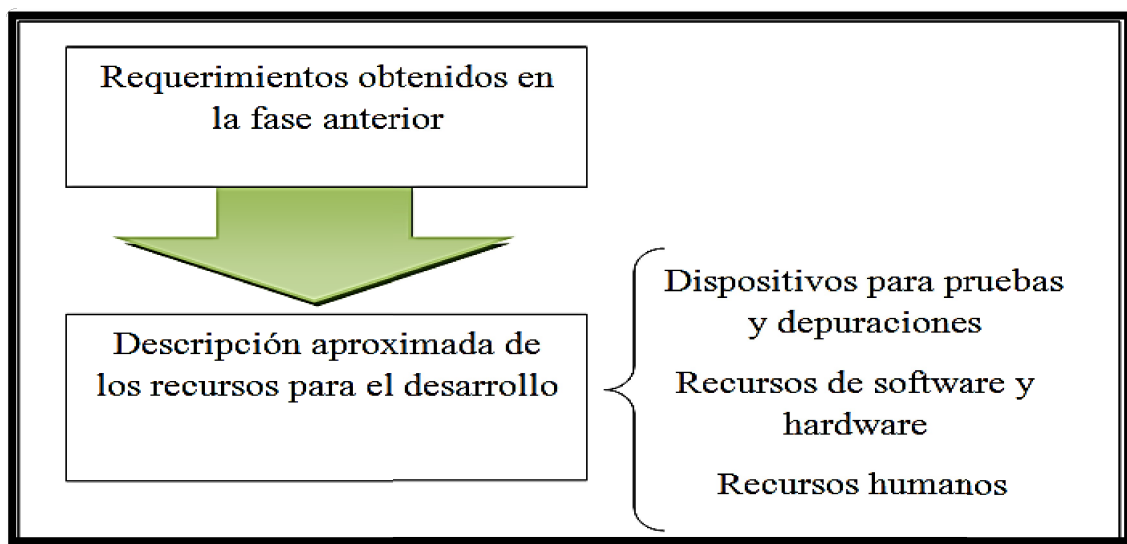


Figura 5-1: Tarea de configuración del entorno

Utilizando como base los requerimientos obtenidos en la fase anterior, se realiza una estimación de los recursos técnicos y físicos que serán necesarios para el desarrollo, pruebas y depuraciones del sistema. Dicha estimación se presenta en la tabla 5.1.

	Recurso	Características
Hardware	Estación de trabajo	Sistema operativo: Windows 10 Pantalla: 14.1” Disco: 1TB S-ATA RAM:4GB DDR3 1600MHZ Procesador: Intel Core I5 2,7GHz 3M Cache
	Dispositivo Android	Tableta: Samsung Galaxi Tab 3 Pantalla: TFT a 1280 x 800 pixels (10.1”) RAM: 1GB Almacenamiento interno: 16GB Procesador: un procesador Intel -Atom dual-core de 1.6GHz. Sistema operativo: Android v4.2 Jelly Bean Batería: Standard, Li-Po 6800 mAh

Tabla 5-1: Recursos necesarios para el desarrollo del SAC

	Recurso	Características
Software	Sistema de creación de pantallas interactivas	Justinmind Prototyper ¹⁶
	Plataforma de desarrollo	Android Studio versión 1.5.1 ¹⁷ Requerimientos generales para instalación en el SO Microsoft Windows: <ul style="list-style-type: none"> • Versiones:10/8.1/8/7/Vista/2003/XP (32 o 64 bit). • RAM: 2 GB mínimo, 4 GB recomendado. • Espacio en disco: 500 MB. • Espacio para SDK de Android: 1 GB mínimo para Android SDK, imágenes del sistema de emulador y cachés. • Versión del kit de desarrollo de java:7 o superior • Resolución de pantalla: mínimo de 1280x800
	Almacenamiento	Almacenamiento de datos primitivos privados en pares clave-valor.
Humanos	Orientación Especifica	Especialistas en desarrollo de aplicaciones móviles

Tabla 5.1. Recursos necesarios para el desarrollo del SAC (Continuación)

Como se explicita en la tabla 5.1, se decide utilizar Android Studio para el desarrollo del SAC. Esto requiere planificar un entrenamiento para el equipo de desarrollo en este caso en particular, lo cual se especifica a continuación.

¹⁶ <https://www.justinmind.com/> [consultado ultima vez 5 Octubre 2016]

¹⁷ <http://tools.android.com/download/studio/builds/1-5-1> [consultado ultima vez 15 Agosto 2016]

5.1.2. Entrenamiento

Esta tarea es muy importante dado que en ella se planifica cómo entrenar al equipo de desarrollo de software según sus necesidades específicas. Esto puede incluir capacitación en cuestiones de proceso de desarrollo de software, recolección de datos para el monitoreo de proyectos, o cuestiones técnicas relacionadas con el desarrollo de software en sí mismo, tales como nuevas herramientas y métodos.

En el caso de este proyecto para desarrollar un SAC, la tarea de entrenamiento implicó varias partes. Por un lado fue necesario realizar un estudio profundo sobre el método Mobile-D, sus características, patrones y lineamientos. Esto permitió decidir qué etapas dentro de cada fase sería posible aplicar y cómo hacerlo. Por otra parte, fue necesario revisar métodos de captura de requerimientos, pues el proyecto posee características difíciles de visualizar con técnicas comunes. Esto conllevó a dedicar un tiempo adicional a la preparación de reuniones con los asesores de desarrollo, surgiendo la necesidad de adquirir un software de generación de prototipos funcionales como lo es *Justinmind Prototyper*, de manera de obtener un prototipo que permita simular el funcionamiento real del SAC en cuanto a la organización de las interfaces y los esfuerzos de desplazamiento motriz necesario para navegar a través del SAC. En el caso de la aplicación *Justinmind Prototyper*, el plan de entrenamiento estuvo basado en el seguimiento y ejecución de tutoriales de uso provistos por el mismo sistema.

Además, fue necesaria una capacitación en desarrollo para el SO Android, específicamente en el lenguaje a utilizar para la programación del prototipo Android Studio. Al carecer de experiencia previa en desarrollos con mencionado SO se tomaron cursos básicos de iniciación, complementados con el uso de literatura específica para el desarrollo on-line¹⁸ sobre el desarrollo de aplicaciones móviles. El entrenamiento estuvo apoyado en el aprendizaje autónomo en gran medida, guiado por tutoriales disponibles en la web, permitiendo ahondar sobre la aplicación de funciones específicas a este desarrollo, como por ejemplo manejo de ventanas de dialogo, *widget* y sus propiedades, manejo de *activity*, comunicación entre ellas, entre otras. El entrenamiento sobre Android fue orientado por el asesor el desarrollo de la investigación.

¹⁸ <https://developer.android.com/index.html> [consultado por última vez 17 de febrero 2017]

5.1.3. Establecimiento de la comunicación

El establecimiento de la comunicación se identifica como “*tipo: patrón de tarea de clasificación esencial*” dentro del método Mobile-D. La meta de la tarea es acordar formas de comunicación entre el grupo de interesados y el equipo de desarrollo para discutir sobre las necesidades inminentes del proyecto.

Los medios de comunicación que se emplearon en esta instancia entre el equipo de desarrollo y el grupo de interesados fueron: reuniones personales, reuniones por *skype*, y llamadas telefónicas. Los dos primeros se utilizaron para establecer los requerimientos iniciales, y serán de utilidad posteriormente para refinar los mismos o cuando surjan dudas que requieran un esfuerzo de entendimiento mayor, mientras que las llamadas telefónicas se utilizaron en la fase de exploración para evacuar dudas.

A diferencia de la fase anterior donde la comunicación entre el grupo de interesados y el equipo de desarrollo fue frecuente, en esta fase de inicialización se prevé esta frecuencia incrementada, pues es en ella donde se clarifican las necesidades reales del desarrollo. En lo que respecta a la fase de producción, se estima que la comunicación será un poco menos asidua, dado que quedarían por evaluar cuestiones técnicas referidas al funcionamiento del sistema y las pruebas de funcionamiento del desarrollo. La frecuencia de comunicación estimada para las fases de inicialización y producción están basadas en el hecho de que, para clarificar las funcionalidades específicas de desarrollo, se requerirá el uso de historias de usuario, entrevistas, encuestas y pruebas de prototipo. A consecuencia, será preciso interactuar con el grupo de interesados por ser ellos los responsables de guiar y delinear cuestiones específicas que el sistema debería satisfacer.

Los acuerdos inherentes a la comunicación que se hicieron entre ambos grupos, se resumen a continuación:

- Las reuniones personales se realizarían en el IPRI, en el horario de jornada laboral de los miembros del grupo de interesados, por cobertura de seguro laboral.
- Para coordinar el horario y fecha en el que se realizarán las interacciones sucesivas, se utilizarán los números telefónicos personales y las direcciones de correo electrónico.

- Se planifica realizar varias reuniones de monitoreo del desarrollo con el grupo de interesados, que se irán ejecutando a medida que sean necesarias.

Con estos elementos en juego, el equipo de desarrollo tiene la certeza que puede acudir al grupo de interesados tanto para evacuar dudas como para monitorear, probar, y readecuar el funcionamiento del SAC para niños con PCI que se está desarrollando.

5.2 DÍA DE PLANIFICACIÓN INICIAL O ITERACIÓN 0

La etapa *día de planificación inicial* se identifica como “*tipo: patrón de etapa de clasificación esencial*” dentro del método Mobile-D. Se puede presentar la misma con el siguiente resumen:

Motivación de la definición del día de planificación inicial:

- Obtener una comprensión general profunda del producto que se va a desarrollar, así como también preparar y refinar los planes para las próximas fases del proyecto. Preparar planes para verificar y resolver todos los problemas críticos de desarrollo hacia el final de esta fase.

Metas de la definición del día de planificación inicial:

- Obtener una comprensión general del producto a ser desarrollado.
- Preparar planes de proyectos y planes de arquitectura para las próximas fases del proyecto.
- Preparar planes para revisar y solucionar todos los problemas críticos de desarrollo.

Entrada a la definición del día de planificación inicial:

- El documento de requerimiento iniciales (funcional y no funcional).
- La descripción de la línea de arquitectura.
- El plan inicial del proyecto.

Salida de la etapa definición del día de planificación inicial:

- El documento de requerimientos iniciales actualizado.
- El plan actualizado del proyecto.
- El plan de línea de la arquitectura.
- La cartera de productos con los requerimientos iniciales del software.
- Las notas del desarrollador e interfaz de usuario - ilustraciones de cada requerimiento tratado.
- Las pruebas de aceptación para cada requerimiento.
- La primera versión del documento de descripción del diseño y de la arquitectura de software.

Tareas específicas a realizar en la definición del día de planificación inicial:

- Análisis de necesidades iniciales, implica priorizar y analizar cuidadosamente los requerimientos seleccionados para la implementación durante la fase de inicialización.
- Planificación de líneas de arquitectura, requiere preparar las cuestiones arquitectónicas críticas.
- Planificación de la iteración, en ella se genera la programación y el contenido para que la iteración se ejecute. El contenido se define en términos de:
 - ✓ Historias de usuario y tareas para producir código mediante la implementación de requerimientos funcionales seleccionados en el análisis de necesidades iniciales.
 - ✓ Tareas para comprobar y resolver problemas críticos de desarrollo, sin producir ningún código de trabajo.
- Generación de prueba de aceptación, implica la verificación de los requerimientos que se han establecido. Éstas son generadas para historias y tareas que producen código.
- Revisión de pruebas de aceptación para mejorar la calidad.

Roles implicados en la del día de planificación inicial:

- Equipo de desarrollo
- Equipo de arquitectura (si existiera)
- Grupos de interesados

5.2.1. Análisis de Necesidades Iniciales

El análisis de necesidades iniciales se identifica como “tipo: patrón de tarea de clasificación esencial” dentro del método Mobile-D. Las metas de esta tarea son: i) identificar un conjunto de requerimientos que obligan a crear los componentes e interfaces más importantes del sistema, ii) garantizar que estos requerimientos aporten un importante valor comercial.

Para completar esta tarea se selecciona un conjunto de requerimientos básicos simples con los que se crean los componentes más importantes y las interfaces del sistema. Luego se analizan estos requerimientos en forma individual, usando ilustraciones de la interfaz de usuario como material de soporte gráfico. Esto se realiza para obtener una comprensión completa de la funcionalidad requerida por los desarrolladores. El objetivo de esta tarea es profundizar la comprensión de las funciones necesarias para el sistema.

Siguiendo estos lineamientos, el grupo de interesados procede a priorizar los requerimientos iniciales que ellos consideran que deben ser satisfechos en la primera instancia del desarrollo. Con ésta actividad se intenta profundizar sobre cuáles son las necesidades comunicativas de la aplicación y cuál sería la navegación más adecuada, así como también se pretende explorar la existencia de requerimientos que hasta el momento no hayan sido visualizados y que puedan estar encapsulados en los descritos como iniciales, clarificando las necesidades del sistema a desarrollar.

Junto a los profesionales del IPRI se realiza un primer análisis de los requerimientos iniciales planteados, definiendo en base a ellos las necesidades principales de comunicación que tienen los niños que asisten al Instituto. Estos requerimientos iniciales priorizados se expresan en la tabla 5.2.

Nº	Se requiere un SAC que:	Prioridad
1	Se ejecute en dispositivos móviles. Requerimiento no funcional.	1
2	Presente en su interfaz íconos o imágenes seleccionables que representen acciones que el usuario desea llevar a cabo. Requerimiento funcional.	4
3	Contenga íconos, imágenes o fotos pre-obtenidas como parte de la aplicación. Requerimiento no funcional	7
4	Muestre imágenes o pictogramas según el género y el rango de edad del paciente. Requerimiento funcional.	5
5	Ofrezca imágenes seleccionables al usuario diferenciando si el usuario se encuentra en la casa o la terapia. Requerimiento funcional.	6
6	Reproduzca en forma sonora una acción (frase o palabra) cuando el usuario seleccione un ícono de la interfaz. Requerimiento funcional.	2
7	No requiera en ningún caso grandes esfuerzos motrices del usuario (movimientos de brazos o de cuerpo entero) para ejecutar sus opciones. Requerimiento no funcional.	3

Tabla 5-2: *Requerimientos priorizados por el grupo de interesados*

Los requerimientos más importantes son valorados en una escala del 1 al 7, donde el 1 representa el requerimiento de prioridad más alta y el 7 el requerimiento que se satisfará al final del desarrollo.

5.2.2. Planificación de Líneas de Arquitectura

La planificación de líneas de arquitectura se identifica como “*tipo: patrón de tarea de clasificación recomendada*” dentro del método Mobile-D, no es obligatoria. La meta de esta tarea es crear un plan de línea de arquitectura.

En esta tarea se define el contexto del sistema, se incluye la definición en el documento de descripción de diseño, así como también la arquitectura de software. Además, se describen las herramientas para la arquitectura y se diseña una arquitectura inicial para el software que se va a desarrollar.

En la fase de exploración se realizó una descripción general de la aplicación desde el punto de vista arquitectónico, identificando a la aplicación como: nativa, a desarrollarse para el sistema operativo Android en Android Studio, y una arquitectura monolítica (Najar Ruiz, 2014)

Considerando esta descripción y los requerimientos planteados se determina la necesidad de utilizar *perfiles de usuario*. Por ello se hace necesario almacenar datos de los diferentes usuarios de la aplicación y sus configuraciones de sistema. A

partir de estos datos se permite que varios usuarios de manera alternativa puedan iniciar sesión en un mismo dispositivo, pero sólo un usuario en un momento dado. Así el archivo general de configuración de sistema se entiende como una plantilla general, que se particulariza para un usuario con características específicas en un momento dado. En este caso se genera un archivo de configuración particularizado a partir del general, que contiene todas las configuraciones que le corresponden al usuario según lo delimita su perfil. La figura 5.2 aclara la configuración de archivos para N usuarios diferentes.

En cuanto al desarrollo de la aplicación propiamente dicha, es importante mencionar que la misma estará compuesta por el conjunto de interfaces de usuario, una lógica de navegación entre ellas, y las configuraciones de usuario particulares.

Para satisfacer las necesidades del sistema se requiere detallar la composición de las pantallas para mostrar al grupo de interesados que serán los evaluadores del contenido de imágenes y texto correspondiente a cada una.

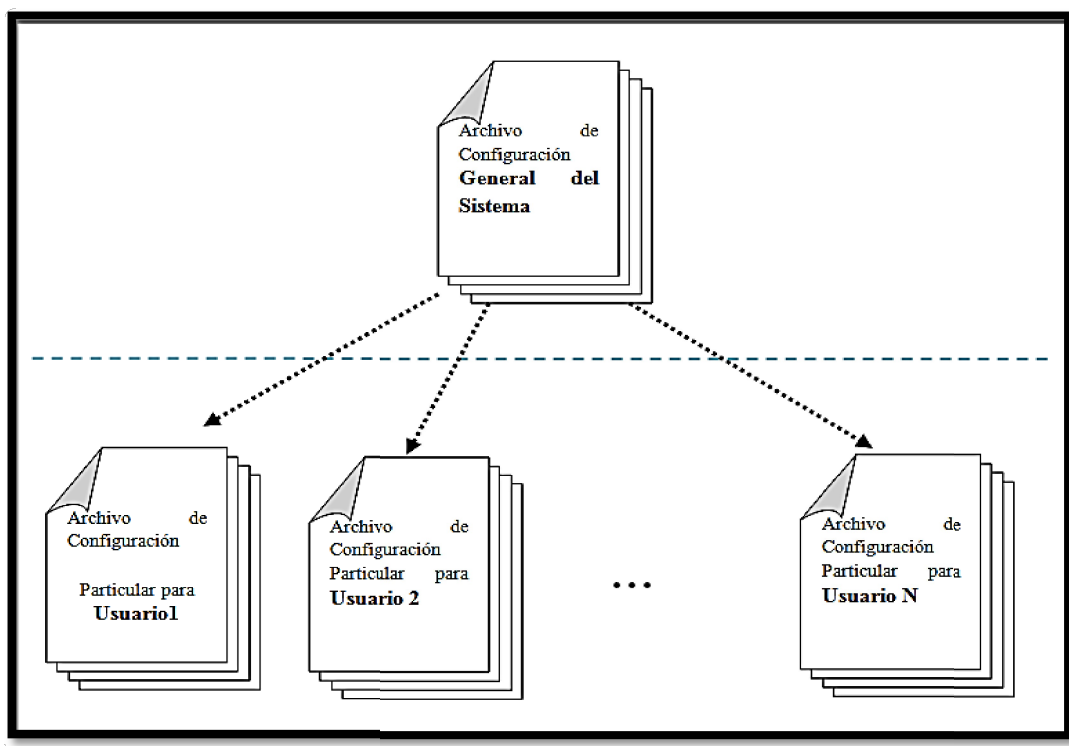


Figura 5-2: Representación gráfica de la configuración de usuarios

En lo que respecta a la navegación, se debe definir la lógica de sintaxis o conexión de las diferentes pantallas según su contenido. Por último, para la configuración del usuario se hace necesario evaluar el género y la edad del mismo

cuando inicia la aplicación, lo que llevará a redefinir las conexiones entre las pantallas.

5.2.3. Planificación de la Iteración 0

Para planificar esta experiencia de desarrollo y la posterior evaluación, se elabora una ficha o historia de usuario cuyo formato general se muestra en la tabla 5.3, y que deberá ser completada por el grupo de interesados. De manera de obtener una colaboración concreta de los terapeutas en el llenado de las historias de usuario, el equipo de desarrollo mantuvo reuniones con el grupo de interesados en las que el proceso de comunicación tendió a explicar la funcionalidad principal de este documento. Este detalle no es menor, puesto que se trata de un trabajo conjunto interdisciplinario en el que se debe diferenciar tanto el rol del paciente como el del terapeuta. Cada historia de usuario permitirá desarrollar las funciones de desplazamiento por el prototipo y - al mismo tiempo - documentar los requerimientos. En las Tablas 5.3 a 5.7 se presentan las fichas obtenidas como resultado de esta actividad.

Numero/ ID	Tipo	Dificultad		Esfuerzo		Prioridad	Notas
		Antes	Después	Estimado	Gastado		
{ N° de historia}	Nueva	Fácil	Fácil			Alta	{nombre de la historia de usuario}
	Arreglada	Moderado	Moderado			Media	{a quien va dirigida}
	Extendida	Difícil	Difícil			Baja	{requerimiento/s al /a los que se vincula}
Descripción							
“Como [rol] quiero [funcionalidad] para [beneficio]”							
Fecha	Estado		Comentario				
	Definido						
	Implementando						
	Hecho						
	Verificado						
	Pospuesto/Cancelado/ Combinada						

Tabla 5-3: Formato general de la ficha de historia de usuario

Numero/ ID	Tipo	Dificultad		Esfuerzo		Prioridad	Notas
		Antes	Después	Estimado	Gastado		
1	Nueva Arreglada Extendida	Fácil Moderado Difícil	Fácil Moderado Difícil	--	--	Alta Media Baja	Agrupamiento de palabras Niño con PCI (usuario final) #Requerimientos iniciales vinculados: N°2, N°6 y N°7
Descripción							
<p><u>Funcionalidad deseada:</u> Como terapeutas consideramos que las palabras disponibles en el SAC deben estar ordenadas según un esquema lógico de selección, es decir agrupadas por categorías o tipos de acción, para reducir los desplazamientos necesarios al momento de comunicarse y que se facilite encontrar lo que quiere expresar.</p>							
Fecha	Estado		Comentario				
07/16	Definido		Fase de inicialización Esfuerzo estimado para satisfacer ésta HU: 3 días				

Tabla 5-4: *Historia de usuario 1 “agrupamiento de palabras”*

Numero/ ID	Tipo	Dificultad		Esfuerzo		Prioridad	Notas
		Antes	Después	Estimado	Gastado		
2	Nueva Arreglada Extendida	Fácil Moderado Difícil	Fácil Moderado Difícil	--	--	Alta Media Baja	Navegación entre pantallas del prototipo Paciente Niño con PCI (usuario final) #Requerimientos iniciales vinculados: N°5 y N°7
Descripción							
<p><u>Funcionalidad deseada:</u> Como terapeutas necesitamos que el sistema no necesite en ningún caso grandes esfuerzos motrices del usuario para ejecutar sus opciones, para mejorar la fluidez en la comunicación.</p>							
Fecha	Estado		Comentario				
07/16	Definido		Fase de Inicialización Esfuerzo estimado para satisfacer ésta HU: 4 días				

Tabla 5-5: Historia de usuario 2 “navegación entre pantallas”

Numero/ ID	Tipo	Dificultad		Esfuerzo		Prioridad	Notas
		Antes	Después	Estimado	Gastado		
3	Nueva Arreglada Extendida	Fácil Moderado Difícil	Fácil Moderado Difícil	--	--	Alta Media Baja	Complejidad del vocabulario según la edad Paciente Niño con PCI (usuario final) #Requerimiento inicial vinculado:Nº4
Descripción							
<p><u>Funcionalidad deseada:</u> Como terapeutas necesitamos que el SAC tenga un vocabulario más completo y complejo a mayor edad del paciente. Para acompañar el proceso de evolución del lenguaje que utilicen para comunicarse.</p>							
Fecha	Estado		Comentario				
07/16	Definido		Fase de Inicialización Esfuerzo estimado para satisfacer ésta HU: 4 días				

Tabla 5-6: Historia de usuario 3 “complejidad del vocabulario”

Numero/ ID	Tipo	Dificultad		Esfuerzo		Prioridad	Notas
		Antes	Después	Estimado	Gastado		
4	Nueva Arreglada Extendida	Fácil Moderado Difícil	Fácil Moderado Difícil	--	--	Alta Media Baja	Alertas de situaciones predecibles Paciente Niño con PCI (usuario final) #Requerimiento vinculado: Nuevo Requerimiento
Descripción							
<p><u>Funcionalidad deseada:</u> Como terapeutas necesitamos que el SAC sugiera actividades posibles según el horario del día, por ejemplo comer al medio día y a la noche. Esto permitirá anticipar situaciones comunicativas y reducir procesos de búsqueda.</p>							
Fecha	Estado		Comentario				
07/16	Definido		Fase de Inicialización Esfuerzo estimado para satisfacer ésta HU: no definido				

Tabla 5-7: Historia de usuario 4 “alertas según horarios”

Como resultado del trabajo con las fichas entregadas a los interesados se obtuvieron cuatro historias que ayudan a reforzar los requerimientos hasta ahora descritos y a clarificar lo que se necesita para el desarrollo. Asimismo, con estas historias y las acotaciones que se hicieron en las distintas reuniones, se ahonda sobre algunos aspectos específicos, por ejemplo:

- Los niños con PCI deberían poder seleccionar el género y la edad al iniciar la aplicación. Esto permitiría que los pacientes accedan a una interfaz personalizada que les ofrezca imágenes acordes a esta selección. Éste requerimiento se origina en que las necesidades comunicativas de un niño de 2 años no son las mismas que las de un niño de 7 por ejemplo, pues éste ya está escolarizado. Un análisis similar puede realizarse con respecto al género de los pacientes.
- Sobre la complejidad y riqueza del *vocabulario* a usar en la aplicación según el área trabajada: se determinó la necesidad de que los usuarios puedan comunicar sus *sentimientos* tales como alegría, tristeza, o dolor; las *necesidades básicas* como comer, tener sed, ir al baño; y *deseos* como por ejemplo moverse, leer, jugar, entre otros, siempre manteniendo un orden lógico. Esto implica que, cuando se indique la acción *comer*, la aplicación ofrezca una lista de alimentos tentativos, o si se selecciona la opción *jugar*, se facilite el acceso a un conjunto de juegos potenciales, para citar algunos ejemplos.
- Utilizar una *jerarquía de palabras* para las acciones determinadas que se intente comunicar, lo cual incide directamente en cómo se agruparán las palabras según el tipo de acción a realizar. Esto nos brinda una mirada más amplia a cerca del grado de detalle a considerar a la hora de seleccionar las imágenes que conformarán la interfaz. El primer nivel de la jerarquía representaría el nombre de la acción más general, que contendría o agruparía acciones más específicas, como ejemplo un caso puntual se muestra en la tabla 5.8:

Acción general	Acción intermedia	Acción específica
Almorzar	Comer	(Opciones de comidas específicas) Fideos Pizza Pollo Arroz Sopa ...
	Beber	(Opciones específicas de bebidas) Agua Jugo de naranja Jugo de pera Jugo de manzana Gaseosa ...
	Comer Postre	(Opciones específicas de postres) Manzana Flan Gelatina Helado ...

Tabla 5-8: *Ejemplo de jerarquía de palabras para el caso puntual de Almuerzo*

Como resultado de la tarea realizada hasta el momento se obtuvo una primera aproximación a una jerarquía de palabras a considerar en el diseño de la interfaz, junto con las aclaraciones obtenidas sobre los requerimientos iniciales.

Con toda la información recolectada y buscando una retroalimentación sobre las expectativas funcionales de la aplicación, consolidar y validar de las necesidades planteadas, se procederá a diseñar un soporte gráfico de las interfaces que permitirán satisfacer los requerimientos iniciales. Para ello se realizará un prototipo funcional que plasme las interfaces necesarias y el recorrido entre ellas. Los aspectos a evaluar con este prototipo son:

HU1: Correcto agrupamiento de palabras.

HU2: Esfuerzo de navegación adecuado entre las pantallas.

HU3: Adecuación de la complejidad del vocabulario a la edad del paciente.

Para diseñar las interfaces de esta primera iteración se utilizará la herramienta *Justinmind Prototyper*¹⁹ que permite crear pantallas interactivas. Dado que la aplicación está destinada a ejecutarse en dispositivos móviles, se utilizará una tableta para que el grupo de interesados realice las pruebas de navegación y de adecuación de imágenes y vocabulario en este primer prototipo.

Para el diseño de las pantallas se recopilan y organizan imágenes representativas de lo que los usuarios necesitan comunicar, las cuales se muestran en la figura 5.3. Cabe destacar que estas imágenes fueron adquiridas mediante búsquedas (basadas en sistemas de funcionamiento similar al deseado para el desarrollo) en la web, ya que los terapeutas no especificaron de dónde obtenerlas en esta primera instancia.


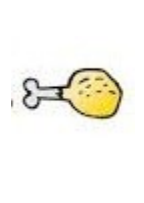




Palabra/frase	Imagen representativa	Palabra/frase	Imagen representativa	Palabra/frase	Imagen representativa
Fideos		Pollo		Bebidas	
Comidas		Sandwich		Pizza	
...

Figura 5-3 : *Imágenes utilizadas en el diseño de las interfaces – Iteración 0*

A continuación, en la figura 5.4 se ilustra una captura de pantalla del proceso de construcción del prototipo haciendo uso de *Justinmind Prototyper*, donde se puede observar un diseño de interfaz inicial para el prototipo.

¹⁹ <https://www.justinmind.com/enterprise> [Consultado el 5 de diciembre de 2016]



Figura 5-4: Construcción del prototipo funcional

En el diseño de las interfaces y en la lógica de navegación de esta iteración se tuvieron en cuenta las indicaciones recibidas por parte del grupo de interesados: género del usuario, su edad, las necesidades comunicativas descritas por las historias de usuario, y las consideraciones generales acerca la navegación. Se decide incluir en la Fase de Producción la implementación del requerimiento de reproducción sonora de una acción cuando el usuario seleccione un ícono de la interfaz y el ofrecimiento de actividades según las horas del día.

Algunas de las interfaces obtenidas se muestran en la figuras 5.5.



Figura 5-5: Ejemplo de interfaces para iteración 0

En la figura 5.6 se ilustra la interfaz y la navegación del prototipo del SAC para el ejemplo de niños varones de 2 a 4 años de edad. En la misma se aprecia el ofrecimiento de acciones que el paciente puede elegir y que tendrán asociada una reproducción sonora (a implementar en la próxima iteración), como así también la secuencia de acciones que se deben ejecutar para alcanzar dicha reproducción. Las acciones incluidas en la interfaz dependen de la edad configurada.

En el prototipo diseñado en esta iteración la selección del grupo de edad y el género al que pertenece el paciente se realiza cada vez que se inicia la aplicación dado que no se cuenta con una memoria de los datos del usuario. En caso de seleccionar género femenino, la interfaz ofrecerá las imágenes de acciones representadas por una mujer, mientras que al seleccionar género masculino las imágenes son referentes a un varón.

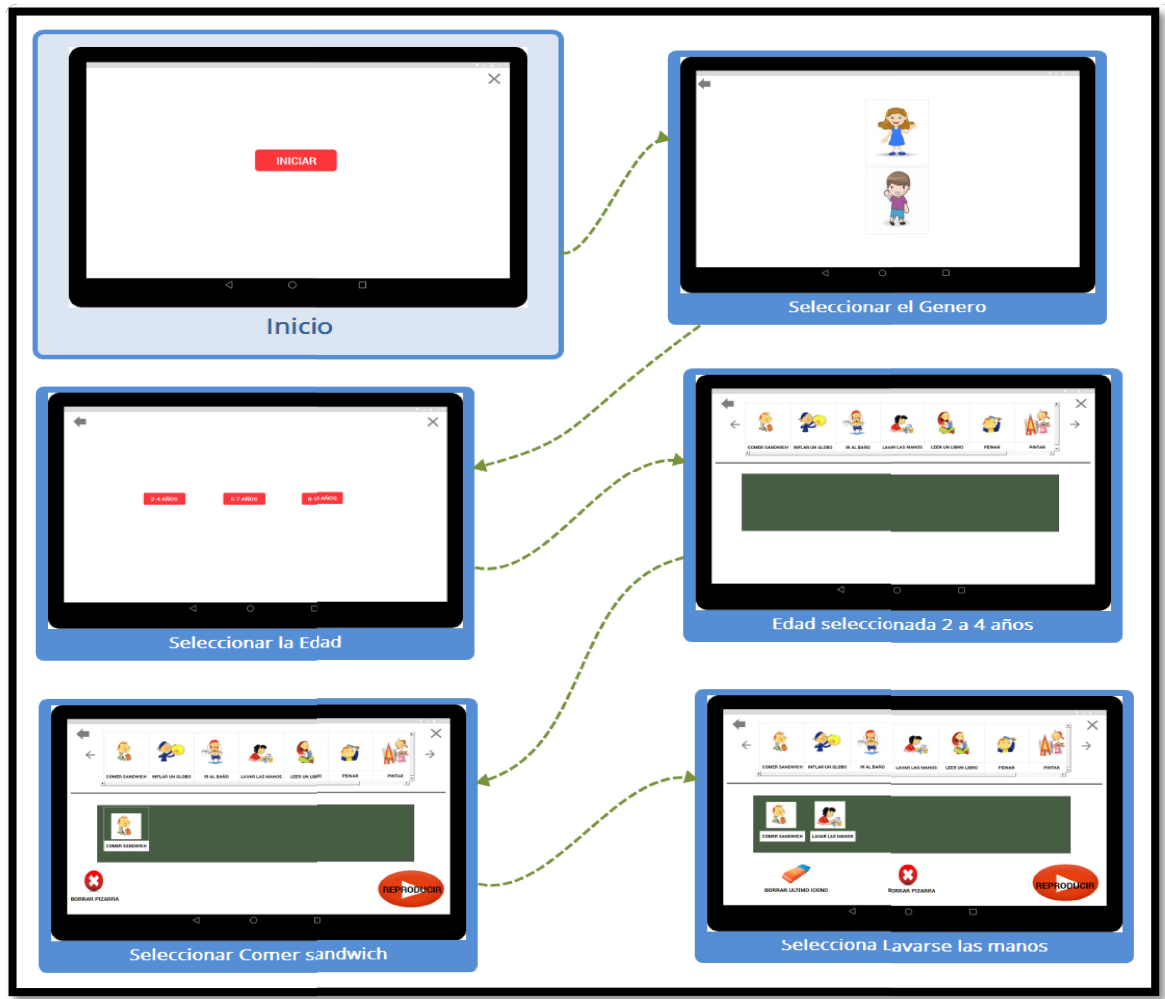


Figura 5-6: Ejemplo interfaces/ navegación del prototipo, niños de 2 a 4 años

En la figura 5.7 se puede observar un ejemplo de cómo se organizaría la jerarquía de palabras propuesta en la tabla 5.8 para la acción general “Almorzar”. En primer lugar se elige la actividad “Almorzar”, luego se despliega una lista de opciones inherentes a dicha actividad, como por ejemplo comidas específicas, “Bebidas” y “Comer postres”. El paciente selecciona una alternativa de las opciones ofrecidas, en el ejemplo se observa que selecciona “Comer postres”, y el prototipo SAC reproduce la salida de audio correspondiente. En este ejemplo en particular, puede existir un submenú de opciones “postres” antes de que se presenten las alternativas con salidas de voz asociadas.

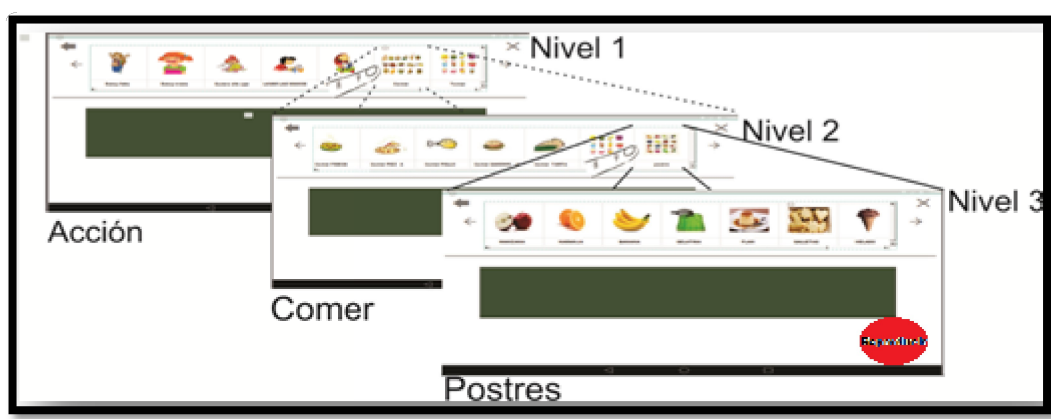


Figura 5-7: Ejemplo de jerarquía de palabras para el SAC

Este prototipo será presentado a los profesionales del IPRI para realizar pruebas de aceptación para función solicitada.

5.2.4. Generación de Pruebas de Aceptación

Evaluar la aceptación del prototipo inicial implica que el grupo de interesados interactúe con la aplicación de manera de apreciarla desde las áreas específicas: fonoaudiología, terapia ocupacional, psicomotricidad, entre otras. Los resultados de esta interacción se ven reflejados en las historias de usuario²⁰ que fueron completadas por los terapeutas describiendo la funcionalidad real aplicada en el prototipo, y que se resumen en las Tablas 5.9 a 5.11, así como también en la ficha de verificación, consignada en el anexo del presente trabajo.

²⁰ En estas historias de usuario se eliminan algunas columnas de las presentadas precedentemente a los fines de facilitar la lectura.

ID	Tipo	Dificultad	Prioridad	Notas
1	Nueva Arreglada Extendida	Fácil Moderado Difícil	Alta Media Baja	Agrupamiento de palabras Niño con PCI (usuario final) #Requerimientos iniciales vinculados: N°2, N°6 y N°7
Descripción				
<p><u>Funcionalidad deseada:</u> Como terapeutas consideramos que las palabras disponibles en el SAC deben estar ordenadas según un esquema lógico de selección, es decir agrupadas por categorías o tipos de acción, para reducir los desplazamientos necesarios al momento de comunicarse y que se facilite encontrar lo que quiere expresar.</p>			<p><u>Funcionalidad real del prototipo:</u> La secuencia lógica de la organización es adecuada, pero la cantidad de selecciones para reproducir una acción no lo es. Las imágenes seleccionadas deberían reproducirse de manera automática. No es adecuado seleccionar la imagen y luego seleccionar reproducir. Las imágenes a utilizar deberían ser las que se usan hoy en las terapias porque son las que el paciente identifica.</p>	
Fecha		Estado		Comentario
08/16		Verificado		Fase de Inicialización Esfuerzo gastado en ésta HU: 4 días

Tabla 5-9: Validación del “agrupamiento de palabras” según terapeutas

ID	Tipo	Dificultad	Prioridad	Notas
2	Nueva Arreglada Extendida	Fácil Moderado Difícil	Alta Media Baja	Navegación entre pantallas del prototipo Paciente #Requerimientos iniciales vinculados: N°5 y N°7
Descripción				
<p><u>Funcionalidad deseada:</u> Como terapeutas necesitamos que el sistema no necesite en ningún caso grandes esfuerzos motrices del usuario para ejecutar sus opciones, para mejorar la fluidez en la comunicación.</p>			<p><u>Funcionalidad real del prototipo:</u> La cantidad de pantallas por las que el paciente navega desde que inicio la aplicación hasta la reproducción de una necesidad es demasiada y complicada de manejar. Se requiere que el paciente: pueda abrir la aplicación y ésta reconozca su configuración personal en una única selección. No deben existir botones para retroceder por el SAC o para cambiar de pantalla, sino que esto debe ser automático.</p>	
Fecha		Estado		Comentario
08/16		Verificado		Fase de Inicialización Esfuerzo gastado en ésta HU: 6 días

Tabla 5-10: Validación de la “navegación entre pantallas” según terapeutas

ID	Tipo	Dificultad	Prioridad	Notas
3	Nueva Arreglada Extendida	Fácil Moderado Difícil	Alta Media Baja	Complejidad del vocabulario según la edad Paciente #Requerimiento vinculado: N°4
Descripción				
<u>Funcionalidad deseada:</u> Como terapeutas necesitamos que el SAC tenga un vocabulario más completo y complejo a mayor edad del paciente. Para acompañar el proceso de evolución del lenguaje que utilicen para comunicarse.			<u>Funcionalidad real del prototipo:</u> Esto es adecuado en el prototipo funcional del comunicador.	
Fecha	Estado		Comentario	
08/16	Verificado Implementado		Fase de Inicialización Esfuerzo gastado en ésta HU: 4 días	

Tabla 5-11: Validación terapéutica de la “complejidad del vocabulario”

Como resultado de este proceso de evaluación se recabó la siguiente información:

- La organización lógica del vocabulario resulta demasiado compleja de acceder para niños con PCI, ya que requiere varios desplazamientos por la aplicación.
- Se requiere el vocabulario asociado a las imágenes propias de ARASAAC²¹, pues son ellas las que actualmente utilizan para comunicarse.
- En cuanto al esquema de navegación, se considera inadecuado el ingreso a la aplicación ya que requiere esfuerzos innecesarios. El procedimiento para

²¹ <http://www.arasaac.org/> [Consultado última vez 22 de Febrero del 2017]

reproducir una frase o acción es engorroso ya que requiere varias acciones (buscar la imagen, seleccionar y luego reproducir).

- Cuando sólo se requiere borrar una acción seleccionada por equivocación, el prototipo incurre en poder borrar todo el tablero o borrar una por una las acciones que no se desee reproducir. Esto es inadecuado.
- El paciente puede requerir expresar una acción compuesta. Esto demanda un proceso de selección complejo en este prototipo, ya que implica seleccionar varias acciones seguidas. Si se produjera un error de orden en una de las acciones intermedias, se deben borrar tanto la acción errónea como las posteriores, y volver a insertarlas en el orden correcto. Para volver a insertarlas hay que buscarlas nuevamente, seleccionarlas y reproducirlas. Esto es inadecuado.
- El seleccionar acciones conjuntas como por ejemplo comer arroz y beber agua se encuentran en diferentes pantallas. Esto obliga a realizar acciones de retroceso en la interfaz, complicando el proceso de comunicación.

Como resumen de la experiencia de evaluación, en forma conjunta el grupo de interesados y el equipo de desarrollo determinaron las siguientes modificaciones a incorporar en la siguiente iteración:

- Las frases o palabras (representadas por imágenes) en el comunicador deben reproducirse automáticamente ante la selección de una imagen, con el objeto de reducir el número de desplazamientos por la aplicación, eliminando el uso de botones de reproducción, es decir, con solo presionar la imagen debe reproducirse el sonido.
- Para simplificar el uso del SAC se deben eliminar los botones de reproducir y borrar, ya que éstos no le agregan funcionalidad a la aplicación.
- Es necesaria la identificación unívoca de cada perfil de paciente. La información personal de cada uno debe estar almacenada de manera persistente en el dispositivo, evitando que en cada ingreso a la aplicación se deba reingresar todos los datos correspondientes a la preferencia del usuario.
- El SAC necesita anticipar acciones al usuario. Esto implica identificar el horario del día en el que se puede realizar una actividad específica.

En base a estas observaciones y al trabajo que se desarrolló hasta el momento, se pueden refinar los requerimientos iniciales. De esta manera, se propone un nuevo catálogo de requerimientos, en donde cada uno de ellos se puede clasificar como un requerimiento general de diseño, denotado como RG, o como una funcionalidad del sistema, denotada como RF. Además, en los casos que corresponda, se identifica la sensibilidad al contexto temporal.

- RG_1. El SAC se ejecutará en dispositivos móviles.
- RG_2. El SAC deberá permitir registrar a cualquier paciente que concurre al IPRI con PCI leve o moderada.
- RG_3. El SAC no deberá requerir en ningún caso grandes esfuerzos motores del usuario (movimientos de brazos o de cuerpo entero).
- RG_4. El paciente no deberá recibir informes sobre errores técnicos de la aplicación que puedan distraer su atención.
- RF_1. El SAC ofrecerá en su interfaz íconos o imágenes seleccionables que representen acciones que el usuario desea llevar a cabo
- RF_2. Los íconos o imágenes seleccionables serán pictogramas o fotos pre-obtenidas, es decir, almacenadas en la aplicación.
- RF_3. Los íconos o imágenes seleccionables deberán ser pictogramas de ARASAAC, con excepción de las imágenes propias de nuestra cultura.
- RF_4. El SAC deberá ofrecer las imágenes seleccionables al usuario acordes al contexto temporal (hora del día).
- RF_5. El SAC permitirá la configuración personal de horarios en los que el usuario realice determinadas actividades. *Requerimiento sensible al contexto temporal.*
- RF_6. El SAC deberá diferenciar el ofrecimiento de las imágenes seleccionable dependiendo de si el usuario se encuentra en la *casa* o la *terapia*.
- RF_7. El SAC deberá ofrecer alertas de alimentación considerando un horario configurado en su registro de usuario. *Requerimiento sensible al contexto temporal.*
- RF_8. El SAC reproducirá en forma sonora una acción (frase o palabra) cuando el usuario seleccione un ícono de la interfaz (imagen).
- RF_9. El SAC reproducirá una salida sonora que represente la acción seleccionada (pictograma) en nuestro idioma y con los modismos de la cultura santiagueña.
- RF_10. El SAC mostrará imágenes o pictogramas según el género y el rango de

edad del paciente.

Una vez redefinidos los requerimientos con el grupo de interesados, el equipo de desarrollo define la denominada *pila de productos*, que se considerará entrada de la siguiente etapa de Mobile-D, por lo que será presentada posteriormente.

5.3 DÍA DE PRUEBA DE LA ITERACIÓN 0

La etapa *día de prueba para iteración 0* se identifica como “*tipo: patrón de etapa de clasificación recomendada*” dentro del método Mobile-D. Puede ser presentada con el siguiente resumen:

Motivación del día de prueba:

- Probar y configurar aún más el entorno de desarrollo técnico y asegurarse de que todo esté listo para implementar el producto. Además, de implementar alguna funcionalidad básica del sistema. También es posible realizar nuevas investigaciones tecnológicas en esta etapa. Los días de prueba forman la fase previa para los días reales de desarrollo.

Metas del día de prueba:

- Asegurar la funcionalidad del entorno de desarrollo técnico.
- Implementar alguna funcionalidad básica del sistema.

Entrada al día de trabajo para iteración 0:

- La pila de Productos
- El plan del proyecto (incluyendo el plan de línea de la arquitectura)

Salida del día de prueba:

- La funcionalidad implementada que se planificó en las iteraciones. Sin embargo, la salida del Día de Prueba no necesariamente puede ser un producto de software como es el caso en la etapa de Día de Trabajo.

Tareas específicas a realizar en el día de prueba:

- Implementar historias, es una selección opcional de implementación para el día de prueba, si se opta por esta se produce código de trabajo tal como se describe en la *fase de Producción en el Día de trabajo*- implementando requerimientos funcionales seleccionados en el análisis de requerimientos iniciales durante el Día de Planificación.
- Implementar otras tareas es para implementar historias de usuarios, por lo cual se decide revisar y resolver problemas críticos de desarrollo en lugar de producir código de trabajo para las historias.
- Informar a los interesados, en esta tarea se proporciona una visión honesta del progreso y se da la posibilidad de dar retroalimentación sobre las características implementadas para guiar el desarrollo.

Roles implicados en el día de prueba:

- Equipo de exploración
- Grupos de interesados

Puede considerarse que el desarrollo de la presente etapa se realiza mediante la implementación de otras tareas, pues no se desarrolla código, en cambio se refinan detalles que afectan a la etapa posterior. Se define en esta instancia la lista final de funcionalidades que orientará el desarrollo de la próxima fase. En la tabla 5.12 se contemplan todas las funcionalidades del SAC y, siguiendo el orden de prioridades asignado por los interesados, se consigna en qué iteración se prevé implementar cada una.

Funcionalidades del SAC	# iteración en que se implementará
El SAC deberá permitir registrar a cualquier paciente que concurre al IPRI con PCI leve o moderada.	2
El paciente no deberá recibir informes sobre errores técnicos de la aplicación que puedan distraer su atención.	4
El SAC ofrecerá en su interfaz íconos o imágenes seleccionables que representen acciones que el usuario desea llevar a cabo.	1
El SAC deberá ofrecer las imágenes seleccionables al usuario acordes al contexto temporal (hora del día).	3
El SAC permitirá la configuración personal de horarios en los que el usuario realice determinadas actividades.	3
El SAC deberá diferenciar el ofrecimiento de las imágenes seleccionable dependiendo de si el usuario se encuentra en la casa o la terapia.	1
El SAC deberá ofrecer alertas de alimentación considerando un horario configurado en su registro de usuario.	3
El SAC reproducirá en forma sonora una acción (frase o palabra) cuando el usuario seleccione un ícono de la interfaz (imagen).	1
El SAC reproducirá una salida sonora que represente la acción (pictograma) seleccionada en nuestro idioma y con los modismos de la cultura santiagueña.	1
Mostrará imágenes o pictogramas según el género y el rango de edad del paciente.	2

Tabla 5-12: *Funcionalidades de SAC priorizadas*

A continuación se organiza una *pila de productos* que indica en qué iteración se planifica alcanzar un determinado grupo de requerimientos, para ello se lo expresa gráficamente mediante la figura 5.8.

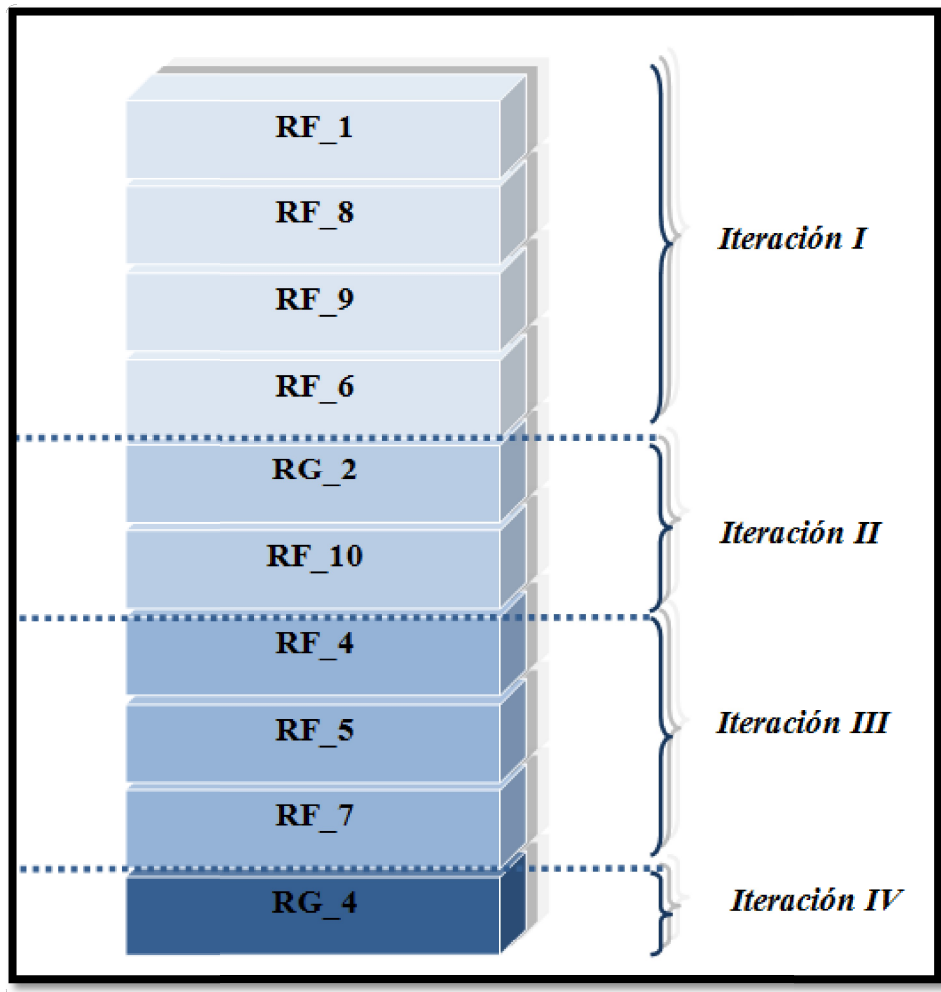
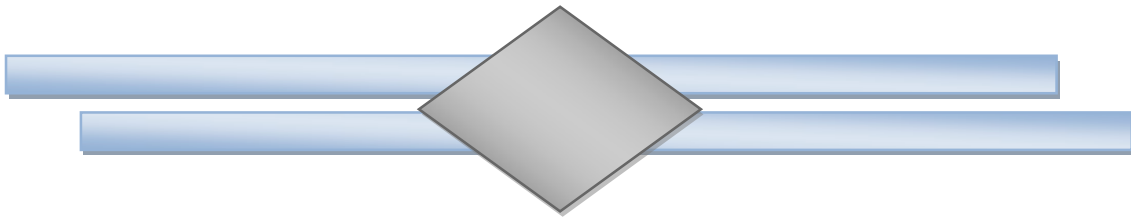


Figura 5-8: Pila de productos para el desarrollo del SAC

5.4 CONCLUSIÓN

En este capítulo se ha desarrollado la *Fase de Inicialización* propuesta por el método Mobile-D, obteniendo como documentación principal funcionalidades del SAC expresadas en historias de usuario, implementadas mediante el prototipo inicial, y evaluadas a través de la interacción del grupo de interesados con mencionado prototipo. Todo este proceso ha dado lugar a una primera versión de la cartera o pila de productos a obtener y al documento de requerimientos iniciales actualizado. Este último con importantes aclaraciones para guiar la siguiente Fase de Producción.

Capítulo 6 - Fase de Producción



La fase de Producción se identifica como “tipo: patrón de fase” dentro del método Mobile-D y el propósito de ella es implementar las funcionalidades requeridas en el producto, aplicando ciclo de desarrollo iterativo y incremental. Se presenta a continuación el resumen de esta fase:

Metas de la fase de producción:

- Implementar la funcionalidad del producto, según priorización del cliente.
- Centrarse en la funcionalidad básica crucial implementándola en el incremento inicial para permitir múltiples ciclos de mejora.

Entrada de la fase de producción, antes de la iteración 0:

- Plan de proyecto actualizado y plan de línea de arquitectura.
- Primera versión del documento de Arquitectura y Diseño de Software.
- Planes para el control de cuestiones críticas de desarrollo.
- Funcionalidad implementada.
- Cartera de productos que contiene todos los requerimientos identificados del proyecto (funcionales como no funcionales).
- Datos de las métricas, si existieran.
- Experiencia del equipo del proyecto.
- Documentación de la prueba de aceptación.
- Tarjetas de historias y tareas.
- Datos sobre los recursos gastados.
- Manuales, especificaciones API y otro material de apoyo necesarios.
- Pruebas unitarias.

Entrada de la fase de producción, antes de la sub-secuencia de la iteración 0:

- El resultado de la iteración de producción precedente.

Salidas de la fase de producción:

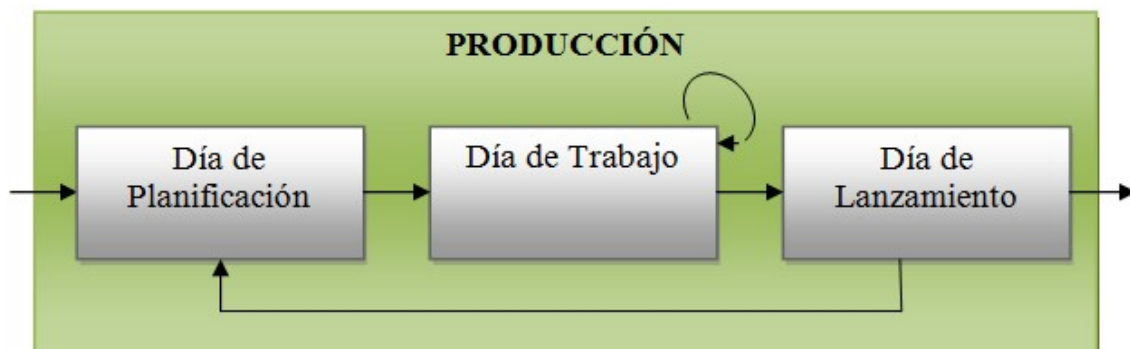
- Funcionalidad implementada.
- Documentación de prueba de aceptación.

- Notas del desarrollador.
- UI – ilustraciones.
- Lista de puntos de acción
- Cartera de productos actualizada.
- Plan actualizado del proyecto.
- Tarjetas de historias y tareas.
- Datos de la experiencia del equipo del proyecto.
- Conocimiento de los requerimientos del sistema y de las pruebas de aceptación.
- Lista de defectos.
- Lanzamiento de la lista de verificación de auditoría.
- Documento de requerimientos iniciales.
- Informe de estado diario

Tareas de la fase que permiten alcanzar las metas y obtener las salidas:

- Día de planificación.
- Día de trabajo.
- Día de lanzamiento.

Resumen gráfico de la fase de producción comprendida como un proceso:



A continuación se describen cada una de estas fases.

6.1 DÍA DE PLANIFICACIÓN

La etapa de *día de planificación* se identifica como “*tipo: patrón de etapa de clasificación esencial*” dentro del método Mobile-D. Se puede presentar la etapa con el siguiente resumen:

Motivación del día de planificación:

- Seleccionar y planificar los contenidos de trabajo para la iteración. Al participar activamente en las actividades de planificación, el cliente garantiza que se identifican los requerimientos que proporcionan la mayor parte del valor del negocio y que dichos requerimientos se entienden correctamente.

Metas del día de planificación:

- Optimizar el proceso de desarrollo de software, según necesidades actuales del proyecto.
- Priorizar y analizar los requerimientos seleccionados para la iteración.
- Planificar el contenido de la iteración (historias y tareas).
- Generar pruebas de aceptación para los requerimientos.

Entrada a la etapa del día de planificación:

- Cartera de productos con los requerimientos identificados del proyecto.
- Datos de las métricas.
- Experiencia del equipo del proyecto.
- Pruebas de aceptación y documentación.

Salida de la etapa del día de planificación:

- Tarjetas de historia y tareas.
- Pruebas de aceptación y documentación.
- Notas para el desarrollador.
- Ilustraciones de interfaz de usuario.
- La lista de puntos de acción de las acciones de SPI.
- Registro actualizado del producto - Plan de proyecto actualizado.

- Datos de experiencia del equipo del proyecto.
- Artefactos de desarrollo generados para apoyar la estimación y la comunicación.
- Conocimiento de los requerimientos del sistema y las pruebas de aceptación aumentadas.

Tareas específicas a realizar en el día de planificación:

- *Taller de post-iteración:* busca mejorar iterativamente el proceso de desarrollo de software para adaptarse mejor a las necesidades del proyecto de software. Incluye la identificación de fortalezas y debilidades en proceso y la generación de acciones de mejora para la siguiente iteración.
- *Análisis de requerimientos:* prioriza y analiza cuidadosamente los requerimientos seleccionados para cada iteración. Durante esta tarea se asegura que los requerimientos que proporcionan la mayor parte del valor del negocio se identifican y se entienden correctamente.
- *Planificación de iteración:* genera el programa y el contenido para que la iteración se ejecute. Los contenidos se definen en términos de tareas, es decir, órdenes de trabajo para el equipo.
- *Generación de pruebas de aceptación:* verifica que los requerimientos establecidos por el cliente se implementan correctamente. Las pruebas de aceptación se generan durante el día de planificación. Posteriormente, se ejecutan con el cliente el día de lanzamiento y se documentan los problemas encontrados (defectos y mejoras).
- *Revisión de las pruebas de aceptación:* se difunde el conocimiento de las pruebas y los requerimientos del sistema a todo el equipo, además se permite a los miembros del equipo comentar sobre las pruebas para mejorar su calidad.

Roles a definir en del día de planificación:

- Equipo de exploración
- Grupos de interesados.

6.1.1. Taller Post-Iteración

El taller post-iteración es una tarea de soporte cuyo propósito es mejorar iterativamente el proceso de desarrollo de software para adaptarse mejor a las necesidades del equipo de proyecto. Con ella se busca identificar fortalezas y debilidades en proceso, generar acciones de mejora para la siguiente iteración, y hacer un seguimiento de la eficacia y ejecución de las acciones de mejora.

Esta tarea no se lleva a cabo dado que está destinada a grandes equipos de desarrollo. Al tratarse de un desarrollo de pequeño, la misma no se considera pertinente.

6.1.2. Análisis de Requerimientos

En esta tarea se define la prioridad de los requerimientos para cada iteración, indicando en qué iteración se implementará cada uno. Para ello se identifican los requerimientos que proporcionan mayor valor agregado a los interesados, según la priorización realizada en la *fase de inicialización*. La tabla 6.1 refleja cómo se mantiene ésta prioridad que determina las implementaciones por iteración tras la iteración de refinamiento realizada en la sección 5.2.3. Además, luego del refinamiento de la *fase de Inicialización*, surgieron requerimientos nuevos que no se habían contemplado en primera instancia. Con respecto a esto, la implementación de los nuevos requerimientos se realiza luego de haber implementado los requerimientos priorizados. Además, la tabla 6.1 lista la pila de productos a desarrollar.

	Requerimiento funcional priorizado por los interesados	Requerimientos refinados en fase de inicialización	Producto
Iteración I	Se desea un SAC que reproduzca en forma sonora una acción (frase o palabra) cuando el usuario seleccione un ícono de la interfaz.	El SAC reproducirá en forma sonora una acción (frase o palabra) cuando el usuario seleccione un ícono de la interfaz (imagen).	Comunicador Básico
		El SAC reproducirá una salida sonora que represente la acción seleccionada (pictograma) en nuestro idioma y con los modismos de la cultura santiagueña.	
	Se desea un SAC que presente en su interfaz íconos o imágenes seleccionables que representen acciones que el usuario desea llevar a cabo.	El SAC ofrecerá en su interfaz íconos o imágenes seleccionables que representen acciones que el usuario desea llevar a cabo.	
	Se desea un SAC que ofrezca imágenes seleccionables al usuario diferenciando si el usuario se encuentra en la casa o la terapia.	El SAC deberá diferenciar el ofrecimiento de las imágenes seleccionable dependiendo de si el usuario se encuentra en la casa o la terapia.	

Tabla 6-1: *Requerimientos y productos por iteración*

	Requerimiento funcional priorizado por los interesados	Requerimientos refinados en fase de inicialización	Producto
Iteración II	Se desea un SAC que muestre imágenes o pictogramas según el género y el rango de edad del paciente.	El SAC mostrará imágenes o pictogramas según el género y el rango de edad del paciente	Comunicador adaptable a las preferencias del usuario
		El SAC deberá permitir registrar a cualquier paciente que concurre al IPRI con PCI leve o moderada.	
Iteración III	Requerimiento Adicional surgido luego del refinamiento	El SAC deberá ofrecer las imágenes seleccionables al usuario acordes al contexto temporal (hora del día).	Comunicador sensible al tiempo.
		El SAC permitirá la configuración personal de horarios en los que el usuario realice determinadas actividades.	
		El SAC deberá ofrecer alertas de alimentación considerando un horario configurado en su registro de usuario.	
Iteración IV	Requerimiento Adicional surgido luego del refinamiento	El paciente no debe recibir informes sobre errores técnicos de la aplicación que puedan distraer su atención.	Comunicador libre informe de errores

Tabla 6-1. *Requerimientos y productos por iteración (Continuación)*

En esta instancia también se indica qué HU se satisfará en cada iteración. En la figura 6.1 se puede observar cómo, al ser iteraciones incrementales, se va agregando funcionalidad al prototipo en cada refinamiento.

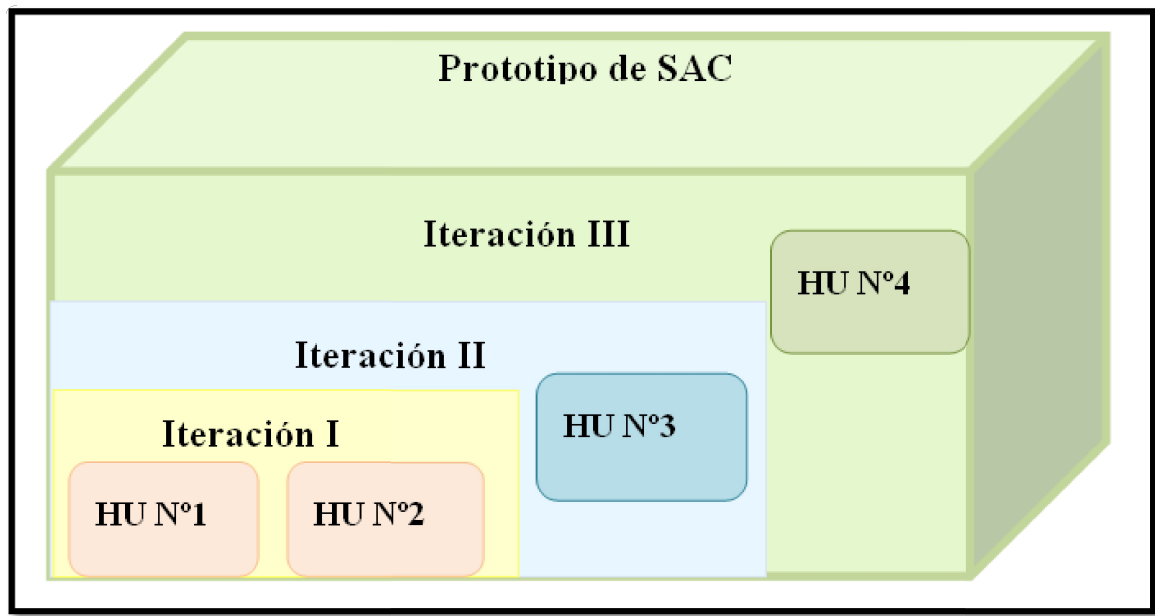


Figura 6-1: *Implementación de historia de usuario en iteraciones*

En la iteración I se implementan las HU N° 1 y N° 2. En base a este desarrollo se incorpora una nueva función al sistema en la iteración II, en este caso la HU N° 3. De la misma manera, en la iteración III, se agrega la HU N° 4 a la funcionalidad del sistema. Es así cómo se van incorporando las necesidades planteadas por los interesados al prototipo, sobre la base de los desarrollos anteriores.

6.1.3. Planificación de la Iteración

La planificación de la iteración es una tarea de tipo esencial para generar la programación y el contenido software. El contenido se define en términos de “órdenes de trabajo para el equipo”, es decir, tareas a desarrollarse en no más de 3 días y que se puedan probar de manera individual. La planificación de la iteración transforma las historias en un conjunto de tareas definidas, y produce borradores de diseño. Además, se estima el esfuerzo requerido para realizar las tareas.

Al finalizar la planificación de la iteración, se debe tener disponibles la lista de historias seleccionadas y tareas que se deben realizar, así como también el desarrollo de artefactos generados para apoyar la estimación y la comunicación.

A partir de las historias de usuario se definen las órdenes de trabajo que se remitirán al equipo de desarrollo, así como también las acciones necesarias para llevar a cabo en cada una de las iteraciones. En esta tarea también se obtienen los borradores de diseño o aproximaciones a los componentes del sistema, en los que se intenta plasmar cómo interactúan los componentes entre sí y los posibles flujos de datos (tipos, origen/destino, etc.). Dependiendo de la iteración, se pueden determinar posibles dependencias entre las tareas necesarias para cada iteración.

En esta instancia se realiza un análisis respecto del contenido de las diferentes interfaces que formarán parte de la aplicación, pues a través de ellas se podrán identificar los flujos de datos necesarios entre los componentes del sistema para poder satisfacer las necesidades de cada desarrollo.

Se describen a continuación las tareas necesarias para satisfacer cada HU y los requerimientos detallados en la tabla 6.1 en cada iteración. También se asocia a cada tarea un esfuerzo estimado en días (EE). La lista de tareas de diseño y programación para cada iteración se encuentra detallada en las tablas 6.2, 6.3 y 6.4.

ITERACIÓN I - HU N° 1 y 2	
TAREAS DE DISEÑO	EE
Determinar los pictogramas o acciones a utilizar para el diseño de las interfaces.	3
Registrar las grabaciones de voz necesarias para la reproducción de los pictogramas que representan acciones del sistema.	1
Determinar qué acciones serán generales y cuáles serán sus acciones específicas. Identificar cada acción ya sea general o específica con pictograma y una grabación de voz.	3
Determinar los pictogramas o acciones a incorporar en la pantalla principal. Representa lo que el usuario puede necesitar en la Casa o en la Terapia.	1
Determinar la cantidad de pantallas necesarias para poder expresar todas las acciones específicas consideradas para esta iteración.	2
Determinar las conexiones entre pantallas según las acciones generales y específicas.	1
TAREAS DE PROGRAMACIÓN	
Reproducir la grabación de voz asociada a un pictograma al seleccionarlo en la interfaz. El pictograma puede representar una acción general o una específica.	2
Abrir una nueva interfaz al seleccionar una acción general representada por un pictograma. La interfaz desplegada con esta selección debe contener las acciones específicas asociadas a la acción general. Por ejemplo al seleccionar la acción general “Jugar” se despliega una pantalla con pictogramas seleccionables que representan diferentes juegos como ser “pelotas”, “muñecos”, “bloques”, etc.	3
Ocultar todas las acciones específicas una vez que cualquiera de ellas sea seleccionada y reproducida. Luego mostrar la interfaz principal o secundaria que contiene la acción general que despliego las opciones específicas. Es decir la aplicación debe regresar a “Casa” o “Terapia” dependiendo de donde se encontraba la acción general que despliego las opciones.	3

Tabla 6-2: *Tareas a llevar a cabo durante la Iteración I*

ITERACIÓN II - HU N° 3	
TAREAS DE DISEÑO	EE
Determinar los pictogramas o acciones que permiten incorporar mayor amplitud de vocabulario a las interfaces.	3
Registrar las grabaciones necesarias para todos los pictogramas que se involucran en esta iteración.	1
Definir las acciones generales y las específicas correspondientes según la edad del usuario. Para ello considerar el conjunto total de acciones en esta iteración.	3
Identificar para cada edad la cantidad de pictogramas por pantalla.	3
Determinar la cantidad de pantallas a incorporar para las edades consideradas.	3
Establecer las conexiones entre las diferentes pantallas según la edad y el género considerado en la iteración.	2
TAREAS DE PROGRAMACIÓN	
Solicitar inicio de sesión al abrir la aplicación y el registro de datos personales del usuario (nombre del usuario, género y su edad).	2
Mantener la sesión de usuario iniciada hasta que el usuario indique que desea cerrar su sesión. La aplicación no solicitará reingreso de los datos personales.	1
Mantener iniciada la sesión de usuario por más que la aplicación sea cerrada o el dispositivo se apague.	1
Para cerrar sesión el usuario indica la acción al sistema. Este borrará los datos personales de la aplicación. Para poder ingresar a la funcionalidad del comunicador, el sistema necesita contar con los datos personales de un usuario.	1
Ofrecer los pictogramas en la interfaz filtrados por el género del usuario.	3
Ofrecer los pictogramas en la interfaz filtrados según la edad del usuario.	5

Tabla 6-3: *Tareas a llevar a cabo durante la Iteración II*

ITERACIÓN III - HU N° 4	
TAREAS DE DISEÑO	EE
Determinar los pictogramas o acciones necesarios para incorporar las alertas de alimentación	1
Registrar las grabaciones de voz necesarias para los pictogramas agregados en esta iteración.	1
Determinar los pictogramas necesarios para las pantallas de alertas	1
Determinar las conexiones entre pantallas	1
TAREAS DE PROGRAMACIÓN	
Solicitar en el inicio de sesión el ingreso de 4 horarios para configuración de actividades para alimentación.	3
Registrar y mantener la hora configurada el alerta por un usuario al iniciar sesión. Es decir, la aplicación no solicitará reingreso de horarios de alerta si la aplicación se cierra o se apaga el dispositivo. Los horarios están asociados a la sesión activa del usuario.	2
Registrar y controlar el horario del dispositivo cuando la aplicación está siendo usada	1
Controlar que la hora actual del dispositivo se encuentre dentro de un rango determinado para activar un alerta configurada en el inicio de sesión. El rango de activación de un alerta será de 60 minutos posterior al horario configurado para el alerta. Es decir si se configura a las 07:35, cuando la hora del sistema se encuentre entre las 07:35 y las 08:35, el alerta estará activada.	3
Indicar qué alerta se encuentra activa.	1
Ofrecer pictogramas de alimentos para comidas principales en solo en la interfaz de <i>Casa</i> cuando un alerta este activa.	2
Ofrecer en la interfaz de <i>Casa</i> los pictogramas de alimentos acordes a la alerta activa. Es decir si se activa un alerta de <i>almuerzo</i> , ofrecer opciones para almorzar.	2

Tabla 6-4: *Tareas a llevar a cabo durante la Iteración III*

En esta instancia se preparan los recursos visuales necesarios para realizar el desarrollo de código ordenadamente. Es importante recordar que las tareas de programación van incorporando funcionalidades nuevas al prototipo en cada iteración por lo tanto se incrementan las funcionalidades a medida que aumenta el número de iteraciones. Asimismo, se realizan las tareas de diseño para cada una de las 3 iteraciones planeadas. Para ello, se comienza seleccionando las imágenes y pictogramas requeridos en cada iteración, lo cual se representa a modo de ejemplo en la tabla 6.5, en donde se empareja cada “Palabra” que representa una necesidad de comunicación con su respectivo pictograma, definiendo cuáles son los pictograma-acción que corresponden a cada iteración





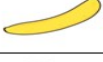







Pictograma	Acción que representa	Pictograma	Acción que representa
	YO ME LLAMO		CHAU
	MAMÁ		SUBIR
	PAPÁ		BAJAR
	ESCUCHAR MÚSICA		MUÑECO
	BANANA		NARANJA
	TOMAR		OSO DE PELUCHE
	CUATRO		PERDON
	SI		VER TELEVISIÓN
	PONER		MANZANA
	BLOQUES		NO

Tabla 6-5: Ejemplo de pictogramas y acciones del comunicador

En la tabla anterior se ilustran algunos de los recursos gráficos acompañados con una palabra o frase que lo describe. Éstos conforman parte de la lista general que contiene todos los “pictograma-acción” para cada una de las iteraciones. La frase que describe la imagen se corresponde con la representación semántica de ella dentro de la cultura santiagueña, es decir, se incorporan al SAC interpretaciones y significados acordes al contexto social y cultural propio.

Basándose en esta lista se realizaron las grabaciones de voz que representan la

semántica asociada al pictograma en el idioma nativo y con los modismos santiagueños. Se pudo acceder a la mayoría de los recursos presentados en la misma por recomendación de los interesados, pues son los que se utilizan para la comunicación con los niños que asisten al IPRI. Estos pictogramas se seleccionaron y filtraron del portal de ARASAAC, el cual contiene un archivo comprimido con un conjunto de 15.396 pictogramas. Estos están disponibles bajo licencia *Creative Commons (CC-BY/NC/SA)*, cuya autorización de uso es para fines sin ánimo lucrativo. Los símbolos pictográficos utilizados son propiedad de CATEDU²² y han sido creados por Sergio Palao. Sin embargo, otros pictogramas que representan imágenes propias de nuestra cultura, no están disponibles en éste portal y sus imágenes se seleccionaron de la web, como ser el caso de alimentos y animales.

La búsqueda, selección y filtrado de pictogramas para el prototipo insumió una labor importante. Esto se debe a que:

- Los recursos gráficos proporcionados por ARASAAC están nombrados con palabras o frases propias a la semántica el contexto donde fueron creados, es decir, dentro de un contexto europeo, pues el portal al que pertenece es del Gobierno de Aragón, España. La interpretación de un mismo pictograma para una persona de nuestra cultura puede no ser la misma que la colocada por el creador de la imagen.
- Hay varios casos donde los nombres son similares pero se provee diferentes pictogramas, como por ejemplo el que se presenta en la figura 6.2.

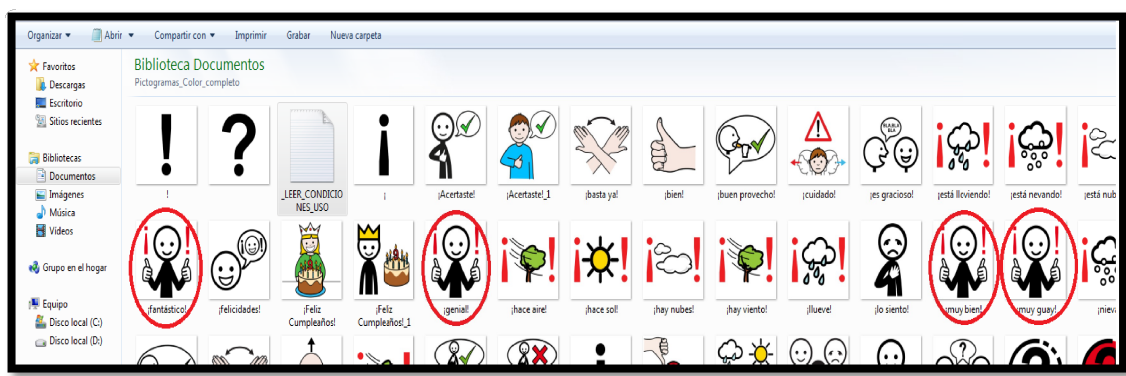


Figura 6-2: Ejemplo de pictogramas iguales con nomenclatura diferente

²² <http://arasaac.org/> Portal de Aragón de la Comunicación Aumentativa y Alternativa

[consultado ultima vez 22 de Febrero del 2017]

Una vez seleccionados los pictogramas y sus acciones asociadas, se clasifican los pictogramas-acción según su generalidad, para cada una de las 3 iteraciones a desarrollar. Es decir, se agrupan para la visualización en las interfaces de cada iteración. Para ello se definen 2 tipos de acciones tal como se muestran en la figura 6.3, considerando la frecuencia de uso en una comunicación de cada pictograma-acción y la adquisición natural del lenguaje en las diferentes etapas evolutivas de los infantes. En base a esta clasificación se realiza una lista de todas las acciones generales y específicas para cada iteración.

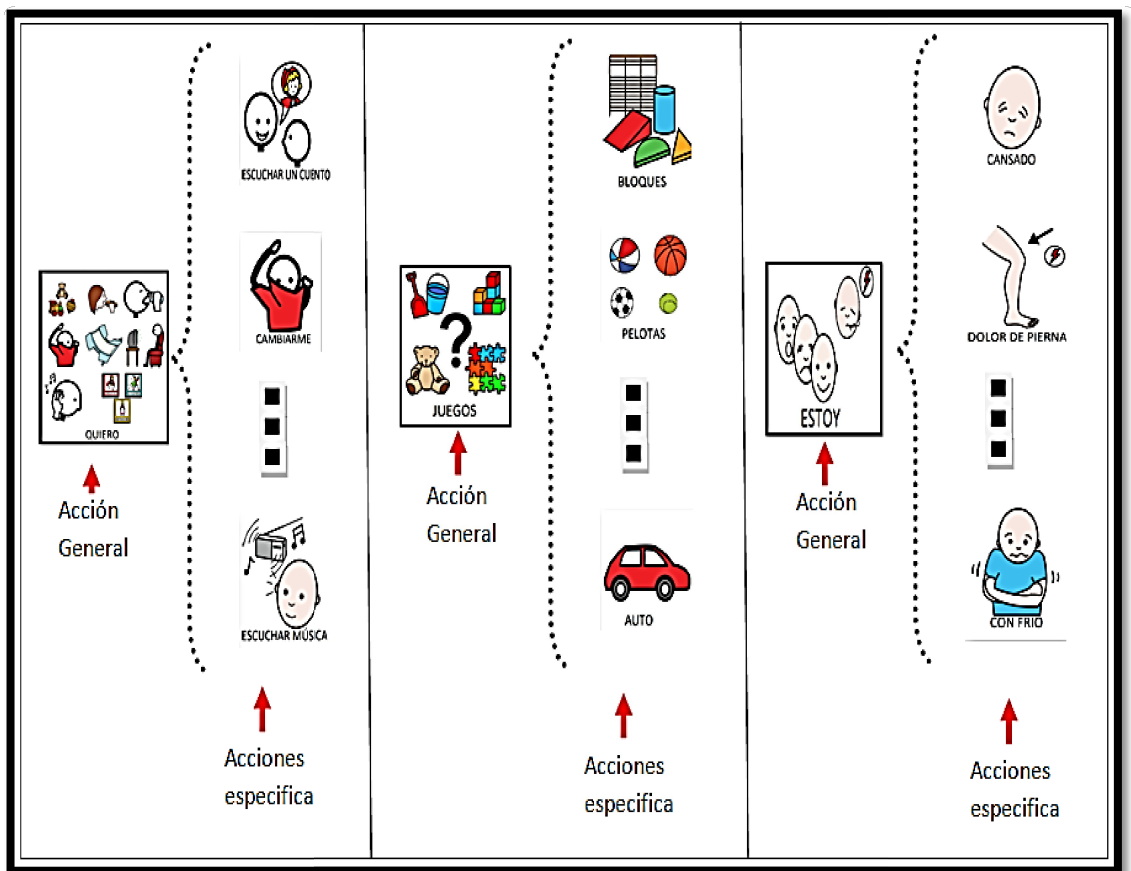


Figura 6-3: Ejemplo de clasificación de acciones según su generalidad

Basados en la clasificación de cada pictograma-acción, se procede al diseño de las interfaces del prototipo, definiendo los contenidos de cada pantalla y determinando la cantidad de pantallas necesarias, según la cantidad de acciones generales y específicas que se incorporen en la iteración.

Se realiza un soporte gráfico de las interfaces, para obtener una comprensión completa y consistente en el desarrollo. En la figura 6.4 se ejemplifican los contenidos gráficos y una distribución visual aproximada de las diferentes interfaces para las 3

iteraciones. Los contenidos de las interfaces se mantendrán a lo largo de todo el desarrollo, es decir los resultados de la programación del prototipo deben ajustarse a lo descrito en esta instancia. La distribución de cada pictograma-acción dentro de la pantalla puede llegar a variar, siempre y cuando haya una adecuada justificación para ello desde el punto de vista de la programación.

En esta instancia se determina que el SAC necesita diferenciar los tipos de acciones de ejecución, por un lado las acciones tipo “reproducción” de acciones específicas, en las que el comunicador solo deberá reproducir un sonido ante su selección; y por otro las acciones de tipo “reproducción y despliegue” de acciones generales, en las que el comunicador debe reproducir la acción seleccionada y abrir una nueva interfaz asociada a la selección.



Figura 6-4: Ejemplos de diferentes interfaces del sistema

Habiendo descrito las consideraciones generales para todas iteraciones sobre el diseño de las interfaces y su composición, se plantea el recorrido de navegación para cada iteración. A modo de ejemplo se ilustra en la figura 6.5 el correspondiente a la desplazamiento de pantallas del “Comunicador básico”, iteración I.

En ella se muestra cómo el usuario percibiría cada pantalla tras su interacción con la aplicación. Por ejemplo, en la pantalla de “Casa” podrá observar dos tipos de acciones: por un lado las de “reproducción” como son *yo me llamo, mamá, papá, si, no, subir, bajar, poner, sacar, hola y chau*, mientras que *Quiero, Estoy y Terapia* son de tipo “reproducción y despliegue”. En todos los casos en los que se seleccione un pictograma de tipo “reproducción”, el comunicador regresa a la interfaz de *Casa*. En caso de seleccionar un pictograma de “reproducción y despliegue”, el comunicador se desplazará hacia otra interfaz y regresará a *Casa* luego de alcanzar una acción de “reproducción”.

Cuando el usuario haya seleccionado *Terapia* considerada como “reproducción y despliegue”, el comunicador registrará que el usuario ingresa en una sesión de terapia y cambiará su comportamiento: mantendrá la interfaz de terapia tras las selecciones, ya sean de “reproducción y despliegue” o de “reproducción”. Cuando la terapia haya concluido y se quiera regresar a la interfaz original de *Casa*, el usuario debe informarle esto al SAC mediante la selección del pictograma “Casa”. Si no lo hace, el comunicador siempre regresará a la interfaz de terapia al seleccionar un pictograma de cualquier tipo.

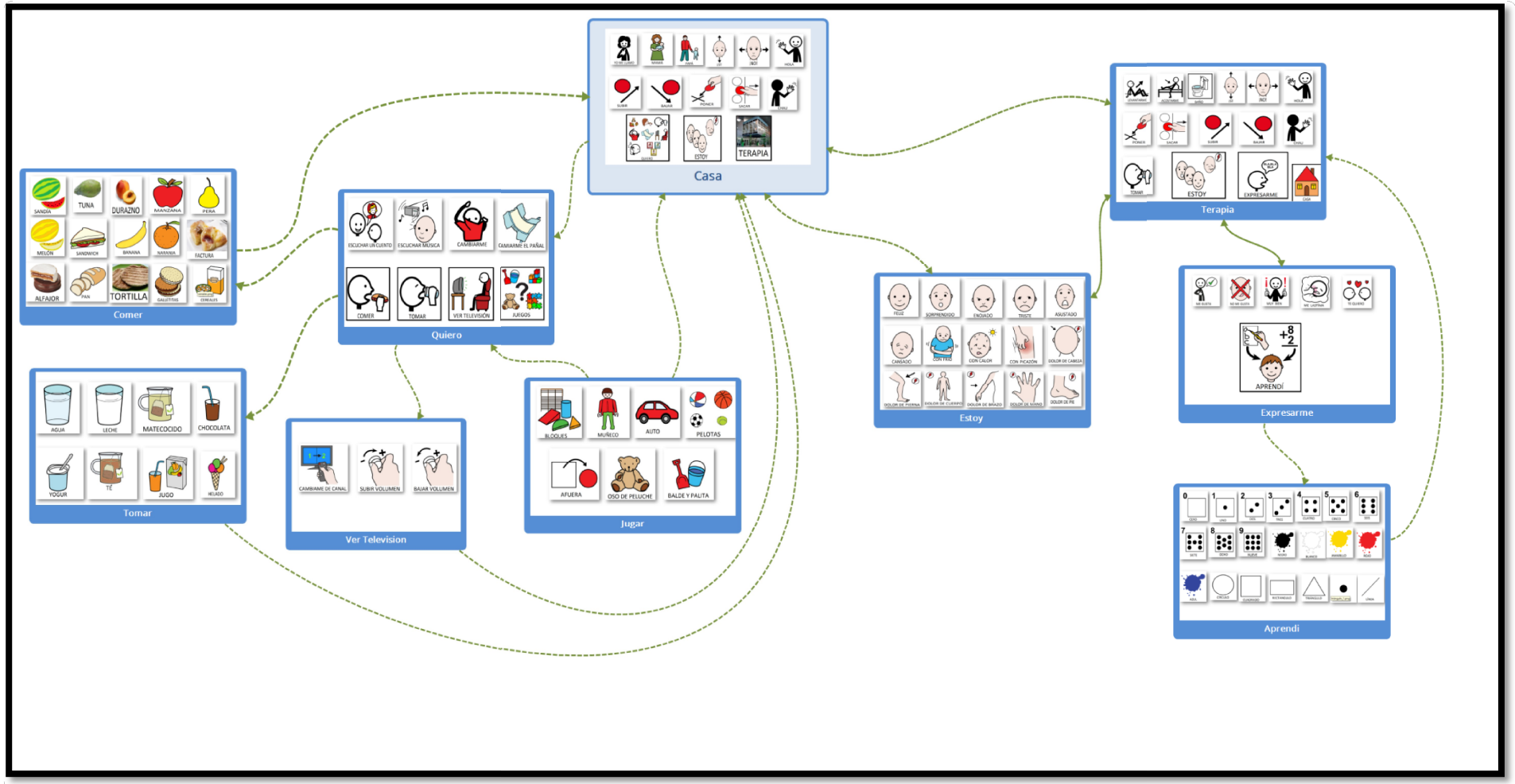


Figura 6-5: Esquema de navegación del comunicador Básico (Iteración I)

Los esquemas de navegación entre pantallas que corresponden a las iteraciones II y III aumentan en complejidad respecto al mostrado para la iteración I. En ellos, el comunicador básico requiere que la interfaz de las iteraciones II y III se adapte a la edad y al género del usuario de la aplicación, por lo cual no se pueden mostrar todas las posibilidades de navegación entre pantallas.

En la figura 6.6 se esquematiza de forma muy general el desplazamiento entre las interfaces que componen la iteración II, presentando una navegación en el comunicador “adaptable a las preferencias del usuario”. A través de esta figura se puede seguir el recorrido de la aplicación sin la necesidad de especificar los detalles inherentes a la interfaz de a cada pantalla. Esto se fundamenta en que la cantidad de interfaces posibles para la iteración II es sustancialmente mayor que las mostradas en el comunicador básico y las conexiones entre ellas pueden llegar a confundir al lector. Para facilitar el seguimiento, se le asigna un nombre a las diferentes interfaces y la navegación se esquematiza con pequeñas “cajas de texto” con el nombre de la interfaz y una serie de flechas que indican cómo serán conectadas, es decir cómo se realizará el recorrido entre las diferentes pantallas. En esta figura y para evitar saturar de flechas de desplazamiento, no se visualiza una transición desde las interfaces “*abecedario*”, “*animales*”, “*cuerpos geométricos*”, “*números*”, “*aprendí*”, “*colores*” y “*dolorido*” hacia “*casa*” pero cabe destacar que estas conexiones sí se consideran para los recorridos de la iteración II.

Dado que en esta iteración la interfaz se adecua a cada usuario en función de la edad y del género del mismo, es necesario contar con un “*login*” para el registro de ciertos parámetros que permitirán adaptar el SAC a la necesidad de cada paciente. Así, para mejor comprensión de la complejidad del general del sistema en esta iteración, en la figura 6.6 las “cajas de texto” de color azul son las que representan un conjunto de pantallas personalizables, mientras que las de color gris son estáticas, es decir no varían para las diferentes sesiones de usuario, o dicho de otra manera, no contienen pictogramas personalizables.

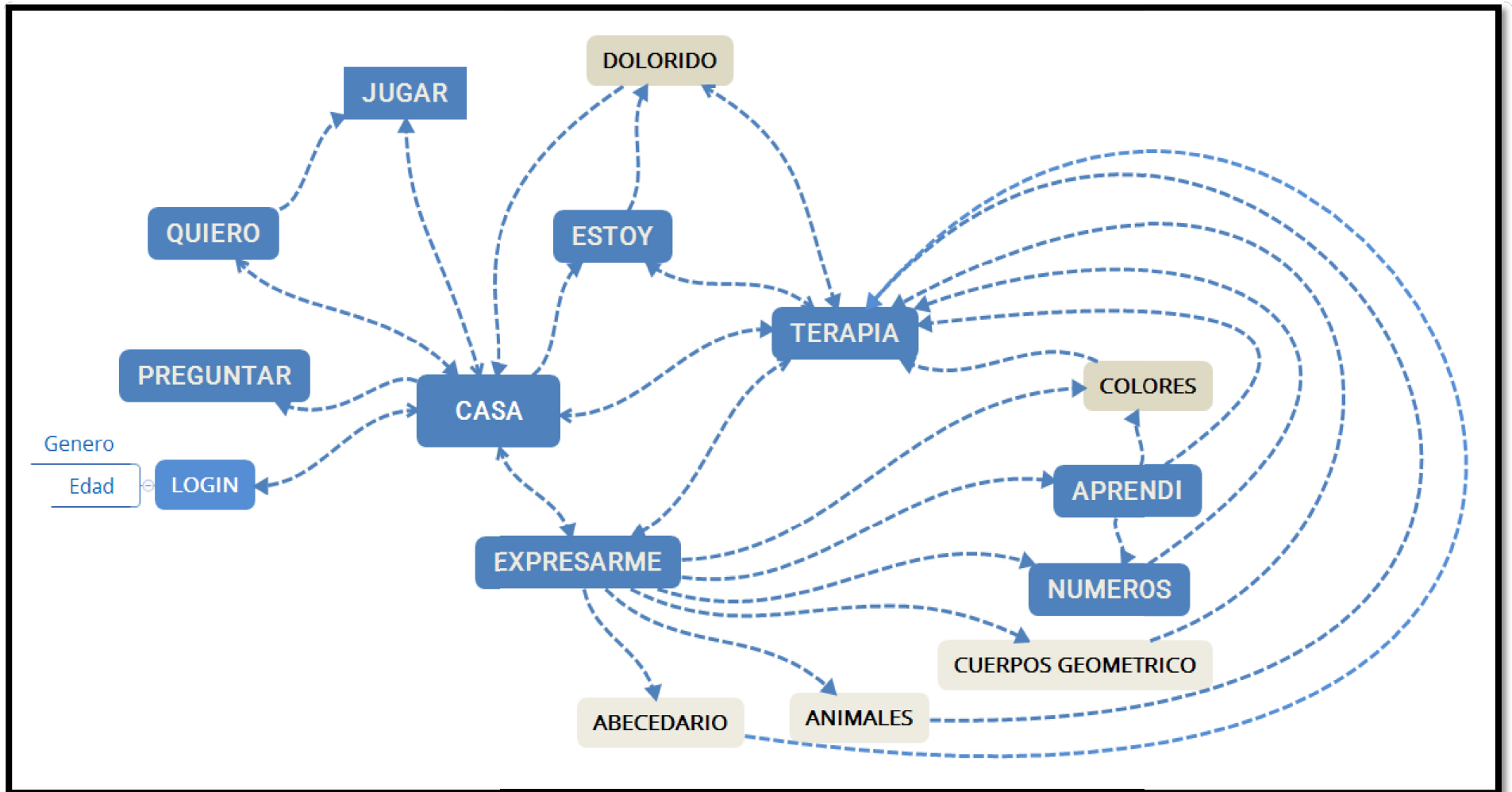


Figura 6-6: Navegación entre pantallas para el “Comunicador adaptable a las preferencias del usuario” (iteración II)

Para la iteración III es decir para el “Comunicador sensible al tiempo”, se sigue un modelado similar a la segunda iteración. Habiendo aclarado esto, en la figura 6.7 se presenta el recorrido entre las interfaces.

En la figura 6.7 se aprecia que la principal variación respecto de la iteración anterior es la presencia de una interfaz denominada “*alerta*”. Ella representa la nueva funcionalidad que se agrega a la iteración anterior, y su función es dejar accesibles nuevos pictogramas en determinados horarios. Estas *alertas* cumplen la función de *sugerentes*, pues se ponen a disposición en la interfaz - alertas visuales - en diferentes momentos del día, según se configure al inicio de sesión.

Para esta iteración las alertas se restringen, proporcionando al usuario solamente una serie de pictogramas que representan aquellas necesidades relacionadas con la alimentación del niño. Por ello, se centra en las comidas principales a lo largo del día, permitiendo que usuario tenga acceso a una serie de pictogramas le permitan expresar las necesidades correspondientes al desayuno, almuerzo, merienda o cena, dependiendo el horario del día.

Para esta iteración se considera la relación existente entre las necesidades de alimentación programadas y el horario actual de ejecución del sistema. Se agregan en el *login* recursos que permiten parametrizar cada horario que el tutor del niño configura para las alertas. De igual manera que se hizo anteriormente, en la figura 6.7 no se representa el desplazamiento desde las pantallas “*abecedario*”, “*animales*”, etc., pero se deja claro que en el recorrido de la iteración esta conexión existe y que por cuestiones de claridad no se grafican.

En lo que respecta a la iteración IV, se trata de una cuestión técnica de desarrollo, es decir de programación del código, e involucra la lógica interna del prototipo. Por este motivo no se especifican interfaces ni desplazamientos de la iteración.

Los resultados obtenidos de la ejecución de las tareas de diseño para cada una de las iteraciones quedan disponibles para ser utilizados en la *etapa de día de trabajo*.

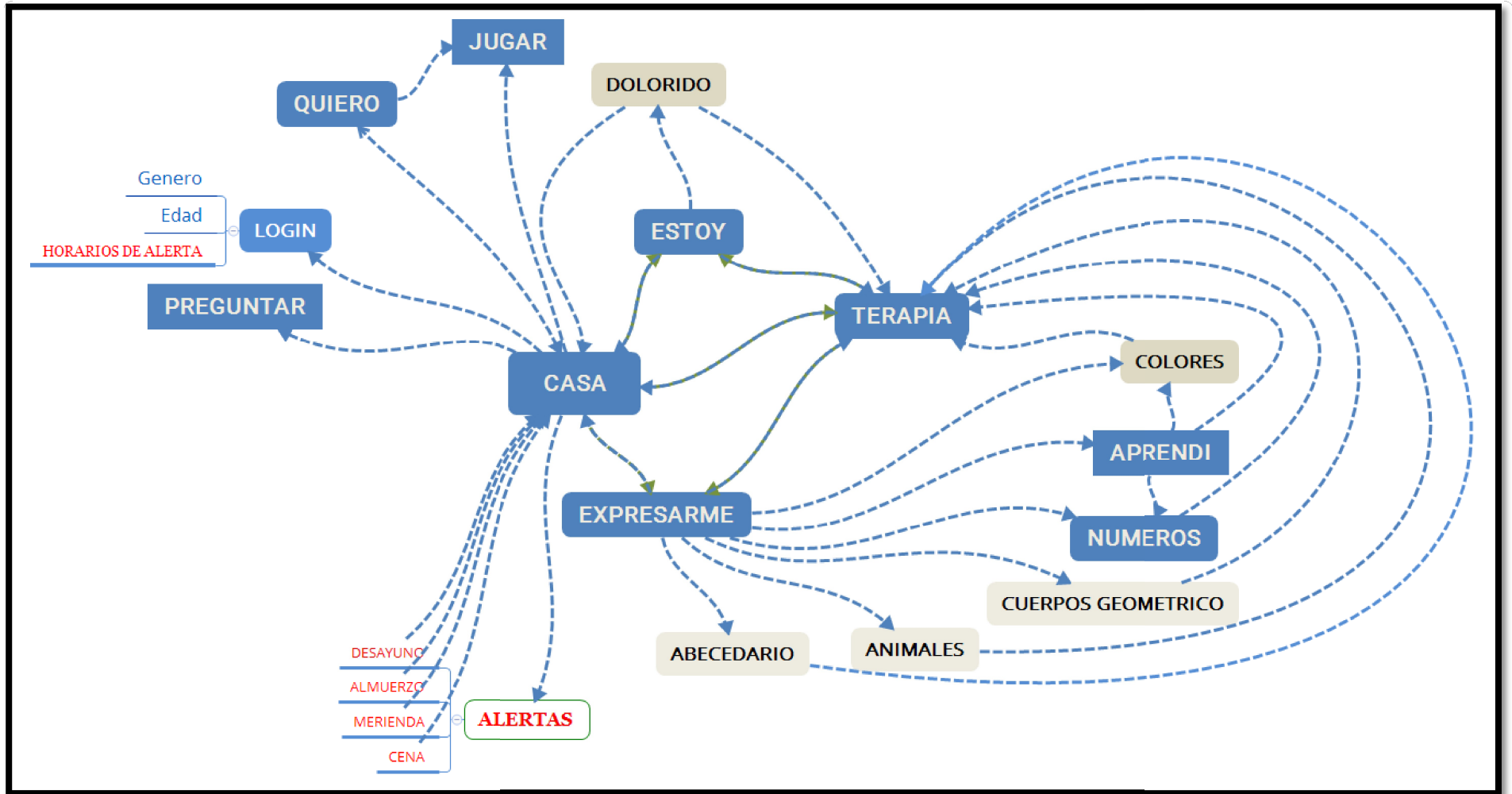


Figura 6-7: Navegación en la iteración III del “SAC sensible al tiempo”

6.1.4. Generación de Pruebas de Aceptación

La generación de pruebas de aceptación es una tarea para verificar los requerimientos establecidos para el software. Actúa como una herramienta de comunicación entre los interesados y el equipo de desarrollo. Para realizarla es necesario considerar los requerimientos del software por cada iteración, además de las HU y las de tareas previstas por iteración. Como parte de esta tarea también se incorpora un paso intermedio: la revisión de la prueba de aceptación durante la redacción, es decir los interesados participan de la redacción de las pruebas de aceptación.

Para la escritura de las pruebas de aceptación se selecciona un requerimiento de la pila de productos y, conjuntamente con los interesados, se procede a describir la prueba pertinente. Como resultado de este trabajo, la tabla 6.6 lista las pruebas de aceptación por requerimiento y resume todas las pruebas de aceptación realizadas para las diferentes iteraciones del desarrollo. Cabe aclarar que el identificador de la prueba sigue el siguiente formato: {Nº de Iteración que se prueba}+{Nombre del requerimiento probado}, para poder ser referenciado de forma más representativa.

PRUEBAS DE ACEPTACION			
Fecha de escritura: Octubre de 2016			
Requerimiento	ID de la prueba	{Qué funcionalidad debe implementar el sistema para dar por satisfecho el requerimiento}	
Iteración I	El SAC reproducirá en forma sonora una acción (frase o palabra) cuando el usuario seleccione un ícono de la interfaz (imagen).	IRF8	Al seleccionar un icono que representa una acción se reproduce automáticamente de forma sonora en el dispositivo.
	El SAC reproducirá una salida sonora que represente la acción seleccionada (pictograma) en nuestro idioma y con los modismos de la cultura santiagueña.	IRF9	La reproducción sonora de cada acción es comprendida sin necesidad de aclaraciones sobre su significado en nuestra cultura.
	El SAC ofrecerá en su interfaz íconos o imágenes seleccionables que representen acciones que el usuario desea llevar a cabo.	IRF1	Las acciones consideradas como parte de la interfaz representan necesidades de comunicación posibles para un niño con PCI leve o moderada.
	El SAC deberá diferenciar el ofrecimiento de las imágenes seleccionable dependiendo de si el usuario se encuentra en la <i>casa</i> o la <i>terapia</i>	IRF6	Al iniciar la sesión de usuario, la interfaz principal corresponde a la de <i>Casa</i> . Cualquier acción que se ejecute (selección de pictograma) en esta interfaz debe retornar siempre a la interfaz de <i>Casa</i> , excepto cuando el usuario indique que va a entrar en <i>terapia</i> , mediante la selección de la imagen representativa. Cuando el usuario le indica a la aplicación que entra en <i>Terapia</i> , la interfaz ofrecida es diferente a la interfaz de <i>Casa</i> . Cualquier acción ejecutada (selección de pictograma) dentro de esta interfaz retorna siempre a <i>Terapia</i> , excepto cuando el usuario indica que regresa a su <i>Casa</i> , mediante la selección del icono que lo representa. En este caso la aplicación retorna a la interfaz de <i>Casa</i> .

Tabla 6-6: Pruebas de aceptación para cada requerimiento por iteración

PRUEBAS DE ACEPTACION		
Fecha de escritura: Noviembre de 2016		
Requerimiento	ID de la prueba	{Qué funcionalidad debe implementar el sistema para dar por satisfecho el requerimiento }
El SAC mostrará imágenes o pictogramas según el género y el rango de edad del paciente	IIRF10	Los pictogramas visibles en la interfaz de género femenino son diferentes a los presentados al masculino.
		Los pictogramas disponibles en las interfaces varían si el usuario es un niño pequeño (por ejemplo 2 años), o mayor (por ejemplo 5 o 9 años).
El SAC deberá permitir registrar a cualquier paciente que concurre al IPRI con PCI leve o moderada.	IIRG2	La funcionalidad del SAC es accesible con el registro de los datos personales del usuario, permitiendo iniciar una sesión al sistema.
		La sesión de un usuario iniciada permanece activa hasta que el usuario indique lo contrario.
		En caso de que se cierre la aplicación voluntariamente o se apague el dispositivo, la sesión permanece abierta sin necesidad de reingreso de datos personales.
		En cualquier instancia durante el funcionamiento de la aplicación el usuario puede cambiar datos de su registro o salir de su sesión.

Tabla 6.6. Pruebas de aceptación para cada requerimiento por iteración (Continuación)

PRUEBAS DE ACEPTACION		
Fecha de escritura: Noviembre de 2016		
Requerimiento	ID de la prueba	{Qué funcionalidad debe implementar el sistema para dar por satisfecho el requerimiento }
Iteración III	IIRF4	La aplicación encuentra coincidencia entre el horario del dispositivo y el horario configurado como alerta y que el usuario interactúa con el sistema, entonces activa el alerta correspondiente.
		Cuando un alerta esté activo el usuario y se encuentre (en la interfaz de <i>Casa</i>), tendrá acceso a pictogramas que representan opciones de alimentos para esa alerta.
	IIRF5	El usuario puede registrar diferentes horarios de alertas relacionados con sus comidas principales: desayuno, almuerzo, merienda y cena.
El registro de las alertas permanecerá relacionado a su sesión de usuario. En caso que la aplicación se cierre o el dispositivo se apague (sin cerrar sesión) los horarios configurados persistirán en su registro.		
	IIRF7	La aplicación detecta que el usuario está interactuando con la interfaz de <i>Casa</i> , y el horario configurado para un alerta coincide con el horario del dispositivo, entonces se ofrece al usuario opciones relacionadas a la alimentación programada en su sesión
Iteración IV	IVRG4	El usuario no visualiza durante el uso intensivo de la aplicación mensajes emergentes de errores internos por defectos en la programación del código.

Tabla 6.6. Pruebas de aceptación para cada requerimiento por iteración (Continuación)

Concluido este trabajo de verificación, se puede trabajar en la implementación en la siguiente etapa.

6.2 DÍA DE TRABAJO

La etapa de *día de trabajo* se identifica como “*tipo: patrón de etapa de clasificación esencial*” dentro del método Mobile-D. Se puede presentar la etapa con el siguiente resumen:

Motivación del día de trabajo:

- El propósito de esta etapa es implementar la funcionalidad del sistema planeada durante el día de planificación. El equipo de desarrollo se centra en la funcionalidad de máxima prioridad definida por los interesados. Una iteración puede contener 1 o n días de trabajo. Los días de trabajo forman los días reales de desarrollo de la iteración.

Metas del día de trabajo:

- Implementar la funcionalidad (es decir, las tareas) asignadas para la iteración
- Mantener al cliente al día con el estado de desarrollo

Entrada a la etapa del día de trabajo:

- Las tarjetas de historias de usuario y las de tareas.
- La pila de productos.
- Los datos sobre los recursos gastados.
- Manuales, especificaciones API y otro material de apoyo.
- Las pruebas unitarias.
- La funcionalidad implementada.

Salida de la etapa del día de trabajo (para el último día laborable en la iteración):

-
- La funcionalidad implementada.
 - El informe de estado diario.

Tareas a realizar en el día de trabajo:

- **Recapitulaciones:** es una sesión interactiva para comunicar el progreso y los problemas dentro del equipo de desarrollo. Suele realizarse como la primera y/o la última actividad de la jornada de trabajo (recapitulación de la mañana y conclusión de la tarde).
- **Desarrollo guiado por pruebas (*Test-Driven Development* -T.D.D.-):** En él las pruebas unitarias se escriben antes del código del programa, entonces éste se desarrolla para trabajar con las pruebas ya escritas.
- **Programación en parejas:** Es un estilo de programación en el que dos desarrolladores codifican juntos. El conductor es la persona que usa el teclado y el ratón durante una sesión de programación de pares. El navegador se sienta delante de la computadora al lado del conductor. El propósito de la programación en parejas es mejorar la comunicación, difundir el conocimiento dentro del equipo y garantizar la calidad del código.
- **Integración continua:** Busca integrar continuamente el nuevo código con el código existente en un repositorio de código. Al integrar continuamente las integraciones masivas se pueden evitar tomar mucho tiempo y esfuerzo.
- **Refactorización:** es el proceso de mejorar la estructura interna del software existente sin modificar su comportamiento externo. Con pequeñas mejoras en el código, la refactorización garantiza que el software sea más modificable, extensible y legible.
- **Informar al Cliente:** Busca proporcionar una visión honesta del progreso a los interesados y obteniendo de ellos una retroalimentación sobre las características implementadas y guiar el desarrollo.

Roles a definir en el día de trabajo:

- Equipo de exploración
- Grupos de interesados.

6.2.1. Recapitulaciones

En las sesiones de recapitulación se busca mejorar la comunicación dentro del equipo y medir el progreso de la iteración que se está realizando. Cada día de trabajo comienza con una reunión de recapitulación, donde se decide y discuten las tareas a implementar. Mientras que al finalizar el día de trabajo se lleva a cabo otra reunión de recapitulación para revisar el progreso y las tareas reales realizadas durante el día, aquí también se plantean los problemas existentes y se busca soluciones a los mismos.

La tarea descrita como parte del día de trabajo en este desarrollo no se lleva a cabo dado que la investigación está centrada en mejorar la comunicación dentro de los equipos de desarrollo. Por tratarse de un desarrollo de pequeña envergadura que involucra un solo desarrollador la misma no genera valor agregado al desarrollo del proyecto.

6.2.2. Desarrollo de Componentes Funcionales

Siguiendo con lo descrito en el apartado 6.1.3 en base a las tareas de programación identificadas para cada iteración se lleva a cabo el proceso de desarrollo de funciones que permitan alcanzar cada una de ellas. Para ello se utiliza Android Studio. En este apartado del trabajo se procede a codificar todas las acciones necesarias para satisfacer las tareas propuestas en cada iteración. Se utilizaron los recursos que provee el lenguaje de programación, y el producto de las *tareas de diseño* realizadas para cada iteración en la sección 6.1.3 como ser: la lista clasificada de los pictogramas-acciones, las grabaciones de audio correspondientes, la lógica de navegación entre pantallas, los contenidos de cada una, y la cantidad de pantallas diferentes.

A continuación se presentan detalles inherentes a la funcionalidad que se incorpora en cada iteración. Para ello se consideran como guías las tareas de programación descritas en la tabla 6.2, 6.3 y 6.4 que permiten satisfacer las necesidades de desarrollo. En esta instancia se implementa cada historia de usuario de acuerdo a lo descrito por la figura 6.1 en el *día de planificación*.

Iteración I

Para la iteración I las tareas propuestas para la programación son:

1. Reproducir la grabación de voz asociada a un pictograma seleccionado en la interfaz. El pictograma puede representar una acción general o una específica.
2. Abrir una nueva interfaz (pantalla) al seleccionar una acción general representada por un pictograma. La interfaz desplegada con esta selección debe contener las acciones específicas asociadas a la acción general. Por ejemplo al seleccionar la acción general “*Jugar*” se despliega una pantalla con pictogramas seleccionables que representan diferentes juegos como ser “*pelotas*”, “*muñecos*”, “*bloques*”, etc.
3. Ocultar todas las acciones específicas una vez que cualquiera de ellas sea seleccionada y reproducida. Luego mostrar la interfaz (principal o secundaria) que contiene la acción general que había desplegado las opciones específicas. Es decir la aplicación debe regresar a “*Casa*” o “*Terapia*” dependiendo de donde se encontraba la acción general que desplegó las opciones.

Las librerías principales que se utilizaron para el desarrollo del código fuente de este prototipo fueron las relacionadas con el manejo de *widget* (*Button*, *ImageView*, *TextView*, entre otras), *View*, *WindowManager*, *Bundle*, *Media*, *Intent* y *Context*. Con ellas se incorporaron funciones al desarrollo del prototipo que permitieron el manejo de botones, tablas, vistas de cada interfaz, manejo de parámetros entre las interfaces, manejo de la reproducción de sonido, inicio de nuevas interfaces, y manejo del contexto en cada situación.

Para la distribución visual de los pictogramas en la interfaz se emplearon diferentes controles de flujo de datos, variables y métodos. En este caso se consideraron las dimensiones del dispositivo en el cual se visualizarán las interfaces y, en base a ello, se proporcionaron mediante código las características visuales de los botones que contienen los pictogramas seleccionables, como por ejemplo el tamaño y la distribución dentro de la pantalla. Los recursos como pictogramas, acciones y grabaciones, fueron asociados a los botones que forman parte de la interfaz mediante código.

Como resultado del trabajo de codificación de la iteración I se comenta a continuación la lógica de programación interna del SAC. Sin embargo, antes de ello es necesario aclarar que, por tratarse de un prototipo y fundamentalmente por necesidades inherentes al lenguaje de programación, en el desarrollo se modifican levemente los

espacios entre los contenidos de cada interfaz, afectando su distribución visual pero sin variar su orden ni la funcionalidad principal requerida. La funcionalidad aplicada a cada pictograma se mantiene de acuerdo a lo previsto, es decir respecto a su posición en la interfaz. Los pictogramas visibles que se ubican cercanos a la izquierda superior de la pantalla, son los correspondientes a las acciones de “reproducción”, mientras que los que se encuentran cercanos a la esquina inferior derecha representan acciones de “reproducción y despliegue”.

El sistema incorpora en cada interfaz un conjunto de botones mediante un *control de visualización*. Cada botón se visualiza con el pictograma que tiene asociado y su acción. El *control de visualizaciones* permite el correcto dimensionamiento del tamaño de cada botón y se encarga de seleccionar el conjunto de botones que conforman cualquier interfaz, de manera de asegurar que el contenido de la misma sea el definido en la etapa de planificación.

En el prototipo se manejan 3 tipos de interfaces: *Casa* (principal), *Terapia* (secundaria), y *auxiliares*. Esta última permite visualizar los botones que contienen los pictogramas que representan acciones específicas tras la selección de una acción general.

La selección de un botón en la interfaz principal o secundaria con pictograma asociado a una “reproducción”, ejecuta una salida de sonido que describe el pictograma y queda en espera de una nueva selección. Mientras que la selección de un botón con pictograma asociado “reproducción y despliegue” reproduce el sonido que describe el pictograma, luego despliega una interfaz auxiliar que contiene todas las opciones específicas y, una vez seleccionada una opción, cierra la interfaz.

Habiendo detallado la lógica asociada al prototipo que se desarrolla en la primera iteración, se muestra parte del código de la funcionalidad en el anexo.

Iteración II

Se realizan las siguientes acciones para satisfacer las tareas propuestas para la programación detalladas en la tabla 6.3:

1. Solicitar inicio de sesión al abrir la aplicación, requiriendo el registro de datos personales del usuario (nombre del usuario, género y su edad).
2. Mantener la sesión de usuario iniciada hasta que el usuario indique que desea cerrar su sesión, es decir la aplicación no solicitará reingreso de los datos personales.

3. Mantener iniciada la sesión de usuario por más que la aplicación sea cerrada o el dispositivo se apague.
4. Para cerrar sesión el usuario indica la acción al sistema, este borra los datos personales de la aplicación. Para poder ingresar a la funcionalidad del comunicador, el sistema necesita contar con los datos personales de un usuario.
5. Ofrecer los pictogramas filtrados por el género del usuario.
6. Ofrecer los pictogramas filtrados según la edad del usuario.

En esta tarea, se agregan varios controles al código desarrollado en la iteración I, para satisfacer todas las tareas descritas en la planificación correspondiente, se integran las funcionalidades de las iteraciones I y II al finalizar la etapa de codificación de esta iteración, y se satisfacen los requerimientos planteados en la iteración I y además lo que corresponden a la iteración II.

Se utilizan los recursos provistos como en la planificación de la iteración II, es decir, los nuevos “pictograma-acción”, sus correspondientes grabaciones de voz, la distribución visual de cada pantalla, la cantidad de pantallas y la navegación entre ellas (sección 6.1.3). Se utilizan las mismas librerías de la iteración anterior (I) más las relacionadas con el manejo de la persistencia de los datos, manejo de fechas y opciones para cambios de sesiones como por ejemplo *SharedPreferences*, *Calendar*, *Dialog*, *DatePickerDialog*, *Menú* y *MenuItem*.

El manejo de funciones de creación de botones, de vistas y desplazamientos en esta iteración es igual a la explicitada en la primera, pues se integra la funcionalidad desarrollada nuevos controles para la redistribución y cambios de botones en las vistas y creación de nuevas interfaces. El manejo de desplazamiento varía respecto a la primera iteración, pues la creación de nuevas pantallas agrega y direcciona las conexiones establecidas al desarrollo de esta iteración.

A continuación se presenta el resultado de la codificación, explicando la lógica interna de esta iteración. Se agregan nuevos controles de flujo de datos al código, como así como también clases propias de Android para el manejo y persistencia de los mismos. Se utilizan las funciones propuestas por la librería *SharedPreferences*, que permite dotar al sistema de *memoria* sobre datos específicos mediante la asociación de claves y valores almacenados en un fichero interno del dispositivo. En esta segunda

iteración estas prestaciones se utilizan para que el sistema sea capaz de recordar la sesión del usuario activo, por más que la aplicación sea cerrada por cualquier motivo.

Una de las interfaces desarrolladas e incorporadas en esta iteración es el *login* para el ingreso de datos personales del usuario, permitiendo cargar su identificación personal, fecha de nacimiento y género. En esta pantalla el sistema espera el evento de inicio de sesión mediante el clic del botón correspondiente. Ante la selección del botón de inicio de sesión, el sistema toma la información proporcionada por el usuario, calcula la edad del mismo en base a la fecha de nacimiento ingresada y pasa esta información a la interfaz de *Casa*, encargada de la creación del contenido de la vista y, en este único caso, de asociar el identificador del usuario o nombre al pictograma que lo reproducirá en la pantalla *Casa*.

La visualización de cada una de las interfaces siguientes (principal, secundaria y auxiliares) se delega al gestor de la vista. Éste tiene como tarea decidir la cantidad, el contenido y la distribución de los diferentes pictogramas que se mostrarán en la pantalla, además de proporcionar al usuario la opción de “cerrar sesión” o “cambiar datos” de su registro. El gestor de la vista utiliza la información proporcionada en el *login* del sistema, es decir, determina lo que va a ser visualizado por el usuario, según el género y la edad del usuario contemplados en la planificación de la iteración II. Cada actividad encargada de dar inicio a una nueva pantalla, es la encargada de proveer al gestor de la vista la información extraída del *login* para que dicho gestor pueda determinar lo que se mostrará.

Ante la selección del usuario de “*cerrar sesión*”, el sistema quita los datos personales del usuario del *login* y queda a la espera de que se ingresen nuevos datos. Mientras que el clic sobre “*cambiar datos*”, permite que el usuario reescriba la información proporcionada al sistema sobre sus datos personales. En esta situación el sistema se adecuará a la información ingresada.

Descrita la lógica interna del prototipo en la iteración II, se consigna en el anexo de este trabajo el código fuente correspondiente a las nuevas funcionalidades aportadas al prototipo.

Iteración III

Las tareas propuestas para la programación de la iteración III fueron consignadas en la tabla 6.4, y son las siguientes:

1. Solicitar al inicio de sesión el ingreso de 4 horarios para configurar las actividades para alimentación.
2. Registrar y mantener la hora configurada (*alerta*) por un usuario al iniciar sesión. Es decir, la aplicación no solicitará reingreso de horarios de alerta si la aplicación se cierra o se apaga el dispositivo, manteniendo estos horarios asociados a la sesión activa del usuario.
3. Registrar y controlar el horario del dispositivo cuando la aplicación está siendo usada
4. Controlar que la hora actual del dispositivo se encuentre dentro de un rango determinado para activar un alerta configurada en el inicio de sesión. El rango de activación de un alerta será de 60 minutos posterior al horario configurado para el alerta. Es decir si se configura un alerta a las 07:35, el rango de activación para ella será entre las 07:35 y las 08:35.
5. Indicar qué alerta se encuentra activa.
6. Ofrecer pictogramas de alimentos para comidas principales sólo en la interfaz de *Casa* y cuando un alerta este activa.
7. Ofrecer en la interfaz de *Casa* los pictogramas de alimentos acordes a la alerta activa. Es decir si está activo un alerta de almuerzo, ofrecer opciones para almorzar.

Las funcionalidades planteadas en las iteraciones I y II se mantienen, y se integra la funcionalidad del sistema de adaptarse al usuario, según la tarea que éste se encuentre realizando y en un determinado momento del día, lo que implica la incorporación de *sensibilidad al contexto* en el desarrollo. Ello requiere el uso de una lógica algo diferente a la utilizada en las iteraciones anteriores. Así planteado, se aspira a lograr un sistema consiente del contexto en el que está funcionando. Para ello se propone el uso de un sensor que puede identificar la presencia de determinada condición o variable del entorno. En esta iteración la variable de interés del entorno es el *tiempo*, la unidad de medida de esta variable está regida por la necesidad de precisión del sistema, en este caso, segundos.

Considerando lo anterior, la característica del código de esta iteración es la declaración de un servicio que cumple la función de censar el valor que toma la variable *tiempo* a medida que éste transcurre. Este servicio se ejecuta en un hilo paralelamente al resto de las funcionalidades del sistema, es decir controla el paso del tiempo

permanentemente, arrojando los diferentes valores de la variable que son evaluados por el servicio mediante una condición. La condición que controla el sistema determina la activación o no de las alertas configuradas por el usuario en el inicio de sesión. Para implementar esta condición el sistema controla el valor obtenido del censado con el valor definido por el usuario como valor del alerta. Cuando estos valores coinciden y el usuario se encuentra ejecutando acciones correspondientes a la interfaz de Casa, se despliega una notificación que le permite al usuario acceder a los pictogramas correspondientes con la actividad de alimentación programada.

Para capturar el valor de configuración de las alertas, el sistema implementa una serie de controles en el *login* que permiten al usuario registrar horarios. Estos horarios configurados se enlazan con la sesión iniciada, permitiendo que la configuración se mantenga aunque se cierre la aplicación o el dispositivo se reinicie. Para ello se utilizan los recursos provistos por la clase de *SharedPreferences*. Una vez que el usuario define los horarios de alerta para cada actividad de alimentación, e indica al sistema la acción de iniciar sesión, entonces se comienza a censar el estado de la variable *tiempo* desde el servicio. Cuando se cumple la condición para que el servicio inicie la notificación, el usuario puede acceder durante 60 minutos a los pictogramas referentes a los diferentes alimentos. Luego de cumplido el tiempo de duración, el alerta deja de estar disponible como así también los pictogramas correspondientes. El servicio se encarga de controlar el valor de la variable tiempo, activando y desactivando las alertas correspondientes con la sesión activa. Mientras no haya ningún alerta activa, el sistema tiene las funciones descritas para la iteración II.

En esta iteración se utilizan las librerías *TimePickerDialog* y *Service*, los resultados de las tareas de diseño para la iteración III, es decir lógica de navegación, pictogramas, grabaciones, así como también los productos de la iteración anterior como el código fuente, navegación, pictogramas y grabaciones. Al finalizar el desarrollo de esta iteración se satisfacen los requerimientos planteados por la iteración I, II y III respectivamente. El código fuente que permite obtener la nueva funcionalidad propuesta por esta iteración se encuentra disponible en el anexo del trabajo.

6.2.3. Informar al Cliente

En esta tarea se pretende proporcionar una visión del progreso al cliente dándole la posibilidad de proveer una retroalimentación sobre las características implementadas

en el desarrollo. Se utilizan informes que reflejen el estado real de las tareas previstas y el grado de cumplimiento del cronograma planificado. Dado el carácter académico y la pequeña envergadura de este desarrollo, los informes de avance de cada iteración se realizaron de manera informal

6.3 DÍA DE LANZAMIENTO

La etapa de *día de lanzamiento* se identifica como “*tipo: patrón de etapa de clasificación esencial*” dentro del método. Se la presenta con el siguiente resumen:

Motivación del día de lanzamiento:

- Liberar el funcionamiento del sistema en pleno desarrollo.

Metas del día de lanzamiento:

- Garantizar que todos los subsistemas se integren con éxito en un único sistema.
- Ejecutar las pruebas de aceptación para verificar que los requerimientos se implementan correctamente.
- Elaborar una lista de nuevos requerimientos y defectos en el software.
- Proporcionar al cliente y al equipo de desarrollo la oportunidad de discutir cómo debe funcionar el software.
- Asegurar que el equipo ha realizado todas las actividades pertinentes antes de hacer una liberación de software (mediante la realización de una auditoría de liberación).
- Asegurar la base para un mayor desarrollo (mediante la creación de una línea de base).

Entrada a la etapa del día de lanzamiento:

- Funcionalidad implementada
- Pruebas de aceptación y documentación.

Salida de la etapa del día de lanzamiento:

- Lista actualizada de defectos,

- Lista de verificación de la liberación de la liberación llena,
- Línea de base (por ejemplo, en el sistema de control de versiones)
- Funcionalidad implementada,
- Documento de requerimientos iniciales.
- Producto integrado operativo.

Tareas específicas a realizar en el día de lanzamiento:

- Integración de sistema: En el caso de un proyecto multi-equipos, se integran en un solo producto los subsistemas generados en equipos separados.
- Pruebas de Pre-Liberación: se asegura que el software que se está produciendo está listo para la prueba de aceptación y la liberación. Las pruebas previas a la liberación se llevan a cabo dentro del equipo ejecutando todas las pruebas de aceptación por escrito.
- Pruebas de aceptación: se verifica que los requerimientos establecidos para el software se implementen correctamente. Durante el día de lanzamiento se realizan pruebas de aceptación con los interesados y se documentan los problemas encontrados.
- Ceremonia de lanzamiento: forma parte de las tareas finales antes de hacer una versión del software. En la práctica, consiste en dos actividades esenciales; por un lado están auditorías de lanzamiento y por otro la creación de línea de base. Ambas se realizan para confirmar que todo se ha hecho correctamente en la iteración actual y se garantiza la base para un mayor desarrollo.

Roles a definir en el día de lanzamiento:

- Equipo de exploración
- Grupos de interesados.

6.3.1. Integración del Sistema y Pruebas Previas a la Liberación

La tarea correspondiente a la *integración del sistema* pretende unificar con éxito, es decir libre de errores y garantizando un correcto funcionamiento, un producto desarrollado por partes o subsistemas por diferentes integrantes del equipo.

Dado que esta tarea encuentra aplicación para equipos de desarrollo más importantes, en el presente trabajo la integración se realizó en la etapa de día de trabajo, proponiendo un prototipo incremental, y los desarrollos de cada iteración tuvieron en cuenta las bases y códigos de las iteraciones precedentes. Como se consignó en la presentación de la etapa, la tarea de *pruebas previas a la liberación* pretende asegurar que el software producido no se presente con defectos al cliente para que éste realice las pruebas correspondientes. En esta investigación también se realizó esta tarea en la etapa de día de trabajo dado que el trabajo de codificación se desarrolla por una sola persona.

6.3.2. Pruebas de Aceptación

En esta tarea se verifica que los requerimientos del sistema se implementen correctamente, para ello se ejecutan las pruebas de aceptación y se actualiza la lista de requerimientos, así como también se listan los defectos detectados en el funcionamiento del software. Para ello, se cuenta con el código fuente de desarrollo y con las pruebas correspondientes para cada iteración. La ejecución de las pruebas de aceptación permite verificar la satisfacción de cada HU correspondiente a cada iteración probada, esto queda reflejado en las tarjetas de HU consignadas en el anexo del trabajo.

A medida que se fue completando cada una de las iteraciones, se fue validando la funcionalidad real implementada comparándola con la funcionalidad deseada para cada iteración. Se presentaron los ejecutables²³ de los diferentes prototipos a los interesados para su valoración. Para las pruebas se utilizó una tableta cuyas características técnicas se detallaron en la sección 5.1.1, con sistema operativo Android. Además, se instaló el software en un dispositivo diferente, un teléfono con sistema operativo Android, para verificar que el diseño de la interfaz se adaptara a los diferentes tamaños de pantalla y que no estuviere ligado a las dimensiones del móvil en el que se ejecuta.

Primero se comprobó la lógica del prototipo por cada iteración en cuanto a desarrollo (pruebas previas a la liberación), luego fue probado por los profesionales que interactuaron con la aplicación y validaron las funciones requeridas por iteración, registrando sus observaciones en las planillas denominadas *pruebas de aceptación*,

²³ Disponible en <http://elcomunicadoralternativo.com>

dando lugar al fundamento para describir nuevas funciones a incorporar en el sistema final en trabajos futuros. Estas pruebas se consignan en las tablas 6.7 a 6.9.

PLANILLA DE PRUEBA DE ACEPTACIÓN – ITERACIÓN I				
Fecha de escritura: Octubre de 2016				
Fecha de ejecución: Noviembre de 2016				
ID	{Qué funcionalidad debe implementar el sistema para ser validado}	RESULTADOS		
		Pasa	No pasa	Observaciones sobre la funcionalidad
IRF8	Al seleccionar un icono que representa una acción se reproduce automáticamente de forma sonora en el dispositivo.	X		
IRF9	La reproducción sonora de cada acción es comprendida sin necesidad de aclaraciones sobre su significado en nuestra cultura.	X		
IRF1	Las acciones consideradas como parte de la interfaz representan necesidades de comunicación posibles para un niño.	X		
IIRF6	Al iniciar la sesión de usuario la interfaz principal corresponde a <i>Casa</i> . Cualquier acción que se ejecute en esta interfaz debe retornar siempre a la interfaz de <i>Casa</i> , excepto cuando el usuario indique que va a entrar en <i>terapia</i> , mediante la selección de la imagen representativa.	X		
	Cuando el usuario le indica a la aplicación que entra en <i>Terapia</i> , la interfaz ofrecida es diferente a la interfaz de <i>Casa</i> . Cualquier acción ejecutada dentro de esta interfaz retorna siempre a <i>Terapia</i> , excepto cuando el usuario indica que regresa a su <i>Casa</i> . En este caso la aplicación retorna a <i>Casa</i> .	X		
<p>Observaciones sobre la interacción: i) Algunas acciones se pueden expresar de manera más comprensible. Por ejemplo, en vez de “poner” o “sacar”, sería mejor poder especificar <i>qué es lo que se puede poner o sacar</i>. La misma observación sobre “subir” y “bajar”. ii) Al seleccionar “yo me llamo” debería proveerse la opción de indicar <i>cómo te llamas</i>, sino queda una sensación de vacío e incompleto. iii) Se podría agregar también la opción de decir cuántos años tiene el usuario. iv) En cuanto a los pictogramas que tienen opciones, la funcionalidad es buena y su contenido también, solo que el pictograma de “cambiar el pañal” suena fuera de contexto si quien usa el sistema fuese un niño que controla sus esfínteres. Quizás sea mejor poner “<i>ir al baño y cambiar el pañal</i>” en un solo pictograma seleccionable que reproduzca “<i>quiero ir al baño</i>”. v) Ampliar las opciones en los juegos.</p>				

Tabla 6-7: Validación de las funcionalidades del prototipo de la iteración I

PLANILLA DE PRUEBA DE ACEPTACIÓN – ITERACIÓN II				
Fecha de escritura: Noviembre de 2016				
Fecha de ejecución: Diciembre de 2016				
ID	{Qué funcionalidad debe implementar el sistema para ser validado}	RESULTADOS		
		Pasa	No pasa	Observaciones sobre la funcionalidad
IIRF 10	Los pictogramas visibles en la interfaz de género femenino son diferentes a los presentados al masculino.	X		La reproducción por género está en voz femenina. Es deseable que para los varones se reproduzcan voces de varón.
	Los pictogramas disponibles en las interfaces varían si el usuario es un niño pequeño (por ejemplo 2 años), o mayor (por ejemplo 5 o 9 años).	X		Debería existir diferencias entre lo que se muestra a niños de 7 y a niños de 9, lo mismo con los de 4 y 5.
IIRG 2	La funcionalidad del SAC es accesible con el registro de los datos personales del usuario, permitiendo iniciar una sesión al sistema.	X		
	La sesión de un usuario iniciado permanece activa hasta que éste indica lo contrario.	X		
	En caso de que se cierre la aplicación voluntariamente o se apague el dispositivo, la sesión permanece abierta sin necesidad de reingreso de datos personales.	X		
	En cualquier instancia durante el funcionamiento de la aplicación el usuario puede cambiar datos de su registro o salir de su sesión.	X		
<p>Observaciones sobre la interacción: i) Los pictogramas “subir”, “bajar”, “poner”, “sacar”, “acercar”, “alejar” son poco funcionales, incorporar acciones específicas sobre lo que pueden realizar con ellas. ii) La reproducción de sonidos es repetitiva para los pictogramas que presentan opciones intermedias. Grabar la voz con la frase completa. iii) No permitir el ingreso de número al escribir el nombre del usuario. iv) Para poder utilizar las letras del abecedario, el usuario no debería repetir todo el circuito por letra, sino que debería permanecer en la pantalla de “abecedario” e indicarle al sistema cuando no quiere estar más ahí, de la misma manera que hace con la “terapia” y “casa”. Lo mismo con los “colores”, “animales” y “números”. v) Los pictogramas de “números” deben tener un tamaño más grande, estar más espaciados o una forma de destacar el botón para diferenciar bien cada uno de los números. vi) En “terapia” incorporar la opción de “preguntas”, así como existe para “casa”. vii) En las opciones “Leer”, “cambiarme”, “escuchar música” o “escuchar un cuento” incorporar opciones, como por ejemplo los diferentes cuentos. viii) Cuando se “cierra sesión”, y quiere ingresar nuevamente al sistema, sería bueno que se pueda ingresar solo un nombre o alias como lo que ocurre con las cuentas de correo una vez que ya has iniciado sesión una vez antes.</p>				

Tabla 6-8: Validación de las funcionalidades del prototipo en la iteración II

Tabla 6-8: Validación de las funcionalidades del prototipo en la iteración II

PLANILLA DE PRUEBA DE ACEPTACIÓN – ITERACIÓN III				
Fecha de escritura: Diciembre de 2016				
Fecha de ejecución: Febrero de 2017				
ID	{Qué funcionalidad debe implementar el sistema para ser validado}	RESULTADOS		
		Pasa	No pasa	Observaciones sobre la funcionalidad
IIIRF4	La aplicación determina que el horario del dispositivo coincide con un horario configurado como alerta y que el usuario interactúa con el sistema, entonces activa el alerta correspondiente.	X		
	Cuando un alerta esté activo el usuario tendrá acceso (en la interfaz de <i>Casa</i>) a pictogramas que representan opciones de alimentos para esa alerta.	X		
IIIRF5	El usuario puede registrar diferentes horarios de alertas relacionados con sus comidas principales (desayuno, almuerzo, merienda y cena).	X		**
	El registro de las alertas permanecerá relacionado a su sesión de usuario. En caso que la aplicación se cierre o el dispositivo se apague (sin cerrar sesión) los horarios configurados serán persistentes en su registro.	X		
IIIRF7	La aplicación detecta que el usuario está interactuando con la interfaz de Casa, y el horario configurado para un alerta coincide con el horario del dispositivo, entonces se ofrece al usuario opciones relacionadas a la alimentación programada en su sesión	X		
Observaciones sobre la interacción: **Puede configurar cualquier horario, hasta horarios ilógicos para un desayuno por ejemplo. Además se pueden poner todas las alertas en el mismo horario, eso no debería suceder. El sistema debería proponer un rango para una alerta otro rango para otra, de manera que no se puedan superponer los horarios configurados.				

Tabla 6-9: Validación de las funcionalidades del prototipo en la iteración III

A partir de las planillas de prueba de aceptación, se identifica una la lista de defectos para la iteración considerada. En la tabla 6.10 se presentan todos los defectos encontrados, registrando los siguientes datos:

- Número: número creciente para la identificación del defecto.
- Severidad: Gravedad del defecto, se clasifica en:
 - *Mayor*: tiene mucho impacto pero posee soluciones provisorias; requiere de gran esfuerzo para corregirlo, pero tiene menor impacto en el cronograma.
 - *Menor*: defecto aislado que se manifiesta raramente y que tiene poco impacto.
 - *Cosmético*: pequeños errores sin impacto en el funcionamiento correcto.
- Descripción: descripción del defecto.
- Fecha encontrada: fecha en que se encontró el defecto.
- N° de iteración donde se encontró.
- Observaciones: Información adicional.

Es importante notar que los defectos que no sean corregidos en el presente prototipo, constituirán la base para el desarrollo de futuras iteraciones.

LISTA DE DEFECTOS					
Nota: esta plantilla se utiliza para registrar todos los defectos encontrados en las diferentes iteraciones probadas por los interesados					
Nº	Severidad	Descripción	Fecha	Iteración	Observaciones
1	Cosmético	Insertar más opciones al seleccionar pictogramas	11/16 y 12/16	I y II	Sacar, poner, subir, bajar, alejar, acerar, leer, escuchar música, escuchar cuento, cambiarme, juegos
2	Menor	Reproducción del pictograma “yo me llamo”	11/16	I	Incorporar la opción de reproducción del nombre por el sistema
3	Cosmético	Reproducción de la edad del usuario	11/16	I	Incorporar la opción de reproducción de la edad por el sistema
4	Menor	Reproducción de pictograma “cambiarme el pañal”	11/16	I	Agregar “ir al baño” para los usuarios que ya no usan pañal
6	Cosmético	Reproducción sonora en género masculino	12/16	II	Agregar grabaciones con reproducción para varón
7	Menor	Falta de adaptación en interfaces según edades	12/16	II	Se consideraron rangos de adaptación por ser un prototipo, no edades específicas
8	Cosmético	Reproducción de pictogramas de “reproducción y desplazamiento” repetitiva.	12/16	II	Reproducir solo los pictogramas que no implican desplazamientos de interfaz, regrabar las frases asociadas para que sean coherentes con el camino que llevo la selección.
9	Cosmético	Reproducción de conocimientos adquiridos	12/16	II	Considerar nuevo estado del usuario para permanecer en la pantalla de números por ejemplo, hasta que el indique que quiere salir (igual que se realiza en terapia y casa)

Tabla 6-10: *Listado de defectos encontrados*

LISTA DE DEFECTOS					
Nota: esta plantilla se utiliza para registrar todos los defectos encontrados en las diferentes iteraciones probadas por los interesados					
Nº	Severidad	Descripción	Fecha	Iteración	Observaciones
10	Menor	Cercanía de los pictogramas, y poca definición del límite de cada uno	12/16	II	En los números y letras resulta difícil diferenciar dónde comienza y termina un pictograma. (Revisar el código para mejorar la distribución de los pictogramas para cada pantalla, sin cambiar su contenido)
11	Cosmético	Falta la opción de preguntas en terapia	12/16	II	Todas las terapias contienen la interfaz igual, aunque los desplazamientos pueden variar. Adaptar las preguntas en la pantalla igual que en <i>casa</i>
12	Menor	Reingreso de datos de registro luego de cierre de sesión	12/16	II	Al iniciar sesión generar un id de usuario que asocie todos los datos de registro ingresados la primera vez. Para que luego de cerrar sesión solo ingrese el id y se inicie el sistema.
13	Menor	Ingreso de nombre de usuario	12/16 y 02/17	II y III	Insertar un control en el ingreso del nombre que sólo permita ingresar letras controlando la longitud de las palabras ingresadas.
14	Menor	Ingreso de alarmas iguales	02/17	III	Insertar un control que no permita ingresar alarmas en el mismo horario de otras definidas.
15	Cosmético	Aumentar el número de alertas configurables	02/17	III	Permitir ingresar la cantidad de alertas que desee el usuario, y agregar una configuración para definir que quiere sugerir al usuario.

Tabla 6.10. *Listado de defectos encontrados (continuación)*

A partir de las observaciones obtenidas, es posible redefinir las necesidades del sistema, estableciendo requerimientos pos iteraciones que se detallan en la tabla 6.11.

Requerimientos “fase de inicialización”	Requerimientos Pos iteraciones
El SAC reproducirá en forma sonora una acción cuando el usuario seleccione un ícono de la interfaz	Reproducir en forma sonora un pictograma seleccionado en la interfaz, sólo si es de “reproducción”.
	La salida sonora asociada a la selección de un pictograma debe ser una frase completa con sentido, no sólo objetos o palabras sueltas.
	Permanecer en cada pantalla donde haya solo pictogramas de “reproducción”, hasta que el usuario le indique salir de ella.
El SAC deberá diferenciar el ofrecimiento de las imágenes seleccionable dependiendo de si el usuario se encuentra en la <i>casa</i> o la <i>terapia</i>	Adaptar el ofrecimiento de pictogramas cuando se encuentra en <i>Terapia/Casa</i> dependiendo su edad, su género y las actividades o acciones posibles para cada situación.
El SAC mostrará imágenes o pictogramas según el género y el rango de edad del usuario	Emitir la salida sonora con una reproducción de voz acorde al género del usuario.
El SAC deberá permitir registrar a cualquier paciente que concurre al IPRI con PCI leve o moderada.	Generar un identificador único para cada usuario registrado.
	Permitir el acceso a sus funcionalidades mediante un identificador de usuario.
	No solicitar el reingreso de datos personales de un usuario que se haya registrado.
El SAC deberá ofrecer las imágenes seleccionables al usuario acordes al contexto temporal (hora del día)	Admitir el ingreso de diferentes alertas configurables por el usuario
	Permitir que el usuario determine los pictogramas que corresponderán a cada alerta, en base a una lista de pictogramas opcionales
El SAC permitirá la configuración personal de horarios en los que el usuario realice determinadas actividades	Permitir que el usuario determine la cantidad de alertas que se ejecuten en su sesión, al igual que el contenido que se mostrará en cada una.

Tabla 6-11: *Requerimientos pos iteraciones para trabajos futuros*

6.3.3. Ceremonia de Lanzamiento

Esta es una tarea para confirmar que todo se ha hecho correctamente en la iteración actual y se garantiza la base para un mayor desarrollo.

Dado que el alcance de este trabajo es la elaboración de un prototipo, se puede decir que el desarrollo hasta la iteración III está aprobado, constituyendo la línea de base. Se entiende que el desarrollo de la próxima iteración debería agrupar los nuevos requerimientos, tratando de satisfacer aquellos que sean más sencillos de solucionar desde el punto de vista de la programación.

La afirmación de establecer el prototipo resultante de las integraciones de las iteraciones I, II y III se basa en que los interesados consignaron las siguientes ventajas del uso de la aplicación:

- Los *pictogramas* utilizados son fácilmente interpretables para la todas las edades y géneros.
- La *cantidad de pictogramas* por pantalla es correcta, no satura al usuario con opciones, además permite que se puedan comunicar necesidades básicas desde la pantalla principal.
- La *reproducción* de los pictogramas es comprensible para la forma de comunicación social en la que interactúan los usuarios.
- La *agrupación* de pictograma-palabra es intuitiva por lo cual se puede acceder a una variedad de pictogramas desde la pantalla principal, situación que con los SAC manuales resulta compleja.
- La *organización* de los pictogramas que abren opciones es adecuada, pues se encuentran distribuidos en la parte inferior de la pantalla quedando siempre cercanos al usuario y reduciendo los desplazamientos necesarios para su selección.
- Los *movimientos* necesarios para reproducir una necesidad son mínimos, lo cual favorece la comunicación fluida.
- La *configuración de perfil* es apropiada dependiendo del tipo de usuario, ya que incorpora nuevas funciones de comunicación en la medida que se identifica un usuario de mayor edad.

- El *inicio de sesión* para diferentes usuarios, en un mismo dispositivo, sin necesidad de acceder a internet, es beneficioso para los terapeutas porque pueden utilizar la aplicación personalizada para cada paciente en el mismo equipo móvil.
- El uso de *alertas* es una ventaja importante, pues permite no tener cargada la pantalla con pictogramas que no serán ocupados en todo momento, sino que solo tendrán función para un rango de horario específico.
- La configuración de alertas personales aporta una funcionalidad innovadora y muy útil, porque permite que el usuario elija el momento del día en que desea tener disponibles los pictogramas en la pantalla.
- Los pictogramas de las *pantallas de alertas*, son acordes a las necesidades de un niño y las posibilidades de decisión que pudiera tener, es interesante que pueda seleccionar alimentos propios de la cultura santiagueña.

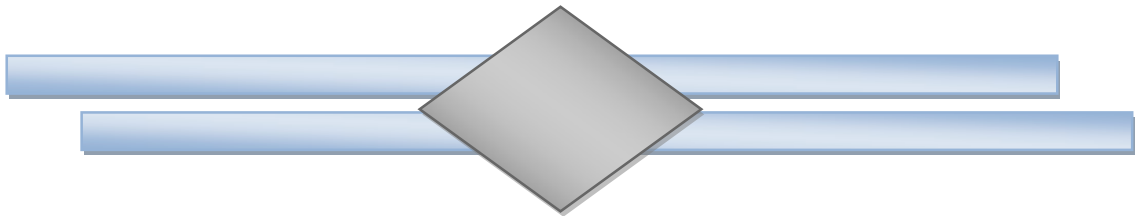
Se procede a dejar a disposición de los interesados el prototipo de SAC desarrollado con fines académico, para el mismo pueda ser usado en las sesiones de terapia con los pacientes del IPRI.

6.4 CONCLUSIÓN

En este capítulo se ha desarrollado la *Fase de Producción* propuesta por el método Mobile-D, obteniendo la implementación de la funcionalidad de cada producto según la Fase de inicialización, así como también la documentación de las pruebas de aceptación realizadas con el grupo de interesados. Esta documentación permitió definir una lista de defectos para cada producto probado. Asimismo, en esta fase se obtuvieron una serie de nuevos requerimientos que podrán ser implementados en trabajos futuros.

Según los resultados obtenidos en las planillas de pruebas de aceptación de cada iteración y los beneficios detectados por los interesados, el prototipo de SAC desarrollado tras tres iteraciones satisface los requerimientos planteados inicialmente. Quedan por resolver cuestiones de estética, que por tratarse de una prototipación se tendrán en cuenta en futuras iteraciones. En la lista de defectos se dejan indicados algunos refinamientos finales para librar al prototipo de notificaciones de errores internos que no afectan a la funcionalidad general, pero que pueden distraer la atención del usuario al momento de la ejecución.

**Capítulo 7 -
Conclusiones y
Trabajo Futuro**



En este trabajo se presentó el desarrollo de un prototipo de SAC que funciona en dispositivos móviles que poseen SO Android, destinado a apoyar la comunicación de los niños con PCI leve a moderada que asisten al IPRI. Para ello en los primeros capítulos se realizó una introducción a los conceptos que dan soporte a la temática de desarrollo propuesta. Esto implicó indagar en cuestiones teóricas referidas a los SAC y a la PCI, pues algunas de las secuelas que apareja la PCI en los niños comprometen la comunicación y la reducción en la movilidad, requiriendo el manejo de SAC para interactuar con el entorno.

Asimismo, se precisaron cuestiones técnicas del área informática como lo son las aplicaciones móviles, el SO Android, la sensibilidad al contexto y el método Mobile-D, que permitieron perfilar el desarrollo. Más precisamente se tomaron en cuenta algunos rasgos tales como el tipo de aplicación a desarrollar, la estructura interna del SO, los aspectos a tener en cuenta para la definición del contexto y las variables de interés. Además el método Mobile-D sustentó la organización y desglose de las diferentes tareas necesarias adaptándose de manera adecuada para el desarrollo de este trabajo.

Seguidamente, en el capítulo 3 se ahondó sobre los SAC y particularmente sobre los de clasificación manual, detallando sus diferentes recursos, complejidad y el funcionamiento general. Se describió el proceso de comunicación apoyado en SAC manuales, determinando que el mismo se ralentiza por la necesidad de intervención casi permanente de una persona para asistir al niño durante el uso del SAC. Posteriormente se puntualizaron las características deseables en un SAC, reforzando la necesidad de integrar las funcionalidades de los SAC manuales en una aplicación móvil, para reducir la asistencia necesaria en la comunicación.

En cuanto al desarrollo de la aplicación, en el capítulo 4 se inició la primera fase del método Mobile-D, denominada fase de exploración, describiendo el alcance del proyecto, determinando el grupo de interesados, y redactando el documento de requerimientos iniciales. Con ello se propuso una definición general sobre la arquitectura del software y el plan de proyecto para el desarrollo del SAC. El método organiza las fases de manera secuencial, de manera que los resultados obtenidos en esta fase de exploración sustentaron las fases posteriores.

A continuación, en el capítulo 5 o fase de inicialización, se expresó la funcionalidad deseada por los interesados respecto al SAC en 4 historias de usuario.

Con base en ellas se diseñó e implementó un prototipo funcional, cuyo diseño fue sometido a consideración de los interesados. Como resultado se obtuvieron aportes importantes útiles para refinar los requerimientos descritos en la fase anterior, permitiendo la actualización del documento de requerimientos. En función a este nuevo documento se dio origen a una pila de productos que se desarrollaron mediante iteraciones incrementales en el capítulo 6.

El capítulo 6 se efectuó la fase final aplicada para este desarrollo, o fase de producción, implementando las funcionalidades propuestas en la fase anterior. Para ello se especificaron cada una de las iteraciones a considerar, a fin de obtener un prototipo con las características deseadas por los interesados. Se describieron las tareas necesarias de diseño y de programación, así como también las planillas de pruebas de aceptación para cada iteración. Como resultado de la ejecución de las tareas de diseño se obtuvieron los recursos gráficos que forman parte de las interfaces, así como también el contenido y la distribución en cada interfaz, lo cual permitió definir los desplazamientos entre ellas. Con base en ello y en la ejecución de tareas de programación se implementaron las funcionalidades necesarias para cada iteración, mediante la programación en Android Studio. Como resultado de la programación se obtuvieron tres ejecutables o productos de cada iteración, los cuales se sometieron a prueba con los interesados. Así, se logró contrastar la funcionalidad deseada con la obtenida, se documentó esta actividad con las planillas pruebas de aceptación, la lista de defectos para cada producto probado y una lista de nuevos requerimientos detectados.

Como resultado de esta fase se concluyó que el prototipo final desarrollado cumple de manera general con las expectativas de funcionamiento propuestas por los interesados, si es utilizado por niños con disminución motriz leve o moderada. Los resultados documentados en este capítulo plasman necesidades de mejora para que el SAC pueda ser utilizado en casos más severos.

Se puede concluir que el análisis de generalidad de las palabras y la organización dentro de la interfaz, según las necesidades posibles y dependiendo del lugar donde se encuentre el niño, permiten reducir los desplazamientos necesarios para comunicar una determinada necesidad, disminuyendo así el tiempo invertido y la asistencia externa de un tutor para comunicarse de manera fluida con el entorno social próximo.

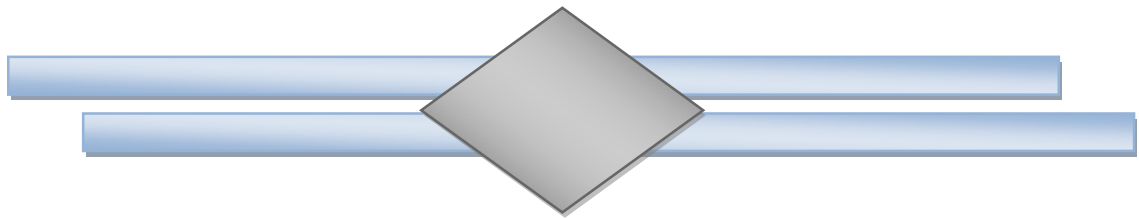
A continuación se describen algunas posibles líneas de trabajo futuro:

- Incorporación de sensibilidad contextual respecto a la ubicación del usuario. En

el presente trabajo el usuario debe indicar al prototipo si se encuentra en la *casa* en la *terapia*, con una acción concreta. En función a ello, el SAC se mantiene en la interfaz hasta que el usuario indique lo contrario. Esta línea de investigación permitiría, mediante la incorporación de un sensor, dotar a la aplicación de la capacidad de determinar en qué lugar se encuentra el usuario en determinado momento y en función a ello adaptar su interfaz.

- Definir una función de asistencia remota, para que si el usuario se encuentra lejos de su tutor y necesitara la asistencia urgente de éste último, el sistema se encargue de enviar un alerta al dispositivo móvil del tutor para avisarle de la situación.
- Desarrollo de modulo de adaptación de switch viacam o pulsador, que permitiera seleccionar y pulsar dentro de la interfaz del dispositivo móvil donde se encuentra el SAC, ya que actualmente el prototipo solo registra la selección y pulsado mediante la pantalla táctil. La incorporación de este modulo permitiría ampliar el tipo de usuarios de la aplicación a aquellos con motricidad aun más reducida.

Bibliografía

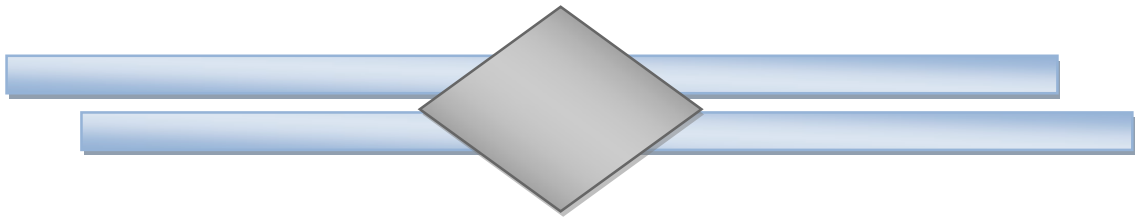


- Abadín, A., Delgado Santos, C. & Vigara Cerrato, A. (2010) Comunicación Aumentativa y Alternativa. Guía de Referencia. Edición CEPAT
- Alonso, M. Á. V., Modinos, F. S., & Sánchez, L. E. G (2009). Bases para el desarrollo de un modelo de evaluación para personas con discapacidad intelectual adultas que viven en servicios residenciales
- Argüelles, P. P. (2008). Parálisis Cerebral Infantil. Protocolos diagnóstico terapéuticos de la AEP: Neurología Pediátrica.
- Baldassarri, S., Peña, P., Cerezo, E., & Marco, J. (2014) Estado del arte en sistemas de comunicación alternativa y aumentativa. Revista AUTI: Aplicaciones y Usabilidad de Televisión digital Interactiva. 1(1), 10, 16. Consultado En Mayo 2016 En <http://docplayer.es/1633985-estado-del-arte-en-sistemas-de-comunicacion-alternativa-y-aumentativa.html>
- Barranco Mesa, J. (2012). SML Android Satélite.
- Baumgart, D., Johnson, J., & Helmstetter, E. (1996). Sistemas Alternativos De Comunicación para personas con discapacidad. Alianza Editorial.
- Beltrán, D., De La Caridad López, M., Contreras, J., Quintana, D., Fuentes, L., Hernández, O., & Morales, E. (2013). Histidinemia en niños preescolares con trastornos de la comunicación oral. Revista de investigación en Logopedia, 3(2), 146-162.
- Boehm, B. (2007). A Survey Of Agile Development Methodologies. Laurie Williams.
- Calderón, R. S. (2012). El Síndrome Autista: un acercamiento a sus características y generalidades. Revista Educación, 26(1), 47-61.
- Cataldi, Z., & Lage, F. J. (2012). TICS En educación: nuevas herramientas y nuevos paradigmas. In VII Congreso de Tecnología en Educación y Educación en Tecnología.
- Chin, E., Felt, A. P., Greenwood, K., & Wagner, D. (2011, June). Analyzing Inter-Application Communication In Android. In proceedings of the 9th international conference on mobile systems, applications, and services (Pp. 239-252). ACM.
- Crespo-Eguílaz, N. & Narbona J. (2003). Perfiles clínicos evolutivos y transiciones en el espectro del trastorno específico del desarrollo del lenguaje. Revue Neurologique, 36 (Supl 1): S29-S35.
- Escobar, D. P. O., & Gualteros, A. C. (2013). Estudio de metodologías ágiles para proyectos de software en corto tiempo. Tecnología Investigación Y Academia, 1(2).
- Fábrega, J., & Patricia, A. (2014). Sistemas Alternativos y Aumentativos de Comunicación
- Fernández, Á., Sánchez, J. L. G., Roldán, L. M., Rodríguez-Fórtiz, M. J., Hurtado-Torres, M. V., & Medina-Medina, N. (2009). Generador Sc@ ut: Sistema De Creación de Comunicadores Personalizados para la Integración. Revista Iberoamericana De Tecnologías del/da Aprendizaje/Aprendizagem, 199.
- Ga, D., Djuric, D., & Deved, V. (2006). Model Driven Architecture And Ontology Development. Springer Science & Business Media.

- Galico, D., Natanzon, K., Vega, C., Matalonga, S., & Solari, M. (2015). Software Sensible al Contexto: Definiciones y Desarrollo de un Estudio de Caso en Google Glass.
- Gallego, M. T. (2012). Metodología SCRUM. Gestión de proyectos informáticos, <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigastfc0612memoria.Pdf>.
- García Rodríguez, M. J. (2015). Estudio Comparativo entre las Metodologías Ágiles y las Metodologías Tradicionales Para la Gestión de Proyectos Software.
- González, M., Sosa, H., & Martín, A. (2014). Sistemas De Comunicación No Verbales. Informes Científicos-Técnicos UNPA, 6(2), 30-56.
- Joskowicz, J. (2008). Reglas y prácticas en eXtreme Programming. Universidad De Vigo, 22
- Korkea-Aho, M. (2000). Context-Aware Applications Survey. Department of Computer Science, Helsinki University of Technology.
- Loke, S. (2006). Context-Aware Pervasive Systems: Architectures for a new breed of Applications. CRC Press.
- Méndez Martín, J. (2015). Aplicación Android de Apoyo a Fisioterapeutas.
- Najar Ruiz, P. (2014). Prototipo De Sistema Móvil Para E-Turismo. Trabajo final de Licenciatura en Sistemas de Información, Universidad Nacional De Santiago Del Estero.
- Ortí, C. B. (2002)Tecnologías de Ayuda: Sistemas Alternativos de Comunicación.
- Polanco, K. M., & Taibo, J. L. B. (2011). " Android" El Sistema Operativo de Google para dispositivos móviles.
- Pressman, R. S., & Troya, J. M. (1988). Ingeniería Del Software (No. 001.64 P74s.). McGraw Hill.
- Quishpe Pila, J. P. (2013). Automatización del seguimiento de proyectos y contratos realizados en el Instituto Superior de Investigaciones (ISI), En la facultad de ingeniería en geología, minas, petróleos y ambiental.
- Restrepo Lenis, V. E. (2016). Prototipo de Aplicación Móvil para el corporativo Peña y Peña abogados
- Robaina-Castellanos, G. R., Riesgo-Rodríguez, S., & Robaina-Castellanos, M. S. (2007). Definición y Clasificación de la Parálisis Cerebral:¿Un problema ya resuelto?. *Revneurolog*, 45(2), 110-7.
- Robert J. Bartz, 2015; Mobile computing deployment and management: real world skills for CompTIA mobility+ certification and beyond. John Wiley & Sons.
- Rodríguez Durán, D. A., & Ugalde Briones, O. J. (2012). Evaluación del uso de la redsocial twitter por parte de los estudiantes de la facultad de comunicación social de la universidad de Guayaquil para la creación de un Programa de Estudio sobre manejo (Bachelor's Thesis, Universidad De Guayaquil. Facultad De Comunicación Social.).

- Ryan, N., Pascoe, J., & Morse, D. (1999). Enhanced Reality Fieldwork: The Context Aware Archaeological Assistant. *Bar International Series*, 750, 269-274.
- Santiago, R., Trbaldo, S., Kamijo, M., & Fernández, Á. (2015). *Mobile Learning: nuevas realidades en el aula*. Editorial Oceano
- Sfaello Z, (2015) Clasificación de la PC. Modulo: Alteraciones Neurofisiológicas Del Desarrollo y el Aprendizaje. Diplomatura Superior en Educación Inclusiva- Atención a la Diversidad
- Strang, T., & Linnhoff-Popien, C. (2004, September). A Context Modeling Survey. In *Workshop Proceedings*.
- Valenzuela, S. S. R., & Terriza, J. A. H.(2008) Servicios Sensibles Al Contexto En Sistemas De Computación Ubicua.
- Villa, M. G., & Martínez, L. D. C. T. R. (2008). *Comunicación Aumentativa Y Alternativa*.

Anexo



ENCUESTA A LOS TERAPEUTAS DEL IPRI

Fecha: 5 de Mayo de 2016

Lugar: 24 de Septiembre n°446

Dentro de los pacientes que asisten al IPRI ¿Qué afecciones presentan aquellos que utilizan Sistemas alternativos de Comunicación?

Se presentan pacientes con muchos tipos de afecciones inclusive de diferentes edades. Hay pacientes con trastornos en el habla como dislalias, disartrias, retrasos del lenguaje, trastornos fonéticos, fonológicos, y muchos otros que pueden ser producto de condiciones físicas como es el caso de los que tienen parálisis.

Centrándonos en la población de pacientes menores ¿Cómo se comunican los que tienen dificultades para hablar?

La mayoría utiliza gestos y señas acompañado por sonidos que pueden emitir. También usan cuadernos en los que van indicando que es lo que necesitan y los terapeutas tenemos que interpretar y ayudar a que puedan expresarse. También usan tarjetas con imágenes que representan lo que quieren.

¿Utilizan actualmente alguna aplicación software que apoyen el proceso de comunicación entre ustedes y los pacientes?

No, en algunos casos sabemos que hay padres que si tienen en sus teléfonos, pero nosotros no las usamos. Tenemos conocimiento de su existencia pero no se adaptan a la forma de trabajo nuestra y tampoco tenemos los recursos para hacerlo.

Si tuvieran la posibilidad de implementar el uso de una aplicación de este estilo para trabajar con los niños ¿lo harían?

Seguro, el tema es que tenemos también que enseñar a usar estos otros recursos, porque no siempre van a tener al alcance la aplicación. Pero si tuviéramos la posibilidad de usarla para comunicarnos con ellos durante las terapias acelerando el proceso de comunicación de esta manera, y a la vez enseñarles a usar el sistema manual, por supuesto que lo haríamos.

Si se pudiera implementar un sistema en dispositivos móviles (tabletas, teléfonos) ¿Qué mejora se podría lograr en el proceso de comunicación con sus pacientes?

El tiempo que nos toma comunicarnos, porque el proceso desde que el chico te demuestra que quiere decir algo, hasta que nosotros logramos entender eso que nos está

queriendo decir, es largo, lleva mucho tiempo. Primero nosotros tenemos que más o menos suponer dentro de la actividad que estamos realizando, qué es lo que puede necesitar, y en base a eso tenemos que darles las tarjetas. Pero es complejo y hasta desesperante a veces, imagínense que hasta cuando necesitan ir al baño o están sucios y necesitan un cambio hay que hacer todo ese proceso, de mostrar tarjetas gestos y sonidos (si es que lo pueden hacer). Si tuviésemos a disposición algo que nos acorte ese proceso de comunicación que es lento, para nosotros y para ellos sería una gran ayuda.

¿Qué necesitaría un programa o aplicación para que cumpla con las necesidades de comunicación que tienen?

Y bueno, lo básico imágenes de las necesidades básicas y que hable con eso ya se haría la diferencia.

Si bien hay sistemas que lo hacen, pero son complicados de configurar, entonces no tendría que tener mucha configuración.

Además en lo posible que reproduzca lo que usamos como termino aquí en Santiago, porque por ejemplo hay aplicaciones que reproducen sonidos para expresar las imágenes que seleccionadas, pero lo que “dicen” por ahí no es lo que todos conocemos en nuestra sociedad, como por ejemplo para decir que estas enojado algunos dicen “me embale”, y uno (terapeuta) tienes que explicar al niño que esa expresión significa que estas “trabado y no puedes continuar” para nosotros y que en otros lugares se les dice de esa otra manera. Esto pasa con varias palabras de los comunicadores que hay, por eso mejor que el comunicador, relacione las imágenes con acciones de aquí. Para que cualquiera que escuche lo que el comunicador dice lo entienda y no haya necesidad de explicarlo.

Ustedes ¿conocen más o menos cómo funcionan los que usan los padres con sus niños?

Si son básicos, aunque los más completos son difíciles de configurar, los padres tienen que elegir las imágenes y ordenarlas dependiendo como las van a usar. Pero no son diseñados para que los use cualquier persona, por ejemplo para los niños con dificultades motoras les resulta difícil manejarlos tiene muchas funciones que no están pensadas para que ellos las usen, por eso no son tan aplicables en estos chicos. Para el resto quizás sí.

¿Sabrían ustedes si la mayoría de los padres de los niños tienen acceso a Internet? esta pregunta es para ver el tipo de sistema que se adecuaría mas a la mayoría.

No, los pacientes que asisten son muchos del interior y el servicio de cobertura no siempre está disponible, muchos de ellos vienen de zonas rurales en donde con suerte hay servicio de telefonía celular. Si el sistema no usara internet creemos seria más accesible a todos.

Considerando lo que nos comento sobre los pacientes que usan sistemas alternativos de comunicación ¿Cuál es la población que según ustedes le sería más útil una aplicación que tenga función de comunicador?

Los niños que tienen dificultades motoras. Porque un niño en estas condiciones requiere ayuda permanente para comunicar que necesita y el proceso es más lento que con el resto. Consideren que ellos no se pueden mover bien y sumen encima que no pueden comunicarse de forma oral, es complicado en estos casos. Así que serian ellos los que más le daría utilidad, consideren también que los sistemas que hay no son fáciles de usar para ellos.

¿Cuál consideran que es el grupo de edades representativo de la población de pacientes niños que hace uso de sistemas alternativos de comunicación?

Tenemos pacientes variados, pero en su mayoría se puede decir que son menores a 10 años, mayores casi no vienen. Si bien con todos se trabaja con esta forma de comunicación, comienzan a comprender instrucciones simples de cómo funciona alrededor de los 2 y de ahí en adelante se usa con más frecuencia.

¿Qué sistema de comunicación alternativo usan con los pacientes que no pueden hablar?

Usamos tarjetas, cuadernos, pizarras, en todas ellas el niño debe indicar que es lo que necesita o, por ahí nosotros le preguntamos señalando “¿esto?” y el nos mueve la cabeza o sonrío, eso es muy propio de cada paciente. Pero esa es la forma en general que estamos manejando para comunicarnos dentro de las terapias.

¿Cómo manejan los sistemas alternativos de comunicación en las terapias?

Bueno, no es mucha ciencia, tenemos un montón de fichas o tarjetas con imágenes que representan acciones que nosotros les mostramos, por ejemplo una cara sonriente quiere decir que está contento o feliz, el niño señala la imagen o nos da a entender cual necesita que le acerquemos.

Aunque también en vez de tarjetas usamos impresiones donde hay disponibles varias imágenes y ahí selecciona lo que quiere. Si no usamos un pizarrón donde ponemos pegadas las tarjetas de distintos tamaños y ellos van señalando, o nosotros señalamos preguntando “¿esto necesitas?” y ellos asientan o niegan.

¿Tienen algunas imágenes que usen con más frecuencia en la mayoría de los casos?

Si siempre dejamos más al alcance algunas opciones que otras, por ejemplo las que representan ir al baño, cansancio, o dolor, como es un proceso lento, estas cuestiones siempre tienen prioridad. Son cosas que uno en la práctica de todos los días se va dando cuenta.

Respecto a la forma de comunicación que usan, ¿Usan la misma para todos los pacientes o se basan en grupos de edades?

Para todos usamos la misma forma de comunicarnos, lo que en realidad varía es la cantidad de imágenes que usamos con cada uno, hay que considerar que lo que necesita un niño de 1 año no es lo mismo que lo que necesita uno de 7. Más que nada es eso lo que se hace.

Respecto a las imágenes que usan ¿Son iguales para todas las edades?

Si se intenta de mantener una concordancia en lo que vas usando a lo largo del tiempo en la medida que el paciente va creciendo, lo que cambia es la cantidad de opciones que se le presenta a uno o a otro dependiendo de la edad.

¿Los padres manejan de la misma manera que ustedes los sistemas de comunicación?

Si, solo que en la casa es más complicado, porque hay más cosas que pueden querer, y además tienen que desplazarse en distintos lugares y a todos lados deben ir llevando su cuaderno comunicador. Por ahí es más práctico cuando salen de su casa usar aplicaciones que descargan en el teléfono, pero para los chicos con dificultades motoras no son tan fáciles de usar, así que es complicado igual.

¿Qué resultados les interesaría obtener de la aplicación o programa?

Como les dijimos, es que se acorten los tiempos, desde que existe la necesidad en el niño, te das cuenta, interpretas lo que “puede llegar a querer”, le brindas las posibilidades de lo que puede comunicar y te señala o asienta lo que necesita. Es decir desde que te quiere decir algo hasta que realmente te lo dice. Y si se pudiera que muestre las imágenes dependiendo si es varón o mujer.

¿Usan las mismas imágenes para ambos géneros? o ¿Hay alguna diferencia que consideren?

Cambian algunas imágenes pero no todas. Lo que varía quizás la forma de presentarle las opciones, por ejemplo a una nena le puedes poner como opciones juegos con muñecas y a los varones con audífonos. Pero no más de eso, al ser chicos como que no hay mucha diferencia, pero en la medida que crecen sí, cada uno se identifica más con ciertas acciones pero no es algo rígido.

Si hacen un manejo diferenciado ¿Cómo se maneja cada uno?

Como explicamos antes, varía la cantidad de imágenes que ofrecemos a los chicos después el resto es igual.

En caso de optar por el desarrollo de una aplicación ¿Cómo se podría trabajar? ¿En qué horarios? ¿Dónde?

Para eso sería bueno que estén todas las terapeutas que trabajamos en el instituto, para eso tendrían que coordinar con nosotras para ponernos de acuerdo que horario y día tenemos todas disponibles para reunirnos. Lo ideal sería en el instituto, es en donde normalmente nos hacemos un tiempo para charlar sobre nuestros pacientes y como va cada uno. Así que primero se comunican con nosotros y el día que ustedes puedan vemos de tomarnos un tiempo para que ustedes puedan llegar para hablar mejor sobre el tema.

CÓDIGO DE DESARROLLO

En este apartado se detalla el código correspondiente a las diferentes funciones desarrolladas en las iteraciones de desarrollo del prototipo de SAC

Iteración I:

```

/* _____ CREACION DE LA INTERFAZ PRINCIPAL(CASA) DE LA APLICACIÓN _____ */
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_casa);
    TipoEdad = 1;
    TipoSexo = 1 ;
    // _____ Obtener la Resolucion de la Pantalla en donde correr el APP _____
    Display display = ((WindowManager)
    getSystemService(Context.WINDOW_SERVICE)).getDefaultDisplay();
    float scale = getApplicationContext().getResources().getDisplayMetrics().density;
    //Ancho de la Pantalla
    int ancho = display.getWidth();
    //Alto de la Pantalla

```

```

int alto = display.getHeight();
//Comodeseo hacerlo cuadrado al boton coloco el mismo tamaño
int resulancho = ancho /3;
int resulAlto = alto / 4;
int margen = resulancho / 3;
// _____ Fin de Resolucion de Pantalla _____
terapia2a3 = Estoy2a3 = Quiero2a3 = 0;
Interfaz(TipoSexo,TipoEdad);
//-----Creacion de Botones Dinamicos-----
if (ArrayImagenSeleccionado != null ) {
    mTlayout = (TableLayout) findViewById(R.id.mTlayout);
    int j = 0;
    while (j < ArraytextoSeleccionado.length) {
        if (j % 6 == 0) // Asigno 6 botones por fila
        {
            tr = new TableRow(this);
            //Asigno bordes a cada celda
            TableLayout.LayoutParams tableRowParams =
                new TableLayout.LayoutParams
                    (TableLayout.LayoutParams.FILL_PARENT,
TableLayout.LayoutParams.FILL_PARENT);
            int leftMargin = 0;
            int topMargin = 3;
            int rightMargin = 0;
            int bottomMargin = 3;
            tableRowParams.setMargins(leftMargin, topMargin, rightMargin, bottomMargin);
            tr.setLayoutParams(tableRowParams);
            //tr.setDividerPadding(20);
            tr.setPadding(leftMargin, topMargin, rightMargin, bottomMargin);
            mTlayout.addView(tr, tableRowParams);
        } //Creacion del Boton
        Button btn = new Button(this);
        //Asigno ID
        btn.setId(j);
        //Asigno la Imagen por medio de i y el Arreglo de Imagnes arriba Creado
        btn.setBackground(this.getResources().getDrawable(ArrayImagenSeleccionado[j]));
        //Accion cuando toque el boton
        btn.setOnClickListener(this);
        //Asigno el boton, Ancho y Alto
        tr.addView(btn, resulAlto, resulAlto);
        j++;
    } }
else {Toast.makeText(Casa.this, "arreglo vacio" , Toast.LENGTH_LONG).show();}
}

```

```

/*ACCIONES ANTE UN CLIC EN UN BOTON DE LA PANTALLA*/

```

```

@Override

```

```

public void onClick(View v) {
    int Texto = ArraytextoSeleccionado[((Button) v).getId()];
    //-----inicializo parametros a pasar por actividad-----
    int quiero=0;
    int estoy=0;
    int vacio=0;
    int[] botones={vacio,terapia2a3,Estoy2a3,Quiero2a3};
    idIngresado = v.getId();
    int seleccionado=0;
    int pbtn=0;
    while (pbtn<botones.length) {

```

```

    if (idIngresado==botones[pbtn])
    {
        seleccionado=pbtn;
        pbtn=botones.length;
    } else {pbtn=pbtn+1;}
}
sonido=MediaPlayer.create(this, Texto);
if (seleccionado==0){
    if (sonido!=null) { sonido.start();}
    else{Toast.makeText(this, "NO SE CARGO EL SONIDO " + Texto ,
Toast.LENGTH_LONG).show();}
}
else{if (sonido!=null){sonido.start();}
else {Toast.makeText(this, "NO SE CARGO EL SONIDO " + Texto ,
Toast.LENGTH_LONG).show();}
Intent i= new Intent();
switch (seleccionado)
{
    case 1:// _____ CASA _____
        i = new Intent(Casa.this, Terapia.class);
        break;
    case 2:// _____ ESTOY _____
        estoy=1;
        break;
    case 3:// _____ QUIERO _____
        i = new Intent(Casa.this, Uno.class);
        quiero=1;
        break;
}
i.putExtra("EDAD", TipoEdad);
i.putExtra("GENERO", TipoSexo );
i.putExtra("DETERAPIA",0);
i.putExtra("QUIERO",quiero);
i.putExtra("ESTOY", estoy);
startActivity(i);
} }

```

```

/*CREACION DE LA INTERFAZ SECUNDARIA(TERAPIA) DE LA APLICACIÓN _____ */
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_terapia);
    Intent intent=getIntent();
    Bundle extras=intent.getExtras();
    if (extras==null) {
        Toast.makeText(this, "extras vacio: " + extras, Toast.LENGTH_LONG).show();}
    else {
        ed=extras.getInt("EDAD");
        gen=extras.getInt("GENERO");
        ...
}
/*OBTENGO LA RESOLUCION DE LA PANTALLA PARA DISTRIBUIR LA INTERFAZ*/
...
//INICIALIZO TODAS LAS VARIABLES QUE ME LLEVAN A OTRA ACTIVIDAD-----
estoy2y3=0;
aprendi2y3=0;
expresarme2y3=0;
Interfaz(ed, gen); ...
/*CREACION DE LOS BOTONES DINAMICOS DE LA INTERFAZ*/...}

```

```

/*ACCIONES ANTE UN CLIC EN UN BOTON DE LA PANTALLA*/
@Override
public void onClick(View v) {
    int Texto = ArraytextoSeleccionado[((Button) v).getId()];
    //-----inicializo parametros a pasar por actividad-----
    int estoy=0;
    int expresarme=0;
    idIngresado = v.getId();
    sonido=MediaPlayer.create(this, Texto);
    ...
    /*DETERMINAR DE LOS BOTONES QUE CAMBIAN DE INTERFAZ*/
    ...
    /* DETERMINA SI EL BOTON SELECCIONADO TIENE UN ACCION ASOCIADA*/
        if (sonido!=null)
        {sonido.start();}
        else{Toast.makeText(this, "NO SE CARGO EL SONIDO " + Texto ,
        Toast.LENGTH_LONG).show();}
        switch (seleccionado) {
            case 1:// _____ CASA _____
                i=new Intent(Terapia.this, Casa.class);
                i.putExtra("EDAD",ed );
                i.putExtra("GENERO",gen );
                startActivity(i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP |
Intent.FLAG_ACTIVITY_SINGLE_TOP));
                finish();
                break;
            case 2:// _____ ESTOY _____
                i = new Intent(this, Uno.class);
                estoy=1;
                i.putExtra("ESTOY", estoy);
                i.putExtra("EXPRESARME", expresarme);
                i.putExtra("EDAD", ed);
                i.putExtra("GENERO", gen);
                i.putExtra("DETERAPIA", 1);
                startActivity(i);
                break;
            case 3:// _____ EXPRESARME _____
                i = new Intent(this, Uno.class);
                expresarme=1;
                i.putExtra("ESTOY", estoy);
                i.putExtra("EXPRESARME", expresarme);
                i.putExtra("EDAD", ed);
                i.putExtra("GENERO", gen);
                i.putExtra("DETERAPIA", 1);
                startActivity(i);
                break;
        }
    }
}

```

```

/* _____ CREACION DE LA INTERFAZ AUXILIARES DE LA APLICACIÓN _____ */
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_uno);
    Intent intent=getIntent();
    Bundle extras=intent.getExtras();
    /*RECIBO LOS PARAMETROS PASADOS A ESTA ACTIVIDAD*/
    ...
}

```

```

/*OBTENGO LA RESOLUCION DE LA PANTALLA PARA DISTRIBUIR LA INTERFAZ*/
...
/*INICIALIZO TODAS LAS VARIABLES QUE ME LLEVAN A OTRA ACTIVIDAD*/
...
Interfaz(ed, gen, quiero, estoy, expresarme);
...
/*CREACION DE LOS BOTONES DINAMICOS DE LA INTERFAZ*/
...
}

```

```

/*ACCIONES ANTE UN CLIC EN UN BOTON DE LA PANTALLA*/
@Override
public void onClick(View v) {
    int Texto = ArraytextoSeleccionado[((Button) v).getId()];
    //-----inicializo parametros a pasar por actividad-----
    String accion="";
    idIngresado = v.getId();
    sonido=MediaPlayer.create(this, Texto);
    ...
    /*DETERMINAR DE LOS BOTONES QUE CAMBIAN DE INTERFAZ*/
    ...
    /* DETERMINA SI EL BOTON SELECCIONADO TIENE UN ACCION ASOCIADA*/
    ...
    /* REPRODUCE EL SONIDOS ASOCIADO A LA SELECCIÓN */
    ...

        switch (seleccionado) {
            case 1:// _____COMER_____
                i = new Intent(this, Dos.class);
                i.putExtra("ACCION", "comida");
                break;
            case 2:// _____TOMAR_____
                i = new Intent(this, Dos.class);
                i.putExtra("ACCION", "bebida");
                break;
            case 3:// _____VER TELEVISION_____
                i = new Intent(this, Dos.class);
                i.putExtra("ACCION", "verTelevision");
                break;
            case 4:// _____JUGAR_____
                i = new Intent(this, Dos.class);
                i.putExtra("ACCION", "jugar2y3");
                break;
            case 5:// _____APRENDI_____
                i = new Intent(this, Colores.class);
                i.putExtra("ACCION", "aprendi2y3");
                break;
        }
    i.putExtra("EDAD", ed);
}

```

```

        i.putExtra("GENERO", gen);
        i.putExtra("DETERAPIA", deTerapia);
        startActivity(i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP |
Intent.FLAG_ACTIVITY_SINGLE_TOP));
        finish();
    }

```

```

/*_____ CREACION DE LA INTERFAZ AUXILIARES DE LA APLICACIÓN _____*/
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_dos);
    Intent intent=getIntent();
    Bundle extras=intent.getExtras();
    /*RECIBO LOS PARAMETROS PASADOS A ESTA ACTIVIDAD*/
    ...
    /*OBTENGO LA RESOLUCION DE LA PANTALLA PARA DISTRIBUIR LA INTERFAZ*/
    ...
    /*INICIALIZO TODAS LAS VARIABLES QUE ME LLEVAN A OTRA ACTIVIDAD*/
    ...
    Interfaz(gen, accion);
    ...
    /*CREACION DE LOS BOTONES DINAMICOS DE LA INTERFAZ*/
    ...}

```

```

/*ACCIONES ANTE UN CLIC EN UN BOTON DE LA PANTALLA*/
@Override
public void onClick(View v) {
    int Texto = ArraytextoSeleccionado[((Button) v).getId()];
    //-----inicializo parametros a pasar por actividad-----
    idIngresado = v.getId();
    sonido=MediaPlayer.create(this,Texto);
    Intent i= new Intent();
    /*REPRODUCIR SONIDO*/
    ...
    if (viene==0){
        i=new Intent(Dos.this, Casa.class);
        i.putExtra("EDAD",ed );
        i.putExtra("GENERO",gen );
    }
    else
    { /*REPRODUCIR SONIDO*/

        i=new Intent(Dos.this, Terapia.class);
        i.putExtra("EDAD", ed);
        i.putExtra("GENERO",gen );
    }
    startActivity(i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP |
Intent.FLAG_ACTIVITY_SINGLE_TOP));
    finish();
}

```

Iteración II:

```

/* INTERFAZ DE INGRESO DE PREFERENCIAS DE USUARIO*/
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_inicio);
    /*
        EDAD
    */
    tvFN=(TextView)findViewById(R.id.tv_edad);
    edt_Nya=(EditText)findViewById(R.id.EdtNombyap);
    Date=(Button)findViewById(R.id.button_Date);
    salir=(Button)findViewById(R.id.buttonSalir);
    /*
        GENERO
    */
    fem = (RadioButton) findViewById(R.id.radio_femenino);
    masc = (RadioButton) findViewById(R.id.radio_masculino);
    /*
        PREFERENCIAS
    */
    preferencias = getSharedPreferences(MyPREFERENCES, Context.MODE_PRIVATE);
    /*
        BOTON DE INICIAR COMUNICADOR
    */
    iniciar=(Button)findViewById(R.id.button);
    salir=(Button)findViewById(R.id.buttonSalir);
    /*
        BOTON CONFIGURAR EDAD
    */
    Date.setOnClickListener(this);
    /*
        TODO LISTO => INICIAR
    */
    iniciar.setOnClickListener(this);
    /*
        => CERRAR SESION
    */
    salir.setOnClickListener(this);
    /*
        GUARDAR LAS PREFERENCIAS CARGADAS
    */
    getProfile();
}

```

```

/*ACCIONES ANTE UN CLIC EN UN BOTON DE LA PANTALLA*/
@Override
public void onClick(View v) {
    switch (v.getId())
    { case R.id.button_Date: /*
        CONFIGURAR LA EDAD DEL USUARIO
    */
        showDialog(DATE_ID);
        Calendar c=Calendar.getInstance();
        sYear=c.get(Calendar.YEAR);
        sMonth=c.get(Calendar.MONTH);
        sDay=c.get(Calendar.DAY_OF_MONTH);
        break;
    case R.id.buttonSalir: //
        SALIR DE LA SESION Y CERRAR LA APLICACION
        SharedPreferences.Editor editor = preferencias.edit();
        editor.clear();
        editor.commit();
        getProfile();
        finish();
        break;
    case R.id.button: /*INICIAR SESION DEL COMUNICADOR Y PASAR A LA INTERFAZ DE
    CASA*/
        //
        COMRPUEBA EL GENERO DEL USUARIO
        if ( fem.isChecked())
        { TipoGenero=2;}
        else
        {
            if ( masc.isChecked())
            { TipoGenero=1; }
        }
    }
}

```



```

    }
// _____ GUARDA LOS DATOS DE LA SESION INICIADA _____
    saveProfile();
// _____ PREPARO INTENT / CARGO PARAMETROS / INICIO ACTICIDAD DE CASA _____
    Intent i=new Intent(Inicio.this,Casa.class);
    i.putExtra("EDAD",TipoEdad);
    i.putExtra("GENERO", TipoGenero);
    i.putExtra("NOMBRE", edt_Nya.getText().toString());
    startActivity(i);
    break;
}.....}

```

```

/* _____ CREACION DE LA INTERFAZ PRINCIPAL(CASA) DE LA APLICACIÓN _____ */
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_casa);
    Intent intent = getIntent();
    Bundle extras = intent.getExtras();
    if (extras == null) {
        Toast.makeText(this, "extras vacio: " + extras, Toast.LENGTH_LONG).show();
    } else {
        /* _____ SACO LOS PARAMETROS RECIBIDOS _____ */
        TipoEdad = extras.getInt("EDAD");
        TipoSexo = extras.getInt("GENERO");
        nombre = extras.getString("NOMBRE");
        nombre = "yo me llamo " + nombre;
        //-----Configuracion de Audio-----
        t1 = new TextToSpeech(getApplicationContext(), new TextToSpeech.OnInitListener() {
            @Override
            public void onInit(int status) {
                if (status != TextToSpeech.ERROR) {
                    t1.setLanguage(new Locale("spa", "ESP"));
                }
            }
        });
        /*OBTENER RESOLUCION DE PANTALLA*/
    }
}
...
/* _____ VARIABLES DE DESPLAZAMIENTO _____ */
terapia2a3 = Estoy2a3 = Quiero2a3 = 0;
terapia4a6 = Expresarme4a6 = Preguntar4a6 = Quiero4a6 = Jugar4a6 = Estoy4a6 = 0;
terapia7a10 = Estoy7a10 = Expresarme7a10 = Preguntar7a10 = Quiero7a10 = 0;
/* _____ DEFINR EL CONTENIDO DE LA PANTALLA _____ */
Interfaz(TipoEdad, TipoSexo);
ProbarLongitud();
/*CREACION DE BOTONES DINAMICOS*/
...
}

```

```

/*ACCIONES DE MENU DE OPCIONES DE BARRA DE ESTADO*/
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu, menu); return true;}

```

```
@Override
```

```
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) { case R.id.CerrarSesion: finish(); break;
    default: }
    return super.onOptionsItemSelected(item);}
```

```
/*ACCIONES ANTE UN CLIC EN UN BOTON DE LA PANTALLA PRINCIPAL*/
```

```
@Override
```

```
public void onClick(View v) {
    int Texto = ArraytextoSeleccionado[((Button) v).getId()];
    idIngresado = v.getId();
    /*INICIALIZO PARAMETROS A PASAR POR ACTIVIDAD*/
    ...
    // cargo el sonido
    sonido = MediaPlayer.create(this, Texto);
    if (desplazamiento(idIngresado) == 0) { //determina el tipo de acción asociada al botón seleccionado
        if (sonido != null) {
            if (idIngresado == 0) {
                String toSpeak = nombre;
                t1.speak(toSpeak, TextToSpeech.QUEUE_FLUSH, null);
            } else {
                sonido.start();
            }
        }
        else {Toast.makeText(this, "NO SE CARGO EL SONIDO " + Texto,
        Toast.LENGTH_LONG).show(); }
    } else {
        Intent i = new Intent();
        switch (desplazamiento(idIngresado)) {
            case 1:/*_____ TERAPIA DE 2 Y 3 AÑOS _____*/
                i = new Intent(Casa.this, Terapia.class);
                break;
            case 2:/*_____ ESTOY DE 2 Y 3 AÑOS _____*/
                reproducir(Texto);
                i = new Intent(Casa.this, Uno.class);
                estoy = 1;
                break;
            case 3:/*_____ QUIERO DE 2 Y 3 AÑOS _____*/
                i = new Intent(Casa.this, Uno.class);
                quiero = 1;
                break;
            case 4:/*_____ TERAPIA DE 4 Y 6 AÑOS _____*/
                i = new Intent(Casa.this, Terapia.class);
                break;
            case 5:/*_____ EXPRESARME DE 4 Y 6 AÑOS _____*/
                i = new Intent(Casa.this, Uno.class);
                expresarme = 1;
                break;
            case 6:/*_____ PREGUNTAR DE 6 Y 6 AÑOS _____*/
                i = new Intent(Casa.this, Uno.class);
                preguntar = 1;
                break;
            case 7:/*_____ QUIERO DE 4 Y 6 AÑOS _____*/
                i = new Intent(Casa.this, Uno.class);
                quiero = 1;
                break;
            case 8:/*_____ JUGAR DE 4 Y 6 AÑOS _____*/
                i = new Intent(Casa.this, Dos.class);
                i.putExtra("ACCION", "Jugar4a6");
```

```

        break;
    case 9:/*_____ ESTOY DE 4 Y 6 AÑOS _____*/
        reproducir(Texto);
        i = new Intent(Casa.this, Uno.class);
        estoy = 1;
        break;
    case 10:/*-----TERAPIA DE 7 A 10-----*/
        i = new Intent(Casa.this, Terapia.class);
        break;
    case 11:/*-----ESTOY DE 7 A 10-----*/
        reproducir(Texto);
        i = new Intent(Casa.this, Uno.class);
        estoy = 1;
        break;
    case 12:/*-----EXPRESARME DE 7 A 10-----*/
        i = new Intent(Casa.this, Uno.class);
        expresarme = 1;
        break;
    case 13:/*-----PREGUNTAR DE 7 A 10-----*/
        i = new Intent(Casa.this, Uno.class);
        preguntar = 1;
        break;
    case 14:/*-----QUIERO DE 7 A 10-----*/
        i = new Intent(Casa.this, Uno.class);
        quiero = 1;
    }
    i.putExtra("EDAD", TipoEdad);
    i.putExtra("GENERO", TipoSexo);
    i.putExtra("NOMBRE", nombre);
    i.putExtra("DETERAPIA", 0);
    i.putExtra("QUIERO", quiero);
    i.putExtra("ESTOY", estoy);
    i.putExtra("EXPRESARME", expresarme);
    i.putExtra("PREGUNTAR", preguntar);
    startActivity(i);
}
}
}

```

```

/*CREACION DE LA INTERFAZ SECUNDARIA (TERAPIA) DE LA APLICACIÓN _____*/
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_terapia);
    ...Intent intent=getIntent();
    Bundle extras=intent.getExtras();
    /*RECIBO LOS PARAMETROS PASADOS A ESTA ACTIVIDAD*/
    ...
    /*REALIZO LA CONFIGURACION DE AUDIO*/
    ...
    /*OBTENGO LA RESOLUCION DE LA PANTALLA PARA DISTRIBUIR LA INTERFAZ*/
    ...
    /*INICIALIZO TODAS LAS VARIABLES QUE ME LLEVAN A OTRA ACTIVIDAD*/
    ...
    Interfaz(ed, gen);
    ProbarLongitud();
    /*CREACION DE LOS BOTONES DINAMICOS DE LA INTERFAZ*/
    ...
}
}

```

```

/*ACCIONES ANTE UN CLIC EN UN BOTON DE LA PANTALLA SECUNDARIA */
@Override
public void onClick(View v) {
    int Texto = ArraytextoSeleccionado[((Button) v).getId()];
    /*INIICIALIZO PARAMETROS A PASAR A OTRA ACTIVIDAD*/
    ...
    /*CARGAR EL SONIDO*/
    ...
    /*EXTRAER ACCION ASOCIADA AL BOTON SELECCIONADO*/
    ...
    /* REPRODUCIR SONIDO*/
    reproducir(Texto);
    Intent i = new Intent();
    switch (desplazamiento(idIngresado)) {
        case 1:/* CASA DE 2 Y 3 AÑOS */
            i=new Intent(Terapia.this, Casa.class);
    /*AGREGAR PARAMETROS A PASAR A LA SIGUIENTE ACTIVITY*/
        startActivity(i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP |
Intent.FLAG_ACTIVITY_SINGLE_TOP));
        finish();
        break;
        case 2:/* CASA DE 4 Y 6 AÑOS */
            i=new Intent(Terapia.this, Casa.class);
    /*AGREGAR PARAMETROS A PASAR A LA SIGUIENTE ACTIVITY*/
        startActivity(i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP |
Intent.FLAG_ACTIVITY_SINGLE_TOP));
        finish();
        break;
        case 3:/* CASA DE 7 Y 10 AÑOS */
            i=new Intent(Terapia.this, Casa.class);
    /*AGREGAR PARAMETROS A PASAR A LA SIGUIENTE ACTIVITY*/
        startActivity(i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP |
Intent.FLAG_ACTIVITY_SINGLE_TOP));
        finish();
        break;
        case 4:/* ESTOY DE 2 Y 3 AÑOS */
            i = new Intent(this, Uno.class);
            estoy=1;
    /*AGREGAR PARAMETROS A PASAR A LA SIGUIENTE ACTIVITY*/...
        startActivity(i);
        break;
        case 5:/* ESTOY DE 4 Y 6 AÑOS */
            i = new Intent(this, Uno.class);
            estoy=1;
    /*AGREGAR PARAMETROS A PASAR A LA SIGUIENTE ACTIVITY*/...
        startActivity(i);
        break;
        case 6:/* ESTOY DE 7 Y 10 AÑOS */
            i = new Intent(this, Uno.class);
            estoy=1;
    /*AGREGAR PARAMETROS A PASAR A LA SIGUIENTE ACTIVITY*/ ...
        startActivity(i);
        break;
        case 7:/* DOLORIDO DE 4 Y 6 AÑOS */
            i = new Intent(this, Dos.class);
            i.putExtra("ACCION", "dolorido4a6");
            i.putExtra("EDAD", ed);
            i.putExtra("GENERO", gen);
    }
}

```

```

        i.putExtra("DETERAPIA", 1);
        startActivity(i);
        break;
    case 8:/*_____DOLORIDO DE 7 Y 10 AÑOS_____*/
        i = new Intent(this, Dos.class);
        i.putExtra("ACCION", "dolorido7a10");
        i.putExtra("EDAD", ed);
        i.putExtra("GENERO", gen);
        i.putExtra("DETERAPIA", 1);
        startActivity(i);
        break;
    case 9:/*_____EXPRESARME DE 2 Y 3 AÑOS_____*/
        i = new Intent(this, Uno.class);
        expresarme=1;
/*AGREGAR PARAMETROS A PASAR A LA SIGUIENTE ACTIVITY*/...
        startActivity(i);
        break;
    case 10:/*_____EXPRESARME DE 4 A 6 AÑOS_____*/
        i = new Intent(this, Uno.class);
        expresarme=1;
/*AGREGAR PARAMETROS A PASAR A LA SIGUIENTE ACTIVITY*/...
        startActivity(i);
        break;
    case 11:/*_____EXPRESARME DE 7 A 10 AÑOS_____*/
        i = new Intent(this, Uno.class);
        expresarme=1;
/*AGREGAR PARAMETROS A PASAR A LA SIGUIENTE ACTIVITY*/
...
        startActivity(i);
        break;
    case 12:/*_____PREGUNTAR DE 4 A 6 AÑOS_____*/
        i = new Intent(this, Uno.class);
        preguntar=1;
/*AGREGAR PARAMETROS A PASAR A LA SIGUIENTE ACTIVITY*/
...
        startActivity(i);
        break;
    case 13:/*_____EXPRESARME DE 7 A 10 AÑOS_____*/
        i = new Intent(this, Uno.class);
        preguntar=1;
/*AGREGAR PARAMETROS A PASAR A LA SIGUIENTE ACTIVITY*/
...
        startActivity(i);
        break;
    } } }

```

```

/*CREACION DE LA INTERFAZ AUXILIAR 1 DE LA APLICACIÓN _____*/
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_uno);
    Intent intent=getIntent();
    Bundle extras=intent.getExtras();
/*RECIBO LOS PARAMETROS PASADOS A ESTA ACTIVIDAD*/
...
/*REALIZO LA CONFIGURACION DE AUDIO*/
...
/*OBTENGO LA RESOLUCION DE LA PANTALLA PARA DISTRIBUIR LA INTERFAZ*/
...

```

```

/*INICIALIZO TODAS LAS VARIABLES QUE ME LLEVAN A OTRA ACTIVIDAD*/
...
    /*_____DEFINIR EL CONTENIDO DE LA PANTALLA_____*/
    Interfaz(ed, gen, quiero, estoy, preguntar, expresarme);
    ProbarLongitud();
/*CREACION DE LOS BOTONES DINAMICOS DE LA INTERFAZ*/
... }

```

```

/*ACCIONES ANTE UN CLIC EN UN BOTON DE LA PANTALLA AUXILIAR 1 */

```

```

@Override
public void onClick(View v) {
    int Texto = ArraytextoSeleccionado[((Button) v).getId()];
    idIngresado = v.getId();
/*CARGAR SONIDO*/
...
/*EXTRAER ACCION ASOCIADA AL BOTON SELECCIONADO*/
...
/* REPRODUCIR SONIDO*/
...
    Intent i = new Intent();
    switch (desplazamiento(idIngresado)) {
        case 1:/*_____COMER_____*/
            i = new Intent(this, Dos.class);
            i.putExtra("ACCION", "comida");
            break;
        case 2:/*_____TOMAR_____*/
            i = new Intent(this, Dos.class);
            i.putExtra("ACCION", "bebida");
            break;
        case 3:/*_____VER TELEVISION_____*/
            i = new Intent(this, Dos.class);
            i.putExtra("ACCION", "verTelevision");
            break;
        case 4:/*_____JUGAR 2 Y 3_____*/
            i = new Intent(this, Dos.class);
            i.putExtra("ACCION", "jugar2y3");
            break;
        case 5:/*_____JUGAR DE 7 A 10_____*/
            i = new Intent(this, Dos.class);
            i.putExtra("ACCION", "jugar7a10");
            break;
        case 6:/*_____DOLORIDO DE 4 A 6_____*/
            i = new Intent(this, Dos.class);
            i.putExtra("ACCION", "dolorido4a6");
            break;
        case 7:/*_____DOLORIDO DE 7 A 10_____*/
            i = new Intent(this, Dos.class);
            i.putExtra("ACCION", "dolorido7a10");
            break;
        case 8:/*_____NUMEROS DE 7 A 10_____*/
            i = new Intent(this, Dos.class);
            i.putExtra("ACCION", "numeros7a10");
            break;
        case 9:/*_____ANIMALES DE 7 A 10_____*/
            i = new Intent(this, Dos.class);
            i.putExtra("ACCION", "animales7a10");
            break;
        case 10:/*_____FIGURAS DE 7 A 10_____*/
            i = new Intent(this, Dos.class);

```

```

        i.putExtra("ACCION", "figuras7a10");
        break;
    case 11:/*_____ COLORES DE 7 A 10 _____*/
        i = new Intent(this, Colores.class);
        i.putExtra("ACCION", "colores7a10");
        break;
    case 12:/*_____ ABECEDARIO DE 7 A 10 _____*/
        i = new Intent(this, Dos.class);
        i.putExtra("ACCION", "abecedario7a10");
        break;
    case 13:/*_____ APRENDI DE 2 Y 3 _____*/
        i = new Intent(this, Colores.class);
        i.putExtra("ACCION", "aprendi2y3");
        break;
    case 14:/*_____ APRENDI DE 4 A 6 _____*/
        i = new Intent(this, Dos.class);
        i.putExtra("ACCION", "aprendi4a6");
        break;
    }
    *AGREGAR PARAMETROS A PASAR A LA SIGUIENTE ACTIVITY*/
    ...
    startActivity(i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP |
Intent.FLAG_ACTIVITY_SINGLE_TOP));
    finish();
} }

/* CREACION DE LA INTERFAZ AUXILIAR 2 DE LA APLICACIÓN */
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_dos);
    Intent intent=getIntent();
    Bundle extras=intent.getExtras();
    /*RECIBO LOS PARAMETROS PASADOS A ESTA ACTIVIDAD*/
    ...
    /*REALIZO LA CONFIGURACION DE AUDIO*/
    ...
    /*OBTENGO LA RESOLUCION DE LA PANTALLA PARA DISTRIBUIR LA INTERFAZ*/
    ...
    /*INICIALIZO TODAS LAS VARIABLES QUE ME LLEVAN A OTRA ACTIVIDAD*/
    ...
    /*_____ DEFINIR EL CONTENIDO DE LA PANTALLA _____*/
    Interfaz(gen, accion);
    ProbarLongitud();
    /*CREACION DE LOS BOTONES DINAMICOS DE LA INTERFAZ*/
    ... }

/*ACCIONES ANTE UN CLIC EN UN BOTON DE LA PANTALLA AUXILIAR 2*/
@Override
public void onClick(View v) {
    int Texto = ArraytextoSeleccionado[((Button) v).getId()];
    idIngresado = v.getId();
    String accion="";
    /*CARGAR SONIDO*/
    ...
    /*EXTRAER ACCION ASOCIADA AL BOTON SELECCIONADO*/
    ...

```

```

/* REPRODUCIR SONIDO*/
...
Intent i= new Intent(Dos.this,Colores.class);
switch (desplazamiento(idIngresado))
{
case 1:/*_____COLORES DE 4 A 6_____*/
accion="colores4a6";
break;
case 2:/*_____NUMEROS MAS DE 30_____*/
accion="masde30";
break;
}
/*AGREGAR PARAMETROS A PASAR A LA SIGUIENTE ACTIVITY*/
...
startActivity(i);
} }

/* CREACION DE LA INTERFAZ AUXILIAR COLORES DE LA APLICACIÓN */
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_colores);
Intent intent=getIntent();
Bundle extras=intent.getExtras();
/*RECIBO LOS PARAMETROS PASADOS A ESTA ACTIVIDAD*/
...
/*REALIZO LA CONFIGURACION DE AUDIO*/
...
/*OBTENGO LA RESOLUCION DE LA PANTALLA PARA DISTRIBUIR LA INTERFAZ*/
...
/*INICIALIZO TODAS LAS VARIABLES QUE ME LLEVAN A OTRA ACTIVIDAD*/
...
/*_____DEFINIR EL CONTENIDO DE LA PANTALLA_____*/
seleccionarPantalla( accion);
ProbarLongitud();
/*CREACION DE LOS BOTONES DINAMICOS DE LA INTERFAZ*/
...
}

/*ACCIONES ANTE UN CLIC EN UN BOTON DE LA PANTALLA AUXILIAR COLORES* /
@Override
public void onClick(View v) {
int Texto = ArraytextoSeleccionado[((Button) v).getId()];
idIngresado = v.getId();
Intent i= new Intent();
/*CARGAR SONIDO*/...
/*EXTRAER ACCION ASOCIADA AL BOTON SELECCIONADO*/
...
/* REPRODUCIR SONIDO*/
...
/*DEFINIR SIGUIENTE ACTIVITY Y AGREGAR PARAMETROS A PASAR */
...
startActivity(i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP |
Intent.FLAG_ACTIVITY_SINGLE_TOP));
finish();
}

```


Iteración III:

```

/* INTERFAZ DE INGRESO DE PREFERENCIAS DE USUARIO CON ALERTAS
PROGRAMABLES*/
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_inicio);
/*EDAD*/
...
/*GENERO*/
...
/*
    _____ALERTAS_____ */
    tvAdesayuno=(TextView)findViewById(R.id.tv_desayuno);
    tvAalmuerzo=(TextView)findViewById(R.id.tv_almuerzo);
    tvAmerienda=(TextView)findViewById(R.id.tv_merienda);
    tvAcena=(TextView)findViewById(R.id.tv_cena);
    des = (CheckBox) findViewById(R.id.checkBox);
    alm = (CheckBox) findViewById(R.id.checkBox2);
    mer = (CheckBox) findViewById(R.id.checkBox3);
    cen = (CheckBox) findViewById(R.id.checkBox4);
/*
    _____PREFERENCIAS_____ */
    preferencias = getSharedPreferences(MyPREFERENCIAS, Context.MODE_PRIVATE);
/*BOTON DE INICIAR COMUNICADOR*/
...
/*BOTON CONFIGURAR EDAD*/
...
/*INICIAR SESION*/
...
/*CERRAR SESION */
...
/*
    _____GUARDAR LAS PREFERENCIAS CARGADAS_____ */
    getProfile(); }

```

```

/* ACCIONES ANTE UN CLIC EN UN BOTON DE LA INTERFAZ DE INGRESO DE
PREFERENCIAS*/
@Override
public void onClick(View v) {
    switch (v.getId())
    {
        case R.id.button_Date:
            /*CONFIGURAR LA EDAD DEL USUARIO*/
            ...
            break;
        case R.id.buttonSalir:
            /*SALIR DE LA SESION Y CERRAR LA APLICACIÓN*/
            ...
            break;
        case R.id.button: /*INICIAR SESION*/
/* INICIAR EL SERVICIO DEL COMUNICADOR Y PASAR A LA INTERFAZ DE CASA*/
/*COMRPUEBA EL GENERO DEL USUARIO*/
...
/*GUARDA LOS DATOS DE LA SESION INICIADA*/
...
/* CREAMOS UN INTENT PARA INICIAR SERVICIO */
//
        Intent objIniciarServicio = new Intent();
        objIniciarServicio = new Intent(Inicio.this, ServiciosAlarmas.class);
        // Inicilizamos-BD

```

```

if (!ExisteBD("1")){ alta();}
// _____ COLOCO LOS DATOS DE LAS ALRMAS PROGRAMADAS EN LISTAS _____
ArrayList<String> alerta1=new ArrayList<String>();
ArrayList<String> alerta2=new ArrayList<String>();
ArrayList<String> alerta3= new ArrayList<String>();
ArrayList<String> alerta4=new ArrayList<String>();
String duracion;
// _____ CONTROLLO LAS ALARMAS QUE EL USUARIO QUIERE ACTIVAR _____
if (des.isChecked()){
    duracion=MiliseundosParaActivar(tvAdesayuno.getText().toString(),"Desayunar");
    alerta1.add(duracion);
    alerta1.add("HORARIO DE DESAYUNO");//titulo
    alerta1.add("ABRIR OPCIONES DE DESAYUNO");//cuerpo
    alerta1.add( "1");//tipo de alerta
}
if (alm.isChecked()){
    duracion=MiliseundosParaActivar(tvAalmuerzo.getText().toString(),"Almorzar");
    alerta2.add(duracion);
    alerta2.add( "HORARIO DE ALMUERZO");
    alerta2.add( "ABRIR OPCIONES DE ALMUERZO");
    alerta2.add( "2");
}
if (mer.isChecked()){
    duracion=MiliseundosParaActivar(tvAmerienda.getText().toString(),"Merendar");
    alerta3.add(duracion);
    alerta3.add("HORARIO DE MERIENDA");
    alerta3.add("ABRIR OPCIONES DE MERIENDA");
    alerta3.add("3");
}
if (cen.isChecked()){
    duracion=MiliseundosParaActivar(tvAcena.getText().toString(), "Cenar");
    alerta4.add(duracion);
    alerta4.add("HORARIO DE CENA");
    alerta4.add("ABRIR OPCIONES DE CENA");
    alerta4.add("4");
}
// PASAR PARAMETROS AL SERVICIOS DE ALERTAS _____
objIniciarServicio.putStringArrayListExtra("Alerta1", (ArrayList<String>) alerta1);
objIniciarServicio.putStringArrayListExtra("Alerta2", (ArrayList<String>) alerta2);
objIniciarServicio.putStringArrayListExtra("Alerta3", (ArrayList<String>) alerta3);
objIniciarServicio.putStringArrayListExtra("Alerta4", (ArrayList<String>) alerta4);
objIniciarServicio.putExtra("EDAD",TipoEdad);
objIniciarServicio.putExtra("GENERO", TipoGenero);
// _____ INICIAR SERVICIO _____
startService(objIniciarServicio);
// _____ PREPARO INTENT / CARGO PARAMETROS / INICIO ACTICIDAD DE CASA _____
Intent i=new Intent(Inicio.this,Casa.class);
i.putExtra("EDAD",TipoEdad);
i.putExtra("GENERO", TipoGenero);
i.putExtra("NOMBRE", edt_Nya.getText().toString());
startActivity(i);
break;
} }

```

```

/* INICIA EL HILO DE CONTROL DEL TIEMPO CORRESPONDIENTE AL SERVICIO*/

```

```

@Override

```

```

public int onStartCommand(Intent intent, int flags, int startId) {

```

```

    int j=0;

```

```

    //RECIBE LAS ALERTAS CONFIGURADAS POR EL USUARIO _____

```

```

alertas.add(j,intent.getExtras().getStringArrayList("Alerta1")); j++;
alertas.add(j,intent.getExtras().getStringArrayList("Alerta2")); j++;
alertas.add(j,intent.getExtras().getStringArrayList("Alerta3")); j++;
alertas.add(j,intent.getExtras().getStringArrayList("Alerta4"));
TipoEdad = intent.getExtras().getInt("EDAD");
TipoSexo = intent.getExtras().getInt("GENERO");
// ___ORDENA LA LISTA DE ACUERDO AL TIEMPO DE DURACION DE LAS ALERTAS
// _____SE COLOCA PRIMERO LA QUE SE EJECUTA PRIMERO_____
Collections.sort(alertas, new Comparator<List<String>>() {
    @Override
    public int compare(List<String> a, List<String> b) {
        return a.get(0).compareTo(b.get(0));
    }
});
// _____DETERMINA LOS ICONOS DE CADA UNA DE LAS ALERTAS_____
DeterminarTipoIconoAlarma(alertas);
//CREA E INICIA EL HILO DE CONTROL DEL TIEMPO_____
Thread t = new Thread(new Hilo());
t.start();
return START_NOT_STICKY;
}

```

```

/*HILO QUE CONTROLA EL TIEMPO Y EL LANZAMIENTO DE LAS ALERTAS*/
public class Hilo implements Runnable
{public void run() {
    while (true) {
        //permite que se ejecute permanentemente mientras esté abierta la aplicación
        indice = 0;
        long anterior=0;
        long duracion=0;
        Iterator<ArrayList<String>> itrArrayListAlarmas = alertas.iterator();
        while (itrArrayListAlarmas.hasNext()) {
            alarmai = itrArrayListAlarmas.next();
            duracion = Long.parseLong(alarmai.get(0))-anterior;
            if (indice!=0)
            {duracion=duracion-hs;}
            try {Thread.sleep(duracion); } catch (Exception e) {e.getMessage();}
            /*verifica que no se encuentre en Terapia para ejecutar el alerta mediante la notificación*/
            if (consulta("1")==1){
                GenerarNotificacion(alarmai.get(1), alarmai.get(2), ilcono.get(indice), abrir.get(indice));
                try {Thread.sleep(hs);} catch (Exception e) {e.getMessage(); }
                PararNotificacion();
            }
            else{ try {Thread.sleep(hs);} catch (Exception e) {e.getMessage();} }
            indice++;
            anterior=duracion;
        }
    }
}
}

```

PRUEBAS DE VERIFICACIÓN DE ITERACIÓN 0

PRUEBA DE ACEPTACIÓN				
Fecha de escritura: Julio de 2016				
Fecha de ejecución: Agosto de 2016				
{Qué funcionalidad debe implementar el sistema para ser validado}	RESULTADOS			
	Pasa	No pasa	Observaciones sobre la funcionalidad	
Iteración N° 0	Las acciones posibles a comunicar por el usuario están representadas por las imágenes del prototipo	X		
	Las imágenes del prototipo son conocidas por los usuarios		X No todas, algunas son un poco confusas. Usaría pictogramas de ARASAAC	
	Las imágenes del prototipo son fácilmente interpretables para el usuario		X	
	La cantidad de imágenes (acciones) es adecuado para una comunicación usando el prototipo	X		Agregaría algunas acciones de uso más frecuente como ser “ir al baño”, “estoy cansado”
	El tamaño de las imágenes es adecuado		X	Están demasiado chicos
	La distribución de las imágenes en la pantalla del prototipo es adecuada		X	La pizarra es más grande que las imágenes, no tendría que haber pizarra

PRUEBA DE ACEPTACIÓN				
Fecha de escritura: Julio de 2016				
Fecha de ejecución: Agosto de 2016				
	{Qué funcionalidad debe implementar el sistema para pasar}	RESULTADOS		
		Pasa	No pasa	Observaciones sobre la funcionalidad
Iteración N° 0	La selección de acciones no representan grandes dificultades de movimientos para el paciente		X	Hay que hacer desplazamientos en las opciones para seleccionar, y lo ideal sería que solamente las seleccione y no tenga que hacer barrido de la pantalla para encontrarla
	La agrupación del contenido (acciones y sub acciones) de las diferentes pantallas, se asemeja a como se secuencian lógicamente las acciones en una comunicación real	X		Pero habría que sacar el tema de los barridos para seleccionar y poder encontrar la imagen que necesitas. Fijarse si se pueden hacer botones que se abran cuando los toques y te muestre otras opciones más.
	La simulación para la reproducción de sonido es adecuada		X	Tiene que hacer muchos pasos para que se reproduzca. Sacaría el botón de reproducción y reproduciría solo con seleccionar la imagen.
Observaciones sobre la iteración: Los botones de regreso entorpecen a la funcionalidad y quitan espacio igual que el de reproducir. No es necesario borrar imágenes seleccionadas en ningún caso, para eso directamente debería seleccionarse la imagen y reproducirse, en el caso de que por error se selecciona una que no se quería no importa después se puede seleccionar otra y el que acompaña al paciente sabrá entender.				

FICHAS DE HU

Numero/ ID	Tipo	Dificultad		Esfuerzo ²⁴		Prioridad	Notas
		Antes	Después	Estimado	Gastado		
1	Nueva Arreglada Extendida	Fácil Moderado Difícil	Fácil Moderado Difícil			Alta Media Baja	Agrupamiento de palabras Niño con PCI (usuario final)
Descripción							
<p><u>Funcionalidad deseada:</u> Como terapeutas consideramos que las palabras disponibles en el SAC deben estar ordenadas según un esquema lógico de selección, es decir agrupadas por categorías o tipos de acción, para reducir los desplazamientos necesarios al momento de comunicarse y que se facilite encontrar lo que quiere expresar.</p>							
Estado	Fecha	Comentario					
Definido	07/16	Fase de inicialización: #Requerimientos iniciales vinculados: N°2, N°6 y N°7 Esfuerzo estimado: 3 días/ Tipo: Nueva					
Implementando	10/16	Fase de Producción: #Requerimiento redefinidos vinculados: RF8, RF9, RF1 y RF6 Esfuerzo estimado: 19 días/ Tipo: Extendida					
Verificado	08/16	Fase de inicialización: #Requerimientos iniciales vinculados: N°2, N°6 y N°7 Esfuerzo gastado: 4 días /Tipo: Extendida					
	11/16	Fase de Producción: #Requerimiento redefinidos vinculados: RF8, RF9, RF1 y RF6 Esfuerzo gastado: 25 días / Tipo: Extendida					

²⁴ En el esfuerzo estimado e insumido para la implementación y verificación de las HU 1 y 2, se considera unificado por lo definido en la Fase de Producción para sección 6.1.3.

Numero/ ID	Tipo	Dificultad		Esfuerzo		Prioridad	Notas
		Antes	Después	Estimado	Gastado		
2	Nueva	Fácil	Fácil			Alta	Navegación entre pantallas del prototipo
	Arreglada	Moderado	Moderado			Media	Paciente
	Extendida	Difícil	Difícil			Baja	Niño con PCI (usuario final)
Descripción							
<p><u>Funcionalidad deseada:</u> Como terapeutas necesitamos que el sistema no necesite en ningún caso grandes esfuerzos motrices del usuario para ejecutar sus opciones, para mejorar la fluidez en la comunicación</p>							
Estado	Fecha	Comentario					
Definido	07/16	Fase de inicialización: #Requerimientos iniciales vinculados: N°5 y N°7 Esfuerzo estimado: 4 días/ Tipo: Nueva					
Implementando	10/16	Fase de Producción: #Requerimiento redefinidos vinculados: RF8, RF9, RF1 y RF6 Esfuerzo estimado: 19 días/ Tipo: Extendida					
Verificado	08/16	Fase de inicialización: #Requerimientos iniciales vinculados: N°5 y N°7 Esfuerzo gastado: 6 días /Tipo: Extendida					
	11/16	Fase de Producción: #Requerimiento redefinidos vinculados: RF8, RF9, RF1 y RF6 Esfuerzo gastado: 25 días / Tipo: Extendida					

Numero/ ID	Tipo	Dificultad		Esfuerzo		Prioridad	Notas
		Antes	Después	Estimado	Gastado		
3	Nueva	Fácil	Fácil			Alta	Complejidad del vocabulario según la edad
	Arreglada	Moderado	Moderado			Media	Paciente
	Extendida	Difícil	Difícil			Baja	Niño con PCI (usuario final)
Descripción							
Funcionalidad deseada: Como terapeutas necesitamos que el sistema no necesite en ningún caso grandes esfuerzos motrices del usuario para ejecutar sus opciones, para mejorar la fluidez en la comunicación							
Estado	Fecha	Comentario					
Definido	07/16	Fase de inicialización: #Requerimiento vinculado: N°4 Esfuerzo estimado: 4 días/ Tipo: Nueva					
Implementando	08/16	Fase de inicialización: #Requerimiento vinculado: N°4 Esfuerzo gastado: 4 días /Tipo: Extendida					
	11/16	Fase de Producción: #Requerimiento redefinidos vinculados: RF10 y RG2 Esfuerzo estimado: 28 días/ Tipo: Extendida					
Verificado	08/16	Fase de inicialización: #Requerimiento vinculado: N°4 Esfuerzo gastado: 4 días /Tipo: Extendida					
	12/16	Fase de Producción: #Requerimiento redefinidos vinculados: RF10 y RG2 Esfuerzo gastado: 30 días / Tipo: Extendida					

Numero/ ID	Tipo	Dificultad		Esfuerzo		Prioridad	Notas
		Antes	Después	Estimado	Gastado		
4	Nueva	Fácil	Fácil			Alta	Alertas de situaciones predecibles
	Arreglada	Moderado	Moderado			Media	Paciente
	Extendida	Difícil	Difícil			Baja	Niño con PCI (usuario final)
Descripción							
<p><u>Funcionalidad deseada:</u> Como terapeutas necesitamos que el SAC sugiera actividades posibles según el horario del día, por ejemplo comer al medio día y a la noche. Esto permitirá anticipar situaciones comunicativas y reducir procesos de búsqueda.</p>							
Estado	Fecha	Comentario					
Definido	07/16	Fase de inicialización: #Requerimiento vinculado: Nuevo Requerimiento Esfuerzo estimado: No definido / Tipo: Nueva					
Implementando	12/16	Fase de Producción: #Requerimiento redefinidos vinculados: RF4, RF5 y RF7 Esfuerzo estimado: 28 días/ Tipo: Extendida					
Verificado	02/17	Fase de Producción: #Requerimiento redefinidos vinculados: RF4, RF5 y RF7 Esfuerzo gastado: 35 días / Tipo: Extendida					