

UNIVERSIDAD MAYOR DE SAN ANDRES
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMATICA



TESIS DE GRADO

**"APLICACIÓN UNIVERSAL DE WINDOWS PARA LA ENSEÑANZA
DEL LENGUAJE DE SIGNOS DE BEBÉ"**

PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA
MENCIÓN: INGENIERÍA DE SISTEMAS INFORMÁTICOS

POSTULANTE : CHRISTOFER TOLEDO LUNA
TUTORA METODOLOGICA : Lic. MENFY MORALES RIOS
ASESORA : Lic. BRIGIDA CARVAJAL BLANCO

LA PAZ – BOLIVIA

2015



**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA**



LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.

DEDICATORIA

A Dios por iluminar mi camino por darme fortaleza cuando más lo necesitaba, por no dejarme caer, por hacerme el hombre que soy hoy.

A mi madre, espero que esta tesis sea solo una de las muchas alegrías y satisfacciones que quiero darte, a ti que siempre estas para apoyarme, que me muestras que el amor de madre no tiene límite.

A ti, que de alguna manera, por algún motivo estás leyendo esta tesis que escribí, muchísimas gracias, me siento honrado de que dediques parte de tu tiempo a conocer mi trabajo, espero que sea de gran ayuda para ti.

A todas aquellas personas que me apoyaron en mi vida.

AGRADECIMIENTOS

A Dios por darme la fortaleza necesaria para seguir adelante, por iluminarme y no dejarme caer, por todo lo que ha traído a mi vida, que de una u otra manera me impulso hacia adelante, por cada día de vida y nueva oportunidad. Gracias Dios por enseñarme a pescar y no haberme dado un solo pez.

A mi familia, por entenderme, por el cariño incondicional, por soportarme en mis malos momentos, que aun a pesar de todo están en mi vida. A ti mamita querida Margarita Luna Humerez, que incondicionalmente confiaste en mí y me has apoyado todos los días de mi vida, tu que eres mi mayor ejemplo e inspiración, por demostrarme el valor del esfuerzo y el amor a la familia y que con dedicación todo es posible en esta vida. A ti papá David Toledo Valencia, por despertar en mí la curiosidad que me ha impulsado cada día a querer saber más, por enseñarme y cultivarme en mente y espíritu. Estoy muy agradecido con ambos por no haberme dado todo lo que quise, más si lo que necesitaba. A mis hermanos Reyna y Víctor, que me han cuidado de que tengo memoria y que a pesar de cualquier mal momento que pueda suceder siempre están presentes para brindarme su apoyo, compañía y amor.

A mi asesora: Lic. Brígida Carvajal y mi tutora: Lic. Menfy Morales. No saben cuan agradecido estoy con ustedes, por su tiempo y dedicación para que esta tesis haya salido adelante. Muchísimas gracias, por todas sus observaciones, sugerencias, recomendaciones e ideas, nunca olvidare todo lo que me inculcaron, gracias por su apoyo y guía en esta última etapa de mi vida universitaria.

A mi grupo de estudios, gracias por dejarme ser uno más de ustedes y ayudar a despertar más y mejores habilidades en mí. Jorge Cupi, Yolger Pérez, Rodrigo Quispe, Edwin Moldez, D'jalmar Gutiérrez, Daniel Pedernera. Gracias compañeros, pero sobre todo amigos.

RESUMEN

Esta tesis propone la creación de una aplicación cuyo objetivo es el de enseñar el lenguaje de signos de bebé a padres de familia y/o personas interesadas en este medio de comunicación gestual a través de la práctica mediante la utilización del sensor Kinect V2, que cuente con material como figuras y cartas de los signos en formato digital en la aplicación y que además realice un seguimiento del aprendizaje a través de registros de signos realizados en las practicas. Para llevar a cabo la aplicación se utilizó C# como lenguaje de programación y XAML como lenguaje de diseño para las plataformas móviles, Azure Mobile Service que es un servicio de la plataforma de Microsoft Azure el cual actúa como backend de la aplicación, y es a través de este servicio que todos los datos de la aplicación pasaran a alojarse en SQL Azure, una base de datos relacional en la Nube de Azure, permitiendo que los datos de los usuario esté disponible en cualquier momento.

ABSTRACT

This thesis proposes the creation of an application whose goal is to teach baby sign language to parents or people interested in this form of gestural communication through practice using the Kinect sensor V2, that has material as figures and letters of the signs in digital format in the application and also to track learning records through signs made in the practices. To carry out the application used C # programming language and XAML as a language of design for mobile platforms, Azure Mobile Service is a service platform Microsoft Azure which acts as a backend application, and it is through this service all pass application data to SQL Azure stay in a relational database in the cloud Azure, allowing the user data is available at any time.

Índice

DEDICATORIA	1
AGRADECIMIENTOS	2
RESUMEN	3
ABSTRACT	4
CAPITULO 1	13
INTRODUCCION	13
1.1. Introducción	13
1.2. Antecedentes	14
1.2.1. Baby signing Lite.....	14
1.2.2. Baby Sign and Learn.....	15
1.2.3. Baby sign Language.....	15
1.3. Planteamiento del Problema.....	16
1.3.1. Formulación del problema.....	17
1.4. Objetivos	17
1.4.1. Objetivo principal	17
1.4.2. Objetivos Secundarios	17
1.5. Justificación.....	18
1.5.1. Justificación técnica	18
1.5.2. Justificación social	18
1.5.3. Justificación académica	18
1.6. Alcances y limites	19
1.6.1. Alcances.....	19
1.6.2. Limites	19
1.7. Aportes	19
1.8. Herramientas	20
1.8.1. Software	20
1.8.2. Hardware.....	20
1.8.3. Plataformas	20

1.9.	Metodología	21
1.9.1.	Metodología de Investigación.....	21
1.9.2.	Metodología de Desarrollo	21
CAPITULO 2		23
MARCO TEORICO		23
2.1.	Lengua de Signos de Bebé	23
2.2.	Lengua de Signos americana.....	23
2.3.	Aplicación Móvil.....	24
2.3.1.	Aplicación Universal de Windows	24
2.4.	Model View Controller (MVC).....	25
2.5.	Model View ViewModel (MVVM).....	26
2.6.	Dispositivo Kinect V2 para Windows.....	27
2.6.1.	Especificaciones técnicas.....	28
2.7.	Microsoft Azure.	29
2.7.1.	Azure Mobile Service.	29
2.7.2.	SQL Azure.	29
2.7.3.	Azure Storage Service	31
2.8.	Desarrollo móvil Ágil (MAD)	33
2.8.1.	Introducción a MAD.....	33
2.8.2.	De donde viene MAD?	33
2.8.3.	Estaciones MAD.....	33
2.8.3.1.	Estudio de Mercado.....	33
2.8.3.2.	Elección de una arquitectura, patrones y tecnologías para la aplicación	34
2.8.3.3.	Definición de módulos de la aplicación	35
2.8.4.	Porque escoger una metodología ágil para el desarrollo móvil?	37
CAPITULO 3		39
MARCO APLICATIVO.....		39
3.1.	Estaciones de MAD.....	39
3.1.1.	Estudio de aplicaciones disponibles en el mercado	39

3.2.	Elección de tecnologías y patrones de desarrollo.....	40
3.2.1.	Lenguaje de Programación	40
3.2.2.	Entorno de desarrollo.....	42
3.2.3.	Elección de un patrón de diseño o arquitectura	42
3.2.4.	Base de Datos.....	43
3.2.5.	Presupuesto.	45
3.2.6.	Conclusión sobre las tecnologías utilizadas.....	47
3.3.	Definición y desarrollo de segmentos	48
3.4.	Modelado de datos.....	53
3.5.	Desarrollo de la interfaz	55
3.6.	Desarrollo de módulos.....	60
3.6.1.7.	Azure Mobile Service.....	72
3.6.1.8.	Creación de formulario para obtener retroalimentación.	73
3.6.1.9.	Creación del Mobile Service	74
3.7.	Prueba Completa	81
3.7.1.	Planeación de la siguiente actualización.....	81
3.8.	Publicación de la aplicación.....	82
3.8.1.	Definición de un modelo de negocio.	82
3.8.2.	Establecimiento de precios en base al modelo o modelos de negocio escogidos	83
3.8.3.	Publicación de la aplicación en la tienda de Windows.	84
3.8.4.	Publicación de la Aplicación.	87
CAPITULO 4		89
PRUEBAS Y RESULTADOS.....		89
4.1.	Pruebas.	89
4.2.	Análisis de los comentarios de los usuarios.	93
4.2.1.	Obtención de estadísticas en base al Test de Usabilidad.	94
4.3.	Pruebas de funcionalidad y certificación.	99
4.3.1.	Realización de la prueba.	101

CAPITULO 5	104
CONCLUSIONES Y RECOMENDACIONES	104
5.1. Conclusiones	104
5.1.1. Objetivos secundarios.	104
5.2. Recomendaciones.....	105
5.2.1. Migrando a nuevas tecnologías.....	105
5.2.2. Expansión a nuevos mercados	105
CAPITULO 6	107
BIBLIOGRAFIA	107
6.1. Referencias Bibliográficas	107
CAPITULO 7	110
ANEXOS	110
ANEXO A.	110
ANEXO B.	111
ANEXO C.	112
DOCUMENTACION.....	113

Índice de figuras

Figura 2.1. Signo equivalente a MIEDO	23
Figura 2.2. Signo equivalente a MAMÁ	23
Figura 2.3. Arquitectura de una Aplicación Universal de Windows.....	25
Figura 2.5. Estructura del Patrón MVVM	26
Figura 2.6. Flujo de captura de Data del Kinect V2.	28
Figura 2.7. Parte del Kinect V2	28
Figura 2.8. Arquitectura de Azure Mobile Services (Microsoft Azure, 2015).....	29
Figura 2.9. Estructura de un Blob. (Microsoft Azure, 2015).....	32
Figura 3.1. Flujo completo y offline de la Aplicación.....	48
Figura 3.2. Áreas del Diccionario/Agrupación Alfabética	49

Figura 3.3. Áreas del Diccionario.....	49
Figura 3.4. Áreas del módulo de preferencias de signos	50
Figura 3.5. Áreas del módulo de práctica de signos	51
Figura 3.6. Áreas del Historial de aprendizaje	51
Figura 3.7. Áreas del formulario de retroalimentación.	52
Figura 3.8. Diagrama de navegación de la Aplicación.....	53
Figura 3.9. Clases principales para el funcionamiento de la Aplicación.....	54
Figura 3.10. Pantalla principal de la aplicación.....	55
Figura 3.11. Menú principal de la aplicación.	56
Figura 3.12. Inicio de sesión en la red social Facebook.	56
Figura 3.13. Diccionario de signos.....	57
Figura 3.14. Agrupación de los signos de preferencia del usuario.....	57
Figura 3.15. Lecciones disponibles en la aplicación	58
Figura 3.16. Módulo de reconocimiento y práctica de Signos	58
Figura 3.17. Historial de Practica de signos.	59
Figura 3.18. Formulario de retroalimentación.....	59
Figura 3.20. Creación de un Proyecto Azure Mobile Service	74
Figura 3.21. Configuración de Proyecto Mobile Service	75
Figura 3.22. Ingreso de credenciales de la cuenta de Azure.....	75
Figura 3.23. Configuración de propiedades del Mobile Service	76
Figura 3.24. Estructura del proyecto BabySigns.MobileService.....	76
Figura 3.25. Descargas de la aplicación en las últimas 2 semanas.....	85
Figura 3.26. Estadísticas de Mercado y de grupos de interés.....	86
Figura 3.27. Grafica acumulativa de descargas de la Aplicación.....	86
Figura 3.28. Mercados que han descargado la Aplicación (Aplicación Dev Center)	87
Figura 4.1. Madre Practicando Signo Leche (Aplicación)	90
Figura 4.2. Madre Practicando Signo Gato	91
Figura 4.3. Padre Practicando Signo Coma (Aplicación).....	91
Figura 4.4. Padre Practicando Signo Coma.....	92

Figura 4.5. Padres de Familia Probando la Aplicación	92
Figura 4.6. Cumplimiento de expectativas de la App.....	94
Figura 4.7. Aprendizaje de Lengua de Signos.....	94
Figura 4.8. Efectividad de la Aplicación	95
Figura 4.9. Utilidad de la App	95
Figura 4.10. Recomendación de la App a otros usuarios.	96
Figura 4.11. La Aplicación es confusa?	96
Figura 4.12. Frecuencia de despliegue de mensajes de error en la Aplicación.	97
Figura 4.13. Interacción con el Kinect	97
Figura 4.14. Esfuerzo necesario para aprender a utilizar la aplicación.	98
Figura 4.15. Frecuencia de comportamiento inesperado en la aplicación.....	98
Figura 4.16. Desempeño de la Aplicación.....	99
Figura 4.17. Creación del paquete de Aplicación y Lanzamiento de pruebas.....	101
Figura 4.18. Selección de pruebas a realizarse.	101
Figura 4.19. Validación de la aplicación a partir de las pruebas realizadas	102
Figura 4.20. Finalización de las pruebas del Kit.	102

Índice de Segmentos de Código

Segmento de Código 3.1. Creación de un LiveTile.....	61
Segmento de Código 3.2. Atributos de un UserControl.....	62
Segmento de Código 3.3. Almacenamiento aislado de una variable.....	63
Segmento de Código 3.4. Métodos de autenticación para una sesión en Facebook	65
Segmento de Código 3.5. Inserción de Usuario mediante el Mobile Service	67
Segmento de Código 3.6. Método de cargado de signos en la vista.....	68
Segmento de Código 3.7. Selección de una seña para su despliegue visual	68
Segmento de Código 3.8. Inserción de grupo de signos a través del Mobile Service	69
Segmento de Código 3.9. Carga de recursos del Kinect y de archivo de reconocimiento ...	70
Segmento de Código 3.10. Método de Captura de Frames	71

Segmento de Código 3.11. Declaración de la instancia cliente.....	72
Segmento de Código 3.12. Asignación de llaves a la instancia Cliente.....	72
Segmento de Código 3.13. Inserción de un registro.....	73
Segmento de Código 3.14. Eliminación de un registro	73
Segmento de Código 3.15. Actualización de un registro	73
Segmento de Código 3.16. Entidad APPUser	77
Segmento de Código 3.17. Entidad CustomLesson	77
Segmento de Código 3.18. Selecccion del tipo de Controlador.....	78
Segmento de Código 3.19. Creación del Controller de la Entidad APPUser.....	78
Segmento de Código 3.20. Controller de la Entidad APPUser	79
Segmento de Código 3.21. Controller de APPUser en el Context del Servicio.....	79
Segmento de Código 3.22. Creación de un Custom Controller	80
Segmento de Código 3.23. Custom Controller - Inserción de Usuario	80
Segmento de Código 3.24. Aplicación subida en el Store de Windows	85

Índice de Tablas

Tabla 1.1. Baby signing Lite Características y desventajas	14
Tabla 1.2. Baby sign and Learn Características y desventajas.....	15
Tabla 1.3. Costos de kits de enseñanza del producto Baby sign Language.	15
Tabla 1.4. Baby sign Language Características y desventajas	16
Tabla 1.5. Etapas del Método Sistemico	21
Tabla 1.6. Estaciones M.A.D.....	22
Tabla 2.1. Ventajas y desventajas del Patrón MVC	26
Tabla 2.2. Ventajas y desventajas del patrón MVVM.....	27
Tabla 2.3. Especificación técnica del Kinect V2.....	28
Tabla 2.4. Aspectos útiles de SQL Azure.....	30
Tabla 5. Componentes de Azure BLOB.....	32
Tabla 3.1. Ventajas y desventajas de Lenguajes de programación.	41

Tabla 3.2. Costes de equipos para el desarrollo del proyecto.....	46
Tabla 3.3. Coste de Licencia para el desarrollo del proyecto.....	46
Tabla 3.4. Coste total del proyecto	46
Tabla 3.5. Pasos para la publicación de la aplicación en las Tiendas de Windows	88
Tabla 4.1. Dispositivos móviles de involucrados en la pruebas.....	89
Tabla 4.2. Requerimientos de hardware de la Aplicación.....	89
Tabla 4.3. Pruebas realizadas en aplicaciones de la Tienda Windows.....	100
Tabla 4.4. Resultados del Kit de pruebas funcionalidad y certificación.	103
Tabla 7.1. Requerimientos de hardware y software del Kinect V2.....	110
Tabla 7.2. Principales aplicaciones para dispositivos móviles.....	111
Tabla 7.3. Análisis principales de aplicaciones y/o productos en el mercado.....	112

1.1.Introducción

A medida que los bebés crecen y van desarrollando sus diferentes habilidades, también crece su necesidad de poder expresar, comunicar y hacer conocer sus deseos. Sin embargo aún están limitados por su poco o nulo desarrollo del habla, ya que tanto su comprensión del lenguaje como sus habilidades motoras se desarrollan mucho más rápido que su capacidad de hablar. (PsicoPedagogia, 2014)

La idea de seguir explorando esta habilidad natural de los bebés y enseñarles un vocabulario de signos lleva a las Doctoras Linda Acredolo y Susan Goodwyn, después de una investigación de más de 1 década, desarrollar el programa Baby Signs, este programa ayuda a desarrollar las habilidades de comunicación gestual del bebé a través de signos naturales que estos realizan, además, también toma los signos más importantes de la lengua de signos americana que podría usar el bebé. (Baby Signs, 2015)

Puesto que los bebés aprenden sus primeras habilidades de comunicación natural como las gestuales y/o sonoras imitando y escuchando a sus padres, se requiere que los padres de familia aprendan previamente otras habilidades de comunicación antes de impartirlas o establecerlas con sus hijos. (Healthy Children, 2014)

Es por tanto que se propone desarrollar una Aplicación Universal de Windows para enseñar el lenguaje de signos de bebe a actuales o también a futuros padres, que proporcione el contenido educativo en formato digital como ser imágenes de los signos, ilustraciones de lo representado por el signo y otros, para que a futuro cuenten con el conocimiento para lograr enseñar y establecer esta habilidad de comunicación gestual con sus bebés, que además realice un seguimiento sobre sus avances y que esta información sea almacenada en una base de datos en la nube a través de los servicios provistos por Microsoft Azure para que pueda estar disponible tanto en su dispositivo móvil como en su computadora y/o Tablet que cuente

con los sistemas operativos Windows Phone 8.1 y Windows 8.1 respectivamente, además que proporcione características innovadoras como el de poder usar características que provee el Kinect V2 para reconocer una cantidad limitada de signos.

1.2. Antecedentes

En la actualidad existen varias aplicaciones en relación al uso de lenguaje de signos de bebé para las plataformas móviles Android y Windows Phone; y ninguna para la plataforma de Windows, además se puede observar que ninguna de estas proveen ni cuentan con características y/o servicios extras que permitan realizar seguimiento del aprendizaje del usuario.

A continuación veremos algunos casos de aplicaciones y/o productos de compañías que brindan una funcionalidad similar a la aplicación que se propone en esta tesis.

1.2.1. Baby signing Lite.

Esta aplicación es una versión gratuita de la aplicación completa de paga para dispositivos con Sistema Operativo Android.

Costos.

La aplicación es gratuita.

Tabla 1.1. Baby signing Lite Características y desventajas

Características	Desventajas
<ul style="list-style-type: none"> • Ofrece instrucciones sobre cómo hacer que los signos, así como consejos de memoria para ayudarlo a recordar. • El usuario puede añadir signos a una lista de favoritos para un acceso más rápido y/o inmediato. 	<ul style="list-style-type: none"> • No lleva registro del avance del bebé. • Solo incluye 50 tipos de signos de los más de 400 signos tomados de la lengua de signos americana y que son usados actualmente. <p>(Baby Signs®, (s.f.))</p>

Fuente: (Aplicacion Baby signing Lite, 2015)

1.2.2. Baby Sign and Learn.

Esta aplicación utiliza lenguaje de señas americano (ASL) signos de palabras clave y contiene más de 300 signos para enseñar a su bebé. Está disponible para dispositivos con Sistema Operativo Android

Costos.

Es precio de la aplicación es de \$ 2.6.

Tabla 1.2. Baby sign and Learn Características y desventajas

CARACTERISTICAS	DESVENTAJAS
<ul style="list-style-type: none">• Demostración del cómo realizar correctamente los signos a través de videos animados.• Realiza exámenes de selección correcta de los signos.	<ul style="list-style-type: none">• No lleva registro del avance del bebé.• Aplicación muy básica, en el sentido de que presenta los signos de forma cruda• sin embargo, todas estas aplicaciones evidencian de manera inmediata una estructura de catálogo como se puede observar en sus respectivas descripciones e imágenes de muestra

Fuente: (Aplicacion Baby Sign and Learn, 2015)

1.2.3. Baby sign Language.

BabySignLanguage.com es un proyecto para los padres que enseñan bebés y niños pequeños temprana habilidades de comunicación. (*Baby Sign Language*).

Tabla 1.3. Costos de kits de enseñanza del producto Baby sign Language.

COSTOS
<ul style="list-style-type: none">• Ultra Kit \$202.87• Premium Kit \$162.91.• Delux Kit \$95.95• Standart Kit \$74.95• Flash Cards \$24.99• Diccionario \$19.99

Fuente: (Proyecto BabySignLanguage.com, 2015)

Tabla 1.4. Baby sign Language Características y desventajas

CARACTERISTICAS	DESVENTAJAS
<ul style="list-style-type: none"> • Uso de tarjetas con ilustraciones de los signos. • Uso de DVDs con material multimedia. • Uso de una guía de aprendizaje con consejos. • Cuadros en las paredes como ayuda. 	<ul style="list-style-type: none"> • No existen recursos como libretas de registro que sirvan para llevar un seguimiento del aprendizaje del bebé. • Se puede observar que el costo de cada Kit de Aprendizaje es elevado, tomando como caso Bolivia, donde que el salario mínimo nacional es de Bs. 1440, 210 \$ Aproximadamente.

Fuente: (Proyecto BabySignLanguage.com, 2015)

1.3.Planteamiento del Problema

Investigaciones como el de las Doctoras Linda Acredolo y Susan Goodwyn apoyada por el Instituto Nacional de Saludo de USA, como el de la Licenciada Chamarrita Farkas de la Pontifica Universidad de Católica de Chile, entre otros, tuvieron como resultado que la enseñanza del lenguaje de signos de bebe en bebés, como medio para establecer canal de comunicación gestual entre los padres y sus bebés, a su vez esto brinda grandes efectos positivos tanto en los padres como en sus hijos, como un claro aumento del Lenguaje tanto expresivo como comprensivo del bebé, así también como una mayor interacción entre las madres y sus hijos, por mencionar algunos de los beneficios. (Farkas, 2009) (Vallotton, 2014)

Sin embargo cabe destacar que el aprendizaje de esta lengua debe iniciar en los padres o incluso personas particulares que deseen establecer este medio de comunicación gestual con los infantes, pues los bebés adquieren sus habilidades de comunicación a través de la interacción con sus padres y/o personas de su entorno.

En relación a la enseñanza de la lengua de signos para bebé, se puede observar los siguientes escenarios:

- Centros de capacitación que ofrecen talleres de enseñanza a padres acerca del lenguaje de signos de bebé.
- Compañías online que ofrecen kits de enseñanza de lengua de signos de bebé que contienen material didáctico, libros, fichas, videos, cuadros.

- Aplicaciones móviles que únicamente contienen las ilustraciones de signos de la lengua de signos de bebé

Se puede constatar que tanto el precio de los talleres como del kit de enseñanza es elevado, además que estos sistemas de aprendizaje no cuentan con alguna herramienta o libreta de registro para hacer seguimiento al aprendizaje del usuario sobre estos signos.

La necesidad de desarrollar una aplicación universal de Windows para todos los dispositivos con Sistema Operativo Windows radica en los siguientes puntos:

Brindar una aplicación que pueda ser descargada para su uso por cualquier persona que cuente con un dispositivo con sistema operativo Windows 8.1 y Windows Phone 8.1.

1.3.1. Formulación del problema.

¿Cómo se puede enseñar el lenguaje de signos de bebé a los padres de familia?

1.4. Objetivos

1.4.1. Objetivo principal

Desarrollar una aplicación para enseñar el lenguaje de signos de bebe a padres de familia y/o también a personas interesadas en este medio de comunicación gestual.

1.4.2. Objetivos Secundarios

- Buscar antecedentes de proyectos previos incluyendo tesis que hayan tocado el mismo tema y/o productos que ofrezcan el servicio.
- Definir los requerimientos de la aplicación para establecer los límites y alcances de la misma.
- Construir una aplicación interactiva que también que utilice características que provee el Kinect V2 como movimientos y gestos.
- Realizar un seguimiento a los padres sobre su aprendizaje de los signos por medio de la aplicación a través de registros de avance.
- Aplicar una metodología ágil para al desarrollo de la aplicación.
- Realizar pruebas de funcionalidad de la aplicación acordes a la metodología ágil aplicada en el desarrollo de la aplicación.

1.5. Justificación

1.5.1. Justificación técnica

Contar con este tipo de aplicación en la cual se tendrán todos los recursos correspondientes para enseñar al padre de familia, hace posible el no tener que cargar con libros, videos, fichas y todo lo relacionar a métodos interactivos de aprendizaje, además que se hará el seguimiento y control del aprendizaje del padre en la misma aplicación, además que el padre podrá acceder a estos recursos e información de sus avances desde un Smartphone y PC/Laptop con sistemas operativos Windows Phone 8.1 y Windows 8.1 respectivamente.

1.5.2. Justificación social

Dado el alto impacto social y el beneficio que representa para familias con bebés (mayores a 6 meses), se ve necesaria la implementación de una “Aplicación para la Enseñanza del lenguaje de signos de bebé”, la cual permitirá a padres aprender el lenguaje de signos de bebé, de esta manera conocer otra habilidad de comunicación que podrá transmitir y/o enseñar de manera natural a sus propios bebés antes del desarrollo del habla de los mismos. Al ser este un método de comunicación gestual natural que se produce en los bebés a partir de los 6 - 10 meses, su implementación a corto plazo en la vida diaria del bebé es mucho más fácil, factible y adecuada a medida que crece la necesidad que tienen los bebés y niños pequeños de comunicar sus necesidades y deseos; y no logran hacerlo con claridad ya que su capacidad cognitiva no es suficiente en los primeros meses y años de vida.

1.5.3. Justificación académica

Se vio la importancia de que bebés y niños pequeños desarrollen desde temprana edad, habilidades de comunicación, ya que la comunicación temprana establece las bases para el aprendizaje acelerado, revela cuan inteligentes son los bebés, estimula el desarrollo intelectual y ayuda a los bebés a aprender a hablar. (Martin/Pitt, Partnership for Children)

1.6. Alcances y límites

1.6.1. Alcances

Para cumplir con el objetivo general del presente proyecto, la aplicación de Windows 8.1 contara con los siguientes Módulos.

- Diccionario de signos.
- Preferencia de signos.
- Historial de aprendizaje del Usuario.
- Practica y reconocimiento de signos utilizando el Kinect v2.

1.6.2. Límites

- La aplicación está disponible para su descargar, en la tienda de Windows y también en la tienda de Windows Phone.
- El Modulo de reconocimiento y practica de signos se limita a personas que cuenten con el Kinect V2 y una PC que tenga las características presentadas en la Tabla 6.3 de la sección de Anexo.
- Si el tesista agregase más signos que sean reconocidos por el Kinect, será necesario que los usuarios actualicen la aplicación a través de la tienda de Windows y Windows Phone.

1.7. Aportes

En el proceso de diseño y desarrollo de la presente tesis, se implementó diversas librerías para:

- La conexión con Microsoft AZURE además de las respectivas operaciones CRUD, pueden ser portadas y utilizadas en cualquier otro proyecto que implemente Azure Mobile Services y Azure Storage Blobs.
- Implementación de librería para el manejo de archivos de reconocimiento de signos de Visual Gesture Builder con Kinect V2.

1.8.Herramientas

1.8.1. Software

- Microsoft Windows 8.1 Pro, 64 bits Edition (Sistema operativo del equipo de desarrollo)
- Microsoft Visual Studio 2013 Professional Edition (Entorno de desarrollo)
- Notepad++ (Editor de código)
- Azure Storage Explorer (Visor y editor *de tablas de Windows Azure*)
- Microsoft Azure (Plataforma de servicios *de computación en la nube*)
- Windows Phone *8.1* SDK (Set de herramientas de *desarrollo para Windows Phone*)
- Windows *8.1* SDK (Set de herramientas de desarrollo para Windows *8.1*)
- ***Kinect SDK v2.0 Public Preview.***

1.8.2. Hardware

El equipo para el desarrollo de la aplicación:

- Laptop HP ENVY dv6.

Los equipos para realizar las pruebas de la aplicación.

- Laptop HP ENVY dv6.
- Nokia Lumia 820.
- Nokia Lumia 920.
- Nokia Lumia 1020.

Sensor de Reconocimiento:

- Kinect for Windows V2.

1.8.3. Plataformas

El presente proyecto, siendo una Aplicación Universal de Windows, está disponible en las siguientes plataformas.

- Windows 8.1
- Windows Phone 8.1

1.9. Metodología

1.9.1. Metodología de Investigación

Para la investigación se utilizó el Método de la Investigación Sistémica.

El Método Sistémico es uno de los instrumentos lógicos más contemporáneos en el ámbito de la metodología, orientado a la percepción holística (total) de la realidad de donde se extraerá la propia problemática y las soluciones correspondientes. (Escuela Superior de la Fuerza Aérea del Perú)

El Enfoque Sistémico es para la Informática una herramienta conceptual y de acción, interpreta lo concreto y facilita el pasaje de la teoría a la práctica, logrando mejores resultados. Si bien el enfoque sistémico no es privativo de la Informática, es en ella donde adquiere, como soporte metodológico, su máxima validez. (Graciela E. Barchini, 2004)

El enfoque sistémico tiene tres etapas que son las siguientes:

Tabla 1.5. Etapas del Método Sistémico

ETAPAS
Identificar un todo contenedor del cual el objeto a explicar es una parte
Explicar el comportamiento o propiedades del todo contenedor.
Explicar entonces el comportamiento o las propiedades del objeto por explicar en términos de su función dentro de su todo contenedor.

Fuente: (Universo Sistemico para Ingenieros de Sistemas, 2015)

1.9.2. Metodología de Desarrollo

Para la parte de desarrollo de la aplicación se ha utilizado la metodología ágil MAD, basada ligeramente en la metodología MASAM, esta metodología está orientada al desarrollo de aplicaciones móviles basado en la Metodología MASAM para el diseño, implementación y desarrollo de aplicaciones. (Cupi Veliz, 2014).

A continuación se detallan las estaciones de MAD y los objetivos de cada estación.

Tabla 1.6. Estaciones M.A.D.

Estaciones MAD	Objetivos
Estudio de aplicaciones en el Mercado	<ul style="list-style-type: none"> • Análisis de aplicaciones similares existentes • Análisis de requerimientos por usuarios • Definición de características • Plataformas a las que apuntamos • Conclusiones sobre el estudio realizado
Elección de tecnologías y patrones de desarrollo	<ul style="list-style-type: none"> • Se realiza la búsqueda de una arquitectura que se acomode a las necesidades de la aplicación, en términos de dimensión de la aplicación, alcances, y escalabilidad.
Definición y desarrollo de segmentos de la aplicación	<ul style="list-style-type: none"> • Jerarquización de los segmentos en base a su prioridad de desarrollo • Modelado de datos • Diseño de la interfaz
Prueba completa de la aplicación.	<ul style="list-style-type: none"> • Encontrar fallas dentro de los segmentos independientes.
Publicación de la aplicación	<ul style="list-style-type: none"> • Definición de un modelo de negocio • Establecimiento de precios en base al modelo o modelos de negocio escogidos • Publicación de la aplicación • Obtención de retroalimentación, • Planeando la siguiente actualización.

Fuente: (Cupi Veliz, 2014)

2.1. Lengua de Signos de Bebé

La lengua de signos para bebés, es en realidad el desarrollo de la comunicación gestual natural que se produce en los bebés a partir de los 6-10 meses, en el sentido de que se utilizan signos que los bebés realizan de manera espontánea o naturalmente para expresar alguna necesidad, además que se incluyen otros signos tomados de la lengua americana de signos para que puedan ser enseñados a bebés antes de su desarrollo del habla, de esta manera se impulsan las habilidades de comunicación de los bebés. (BabySignsToo, 2014)



Figura 2.1. Signo equivalente a MIEDO (Baby sign language, 2014)

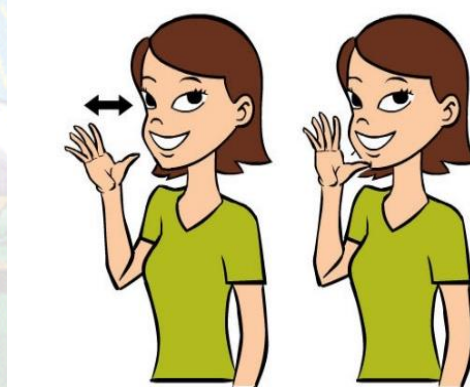


Figura 2.2. Signo equivalente a MAMÁ (Baby sign language, 2014)

2.2. Lengua de Signos americana.

La lengua de signos es la lengua natural de las personas sordas. Mediante ella pueden relacionarse con su entorno social pues les permite establecer un canal de comunicación básico que es visual y espacial.

Conviene aclarar que la lengua de signos no es simple mímica y tampoco es una reproducción visual de alguna versión simplificada de lengua oral. Tiene una estructura gramatical rica y propia caracterizada por la configuración de las manos, de sus movimientos, de sus orientaciones, de su ubicación espacial y de los elementos no manuales como son los

movimientos labiales, faciales, linguales, etc. Como cualquier otra lengua se puede usar para discutir cualquier tema, ya sea sencillo y concreto o denso y abstracto. Además, al igual que la lengua oral, se organiza por unidades elementales sin significado. (González, 2001)

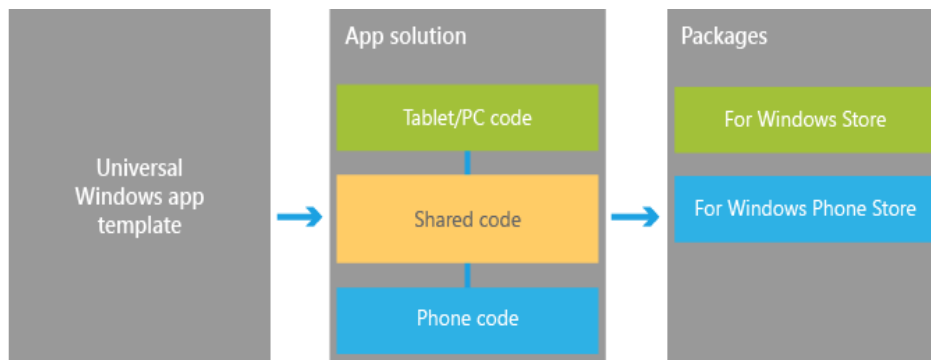
2.3. Aplicación Móvil

Las aplicaciones también llamadas Apps, es simplemente un programa informático creado para llevar a cabo o facilitar una tarea en un dispositivo informático. Cabe destacar que aunque todas las aplicaciones son programas, no todos los programas son aplicaciones. Existe multitud de software en el mercado, pero sólo se denomina así a aquel que ha sido creado con un fin determinado, para realizar tareas concretas. No se consideraría una aplicación, por ejemplo, un sistema operativo, ni una suite, pues su propósito es general. (AppDesingBook, 2015)

2.3.1. Aplicación Universal de Windows

Aplicación Universal de Windows implica una convergencia en las plataformas Windows Phone 8.1 y Windows 8.1, desde el punto de vista del desarrollador esto significa que ahora se puede construir aplicaciones universales que funcionan tanto en Smartphone como en tabletas y PC mediante el intercambio de una cantidad significativa de código y contenido, puesto que Windows Phone 8.1 ejecuta sus aplicaciones con un motor cada vez más parecido al de Windows 8.1, así cada día los dos sistemas operativos están más unidos y eso favorece el desarrollo, porque ahora, se puede compartir incluso más de un 80% del código.

Las versiones actuales de Visual Studio incluyen una plantilla de aplicación universal de Windows que permite crear en un mismo proyecto una aplicación de la Tienda Windows y una aplicación de la Tienda de Windows Phone. Cuando hayas terminado, podrás generar los paquetes de la aplicación para la Tienda Windows y la Tienda de Windows Phone mediante una única acción y tu aplicación llegará a los usuarios de cualquier dispositivo de Windows. (Centro de Desarrollo de Windows, 2015)



*Figura 2.3. Arquitectura de una Aplicación Universal de Windows
(Centro de Desarrollo de Windows, 2015)*

Dicho de otra manera más simple, ahora no solo se pueden establecer núcleos de código lógico, si no establecer también núcleos de interfaces de desarrollo. Con esto, el concepto de reutilización de código fuente va quedando atrás, dando paso ahora al concepto de unificación de código de desarrollo y diseño, lo cual tiene como objetivo reducir el tiempo y dinero invertido en desarrollar aplicaciones móviles que corran bajo diferentes plataformas.

Para que una App pueda ser instalada en un dispositivo móvil, primero tendrá que estar publicada en la tienda de aplicaciones del sistema operativo sobre el cual funciona el dispositivo, en este caso la aplicación desarrollada para la presente tesis, estará alojada en Windows Store y Windows Phone Store. (Cupi Veliz, 2014)

2.4. Model View Controller (MVC)

Es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. En líneas generales, MVC es una propuesta de diseño de software utilizada para implementar **sistemas donde se requiere el uso de interfaces de usuario**. Surge de la necesidad de crear software más robusto con un ciclo de vida más adecuado, donde se potencie la facilidad de mantenimiento, reutilización del código y la separación de conceptos.

En la tabla a continuación se listan algunas ventajas y desventajas de este modelo.

Tabla 2.7. Ventajas y desventajas del Patrón MVC

MVC	Ventajas	<ul style="list-style-type: none"> • Años de madurez y una comunidad inmensa detrás del proyecto, • Un sin número de frameworks que facilitan su implementación.
	Desventajas	<ul style="list-style-type: none"> • No está originalmente pensado como patrón de diseño / arquitectura para el desarrollo de aplicaciones móviles.

Fuente: (Model-View-Controller, 2015)

2.5. Model View ViewModel (MVVM)

Es un patrón de diseño de aplicaciones para desacoplar código de interfaz de usuario y código que no sea de interfaz de usuario. Con MVVM, defines la interfaz de usuario de forma declarativa (por ejemplo, mediante XAML) y usas el marcado de enlace de datos para vincularla a otras capas que contengan datos y comandos de usuario. La infraestructura de enlace de datos proporciona un acoplamiento débil que mantiene sincronizados la interfaz de usuario y los datos vinculados, y que enruta todas las entradas de usuario a los comandos apropiados.

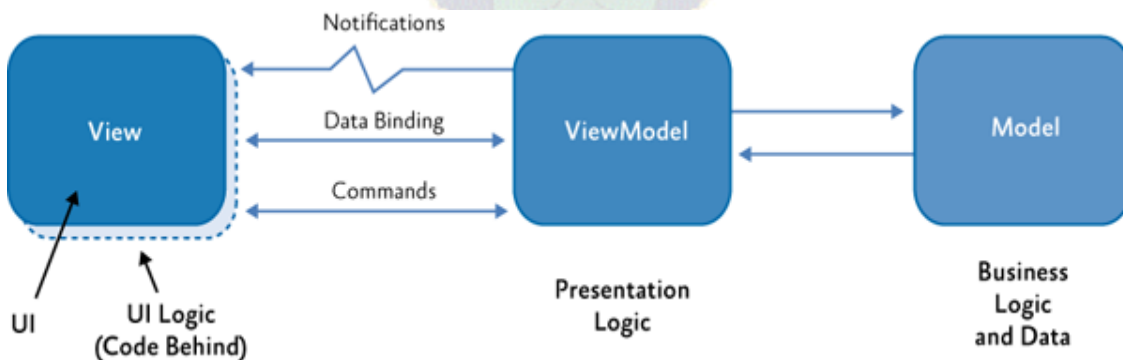


Figura 2.4. Estructura del Patrón MVVM

(Uso del patrón Model-View-ViewModel (MVVM), 2015)

El patrón MVVM organiza el código de tal forma que es posible cambiar partes individuales sin que los cambios afecten a las demás partes. Se listan las siguientes ventajas y desventajas:

Tabla 2.8. Ventajas y desventajas del patrón MVVM

MVVM	Ventajas	<ul style="list-style-type: none"> • Es la adaptación de MVC para la construcción de aplicaciones móviles. • Adapta el concepto de Binding del lenguaje de diseño XAML para una mejor conexión entre las interfaces gráficas y los datos obtenidos desde nuestras diferentes fuentes de datos. • También tiene un buen número de frameworks que facilitan su implementación, y la comunidad que opta por MVVM como patrón de diseño va haciéndose más grande.
	Desventajas	<ul style="list-style-type: none"> • El concepto de Binding no es directamente aplicable a plataformas que no utilicen XAML como lenguaje de diseño, como iOS y Android por ejemplo.

Fuente: (Uso del patrón Model-View-ViewModel (MVVM), 2015)

Por el contrario, una aplicación con una estructura más convencional utiliza el enlace de datos únicamente en controles de lista y texto, y responde a la entrada del usuario mediante el control de eventos expuestos por los controles. Los controladores de eventos están estrechamente acoplados a los controles y suelen contener código que manipula la interfaz de usuario directamente. Esto hace que sea difícil o imposible reemplazar un control sin tener que actualizar el código de control de eventos. (Uso del patrón Model-View-ViewModel (MVVM), 2015).

2.6. Dispositivo Kinect V2 para Windows.

Kinect V2 es el sensor de movimiento de Microsoft para la consola de juegos Xbox ONE. Sin embargo también se sacó a la venta una versión para Windows, el cual proporciona una interfaz de usuario natural (NUI) que permite a los usuarios interactuar intuitivamente y sin ningún dispositivo intermediario, tal como un controlador.

Además permite a desarrolladores crear aplicaciones interactivas que responden a gestos, voz y movimiento. Con el sensor v2 y SDK 2.0, puede crear aplicaciones Kinect v2 para su distribución comercial a través de la tienda de Windows, que le permite llegar a más clientes potenciales que nunca.



Figura 2.5. Flujo de captura de Data del Kinect V2. (Centro de Desarrollo de Windows, 2015)



Figura 2.6. Parte del Kinect V2

2.6.1. Especificaciones técnicas.

Las principales características se muestran en la tabla a continuación.

Tabla 2.9. Especificación técnica del Kinect V2

KINECT V2 XBOX ONE	
Cámara de color	1920x1080 / 30fps
Cámara de profundidad	512x424
Distancia Máxima	4,5 Metros
Distancia Mínima	50 cm
Campo de Visión horizontal	70 Grados
Campo de visión Vertical	60 Grados
Articulaciones del esqueleto definidas	25 Articulaciones
Cuerpos reconocidos	6
Estándar USB	3,0

Fuente: (Kinect for Windows, 2015)

2.7. Microsoft Azure.

Azure es la plataforma de informática en la nube de Microsoft, una colección cada vez mayor de servicios integrados de análisis, proceso, bases de datos, móviles, redes, almacenamiento y web. Proporciona un **entorno gestionado para la ejecución y el despliegue de aplicaciones y servicios** en la nube, además brinda a los desarrolladores un entorno de computación bajo demanda y almacenamiento alojado en los centros de datos de Microsoft para aplicaciones en la web. (Microsoft y la computación en nube, 2015)

2.7.1. Azure Mobile Service.

Microsoft Azure Mobile Service es una oferta de servicios Azure diseñado para hacer más fácil la creación de aplicaciones móviles altamente funcionales utilizando Azure. Servicios móviles reúne un conjunto de servicios de Azure que permiten capacidades de back-end para sus aplicaciones.

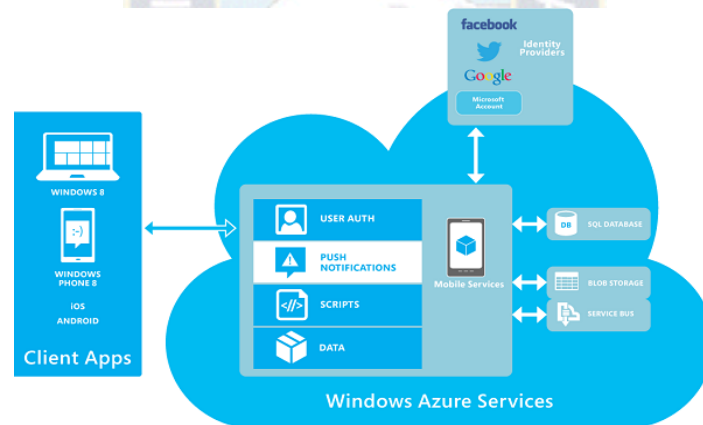


Figura 2.7. Arquitectura de Azure Mobile Services (Microsoft Azure, 2015)

2.7.2. SQL Azure.

Es parte de la plataforma Windows Azure: un conjunto de servicios que ofrecen computación, infraestructura, servicios web y servicios de datos hospedados. El componente SQL Azure ofrece toda la funcionalidad de base de datos relacional de SQL Server, pero también ofrece funcionalidad como servicio de cloud computing, hospedado en centros de datos de Microsoft en todo el mundo.

SQL Azure generalmente viene bien siempre que necesite servicios de base de datos. Existen algunos aspectos en los que SQL Azure sería especialmente útil, estos se detallan en la tabla a continuación.

Tabla 2.10. Aspectos útiles de SQL Azure

Colaboración	Al pasar los datos a la nube, SQL Azure ayuda a crear un concentrador de colaboración. Cuando necesite compartir acceso a información entre límites organizativos, como por ejemplo, entre sucursales, los servicios hospedados permiten reducir los obstáculos en la red, a la vez que aseguran la protección de los datos.
Escala	Uno de los beneficios inherentes de cloud computing es la capacidad de agregar infraestructura a petición, lo que permite escalar dinámicamente la capacidad de la aplicación a medida que surjan necesidades. Esto es particularmente útil cuando el uso máximo de las aplicaciones hospedadas cambia de manera significativa a lo largo del tiempo.
Concentración	A medida que la funcionalidad de base de datos relacional se hace más granular y se distribuye a departamentos y grupos de trabajo individuales, la administración interna de la “expansión de base de datos” puede convertirse en un problema. SQL Azure permite consolidar bases de datos de departamentos y grupos trabajo, aprovisionamiento llave en mano y administración simplificada, de manera que los administradores puedan satisfacer las necesidades de los departamentos más fácilmente.
Aplicaciones hospedadas	Si ya hospeda otras cargas de trabajo en la plataforma Windows Azure (como aplicaciones web), la integración de cargas de trabajo de bases de datos relacionales en SQL Azure es muy simple. También ayudan a maximizar el rendimiento de la aplicación reduciendo el tráfico de red.
Eficacia de costos	Según las cargas trabajo que desee instalar, el modelo de pago por uso que ofrece cloud computing puede proporcionar el enfoque más rentable.

Fuente: (Microsoft Azure, 2015)

En general, la funcionalidad fundamental de SQL Azure y de SQL tradicional es intercambiable, con la excepción de algunas limitaciones de tamaño de la base de datos en la plataforma Windows Azure. Cada enfoque ofrece diferentes beneficios y costos en términos de infraestructura, de manera que es sólo cuestión de seleccionar la mejor opción según el caso de negocios individual. También puede extender las cargas de trabajo entre ambas plataformas de manera muy integrada.

2.7.3. Azure Storage Service

Es un servicio de almacenamiento no relacional provisto por Microsoft Azure Storage Service cuenta con tres formas básicas de almacenar datos:

- Tables,
- Queues,
- Blobs.

Dado que solo se utilizaran los Azure Blobs, no se entrara en detalle sobre el funcionamiento de los Tables y Queues.

El almacenamiento de blobs de Azure es un servicio para almacenar grandes cantidades de datos no estructurados, como texto o datos binarios, a los que se puede obtener acceso desde cualquier lugar del mundo a través de HTTP o HTTPS. El almacenamiento de blobs se puede usar para exponer datos públicamente al mundo o para almacenar los datos de la aplicación en privado.

Los Azure Blobs funcionan como contenedores que almacenan archivos, para su acceso distribuido, este servicio de almacenamiento cuenta con los siguientes componentes:

Tabla 11. Componentes de Azure BLOB

Cuenta de almacenamiento	Se realiza todo el acceso al almacenamiento de Azure a través de una cuenta de almacenamiento.
Contenedor	Un contenedor proporciona una agrupación de un conjunto de blobs. Todos los blobs deben residir en un contenedor. Una cuenta puede contener una cantidad ilimitada de contenedores.
Blob	Un archivo de cualquier tipo y tamaño. Hay dos tipos de blobs que pueden almacenarse en el almacenamiento de Azure: blobs en páginas y en bloques.
Formato de dirección URL	Los blobs son direccionables mediante la siguiente dirección URL: “http://<storage account>.blob.core.windows.net/<container>/<blob>”

Fuente: (Microsoft y la computación en nube, 2015)

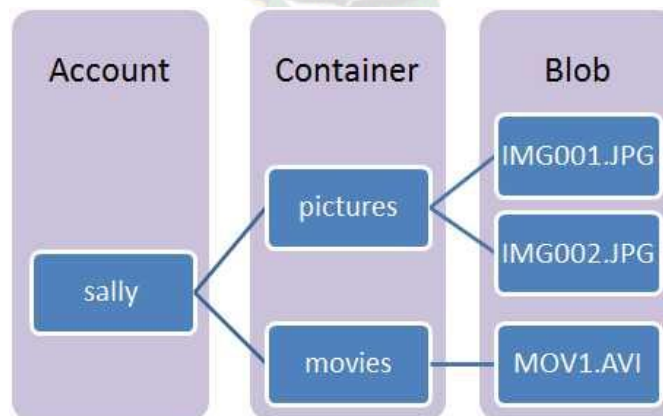


Figura 3.8. Estructura de un Blob. (Microsoft Azure, 2015)

2.8.Desarrollo móvil Ágil (MAD)

2.8.1. Introducción a MAD

Desarrollo ágil y móvil, o MAD por sus siglas en inglés (Mobile Agile Development), es una metodología de desarrollo ágil para aplicaciones móviles, propuesta en la Tesis de grado “Georreferenciación y su uso en Smartphones para la localización de personas” del Licenciado Jorge Cupi Veliz.

2.8.2. De donde viene MAD?

MAD está inspirada en la metodología ágil orientada a móviles MASAM y en la metodología ágil XP. Obtiene de estas dos metodologías la idea de un desarrollo rápido con frecuentes actualizaciones a una aplicación móvil ya publicada también de manera rápida en base a los requerimientos de un mercado específico dentro de las aplicaciones móviles.

2.8.3. Estaciones MAD

2.8.3.1.Estudio de Mercado

En esta sección el desarrollador se dedica a revisar aplicaciones iguales o al menos similares a las que tenemos en mente, esto debido a que no queremos invertir tiempo y dinero en una aplicación que creemos podría ser la idea del millón de dólares, y sin embargo ya existe y tiene usuarios que no piensan considerar alternativas, o el otro caso: Quizás la idea que tenemos existe en el mercado y no genera nada de ingresos debido a la pobre aceptación por parte de usuarios.

Por tales razones, debemos de revisar al menos tres o cuatro mercados actuales y analizar de esta manera si nuestra idea vale la pena, debemos analizar qué cosas le faltan a las soluciones actuales y cómo podemos mejorar la oferta existente. Cuando ya tengamos estos datos, es momento de definir las características con las que contara nuestra aplicación, necesitamos definir esto antes de poder escoger las tecnologías con las cuales trabajaremos.

El siguiente paso es definir el número de plataformas a las cuales queremos apuntar para el desarrollo de nuestra aplicación.

En resumen, los puntos más importantes de este segmento son:

- Análisis de requisitos por parte del mercado seleccionado.
- Búsqueda de soluciones existentes en al menos tres plataformas móviles diferentes.
- Establecimiento de características que tendrá la aplicación en base a los dos anteriores puntos.

2.8.3.2. Elección de una arquitectura, patrones y tecnologías para la aplicación

Se realiza la búsqueda de una arquitectura que se acomode a las necesidades de la aplicación, en términos de dimensión de la aplicación, alcances, y escalabilidad. Se tiene que tener mucha cautela al momento de elegir una arquitectura apta para una aplicación en específico, ya que no todas las arquitecturas son aptas para ser la base de una aplicación, y no todas las aplicaciones pueden adaptarse para trabajar sobre una misma arquitectura. (Gossman, 2006)

Al momento de realizar la elección de las tecnologías, herramientas, patrones de diseño y desarrollo con que se trabajara, se recomienda al menos tener dos opciones y establecer una tabla de pros y contras de cada alternativa. De esta manera se lograr dar con el mejor candidato que se acomode a las necesidades de la aplicación a desarrollarse en lugar de escoger una opción por prisa pura.

Se tiene que tener mucha cautela al momento de elegir una arquitectura apta para una aplicación en específico, ya que no todas las arquitecturas son aptas para ser la base de una aplicación, y no todas las aplicaciones pueden adaptarse para trabajar sobre una misma arquitectura. Lo mismo va para la elección de un lenguaje de programación, un framework de desarrollo, patrones y filosofías de diseño, etc.

2.8.3.3. Definición de módulos de la aplicación

En esta sección se procede a definir los segmentos que existirán dentro de la aplicación en base a las características que fueron identificadas en la estación de “Estudio de aplicaciones en el mercado”.

De forma casi natural, se identificó poco a poco a los segmentos relevantes e importantes para la aplicación, el paso siguiente fue ordenar a los segmentos por medio de las prioridades identificadas. Similarmente, se identificaron sub segmentos dentro de cada segmento, a los que también se ordenaron en base a su prioridad en el desarrollo para que los demás segmentos y sub segmentos respectivos puedan funcionar.

Simultáneamente se empezó a identificar los tipos de datos que serían necesarios y estos fueron agrupados en forma de clases, interfaces, etc. De esta manera se pudo ir pensando en el tipo de base de datos que se necesitaba para la aplicación, si es que acaso se iba a necesitar una.

Por cada segmento de la aplicación, se comenzó a tener esbozos de las interfaces de usuario, dicho de otra manera, se necesitó contar con bosquejos de cómo se debería de ver la aplicación una vez esté terminada. Esto ayudo a identificar nuevas características que no se habían identificado previamente, además de desechar ciertas funcionalidades que aparentemente no serán necesarias para el usuario.

Luego de una serie de iteraciones en los pasos a realizar por cada segmento, se llegó a concluir el desarrollo de la aplicación aún en fase de prueba. Llego el momento de ver si se está conforme o no con la interfaz que será presentada al usuario y si de ser necesario, se tendrán que realizar cambios.

En resumen, en esta sección el desarrollador se dedica a definir cada módulo que existirá dentro de nuestra aplicación, además de identificar dependencias entre módulos para encontrar prioridades al momento de comenzar a desarrollar cada uno de estos. Los puntos más importantes dentro de este módulo son los siguientes:

2.8.3.3.1. Diseño de la interfaz de usuario

- Diseño básico de la interfaz de usuario.
- Navegación de las páginas dentro de la aplicación.
- Diseño final de la interfaz de usuario, que puede ser realizado durante la implementación lógica de un módulo en específico.
- Iconografía de la aplicación.

2.8.3.3.2. Desarrollo y prueba de un Modulo

- Análisis de funcionalidades que presentara el modulo.
- Identificación de funcionalidades dependientes.
- Desarrollo de una funcionalidad.
- Prueba de funcionalidad sobre la interfaz apropiada en su estado actual.

2.8.3.3.3. Prueba completa de la Aplicación.

En teoría, en esta etapa del desarrollo, el desarrollador debería de encontrarse con fallas dentro de los segmentos independientes que generalmente vienen a ser aquellos módulos prioritarios con los que se comenzó el desarrollo. En cualquiera sea el caso, se deberían considerar las siguientes pruebas:

- A nivel de módulos independientes.
- A nivel de módulos dependientes.
- Prueba en distintos dispositivos, de preferencia físicos y no virtuales.
- Prueba exhaustiva de manipulación de la interfaz de usuario (cambio de fuente de letras, colores, resolución de pantalla, etc.) si es posible dependiendo al dispositivo y plataforma móvil sobre la cual se está trabajando.
- Prueba exhaustiva de la aplicación frente a agentes externos (reproducción de multimedia en el dispositivo, llamadas entrantes, sistema de geolocalización desactivado, plan de datos desactivado, etc.)

2.8.3.3.4. Publicación de la Aplicación.

La última pero no menos importante estación en la metodología MAD es la de la publicación de la aplicación en el mercado de aplicaciones de la plataforma a la que se está apuntando.

Pero antes de pensar en publicar la aplicación, es hora de definir el modelo de negocios que será utilizado para monetizar la idea.

Una vez que ya se está totalmente convencido de cuál es el mercado al que se está apuntando, es hora de publicar la aplicación, o comenzar a hacerla conocer. Para esto se sugiere que se tenga una etapa Beta, e incluso una etapa Alpha para comenzar a recibir retroalimentación de los que serán los futuros usuarios de la aplicación.

Y es que en base a esta retroalimentación se puede planear las siguientes actualizaciones que incluirán nuevas funcionalidades o tendrán corregidos ciertos errores encontrados. Lo que menos quiere uno es lanzar al mercado una aplicación defectuosa o inconclusa. La retroalimentación juega un gran papel en esta metodología, se recomienda realizar pequeñas pero continuas actualizaciones en base a la información que vayan brindando los usuarios.

2.8.4. Porque escoger una metodología ágil para el desarrollo móvil?

“Los métodos del desarrollo ágil favorecen el planeamiento ligero y continuo en lugar del planeamiento detallado a largo plazo por el simple hecho de que, especialmente en el mundo del desarrollo móvil, los cambios ocurren demasiado rápido. El compromiso del desarrollo ágil de la entrega de software funcional y testeado en frecuentes intervalos (generalmente de 1 a 4 semanas) de tiempo asegura mucha más confiabilidad y oportunidad de incorporar retroalimentación obtenida del usuario, así como de la tecnología.” (Robert Holler, CEO de VersionOne, 2006)

“Uno de los retos que afrontan los desarrolladores de aplicaciones móviles es el hecho de que el hardware y la infraestructura para los móviles está evolucionando constantemente, lo que resulta en un promedio de vida de 12 meses aproximados para una aplicación móvil. Para

que un equipo de desarrolladores móviles pueda dar aplicaciones relevantes y funcionales al mercado, necesitan trabajar rápidamente para desarrollar una solución de software. Los principios del desarrollo ágil de software establecen un marco de trabajo para un equipo de desarrollo pueda desarrollar una aplicación móvil con mucho más tiempo de vida en el mercado de aplicaciones.” (Nenu Tech, compañía de desarrollo de aplicaciones móviles y web, 2012).



3.1. Estaciones de MAD

3.1.1. Estudio de aplicaciones disponibles en el mercado

Como se puede observar en el Capítulo 1, en la sección de antecedentes se han observado y analizado a detalle a cuatro soluciones y/o productos existentes, elegidos porque son las más relevantes del mercado:

- Baby Signing Lite.
- Baby Sign And Learn
- Baby Sign Language

3.1.2. Análisis de aplicaciones similares existentes

En base a lo observado en las soluciones y/o productos existentes, los usuarios buscan mayor incorporación del producto en sus Smartphones, les gusta el producto, pero no ven un esfuerzo de los proveedores por ofrecerles la versión móvil, o al menos mejorar las versiones existentes y hacerlas más completas.

También se identifica que tanto Baby signs Lite, ASL Dictionary y las demás soluciones analizadas, carecen de propiedades y características que las hagan atractivas para el usuario, y que los productos físicos como “Baby Signs and Learn” y “Baby Signs Language” no se encuentran disponibles a primera mano en varios países y que en muchos casos habrá que realizar un pedido internacional para obtenerlos, sin mencionar el coste extra que presentaría en adición al alto precio del producto. Todo esto deja un mercado mundial sin soluciones similares, por tanto la idea de tener una aplicación que funcione tanto en Bolivia como en Inglaterra o Australia, por dar algunos ejemplos, es algo más que interesante para cualquier usuario.

Finalmente, los precios de ambos servicios nos dan una clara posibilidad de sacar ventaja al momento de ofrecer una aplicación similar por casi un tercio del precio original. Después de todo, la idea de esta tesis es demostrar que se puede ofrecer un producto similar, con un precio mucho más bajo para que sea accesible a más personas alrededor del mundo, sin tratar de cerrarse a un solo país, región o continente.

3.1.3. Análisis de requerimientos por usuarios

Las características que tiene la aplicación están definidas en la sección de Alcances y Límites en el Capítulo 1: Marco Referencial. Estas características fueron definidas posteriormente al análisis de aplicaciones disponibles en el mercado y al análisis de requerimientos de usuarios.

3.1.4. Conclusión del estudio de aplicaciones

Existe al menos cuatro soluciones presentes en el mercado actual para las tecnologías móviles de Android y iOS. Eso sí, los usuarios actuales no están nada conformes con la aplicación móvil ya que quieren una mejor integración su dispositivo móvil, además de contar con más y mejores características que las que ofrecen las soluciones actuales.

En tal efecto, se desarrolló la aplicación para las plataformas Windows 8.1 y Windows Phone 8.1, cubriendo también de esta manera la falta de esta aplicación para plataformas del ecosistema de Microsoft.

3.2. Elección de tecnologías y patrones de desarrollo

3.2.1. Lenguaje de Programación

Ahora que ya se tiene definida la necesidad de desarrollar una aplicación multiplataforma, es momento de ver que lenguajes de programación pueden ayudar a lograr este cometido:

Tabla 3.12. Ventajas y desventajas de Lenguajes de programación.

	Ventajas	Desventajas
C#	<ul style="list-style-type: none"> • Los sistemas operativos Windows Phone 8 y Windows 8.1 no tienen como lenguaje nativo. • La programación asíncrona se encuentra en un mayor nivel de madurez en este lenguaje, • Basta documentación en la Nube. • Librerías de clases portables son bastante útiles para el desarrollo multiplataforma. 	<ul style="list-style-type: none"> • Baja aceptación por parte de la comunidad de desarrollo multiplataforma. • Encontrar ayuda ante la presencia de bugs y errores es más complicado. • Encontrar recursos en términos de librerías y material de terceros es más difícil debido a que la comunidad no es tan grande como la de desarrolladores que usan JavaScript.
JavaScript+ HTML5	<ul style="list-style-type: none"> • Cuenta con una amplia gama de frameworks y herramientas que impulsan el desarrollo multiplataforma, • Las plataformas a las que apunta el tesista aceptan a JavaScript como lenguaje de programación para sus aplicaciones. • Es seguro y fiable. Por lo que puede ser filtrado para el mismo Javascript. 	<ul style="list-style-type: none"> • La plataforma a la que se apunta no tiene a JavaScript como lenguaje de programación nativo, • Está demostrado que JavaScript impacta en el rendimiento de una aplicación cuando es utilizada desde un dispositivo móvil (Crawford, 2013), • JavaScript no estaba planeado en sus principios para el desarrollo de aplicaciones móviles, • Si bien hay una comunidad extensa de desarrolladores que utilizan JavaScript, el número de personas dentro de la comunidad que puede ayudar al momento de surgir un problema disminuye dependiendo al framework que se esté utilizando.

Fuente: (Cupi Veliz, 2014) (Centro de Desarrollo de Windows, 2015)

Se eligió C# como lenguaje de programación. Es claro que la comunidad de desarrolladores multiplataforma no es tan grande como la de JavaScript, sin embargo C# en un lenguaje mucho más maduro para el desarrollo de aplicaciones móviles y el más utilizado para el desarrollo de aplicaciones universales de Windows, presenta características interesantes y sobresalientes como la programación asíncrona y la creación de librerías de clases reutilizables que llega a ser el factor importante y decisivo al momento de desarrollar un núcleo reutilizable de código fuente, que es uno de los fines al cual apunta el tesista.

3.2.2. Entorno de desarrollo

Como entorno de desarrollo, se escogió Visual Studio 2013 versión Professional debido a las bondades y características con las que cuenta al momento de escribir, depurar, re factorizar, probar, analizar y mantener código fuente. Además de contar con la integración que tiene con distintos servicios, como por ejemplo GitHub para el versionamiento de código, Nuget para la descarga de componentes de terceros o Add-ins como CodeMaid para la depuración de código.

Además de contar con Team Foundation Server que ofrece funciones de control de código fuente, seguimiento de elementos de trabajo; y sobre todo, para desarrollar una aplicación universal de Windows (Windows Phone 8.1 y Windows 8.1), se necesita contar con este entorno de desarrollo.

3.2.3. Elección de un patrón de diseño o arquitectura

Llegada la búsqueda de un patrón de diseño idóneo para desarrollar la aplicación universal de Windows y que además su futura portación a otras plataformas sea lo más sencilla posible. Estaba claro que hasta el momento se habían identificado dos fases grandes en el desarrollo:

- La elaboración de un núcleo de código fuente reutilizable.
- El desarrollo de la interfaz gráfica de las distintas versiones de nuestra aplicación para cada plataforma, consumiendo el núcleo de código.

Se escogió el patrón de diseño MVVM. La marcada tendencia entre los desarrolladores de aplicaciones para las plataformas de Windows es la de utilizar MVVM como patrón de diseño de facto debido a su naturaleza multiplataforma que se adapta adecuadamente a la construcción de aplicaciones universales de Windows. Será necesario ver si tal naturaleza es expandible a plataformas cuyo lenguaje nativo no es C# y no utilizan XAML como lenguaje de diseño.

3.2.4. Base de Datos

Es el momento de tocar el tema de una elección de base de datos que es usada no solo para administrar en cierta manera a los usuarios de la aplicación, si no también que se encargue de la enorme tarea que conlleva el administrar los progresos que tienen los usuarios en sus clases, los signos utilizados en estas, las agrupaciones creadas por el usuario, además de archivos de imágenes y de reconocimiento.

La base de datos elegida: SQL Azure y Azure Storage Service.

SQL Azure

Este modelo se adapta adecuadamente a la aplicación propuesta, ya que:

- Reduce el nivel complejidad de las consultas que las que se hubiesen realizado de usar una base de datos relacional de Azure.
- Va de la mano con Azure Mobile Service, que es el medio por el cual la aplicación de la presente tesis accede a los datos.
- Se adapta al modelo Entity Framework, permitiendo al desarrollador trabajar con datos relacionales usando objetos específicos del dominio.

Azure Storage Service

Si bien la aplicación propuesta en la presente tesis, no tiene como objetivo la administración y manejo de archivos, esta característica está presente en uno de sus módulos, específicamente, en el que verificara la existencia de nuevas clases disponibles para su

descarga, en el cual podrán visualizarse la disponibilidad de nuevos módulos de enseñanza de signos de bebé, los cuales, a disposición del usuario podrán descargarse y añadirse a las clases incluidas en la aplicación, sin la necesidad de que el tesista como desarrollador de la aplicación, deba generar una nueva versión de la aplicación que contenga las nuevas clases, ni que el usuario tenga que actualizar a una nueva versión de la aplicación.

Una base de datos no relacional está preparada para almacenar cantidades masivas de datos sin sufrir estos problemas a cambio de ciertas pérdidas, por ejemplo queda desechada la idea de una posible normalización de datos, o el no poder realizar consultas de un nivel de complejidad considerable. Además, si bien la aceptación de la industria ha ido creciendo en torno a la adaptación de bases de datos no relacionales, está aún no llega a una etapa madura como para comenzar a considerar a dichas bases como una opción totalmente viable para cualquier escenario.

3.2.4.1. Gestión de usuarios

Hay un par de frases que escuchamos muy frecuentemente en nuestro campo como: “Ya no es necesario redescubrir la rueda” y “Es mejor descansar sobre los hombros de gigantes”. Apoyándonos en ambas frases, muy sabias por cierto, resulta mucho más práctico utilizar una base de datos ya existente que se ocupe del manejo y gestión de usuarios que vamos a tener en nuestra aplicación. ¿Y quién tiene actualmente la base de datos de personas más grande en el mundo? La respuesta es Facebook.

Hay muchas cosas respecto a Facebook SDK que pueden ser muy engorrosas, sin embargo no debemos preocuparnos demasiado por ello, hoy en día existe una amplia gama de componentes para Windows / Windows Phone que se han encargado de resolver varios temas, especialmente haré referencia a Facebook SDK for Windows un SDK desarrollado y mantenido por Microsoft. Para autenticarse en Facebook es necesario hacer uso de su API REST, y desde luego como siempre el primer paso es autenticarnos en Facebook para lograr obtener un token, de allí en más es solo utilizar los endpoint de Facebook para cada operación

que se vaya a realizar como acceder a información del usuario que va desde el nombre y foto hasta los amigos de los que vayan a ser usuarios de la aplicación.

Por las razones mencionadas previamente, se ha utilizado Facebook para la gestión de usuarios, lo cual da dos ventajas que se admiran a simple vista:

- Los usuarios no tienen que tomarse el tiempo y la molestia de crear una nueva cuenta solo para usar esta aplicación, por tanto es una contraseña y un nombre de cuenta menos que tienen que recordar.
- Una tarea menos al momento de desarrollar la aplicación, ya que la autenticación de los usuarios, así como el almacenamiento de sus datos personales y los de sus contactos son manejados por Facebook.
- Una tarea menos al momento de desarrollar la aplicación, ya que la autenticación de los usuarios, así como el almacenamiento de sus datos personales y los de sus contactos son manejados por Facebook.

3.2.5. Presupuesto.

Una vez elegido el entorno de desarrollo y tecnologías a utilizarse, se procederá a detallar los gastos relacionados a la realización del proyecto, en ese sentido solo se incluirán costes directos.

Puesto que el tiempo invertido en el desarrollo del proyecto está comprendido entre principios de Septiembre de 2014 y finales de Mayo del presente año, teniendo así 9 meses de duración, es a partir de este dato que se obtuvieron los diferentes costes asociados al proyecto, a continuación se procede con el detalle.

Costes directos.

Equipos:

Fue necesario adquirir un ordenador portátil así como un sensor Kinect V2 para desarrollador tal y como se detalla en la siguiente tabla.

Tabla 3.13. Costes de equipos para el desarrollo del proyecto

Descripción	Coste (Bolivianos)	% de uso dedicado	Dedicación (Meses)
Ordenador	7000 Bs	100 %	9 Meses
Sensor Kinect V2	2100 Bs	88 %	8 Meses
Smartphone	1200 BS	55 %	5 Meses
Coste Total	10300 Bs		

Software

A continuación se detalla el coste del software que fue necesario utilizar para el desarrollo del presente proyecto.

Tabla 3.14. Coste de Licencia para el desarrollo del proyecto

Descripción	Licencias	Coste / Licencia (Bolivianos)	Coste
Visual Studio 2013 Professional	1	0 Bs	0 Bs
SDK Kinect V2 for Windows	1	0 Bs	0 Bs
Microsoft Office 2013	1	0 Bs	0 Bs
Coste Total	10300 Bs		

Cabe aclarar que el proyecto no tuvo coste de licencias, puesto que el tesista al miembro activo de la comunidad de Microsoft Student Partner, tiene acceso a licencias de los productos utilizados en el desarrollo del presente proyecto de manera gratuita.

Coste Total.

A continuación se resumen los diferentes costes mencionados y se calculara el coste total del proyecto.

Tabla 3.15. Coste total del proyecto

Concepto	Coste Total (Bolivianos)
Equipos	10300 Bs
Software	0 Bs
Total	10300 Bs

De tal manera, el coste total del proyecto asciende a 10300 Bs.

3.2.6. Conclusión sobre las tecnologías utilizadas

Se eligió a Microsoft y su solución en la nube denominada Microsoft Azure, de la cual se hará uso del servicio de esta plataforma llamado Azure Mobile Service, el cual accede a datos almacenados en una base de datos relación en la nube denominada SQL Azure la cual está alojada en la misma plataforma, además, de esta plataforma, también se hará uso de la solución Azure Storage Blob, la cual permitirá almacenar archivos de datos en la nube. De esta manera se garantiza que la aplicación tenga un funcionamiento asegurado de 24 horas por día, los 365 días del año.

Es importante mencionar que por efectos prácticos para la implementación de la base de datos SQL Azure, esta se implementó a través Entity Framework, el cual permite crear un modelo a partir de código C#, este enfoque es llamado también Code First, además cabe aclarar que este enfoque es utilizado en Azure Mobile Service, que como se detalló anteriormente, es el que fue utilizado por el tesista.

En adición, se hará uso de la red social Facebook, siendo más específico, se utilizara su base de datos de usuarios, lo cual directamente permite descartar el desarrollo y creación de una base de datos de usuarios propia, dejando de lado la creación y gestión de cuentas a un tercero. Además los usuarios podrán utilizar la aplicación con la única necesidad de tener una cuenta en dicha red social.

Toda esta data almacenada en la nube podrá ser accedida y consumida por dispositivos del ecosistema Microsoft, más propiamente, dispositivos con sistema operativo Windows 8.1 y Windows Phone 8.1.

Como patrón de diseño de la aplicación se optó por MVVM, por ser este el más apropiado y el que mejor se ajusta en la implementación de una Aplicación Universal de Windows, más cuando este tipo de implementación comparte gran parte de código por no decir la mayoría, entre las plataformas que integra, además que este patrón permite una escalabilidad, fácil expansión y mantenimiento del proyecto.



Figura 3.9. Flujo completo y offline de la Aplicación.

3.3. Definición y desarrollo de segmentos

Ya habiendo pasado el proceso de estudio, análisis y proceso de selección basado en pros versus contras, se eligieron a las tecnologías que se utilizaran, además de que ya se tienen definidas las características de la aplicación, es momento de agrupar estas últimas en segmentos y ordenar dichos segmentos por prioridad de desarrollo. Esto es, identificar a aquellos segmentos que son necesarios para el correcto funcionamiento de los segmentos que les sigan.

3.3.1. Manejo de sesión en Facebook

Si bien en se proveerán sesiones offline a futuro, la versión actual de la aplicación en su integridad requiere que el usuario inicie sesión en la aplicación vía cuenta de Facebook. Se necesita entonces que los usuarios tengan una manera fácil y rápida de acceder a su cuenta en dicha red social, otorgar una serie de permisos a la aplicación, y proceder luego a

almacenar los datos de su sesión para no tener que obligarlos a realizar los mismos pasos cada vez quieran usar la aplicación.

Las tareas que se pueden identificar para este segmento son tres:

- Permitir que un usuario pueda iniciar una sesión en Facebook desde la aplicación.
- Almacenar la información de la sesión actual.
- Permitir que un usuario pueda cerrar la sesión en Facebook desde la aplicación.

3.3.2. Diccionario de signos disponibles en la aplicación.

Una vez el usuario haya iniciado sesión en la aplicación, debería poder realizar las siguientes operaciones:

- Buscar y/o escoger de entre todos los signos que se encuentra en la aplicación el de su preferencia.
- Una vez elegido el signo el usuario podrá visualizar una imagen en la que se indica la manera en la que debe realizar el signo.
- Todos los signos deben estar disponibles de manera offline.

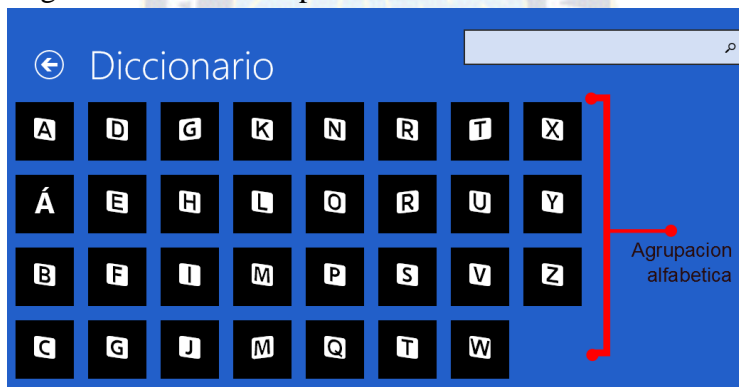


Figura 3.10. Áreas del Diccionario/Agrupación Alfabética

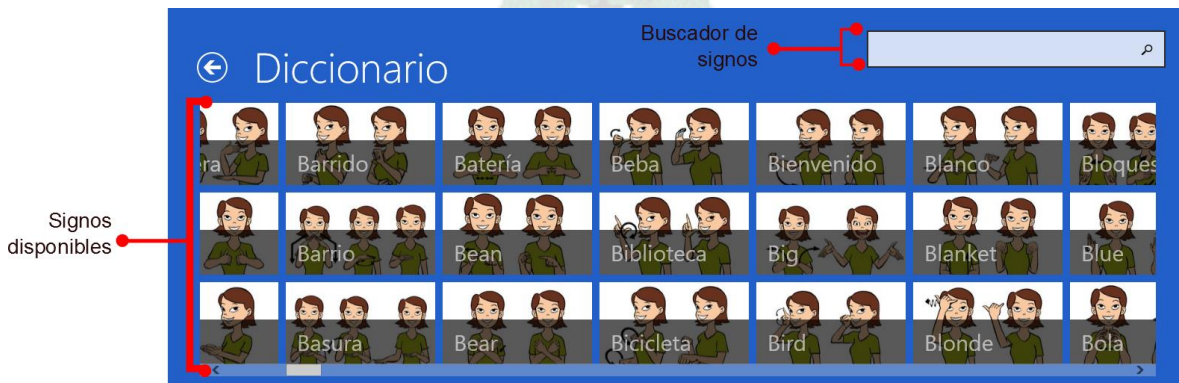


Figura 3.11. Áreas del Diccionario

3.3.3. Preferencia de signos

En este segmento, el usuario debe poder realizar las siguientes tareas:

- Permitir al usuario agrupar los signos de su preferencia y/o conveniencia incluidos en la aplicación en secciones a la que él pueda acceder de manera inmediata.
- Sugerir al desarrollador los signos agrupados, para que el desarrollador incluya estas en las clases de la aplicación.



Figura 3.12. Áreas del módulo de preferencias de signos

3.3.4. Practica de signos

Los signos utilizados en las clases presentes en la aplicación, son para que el usuario pueda empezar con signos que se pueden utilizar con frecuencia para su uso cotidiano, como ser los signos que se refieren familiares y/o a personas importantes, hasta alimentación y animales entre otros.

En este segmento, las tareas deben ser discriminadas respecto al tipo de dispositivo en el que el usuario este haciendo uso de la aplicación.

- PC y computador portátil.
 - Practicar los signos de manera didáctica haciendo uso del Kinect V2 para el reconocimiento de signos.
 - Verificar que esta signando apropiadamente.



Figura 3.13. Áreas del módulo de práctica de signos

- Smartphone.
 - Aprender a través de imágenes que indican al usuario como signar apropiadamente.

3.3.5. Registros de Progreso.

Si bien no cuenta con un alto nivel de complejidad en el desarrollo, este segmento es importante, ya que indica al usuario el avance que ha obtenido en el desarrollo que ha tenido en el uso de la aplicación y de los cursos que incluye.

Las tareas de este segmento son:

- Permitir al usuario visualizar las clases de la aplicación que ha tomado.
- Visualizar la los signos que ha aprendido además de los intentos realizados en el proceso de aprendizaje.

LECCION	SIGNO	CANTIDAD
Animales	Cat	5
Comida	Leche	2

Figura 3.14. Áreas del Historial de aprendizaje

3.3.6. Personalización de la aplicación de acuerdo a las necesidades del usuario

Este segmento, si bien no es totalmente necesario, es importante brindar opciones de personalización al usuario, por ejemplo, las características que contiene la aplicación:

- Actualización periódica de Tiles.
- Enviar información de retroalimentación al desarrollador.
- Terminar su sesión (logoff) en la aplicación.

3.3.7. Creación de formularios para obtener retroalimentación

Es una parte totalmente independiente a casi todos los demás segmentos, para funcionar solo se necesita que el usuario tenga una sesión iniciada en Facebook. En este segmento, le pedimos al usuario que ayude a mejorar la aplicación brindando retroalimentación en forma de cuestionarios llenados o calificaciones en la tienda de aplicaciones de la plataforma desde donde este. Para que ambas tareas puedan ser realizadas, también se tiene que verificar que el usuario tenga una conexión activa a Internet en ese instante.

The image shows a feedback form titled "FeedBack" with a back arrow icon. It is divided into two main sections:

- Facilidad de uso de la Aplicacion (Area de Facilidad del Formulario):**
 - La aplicacion es confusa?
 - Si
 - No
 - Tiende a desplegar muchos mensajes de error?
 - Si
 - No
 - No tan frecuentemente
 - Es fluida la interaccion con el Kinect?
 - Si

- Utilidad de la Aplicacion (Area de Utilidad del Formulario):**
- Cumple con mis expectativas?
 - Para nada
 - En cierta manera
 - Si cumple
 - La excede
- Estoy aprendiendo lengua de Signos de bebé?
 - Para nada
 - Posiblemente
 - Efectivamente

At the bottom right, there is a button labeled "Envía los resultados al desarrollador" with an "Enviar" icon.

Figura 3.15. Áreas del formulario de retroalimentación.

3.3.8. Diagrama navegación de la aplicación.

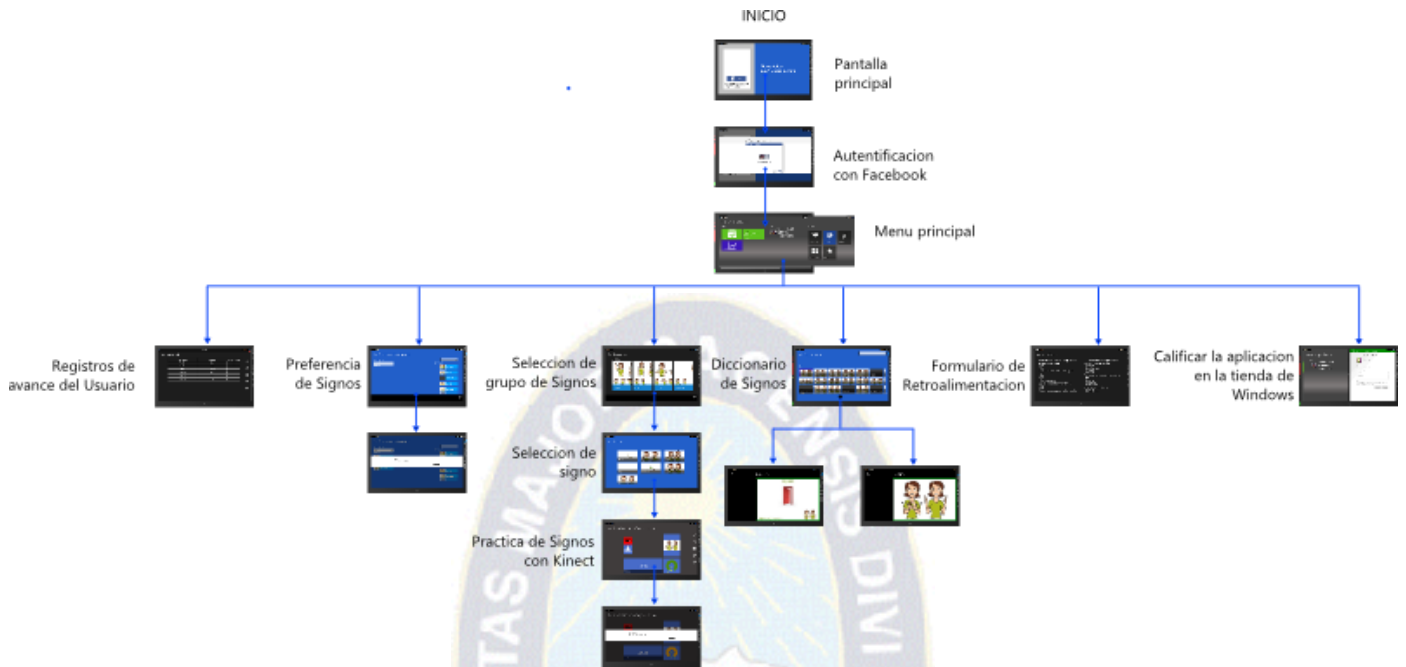


Figura 3.16. Diagrama de navegación de la Aplicación.

3.4. Modelado de datos

Tal y como se indicó en anteriormente en la estación “Elección de tecnologías y patrones de desarrollo” de la presente tesis, se optó por tener una base de datos relación y una no relacional, ambas alojadas en la plataforma Azure de Microsoft, además se está utilizando los servicios de Facebook, para que a través de esta red social, se pueda obtener la información necesaria de los usuarios.

Es importante mencionar que para la implementación de la base de datos SQL Azure, se utilizó Entity Framework, el cual permite crear un modelo a partir de código C#, este enfoque es llamado también Code First, además cabe aclarar que este enfoque es utilizado en Azure Mobile Service, que como se detalló anteriormente, es el que está utilizando el tesista.

Dado que es un tanto complicado el graficar un diagrama con las clases y sus atributos correspondientes, lo mejor es definir primero que clases son necesarias para que la aplicación funcione correctamente:

- Una clase que represente a los usuarios y su información proveniente de Facebook, la clase es llamada APPUser y entre sus atributos contiene por ejemplo cadenas que representan el nombre, ID y la url de la foto del usuario.
- Una clase que represente a las agrupaciones de signos creados por el usuario a la cual se denominó CustomLesson.
- Una clase a la que se denominó Sign, la cual representa a los signos en la cual se aprecian atributos como ser descripción, el path de la imagen que representa el signo y también el de la imagen que indica como signar.
- Una clase que contenga información de los signos que contienen las agrupaciones de signos creados por el usuario.
- Finalmente pero no menos importante, una clase denominada Feedback, la cual contendrá la información de retroalimentación que enviaran los usuario al desarrollador.

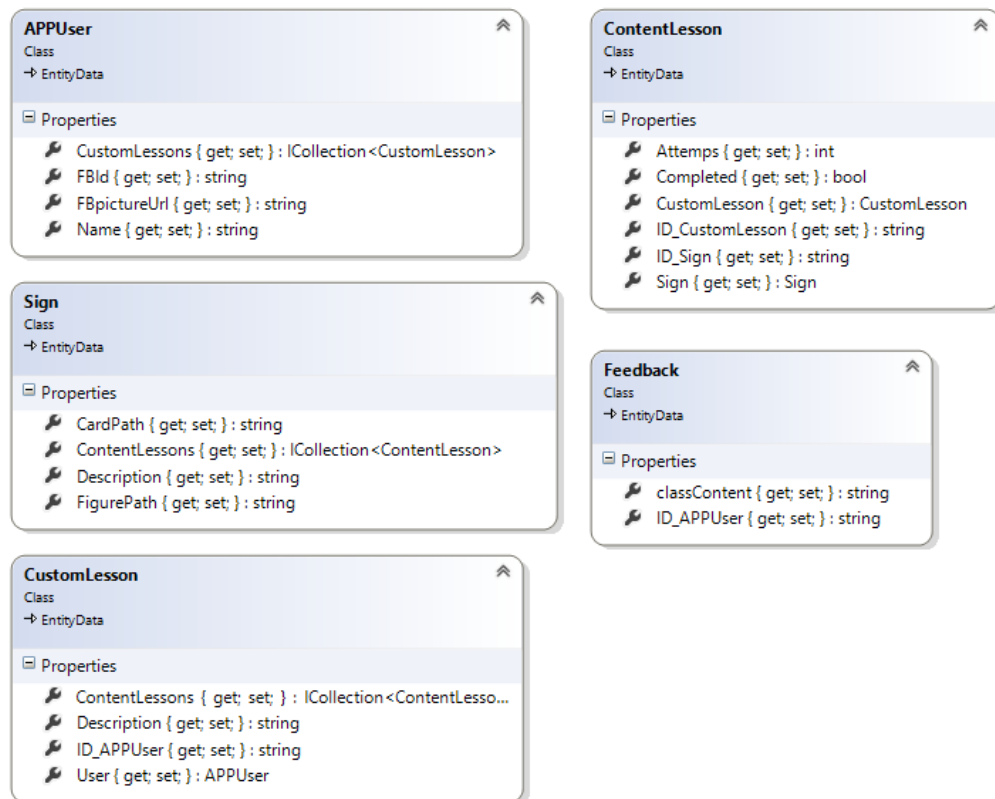


Figura 3.17. Clases principales para el funcionamiento de la Aplicación

En la sub estación Desarrollo de módulos se entrara en mayor detalle en más clases que se irán utilizando dentro de cada segmento en específico, las cinco clases ilustradas en la Figura 3 son las importantes pero no las únicas.

3.5.Desarrollo de la interfaz

Las interfaces que se proponen en esta sección no tienen necesariamente el objetivo de revelar la apariencia de lo que será el producto final, si no sirven más como una base para comenzar a trabajar y empezar a dar énfasis a las prioridades de desarrollo que fueron identificadas en la anterior sección. Entonces, más que haber comenzado a desarrollar una interfaz, se tiene en realidad una colección de bosquejos que pretenden dar el paso inicial del desarrollo de la aplicación.

Para este efecto, se tratara de ilustrar como lucirá la aplicación tanto en las dos plataformas iniciales: Windows Phone 8.1 y Windows 8.1.

3.5.1. Pantalla principal

3.5.1.1.Pre Inicio de sesión en Facebook

Dado que es necesario que haya un usuario con sesión iniciada en Facebook, la primera vez que el usuario abra la aplicación, no podrá hacer otra cosa que no sea iniciar sesión. (Figura 3.10).



*Figura 3.18. Pantalla principal de la aplicación.
a) Aplicación de Windows 8.1, b) Aplicación de Windows Phone 8.1*

3.5.1.2. Post inicio de sesión en Facebook.

Ya iniciada la sesión a través de su cuenta de Facebook, el usuario puede acceder a las características y diferentes opciones que disponibles en aplicación.

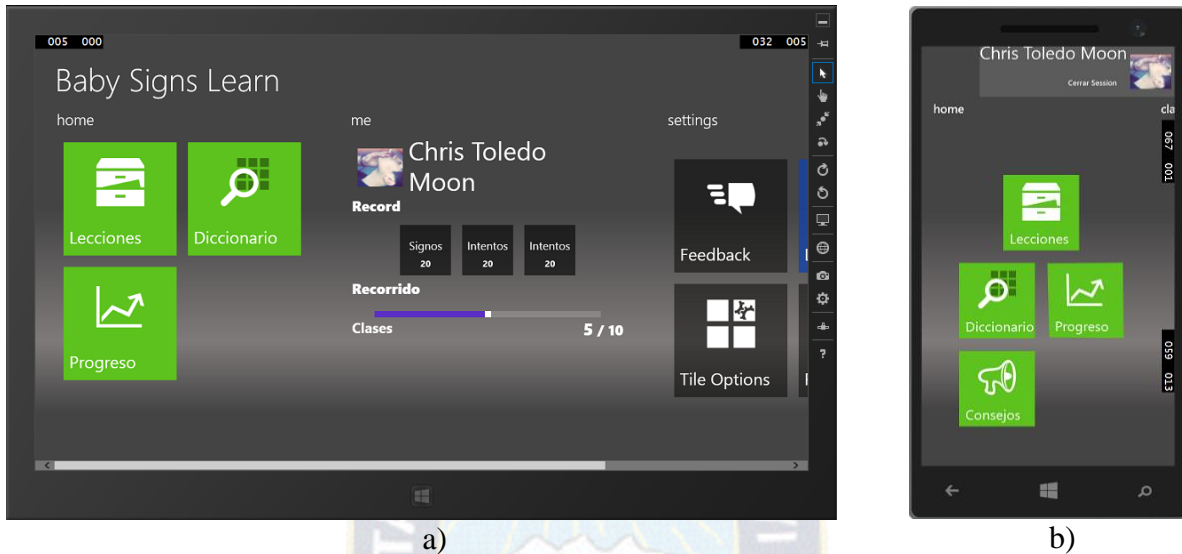


Figura 3.19. Menú principal de la aplicación.
a) Aplicación de Windows 8.1, b) Aplicación de Windows Phone 8.1.

3.5.1.3. Inicio de sesión en Facebook

Es necesario facilitar el acceso a Facebook para el usuario, para esta tarea se hace uso de la API y SDK de Facebook. (Figura 3.12).

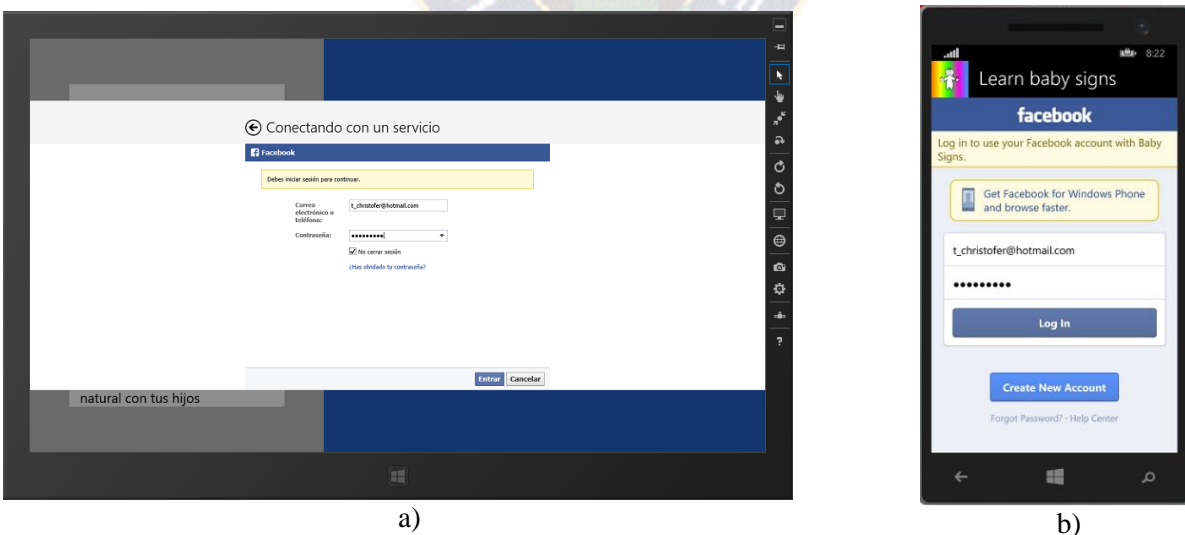


Figura 3.20. Inicio de sesión en la red social Facebook.
a) Aplicación de Windows 8.1, b) Aplicación de Windows Phone 8.1

3.5.1.4. Pantalla de Diccionario.

Puesto que las lecciones de la aplicación en primera instancia solo cuentan con los signos principales y/o esenciales para el aprendizaje del usuario, es necesario también tener a su disposición un diccionario que incluya gran cantidad de los signos existentes actualmente, de esta manera el usuario pueda acceder a estos cuando así lo desee. (Figura 3.13)

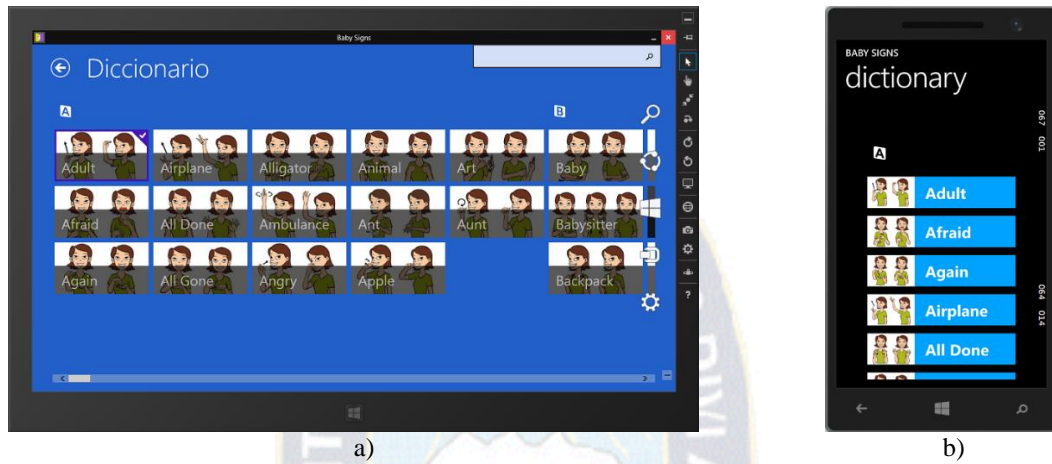


Figura 3.21. Diccionario de signos.

a) Aplicación de Windows 8.1, b) Aplicación de Windows Phone 8.1

3.5.1.5. Pantalla de preferencia de signos

En esta sección el usuario tendrá la posibilidad de agrupar varios signos de su preferencia y acceder rápidamente a estos. (Figura 3.12).

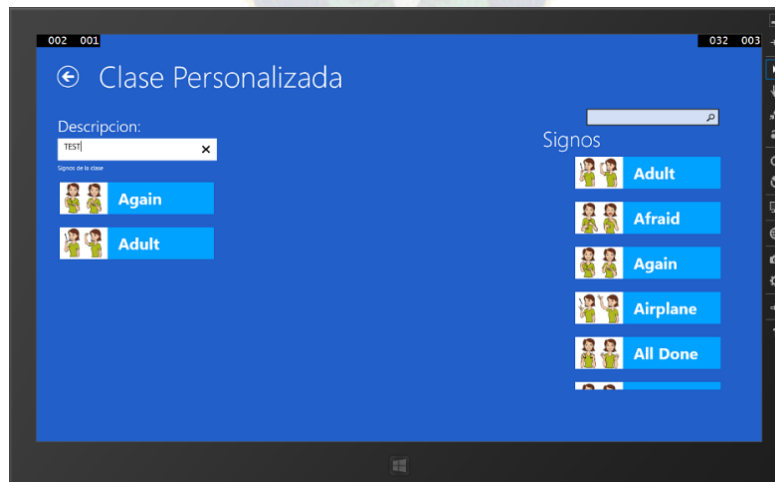
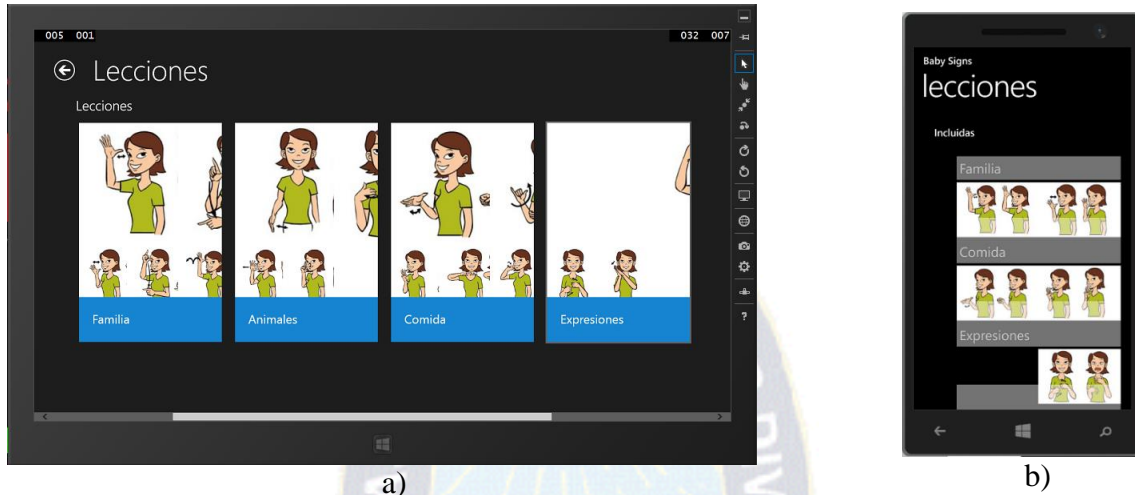


Figura 3.22. Agrupación de los signos de preferencia del usuario

3.5.1.6. Pantalla de Lecciones

En esta pantalla el usuario puede ver las clases disponibles en la aplicación y también los grupos de preferencia de signos que creó. (Figura 3.15).



a) *Figura 3.23. Lecciones disponibles en la aplicación*

a) *Aplicación de Windows 8.1, b) Aplicación de Windows Phone 8.1*

3.5.1.7. Módulo de práctica y reconocimiento

Este módulo es el más complejo de todos y el que presento mayor tiempo de desarrollo, además que es el más importante de la aplicación, pues es en este en que el usuario practica los signos con ayuda del Kinect el cual solo registra el avance después de que el usuario haya realizado correctamente el signo.



Figura 3.24. Módulo de reconocimiento y práctica de Signos

3.5.1.8. Historial de aprendizaje.

Este módulo despliega el avance y/o practica que tuvo el usuario en relación a los signos presentes en las clases con Kinect incluidas en la aplicación.

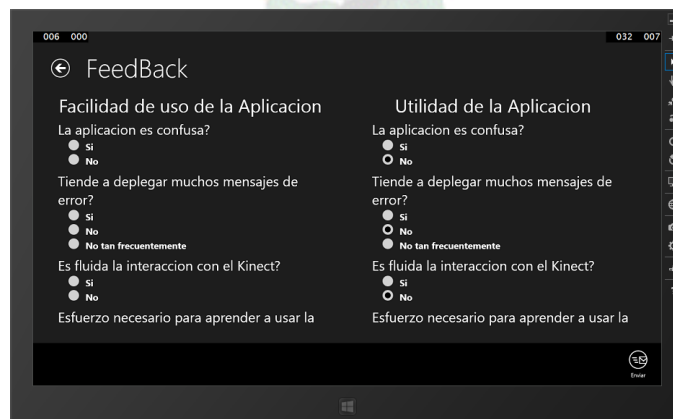


LECCION	SIGNO	CANTIDAD
Familia	Papá	4
Familia	Mamá	4
Familia	Hermano	2
Familia	Sister	2
Familia	Abuelo	5
Animales	Perro	2
Animales	Gato	3
Animales	Ratón	6
Comida	Leche	5
Comida	Agua	2
Comida	Cereal	3
Comida	Yogur	1

Figura 3.25. Historial de Practica de signos.

3.5.1.9. Formularios de Retroalimentación

Es una parte totalmente independiente a casi todos los demás segmentos, para funcionar solo se necesita que el usuario tenga una sesión iniciada en Facebook. En este segmento, le pedimos al usuario que ayude a mejorar la aplicación brindando retroalimentación en forma de cuestionarios llenados o calificaciones en la tienda de aplicaciones de la plataforma desde donde este. Para que ambas tareas puedan ser realizadas, también se tiene que verificar que el usuario tenga una conexión activa a Internet en ese instante.



006 000 032 007

← FeedBack

Facilidad de uso de la Aplicacion

La aplicacion es confusa?

Si
 No

Tiende a desplegar muchos mensajes de error?

Si
 No
 No tan frecuentemente

Es fluida la interaccion con el Kinect?

Si
 No

Esfuerzo necesario para aprender a usar la

Utilidad de la Aplicacion

La aplicacion es confusa?

Si
 No

Tiende a desplegar muchos mensajes de error?

Si
 No
 No tan frecuentemente

Es fluida la interaccion con el Kinect?

Si
 No

Esfuerzo necesario para aprender a usar la

Figura 3.26. Formulario de retroalimentación.

3.6.Desarrollo de módulos

Ya diseñados los primeros esbozos de cómo debería quedar la aplicación se procedió a comenzar con el desarrollo de la aplicación. A modo de recordar, se utilizó el entorno de desarrollo Visual Studio 2013 con Update 3. Al ser una aplicación universal de Windows, la plantilla de este tipo de proyecto, separa el código en tres proyectos distintos, adicionalmente se incluye el proyecto del Mobile Service:

- Baby Sign Language.Shared, representa el núcleo de código reutilizable para nuestras plataformas de Windows.
- Baby Sign Language.Windows, representa el código de la versión de la aplicación para Windows 8.1.
- Baby Sign Language.WindowsPhone, representa el código de la versión de la aplicación para Windows Phone 8.1.
- MobileServiceBabySigns, representa el código del Mobile Service que estará alojado en la Nube de Microsoft Azure.

Previamente al detalle del desarrollo de los módulos de la aplicación, cabe recordar que de acuerdo al patrón de diseño MVVM, tenemos tres segmentos importantes:

- Modelado de datos (Models).
- Manipulación de datos (ViewModels).
- Presentación de datos al Usuario (Views).

La mayor parte del desarrollo de los módulos están enfocados en los dos primeros incisos mencionados previamente, tanto los Models como los ViewModels se encuentran en el proyecto Baby Sign Language.Shared, las Vistas o Views son las que cambian de un sistema operativo a otro. Se hizo todo lo posible por una parametrización de código, sin embargo hay ciertas partes del código fuente que no pueden ser escritas en el núcleo de código, estas partes por tanto estarán en los diferentes proyectos de cada plataforma.

3.6.1. Desarrollo de interfaces de cada módulo

Para la implementación de la parte grafica de cada plataforma, es decir las vistas, se utilizó el lenguaje de diseño XAML, nativo de las plataformas de Windows 8.1 y Windows Phone 8.1. Además se implementaron UserControls propios del tesista que sigan el paradigma del diseño de interfaces del estilo Metro de Windows. La idea de implementar UserControls propios, se debe en gran medida debido a que los existentes, tanto de manera nativa como desarrollados por terceros no eran del agrado del tesista.

3.6.1.1. Estructura de un Use Control

Puesto que solo se desarrolló un Live Tile Personalizable usado tanto en la plataforma de Windows 8.1 como la de Windows Phone 8.1, se describirá y analizara de manera completa la construcción de este.

3.6.1.1. Parte Grafica

Como se observa en el segmento de código 3.1, se tienen dos partes al momento el UserControl: una sección encargada de la animación que representa las transiciones del movimiento del icono y una sección en la que definimos los controles que compondrán al UserControl.

```
<UserControl
  x:Class="Baby_Sign_Language.Views.UserControls.MainMenuOption"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  xmlns:interactivity="using:Microsoft.Xaml.Interactivity"
  xmlns:arcanaStudio="using:ArcanaStudio" mc:Ignorable="d" Name="userControl">
  <Grid Name="myGrid" Background="{Binding ElementName=userControl, Path=Background}">
    <interactivity:Interaction.Behaviors>
      <arcanaStudio:UniversalTilt />
    </interactivity:Interaction.Behaviors>
    <Grid.RowDefinitions>
      <RowDefinition Height="7*" />          <RowDefinition Height="3*" />
    </Grid.RowDefinitions>
    <Image Name="ImageOption" Grid.Row="0" Source="ms-appx:///Assets/Icons/icon-
cabinet.png" Width="85" VerticalAlignment="Center" HorizontalAlignment="Center" />
    <TextBlock Name="Title" Grid.Row="1" Text="LECCIONES" FontSize="30"
      VerticalAlignment="Center" HorizontalAlignment="Left" />
  </Grid>
</UserControl>
```

Segmento de Código 3.1. Creación de un LiveTile

3.6.1.2. Parte Lógica

Al ser este un UserControl personalizable, necesita recibir parámetros desde donde sea instanciado y también para modificar sus propiedades, para esto, necesita que sus atributos tengan la forma de propiedades.

```
public string TextTitle
{
    get { return (string)this.GetValue(TextTitleProperty); }
    set
    {
        this.Title.Text = value;
        this.SetValue(TextTitleProperty,value);
    }
}
public ImageSource ImageOp
{
    get { return (ImageSource)this.GetValue(ImageOpProperty); }
    set
    {
        this.ImageOption.Source = value;
        this.SetValue(ImageOpProperty,value);
    }
}
public double ImageOptionSize
{
    get { return (double)this.GetValue(ImageOptionSizeProperty); }
    set
    {
        this.ImageOption.Width = value;
        this.SetValue(ImageOptionSizeProperty, value);
    }
}
public double gridWith
{
    get { return (double)this.GetValue(gridWithProperty); }
    set
    {
        this.myGrid.Width = value;
        this.SetValue(gridWithProperty, value);
    }
}
public double gridHeight
{
    get { return (double)this.GetValue(gridHeightProperty); }
    set
    {
        this.myGrid.Height = value;
        this.SetValue(gridHeightProperty, value);
    }
}
```

Segmento de Código 3.2. Atributos de un UserControl

3.6.1.2. Almacenamiento Aislado

Una de las características importantes, de la cual fue utilizada frecuentemente es la del almacenamiento aislado, se almacenaron principalmente dos tipos de valores en la aplicación:

- Variables que indican opciones del usuario.
- Variables que almacenan llaves y códigos de acceso para utilizar la API de Facebook y la de los Azure Tables.

3.6.1.2.1. Estructura de una variable en el Almacenamiento Aislado.

El almacenamiento aislado funciona de la misma manera para ambos sistemas operativos, este representa un contenedor de valores de configuración de la aplicación, así que para efectos prácticos, se ilustra a continuación la manera de construir una variable que aloja sus datos en el almacenamiento aislado de Windows Phone 8. (Segmento de código 3.4).

```
public static string APPUserID
{
    get
    {
        if
        (ApplicationData.Current.LocalSettings.Containers.ContainsKey("APPUserID"))
        {
            return
            ApplicationData.Current.LocalSettings.Values["APPUserID"].ToString();
        }
        else return String.Empty;
    }
    set
    {
        ApplicationData.Current.LocalSettings.Values["APPUserID"] = value
    }
}
```

Segmento de Código 3.3. Almacenamiento aislado de una variable

Se puede observar que son variable de tipo estáticas, que al ser invocadas buscan el valor y/o contenedor asociado a una llave única (key), estas opciones son parte del segmento llamado AppicationData de Windows Phone y Windows, la cual proporciona acceso a los valores almacenados en el contenedor.

En el segmento de código presentado se puede observar que la propiedad APPUserID, al ser invocado busca en el contenedor si existe un valor asignado a esa llave, de no encontrarlo, devolverá una cadena vacía. Sin embargo cuando se asigne un valor a esta propiedad, en lugar de determinar si la llave existe, automáticamente asignara una casilla del diccionario con la nueva llave y su valor correspondiente. Todas las variables que tengan sus valores en el almacenamiento aislado se encuentran declaradas en el archivo AppSettings.cs que contiene los métodos y atributos principales para el inicio de la aplicación.

Se puede comparar al almacenamiento aislado como una mini base de datos bastante útil para almacenar valores simples como cadenas, números y booleanos, sin la necesidad de crear una base de datos relacional y/o implementar más lógica para almacenar datos básicos.

En los distintos segmentos de código que se presentan en esta sección de la tesis, cada vez que se contemple al objeto AppSettings, se está haciendo referencia precisamente al objeto declarado en el archivo AppSettings.cs que contiene a las variables alojadas en el almacenamiento aislado. Por ejemplo: Se observara frecuentemente en el código, variables similares a “AppSettings.APPUserID” o “AppSettings.IsAuthenticated”.

3.6.1.3. Manejo de Sesión en Facebook

El primer paso que se tuvo que dar para que la aplicación comience a tomar forma fue el de lograr que un usuario inicie sesión en Facebook. Para lograr este objetivo se tenían cinco pasos por completar:

- Tener un botón que se encargue de iniciar sesión en Facebook en la pantalla principal.
- Iniciar una sesión en Facebook para obtener las credenciales que permitan acceder a la información del usuario.

- Almacenar las credenciales del inicio de sesión en la aplicación de tal manera que el usuario no tenga que iniciar sesión cada vez que desee hacer uso de la aplicación.
- Tener un botón que se encargue de cerrar la sesión de Facebook en el Menú principal de la aplicación.

Para todos los pasos, se hizo uso del mismo comando asociado al evento de un botón, así, cada vez que el botón fuese presionado inicia una sesión en Facebook o pregunta al usuario si desea cerrar la sesión existente, sin embargo cabe especificar que tiene un comportamiento casi distinto respecto a la plataforma en la que se esté ejecutando, es por tanto que se puede observar que se hace una validación de plataforma.

```
public ICommand LoginCommand
{
    get{
        #if WINDOWS_PHONE_APP
        return new DelegateCommand(() =>
        #else
        return new DelegateCommand(async () =>
        #endif{
            if (AppSettings.isAuthenticated){
                AppSettings.fbToken = null;
                AppSettings.fbProfileName = null;
                AppSettings.fbProfileID = null;
                AppSettings.fbProfilePicture = null;
                AppSettings.isAuthenticated = false;
                this.SessionState = AppSettings.isAuthenticated;
            }else{
                #if WINDOWS_PHONE_APP
                Authenticate.LoginAndContinue();
                #else
                var response = await Authenticate.LoginAsync();
                var Token = Authenticate.ValidateFacebookResponse(response);
                if (!string.IsNullOrEmpty(Token)){
                    dynamic user = await Authenticate.UserInfo(Token);
                    AppSettings.fbProfileName = user.name;
                    AppSettings.fbProfileID = user.id;
                    AppSettings.fbToken = Token;
                    this.ServicesAuthentication();
                    AppSettings.APPUserID = await CRUD.insertUser(new APPUser { FBID = user.id, Name =
                    user.name, FBpictureUrl = user.photoUrl });
                    AppSettings.isAuthenticated = true;
                    this.SessionState = AppSettings.isAuthenticated;
                    this.LoadUserProfile();}
                #endif}});}}}
```

Segmento de Código 3.4. Métodos de autenticación para una sesión en Facebook

En este segmento de código se puede apreciar que primero se verifica en que plataforma se está ejecutando la aplicación, ya que de acuerdo a esto es que varía el comportamiento del método, mas no el fin, luego se verifica si hay una sesión iniciada en la aplicación, esto mediante una revisión del estado de variables que obtienen su información del almacenamiento aislado del sistema operativo. De encontrarse un valor positivo, esta cierra la sesión del usuario y regresa a la pantalla inicial, caso contrario se llama a un método de inicio de sesión para desplegar al usuario una ventana en donde pueda ingresar los datos de su cuenta de Facebook. Finalmente, tras validar si el inicio de sesión fue exitoso, se almacenan los valores obtenidos en el almacenamiento aislado y se procede a dar visibilidad a las características de la aplicación.

Es necesario que este método u operación se encuentra en el núcleo de nuestro código, aunque este varíe ligeramente respecto de la plataforma, es por esto que se realiza la validación, de esta manera implementarlas en un mismo método, para que sea reutilización en ambas plataformas.

A grandes rasgos, se puede observar que el método descrito realiza una petición HTTP de tipo GET a la API de Facebook con una cadena que contiene la consulta que desea ser realizada y en respuesta contienen un texto que contiene un objeto JSON. Este texto es propiamente trasladado utilizando las librerías correspondientes para los métodos devuelvan un objeto común y corriente: Un objeto “APPUser” que representa al usuario como tal, una vez obtenido este objeto, además del cambio de estado de los valores en los contenedores, se procede almacenar y/o registrar este objeto que representa al usuario, en la base de datos alojada en Azure haciendo uso de nuestro Mobile Service que actúa como el BackEnd de la aplicación. Por su naturaleza trivial, el cierre de sesión de Facebook no será detallado en esta sección.

```

public static async Task<string> insertUser(models.APPUser user)
{
    var UserResult = await App.MobileService.GetTable<APPUser>().Where(w =>
w.FBId == user.FBId).ToListAsync();
    if (UserResult.FirstOrDefault() == null)
    {
        await
App.MobileService.GetTable<models.APPUser>().InsertAsync(user);
    }
    var registers = await App.MobileService.GetTable<APPUser>().Where(w =>
w.FBId == user.FBId).ToListAsync();
    return registers.FirstOrDefault().Id;
}

```

Segmento de Código 3.5. Inserción de Usuario mediante el Mobile Service

El segmento de código 3.6, se puede apreciar la inserción de un nuevo usuario a la base de datos alojada en la plataforma de Azure a través de nuestro Mobile Service, previamente se valida que el usuario no exista en la base de datos, de no existir, se inserta en la base de datos, finalmente se obtenemos el ID único del usuario para retornar este valor, para finalmente como se detalló en el segmento de código 3.4., este se guarda en el almacenamiento asilado.

Si bien este segmento es breve tomando en cuenta varias consideraciones, es uno de los más importantes ya que sin el manejo apropiado del inicio de sesión de Facebook, la aplicación no puede funcionar.

3.6.1.4.Despliegue de Signos para el acceso del usuario.

Puesto que es necesario que el usuario tenga acceso inmediato a los signos y para evitar un tráfico trivial de datos, es que se optó por tener tanto la lista de signos serializada en un archivo JSON como los recursos gráficos que esta requiere, dentro de la aplicación, es por tanto que el segmento de código a continuación describe la manera de deserializar esta lista y asignarla a una propiedad que está asociada a los elementos gráficos en la vista, de tal manera que esta sea visualizada por el usuario. Este método se ejecuta cada vez que se carga la vista del módulo de diccionario.

```

async void getSigns()
{
var uri = new System.Uri("ms-appx:///Assets/Data/signs.json");
var SignJson = await Util.getJsonContentThisJsonFileUri(uri);
Signs = Util.Transform.toObject<ObservableCollection<Sign>>(SignJson);
var SignsGrouped = Signs.GroupBy(f => f.Description[0]).Select(f => new
SignsbyLetter { Letter = f.Key, Signs = f.ToList() });
SignsByGroup = SignsGrouped.ToList();
}

```

Segmento de Código 3.6. Método de cargado de signos en la vista

A grandes rasgos y sin ser demasiado redundante, el segmento de código 3.7, lee el contenido del archivo de formato JSON almacenado en los recursos de la aplicación, a continuación se procede a parsear este contenido para devolver un objeto de tipo List<Sign>, paso siguiente se clasifica esta lista por grupos tomados respecto a la inicial, para luego pasar esta agrupación a la propiedad SignsByGroup el cual está asociado a un componente de la vista que se encarga de presentar esta data de manera gráfica.

```

public RelayCommand<Models.Sign> GridView_ItemClick{
get{
if (gridView_ItemClick == null){
gridView_ItemClick = new RelayCommand<Models.Sign>(
(item) =>{
(Window.Current.Content as
Windows.UI.Xaml.Controls.Frame).Navigate(typeof(Views.SignPage), item);
});}
return gridView_ItemClick;}}

```

Segmento de Código 3.7. Selección de una seña para su despliegue visual

Cabe recalcar que esta lista también se encuentra en la base de datos colgada en la nube de Azure y que su despliegue se reduce a una simple petición al Mobile Service, sin embargo previamente se aclaró el motivo por el esta se encuentra en la también aplicación.

3.6.1.5. Almacenamiento de preferencia de signos.

El comando descrito a continuación se encarga de almacenar en la base de datos la selección de signos de preferencia del usuario, como en procesos de inserción anteriores, este proceso se lo ejecuta haciendo uso de nuestro Mobile Service que actúa como el BackEnd de la aplicación. No se describe el proceso de selección de signos, ya que este representa otro tipo de complejidad en el manejo de las propiedades de los componentes de la vista.

```

public ICommand SaveClassCommand{
    Get{
        return new DelegateCommand(async() =>{
            if (!string.IsNullOrEmpty(this.Description) &&
                !string.IsNullOrWhiteSpace(this.Description) && this.CustomSigns.Count > 0){
                this.LoadingClass_status = Visibility.Visible;
                CustomLesson customLesson = new CustomLesson {
                    Description=this.Description,
                    ID_APPUser=AppSettings.APPUserID};
                customLesson.Id = await
                    Services.AzureMobileServices.CRUD.InsertCustomClass(customLesson);
                var LContentLesson = this.CustomSigns.Select(s => new ContentLesson {
                    ID_CustomLesson = customLesson.Id, ID_Sign = s.Id });
                await Services.AzureMobileServices.CRUD.InsertContentLesson(LContentLesson);
                this.LoadingClass_status = Visibility.Collapsed;
                var dialog = new Windows.UI.Popups.MessageDialog("Se ha guardado su Clase
                    Personal", this.Description);
                await dialog.ShowAsync();
                (Window.Current.Content as Windows.UI.Xaml.Controls.Frame).GoBack();}
            Else{
                var dialog = new Windows.UI.Popups.MessageDialog("Verifique si los campos son
                    correctos", "ERROR");
                await dialog.ShowAsync();}
        });}}

```

Segmento de Código 3.8. Inserción de grupo de signos a través del Mobile Service

El segmento de código 3.9., en primera instancia verifica la validez del título que se asignara a la agrupación de signos y que existan dichos signos, pasada la validación, crea un objeto de tipo CustomLesson el cual tiene como identificador el id del usuario además de un id autogenerado de tal manera que puedan identificarse las agrupaciones del usuario y este se inserta en la base de datos a través de nuestro Mobile Service, paso seguido a partir del grupo de signos crea objetos de tipo ContentLesson que se almacenaran en una tabla intermedia la cual contiene como llaves el id del signo y el id del objeto CustomLesson, estos también se insertaran en la base de datos.

3.6.1.6.Reconocimiento de Signos con Kinect V2

Es importante aclarar que el código de este segmento en específico no se encuentra en el proyecto núcleo de código compartido de la aplicación, si no en el proyecto de Windows 8.1 puesto que esta característica de reconocimiento con el sensor Kinect solo puede ser usada en computadores y laptops que cuenten con el Sistema Operativo mencionado.

El desarrollo de este segmento fue realmente el más que complejo de ser elaborado debido al complejo uso que representa el uso del dispositivo Kinect en una aplicación, además del proceso de aprendizaje que este conlleva.

En primera instancia se desarrolló un modelo al que se denominó KinectCore el cual abstraerá la complejidad que conlleva el uso de los recursos que podemos extraer del Kinect, para así reutilizar este core en diferentes ViewModels, de tal manera que futuras implementaciones sean menos complejas. No se entrará en la descripción de KinectCore debido a su complejidad y sobre todo a que no es objetivo del segmento describir el manejo del SDK del Kinect, además que el código fuente está disponible para su descarga.

```
public void SetSetting(){
this.kinectSensor = KinectSensor.Default();
this.Kinect_Color_Loaded();
this.Kinect_Body_Loaded();
this.kinectSensor.Open();
this.kinectSensor.IsAvailableChanged += kinectSensor_IsAvailableChanged;
this.multiSourceFrameReader =
this.kinectSensor.OpenMultiSourceFrameReader(FrameSourceTypes.Body |
FrameSourceTypes.Color);
this.multiSourceFrameReader.MultiSourceFrameArrived +=
MultiSourceFrameReader_MultiSourceFrameArrived;
this.visualGestureBuilderDatabase = new
VisualGestureBuilderDatabase(this.DataBasePath);
_gestureFrameSource = new VisualGestureBuilderFrameSource(this.kinectSensor, 0);
_gestureFrameSource.AddGestures(this.visualGestureBuilderDatabase.AvailableGestures);
this.ListGesture = visualGestureBuilderDatabase.AvailableGestures.ToList();
_gestureFrameReader = _gestureFrameSource.OpenReader();
_gestureFrameReader.IsPaused = true;
_gestureFrameReader.FrameArrived += reader_FrameArrived;}
```

Segmento de Código 3.9. Carga de recursos del Kinect y de archivo de reconocimiento

El segmento de código 3.10., describe un método que inicializa métodos de KinectCore heredados en el ViewModel, primero se captura el sensor Kinect conectado al equipo, después se cargan los frames ColorFrame y BodyFrame, a continuación se empiezan a capturar los metadatos de los frames mencionados, se asigna el método kinectSensor_IsAvailableChanged que nos informara el estado del Kinect y también se asigna el método MultiSourceFrameReader_MultiSourceFrameArrived el cual que describió anteriormente, a continuación se carga el archivo de signos a través del método

VisualGestureBuilderFrameSource y se asigna el método reader_FrameArrived en el cual se realizan las operaciones de reconocimiento.

```
void MultiSourceFrameReader_MultiSourceFrameArrived(MultiSourceFrameReader sender,
MultiSourceFrameArrivedEventArgs args)
{
var reference = args.FrameReference.AcquireFrame();
#region ColorFrame
bool colorFrameProcessed = false;

using (ColorFrame colorFrame = reference.ColorFrameReference.AcquireFrame()){
if (colorFrame != null){
FrameDescription colorFrameDescription = colorFrame.FrameDescription;
if ((colorFrameDescription.Width == this.bitmap.PixelWidth) && (colorFrameDescription.Height ==
this.bitmap.PixelHeight)){
if (colorFrame.RawColorImageFormat == ColorImageFormat.Bgra){
colorFrame.CopyRawFrameDataToBuffer(this.bitmap.PixelBuffer);
}
}
Else
{
colorFrame.CopyConvertedFrameDataToBuffer(this.bitmap.PixelBuffer, ColorImageFormat.Bgra);
}
colorFrameProcessed = true;
}}
if (colorFrameProcessed)
{
this.bitmap.Invalidate();
}
#endregion

#region BodyFrame
bool dataReceived = false;

using (BodyFrame bodyFrame = reference.BodyFrameReference.AcquireFrame())
{
if (bodyFrame != null){
bodyFrame.GetAndRefreshBodyData(this.bodies);
dataReceived = true;
}
}
if (dataReceived)
{
this.BeginBodiesUpdate();
int count = 0;
for (int bodyIndex = 0; bodyIndex < this.bodies.Length; bodyIndex++)
{
Body body = this.bodies[bodyIndex];
if (body.IsTracked)
{
this.UpdateBody(body, bodyIndex);
_gestureFrameSource.TrackingId = body.TrackingId;
count++;}
else{
this.ClearBody(bodyIndex);
}
}
this.RecognitionState = count==1;
_gestureFrameReader.IsPaused = count==0?true:false;
}
}
```

Segmento de Código 3.10. Método de Captura de Frames

El segmento de código 3.10 describe es un método el cual recibe dos tipos de Frames, ColorFrame, el cual contiene metadatos sobre las imágenes que captura el Kinect, en este caso será usado como una cámara de video, y BodyFrame, este en particular contiene metadatos sobre los cuerpos detectados por el sensor, cantidad de cuerpos detectados y las coordenadas de cada articulación, este es el más importante pues es a través de él que se hará uso de estas coordenadas para dibujarlas en el ColorFrame, estos eventos se comunican con propiedades asociadas a elementos de la vista para su despliegue y/o visualización.

3.6.1.7. Azure Mobile Service

3.6.1.7.1. Estableciendo conexión con Azure Mobile Services

Se a declarado una instancia en el en el archivo App.xaml.cs, que es donde se encuentra la configuración base de la aplicación, a esta instancia se asigna por decirlo de alguna manera, la conexión hacia el Mobile Service, de esta manera esta instancia cliente ahora tendrá acceso al servicio móvil publicado en Azure.

```
public static MobileServiceClient MobileService { get; set; }
```

Segmento de Código 3.11. Declaración de la instancia cliente

```
public static void CheckAuthentication(){
    App.MobileService = new MobileServiceClient(
        App.Current.Resources["AzureMSApplicationURL"].ToString(),
        App.Current.Resources["AzureMSApplicationKEY"].ToString());
}
```

Segmento de Código 3.12. Asignación de llaves a la instancia Cliente

Como se puede apreciar, en segmento de código 3.12 asignamos a la instancia cliente la conexión, a través de la URL y KEY almacenados en la configuración de la aplicación estos parámetros son generados en la plataforma de Microsoft Azure.

Puesto que en segmentos anteriores se describieron métodos que realizaban inserciones a través del Mobile Service, ahora se hará descripción genérica.

```
await App.MobileService.GetTable<MODEL>().InsertAsync(REGISTO);
```

Segmento de Código 3.13. Inserción de un registro

```
await App.MobileService.GetTable<MODEL>().DeleteAsync(REGISTO);
```

Segmento de Código 3.14. Eliminación de un registro

```
await App.MobileService.GetTable<MODEL>().UpdateAsync(REGISTO);
```

Segmento de Código 3.15. Actualización de un registro

Como se observa en los segmentos de código 3.13, 3.14, 3.15, las operaciones son bastante triviales, basta con invocar a la instancia que actúa como cliente declarado anteriormente en la configuración de la aplicación, basta hacer uso de los métodos `InserteAsync`, `DeleteAsync` y `UpdateAsync` previa especificación del tipo de tabla a la cual se hará la inserción.

3.6.1.8. Creación de formulario para obtener retroalimentación.

Se creó un cuestionario que fue realizado en base al “modelo tecnológico de aceptación” o TAM por sus siglas en inglés (Abu-Dalbouh, 2013).

Dicho modelo intenta determinar cómo los usuarios están dispuestos a aceptar o rechazar una nueva tecnología, que cobra forma de aplicación, página web, sistema, u otros.

El cuestionario cuenta con dos segmentos: Percepción de usabilidad y percepción de facilidad de uso. Y de acuerdo a estudios y encuestas realizadas por el autor del modelo TAM, la usabilidad es más importante que la facilidad de uso en un factor de 1.5.

3.6.1.8.1. Cuestionario de Usabilidad

Utilidad percibida de la Aplicación

1. Cumple con mis expectativas?
a) Para nada. b) En cierta manera. c) Si cumple. d) La excede.
2. Estoy aprendiendo lengua de Signos de bebé?
a) Para nada. b) Posiblemente. b) Efectivamente
3. Es una aplicación efectiva?
a) Para nada b) En cierta manera c) Si cumple
4. Es una aplicación útil?
a) Si. b) No
5. Recomendaría la aplicación a otras personas?
a) Si. b) No

Facilidad de Uso de la Aplicación

1. La aplicación es confusa?
a) Si. b) No
2. Tiende a desplegar muchos mensajes de error?
a) Si. b) No. c) No tan frecuentemente
3. Es fluida la interacción con el Kinect?
a) Si. b) No
4. Esfuerzo necesario para aprender a usar la aplicación.
a) Mucho esfuerzo. b) Poco esfuerzo. c) Ningún esfuerzo
5. Tiende a realizar comportamientos inesperados?
a) Nunca. b) Casi nunca. c) De vez en cuando. d) Casi siempre. e) Siempre
6. Desempeño de la Aplicación
a) Muy malo. b) Malo c) Regular d) Bueno e) Muy Bueno

3.6.1.9. Creación del Mobile Service

El Mobile Service es el corazón en sí de la aplicación, esta provee una infraestructura cómoda, flexible y sencilla de implementar, pero de extrema complejidad si se desea, es a través de esta que realizamos todas las operaciones de inserción, actualización y eliminación de registros hacia nuestra base de datos.

A continuación se describe la creación del proyecto y a la implementación de uno de los modelos usados en la aplicación.

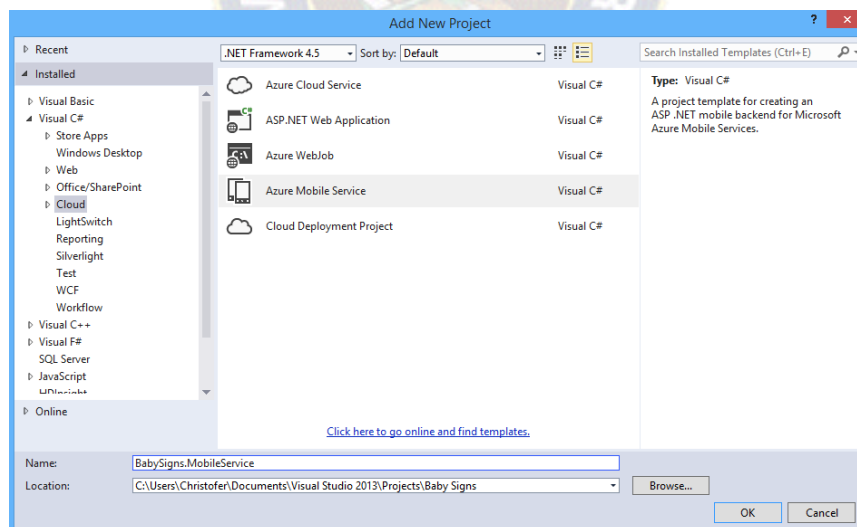


Figura 3.27. Creación de un Proyecto Azure Mobile Service

Es tan sencillo como crear un proyecto en Visual Studio 2013, previa instalación del SDK de AZURE y seleccionar el tipo de proyecto Azure Mobile Service.

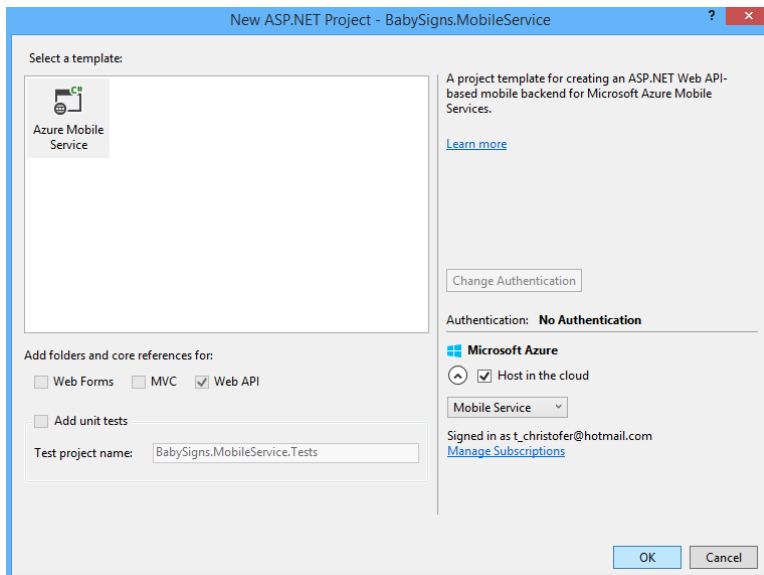


Figura 3.28. Configuración de Proyecto Mobile Service

Es preciso aclarar que no es necesario contar con una cuenta en Azure para realizar pruebas de manera local, sin embargo si es necesario si desean publicar su servicio en la Nube, sin embargo, puesto que la que el servicio ya está publicado en la nube, se detallara este proceso.

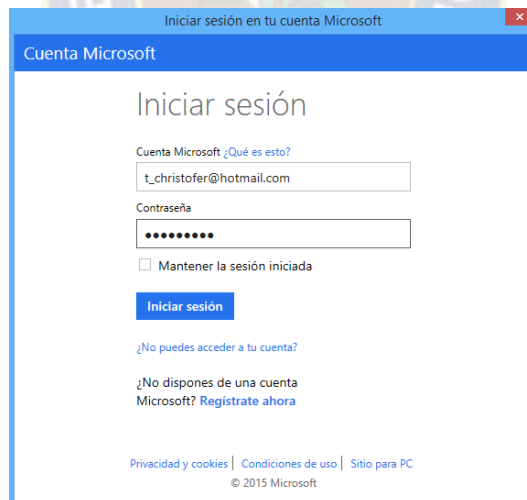


Figura 3.29. Ingreso de credenciales de la cuenta de Azure

El siguiente paso es asignarle un nombre al servicio, el de la aplicación es MobileServiceBabySigns y asignar un usuario y password a la base de datos que se creara.

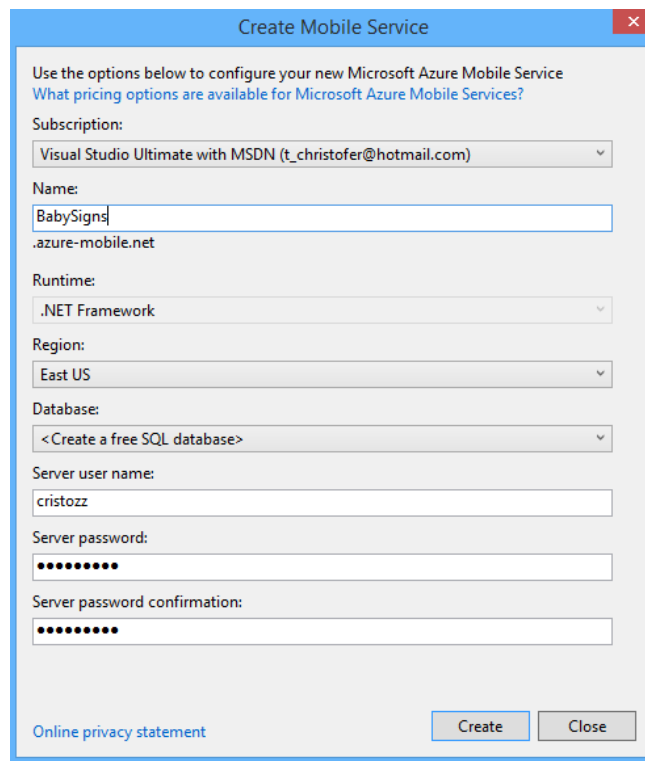


Figura 3.30. Configuración de propiedades del Mobile Service

Este proceso creará la base de datos en la plataforma de Azure además genera un proyecto con la siguiente estructura:

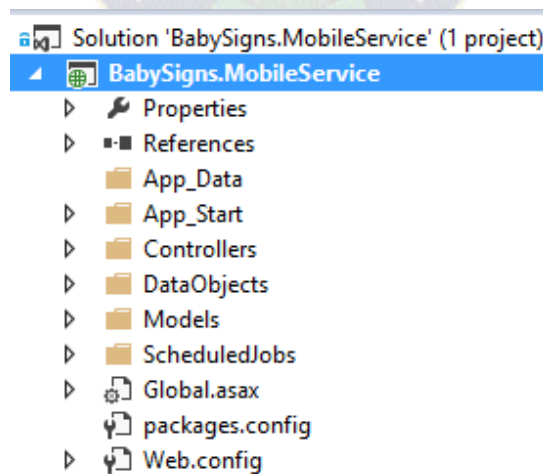


Figura 3.31. Estructura del proyecto BabySigns.MobileService

3.6.1.9.1. Creación de DataObjects

Se agrega una clase a este a este folder el proyecto a la cual denominaremos APPUser, esta clase deberá heredar de EntityData propia del SDK de Azure, la cual abstrae las propiedades necesarias para la creación del objeto, como ser el ID que actuara como Primary Key y su declaración no será necesaria.

```
public class APPUser : EntityData
{
    public string FBId { set; get; }
    public string FBpictureUrl { get; set; }
    public string Name { set; get; }
    public virtual ICollection<CustomLesson> CustomLessons { get; set; }
}
```

Segmento de Código 3.16. Entidad APPUser

Para explicar la relación entre entidades se describirá la clase denominada CustomLesson, la cual al ser una entidad, también hereda de EntityData, esta entidad también guarda relación con la entidad ContentLesson, sin embargo su explicación resultaría trivial y repetitiva.

```
public class CustomLesson : EntityData
{
    public string Description { get; set; }
    public string ID_APPUser { get; set; }
    [ForeignKey("ID_APPUser")]
    public virtual APPUser User { get; set; }
    public virtual ICollection<ContentLesson> ContentLessons { get; set; }
}
```

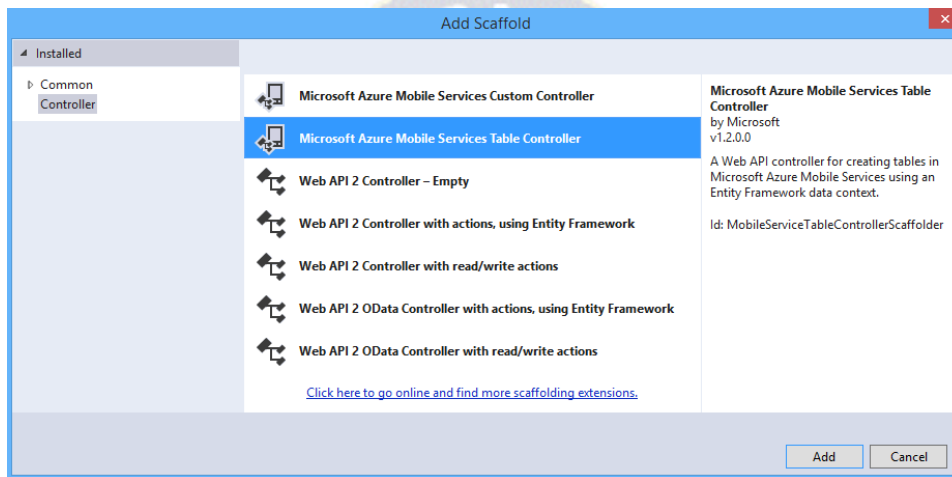
Segmento de Código 3.17. Entidad CustomLesson

Como se observa en el segmento de código 16.17, esta presenta el atributo de tipo APPUser denominado User, el cual está marcado mediante su cabecera como ForeignKey.

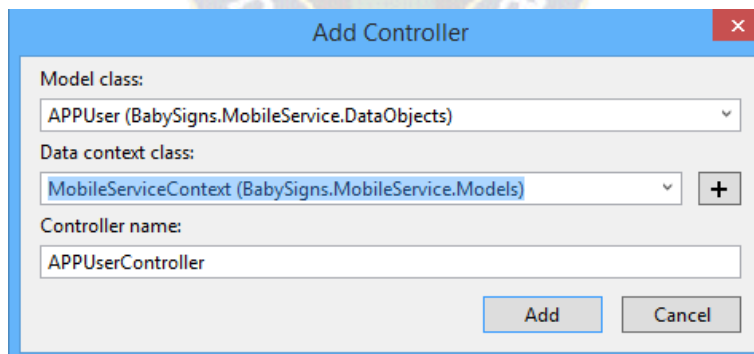
Este enfoque de creación de tablas a través de entidades declaradas a través de código se denomina CodeFirst, el cual es flexible y aliviana mucho el trabajo que tendría el tesista de tener que crear las tablas primero en SQL Server y publicarlas en Azure.

3.6.1.9.2. Creación de Controllers de las Entidades

Los Controller se encargan de servir a las peticiones HTTP. Este controlador se deriva de una clase `TableController<ENTIDAD>` y contiene los métodos de acción para cada acción HTTP, como GET, POST, este puede ser personalizado como el usuario vea conveniente, sin embargo el proyecto tiene la capacidad de generarnos una plantilla genérica con las operaciones básicas que necesitaremos para las entidades que necesitemos.



Segmento de Código 3.18. Selección del tipo de Controlador



Segmento de Código 3.19. Creación del Controller de la Entidad APPUser

Este proceso creara el siguiente segmento de código.

```
public class APPUserController : TableController<APPUser>
{
    protected override void Initialize(HomeControllerContext controllerContext){
        base.Initialize(controllerContext);
        MobileServiceContext context = new MobileServiceContext();
        DomainManager = new EntityDomainManager<APPUser>(context, Request,
Services);
    }
    public IQueryable<APPUser> GetAllAPPUser()
    {
        return Query();
    }
    public SingleResult<APPUser> GetAPPUser(string id)
    {
        return Lookup(id);
    }
    public Task<APPUser> PatchAPPUser(string id, Delta<APPUser> patch)    {
        return UpdateAsync(id, patch);
    }
    public async Task<IHttpActionResult> PostAPPUser(APPUser item) {
        APPUser current = await InsertAsync(item);
        return CreatedAtRoute("Tables", new { id = current.Id }, current);
    }
    public Task DeleteAPPUser(string id)    {
        return DeleteAsync(id);
    }
}
```

Segmento de Código 3.20. Controller de la Entidad APPUser

A su vez, este el Controller debería quedar instanciado en el Model Context del Servicio de la siguiente manera.

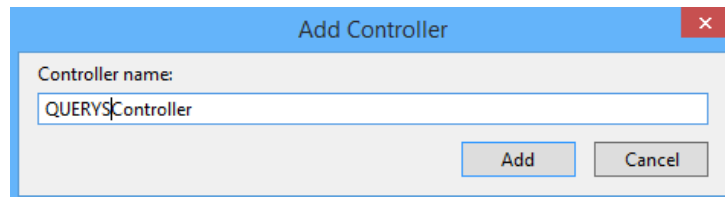
```
public DbSet<DataObjects.APPUser> APPUsers { get; set; }
```

Segmento de Código 3.21. Instancia del Controller de APPUser en el Context del Servicio

Por ser un proceso repetitivo, el tesista cree poco práctico detallar el desarrollo de la creación de las demás Entidades y sus respectivos Controllers.

3.6.1.9.3. Creación de Controllers Personalizados

Esta se lo realiza a través de una clase denominada ApiController, la cual es la base para los controladores que manejan las peticiones HTTP de los clientes, es por medio de esta que podremos generar nuestras consultas y peticiones de acuerdo a las necesidades que tenga el tesista.



Segmento de Código 3.22. Creación de un Custom Controller

A continuación se detalla uno de los métodos que se encarga de insertar un nuevo usuario en la base de datos. El método es contiene tres cabeceras, HttpPost que indica el tipo de acción HTTP que ejecutara, en este caso POST, ActionName en el cual se asigna un nombre por el cual el método podrá ser invocado a través del cliente que está en la aplicación y finalmente Route que indica directorio de esta.

```
public class QUERYController : ApiController{
    public ApiService Services { get; set; }
    public MobileServiceContext context { get; set; }
    [HttpPost]
    [ActionName("alluserlist")]
    [Route("api/customs/alluserlist/")]
    public async Task<string> AddUser(APPUser _AppUser)    {
        string FBId_Response = string.Empty;
        using (this.context = new MobileServiceContext())
        {
            var result = this.context.APPUsers.Where(x =>
x.FBId.Equals(_AppUser.FBId));
            if (result.FirstOrDefault() == null)
            {
                this.context.APPUsers.Add(_AppUser);
                await this.context.SaveChanges();
            }
            FBId_Response = this.context.APPUsers.Where(x =>
x.FBId.Equals(_AppUser.FBId)).FirstOrDefault().Id;
        }
        return FBId_Response;
    }
}
```

Segmento de Código 3.23. Custom Controller - Inserción de Usuario

Como se puede apreciar en el segmento de código 3.23, el método AddUser recibe un objeto de tipo APPUser, entonces se crea la instancia MobileServiceContext la cual accede a todos los Controller de las Entidades, a través de este verificamos que el nuevo usuario no este registrado, para ello se efectúa una consulta de búsqueda usando su ID de Facebook puesto que este también es único, de no encontrarse, se lo registra, finalmente se efectúa otro consulta para obtener el id único del usuario y este es retornado.

3.7. Prueba Completa

Este segmento es detallado en el capítulo 4 de la presente tesis.

3.7.1. Planeación de la siguiente actualización.

En base a la retroalimentación obtenida de los usuarios gracias a sus comentarios y test de usabilidad llenados, además de observaciones propias del tesista, se tienen las siguientes características a ser consideradas en una siguiente actualización:

- Contar con la aplicación en otros idiomas como el alemán, francés, italiano e inglés principalmente.
- Considerar agregar más y mejores características a la aplicación con Windows Phone 8.1 como la integración con Cortana y que esta pueda ser utilizada por medio de comandos de voz.
- Desarrollar un método de autenticación alternativo a Facebook. Los dos candidatos más fuertes son: Autenticación vía cuenta de correo y creación de cuenta propia para la App.
- Comenzar la versión preliminar para Android.
- Incrementar la confiabilidad del módulo de práctica y reconocimiento de signos con Kinect de la Aplicación Windows 8.1.

3.8.Publicación de la aplicación

El proceso de publicación de una App de Windows/Windows Phone es similar al de otros mercados en general, se tienen los siguientes pasos:

- Registro del desarrollador en la tienda de aplicaciones.
- Subida de una App a la tienda de aplicaciones.
- Revisión de la App por parte de personal de la tienda de aplicaciones.
 - En caso de encontrarse errores o problemas en general, la App no puede ser publicada y tiene que ser subida de nuevo, con las correcciones respectivas.
 - En caso de no encontrarse errores, la App está lista para ser publicada.
- Publicación de la App en la tienda de aplicaciones.

3.8.1. Definición de un modelo de negocio.

Se tuvo que escoger un modelo de negocios híbrido que combina aspectos del modelo “Freemium” y el de “Ads” o conocido también como propagandas en la aplicación. Se descartó al modelo “In-App purchasing” por su concepto de ofrecer características limitadas de la aplicación de manera gratuita hasta que el usuario realice un pago.

Entendiendo al modelo “Freemium”

Este modelo de negocios sugiere la habilitación de nuevas características para el usuario una vez que este compre una versión completa de la aplicación. Si bien esto va en contra de lo que se planteó en un principio de la aplicación, cabe destacar que en ningún momento el usuario contara con características reducidas.

El único pago que tendrá que realizar el usuario es el de una cuota única para deshacerse de las propagandas que se desplieguen en la aplicación. En todo momento, desde su descarga, el usuario contara con todas las características que tiene la aplicación.

3.8.2. Establecimiento de precios en base al modelo o modelos de negocio escogidos
Entonces, la aplicación es de descarga gratuita y viene con propagandas incrustadas no intrusivas en diferentes páginas. El precio que el usuario debe pagar para poder deshacerse de estas propagandas es de 1\$ dólar americano. Dicho precio es casi un estándar en el mundo de las aplicaciones móviles.

Las ganancias que se obtienen por medio de las propagandas funcionan en base a una fórmula similar a la siguiente:

$$\text{Ganancia} = \text{NroClicks} * \text{CteG}$$

Donde:

- NroClicks = Numero de clicks o taps realizados por los usuarios sobre una propaganda dentro de la aplicación.
- CteG = Dinero otorgado por la compañía por cada “click” realizado sobre una propaganda.
- Ganancia = El total de dinero obtenido.

Ahora bien, la constante CteG suele tener un valor muy bajo que ronda los 0.5 centavos de dólar americano dependiendo a la empresa que se encargue de la monetización de la aplicación.

Constante CTR

$$\text{CTR} = \frac{\text{Clicks}}{\text{Impresiones}}$$

La constante CTR o “Click through rate” por sus siglas en ingles es un valor que indica el promedio de “clicks” que realizar un usuario después de haber visto N propagandas, también conocidas como “Impresiones”. El valor de la constante CTR determina que tan probable es que los usuarios de la aplicación hagan click sobre una propaganda. (Bea, 2013)

El valor promedio de la constante CTR es de 2.3% para el tipo de propaganda que se está utilizando en la aplicación (propagandas de tipo Banner). Haciendo los cálculos para obtener una ganancia promedio de 20\$ dólares americanos por día, se tendría lo siguiente:

$$NroClicks = \frac{20\$}{0.005 \frac{\$}{click}}$$

$$NroClicks = 4000 \text{ clicks}$$

$$Posibilidad \text{ de } 1 \text{ click} = CTR * Nro \text{ de usuarios de la aplicacion}$$

$$Nro \text{ de usuarios de la aplicacion} = \frac{Posibilidad \text{ de } 1 \text{ click}}{CTR}$$

Asumiendo que queremos tener el 100% de certeza de que al menos un usuario hará click sobre una propaganda:

$$Nro \text{ de usuarios de la aplicacion} = \frac{100\%}{2.3\%}$$

$$Nro \text{ de usuarios de la aplicacion} = 44$$

Entonces:

$$\begin{aligned} Nro \text{ de usuarios necesarios para generar } 20\$ \text{ diarios} \\ = Nro \text{ de usuarios requeridos para un click} * \text{numero de clicks requeridos} \end{aligned}$$

$$Nro \text{ de usuarios necesarios para generar } 20\$ \text{ diarios} = 44 * 4000$$

$$Nro \text{ de usuarios necesarios para generar } 20\$ \text{ diarios} = 176000$$

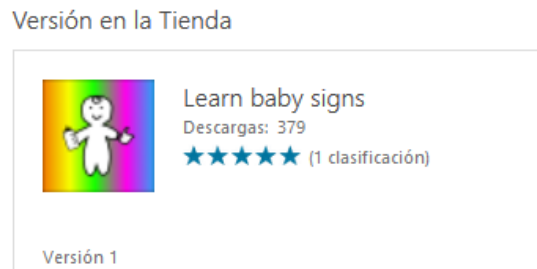
Se necesitan aproximadamente de 176000 usuarios para poder generar ingresos de 20\$ diarios en la aplicación, este no es un número exacto, ya que depende del valor de constantes entregadas por las empresas que se encargan de la monetización de las aplicaciones, así como también del movimiento actual de las aplicaciones móviles. También es importante recalcar que 176000 usuarios no son un número inalcanzable, considerando la tendencia actual del mercado. Solo para rescatar algunos números: Alrededor de 12 millones de descargas son realizadas diariamente en las tiendas de Windows Phone 8.1 y Windows 8.1, con un incremento de 390% de ganancias para los desarrolladores de estas plataformas 110 desde que salieron al mercado, aproximadamente hace un año y medio. (Stott, 2014).

3.8.3. Publicación de la aplicación en la tienda de Windows.

Actualmente ya se encuentra una versión preliminar de la aplicación tanto en la tienda de Windows, como en la de Windows Phone.

Windows Store

Como se observa, en el último mes, la aplicación para computadoras de escritorio, laptops y tablets, alcanzo las 379 descargas.



Segmento de Código 3.24. Aplicación subida en el Store de Windows

Haciendo uso de las herramientas del Centro de Desarrollo de Windows se obtuvo las siguientes métricas y estadísticas:

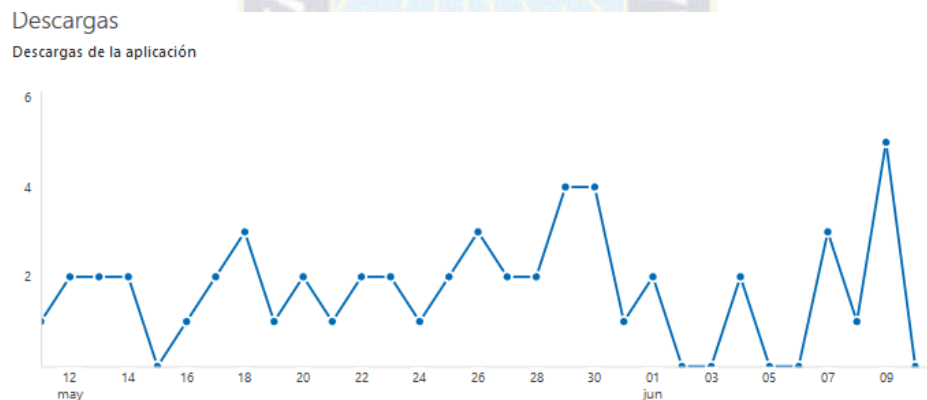


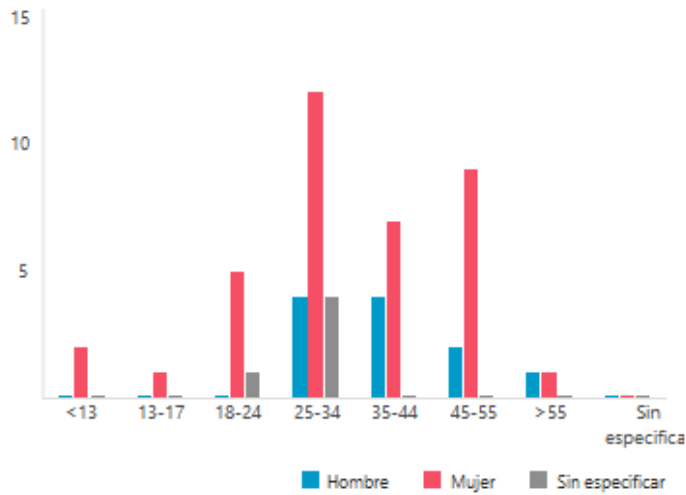
Figura 3.32. Descargas de la aplicación en las últimas 2 semanas

(Centro de Desarrollo de Windows, 2015)

Como se observa en la figura a continuación, la aplicación presenta gran aceptación en 6 mercados internacionales, vale resaltar el mercado estadounidense, además que las principales consumidoras de este son en su mayoría mujeres de entre 25 y 34 años.

Descargas por grupo de edad

Descargas de la aplicación



Descargas por mercado

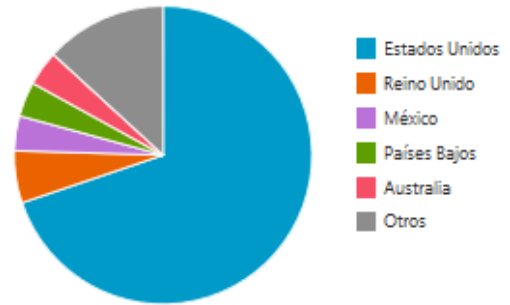


Figura 3.33. Estadísticas de Mercado y de grupos de interés.

(Centro de Desarrollo de Windows, 2015)

Windows Phone

La aplicación para Smartphones del ecosistema de Microsoft, a la fecha tuvo menos aceptación que la de Windows, alcanzando únicamente las 41 descargas.

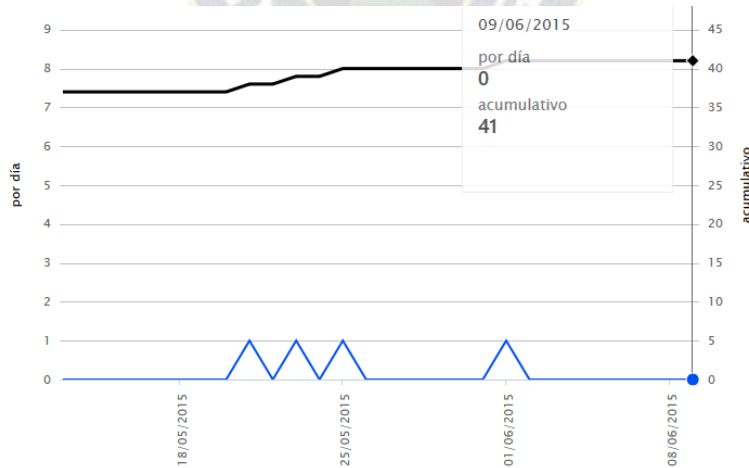


Figura 3.34. Grafica acumulativa de descargas de la Aplicación.

(Centros de Desarrollo de Windows Phone 2015)

Sin embargo se observó que la aplicación para Smartphones obtuvo mayor alcance en relación a mercados, llegando así a 14 países diferentes, y aun resaltando entre estos el país estadounidense.

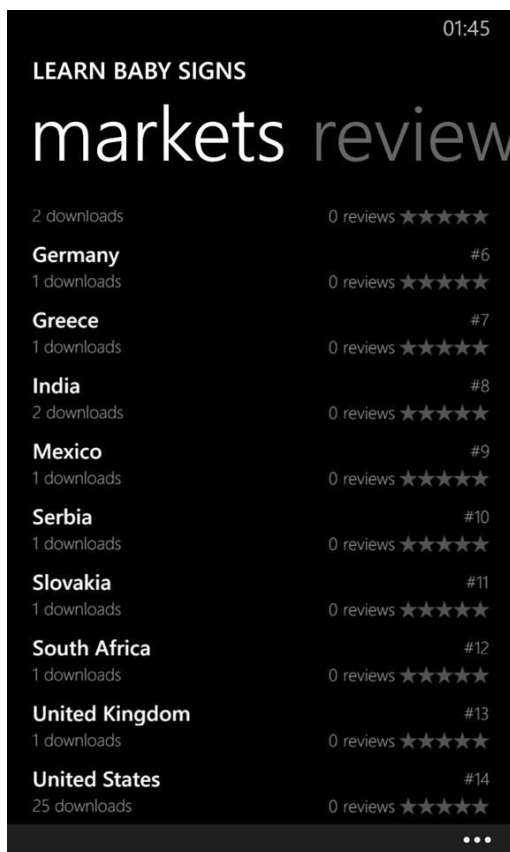


Figura 3.35. Mercados que han descargado la Aplicación (Aplicación Dev Center)

3.8.4. Publicación de la Aplicación.

En el proceso de redacción de este documento, la aplicación se encontraba en su fase final de desarrollo, de esta manera se pudieron corregir errores detectados gracias a la temprana retroalimentación obtenida de una publicación Alpha que se distribuyó a un número reducido de usuarios. Al momento de la defensa de tesis, la aplicación estará publicada en ambas tiendas de aplicaciones Windows y Windows Phone.

A continuación se describe brevemente los pasos del proceso de publicación de una aplicación en la tienda de Windows y Windows Phone, es necesario aclarar que el procesos es el mismo para ambas tiendas.

Tabla 3.16. Pasos para la publicación de la aplicación en las Tiendas de Windows

Paso	Descripción
Cuenta de Microsoft.	<ul style="list-style-type: none"> • Es requerida para acceder al panel de la tienda de Windows y Windows Phone. • Su creación es gratuita.
Abrir una cuenta de desarrollador con la cuenta de Microsoft.	<ul style="list-style-type: none"> • Es gratuito si se está registrado como miembro de la comunidad de Microsoft Student Partner. • Existen dos tipos de Cuenta. <ul style="list-style-type: none"> - Individual / Costo 19 USD - Empresa / 99USD
Elegir y reservar el nombre de la aplicación.	<ul style="list-style-type: none"> • Este proceso se puede realizar en cualquier momento. • El nombre de la aplicación debe ser único. • El nombre también estará disponible en ambas tienda.
Elección de clasificación por edad que se aplicara a la aplicación.	<ul style="list-style-type: none"> • Es posible que deba crear formularios GDF. • Es necesario para contar con restricciones.
Elección de un modelo comercial.	<ul style="list-style-type: none"> • Las tiendas de Windows y Windows Phone admite tanto aplicaciones gratuitas como una serie de franjas de precio.
Países y regiones en lo que se distribuirá la aplicación.	<ul style="list-style-type: none"> • Las tiendas admiten mercados e idiomas que tengan más sentido para ti y para tu aplicación.
Describir la aplicación y tomar capturas de pantalla.	<ul style="list-style-type: none"> • Es un factor clave para animar a los usuarios a descargar la aplicación. • Si es a escala internacional, la descripción deberá hacerse en distintos idiomas.
Prueba de aplicación con el Kit para la certificación de aplicación de Windows.	Puedes usar este kit para probar la aplicación e identificar posibles problemas que pueda tener, antes de que la envíes a la Tienda Windows.
Enviar la aplicación a la tienda.	Cuando se hayan cumplido todos los pasos anteriores, ya puedes enviar la aplicación a la tienda.

Fuente: (Información general sobre la publicación de una aplicación en la Tienda Windows, 2015)

4.1.Pruebas.

Al tratarse de una aplicación diseñada en primera instancia para padres de familia con bebés, se realizó la prueba directamente a 4 parejas de padres con edades de entre 23 a 29 años que tienen hijos de entre los 7 a 19 meses de edad.

Para las pruebas se hizo uso de los siguientes dispositivos móviles:

Tabla 4.17. Dispositivos móviles de involucrados en la pruebas.

Smartphone
Nokia Lumia 520.
Nokia Lumia 820.
Nokia Lumia 920.
Nokia Lumia 1020.

Además de computadores de escritorio y/o portátiles que cumplen con las siguientes características:

Tabla 4.18. Requerimientos de hardware de la Aplicación

Requerimientos	
Sistema Operativo y Arquitectura	<ul style="list-style-type: none"> • Windows 8 (x64) • Windows 8.1 (x64)
Configuración de Hardware	<ul style="list-style-type: none"> • Procesador 64 bits (x64) 3.1 GHz • 4 GB de Memoria • USB 3.0 • DirectX 11

Fuente: (Kinect for Windows, 2015)

La prueba estuvo constituida de los siguientes pasos:

- Descarga de la aplicación de la tienda de Windows y su posterior instalación.
- Registro del usuario en la aplicación a través de su cuenta de Facebook.
- Explicación de los módulos de la aplicación por parte del tesista al usuario.
- Practica de los signos de las lecciones en la aplicación a través del módulo de reconocimiento con Kinect.
- Registro de sus avances por medio de la aplicación.
- Llenado del formulario de retroalimentación presente en la aplicación por parte del usuario.

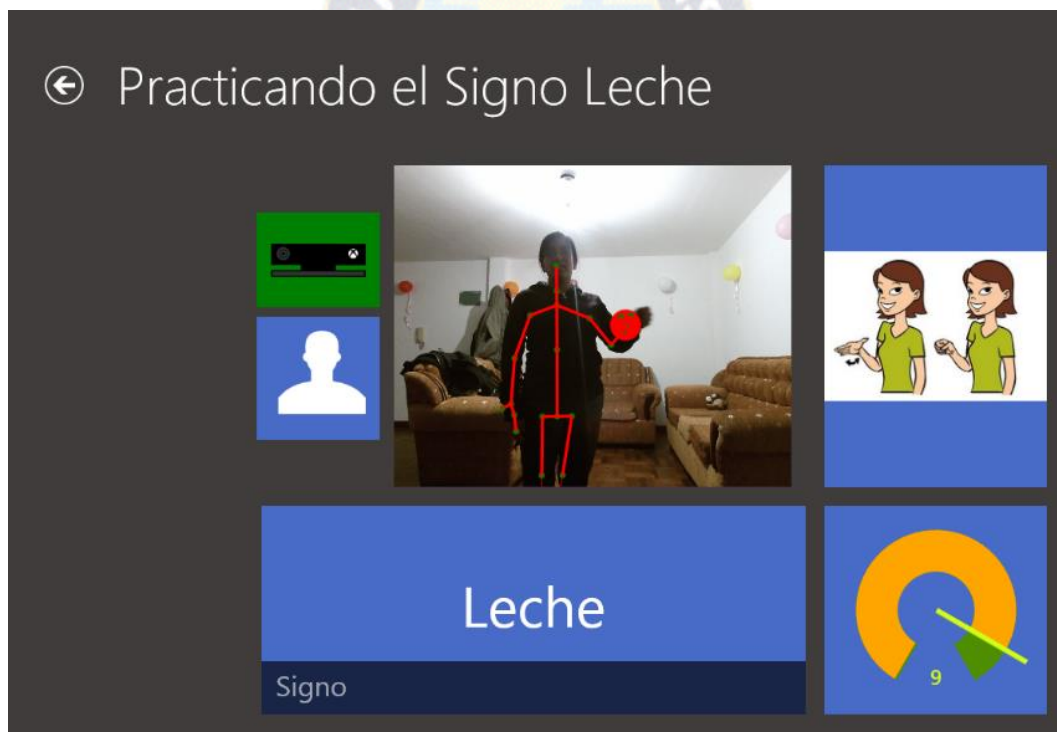


Figura 4.36. Madre Practicando Signo Leche (Aplicación)



Figura 4.37. Madre Practicando Signo Gato

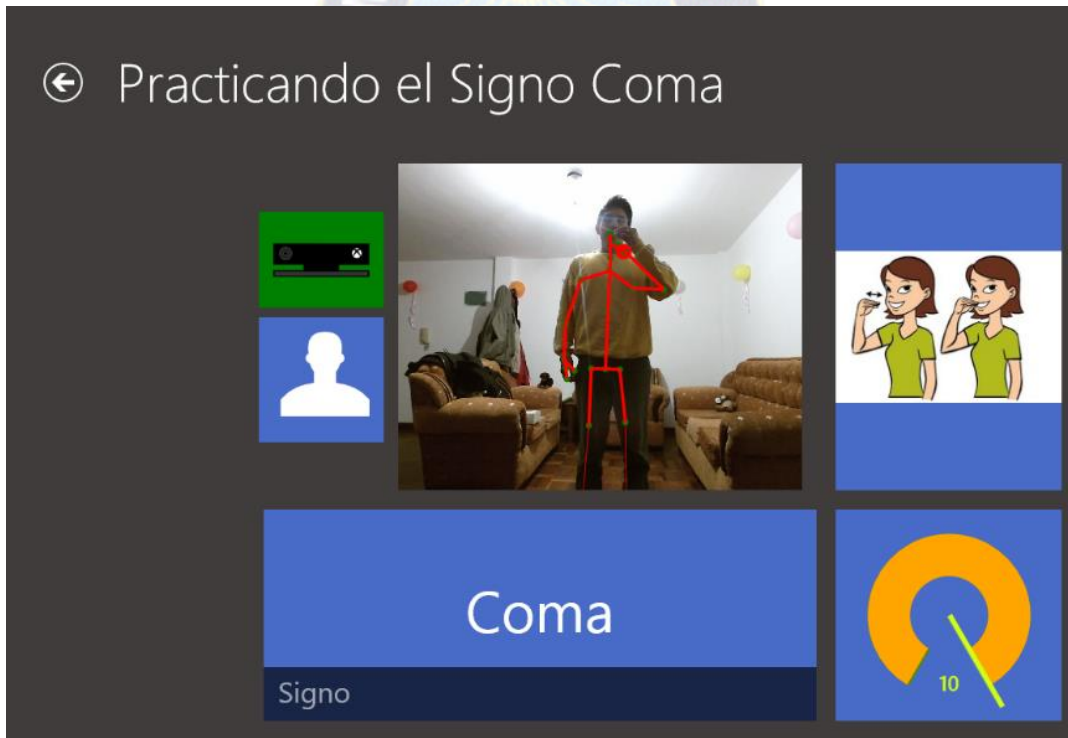


Figura 4.38. Padre Practicando Signo Coma (Aplicación)



Figura 4.39. Padre Practicando Signo Coma



Figura 4.40. Padres de Familia Probando la Aplicación

4.2. Análisis de los comentarios de los usuarios.

De los ocho usuarios que probaron la aplicación se recibieron comentarios más positivos que negativos. Sin embargo, a continuación se citan solo a aquellos comentarios que no fueron completamente positivos para poder realizar un análisis más interesante:

- “Puedo practicar con la mayoría de los signos de las clases de PC con Kinect, sin embargo algunos de estos signos no son interpretados correctamente por la aplicación”. Es una observación válida, pues al momento es una de las grandes limitantes del SDK de Kinect V2, el cual no nos permite crear gestos que sean reconocidos al 100%, más cuando estos son muy complejos, se prevé realizar ajustes e incrementar la lógica del módulo de reconocimiento para poder disminuir al mínimo este inconveniente en una futura actualización de la aplicación.
- “La aplicación es fluida, me gusta mucho, pero quisiera que no fuese necesario tener una cuenta de Facebook para poder usarla”. El uso de Facebook como base de datos ha tenido respuestas buenas y malas por parte de los usuarios, sobre todo porque muchos padres de familia que no tienen una cuenta en esta red social. Por tanto, se deberá de analizar el uso de formas de autenticación alternativas en la siguiente actualización.
- “Me gustaría contar con más funcionalidades en la aplicación Móvil, como prácticas de selección múltiple con las ilustraciones”. Puesto que se puso más énfasis en el desarrollo de la aplicación de PC y su módulo de reconocimiento con Kinect, la aplicación móvil es solo de apoyo técnico, sin embargo se tomarán en cuenta las observaciones en la siguiente versión.
- “¡La aplicación también debería soportar múltiples idiomas!”. La usuaria que utilizó la App en una fase temprana es anglo parlante y solo había podido verla funcionando en español, al momento de defender la tesis la aplicación está disponible solo en español, sin embargo cuenta con la configuración lista para soportar más idiomas.
- “La aplicación móvil estará disponible para dispositivos Android”. Si bien la aplicación fue planeada correr en plataformas con sistemas operativos del ecosistema de Microsoft, es posible analizar el desarrollar una solución Android a partir del actual núcleo de código de la aplicación gracias a frameworks como Xamarin.

4.2.1. Obtención de estadísticas en base al Test de Usabilidad.

En Base al cuestionario de usabilidad se tienen los siguientes resultados:

Utilidad percibida de la aplicación.

Analizando la figura 4.6., se observa que la aplicación cumple con las expectativas de un 60% de los padres de familia que participaron de la prueba, sin embargo aún se debe trabajar en mejorar aspectos de la aplicación para que el 26% de estos a los que no llenó su expectativa disminuya y tenga mayor aceptación.



Figura 4.41. Cumplimiento de expectativas de la App.

Se puede observar que todos los padres de familia, en el tiempo que utilizaron la aplicación, efectivamente estuvieron aprendiendo lengua de signos de bebé, sin embargo, se deberá mejorar el porcentaje de 62% 'Efectivamente', en este sentido se deberá tomar a consideración las opiniones y recomendaciones, de esta manera pensar en una nueva actualización de la aplicación y mejorar su efectividad.

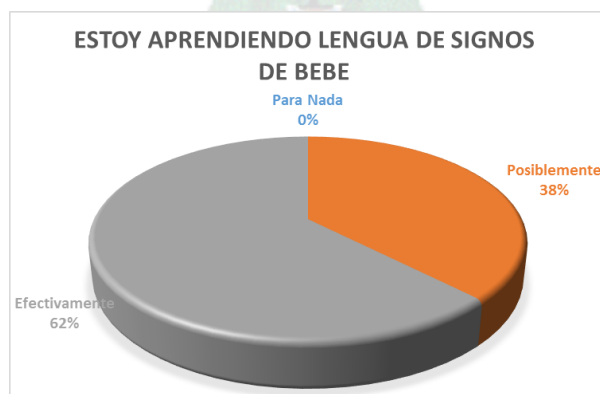


Figura 4.42. Aprendizaje de Lengua de Signos

Tal y como en el detalle de la figura 4.7., se tiene el mismo resultado, de tal manera se debe mejorar la aplicación a partir de las recomendaciones y sugerencias de los padres de familia.

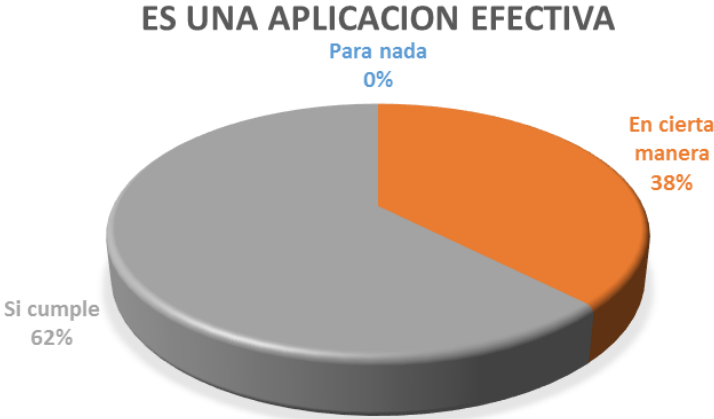


Figura 4.43. Efectividad de la Aplicación

Se puede decir en base a los resultados de la figura 4.9. que la aplicación a presentado gran utilidad a 75% de los padres de familia que utilizaon la aplicacion, lo cual es una cifra bastante aceptable, sin embargo aun hay que pensar en el 25% que respondieron “No”, para cubrir su expectativa y de alguna manera disminuir esta cifra.



Figura 4.44. Utilidad de la App

Es importante que la aplicación tenga mayor alcance de usuarios, tal y como señalan las cifras de la figura 4.10., un 87% de los padres de familia podría recomendar la aplicación.

RECOMENDARIAS LA APLICACION A OTRAS PERSONAS

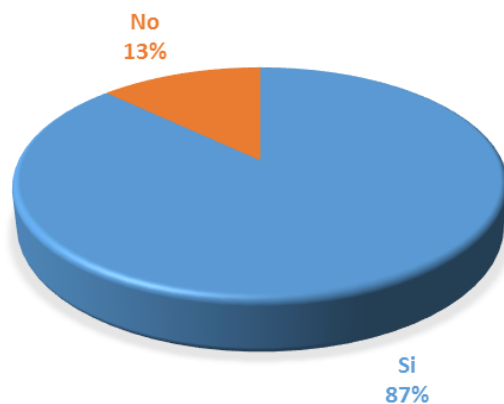


Figura 4.45. Recomendación de la App a otros usuarios.

Facilidad de uso de la aplicación

Es verdaderamente importante que la aplicación sea entendible para los padres de familia, aun cuando solo a un 87% de ellos la aplicación no les resulto confusa, será necesario mejorar la calidad de esta en base a la retroalimentación obtenida.

LA APLICACION ES CONFUSA

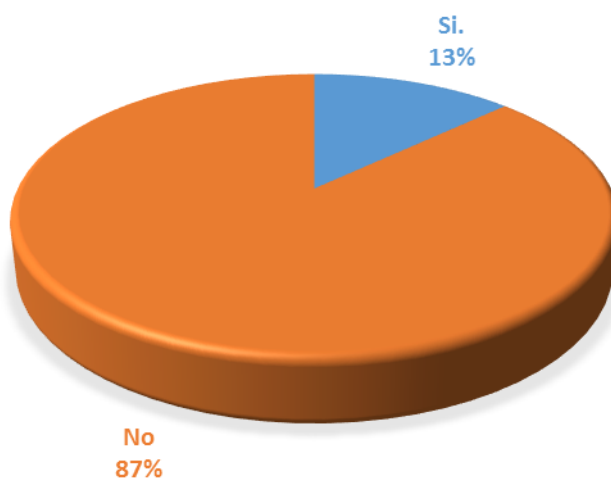


Figura 4.46. La Aplicación es confusa?

Sera necesario trabajar un poco más en la aplicación y así bajar estas cifras de despliegue de errores, aun cuando el porcentaje no es elevado como se muestra en la figura 4.12, se debe pensar en la experiencia de los usuarios.



Figura 4.47. Frecuencia de despliegue de mensajes de error en la Aplicación.

Los resultados presentados en la figura 4.13., son muy importantes puesto que el recurso más sobresaliente e impórtate en la aplicación es la utilización del Kinect, sin embargo se tiene como traba principal las limitaciones del SDK del Kinect V2, entonces se deberá esperar una actualización de dicho SDK que mejore la eficiencia del Kinect, ya que a la fecha no existe kits de desarrollo de terceros para la implementación de Kinect en aplicaciones.

ES FLUIDA LA INTERACCION CON EL KINECT

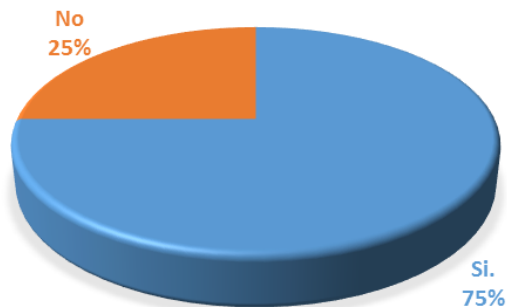


Figura 4.48. Interacción con el Kinect

A partir de la figura 4.14., se puede apreciar que el 100% de los usuarios pudo aprender a usar la aplicación haciendo el menor esfuerzo posible.

ESFUERZO NECESARIO PARA APRENDER A USAR LA APLICACION



Figura 4.49. Esfuerzo necesario para aprender a utilizar la aplicación.

En este caso, será necesario una prueba más profunda y así localizar el origen de estos de estos comportamientos, ya que en el desarrollo de la aplicación, se implementaron múltiples medidas de control y manejo de errores. Según comentarios de los usuario, el comportamiento inesperado más común es el de navegar de una página directamente al menú principal después de terminar una práctica de signos.

TIENDE A REALIZAR COMPORTAMIENTOS INESPERADOS

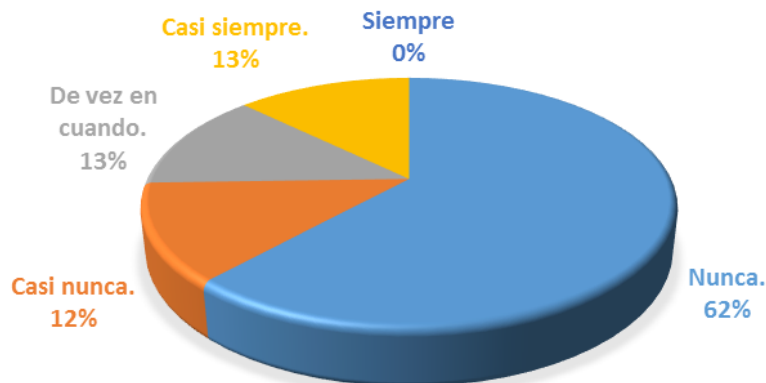


Figura 4.50. Frecuencia de comportamiento inesperado en la aplicación.

A pesar de los resultados que se obtuvieron a través de las estadísticas anteriores, la aplicación presenta un desempeño ‘Muy Bueno’ del 74%, entonces, tomando en cuenta las observaciones de los padres de familia.

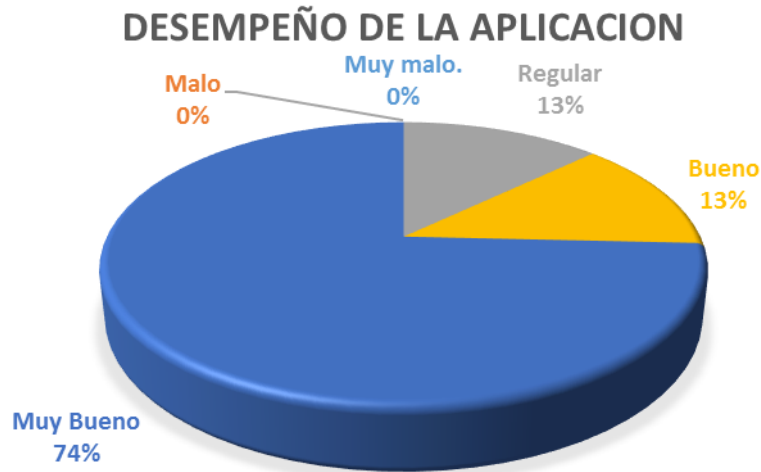


Figura 4.51. Desempeño de la Aplicación.

4.2.1.1. Conclusiones del test de usabilidad.

Los datos más interesantes que pudieron ser extraídos del test de usabilidad son por ejemplo el hecho de que la App cumple las expectativas de más del 60% de los usuarios encuestados, esto apoyándose en que el 75% de los usuarios cree que la aplicación es útil y un 62% de ellos afirmar estar aprendiendo lenguaje de signos de bebé a través de la aplicación. Mientras que solo un 13% de los usuarios cree que se debe realizar esfuerzo para aprender a utilizar la App, el 87% de encuestados concuerda en que la App no es confusa.

4.3. Pruebas de funcionalidad y certificación.

Esta prueba se realizó a través del Kit para la certificación de aplicaciones en Windows, el cual contiene un número de pruebas que pueden garantizar que la aplicación esté lista para publicarse en la Tienda Windows y que se pueda proporcionar aplicaciones de gran calidad en la Tienda Windows y Windows Phone.

La tabla 4.3 refleja las pruebas que el Kit realiza sobre la aplicación.

Tabla 4.19. Pruebas realizadas en aplicaciones de la Tienda Windows

Nombre de la prueba	Windows 8.1	Windows Phone 8.1
Prueba de bloqueos y cuelgues	Sí	No
Manifiesto de la aplicación	Sí	Sí
Prueba de características de seguridad de Windows	Sí	Sí
Prueba de API de la Tienda Windows admitidas	Sí	Sí
Prueba de rendimiento	Sí	No
Prueba de recursos del manifiesto de la aplicación	Sí	Sí
Prueba de configuración de depuración	Sí	Sí
Prueba de codificación de archivos	Sí	Sí
Compatibilidad del nivel de característica de Direct3D	Sí	No
Recorte después de la suspensión en Direct3D	Sí	No
Prueba de funcionalidades de la aplicación	Sí	Sí
Validación de metadatos de Windows en tiempo de ejecución	Sí	Sí
Archivos apropiados para la plataforma	Sí	Sí
Comprobación de estructura de directorios admitida	Sí	No
Integridad del paquete	No	Sí
Prueba de inicio de versión de plataforma.	Sí	No
Validación del controlador de cancelación de tareas en segundo plano.	No	No
Validación previa al lanzamiento de la aplicación.	Sí	No
Prueba de uso de recursos	Sí	Sí

Fuente: (Pruebas del Kit para la certificación de aplicaciones en Windows para aplicaciones de la Tienda Windows, 2015)

4.3.1. Realización de la prueba.

Este proceso es importante como se explicó en el capítulo 4 en la sección de publicación de la aplicación, ya que de no pasar esta prueba, la aplicación no podrá ser publicada en la tienda de Windows.

Como se muestra en la Figura 4.17, una vez creado el paquete de la aplicación, el software presenta la opción de poder analizarla a través del Kit.

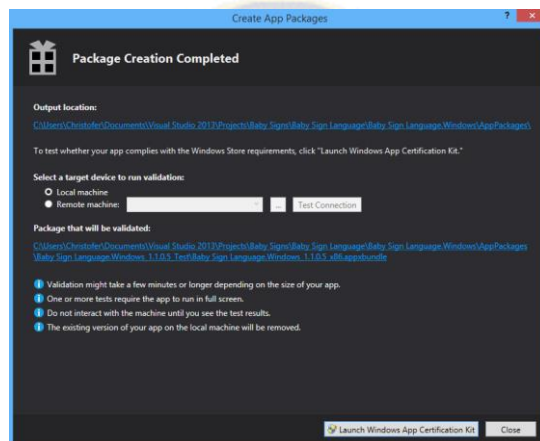


Figura 4.52. Creación del paquete de Aplicación y Lanzamiento de pruebas.

En la figura 4.18, se procede a seleccionar todas las pruebas que puede realizar el kit tal y como muestra la tabla 4.3

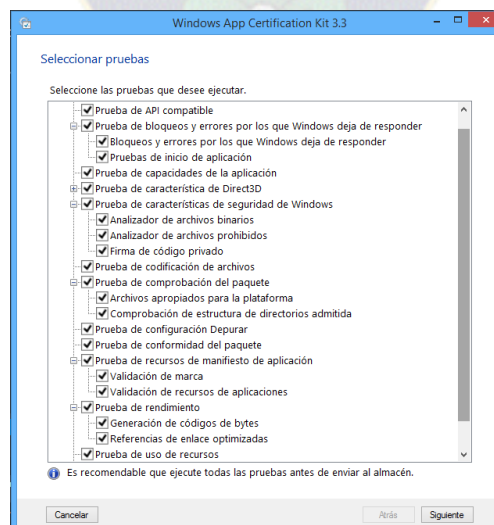


Figura 4.53. Selección de pruebas a realizarse.

Una vez finalizada las pruebas del kit sobre la aplicación, este procede a analizar los resultados de dichas pruebas y decidir si esta cumple con los estándares.

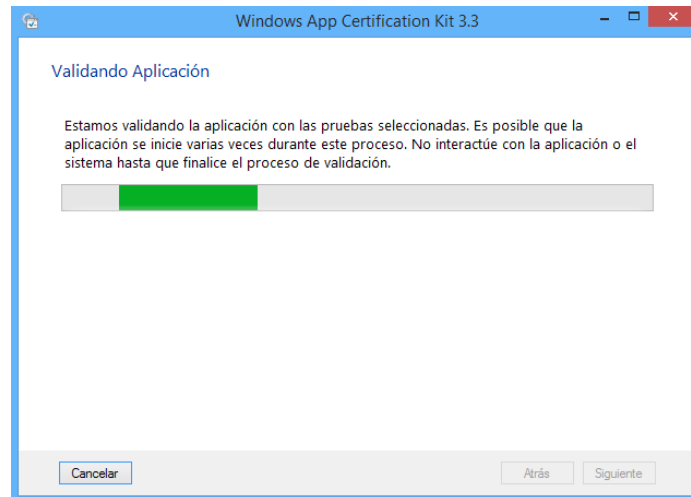


Figura 4.54. Validación de la aplicación a partir de las pruebas realizadas

Finalmente, ya analizados los resultados, el kit de pruebas despliega el resultado y genera un breve informe.

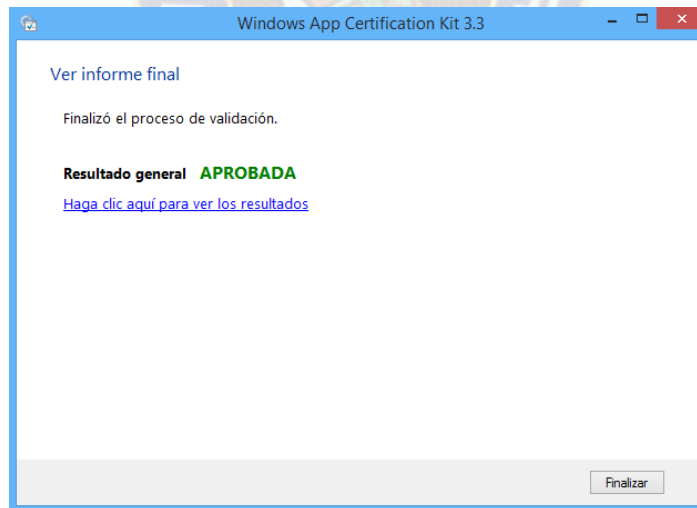


Figura 4.55. Finalización de las pruebas del Kit.

4.3.1.1. Resultados de la pruebas de funcionalidad y certificación.

La aplicación paso todas las pruebas de la tabla 4.3 del Kit de Certificación, cumpliendo así con todos estándares de calidad requeridos por la Tienda de Windows como se muestra en la figura 4.20.

Tabla 4.20. Resultados del Kit de pruebas funcionalidad y certificación.

Tipo de Pruebas	Pruebas	Resultado
Prueba de bloqueos y errores por los que Windows deja de responder	Pruebas de inicio de aplicación	APROBADA
	Bloqueos y errores por los que Windows deja de responder	APROBADA
Prueba de conformidad del paquete	Manifiesto de aplicación	APROBADA
Prueba de características de seguridad de Windows	Analizador de archivos binarios	APROBADA
	Analizador de archivos prohibidos	APROBADA
	Firma de código privado	APROBADA
Prueba de API compatible	API compatibles	APROBADA
Prueba de rendimiento	Generación de códigos de bytes	APROBADA
	Referencias de enlace optimizadas	APROBADA
	Validación de marca	APROBADA
	Referencias de enlace optimizadas	APROBADA
Prueba de configuración Depurar	Configuración Depurar	APROBADA
Prueba de codificación de archivos	Codificación de archivos UTF-8	APROBADA
Prueba de característica de Direct3D	Compatibilidad de nivel de característica de Direct3D	APROBADA
	Recorte Direct3D tras suspensión	APROBADA
Prueba de capacidades de la aplicación	Capacidades de uso especial	APROBADA
Validación de metadatos de Windows en tiempo de ejecución	Prueba del atributo ExclusiveTo	APROBADA
	Prueba de ubicación de tipo	APROBADA
	Prueba de distinción entre mayúsculas y minúsculas del nombre de tipo	APROBADA
Prueba de comprobación del paquete	Archivos apropiados para la plataforma	APROBADA

5.1.Conclusiones

Se ha desarrollado una aplicación para la enseñanza de los signos de bebé para plataformas del ecosistema Microsoft, Windows 8.1 y Windows Phone 8.1, el cual ayuda a personas a aprender en primera instancia varios signos utilizados con frecuencia para su uso cotidiano, como ser los signos que se refieren familiares y/o a personas importantes, hasta alimentación y animales entre otros, puntualmente los signos que están agrupados por tipo en la aplicación y que pueden ser practicados con el Kinect V2, para que el usuario pueda empezar un aprendizaje moderado pero adecuado, dando cumplimiento al objetivo principal de la presente tesis.

5.1.1. Objetivos secundarios.

Se buscaron antecedentes de proyectos previos incluyendo tesis que hayan tocado el mismo tema. (Sección “Antecedentes” del capítulo 1: Introducción) para dar cumplimiento al primer objetivo secundario.

Se han definido los requerimientos de la aplicación para establecer límites y alcances de la misma, además del software y hardware que se utilizó. (Sección “Alcances y límites” del capítulo 1: Introducción) para dar cumplimiento al segundo secundario.

Se desarrolló una aplicación que utiliza características del Kinect como el reconocimiento de cuerpos y movimientos, estas características fueron utilizadas en el módulo de práctica y reconocimiento de signos de la aplicación de Windows 8.1 (Sección “Definición y desarrollo de segmentos” del Capítulo 3: Marco Aplicativo) para dar cumplimiento al tercer objetivo secundario de la aplicación.

Se desarrolló un módulo de registros de avance del usuario en la aplicación (Sección “Definición y desarrollo de segmentos” del Capítulo 3: Marco Aplicativo) para dar cumplimiento al cuarto objetivo secundario de la aplicación.

Se utilizó una nueva metodología ágil orientada al desarrollo de aplicaciones móviles (Sección “Desarrollo móvil Ágil MAD” del Capítulo 2: Marco teórico) para dar cumplimiento al cuarto objetivo secundario.

Se han realizado las pruebas de funcionalidad acordes (Sección “Pruebas de funcionalidad, y certificación” del Capítulo 4: Pruebas y resultados) para dar cumplimiento al quinto objetivo secundario.

5.2.Recomendaciones

5.2.1. Migrando a nuevas tecnologías

Es un hecho de que la tecnología acelera a paso veloz de lo que en muchas ocasiones podemos seguir como desarrolladores de software, siempre se tendrán mejoras, actualizaciones, adaptaciones de mejores prácticas en desarrollo, migraciones a nuevas bases de datos, entre otros.

Sin ir demasiado lejos, en los días del 29 de Abril al 1 Mayo del 2015, tiempo en el que esta tesis estaba siendo escrita, Microsoft anuncio la expansión en el desarrollo de aplicaciones universales con su sistema Operativo Windows 10, llegando así no solo a PC, Smartphones y tablets, sino también a Consolas de juegos XBOX ONE, Surface entre otras. Esto quiere decir que ahora, el concepto de aplicaciones universales ha sido extendido también hacia la unificación de múltiples dispositivos.

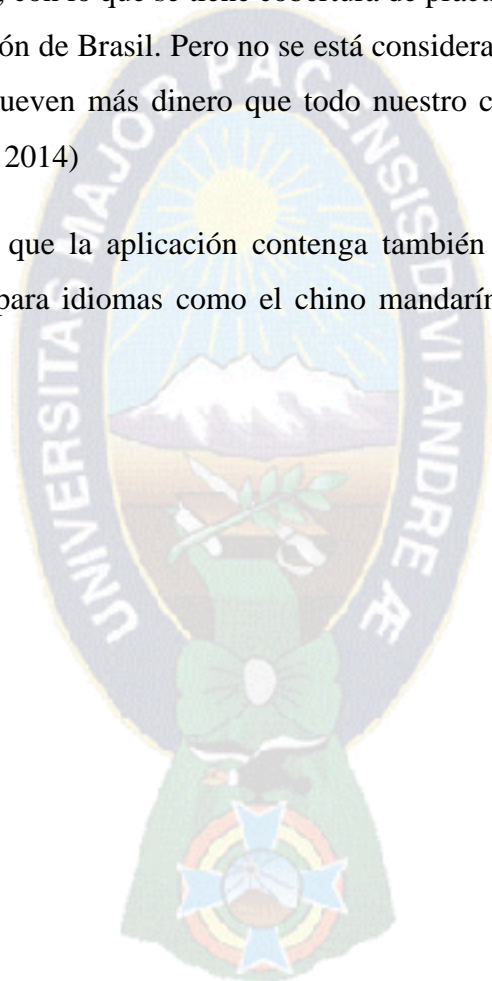
5.2.2. Expansión a nuevos mercados

Seguir con el proceso de investigación de desarrollo para dispositivos móviles utilizando tecnologías web, ya que empresas como Google y Apple están apostando por HTML5, CSS3

y JavaScript para la realización multimedia o de animaciones, entre estos procesos se puede destacar el uso de frameworks como JQuery Mobile, Sencha, Titanium, Phonegap, etc., que facilitan notablemente el desarrollo de aplicaciones móviles, agregando una ventaja más que es la de contar con aplicaciones multiplataforma.

La aplicación desarrollada para la presente tesis, incluye inicialmente dos idiomas para los usuarios: Español e inglés, con lo que se tiene cobertura de prácticamente todo el continente americano, con la excepción de Brasil. Pero no se está considerando a países como China o India, que por sí solos mueven más dinero que todo nuestro continente, sin considerar a Estados Unidos. (Charles, 2014)

Se recomienda por tanto que la aplicación contenga también en futuras actualizaciones diccionarios de recursos para idiomas como el chino mandarín, chino cantones, japonés, coreano, entre otros.



6.1. Referencias Bibliográficas

- Algueró, M. (15 de Octubre de 2011). Las ventajas de usar el lenguaje de signos con los niños. *Guia Infantil*.
- App Flood*. (23 de Diciembre de 2014). Obtenido de What is an average mobile ad click-through rate?:
<http://appflood.com/blog/average-ctr-by-ad-formats>
- AppDesingBook*. (10 de Marzo de 2015). Obtenido de Diseñando Apps para Moviles:
<http://www.appdesignbook.com/es/contenidos/las-aplicaciones/>
- BabsySignsToo*. (23 de Noviembre de 2014). Obtenido de Babsy Signs @:
<http://www.babysignstoo.com/information/faqs>
- Baby Sign and Learn*. (15 de Octubre de 2014). Obtenido de Windows Store:
<http://www.windowsphone.com/es-bo/store/app/baby-sign-and-learn/2b899b09-29c3-4284-9a6f-442fc32a4ecb>
- Baby Sign Language Store*. (2014 de Octubre de 2014). Obtenido de BabySignLanguage:
<http://www.babysignlanguage.com/store/>
- Baby Signing Lite*. (15 de Octubre de 2014). Obtenido de Google Play:
https://play.google.com/store/apps/details?id=com.rbi.msh_dictionary_lite
- Baby Signs – Sign language for babies*. (5 de Noviembre de 2014). Obtenido de Circulo Infantil Chile:
http://www.circuloinfantil.cl/baby_texto.pdf
- BabySings - Señas para bebé*. (11 de Octubre de 2014). Obtenido de BabySings: <http://www.babysigns.cl/>
- Carter, C. (11 de Octubre de 2014). *Signos para Bebés – La guía rápida para enseñar a tu bebé*. Obtenido de Google Books: <http://books.google.com.bo/books?isbn=1300247355>
- Centro de Desarrollo*. (15 de Octubre de 2014). Obtenido de Creación de Aplicaciones de Windows Universales para cualquier dispositivo de Windows: <https://dev.windows.com/es-es/develop/building-universal-Windows-apps#using>
- Centro de Desarrollo de Windows*. (16 de Febrero de 2015). Obtenido de Microsoft:
<https://msdn.microsoft.com/es-es/library/windows/apps/dn726767.aspx>
- Charles, A. (14 de Febrero de 2014). *The Guardian*. Obtenido de Smartphone explosion in 2014 will see ownership in India pass US: <http://www.theguardian.com/technology/2014/jan/13/smart>

- Contreras Aranda, S. (2010). *Metodología de la Investigación Científica y Sistemática para la Toma de Decisiones*. Trujillo, Peru.
- Cupi Veliz, J. M. (2014). *Georreferenciación y su uso en Smartphones para la localización de personas*. La Paz, Bolivia.
- Enséñele gestos a su guagua para saber qué diablos quiere*. (14 de Octubre de 2014). Obtenido de Las Últimas Noticias: <http://www.lun.com/Pages/NewsDetail.aspx?dt=2011-04-19&PaginaId=10>
- Farkas, C. (2009). *Gestos que hablan*. Santiago de Chile: Ediciones suc.
- Gómez Galán, S. O. (1999). *El ciclo del proyecto de cooperación al desarrollo. La aplicación del marco lógico*. Madrid, España.
- González, M. A. (2001). *Lenguaje de Signos*. Mexico.
- Graciela E. Barchini, M. S. (2004). *La informática como disciplina científica*.
- Healthy Children*. (4 de Noviembre de 2014). Obtenido de Como aprenden los bebés?: <http://www.healthychildren.org/Spanish/ages-stages/baby/Paginas/How-Do-Infants-Learn.aspx>
- Jiménez, M. V. (12 de Octubre de 2014). *Lengua de Signos para bebés oyentes – Habla con tu bebé antes de que tu bebé aprenda a hablar*. Obtenido de Google Books: <http://books.google.com.bo/books?isbn=8461658760>
- Kinect for Windows*. (26 de Junio de 2015). Obtenido de Microsoft: <http://www.microsoft.com/en-us/kinectforwindows/meetkinect/features.aspx>
- Lengua de señas en los bebés y niños pequeños*. (2014 de Octubre de 2014). Obtenido de Wikipedia: http://es.wikipedia.org/wiki/Lengua_de_se%C3%B1as_en_los_beb%C3%A9s_y_ni%C3%B1os_peque%C3%B1os
- Lenguaje de signos para bebés y niños - Como comunicarte con tu hijito antes de que empiece a hablar*. (15 de Octubre de 2014). Obtenido de BabyCenter: <http://espanol.babycenter.com/a9200007/lenguaje-de-signos-para-beb%C3%A9s-y-ni%C3%B1os-c%C3%B3mo-comunicarte-con-tu-hijito-antes-de-que-empiece-a-hablar>
- M, K. (15 de Diciembre de 2014). *Venture Beat*. Obtenido de 46M apps are downloaded from Apple's App Store every day: <http://venturebeat.com/2012/05/30/app-store-downloads/>
- Microsoft Azure*. (10 de Junio de 2015). Obtenido de How to use Blob storage from .NET: <https://azure.microsoft.com/en-us/documentation/articles/storage-dotnet-how-to-use-blobs/>
- Microsoft y la computación en nube*. (15 de Mayo de 2015). Obtenido de Microsoft: <http://www.microsoft.com/government/es-xl/iniciativas/Pages/computo-en-nube.aspx>
- Model-View-Controller*. (23 de Abril de 2015). Obtenido de MSDN: <https://msdn.microsoft.com/en-us/library/ff649643.aspx>

Peru, E. S. (Junio de 2014). *ESFAP*. Obtenido de www.esfap.fap.mil.pe/jetpo/docsPDF/MetodoSistemico.pdf.

PsicoPedagogia. (8 de 10 de 2014). Obtenido de Desarrollo del Bebé:

<http://www.psicopedagogia.com/articulos/?articulo=313>

Que es Microsoft Azure? (15 de Octubre de 2014). Obtenido de Microsoft Azure: [http://azure.microsoft.com/es-](http://azure.microsoft.com/es-es/overview/what-is-azure/)

[es/overview/what-is-azure/](http://azure.microsoft.com/es-es/overview/what-is-azure/)

Stott., L. (5 de Enero de 2015). *MSDN Blogs*. Obtenido de UK Faculty Connection:

http://blogs.msdn.com/b/uk_faculty_connection/

Universo Sistemico para Ingenieros de Sistemas. (2015). Obtenido de Enfoque Sistemico:

<https://sites.google.com/site/teoriageneraldesistemas301/project-definition>

Uso del patrón Model-View-ViewModel (MVVM). (1 de Junio de 2015). Obtenido de Microsoft:

<https://msdn.microsoft.com/es-es/library/windows/apps/jj883732.aspx>

Vallotton, C. (5 de 10 de 2014). *Signing Time*. Obtenido de Signing with Babies and Children:

<http://www.signingtime.com/resources/research/>



ANEXO A.

Tabla 7.21. Requerimientos de hardware y software del Kinect V2

Requerimientos	
Sistema Operativo y Arquitectura	<ul style="list-style-type: none">• Windows 8 (x64)• Windows 8.1 (x64)
Configuración de Hardware	<ul style="list-style-type: none">• Procesador 64 bits (x64) 3.1 GHz• 4 GB de Memoria• USB 3.0• DirectX 11

Fuente: (Kinect for Windows, 2015)

ANEXO B.

Tabla 7.22. Principales aplicaciones para dispositivos móviles.

NOMBRE	PLATAFORMA	URL	TIPO	
			GRATUITO	PAGA
Baby Sign Lite	Android	https://play.google.com/store/apps/details?id=com.babysignandlearn.babysign	X	
Baby Signing Lite	Android	https://play.google.com/store/apps/details?id=com.rbi.msh_dictionary		2.5 \$
Children Sing Language	Android	https://play.google.com/store/apps/details?id=com.chldrma.chldrma&hl=es	X	
ASL Dictionary	Android	https://play.google.com/store/apps/details?id=com.babysignandlearn.asldictionary		2.7 \$
Best Baby Apps	Windows Phone	http://www.windowsphone.com/es-bo/store/app/best-baby-apps/dd0a8fe6-ed6b-422c-a220-9836d171eec8		1.9 \$
Baby Language	Windows Phone	http://www.windowsphone.com/es-bo/store/app/baby-language/91833a50-9129-4f8c-8813-65217061b896		0.99 \$
Baby Sign Language	Windows Phone	http://www.windowsphone.com/es-bo/store/app/baby_sign_language/bb571c83-5583-41ca-bbeb-ff095b3466c7	X	
Baby Sign and Learn	Windows Phone	http://www.windowsphone.com/es-bo/store/app/baby-sign-and-learn/2b899b09-29c3-4284-9a6f-442fc32a4ecb		2.49 \$
-	Windows	NO EXISTEN APLICACIONES		

ANEXO C.

Tabla 7.23. Análisis principales de aplicaciones y/o productos en el mercado

Producto	Descripción	Costos	Características	Desventajas
BABY SIGNING LITE	Esta aplicación es una versión gratuita de la aplicación completa de paga para dispositivos con Sistema Operativo Android. (<i>Baby Signing Lite</i>).	<ul style="list-style-type: none"> Gratuito 	<ul style="list-style-type: none"> Ofrece instrucciones sobre cómo hacer que los signos, así como consejos de memoria para ayudarlo a recordar. El usuario puede añadir signos a una lista de favoritos para un acceso más rápido y/o inmediato. 	<ul style="list-style-type: none"> No lleva registro del avance del bebé. Solo incluye 50 tipos de signos de los más de 400 signos tomados de la lengua de signos americana y que son usados actualmente. (Baby Signs®, (s.f.)).
BABY SIGN AND LEARN	Esta aplicación utiliza lenguaje de señas americano (ASL) signos de palabras clave y contiene más de 300 signos para enseñar a su bebé. Está disponible para dispositivos con Sistema Operativo Windows Phone. (<i>Baby Sign and Learn</i>).	<ul style="list-style-type: none"> Precio de la Aplicación \$ 2.6 	<ul style="list-style-type: none"> Demostración del cómo realizar correctamente los signos a través de videos animados. Realiza exámenes de selección correcta de los signos. 	<ul style="list-style-type: none"> No lleva registro del avance del bebé. Aplicación muy básica, en el sentido de que presenta los signos de forma cruda sin embargo, todas estas aplicaciones evidencian de manera inmediata una estructura de catálogo como se puede observar en sus respectivas descripciones e imágenes de muestra
BABY SIGN LANGUAGE	BabySignLanguage.com es un proyecto para los padres que enseñan bebés y niños pequeños temprana habilidades de comunicación. (<i>Baby Sign Language</i>).	<ul style="list-style-type: none"> Ultra Kit \$202.87 Premium Kit \$162.91 Deluxe Kit \$95.95 Standard Kit \$74.95 Flash Cards 24.99 Diccionario \$19.99 	<ul style="list-style-type: none"> Uso de tarjetas con ilustraciones de los signos. Uso de DVDs con material multimedia. Cuadros en las paredes como ayuda. 	<ul style="list-style-type: none"> Entre los materiales que contiene cada Kit de aprendizaje, no existen recursos como libretas de registro que sirvan para llevar un seguimiento del aprendizaje del bebé. El costo de cada Kit de Aprendizaje es elevado, tomando como caso Bolivia, donde que el salario mínimo nacional es de Bs. 1440, 210 \$ Aproximadamente.

The logo of Universitas Major Pacensis Dini Andre is an oval emblem. It features a sun with rays at the top, a mountain range in the middle, and a green landscape with a tree at the bottom. The text "UNIVERSITAS MAJOR PACENSIS DINI ANDRE" is written around the perimeter of the oval. Below the oval is a green ribbon with a white cross and a blue star.

DOCUMENTACION