

UNIVERSIDAD AMAZÓNICA DE PANDO
ÁREA DE CIENCIAS Y TECNOLOGÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS



TESIS DE GRADO PARA OPTAR AL TÍTULO DE
INGENIERO DE SISTEMAS

**“SISTEMA MÓVIL DE COMUNICACIÓN EN DISPOSITIVOS ANDROID PARA LA
TRADUCCIÓN DEL SISTEMA BRAILLE A TEXTO ALFANUMÉRICO EN ESPAÑOL
Y SU EXPRESIÓN FONOLÓGICA CON EL USO DE ALGORITMOS GENÉTICOS Y
METODOLOGÍAS HÍBRIDAS”**

Postulante : Univ. Carlos Alberto Osco Flores
Tutor : M. Sc. Lic. Humberto Fernández Calle
Asesor : Ing. Juan Gabriel Lazcano Balanza

Cobija - Pando – Bolivia

2016

DEDICATORIA

Mi tesis la dedico con todo mi amor y cariño a mis queridos padres y mi hermana, porque ellos me dieron la fe y fortaleza necesaria para salir siempre adelante pese a las dificultades, colocándome en el mejor camino, iluminando cada paso de mi vida brindándome la salud y esperanza para terminar esta tesis.

AGRADECIMIENTO

No puedo pasar la oportunidad de agradecer sinceramente a mis docentes *M. Sc. Lic. Juan Carlos Huanca Guanca* y al *Ing. Juan Gabriel Lazcano Balanza* por sus esfuerzos, dedicación y sus conocimientos, orientaciones, su manera de trabajar, su paciencia y motivación han sido fundamentales en mi proceso educativo sin los cuales no podría tener una formación completa como investigador en TIC y la presente Tesis, también a mi amigo y compañero de clase *Gary Oliver Guarachi Aldunate*, a los mismos se los dedico estas pequeñas líneas de recompensa.

También agradezco a esta prestigiosa Superior Casa de Estudios.

RESUMEN

La computación es una ciencia avocada a resolver problemas de manera eficaz y eficiente en la actualidad existen diversos dispositivos táctiles, los cuales plantean un nuevo paradigma de interacción con las personas.

La Plataforma **Android** provee aplicaciones para dispositivos móviles en paquetes denominados **APK** (Application Package File) los cuales apoyan a nuestra sociedad. Sin embargo en la actualidad existen aplicaciones para las personas con discapacidad visual, pero las disponibles en **Play Store** son para que nosotros nos comuniquemos y aprendamos su forma de comunicarse de ellos pero no así para que ellos se comuniquen de forma **asíncrona** con nosotros.

La presente Tesis, es un modelo táctil traductor para **Android** en base a los **Algoritmos Genéticos, Tabú y Metodologías Híbridas** que contribuye con un sistema de escritura **Braille** y tinta, para lograr una mejor adaptación de una aplicación **Multitáctil** en la comunicación **asíncrona** entre las personas con discapacidad visual y la población en general permitiéndoles una comunicación directa de forma escrita y **fonológica** .

La utilidad de esta **APK** tiene posibles proyecciones en el ámbito de la comunicación, se realizaron diversas evaluaciones de funcionalidad y usabilidad para asegurar su correcto funcionamiento, tanto con usuarios expertos en comunicación diferencial, especialistas en personas con discapacidad visual y los usuarios finales.

Palabras Clave: Algoritmos Genéticos, Android, APK, Multitáctil, Asíncrona, Braille, Fonológica, Play Store.

ABSTRACT

Computer science is doomed to solve problems effectively and efficiently at present there are various touch devices, which pose a new paradigm of interaction with people.

The **Android** platform provides applications for mobile devices in packages called **APK** (Application Package File) which support our society. However at present there are applications for visually impaired people, but the **Play Store** are available for us to communicate and learn how they communicate them but not for them to communicate asynchronously with us.

This thesis is a model for **Android** based on **genetic algorithms**, **Tabu** and **hybrid methodologies** contributing to a writing system **Braille** and ink, to achieve a better adaptation of a multi-application asynchronous communication between people with visual disabilities and the general population allowing direct communication in writing and **phonological** form.

The usefulness of this **APK** is possible projections in the field of communication, different functionality and usability evaluations were performed to ensure correct operation, both in differential communication expert users, specialists visually impaired and end users.

Keywords: Genetic algorithms, Android, APK, Multi-touch, Asynchronous, Braille, Phonological, Play Store.

ÍNDICE

CAPÍTULO I

MARCO REFERENCIAL

1. Introducción	1
1.1. Estado del Arte.....	2
1.1.1. Sistemas Operativos móviles disponibles para teléfonos	2
1.1.2. Elección de sistema apropiado	3
1.1.3. Investigaciones y aplicaciones disponibles	3
1.1.4. Proyecto de interpretación del Braille al español	4
1.1.5. Análisis del entorno	6
1.2. Planteamiento del problema.....	7
1.3. Idea a defender.....	7
1.4. Objetivos	8
1.4.1. Objetivo General.....	8
1.4.2. Objetivos Específicos	8
1.5. Justificación	9
1.5.1. Justificación Social	9
1.5.2. Justificación Técnica	9
1.5.3. Justificación Científica	10
1.6. Metodología	10
1.7. Alcances	12

CAPÍTULO II

MARCO TEÓRICO

2. Introducción	13
2.1. Sistema Braille	15
2.2. Modelo táctil para la traducción en el Sistema Braille	17
2.3. Sistemas Operativos Móviles.....	18
2.3.1. Aplicaciones móviles.....	19
2.4. Tecnologías y herramientas utilizadas	26
2.4.1. Android.....	27
2.4.2. Java	28
2.4.3. XML	29
2.4.4. SQLite.....	30
2.4.5. Eclipse	32
2.4.6. Especificaciones e instalación de la tecnología de desarrollo	36
2.5. Metodologías ágiles para aplicaciones móviles.....	40
2.5.1. Características y requerimientos específicos del entorno móvil	41
2.6. Metodologías utilizadas	43
2.6.1. Diseño de la metodología híbrida (Hybrid Methodology Design).....	44
2.6.2. Búsqueda Tabú	46
2.7. Algoritmos Genéticos	54
2.7.1. Historia de los algoritmos genéticos.....	54
2.7.2. Definiciones.....	55

2.7.3.	Herencia.....	57
2.7.4.	Selección natural.....	59
2.7.5.	Operaciones genéticas en cadenas binarias	62
2.7.6.	Cruzamiento	63
2.7.7.	Mutación.....	65
2.7.8.	Algoritmos Genéticos y TABU con mecanismos de búsqueda local.....	69

CAPÍTULO III

MARCO APLICATIVO

3.	Introducción	76
3.1.	Diseño de la metodología híbrida	76
3.1.1.	Primera iteración - Análisis y Diseño.....	77
3.1.2.	Segunda iteración - Integración.....	81
3.1.3.	Tercera iteración - Motor de Desarrollo.....	84
3.1.4.	Cuarta iteración – Prototipado.....	95

CAPITULO IV

IDEA A DEFENDER

4.	Introducción.....	105
4.1.	Pruebas al prototipo	105
4.1.1.	Fase 1 - Planeación.....	105
4.1.2.	Fase 2 - Preparación	108
4.1.3.	Fase 3 – Desarrollo de Pruebas	110
4.1.4.	Fase 4 - Terminación de las Pruebas	123

4.1.5.	Conclusiones de resultados obtenidos	125
4.1.6.	Comprensión y aplicación	128
4.1.7.	Afirmación de la Idea a Defender.....	128

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1.	Conclusiones	129
5.2.	Recomendaciones	130
5.2.1.	Trabajos futuros.....	131
	Bibliografía.....	132

CAPÍTULO VI

ANEXOS

A - Arbol de Problemas	137
B - Alfabeto Braille.....	138
C - Conversion Braille – Binario – Decimal – Alfanumerico	139
D - Prototipo	140
E - Entrevista con el Dir. discapitados de la vista.....	142
F - Entrevista con especialista interprete de Sordos	143
G - Evaluacion de la APK (ISO/IEC 9126).....	145
H - Abecedario Sudamericano en Alfa-numeric, Braille y Simbolos.....	146
I - Vocabulario en Alfa-numeric, Braille y Simbolos.....	153
J - Numeros en Alfa-numeric, Braille y Simbolos.....	154

ÍNDICE DE FIGURAS

Figura 1 Aplicación lucrativa	4
Figura 2 Discapacitado utilizando el invento	5
Figura 3 Interfaz de “Alfabeto Braille en línea”	6
Figura 4: Configuración de una celda en Braille	15
Figura 5 Regleta y punzón para escribir manualmente en Braille.....	16
Figura 6 Maquina de Perkins para escritura en Braille	17
Figura 7: Cuota mundial de mercado de sistemas operativos moviles.....	18
Figura 8: Sistemas operativos para dispositivos moviles 2014	19
Figura 9: Nivel de evolución y consumo de sistemas operativos móviles	21
Figura 10 Arquitectura interna de Android	28
Figura 11: Vista del emulador de Android	33
Figura 12: Vista del entorno de desarrollo de Android	33
Figura 13: Vista de la plataforma Eclipse	33
Figura 14: Jerarquía que define una interfaz de usuario.....	34
Figura 15: Estructura de un proyecto Android	35
Figura 16: Estructura de la carpeta res de un proyecto Android	35
Figura 17: Descargando el JDK de Java en la pagina de Oracle	36
Figura 18: Lista de JDK dividida por sistemas operativos.....	36
Figura 19 Enlace de descarga de SDK	37
Figura 20: Ejecución de ADT.....	38
Figura 21: Android SDK Manager	39
Figura 22: Instalacion del ADT de Android.....	40
Figura 23: Ciclo de vida clasico para un proceso de desarrollo de software.....	44
Figura 24: Primera iteración en el diseño hibrido (instanciacion de analisis y diseno)	45
Figura 25: Segunda iteración en el diseño hibrido: integracion	45
Figura 26: Tercera iteración en el diseño hibrido: motor de desarrollo	45

Figura 27 Cuarta iteración en el diseño híbrido: prototipado.....	46
Figura 28: Permutacion inicial	48
Figura 29: Intercambio de las tareas 1 y 6.....	49
Figura 30: Estructura de datos Tabu.....	50
Figura 31: Iteracion 0	51
Figura 32: Iteracion 1	51
Figura 33: Iteracion 2	52
Figura 34: Iteracion 3	52
Figura 35: Iteracion 4	53
Figura 36: Iteracion 5	53
Figura 37: Cruzamiento de unico-punto	64
Figura 38: Esquema de un algoritmo genetico simple	66
Figura 39: Flujo del método Propuesto	69
Figura 40 Ejemplo de búsqueda local.....	71
Figura 41: Ejemplo de la base de datos - almacenamiento individual	73
Figura 42: Aspecto de la base de datos del metodo propuesto.....	74
Figura 43: Ciclo de vida clásico para un proceso de desarrollo de software- Traducido.....	76
Figura 44: Primera iteracion en el diseño híbrido (analisis y diseño) - traducido.....	77
Figura 45: Diagrama general del sistema	78
Figura 46 Representacion del diagrama de Procesos de la APK.....	80
Figura 47: Segunda iteracion en el diseño híbrido (integracion) - traducido.....	81
Figura 48: Diseño Navegacional - OOHDM.....	82
Figura 49 Diagrama de clases - funciones.....	83
Figura 50: Diseño de la BD	83
Figura 51: Tercera iteracion en el diseño híbrido (motor de desarrollo) - traducido	84
Figura 52: Modelo navegacional - Teclado Braille.....	86
Figura 53 Modelo navegacional - Teclado Alfanumerico.....	87
Figura 54 Modelo navegacional - Abecedario Vocabulario y Numeros	87
Figura 55 Modelo navegacional - Documentos guardados	88
Figura 56 Modelo navegacional - Manual de Usuario	88
Figura 57: Estructura de la APK.....	89

Figura 58: Cuarta iteracion en el diseño hibrido (prototipado) - traducido.....	95
Figura 59: Relación idea a defender propuesta	105
Figura 60: Reunión con Dir. Discapitados de la vista.....	122
Figura 61: Reunión con la especialista interprete de los Sordos	122
Figura 62: Aplicando el funcionamiento de la APK 1	123
Figura 63: Explicando el funcionamiento de la APK 2.....	123
Figura 64: Explicando el funcionamiento de la APK 3.....	124
Figura 65: Pruebas prácticas 1	124
Figura 66: Pruebas prácticas 2.....	124
Figura 67: Pruebas prácticas 3.....	125
Figura 68 Splash – Bienvenida.....	140
Figura 69 Menu principal	140
Figura 70 Abecedario Vocabulario y Números	140
Figura 71 Documentos guardados	140
Figura 72 Teclado Braille.....	141
Figura 73 Teclado Alfanumérico.....	141
Figura 74 Manual del usuario.....	141

ÍNDICE DE TABLAS

Tabla 1: Sistemas operativos móviles disponibles	2
Tabla 2: Metodologías utilizadas.....	11
Tabla 3: Versiones SDK.....	25
Tabla 4 Expresiones utilizadas en la genética	55
Tabla 5: Primera generación del cruzamiento	68
Tabla 6: Segunda generación del cruzamiento	68
Tabla 7: Requisitos funcionales.....	79
Tabla 8: Requisitos no funcionales.....	79
Tabla 9 Hardware de desarrollo	84
Tabla 10: Software de desarrollo.....	84
Tabla 11: Requerimientos de la APK.....	85
Tabla 12: Prueba de caja blanca, Splash	111
Tabla 13: Resultado de caja blanca, Splash	111
Tabla 14; Caso de prueba caja blanca, Splash.....	111
Tabla 15; Prueba de caja blanca, Menú principal.....	112
Tabla 16: Resultados de caja blanca, Menú principal	113
Tabla 17: Caso de prueba caja blanca, Menú principal.....	113
Tabla 18: Prueba de caja blanca, Menú Braille	114
Tabla 19: Resultados de caja blanca, Menú Braille.....	114
Tabla 20: Caso de prueba caja blanca, Menú Braille	115
Tabla 21: Prueba de caja blanca, Menú Alfa-Numérico	115
Tabla 22: Resultados de caja blanca, Menú Alfa-Numérico	116
Tabla 23: Caso de prueba caja blanca, Menú Alfa-Numérico.....	116
Tabla 24: Prueba de caja blanca, Menú VocAbeNum	117
Tabla 25: Resultados de caja blanca, Menú VocAbeNum	117
Tabla 26: Caso de prueba caja blanca, Menú VocAbeNum.....	117
Tabla 27: Prueba de caja blanca, Menú Diccionarios	118
Tabla 28: Resultados de caja blanca, Menú Diccionarios	118

Tabla 29: Caso de prueba caja blanca, Menú Diccionarios.....	118
Tabla 30: Prueba de caja blanca, Menú Diccionario Símbolos.....	119
Tabla 31: Resultados de caja blanca, Menú Diccionario Símbolos	119
Tabla 32: Caso de prueba caja blanca, Menú Diccionario Símbolos	119
Tabla 33: Prueba de caja blanca, ejemplo Image Adapter	120
Tabla 34: Resultados de caja blanca, ejemplo Image Adapter	120
Tabla 35: Caso de prueba caja blanca, ejemplo Image Adapter.....	120
Tabla 36: Equivalencias en el teclado Braille	121
Tabla 37: Escala de calificación para la APK	125
Tabla 38 Resultados de usabilidad	126
Tabla 39 Resultados de contenido	127
Tabla 40: Resultados de aplicación	128

ÍNDICE DE ECUACIONES

Ecuación 1: Función Tabú.....	48
Ecuación 2: Cadena binaria de longitud fija.....	61
Ecuación 3: Cadenas de tiempo.....	61
Ecuación 4; Seleccion dentro de un algoritmo	62
Ecuación 5: Diferentes salidas con probabilidades	62
Ecuación 6: Funcion de ejemplo	67
Ecuación 7: Probabilidad de cruzamiento	68
Ecuación 8: Valor absoluto	71
Ecuación 9: Caso N individuos - j es parte de I	75
Ecuación 10: Distancia entre individuos	75
Ecuación 11: Expresión cerrada	75

ÍNDICE DE CODIGO

Código 1: Ejemplo simple con texto y un boton	34
Código 2: Estructura de un AG	61
Código 3: Diferentes salidas con probabilidades	63
Código 4: Cruzamiento de único-punto	64
Código 5: Mutación en un individuo.....	65
Código 6: Clase Splash.....	90
Código 7 Clase Menu Principal.....	91
Código 8: Clase Teclado Braille.....	92
Código 9: Clase Teclado Alfanumerico	93
Código 10 Clase Manual Usuario.....	93
Código 11 Clase Vocabulario Abecedario y Numeros.....	94
Código 12: Clase Documentos Guardados.....	94
Código 13: Android Manifest.....	96
Código 14 Modelo Algoritmo Genetico con Busqueda Local Tabu	103
Código 15: Base de datos de la APK.....	104
Código 16: Clase que observa la Base de Datos	104

GLOSARIO

A

Algoritmos Genéticos

El algoritmo genético es una técnica de búsqueda basada en la teoría de la evolución de Darwin, que ha cobrado tremenda popularidad en todo el mundo durante los últimos años.

Android

Sistema operativo basado en el núcleo Linux, diseñado para dispositivos móviles con pantalla táctil.

APK

Application Package File - Paquete para el sistema operativo Android, este formato es una variante del formato JAR de Java.

Asíncrona

La comunicación asíncrona es la que permite la comunicación entre personas de forma no simultánea.

B

Braille

Sistema de lectura y escritura táctil pensado para personas ciegas.

F

Fonológica

Referido al sonido o a la decodificación de sonidos necesarios para la comprensión de palabras, la fonología describe el modo en que los sonidos funcionan en un nivel abstracto o mental.

H

híbrido

El término híbrido hace referencia a todo aquello que sea el resultado de la mezcla de dos o más elementos de diferente naturaleza o tipo. El híbrido es entendido entonces como algo que no es puramente ninguno de las partes que lo compuso si no que toma elementos de todas ellas para convertirse en algo nuevo. Las metodologías híbridas incorporan algunas prácticas existentes dentro de las metodologías tradicionales y ágiles, proporcionando así una gran ventaja.

I

Inherente

Procede del latín inhaerens, una conjugación del verbo inhaerere (“permanecer unido”). El concepto se utiliza para nombrar a aquello que, debido a sus condiciones naturales, resulta imposible separarlo de algo ya que está unido de una manera indivisible a eso.

Intuitivo

Es la facultad de comprender las cosas al instante, sin necesidad de realizar complejos razonamiento. El término también se utiliza para hacer referencia al resultado de intuir “En realidad no sabía que ibas a estar allí; fue pura intuición”

M

MIKY MAIA

Centro de Educación y Rehabilitación de personas con Discapacidad (Cobija-Pando-Bolivia).

Modelo

Un Modelo es un prototipo que sirve de referencia y ejemplo para todos los que diseñan y confeccionan productos de la misma naturaleza. La palabra proviene del italiano “Modello”. Según la Real Academia Española, un modelo es una representación que simboliza la perfección en todos los aspectos naturales que posee y en la forma en la que la sociedad reacciona ante ello.

Multitáctil

Diferencia de las pantallas táctiles convencionales, pueden detectar más de una pulsación al mismo tiempo.

P

Paradigma

Un vocablo que deriva del griego paradeigma - se utiliza como sinónimo de “ejemplo” o referencia como “modelo”.

Paradigmático

Teoría cuyo núcleo central se acepta sin cuestionar y que suministra la base y modelo para resolver problemas y avanzar en el conocimiento.

Play Store

Plataforma de distribución digital de aplicaciones móviles para los dispositivos con sistema operativo Android.

T

Tabu

Tabú es un término polinesio que significa “lo prohibido”. El concepto permite mencionar las conductas o acciones que están prohibidas o censuradas por un grupo humano debido a cuestiones culturales, sociales o religiosas.

Tactil

Para poder entender el significado de pantalla táctil antes es necesario proceder a determinar el origen etimológico de lo que le da forma (Táctil, por su parte procede del latín. En concreto, de “tactilis”, que puede traducirse como “relativo al tacto”. - Pantalla, por su parte tiene un origen incierto. No obstante, se considera que puede emanar del catalán “pantalla”, que es fruto de una mezcla entre “pampol” y “ventalla”)

Traductor

La **traducción** es la actividad que consiste en comprender el significado de un texto en un idioma, llamado *texto origen* o «texto de salida», para producir un texto con significado equivalente, en otro idioma, llamado *texto traducido* o «texto meta». El término traducción puede definirse como la acción de guiar de un lado para otro.

CAPÍTULO I

MARCO REFERENCIAL

El presente capítulo describe en forma general la Tesis, conteniendo los antecedentes, de la misma forma se describe de manera precisa la solución propuesta en base a los problemas identificados en el estado del arte y fruto de un análisis minucioso, de acuerdo a lo expuesto se genera la hipótesis y el objetivo planteado.

1. Introducción

El braille es un sistema de lectura y escritura táctil pensado para personas con discapacidad visual. Fue ideado por el francés Louis Braille a mediados del siglo XIX. El sistema braille consiste en un sistema universal de celdas de 6 puntos con relieves colocados en dos filas verticales contiguas de tres puntos cada una, ligeramente separados entre sí. El braille no es un idioma, sino un alfabeto; Con el braille se pueden representarse las letras, los signos de puntuación, los números, la grafía científica, los símbolos matemáticos y la música.

El sistema Braille es un modo universal en el que las personas con discapacidad visual leen y escriben, de tal forma que pueden adquirir una mayor capacidad para competir en igualdad de condiciones. A pesar del enorme desarrollo del siglo XXI y los avances en la tecnología móvil, la misma no consiguió mejorar las condiciones de vida de las personas discapacitadas, si lo comparamos con el resto de la población, existe límites, en coste y tiempo para poder plasmar sus ideas, pensamientos y difundirlos en cualquier medio escrito o impreso.

Afortunadamente, disponemos en nuestros bolsillos de equipos con una enorme potencia que hace años atrás, deseáramos tener en nuestra oficina o casa. Los teléfonos Android tienen capacidades suficientes para ejecutar aplicaciones complejas, por lo que se espera que las personas discapacitadas en los próximos años puedan disponer de un número significativo de nuevas aplicaciones que les ayuden en su vida diaria.

La presente Tesis, es un modelo táctil traductor para Android en base a los Algoritmos Genéticos, Tabú y Metodologías Híbridas que contribuye con un sistema de escritura Braille y tinta, para lograr una mejor adaptación de una aplicación Multitáctil en la comunicación asíncrona entre las personas con discapacidad visual y la población en general permitiéndoles una comunicación directa de forma escrita y fonológica.

CAPÍTULO I - MARCO REFERENCIAL

El uso de esta plataforma se debe a la enorme aceptación que está teniendo dentro de la sociedad en general como las facilidades que permite en un entorno de desarrollo, software libre y abierto.

1.1. Estado del Arte

Analizando los sistemas operativos móviles para teléfonos como antecedentes se hace un repaso a las tecnologías relacionadas con la tesis y algunas iniciativas cuya misión es mejorar la accesibilidad para personas con discapacidades visuales, para lo cual se analizará los sistemas operativos más utilizados e influyentes, investigaciones, APK disponibles, proyectos de interpretación y un análisis del entorno.

1.1.1. Sistemas Operativos móviles disponibles para teléfonos

Catalogando los distintos sistemas disponibles en el mercado se diferencian en sus características las cuales se demuestran en la *Tabla 1*.

Sistemas operativos móviles	
BlackBerry OS	Correo electrónico tiempo real, servidores RIM, cifrada y segura.
iOS	Mejor diseño, presentación y simplicidad con control de Apple ¹ .
Symbian	Filosofía Nokia impidió Symbian crecer liberando código Symbian.
Android	Google creó sistema operativo móvil propio, gratis en kernel Linux.
Windows Phone²	Sistema operativo para móviles de Microsoft ³ .
MeeGo⁴	Vio la luz como fusión de Maemo ⁵ de Nokia y Moblin ⁶ de Intel.

Tabla 1: Sistemas operativos móviles disponibles
Fuente: Elaboración propia

¹<http://www.apple.com/>

² <http://www.microsoft.com/windowsphone/es-es/default.aspx>

³<http://www.microsoft.com/es/es/default.aspx>

⁴<http://meego.com/>

⁵<http://maemo.org/>

⁶<http://moblin.org/>

1.1.2. Elección de sistema apropiado

Tras analizar los sistemas operativos móviles más importantes y las principales tecnologías de accesibilidad, se llegó a la conclusión de que la herramienta más adecuada se encuentra en la plataforma Android, por los siguientes motivos (La aplicación pueda correr en múltiples dispositivos. Una tienda APK sin restricciones, permitiendo que usuario juzgue la calidad. Precio ajustado para una publicación y accesibilidad de programar en Windows, Linux y MacOS).

1.1.3. Investigaciones y aplicaciones disponibles

- **Universidad:** Escuela Superior de Ingeniería Mecánica y Eléctrica – México
 - Área:** Ingeniero en comunicaciones y electrónica
 - Autores:** Jessica Flores Hernandez y Uziel Gallegos Espinoza
 - Tesis:** Diseño de un intérprete Braille-Español/Español-Braille utilizando la Bam Alfa-Beta.
 - Resumen:** Se diseñó e implemento un intérprete Braille-Español/Español-Braille utilizando el modelo de las memorias asociativas bidireccionales Alfa-Beta procesando las imágenes obtenidas (PDI), las memorias asociativas son el proceso de interrelacionar información con otra partiendo de conocimiento previo. El presente proyecto terminado asegura que el intérprete es una herramienta eficaz con su interfaz gráfica posicionándolo como un software amigable para cualquier usuario.

- **APK:** **BrailleApp Free** para Android (BrailleApp-Free, 2014) es una APK lucrativa, en la *Figura 1*.

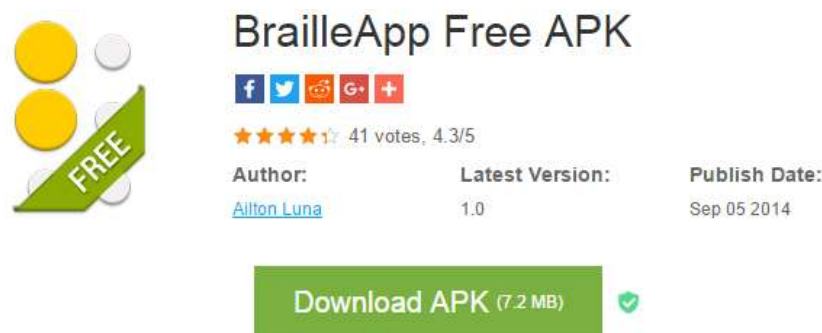


Figura 1 Aplicación lucrativa
Fuente: (BrailleApp-Free, 2014)

Versión: 1.0 for Android 4.0+ (API 14) **Publicado:** Sep. 05 2014

Ayuda en la enseñanza y el aprendizaje del Braille, disponible en GooglePlay, sus características:

- ✓ Transcripción de texto en Braille para el portugués y portugués al Braille.
- ✓ Lista de signos Braille. Muestra la Guía de referencia del sistema Braille, que contenía el Braille gráfico para el idioma portugués.
- ✓ Permite ejercitar la lectura Braille, que se muestran 10 palabras escritas en Braille y se intenta transcribirlos al portugués y viceversa.

1.1.4. Proyecto de interpretación del Braille al español

El proyecto relacionado con la interpretación del Braille al español se efectuó el 2007, Raoul Parienti de origen Francés invento el “Top Braille”, un aparato que permite traducir instantáneamente en Braille los textos impresos. El aparato es un poco más grande que el ratón de un ordenador, pesa 120 gramos, cabe en la palma de la mano o en el fondo de un bolsillo.

El aparato se desplaza por una línea de texto, una micro cámara digital escanea cada carácter, transmite las imágenes a un procesador que dirige una célula Braille bajo el índice de un usuario cuyo uso se demuestra en la *Figura 2*. Los pequeños picos de esta célula bajan o suben para componer una traducción instantánea, fuera de imprimir el texto puede escucharse a su vez por medio de un auricular. (Top Braille, 2007).



*Figura 2 Discapitado utilizando el invento
Fuente: (Top Braille, 2007)*

En la actualidad hay disponible en internet una serie de traductores online que permiten hacer una conversión de español a Braille que solamente se ocupan de traducir texto en español a Braille, efectuando comparaciones simples, los cuales son:

- **Software - Traducción a Braille Duxbury.** Este sistema es líder mundial en software para Braille, el Duxbury Braille Translator (DBT) y MetaDot son utilizados por prácticamente todos los principiantes editores del mundo Braille. Nadie soporta más idiomas que DuxburySystems – DBT apoyada en grado 1 y 2 de traducción en inglés, español, francés, alemán, portugués, árabe, malayo, sueco y otros idiomas. Existe una versión del programa para Windows, de documentos hechos en Word o código ASCII a Código Braille sin la intervención del usuario. Automáticamente agrega todos los caracteres de control necesarios, así como las convenciones utilizadas en el Braille. (Software traductor a Braille Duxbury, s.f.).
- **Alfabeto Braille en Línea.** Es una aplicación desarrollada utilizando tecnología JAVA cuya interfaz se ve en la *Figura 3*, que permite a los usuarios escribir el texto deseado y convertirlo en Braille, no usa un algoritmo muy complejo ya que solamente se basa en hacer comparaciones con una interfaz gráfica amistosa. (Software Alfabeto Braille en Linea, s.f.)

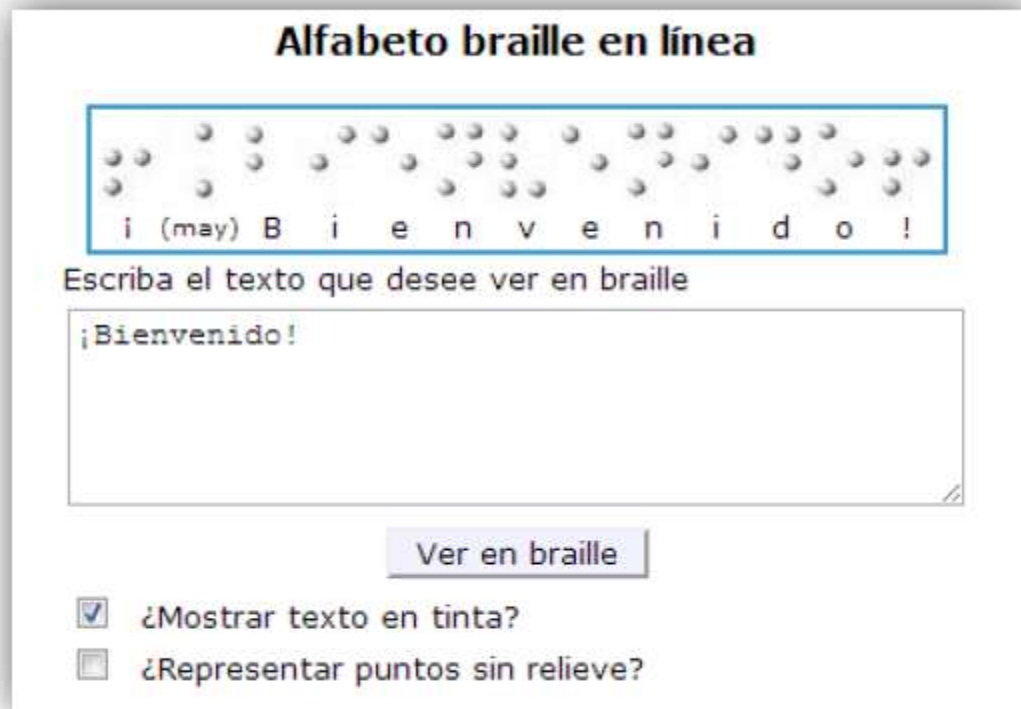


Figura 3 Interfaz de “Alfabeto Braille en línea”
FUENTE: (Software Alfabeto Braille en Línea, s.f.)

1.1.5. Análisis del entorno

Las investigaciones realizadas concernientes a este tema tienen un mayor avance tecnológico en otros países que en el nuestro. Las investigaciones y APK en línea se enfocan directamente al proceso de enseñanza aprendizaje aprendiendo sus sistemas de los discapacitados y no así que ellos puedan aprender y comunicarse con nosotros de manera rápida y directa.

La presente Tesis impulsa a la tiftotecnología con el desarrollo de una APK en una plataforma Android con Algoritmos Genéticos y Tabú para crear una forma de comunicación asíncrona aportando la educación, social y laboral. En el centro **MIKY MAIA** se observó el estado actual de los discapacitados de la vista, sus medios de comunicación, alcances y limitaciones.

CAPÍTULO I - MARCO REFERENCIAL

1.2. Planteamiento del problema

Las causas son que las personas con discapacidad visual tienen dificultades para expresar sus pensamientos y las ideas en documentos de forma escrita y compartirlos en un medio de comunicación.

En el centro de discapacitados **MIKY MAIA** se evidencio que las personas discapacitadas de la vista tienen la necesidad de plasmar sus ideas en documentos y que se puedan publicar en medios impresos y digitales, por lo cual no se fomenta la producción intelectual y se observa su necesidad de adquirir material bibliográfico de otros países al no existir en la Ciudad de Cobija librerías especializadas en Braille y materiales para su uso, lo cual implica un costo muy elevado y conflictivo para adquirir los mismos y más aun con los requisitos que se necesitan para comprar por Internet.

Los efectos son, no se fomenta la producción intelectual de las personas con discapacidad visual, la población al desconocer el Braille, omiten los textos donde se plasman sus ideas y pensamientos de forma asíncrona realizadas por las personas con discapacidad visual no se pueden expresar libremente, como se puede observar en Anexos (**A - árbol de problemas**).

A partir de lo mencionado anteriormente se llega al problema principal:

“Dificultad en la comunicación asíncrona que traduce del Sistema Braille a escritura en tinta y fonológica entre una persona con discapacidad visual y la población en general”

1.3. Idea a defender

El modelo táctil de traducción propuesto en base Algoritmos Genéticos, Tabú, Metodologías Híbridas y Android para los discapacitados de la vista, facilitará la comunicación asíncrona del Sistema Braille a Escritura en Tinta y su expresión

1.4. Objetivos

1.4.1. Objetivo General

Desarrollar un modelo táctil de traducción para Android en una aplicación APK en base Algoritmos Genéticos, Tabú y Metodologías Híbridas, como un medio de comunicación asíncrona, para el Sistema Braille a Escritura en Tinta y su Expresión Fonológica.

1.4.2. Objetivos Específicos

- Diseñar y construir una interfaz gráfica intuitiva con el Sistema Braille para convertirlos en textos alfanuméricos y expresión fonológica.
- Administrar archivos de texto generados y almacenados por el sistema de manera accesible para personas con discapacidad Visual.
- Diseñar y construir un control para administrar el mando de lectura en Braille.
- Desarrollar el prototipo del modelo táctil de traducción para Android en una aplicación APK en base Algoritmos Genéticos, Tabú y Metodologías Híbridas.
- Evaluar el prototipo construido, a través de técnicas de prueba de caja blanca y caja negra.

1.5. Justificación

1.5.1. Justificación Social

El **I.B.C.**⁷ en Cobija carece de funcionarios y de fondos por que los mismos son incomprendidos por parte de la Honorable Alcaldía Municipal de Cobija y la Gobernación de Pando, esto produce que los mismos sientan una separación con la sociedad.

La Tesis se justifica socialmente en beneficio a la población en general, pero particularmente a los discapacitados de la vista, la APK con algoritmos genéticos y Tabú, coadyuvan en la traducción del sistema Braille al español favoreciendo en gran magnitud en transmitir y difundir sus pensamientos y las ideas que serán plasmadas en un texto alfanumérico y poder transmitirlo en forma digital.

La APK puede adaptarse a diferentes entornos, facilitando su uso y aceptación de las personas brindando ayuda a los discapacitados de la vista para integrarse a la sociedad sin sufrir ningún tipo de rechazo.

1.5.2. Justificación Técnica

La Ciencia y Tecnología cuenta con muchas herramientas innovadoras y brillantes que motivan a la escritura de las personas con discapacidad Visual. Tal es el caso que el modelo propuesto está elaborado en función a técnicas del Sistema Braille y técnicas informáticas que combinadas facilitarán la comunicación asíncrona a Escritura en Tinta y su Expresión Fonológica.

⁷I.B.C. - Instituto Boliviano de la Ceguera

1.5.3. Justificación Científica

Actualmente no existen muchos modelos del Sistema Braille particularmente en Android para personas discapacitadas de la vista y sus formas de comunicación, requieren un análisis especial, su pensamiento y difusión lo plasman de forma distinta (Tellez, 2015).

La presente tesis aporta al *desarrollar un modelo táctil de traducción para Android en una aplicación APK con Algoritmos Genéticos, Tabú y Metodologías Híbridas, como un medio de comunicación asíncrona, del Sistema Braille a Escritura en Tinta y su Expresión Fonológica*, al avance constante de la Ciencia y Tecnología.

1.6. Metodología

Para la evaluación de la hipótesis, se utiliza la metodología de investigación científica propuesta por Mario Bunge (Bunge, 1989):

- Observación
- Planteamiento de la hipótesis
- Diseño de la aplicación
- Prueba de hipótesis
- Conclusiones

Para el proceso de desarrollo de la aplicación (APK) se utilizara el proceso y método de la ingeniería de software orientado a objetos, tomando al Lenguaje de Modelado Unificado como herramienta de modelado para garantizar el proceso de aseguramiento de la calidad del software y centrándose en un estudio descriptivo, seleccionando una serie de conceptos y explicación de sus componentes.

Usualmente en el desarrollo de algún software, del tipo que fuera se utiliza una sola metodología, pero para desarrollar esta Tesis se caracteriza particularmente por involucran Algoritmos Genéticos, metodología de búsqueda local Tabú, para el diseño el uso de metodologías Híbridas todo esto para realizar una aplicación APK que sea más eficiente y eficaz. Ambas metodologías se las describe en la *Tabla 3*.

METODOLOGIAS UTILIZADAS

Diseño de la metodología híbrida(Hybrid Methodology Design)		
Utiliza el modelo iterativo incremental para el proceso de desarrollo y así lograr la rápida entrega de software. Se alinean con las necesidades de desarrollo de aplicaciones móviles (Harishankar_S_B, 2016)	Primera iteración	Se divide la fase de análisis con la intención de mitigar riesgos de desarrollo La implementación y las pruebas sin embargo se fusionan introduciendo conceptos de desarrollo orientado a pruebas (Test-Driven Development, TDD). Aparece además una fase de comercialización, hacia el desarrollo de producto que se imponen en el escenario del desarrollo de aplicaciones para plataformas móviles.
	Segunda Iteración	Realiza una integración de ciertas partes de los modelos NPD (New Product Development), añadiendo la generación de ideas en el inicio del ciclo y una prueba de mercado antes de lanzar la fase de comercialización.
	Tercera Iteración	Integra directamente el “motor de desarrollo” de los métodos de desarrollo adaptativo (ASD)
	Cuarta iteración	Se añaden elementos de prototipado; se refina, además, la fase de iniciación del proyecto, sobre la base del mismo elemento de los procesos adaptativos.
Metodología de búsqueda Tabú		
La búsqueda Tabú o TS por sus siglas en ingles tienen antecedentes en métodos diseñados para cruzar cotas de facilidad u optimizad. Una característica distinta de este procedimiento es el uso de memoria adaptativa y estrategias especiales de resolución de problemas.	Memoria de corto plazo	Paso 1. selección de la solución inicial, debe de ser una solución factible
	Memoria a mediano plazo	Paso 2. Elección del entorno o vecindario y generación de una nueva solución.
	Memoria a largo plazo	Paso 3. Evaluación de la función objetivo
		Paso 4. Actualización de la mejor solución
		Paso 5. Criterio de parada

*Tabla 2: Metodologías utilizadas
Fuente: Elaboración Propia*

1.7. Alcances

La APK posee los siguientes alcances:

- Diseño para la comodidad del usuario, adaptándose a sus necesidades y rutinas en el sistema Braille.
- Disponibilidad de almacenamiento individual para cada persona con discapacidad visual para crear y almacenar textos alfa-numéricos
- Subsistema transcriptor, en este módulo el discapacitado visual podrá escribir en Braille y el dispositivo móvil traducirá al español.
- Subsistema de lectura, el dispositivo interactúa de manera directa y el mismo tiene la capacidad de leer los textos generados y guardados por el usuario.
- Subsistema de escritura alfa-numérica, escribe los documentos generados en Braille, también cuando el usuario quiere conocer lo que ha escrito el dispositivo tiene la facultad de hacer conocer el avance de la transcripción.

CAPÍTULO II

MARCO TEÓRICO

Este capítulo presenta las bases teóricas referentes a la investigación como ser: distintos tipos de sistemas operativos para teléfonos inteligentes, sistema Braille, características y versiones de Android, adaptación de los mismos utilizando los Algoritmos Genéticos, Tabú y metodologías Híbridas.

2. Introducción

Según la Organización mundial de la Salud (**OMS**) en el mundo existe aproximadamente 285 millones de personas con discapacidad visual, de las cuales 39 millones son ciegas y 246 millones presentan baja visión. Aproximadamente un 90% de discapacidad visual se concentra en los países de ingresos bajos. (OMS, 2014)

En Bolivia, el Instituto Nacional de Estadística (**INE**) comunico que existe 160 mil personas tienen discapacidad visual en el Estado Plurinacional de Bolivia, de los mismos 84 mil son mujeres y el resto son varones. Santa Cruz, La Paz y Cochabamba son los departamentos que acogen a las personas con discapacidad. (INE, 2014)

Según información proporcionada por parte del **INE** se notificó que el 2,78% de los discapacitados de Bolivia están en Pando. Efectuando el conteo se observa que son 4.448 personas con discapacidad visual en el departamento de **Pando** siendo esta una cifra alarmante.

Bolivia como en otros países del mundo, implanto programas de integración e inclusión educativa con la finalidad de contribuir con el proceso educativo y social de las personas con discapacidad visual creando una institución educativa, con la intención de impartir una educación más homogénea.

Con esta finalidad se creó la escuela Cobija (Centro Ester Campos), pero es ficticio y tergiversado por el SEDUCA por que no funciona como centro, los profesores que llegan a impartir clases son egresados de la normal y no tiene conocimientos explícitos acerca de la educación especial que necesitan los discapacitados y termina resultando que el alumno enseña al profesor. (Tellez, 2015).

CAPÍTULO II - MARCO TEÓRICO

Las personas discapacitadas de la vista en Cobija para integrarse comienzan de forma intuitiva y por instinto a utilizar herramientas tecnológicas que les permitan efectuar de manera más sencilla a sus actividades diarias. Surgieron diferentes herramientas, como los lectores de pantalla lo cuales ayudo de forma parcial. A esto se sumó el aporte de dispositivos móviles, etc. (Tellez, 2015).

Los discapacitados de la vista presentan ciertas necesidades especiales en el ámbito educativo, social, cultural, comunicación y literario para el desarrollo de sus habilidades en su libertad de comunicación y expresión.

Las Tecnologías de la Información y comunicación (TIC) inducen un impacto en todos los ámbitos. Se planteó el cómo utilizar esta herramienta y surgió la idea de un sistema que ayude a los discapacitados de la vista permitiéndoles realizar escritura en tinta y **fonológica**⁸.

La característica principal del capítulo propuesto es; *“La justificación teórica de investigación en la propuesta de un sistema móvil de comunicación en dispositivos Android para la traducción del sistema Braille a texto Alfanumérico en español y su expresión fonológica en base a los Algoritmos Genéticos, Tabú y metodologías híbridas”*. Se requiere destacar también, el cambio paradigmático producido y su repercusión de la APK, pero es con la irrupción de los teléfonos inteligentes de uso común a bajo costo cuando se maximiza su interés y uso como herramienta frecuente y continua.

El capítulo II se refiere al respaldo teórico de la investigación, donde se menciona el Sistema Braille, su uso frecuente y continuo por parte de las personas con discapacidad visual total o parcial y como han ido utilizando las Tecnologías de Información y Comunicación (TIC) para mejorar su entorno.

⁸ Fonología procede de la raíz griega Phonos que significa “Sonido”

CAPÍTULO II - MARCO TEÓRICO

2.1. Sistema Braille

El Sistema Braille es un sistema de lectura y escritura táctil, el cual presenta un conjunto de características específicas que hoy en día lo han hecho posicionarse como un método universal que utilizan personas invidentes para poder comunicarse. (UNESCO, 1954)

El Sistema Braille se expresa en la **Figura 4**, mediante la utilización de puntos de relieve dispuestos ordenadamente en grupos de dos columnas con tres filas cada una, a las cuales se les llama celdas. Para facilitar la descripción de cada uno de los signos dispuestos en la celda, enumerando los puntos convencionalmente de la siguiente manera: la columna izquierda se enumeran 1-2-3, de arriba abajo, y los puntos 4-5-6 corresponden la derecha. La letra “A” se representa con el punto 1; “B” es la agrupación de los puntos 1-2, “C” los puntos 1-4, y así sucesivamente. (Saad, 1998). Las 10 primeras letras están formadas por los cuatro puntos de arriba, las diez siguientes comprenden las diez primeras repetidas agregando el punto 3, hasta llegar a formar el alfabeto Braille. (Ver Anexos B)

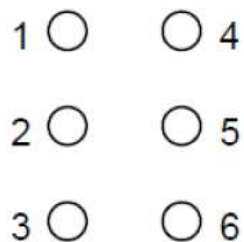


Figura 4: Configuración de una celda en Braille
Fuente: (UNESCO, 1954)

El tacto es un factor sumamente importante para determinar la amplitud del uso del Braille. Los lectores se dividen, según sus capacidades, en grupos muy determinados, como aquellos que nacieron con discapacidad visual en donde el Braille viene a convertirse en algo totalmente natural para ellos, a diferencia de los que perdieron la vista en la edad adulta, los cuales tienen que pasar de una lectura visual a una táctil. Estos últimos son generalmente lectores lentos, siendo los que enfrentan mayores dificultades para dominar el Braille. (Arredondo, 2001).

CAPÍTULO II - MARCO TEÓRICO

La lectura se realiza deslizando las yemas de los dedos de izquierda a derecha a lo largo de las líneas que constituyen el texto. La escritura puede realizarse de dos maneras, de forma manual o utilizando una máquina de escribir llamada “Máquina de Perkins”. (UNESCO, 1954).

La escritura manual se realiza introduciendo la hoja de papel en un bastidor cuyos lados verticales presentan unos orificios en los cuales se inserta la “pauta”, este instrumento se ve en la **Figura 5** que consiste en una regleta en la que aparecen perforadas líneas con orificios rectangulares que corresponden a las “celdas”. La persona escribe presionando con un punzón en las posiciones correspondientes, este procedimiento exige que la escritura se realice de derecha a izquierda y que cada signo deba escribirse rotando 180 grados respecto a su posición cuando se lee. Es decir, los signos aparecen “en espejo” en la lectura respecto a su posición en la escritura. (Huertas, 1988).



*Figura 5 Regleta y punzón para escribir manualmente en Braille
Fuente: (Huertas, 1988)*

La máquina de Perkins (**Figura 6**) es una máquina de escribir que dispone de una tecla para cada uno de los seis puntos que construyen cada “celda” y otra tecla que actúa como espaciador. Los puntos aparecen de abajo hacia arriba en el papel sobre el que se escribe, pero la persona para poder palparlos precisa desplazar su mano desde el teclado hacia la parte posterior de la máquina. Cada celda se escribe presionando simultáneamente las teclas correspondientes al signo que se quiere escribir. (Huertas, 1988) (Arredondo, 2001).



Figura 6 Maquina de Perkins para escritura en Braille

Fuente: (Arredondo, 2001)

El procedimiento de impresión en Braille resulta costoso, encargándose casi totalmente de su ejecución organizaciones privadas, a menudo con ayuda de subsidios del Gobierno (países de primer mundo). Los textos Braille se estereotipan sobre planchas de un metal blando, a mano o por medio de máquinas. Estas planchas se colocan sobre una prensa plana o rotativa que repuja los puntos en un papel grueso y fuerte, generalmente humedecido, para facilitar la impresión de los puntos y evitar que se perfora el papel. (UNESCO, 1954)

El propósito del Braille consiste en ofrecer una forma de lectura y escritura que sirva a las personas discapacitadas de la vista en su educación, trabajo, su correspondencia y goce literario. El objeto del sistema planteado consiste en ofrecer a las personas invidentes el medio más fácil de escritura para aprender, leer y comunicarse con otras lenguas además de la propia.

Por si solo el sistema Braille deja de ser una herramienta eficaz para los invidentes. La tiftotecnología toma como base el sistema Braille para que los invidentes puedan hacer uso de diversas tecnologías entre las que destaca la computadora. (ONCE UAB, 2010)

2.2. Modelo táctil para la traducción en el Sistema Braille

Para que la aplicación de un sistema operativo móvil sea revolucionario tiene que ser universalmente fácil de utilizar, de ahí que se utiliza la pantalla completa, este modelo de aplicación tiene una riqueza visual.

Utiliza una cantidad enorme de información pero que no se muestra al usuario, facilitando su manipulación continua. Utilizando un modelo táctil para la traducción reduce de manera considerable el tiempo de escritura y traducción

CAPÍTULO II - MARCO TEÓRICO

2.3. Sistemas Operativos Móviles

Partiendo de la definición de **Sistema Operativo**: “Capa compleja entre el hardware y el usuario, concebible también como una máquina virtual, que facilita al usuario o al programador las herramientas e interfaces adecuadas para realizar sus tareas informáticas, abstrayéndole de los complicados procesos necesarios para llevarlas a cabo”. (Arturo Baz Alonso, 2013).

Por ende, un **Sistema Operativo Móvil** controla un dispositivo móvil. Sin embargo, estos son mucho más simples, hay más variedad con respecto al ordenador, y el uso de uno u otro determinará la funcionalidad y la forma de interactuar con el usuario.

La popularidad de los Smartphone aumenta día tras día, y con ella, la importancia de los **Sistemas Operativos** con los que funcionan. Actualmente, según estudios realizados por **Statcounter**, la cuota de mercado de sistemas operativos móviles en el año 2011 se divide, lo cual se menciona en la **Figura 7**.

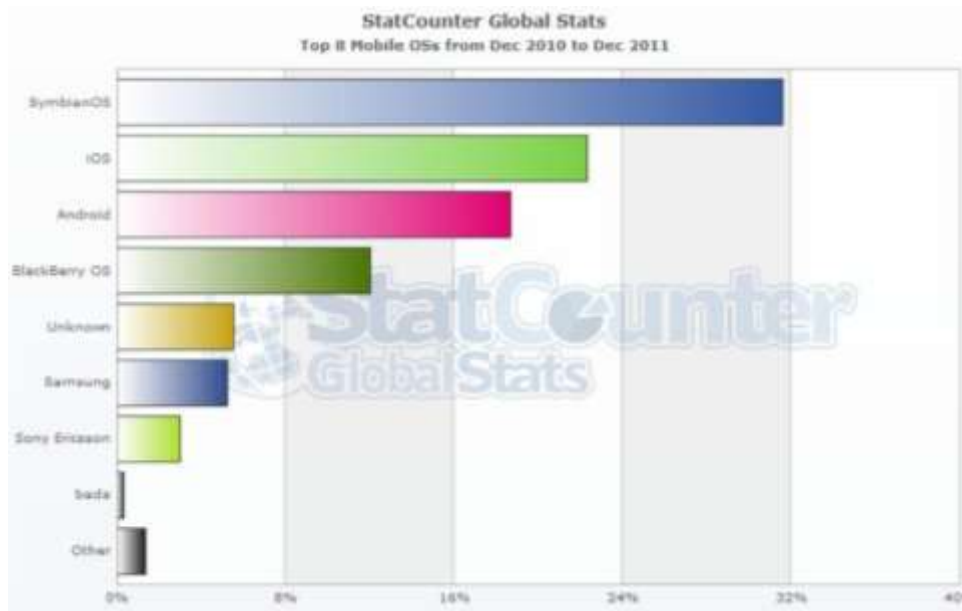


Figura 7: Cuota mundial de mercado de sistemas operativos móviles
Fuente: (Top Braille, 2007) (StatCounter GlobalStats, 2013)

2.3.1. Aplicaciones móviles

El mercado de aplicaciones móviles ha experimentado una rápida expansión durante los últimos 10 años, las plataformas móviles siguen mejorando su desempeño, y la necesidad de los usuarios de una amplia variedad de aplicaciones móviles va en aumento.

El desarrollo de software para las plataformas móviles viene con características únicas y restricciones que se aplican a la mayoría de las etapas del ciclo de vida. Las características distintivas más importantes son identificadas en: un alto nivel de competitividad, cortó tiempo de entrega, movilidad, portabilidad, capacidades específicas y constantemente cambiantes de las terminales, sistemas operativos diferentes e incompatibles, entre otras.

Los sistemas operativos con mayor influencia en el mercado y con un número importante de desarrolladores interesados en participar de las “App Stores”, según un estudio realizado y presentado, muestra según **Figura 8** que Android ha encabezado el interés de los desarrolladores con un 77% junto con iOS que posee el 66%, mientras que BlackBerry posee un 34% frente a un 37% de Windows Phone.

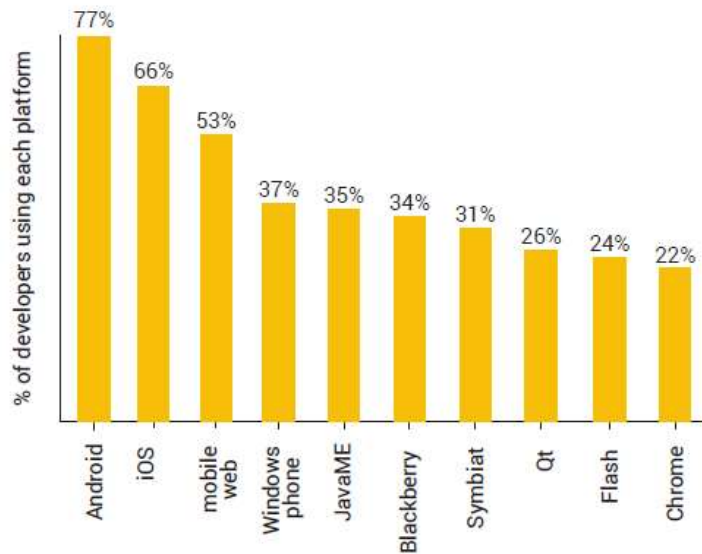


Figura 8: Sistemas operativos para dispositivos móviles 2014

Fuente: (Gartner, “Gartner says android to become no. 2 worldwide mobile operating system, 2014)

En los siguientes apartados los sistemas operativos móviles más usados, sus características y diferencias particulares.

Sistema Operativo BlackBerry OS

El sistema operativo BlackBerry OS, nace de la mano de la empresa canadiense RIM (Research In Motion)⁹ y comparte nombre con su línea de teléfonos inteligentes lanzada al mercado en 1999. Las principales características de este sistema operativo se centran en el mercado de los negocios. Como ejemplo podemos citar que ha sido una de las primeras plataformas en proveer correo electrónico en tiempo real (R. I. M. Limited, “Blackberry service real-time of email messages, 2015).

Con la llegada de otros sistemas operativos móviles, BlackBerry OS comienza a perder cuota de mercado y apuesta por mejorar sus prestaciones añadiendo mejores capacidades multimedia, soporte para MIDP (perfiles informativos del terminal), pantallas táctiles y el resto de elementos que, en general, podemos encontrar en sus competidores.

En la actualidad una de las características que hacen a este sistema operativo único reside en que los servicios de datos pasan por los servidores de RIM, permitiendo así la compresión de la información (R. I. M. Limited, “About compression, 2015) que, además, viaja cifrada por la red, obteniendo velocidad y seguridad. Esto ha supuesto problemas con algunos gobiernos ya que una empresa extranjera sería la única en controlar parte de las comunicaciones nacionales (U. T. R. Authority, “Tra announces the suspension of blackberry messenger, blackberry, 2015).

Como plataforma de desarrollo se encuentra lejos de Android e IOS. Prueba de ello es la escasa cantidad de aplicaciones disponibles.

⁹<http://www.rim.com/>

Sistema operativo iOS

En junio de 2007 sale al mercado el teléfono que cambia la historia de los móviles, se trata del iPhone con su sistema operativo iOS4 (Macworld, “Apple unveils iphone, 2014). IOS se adelanta varios años a su competencia, tanto en diseño como en prestaciones. Su interfaz simple y su capacidad multi táctil han servido como base a imitar para la mayoría de los demás sistemas operativos. Uno de los puntos flacos de su plataforma es el control que Apple¹⁰ mantiene sobre todos los aspectos relacionados con este producto. Este hecho no parece ser problema para los consumidores y, por tanto, mantiene a los desarrolladores interesados.

Entre las demás desventajas podemos citar que no se puede publicar una aplicación que sustituya a parte del sistema de Apple (si bien el navegador Opera ha encontrado la manera de acceder a App Store (Opera, “Opera mini app approved for the app store.”, 2014)), la falta de soporte a la tecnología Flash¹¹, la necesidad de programar desde un Mac para poder publicar aplicaciones o que no podamos programar para nuestros propios terminales sin pagar a Apple la cuota como desarrollador, aunque no queramos publicar nuestra aplicación.

A pesar de los puntos anteriores, el interés en el sistema operativo se mantiene, ya que iOS no solo se encuentra en el iPhone sino en iPod Touch e iPad, por lo que muchas aplicaciones pueden ser reutilizadas ahorrando tiempo de desarrollo a los programadores en la **Figura 9**.

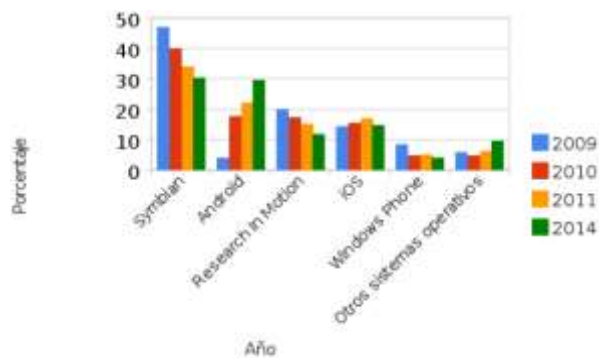


Figura 9: Nivel de evolución y consumo de sistemas operativos móviles
Fuente: (Gartner, “Gartner says android to become no. 2 worldwide mobile operating system, 2014)

¹⁰<http://www.apple.com/>

¹¹<http://www.adobe.com/products/flashplayer/>

Sistema operativo Symbian

En sus orígenes, Symbian¹² es controlado por varias empresas, entre todas intentan cooperar para avanzar rápidamente hacia el entonces incipiente mundo de los teléfonos inteligentes, pero con el paso de los años es la compañía sueca Nokia¹³ la principal encargada del desarrollo del sistema operativo.

A pesar de su despegue y años de mantenimiento, la filosofía de Nokia fue la que impidió a Symbian seguir creciendo y consolidarse como un sistema operativo móvil dominante. Las actualizaciones de sistema tardan en trasladarse a los usuarios, incluso a los poseedores de teléfonos de gama alta, por lo que poco a poco los más interesados en sacar partido a su teléfono han ido migrando a otras opciones.

El éxito de ventas de iPhone y el rápido desarrollo de la plataforma Android ha hecho que Nokia libere el código de Symbian para tratar de hacerlo avanzar más rápido gracias a la comunidad.

¹² <http://www.symbian.org/>

¹³ <http://www.nokia.es/>

Sistema operativo Android

Hace algunos años, Google decidió que debía expandir su negocio hacia los móviles, así que su mejor estrategia fue crear un sistema operativo móvil propio, gratis y con varios de los más grandes fabricantes de celulares como respaldo, de esta manera nace Android. (Vilchez, 2009)

Android¹⁴ es el último gran actor que ha llegado al mercado, cosechando un éxito comparable al de iPhone. Aunque se suele relacionar al mismo en exclusiva con Google¹⁵, el desarrollo es promovido por la OHA (Open Handset Alliance)¹⁶, compuesta por empresas de la talla de HTC, Dell, Intel, Motorola, Qualcomm, Texas Instruments, Samsung, LG, T-Mobile, Nvidia o WindRiverSystems (O. H. Alliance, “Open handset alliance members, 2015), estando disponible gran parte del código bajo licencia Apache v2 (T. A. S. Foundation, “Opera mini approved for app store, 2013).

Con una interfaz de programación Java, diseñado para ser utilizado en dispositivos móviles como teléfonos inteligentes, tabletas, Google TV y otros. Desarrollado por la Open Handset Alliance la cual es liderada por Google. (O.S.L., 2011)

Android permite programar aplicaciones en una variación de Java llamada Dalvik. Proporciona todas las interfaces necesarias para desarrollar aplicaciones que accedan a las funciones del teléfono (GPS, llamadas, sms, agenda, entre otras.) en Java.

Una de las ventajas más claras de esta plataforma respecto a su competencia es la facilidad para desarrollar aplicaciones, existiendo versiones del SDK para los tres sistemas operativos dominantes en el mercado de los ordenadores personales: Windows, MacOS y GNU/Linux. El hecho de que se puede desarrollar usando nuestros propios terminales sin “pagar cuota de desarrollo” ayuda a acercar a los programadores con poca experiencia a este sistema. El pago se realiza así que queremos publicar en la tienda.

¹⁴ <http://www.android.com/>

¹⁵ <http://www.google.com/>

¹⁶ <http://www.openhandsetalliance.com/>

CAPÍTULO II - MARCO TEÓRICO

Si bien Android ofrece una cantidad de herramientas muy útiles como una API potente, intercomunicación de aplicaciones, multitarea y posible sustitución de funcionalidad original del sistema operativo por una personalizada, cuenta con algunos problemas inherentes a su categoría de software libre:

- Fragmentación
- Actualizaciones por parte de los fabricantes
- Personalización

A pesar de lo anterior, la comunidad de desarrollo de Android es enorme y, aprovechando las ventajas del código libre, hay modificaciones personalizadas del sistema que permiten, por ejemplo, tener funcionalidades de las últimas generaciones de Android en teléfonos que no verán actualizado su sistema operativo.

Cada *Versión del sistema operativo Android* es desarrollada bajo un nombre en código de un elemento relacionado con postres, los nombres están en orden alfabético los cuales se puede observar en la *Tabla 4* (Rodríguez, 2012)

La última versión de *Android Marshmallow 6.0* fue presentado recientemente en Octubre del 2015 por sorpresa y mantiene así la tradición de utilizar golosinas para acompañar las distintas versiones de Android: Lollipop (5.0), KitKat (4.4), Jellybean (4.1) o Ice Cream Sandwich (4.0). Se diferencia por las siguientes características:

- Los permisos por aplicación, mejoras en la batería.
- Soporte para la huella dactilar en otras aplicaciones como Google Play. Google Pay.
- Respaldo en la nube con Google Drive.
- Mejoras de administración de memoria.
- Mejoras para el ahorro de batería.
- Google Now integrado a través de todo el sistema operativo con la función ‘Now on Tap’.

Versiones históricas y disponibles de SPK

Versión	Descripción
1.0	Liberado el 23 de septiembre de 2008
1.1	Liberado el 9 de febrero de 2009
1.5 (Cupcake) 	Liberado el 15 de septiembre de 2009. Basado en el kernel de Linux 2.6.27. Con una interfaz sencilla y atractiva, GPS, capacidad de grabar y reproducir videos, entre otras.
1.6 (Donut) 	Liberado el 15 de septiembre de 2009. Basado en el kernel de Linux 2.6.29 Esta actualización se incluyó novedades como la 'Quick Search Box', control de batería, conexión a VPN, entre otras.
2.0 / 2.1 (Eclair) 	Liberado el 26 de octubre de 2009. Basado en el kernel de Linux 2.6.29 En actualización se incluyó un rediseño de la interfaz de usuario, soporte para HTML5, Bluetooth 2.1, soporte para Facebook, entre otras. El SDK 2.0.1 fue liberado el 3 de diciembre de 2009. El SDK 2.1 fue liberado el 12 de enero de 2010.
2.2 (Froyo) 	Liberado el 20 de mayo de 2010. Basado en el kernel de Linux 2.6.32 En esta actualización se incluyó: una optimización general del sistema Android, que mejoraba su rendimiento y memoria, soporte para Adobe Flash entre otras.
2.3 (Gingerbread) 	Liberado el 6 de diciembre de 2010. Basado en el kernel de Linux 2.6.35.7 En esta actualización se incluyó: nuevos efectos, soporte para NFC, mejora en la entrada de datos, audio y gráficos para juegos, etc.
3.0 / 3.1 / 3.2 (Honeycomb) 	Liberado el 22 de febrero del 2011. En esta actualización se incluyó: soporte para tablets, escritorio 3D con widgets rediseñados, Google Talk, entre otras.
4.0 (Ice Cream Sandwich) 	En esta actualización se incluyó: versión que unifica el uso en cualquier dispositivo, tanto en teléfonos, tablets, televisores, netbooks, etc.
4.1 (Jelly Bean) 	En esta actualización se incluyó: mejora de la fluidez y de la estabilidad gracias al proyecto "Project Butter", ajuste automático de widgets cuando se añaden al escritorio, cambiando su tamaño y lugar para permitir que los nuevos elementos se puedan colocar nuevas lenguas no occidentales, fin al soporte de Flash Player para Android a partir de esta versión.
4.2 Jelly Bean_mr1	Liberado el 9 de octubre del 2012. En esta actualización se incluye: soporte de rotación de la pantalla principal, arreglo de fallos y mejoras en rendimiento, notificaciones expansión/contracción con un dedo.

Tabla 3: Versiones SDK
Fuente: (Vilchez, 2009)

Todas las características del sistema operativo Android tienden a mayor incremento según las versiones de los mismos. Fuera de estas características únicas de la versión 6.0, sigue conservando características indispensables de las anteriores versiones. (Vilchez, 2009).

CAPÍTULO II - MARCO TEÓRICO

Android “es una pila de software para dispositivos móviles que incluye un sistema operativo, middleware y aplicaciones clave.” (*Android Developers, Feb 2012*). Sus principales características son: (Android Developers “What is Android?”, 2014).

- Navegador integrado
- Gráficos optimizados
- SQLite
- Bluetooth, EDGE, 3G y WiFi.
- Cámara, GPS, brújula y acelerómetro.

Sistema operativo Windows Phone

Windows Phone¹⁷ ha sido tan solo uno de los nombres que ha tenido el sistema operativo para móviles de Microsoft¹⁸, antes fue conocido como PocketPC y Windows Mobile. A comienzos de la década se disputaba el título de sistema operativo más popular con Symbian. Sin embargo ambos líderes del mercado han visto como con la llegada de nuevos dispositivos con infinidad de opciones y atractivos diseños han mermado su cuota de mercado (A. Technica, “Open handset alliance members.”, 2012).

Sistema operativo MeeGo

Meego¹⁹ es un producto como consecuencia de la fusión de otros dos, Maemo²⁰ de Nokia y Moblin²¹ de Intel. Estos dos sistemas se caracterizaban por un enfoque distinto al de los sistemas mayoritarios y, con su unión, se han visto reforzadas sus posibilidades de conseguir una buena cuota de mercado.

2.4. Tecnologías y herramientas utilizadas

Se pretende explicar cómo se han utilizado las tecnologías y las herramientas de programación, en particular las usadas para crear una aplicación en Android de la que trata esta tesis.

¹⁷ <http://www.microsoft.com/windowsphone/es-es/default.aspx>

¹⁸ <http://www.microsoft.com/es/es/default.aspx>

¹⁹ <http://meego.com/>

²⁰ <http://maemo.org/>

²¹ <http://moblin.org/>

2.4.1. Android

El desarrollo de aplicaciones para Android se hace con un subconjunto de la API de Java²² y algunos añadidos extra, lo que ha propiciado una demanda por parte de Oracle²³, tras la compra de Sun Microsystems (S. Microsystems, “Oracle to buy sun.”, 2012). Además de Java se puede utilizar el lenguaje C de manera nativa para acceso a algunas partes del hardware.

2.4.1.1. Arquitectura

La arquitectura de **Android**, está formada por 5 componentes: (Android Developers “What is Android?”, 2014) lo cual se puede observar en la **Figura 10**.

- **Applications**: conjunto de aplicaciones básicas creadas usando el lenguaje Java.
- **Application Framework**: Al ser una plataforma de código abierto, los desarrolladores tienen acceso total al código fuente de las aplicaciones, ofreciendo la capacidad de crear aplicaciones muy ricas e innovadoras. Además se evita que se generen grandes cantidades de componentes de aplicaciones distintos que responden a la misma acción.
- **Libraries**: Android incluye un conjunto de librerías C/C++ utilizadas por los diversos componentes del sistema Android.
- **Android Runtime**: Android incorpora bibliotecas que proporciona la mayor parte de la funcionalidad disponible en las bibliotecas base del lenguaje de programación Java.
- **Linux Kernel**: Android se basa en la versión 2.6 de Linux para los servicios como seguridad, gestión de memoria, gestión de procesos, pila de red, y el modelo de controlador.

²²<http://www.java.com/es/>

²³<http://www.oracle.com/es/index.html>



Figura 10 Arquitectura interna de Android

Fuente: (Android Developers “What is Android?”, 2014).

2.4.2. Java

Las aplicaciones de **Android** se han desarrollado utilizando el lenguaje **Java**.

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems (propiedad de Oracle). Este lenguaje fue desarrollado siguiendo la sintaxis de C y C++ pero desechando rasgos que inducen a errores como pueden ser los punteros, la herencia múltiple o la asignación de memoria y además, a diferencia de C++, fue creado desde el principio para ser completamente orientado a objetos. Todo en **Java** es un objeto y reside en alguna clase. (openBoxer., 2013).

El lenguaje **Java** se creó con cinco objetivos principales: (openBoxer., 2013)

- Debería usar la metodología de la programación orientada a objetos.
- Debería permitir la ejecución de un mismo programa en múltiples sistemas operativos.
- Debería incluir por defecto soporte para trabajo en red.
- Debería diseñarse para ejecutar código en sistemas remotos de forma segura.
- Debería ser fácil de usar y tomar lo mejor de lenguajes orientados a objetos, como C++.

CAPÍTULO II - MARCO TEÓRICO

Java es fácil de aprender por una gran variedad de razones. Existen infinidad de recursos para ayudar a aprender el lenguaje, incluyendo sitios web, tutoriales o libros. Además, **Java** es uno de los lenguajes de programación más enseñado y usado en el planeta. Se utiliza para muchos tipos diferentes de proyectos de programación, sin importar su escala. Por esto, sus pilares básicos son: (openBoxer., 2013)

2.4.2.1. Orientado a objetos (OO)

Se refiere a un método de programación y al diseño del lenguaje. Es una forma de programar más cercana a una interpretación de cómo lo haríamos en la vida real. La idea es diseñar el software de manera que los tipos de datos estén asociados a sus operaciones. Así pues, los datos y las operaciones se combinan para formar los objetos.

Un objeto será una entidad que contiene unas propiedades (datos) y unas funciones (el código) que es capaz de realizar. El principio es separar lo que cambia de lo que permanece inalterable para así ofrecer una base más estable para el diseño. El principal objetivo es facilitar la gestión y el manejo de grandes proyectos, para así mejorar la calidad y reducir los proyectos fallidos.

2.4.2.2. Independencia de la plataforma

Cualquier programa en el lenguaje **Java** pueden ejecutarse en cualquier tipo de hardware. Para esto, el código fuente escrito y se compila creando un archivo llamado “bytecode” que son interpretados por una máquina virtual Java (**JVM**), un programa escrito en código nativo de la plataforma destino, que interpreta y ejecuta el código.

2.4.3. XML

XML (eXtensible Markup Language) es un metalenguaje cuya labor es definir unas reglas que deberá cumplir un lenguaje de marcas, es decir, ordena, estructura y describe un lenguaje de marcas. Por lo tanto no se trata de un lenguaje en particular, sino una manera de definir lenguajes. Fue desarrollado por el World Wide Web Consortium (W3C) y se trata de una versión abreviada del SGML (Standard Generalized Markup Language). (Marco., 2013).

Un documento **XML** es, simplemente, un documento de texto que ofrece una representación estructural de los datos. Consta de dos estructuras, una física y otra lógica. Físicamente, el documento **XML** está compuesto por entidades (una o más) con contenido e identificadas por un

CAPÍTULO II - MARCO TEÓRICO

nombre. Lógicamente, contiene elementos (uno o más) delimitados por marcas de principio y fin. Cada uno de los elementos tienen un identificador genérico, y dentro de este, puede contener una serie de atributos (Mendoza., 2014).

Los archivos **XML** utilizados en **Android** para elementos como layouts que usará la aplicación para un funcionamiento correcto. Creamos entonces la interfaz de usuario de la aplicación, una apariencia física. El uso de este tipo de archivos no es imprescindible en la programación para **Android** puesto que puedes usar código **Java** para configurar la apariencia, cambiar formatos, mostrar elementos, ocultarlos. El porqué de su uso radica en mantener un diseño limpio y separar el código de la funcionalidad propia de la aplicación del código de diseño, para así facilitar las modificaciones que se realicen. (Condesa., 2013).

2.4.4. SQLite

SQLite es un proyecto de dominio público cuyo lanzamiento fue el 17 de agosto del 2000 creado por D. Richard Hipp²⁴

SQLite es un gestor de base de datos, lo que quiere decir que este programa tiene la versatilidad de manejar bases de datos y puede hacer con ellas un sinnúmero de funciones, las crea las manipula, las robustece, las modifica y las elimina como cualquier gestor de bases de datos. A diferencia de los sistemas de gestión de bases de datos cliente-servidor, el motor no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca se enlaza con el programa pasando a ser parte integral del mismo.

El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos.

En su versión 3, **SQLite** permite bases de datos de hasta 2 Terabytes de tamaño, y también permite la inclusión de campos tipo **BLOB** (La última versión estable 3.10.2 fue lanzada el 20 de enero de 2016).

²⁴ <http://www.sqlite.org>

2.4.4.1. Características SQLite

SQLite tiene la ventaja de ser multiplataforma, lo que quiere decir que funciona en los diversos sistemas operativos que existen, tanto por aquellos llamados de escritorio como por los sistemas operativos móviles ya que como dijimos trabaja con una librería muy liviana de 500 Kb.

SQLite usa un sistema de tipos inusual. En lugar de asignar un tipo a una columna como en la mayor parte de los sistemas de bases de datos SQL, los tipos se asignan a los valores individuales. Varios procesos o hilos pueden acceder a la misma base de datos sin problemas. Varios accesos de lectura pueden ser servidos en paralelo.

2.4.4.2. Lenguajes de programación de SQLite

SQLite tiene versatilidad de funcionar en varios lenguajes de programación gracias a drivers o a módulos creados por desarrolladores o usuarios que quieren que funcione en los lenguajes que ellos utilizan

- La biblioteca puede ser usada desde programas en C/C++
- Python incluye soporte para SQLite nativamente desde la versión 2.5
- Visual Basic 6 llamado VBSqlite
- Desde Delphi se puede usar SQLite a través de los componentes libres ZeosLib.
- PHP incluye SQLite, desde la versión 5.
- Desde Java se puede acceder mediante el driver de SQLite JDBC
- Desde .NET se puede acceder usando System.Data.SQLite
- Desde Lazarus 0.9.8 y Free Pascal 2.0.0, SQLite está disponibles
- Mac OS X v10.4 incluye SQLite
- BlitzMAX posee un MOD que permite trabajar con bases de datos SQLite.

2.4.5. Eclipse

Eclipse es una plataforma de desarrollo Open Source basada en Java. Eclipse fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para VisualAge. Actualmente es desarrollado por la Eclipse Foundation, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto.

En sí mismo Eclipse es un marco y un conjunto de servicios para construir un entorno de desarrollo a partir de componentes conectados, plugins. Existen plugins para el desarrollo en Java, JDT Java Development Tools, así como para el desarrollo en otros lenguajes como C/C++, PHP, Cobol, plataformas como Android, etc.

Eclipse es el IDE recomendado para el desarrollo sobre Android y se incluye de facto en el paquete Bundle de las Android Developer Tools, por lo que el proyecto hará uso de esta herramienta complementándose con los plugins necesarios para facilitar el desarrollo del mismo.

2.4.5.1. Entorno de desarrollo

Android ofrece un plugin para Eclipse que extiende la funcionalidad de éste y facilita el desarrollo de aplicaciones para Android, lo cual se hace mención en las *Figuras 11- 12- 13*. Además, ofrece las herramientas que utiliza este plugin como scripts para que puedan ser utilizados también desde otros entornos como Netbeans o IntelliJ IDEA²⁵. Entre las funcionalidades de este plugin se encuentra:

- Emulador de Android. Permite elegir entre distintos terminales móviles y la versión del sistema operativo.
- El acceso a herramientas de desarrollo de Android como tomar capturas de pantalla, la redirección de puertos, la posibilidad de depurar con puntos de parada o ver el estado de las hebras y los procesos corriendo en el sistema.)
- Editores de código para los distintos archivos de configuración (XML) que facilitan su comprensión y desarrollo.
- Interfaces gráficas que permiten el desarrollo de componentes visualmente.

²⁵ <http://www.jetbrains.com/idea/>



Figura 11: Vista del emulador de Android
Fuente: (S. Microsystems, “Oracle to buy sun.”, 2012)

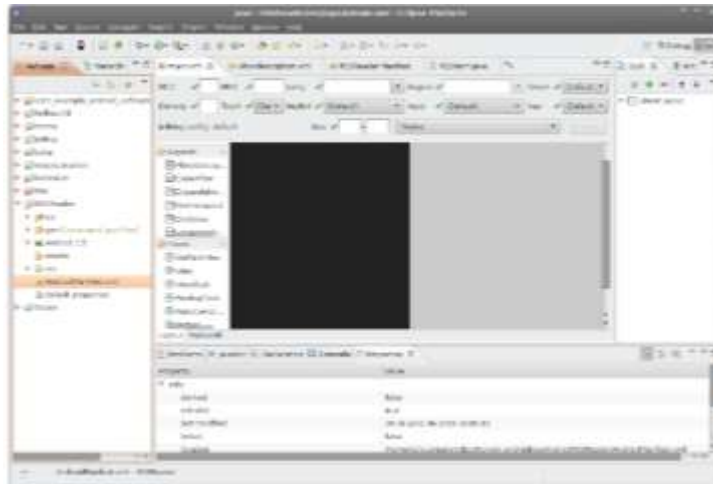


Figura 12: Vista del entorno de desarrollo de Android
Fuente: (S. Microsystems, “Oracle to buy sun.”, 2012)

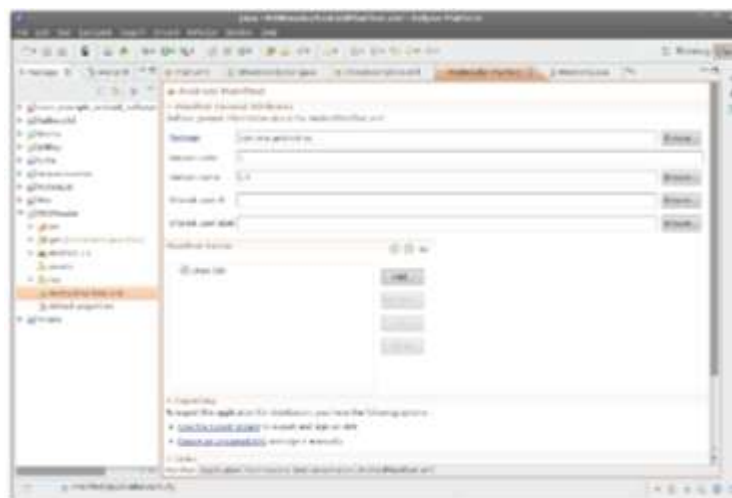


Figura 13: Vista de la plataforma Eclipse
Fuente: (S. Microsystems, “Oracle to buy sun.”, 2012)

2.4.5.2. Interfaz de usuario

En Android, la interfaz de usuario se construye a partir de los tipos **View** y **ViewGroup**. El tipo **View** es la base para los llamados “**widjets**”, objetos totalmente implementados como campos de texto o botones. El tipo **ViewGroup** es la base para los **layout**, que ofrecen diferentes tipos de vista, como **framelayout**, **relativelayout**, **linearLayout**, **tablelayout**. (Alejandro., 2013).

Es decir, colocamos una serie de objetos view (Texto, imágenes, botones) dentro de un viewGroup que nos lo colocará según el tipo de vista que tenga. La **Figura 14** muestra un ejemplo de la estructura comentada.

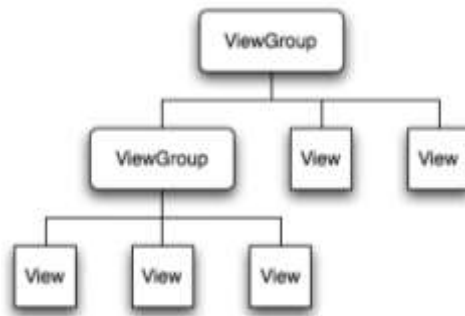


Figura 14: Jerarquía que define una interfaz de usuario
Fuente: (Alejandro., 2013)

La forma más común de establecer esta jerarquía es con un archivo XML que contenga la disposición de los distintos elementos. Por ejemplo un elemento `<TextView>` crea un objeto `TextView` en la interfaz, y un elemento `<LinearLayout>` crea un objeto `ViewGroup` equivalente. Por ejemplo, el fragmento de **Código 1** crearía un `ViewGroup` en el que sus elementos, un cuadro de texto y un botón, se mostrarán uno a continuación del otro:

```
<?xml version="1.0" encoding="utf-8">
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <TextView android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a Button" />
</LinearLayout>
```

Código 1: Ejemplo simple con texto y un boton
Fuente: (Alejandro., 2013)

2.4.5.3. Estructura de un proyecto Android en Eclipse

Un proyecto nuevo con el entorno de desarrollo eclipse, genera una serie de carpetas para posteriormente generar la aplicación. Por ejemplo como se ve en la **Figura 15** la estructura para el proyecto llamado Hola_Mundo (sgoliver, 2012).



Figura 15: Estructura de un proyecto Android
Fuente: (sgoliver, 2012)

El contenido de cada carpeta es indispensable para su funcionalidad y se diferencian por:

- **Carpeta /src/:** Contiene todo el código fuente de la aplicación. Aquí es donde debe agregarse todo el código java.
- **Carpeta /res/:** Contiene todos los ficheros de recursos necesarios para la aplicación: imágenes, audios, cadenas de texto que se puede apreciar en la **Figura 16**.

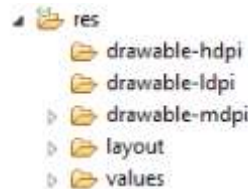


Figura 16: Estructura de la carpeta res de un proyecto Android
Fuente: (sgoliver, 2012)

Res/drawable/: Contienen imágenes de la aplicación dependiendo de su resolución.

Res/layout/: Contienen los archivos XML, pantallas de la interfaz gráfica.

Res/values/: Contiene otros recursos de la aplicación como por ejemplo cadenas de texto (strings.xml), estilos (styles.xml), colores (colors.xml).

- **Carpeta /gen/:** Contiene una serie de elementos de código generados automáticamente al compilar el proyecto.
- **AndroidManifest:** Contiene la definición en XML de los aspectos principales de la aplicación, como su nombre, versión, icono, pantallas, mensajes...o los permisos necesarios para su ejecución.

CAPÍTULO II - MARCO TEÓRICO

2.4.6. Especificaciones e instalación de la tecnología de desarrollo

Para poder desarrollar una aplicación en Android es necesario tener instalado en el ordenador de trabajo una serie de herramientas. (McCracken) (Lee).

Actualmente la instalación de las herramientas se realiza de una forma muy sencilla descargando e instalando las siguientes herramientas:

2.4.6.1. Java SE Development Kit (JDK)

Android hace uso de JDK para el desarrollo de aplicaciones. Este paquete se encuentra de forma libre en la página de Oracle que es la propietaria de Sun Microsystems, la compañía que desarrolló java, lo mencionado se lo puede descargar y apreciar en la *Figura 17- 18*.

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>



Figura 17: Descargando el JDK de Java en la pagina de Oracle
Fuente:<http://www.oracle.com>

Al pulsar sobre el enlace se muestra una lista para descargar JDK según el sistema operativo:



Product File Description	File Size	Download
Linux ARMv7 (32-bit) Float-HW	57.1 MB	jdk-7u51-linux-armv7-32bit.tar.gz
Linux ARMv7 (32-bit) Float-HW	67.68 MB	jdk-7u51-linux-armv7-32bit.tar.gz
Linux x86	115.05 MB	jdk-7u51-linux-i386.tar.gz
Linux x86	122.99 MB	jdk-7u51-linux-i386.tar.gz
Linux x86	116.68 MB	jdk-7u51-linux-x86.tar.gz
Linux x86	121.8 MB	jdk-7u51-linux-x86.tar.gz
Mac OS X x64	179.46 MB	jdk-7u51-macosx-x64.pkg
Ubuntu x86 (32-bit package)	140.02 MB	jdk-7u51-ubuntu-i386.tar.gz
Ubuntu x86	95.73 MB	jdk-7u51-ubuntu-i386.tar.gz
Ubuntu x64 (64-bit package)	24.02 MB	jdk-7u51-ubuntu-amd64.tar.gz
Ubuntu x64	10.28 MB	jdk-7u51-ubuntu-amd64.tar.gz
Solaris SPARC (32-bit package)	128.28 MB	jdk-7u51-solaris-sparc.tar.gz
Solaris SPARC	90.19 MB	jdk-7u51-solaris-sparc.tar.gz
Solaris SPARC-64 (64-bit package)	22.94 MB	jdk-7u51-solaris-sparc64.tar.gz
Solaris SPARC-64	10.22 MB	jdk-7u51-solaris-sparc64.tar.gz
Windows x86	122.44 MB	jdk-7u51-windows-i386.exe
Windows x64	125.46 MB	jdk-7u51-windows-x64.exe

Figura 18: Lista de JDK dividida por sistemas operativos
Fuente:<http://www.oracle.com>

2.4.6.2. Android SDK, Android ADT y Eclipse

Android SDK proporciona las librerías de las APIs, documentación y ejemplos y todas las herramientas para poder construir, comprobar y depurar aplicaciones. Además esta herramienta permite emular las aplicaciones sin necesitar de un dispositivo real.

Android SDK se descarga desde la página del desarrollador de Android:

<http://developer.android.com/sdk/index.html>



Figura 19 Enlace de descarga de SDK
Fuente: <http://developer.android.com/sdk/index.html>

Al pulsar sobre el botón, en la **Figura 19** se descargará un archivo .rar que contiene el directorio de instalación de las herramientas. Para terminar la instalación hay que acceder al directorio de Eclipse y pulsar sobre su ejecutable para después instalar Android ADT, se lo puede apreciar en la **Figura 20** (Al pulsar sobre el ejecutable de Eclipse se ejecuta directamente el ADT y se instala Eclipse).



Figura 20: Ejecución de ADT
Fuente: Elaboración propia

El ADT (Android Development Tools o Herramientas de Desarrollo de Android) es el plug-in de Eclipse. El conjunto de herramientas de ADT suministra:

- El asistente de proyectos de Android para generar los archivos que requiere un proyecto.
- Editores de recursos específicos que necesita Android.
- Gestor de AVD. Esta herramienta que proporciona una interfaz gráfica para el lanzamiento de emuladores para simulación de una aplicación sin necesitar el uso de un dispositivo real.
- DDMS (Dalvik Debug Monitor Server): esta herramienta se usa para acciones tales como el depurado y control de aplicaciones, captura de pantalla, logcat, procesos, llamadas entrantes y gestión de archivos.
- Integración con la utilidad de log LogCat de Android y AHV (Android Hierarchy Viewer).
- Compilación y ejecución de aplicaciones Android para su ejecución en emuladores y dispositivos.
- Herramientas de firma de código con certificados digitales y empaquetado (APK, Android Packages Kit) para la distribución de aplicaciones.

A la vez que se instala Android ADT, se instala Android SDK (**Figura 21**):

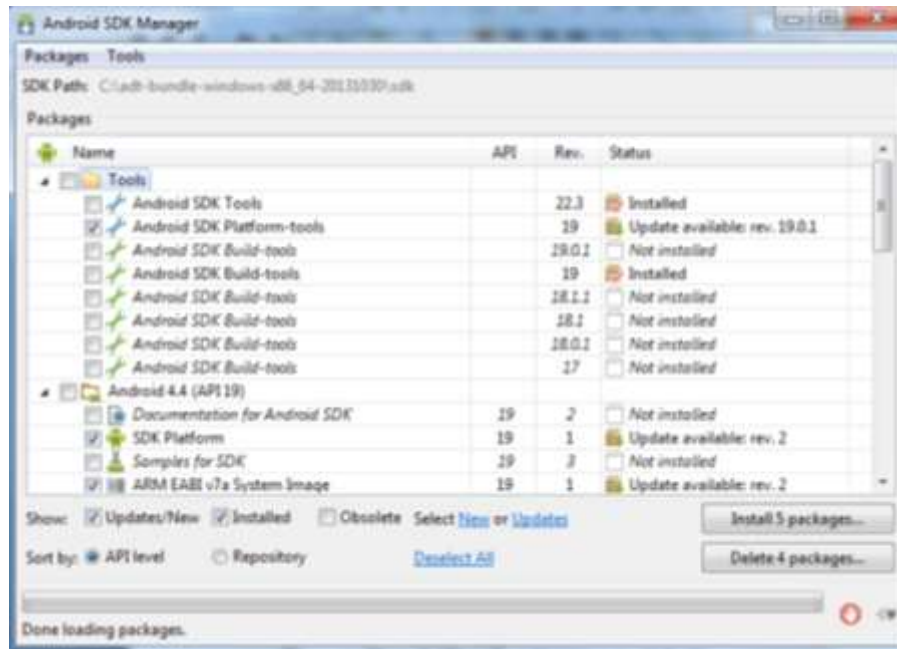


Figura 21: Android SDK Manager
Fuente: Elaboración propia

Con esto, ya estarían instaladas las herramientas necesarias. La forma de instalar estas herramientas han cambiado hace poco, pues cuando se comenzó este tesis, había que descargar cada una de las herramientas por separado.

En el momento en el que se comenzó este proyecto, Eclipse IDE había que descargarlo desde la página:

<http://www.eclipse.org/downloads/>

Eclipse es el IDE totalmente compatible con el plug-in Android Development Tools (ADT). En ese momento, la forma de instalar el ADT (*Figura 22*) era dentro del entorno de Eclipse pulsando **Help->Install New Software**. De esta forma, se abría una nueva ventana que había que rellenar con una serie de datos: (Soluciones móviles de Safelayer S.A, 2014) (Página del desarrollador de Android;, 2014)



*Figura 22: Instalacion del ADT de Android
Fuente: Elaboración propia*

2.5. Metodologías ágiles para aplicaciones móviles

El uso de métodos ágiles de desarrollo de software ha recibido tanto apoyo como argumentos opuestos. El principal argumento en contra es la falta de validación. Hay también una cierta cantidad de incertidumbre para distinguir los métodos ágiles de programación ad-hoc. Sin embargo, algunos autores defienden la teoría de que los métodos ágiles proporcionan un enfoque de desarrollo organizado. (tutkimuskeskus, 2006)

Cuando se trata de comparar las características de las aplicaciones móviles a las de un método ágil, la dificultad proviene, en parte, del hecho de que los límites de las metodologías ágiles no están claramente establecidos. Una visión global de la investigación en ese campo se presenta en (Dingsoyr, 2009) donde los autores dividen su investigación en cuatro categorías: introducción y adaptación, factores humanos y sociales, la percepción de los métodos ágiles, y estudios comparativos. Los resultados indican que la introducción de los métodos ágiles a proyectos de software de tamaño reducido genera grandes beneficios, especialmente si las prácticas ágiles no sustituyen completamente a las tradicionales, sino que trabajan en conjunto.

Otro punto de vista (Abrahamsson, Keynote: Mobile software development–, 2005) realiza una comparación directa entre las características del método ágil y características de las aplicaciones móviles, centrándose en la cantidad de documentación producida, la planificación involucrada, el tamaño del equipo de desarrollo, la identificación del cliente, y la orientación a objetos. Excepto la identificación del cliente, todas las características de las metodologías ágiles son adecuadas para el desarrollo de aplicaciones móviles. El cliente puede ser identificado como el distribuidor de

software, sin embargo, especialmente en el caso de las aplicaciones móviles, el problema de la identificación del cliente es mucho más complejo.

Una metodología de desarrollo nueva, especialmente diseñada para el desarrollo de aplicaciones móviles, recibe el nombre de Mobile-D y es propuesta por Pekka Abrahamsson y su equipo del VTT (Valtion Teknillinen Tutkimuskeskus, en inglés Technical Research Centre of Finland) en Finlandia que lideran una corriente muy importante de desarrollo ágil (Abrahamsson, Agile software development of mobile information systems, 2007) muy centrada en las plataformas móviles, y que se presenta con más detalle en. (P. Abrahamsson, 2004).

El método se basa en prácticas ágiles como Extreme Programming y crystal; las prácticas asociadas a Mobile-D incluyen desarrollo basado en pruebas, la programación en parejas, integración continua y refactorización, así como las tareas de mejora de procesos de software; según Abrahamsson (P. Abrahamsson, 2004) Mobile-D debe ser utilizado por un equipo de no más de diez desarrolladores, trabajando en conjunto para suministrar un producto listo en un plazo máximo de diez semanas.

Otro punto de vista muy importante en la actualidad es la propuesta de Rahimian y Ramsin (Ramsin, 2008) HMD (Hybrid Methodology Design), la cual, se apoya en una combinación del desarrollo adaptativo de software (Adaptive Software Development, ASD) y el diseño de nuevos productos (New Product Development, NPD), parte del ciclo de vida tradicional (análisis, diseño, implementación, pruebas y desarrollo) e incluyen además una fase de comercialización.

La más reciente propuesta de metodologías diseñadas específicamente para aplicaciones móviles y que aún se encuentra en etapa experimental se denomina Mobile Development Process Spiral (A. Nosseir, 2012), el cual es un modelo impulsado por la usabilidad y toma como base el modelo espiral.

2.5.1. Características y requerimientos específicos del entorno móvil

El desarrollo de aplicaciones móviles difiere del desarrollo de software tradicional en muchos aspectos, lo que provoca que las metodologías usadas para estos entornos también difieran de las del software clásico. Esto es porque el software móvil tiene que satisfacer una serie de

CAPÍTULO II - MARCO TEÓRICO

requerimientos y condicionantes especiales (Forman, 1994) (Heyes, 2002) (Dunlop, 2002) que lo hace más complejo:

- **Canal** radio: consideraciones tales como la disponibilidad, las desconexiones, la variabilidad del ancho de banda, la heterogeneidad de redes o los riesgos de seguridad han de tenerse especialmente en cuenta en este entorno de comunicaciones móviles.
- **Movilidad:** aquí influyen consideraciones como la migración de direcciones, alta latencia debido a cambio de estación base o la gestión de la información dependiente de localización. Sobre esta última, de hecho, se pueden implementar un sinnúmero de aplicaciones, pero la información de contexto asociada resulta muchas veces incompleta y varía frecuentemente.
- **Portabilidad:** la característica portabilidad de los dispositivos terminales implica una serie de limitaciones físicas directamente relacionadas con el factor de forma de los mismos, como el tamaño de las pantallas (algo que ha variado sustancialmente con la popularización de las pantallas táctiles), o del teclado, limitando también el número de teclas y su disposición.
- **Fragmentación** de la industria: la existencia de una considerable variedad de estándares, protocolos y tecnologías de red diferentes añaden complejidad al escenario del desarrollo móvil.
- **Capacidades** limitadas de los terminales: aquí incluimos factores como la baja potencia de cálculo o gráfica, los riesgos en la integridad de datos, las interfaces de usuario poco funcionales en muchos aspectos, la baja capacidad de almacenamiento, la duración de las baterías o la dificultad para el uso de periféricos en movilidad.
Factores todos que, por otro lado, están evolucionando en la dirección de la convergencia de los ultra portátiles (*netbooks*) con los dispositivos inteligentes (*smartphones*) constituyendo cada vez menos un elemento diferencial.
- **Diseño:** desde el punto de vista del desarrollo, el diseño multitarea y la interrupción de tareas es clave para el éxito de las aplicaciones de escritorio; pero la oportunidad y frecuencia de éstas es mucho mayor que en el software tradicional, debido al entorno móvil que manejan, complicándose todavía más debido a la limitación de estos dispositivos.

- **Usabilidad:** las necesidades específicas de amplios y variados grupos de usuarios, combinados con la diversidad de plataformas tecnológicas y dispositivos, hacen que el diseño para todos se convierta en un requisito que genera una complejidad creciente difícil de acotar.
- **Time-to-market:** en un sector con un dinamismo propio, dentro de una industria en pleno cambio, los requisitos que se imponen en términos de tiempo de lanzamiento son muy estrictos y añaden no poca dificultad en la gestión de los procesos de desarrollo.

El diseño de sistemas de software móvil es, por tanto, bastante más complejo que el tradicional visto en los otros proyectos de desarrollo, forzando a los investigadores a reconsiderar el uso de las metodologías actuales de desarrollo de software. Como se ha visto, el uso de metodologías ágiles es el medio más apropiado para el desarrollo de tecnología en móviles, aunque las características especiales de los terminales y de las redes de telefonía móvil demandan algunos ajustes sobre las actuales metodologías ágiles.

Los investigadores han intentado mejorar este *statu quo* a través de mejoras o invenciones de metodologías. Por desgracia, estas iniciativas han sido relativamente pocas y lejanas entre ellas. La mayoría del trabajo desarrollado ha estado enfocado a aspectos del desarrollo software de bajo nivel (orientado a implementación), mientras que el alto nivel (orientado a metodologías) aún permanecía sin ser convenientemente atendido. La mayoría del desarrollo de software usa metodologías o elementos de las mismas con más de treinta años de vigencia, que se basan en la introducción de ciertas modificaciones orientadas a las necesidades específicas del desarrollo móvil.

2.6. Metodologías utilizadas

Usualmente en el desarrollo de algún software, del tipo que fuera se utiliza una sola metodología pero esta Tesis se caracteriza por que relaciona Algoritmos Genéticos con Tabú y Metodologías Híbridas para que el modelo del sistema sea eficiente y eficaz.

2.6.1. Diseño de la metodología híbrida (Hybrid Methodology Design)

Esta metodología utiliza el modelo iterativo incremental para el proceso de desarrollo y así lograr la rápida entrega de software y mejorar las capacidades de gestión de riesgos. Algunas de las características ágiles que se destacan y que también se alinean con las necesidades de desarrollo de aplicaciones móviles son según (Harishankar, 2013):

- Desarrollo basado en pruebas.
- Participación continua del cliente.
- Establecimiento de prioridades en los requisitos.
- Comunicación efectiva.
- Calidad garantizada.
- Desarrolladores expertos.
- Revisión de todo el proceso y sesiones de aprendizaje.
- Proceso de adaptación.

2.6.1.1. Iteraciones de diseño de la metodología híbrida

El Diseño de la metodología Híbrida se apoya en una combinación del desarrollo adaptativo de software (*Adaptive Software Development, ASD*) y el diseño de nuevos productos (*New Product Development, NPD*) (Ramsin, 2008). Esto es una decisión crítica para relacionar más del lado del desarrollo de productos que del lado de la gestión de proyectos, lo cual quiere decir que una de las características más sensibles, desde el punto de vista metodológico, para la consolidación de una metodología propia de un entorno móvil, es la presión de los plazos para llegar al mercado, un mercado volátil y altamente dinámico.

A pesar de ser uno de los pocos intentos metodológicos serios recientes, la propuesta (Ramsin, 2008). Se apoya en un proceso iterativo de diseño híbrido de metodologías (Hybrid Methodology Design, HMD) y parte del ciclo de vida tradicional representado en la **Figura 23**.

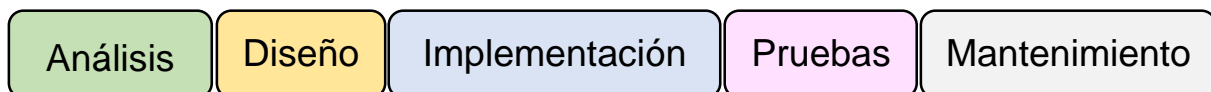


Figura 23: Ciclo de vida clásico para un proceso de desarrollo de software
Fuente: (Ramsin, 2008) - Traducción e interpretación propia

CAPÍTULO II - MARCO TEÓRICO

En la primera iteración se divide la fase de análisis con la intención de mitigar riesgos de desarrollo; de la misma forma, el diseño también se segmenta para introducir algo de diseño basado en arquitectura. La implementación y las pruebas sin embargo se fusionan introduciendo conceptos de desarrollo orientado a pruebas (*Test-DrivenDevelopment, TDD*). Aparece además una fase de comercialización, incidiendo en el sesgo hacia el desarrollo de producto que se imponen en el escenario del desarrollo de aplicaciones para plataformas móviles.



Figura 24: Primera iteración en el diseño híbrido (instanciación de análisis y diseño)

Fuente: (Ramsin, 2008) - Traducción e interpretación propia

Tras la instanciación realizada en la primera iteración metodológica (**Figura 24**), la segunda realiza una integración de ciertas partes de los modelos NPD, añadiendo la generación de ideas en el inicio del ciclo y una prueba de mercado antes de lanzar la fase de comercialización, lo cual se puede apreciar en la **Figura 25**.

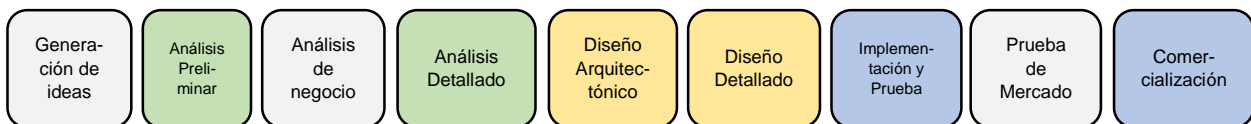


Figura 25: Segunda iteración en el diseño híbrido: integración

Fuente: (Ramsin, 2008) - Traducción e interpretación propia

La tercera iteración (**Figura 26**) integra directamente el "motor de desarrollo" de los métodos adaptativos de desarrollo (ASD) muy orientados al aseguramiento de la calidad en los procesos de desarrollo.

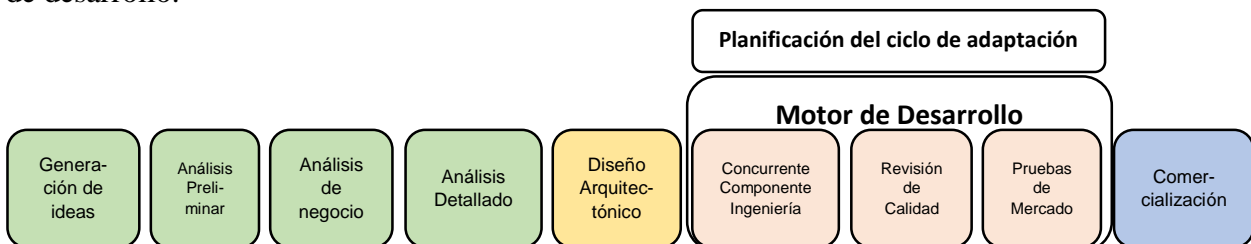


Figura 26: Tercera iteración en el diseño híbrido: motor de desarrollo

Fuente: (Ramsin, 2008) - Traducción e interpretación propia

Pensando en aquella propiedad "ideal" de disponer de la arquitectura física en una fase temprana del proceso, en la cuarta iteración (**Figura 27**) se añaden elementos de prototipado; se refina,

CAPÍTULO II - MARCO TEÓRICO

además, la fase de iniciación del proyecto, sobre la base del mismo elemento de los procesos adaptativos (Harishankar, 2013).

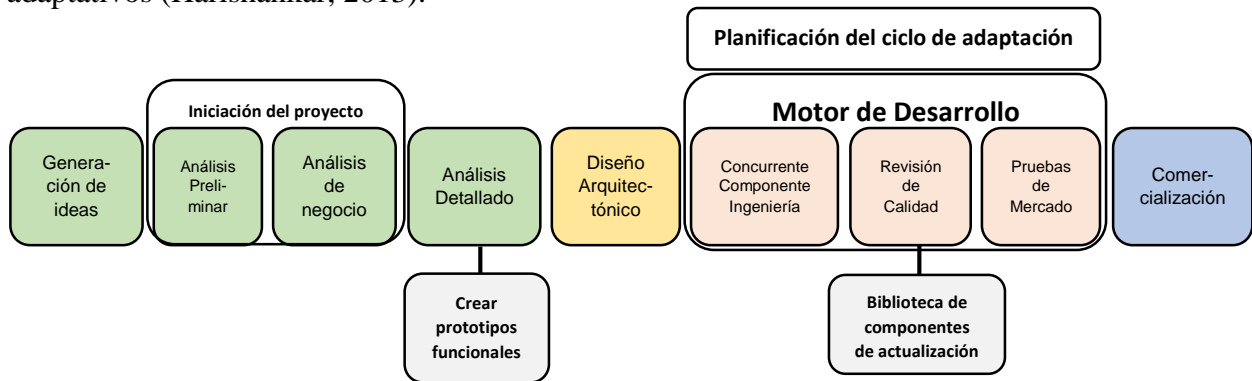


Figura 27 Cuarta iteración en el diseño híbrido: prototipado
Fuente: (Ramsin, 2008) - Traducción e interpretación propia

2.6.2. Búsqueda Tabú

El termino Búsqueda Tabú (*Tabú Search* - TS) fue introducido en 1986 por Fred Glover en el mismo artículo que introdujo el termino meta heurística (Glover F. , 1986). Los principios fundamentales de la búsqueda fueron elaborados en una serie de artículos de finales de los años 80 y principios de los años de 1990, que fueron luego unificados en el libro “*Tabú Search*” en 1997 (Glover F. a., 1997). El destacado éxito de la búsqueda tabú para resolver problemas de optimización duros, especialmente aquellos que surgen en aplicaciones del mundo real, ha causado una explosión de nuevas aplicaciones durante los últimos años. (Glover F. L., 2006).

Su filosofía se basa en derivar y explotar una colección de estrategias inteligentes para la resolución de problemas, basadas en procedimientos implícitos y explícitos de aprendizaje. El marco de memoria adaptativa de la búsqueda tabú no solo explota la historia del proceso de resolución del problema, sino que también exige la creación de estructuras para hacer posible tal explotación. De esta forma, los elementos prohibidos en la búsqueda tabú reciben este estatus por la confianza en una memoria evolutiva, que permite alterar este estado en función del tiempo y las circunstancias. En este sentido es posible asumir que la búsqueda tabú está basada en determinados conceptos que unen los campos de inteligencia artificial y optimización.

CAPÍTULO II - MARCO TEÓRICO

Más particularmente, la búsqueda tabú está basada en la premisa de que para clasificar un procedimiento de resolución como inteligente, es necesario que este incorpore *memoria adaptativa* y *exploración responsiva*. La memoria adaptativa en búsqueda tabú permite la implementación de procedimientos capaces de realizar la búsqueda en el espacio de soluciones eficaz y eficientemente. Dado que las decisiones locales están por tanto guiadas por información obtenida a lo largo del proceso de búsqueda, la búsqueda tabú contrasta con diseños que por contra confían en procesos semi aleatorios, que implementan una forma de muestreo. La memoria adaptativa también contrasta con los típicos diseños de memoria rígidos tales como las estrategias de ramificación y acotación. El énfasis en la exploración responsiva considerada en la búsqueda tabú deriva de la suposición de que una mala elección estratégica puede proporcionar más información que una buena elección realizada al azar, dado que una elección estratégica mala puede proporcionar pistas útiles sobre como guiar la búsqueda hacia zonas prometedoras. Por lo tanto, la exploración responsiva integra los principios básicos de la búsqueda inteligente; explota las características de las soluciones buenas a la vez que explora nuevas regiones prometedoras.

La búsqueda Tabú se caracteriza por estar dividido en pasos:

- Paso 1. Selección de la solución inicial, debe de ser una solución factible
- Paso 2. Elección del entorno o vecindario y generación de una nueva solución.
- Paso 3. Evaluación de la función objetivo
- Paso 4. Actualización de la mejor solución
- Paso 5. Criterio de parada

2.6.2.1. Ejemplo ilustrativo

Los problemas de permutaciones son una clase importante de problemas en optimización, y ofrecen un modo muy útil para demostrar algunas de las consideraciones que deben ser tratadas en el dominio combinatorio. Las instancias clásicas de problemas de permutaciones incluyen los problemas del viajante de comercio, asignación cuadrática, secuenciación de la producción, y una variedad de problemas de diseño. Como base para la ilustración, consideremos el problema de secuenciación de tareas en una única máquina.

CAPÍTULO II - MARCO TEÓRICO

El objetivo de este problema es encontrar un orden para secuenciar las tareas en la máquina de tal forma que se minimice el retraso total en la ejecución de las tareas. Cada tarea j , $j = 1, 2, \dots, n$, tiene asignados un tiempo de procesamiento p_j y un día de finalización d_j . De forma adicional, se podría considerar un valor de penalización por retardo en la finalización de las tareas que dependería de la tarea considerada, w_j . Por tanto, la función a minimizar se expresa en la **Ecuación 1**.

$$F = \sum_{j=1}^n w_j [C_j - d_j]^+$$

Ecuación 1: Función Tabú

Fuente: (Glover F. L., 2006)

Donde C_j es el tiempo de finalización de la tarea j y $[C_j - d_j]^+ = \max\{0, C_j - d_j\}$. El tiempo de finalización de una tarea j , C_j , es igual al tiempo de procesamiento de la tarea j más la suma de los tiempos de procesamiento de todas las tareas que se realizan antes que j . El problema consiste en determinar el orden de secuenciación de las tareas que minimiza el valor de la función objetivo F . Una secuenciación de las tareas, para una permutación, define completamente a una solución.

Nos centramos, por tanto, en el problema de secuenciación de tareas en una única máquina para introducir e ilustrar los componentes básicos de la búsqueda tabú. Supongamos que se consideran 6 tareas para su secuenciación en la máquina. A modo de ilustración, supongamos que este problema de 6 tareas tiene tiempos de procesamiento dados por (5, 8, 2, 6, 10, 3), días de terminación especificados por (9, 10, 16, 7, 20, 23), y penalizaciones por retraso $w_j = 1$ para $j = 1, 2, \dots, 6$. Deseamos diseñar un método capaz de encontrar una solución óptima o cercana a la óptima explorando solo un pequeño subconjunto de todas las permutaciones posibles.

Primero asumimos que puede construirse una solución inicial para este problema de alguna manera inteligente, es decir, tomando ventaja de la estructura específica del problema. Supongamos que la solución inicial de nuestro problema es la que aparece en la **Figura 28**.

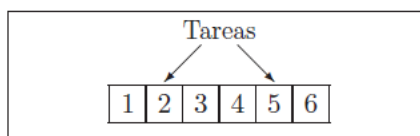


Figura 28: Permutación inicial

Fuente: (Glover F. L., 2006)

CAPÍTULO II - MARCO TEÓRICO

La ordenación de la **Figura 28** especifica que la tarea 1 se realiza en primer lugar, seguida por la tarea 2, etc. El valor de la función objetivo para esta solución es 39. Los métodos TS operan bajo el supuesto de que se puede construir un entorno para identificar “soluciones adyacentes” que puedan ser alcanzadas desde la solución actual. Los intercambios por pares son frecuentemente usados para definir entornos en problemas de permutaciones, identificando movimientos que conducen una solución a la siguiente. En nuestro problema, un intercambio cambia la posición de dos tareas como se ilustra en la **Figura 29**. Por tanto, el entorno completo de una solución en nuestro ejemplo ilustrativo está formado por 15 soluciones adyacentes que pueden ser obtenidas a partir de estos intercambios tal como muestra el Cuadro 1.

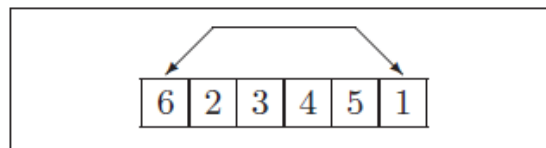


Figura 29: Intercambio de las tareas 1 y 6
Fuente: (Glover F. L., 2006)

Tal como observamos en la tabla 5, asociado a cada intercambio hay un valor de movimiento, que representa el cambio sobre el valor de la función objetivo como resultado del intercambio realizado. Los valores de los movimientos generalmente proporcionan una base fundamental para evaluarla calidad de los mismos, aunque también pueden ser importantes otros criterios. Un mecanismo principal para explotar la memoria en la búsqueda tabú es clasificar un subconjunto de movimientos en un entorno como prohibidos (o tabú). La clasificación depende de la historia de la búsqueda, determinada mediante lo reciente o frecuente que ciertos movimientos o componentes de soluciones, llamados *atributos*, han participado en la generación de soluciones pasadas. Por ejemplo, un atributo de un movimiento es la identidad del par de elementos que cambian posiciones (en este caso, las dos tareas intercambiadas) (**Tabla 5**).

CAPÍTULO II - MARCO TEÓRICO

Tareas		F	valor del movimiento	abs($d_i - d_j$)
i	j			
1	2	40	1	9
1	3	42	3	7
1	4	38	-1	2
1	5	64	25	11
1	6	47	8	14
2	3	41	2	6
2	4	37	-2	3
2	5	49	10	10
2	6	39	0	13
3	4	42	3	9
3	5	54	15	4
3	6	48	9	7
4	5	43	4	13
4	6	38	-1	16
5	6	32	-7	3

Tabla 5: Entorno de intercambios

Fuente: (Glover F. L., 2006)

Como base para evitar la búsqueda desde combinaciones de intercambio repetidas usadas en el pasado reciente, invirtiendo potencialmente los efectos de movimientos anteriores por intercambios que podrían devolver a posiciones previas, clasificaremos como tabú todos los intercambios compuestos por cualquiera de los pares de tareas más recientes; en este caso, para propósitos ilustrativos, las tres más recientes. Esto significa que un par de tareas será tabú durante un periodo de 3 iteraciones. Dado que intercambiar las tareas 2 y 5 es lo mismo que intercambiarlas tareas 5 y 2, ambos intercambios pueden ser representados por el par (2; 5). Por lo tanto, se puede usar una estructura de datos como la usada en la **Figura 30**.

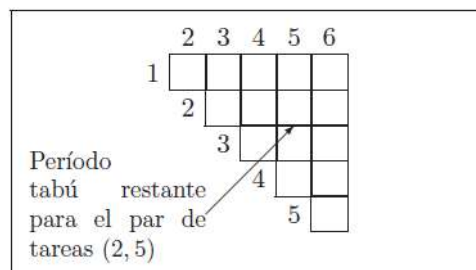


Figura 30: Estructura de datos Tabu

Fuente: (Glover F. L., 2006)

Cada celda de la estructura de la **Figura 30** contiene el número de iteraciones restantes hasta que las capas correspondientes puedan nuevamente intercambiar posiciones. Por tanto, si la celda (2; 5) tuviera un valor de cero, entonces las tareas 2 y 5 estarían disponibles para intercambiar posiciones. Por otro lado, si la celda tuviera un valor de 2, entonces las tareas no podrían intercambiar posiciones durante las dos iteraciones siguientes (es decir, un intercambio que cambia estas tareas es clasificado como tabú).

CAPÍTULO II - MARCO TEÓRICO

Para implementar restricciones tabú, debe tenerse en cuenta una excepción importante: las restricciones tabú no son inviolables bajo cualquier circunstancia. Cuando un movimiento tabú resultara en una solución mejor que cualquiera visitada hasta ese momento, su clasificación tabú podría ser reemplazada. Una condición que permite que ocurra tal reemplazo se llama *criterio de aspiración*. A continuación se muestran 7 iteraciones del procedimiento de búsqueda tabú básico, que usa la restricción tabú de tareas emparejadas.

Solución actual	Estructura tabú	Primeros 5 candidatos Valor de movimiento																																																																																	
<table border="1"> <tr><td>1</td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>2</td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>3</td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>4</td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>5</td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>6</td><td></td><td></td><td></td><td></td><td></td></tr> </table>	1						2						3						4						5						6						<table border="1"> <tr><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr> <tr><td>1</td><td></td><td></td><td></td><td></td></tr> <tr><td>2</td><td></td><td></td><td></td><td></td></tr> <tr><td>3</td><td></td><td></td><td></td><td></td></tr> <tr><td>4</td><td></td><td></td><td></td><td></td></tr> <tr><td>5</td><td></td><td></td><td></td><td></td></tr> <tr><td>6</td><td></td><td></td><td></td><td></td></tr> </table> <p>$F = 39$</p>	2	3	4	5	6	1					2					3					4					5					6					<table border="1"> <tr><td>5,6</td><td>-7*</td></tr> <tr><td>2,4</td><td>-2</td></tr> <tr><td>1,4</td><td>-1</td></tr> <tr><td>4,6</td><td>-1</td></tr> <tr><td>2,6</td><td>0</td></tr> </table>	5,6	-7*	2,4	-2	1,4	-1	4,6	-1	2,6	0
1																																																																																			
2																																																																																			
3																																																																																			
4																																																																																			
5																																																																																			
6																																																																																			
2	3	4	5	6																																																																															
1																																																																																			
2																																																																																			
3																																																																																			
4																																																																																			
5																																																																																			
6																																																																																			
5,6	-7*																																																																																		
2,4	-2																																																																																		
1,4	-1																																																																																		
4,6	-1																																																																																		
2,6	0																																																																																		

Figura 31: Iteración 0
Fuente: (Glover F. L., 2006)

La solución de partida de la **Figura 31** tiene un valor de la función objetivo $F = 39$, y la estructura de los datos tabú esta inicialmente vacía, es decir, está llena de ceros, indicando que ningún movimiento está clasificado como tabú al comienzo de la búsqueda. Después de evaluar los movimientos de intercambio candidatos, se muestran en la tabla para la iteración 0 los cinco primeros movimientos (en términos de valores de movimiento). Para minimizar localmente el retraso total en la ejecución de las tareas, intercambiamos las posiciones de las tareas 5 y 6, como se indica a través del asterisco en la Figura 36. El decremento total de este movimiento es igual a 7 unidades, con lo que el valor de la función objetivo pasa a ser $F = 32$.

Solución tabú	Estructura tabú	Primeros 5 candidatos Valor de movimiento																																																																																	
<table border="1"> <tr><td>1</td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>2</td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>3</td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>4</td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>6</td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>5</td><td></td><td></td><td></td><td></td><td></td></tr> </table>	1						2						3						4						6						5						<table border="1"> <tr><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr> <tr><td>1</td><td></td><td></td><td></td><td></td></tr> <tr><td>2</td><td></td><td></td><td></td><td></td></tr> <tr><td>3</td><td></td><td></td><td></td><td></td></tr> <tr><td>4</td><td></td><td></td><td></td><td></td></tr> <tr><td>5</td><td></td><td></td><td></td><td></td></tr> <tr><td>6</td><td></td><td></td><td></td><td></td></tr> </table> <p>$F = 32$</p>	2	3	4	5	6	1					2					3					4					5					6					<table border="1"> <tr><td>2,4</td><td>-2*</td></tr> <tr><td>1,4</td><td>-1</td></tr> <tr><td>1,2</td><td>1</td></tr> <tr><td>2,3</td><td>2</td></tr> <tr><td>4,6</td><td>2</td></tr> </table>	2,4	-2*	1,4	-1	1,2	1	2,3	2	4,6	2
1																																																																																			
2																																																																																			
3																																																																																			
4																																																																																			
6																																																																																			
5																																																																																			
2	3	4	5	6																																																																															
1																																																																																			
2																																																																																			
3																																																																																			
4																																																																																			
5																																																																																			
6																																																																																			
2,4	-2*																																																																																		
1,4	-1																																																																																		
1,2	1																																																																																		
2,3	2																																																																																		
4,6	2																																																																																		

Figura 32: Iteración 1
Fuente: (Glover F. L., 2006)

CAPÍTULO II - MARCO TEÓRICO

La nueva solución actual tiene un valor de función objetivo $F = 32$ (es decir, el retraso total anterior más el valor del movimiento seleccionado). La estructura tabú de la **Figura 32** ahora muestra que el intercambio de las posiciones de las tareas 5 y 6 se prohíbe durante 3 iteraciones. El movimiento que proporciona la mayor mejora en este paso es el intercambio de las tareas 2 y 4 con un decremento de 2 unidades.

Solución actual	Estructura tabú	Primeros 5 candidatos																																																									
<table border="1"> <tr><td>1</td></tr> <tr><td>4</td></tr> <tr><td>3</td></tr> <tr><td>2</td></tr> <tr><td>6</td></tr> <tr><td>5</td></tr> </table>	1	4	3	2	6	5	<table border="1"> <tr><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr> <tr><td>1</td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td>2</td><td></td><td>3</td><td></td></tr> <tr><td></td><td></td><td>3</td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td>4</td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td>5</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td>2</td></tr> </table> <p>$F = 30$</p>	2	3	4	5	6	1						2		3				3						4						5						2	<table border="1"> <tr><td>1, 4</td><td>-2</td><td>*</td></tr> <tr><td>1, 3</td><td>1</td><td></td></tr> <tr><td>3, 4</td><td>2</td><td></td></tr> <tr><td>2, 4</td><td>2</td><td>T</td></tr> <tr><td>2, 6</td><td>2</td><td></td></tr> </table>	1, 4	-2	*	1, 3	1		3, 4	2		2, 4	2	T	2, 6	2	
1																																																											
4																																																											
3																																																											
2																																																											
6																																																											
5																																																											
2	3	4	5	6																																																							
1																																																											
	2		3																																																								
		3																																																									
			4																																																								
				5																																																							
					2																																																						
1, 4	-2	*																																																									
1, 3	1																																																										
3, 4	2																																																										
2, 4	2	T																																																									
2, 6	2																																																										

Figura 33: Iteración 2
Fuente: (Glover F. L., 2006)

La nueva solución actual tiene un retraso en la ejecución de las tareas de 30. En esta iteración se clasifican como tabú dos intercambios como se indica mediante las entradas distintas de cero en la estructura tabú. Note que la entrada (5; 6) ha disminuido de 3 a 2, indicando que su periodo tabú original de 3 ahora tiene 2 iteraciones restantes. En este momento, el intercambio de las tareas 1 y 4 conduce a una nueva mejora en el valor de la función objetivo, disminuyendo en dos unidades. La **Figura 33** muestra ahora 3 movimientos clasificados como tabú.

Solución actual	Estructura tabú	Primeros 5 candidatos																																																									
<table border="1"> <tr><td>4</td></tr> <tr><td>1</td></tr> <tr><td>3</td></tr> <tr><td>2</td></tr> <tr><td>6</td></tr> <tr><td>5</td></tr> </table>	4	1	3	2	6	5	<table border="1"> <tr><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr> <tr><td>1</td><td></td><td>3</td><td></td><td></td></tr> <tr><td></td><td>2</td><td></td><td>2</td><td></td></tr> <tr><td></td><td></td><td>3</td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td>4</td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td>5</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td>1</td></tr> </table> <p>$F = 28$</p>	2	3	4	5	6	1		3				2		2				3						4						5						1	<table border="1"> <tr><td>1, 4</td><td>2</td><td>T</td></tr> <tr><td>1, 3</td><td>2</td><td>*</td></tr> <tr><td>2, 6</td><td>2</td><td></td></tr> <tr><td>1, 2</td><td>3</td><td></td></tr> <tr><td>2, 3</td><td>3</td><td></td></tr> </table>	1, 4	2	T	1, 3	2	*	2, 6	2		1, 2	3		2, 3	3	
4																																																											
1																																																											
3																																																											
2																																																											
6																																																											
5																																																											
2	3	4	5	6																																																							
1		3																																																									
	2		2																																																								
		3																																																									
			4																																																								
				5																																																							
					1																																																						
1, 4	2	T																																																									
1, 3	2	*																																																									
2, 6	2																																																										
1, 2	3																																																										
2, 3	3																																																										

Figura 34: Iteración 3
Fuente: (Glover F. L., 2006)

En este momento, ninguno de los candidatos tiene un valor de movimiento negativo. Por lo tanto, se realiza un movimiento de no mejora. Dado que el primer movimiento de no mejora es el inverso del movimiento ejecutado en la iteración anterior, que está clasificado como tabú, este movimiento no se selecciona. Entonces se elige el intercambio de las tareas 1 y 3, como indica la **Figura 34**.

CAPÍTULO II - MARCO TEÓRICO

Si siguiendo el mismo procedimiento indicado hasta este momento, se realizarían las siguientes iteraciones mostradas en las **Figuras 35 - 36**. En estas últimas iteraciones observamos que se realizan movimientos de no mejora para escapar de la solución con valor objetivo $F = 28$, que parece ser un óptimo local.

Solución actual	Estructura tabú	Primeros 5 candidatos
4		Valor de movimiento
3	1 2 3 4 5 6	1,3 -2 T
1	2 3 2	3,4 1 *
2	3 1	1,4 2 T
6	4	2,6 2
5	5	1,2 3
	$F = 30$	

Figura 35: Iteración 4
Fuente: (Glover F. L., 2006)

Solución actual	Estructura tabú	Primeros 5 candidatos
3		Valor de movimiento
4	1 2 3 4 5 6	3,4 -1 T
1	2 1	1,3 -1 T
2	3 3	1,4 1 T
6	4	2,6 2 *
5	5	1,2 3
	$F = 31$	

Figura 36: Iteración 5
Fuente: (Glover F. L., 2006)

Si durante el proceso explicado hubiera habido algún movimiento clasificado como tabú que condujera a una solución con un retraso en la finalización de las tareas menor que el de la mejor solución encontrada hasta el momento, se podría haber usado un criterio de aspiración. En este caso, hubiéramos usado el criterio de aspiración por objetivo, que selecciona como solución actual la que tenga un menor valor objetivo, independientemente de que los movimientos requeridos para alcanzarla sean tabú.

En algunas situaciones, puede ser deseable incrementar el porcentaje de movimientos disponibles que reciben una clasificación tabú. Además, a pesar del tipo de restricción seleccionado, a menudo se obtienen mejores resultados por los plazos tabú que varían dinámicamente.

2.7. Algoritmos Genéticos

Un Algoritmo Genético (AG) es una técnica de búsqueda iterativa inspirada en los principios de selección natural. Los AG no buscan modelar la evolución biológica sino derivar estrategias de optimización. El concepto se basa en la generación de poblaciones de individuos mediante la reproducción de los padres. (Cruz, 2010)

Durante el curso de la evolución, los genes con evolución lenta fueron remplazados por genes con mejor estrategia evolutiva. Por lo tanto, se esperan estrategias altamente eficientes en la fauna y la flora modernas.

Muchos problemas tienen funciones objetivo complejas y la optimización tiende a finalizar en mínimos/máximos locales. La idea de los AG es optimizar (hallar el máximo o mínimo) una función objetivo utilizando los principios de la selección natural sobre los parámetros de la función.

2.7.1. Historia de los algoritmos genéticos

La primera idea surgió en la tesis de J. D. Bagley: “El funcionamiento de los sistemas adaptables empleando algoritmos genéticos y correlativos”, en 1967. Esta tesis influyó decisivamente en J. H. Holland, quien se puede considerar como el pionero de los AG. Los AG pueden concebirse como métodos de optimización. En general, se plantean de la siguiente manera:

$x_0 \in X$ tal que f es un máximo en x_0 , donde $f: X \rightarrow \mathfrak{R}$, por lo tanto:

$$f(x_0) = \max_{x \in X} f(x)$$

Es casi imposible obtener una solución en sentido estricto. Dependiendo del problema planteado, puede ser suficiente encontrar el máximo valor o el más cercano al valor máximo; f es una función asignada para definir el valor de “aptitud” para cada individuo (esto es, por supuesto, una gran ayuda para simplificar).

CAPÍTULO II - MARCO TEÓRICO

2.7.2. Definiciones

La siguiente lista contiene diferentes expresiones utilizadas en la genética y su estructura equivalente en AG, lo cual se puede apreciar en la **Tabla 6**.

Evolución natural	Algoritmo genético
genotipo	código de cadena
fenotipo	punto sin codificar
cromosoma	cadena
gen	posición de cadena
alelo	valor en una posición determinada
función de aptitud o aptitud	valor de la función objetivo

Tabla 4 Expresiones utilizadas en la genética

Fuente: (Cruz, 2010)

Genotipo: expresión genética de un organismo o estructura genética del organismo. La información contenida en el genoma.

Fenotipo: características físicas de un organismo, atribuibles a la expresión de su fenotipo. Contiene tanto los rasgos físicos como los conductuales. Es el resultado de la interacción entre el genotipo y el ambiente; se interpreta como la suma de los caracteres observables en un individuo. Es la manifestación externa del genotipo.

Cromosoma: es la molécula única de ADN, unida a histonas (proteínas básicas) y otras proteínas que se condensa durante la mitosis (proceso de división celular- reparto equitativo del material hereditario) y la meiosis (proceso de fragmentación —divisiones pequeñas), formando una estructura compacta.

Gen: especifica la herencia de un carácter; está formado por una secuencia de aminoácidos de una o más cadenas de ARN (ácido ribonucleico: interviene en diferentes neuronas, en la expresión de la información genética), que realizan diferentes funciones en la cadena.

Alelo: el valor de un gen. Una de las dos o más formas alternativas de un gen; determina el carácter controlado por el gen. Un ejemplo es el diploide que contiene dos juegos de cromosomas, por lo tanto tiene dos copias de cada gen.

CAPÍTULO II - MARCO TEÓRICO

Función de aptitud: es un tipo especial de función que cuantifica la optimización de una solución. Se traduce en un cromosoma óptimo para que sus bases sean combinadas con cualquier otra técnica para la producción de una nueva generación que sea mejor a las anteriores.

En términos generales, las cadenas de los AG son análogas a los cromosomas en el sistema biológico.

En los organismos naturales, uno o más cromosomas se combinan para formar la prescripción genética total, para la construcción y operación del organismo. En los organismos naturales el “paquete total” de genética se denomina genotipo. En los AG el “paquete total” de las cadenas se llama estructura (la estructura está compuesta por varias cadenas). En los organismos naturales, la creación de los organismos se realiza mediante la interacción de “paquetes” genéticos con su medio y se llama fenotipo.

En la terminología natural, se dice que los cromosomas están compuestos por genes, los cuales permiten tomar distintos valores llamados alelos. En la genética la posición del gen (llamado locus) se identifica en forma separada de la función del gen. Así, se puede hablar de un gene en particular, por ejemplo el gen que da color a los ojos de los animales: su lugar es la posición 10 y el valor del alelo es “ojos azules”.

En los AG las cadenas están compuestas por características, que toman diferentes valores localizados en distintas posiciones de la cadena. También fueron desarrollados por John Holland y sus colegas en la Universidad de Michigan (1975). Estos algoritmos se basan en la mecánica de la selección natural, la cual afirma que sólo los organismos que mejor se adaptan sobreviven. Parte de la historia de los algoritmos se describió en los antecedentes de la IA, de manera que ahora se abordarán otros aspectos importantes para entender mejor esta parte de la ciencia.

- Los componentes de un algoritmo genético son:
- Una función que se desea optimizar.
- Un grupo de candidatos para la solución.
- Una función de evaluación que mida cómo los candidatos optimizan la función.
- Función de reproducción.

CAPÍTULO II - MARCO TEÓRICO

El AG funciona porque hay competencia por los recursos y se hereda la mejor configuración genética en cada generación. Ciertas características de un organismo de la misma especie, en las que el éxito se manifiesta como éxito reproductivo, lo convierten en el mejor adaptado, por lo que deja más hijos que los otros. De esta manera, los rasgos hereditarios que favorecen el éxito tienden a estar más representados en la población.

2.7.3. Herencia

Desde tiempos inmemoriales, el hombre se ha percatado de la herencia de los rasgos físicos o psicológicos.

En general, los hijos se parecen a los padres, y que esto no sólo ocurre en el hombre, sino también en los demás organismos; ya los filósofos griegos de la Antigüedad meditaban sobre estos hechos. Pero el desarrollo de la genética no se inició sino hasta el comienzo del siglo XX, época en que los científicos conocían ya numerosos detalles de la constitución de la célula. Antes, en el año de 1860, tuvieron lugar los famosos experimentos del monje austriaco Gregor Mendel, quien cruzó diversas plantas de jardín de su convento y pudo establecer algunas de las leyes más importantes de la genética. Sin embargo, la importancia de la obra de Mendel no fue reconocida hasta que los resultados obtenidos por éste no se compararon con los que el estadounidense Thomas Morgan realizó con la mosca de fruta. La genética, es decir, la ciencia de la herencia, junto con la investigación de la biología celular, ha podido establecer que los genes, portadores de los rasgos hereditarios, se encuentran en los cromosomas del núcleo celular (Cruz, 2010).

Incluso se ha mostrado que las cualidades dependientes de dichos genes se heredan según ciertas leyes derivadas de las posibles combinaciones de los genes de las células sexuales. Hoy se sabe que la información sobre la herencia está almacenada en una especie de “código” en los ácidos nucleicos (ADN y ARN). La moderna genética dedica gran atención al estudio de estos ácidos. Se ha demostrado que aunque sean muy estables, pueden ser transformados mediante ciertos productos químicos, rayos X, etc., de tal manera que al transmitirse a las células descendientes, provocan en éstas la aparición de nuevas cualidades. Estas modificaciones, positivas o negativas, se llaman “mutaciones” y pueden surgir también en forma espontánea.

CAPÍTULO II - MARCO TEÓRICO

¿Qué es herencia?

En la reproducción sexual los genes se combinan de diferentes maneras. Los cromosomas paternos se reparten entre las células sexuales, y cuando tiene lugar la fecundación surgen combinaciones de factores hereditarios diferentes de los que poseen los padres. A través de constantes combinaciones, la masa hereditaria se transmite por medio de las células sexuales, de generación en generación; podría decirse que es potencialmente inmortal.

En las plantas y animales superiores, los cromosomas de las células normales se agrupan por parejas.

En cada pareja, uno de los cromosomas procede del padre y el otro de la madre. Al formarse las células sexuales, los dos cromosomas de cada pareja van a parar a células distintas, de modo que los gametos sólo reciben la mitad de los cromosomas contenidos en una célula normal. En el momento de la fecundación, cuando los gametos se fusionan y reúnen sus cromosomas, el número de éstos vuelve a ser el de una célula normal. En los seres humanos, este número es 46 (23 parejas).

Al juntarse las dos células sexuales, cada cromosoma se reúne con su pareja y cada gen con su homólogo del otro cromosoma. Los genes así emparejados no siempre son iguales. Uno puede ser dominante, impidiendo que se manifieste el otro, que se llama recesivo. Se debe considerar que no se heredan directamente las cualidades, sino sólo los genes que las determinan; la aparición de un rasgo depende de la interacción entre los genes y la influencia del medio. El conjunto de los genes o predisposiciones hereditarias de un individuo es su genotipo, mientras que el conjunto de sus caracteres externos, es decir, el resultado de la interacción del genotipo con el medio, es el fenotipo. Las cualidades adquiridas sólo por la influencia del medio no pueden heredarse.

El código genético El material genético está formado por ácidos nucleicos. Éstos se hallan en todas las células vivas, en las que determinan la constitución de las proteínas y de los genes. Se distinguen dos tipos de ácidos nucleicos: ácido desoxirribonucleico (ADN) y ácido ribonucleico (ARN). Ambos están formados por fosfatos, azúcares y bases nitrogenadas. Alrededor de 1940 se consiguió demostrar que la función de los cromosomas, como portadores de los factores hereditarios, corresponde a los ácidos nucleicos. En 1962 se otorgó el premio Nobel por el descubrimiento de la estructura y función de la molécula de ADN. Ésta tiene forma de hélice doble.

CAPÍTULO II - MARCO TEÓRICO

Las dos hélices de la molécula están unidas mediante bases nitrogenadas enlazadas por puentes de hidrógeno. Cada unión está integrada por dos bases, formando cuatro tipos de uniones cuya ordenación en la molécula constituye el código genético. Éste debe ser transmitido, con precisión, de célula a célula, en sucesivas generaciones. Dicha transmisión tiene lugar cuando, al dividirse la célula, las hélices de la molécula de ADN se separan y cada una sintetiza de nuevo su cadena complementaria hasta formar otra vez hélices dobles que luego se reparten entre las células hijas.

Las moléculas de ARN, homólogas a las del ADN, se forman en el núcleo celular. Después se trasladan a los centros de elaboración de proteínas, los ribosomas del citoplasma, donde se dirige la formación de las diferentes clases de proteínas. Hoy el “código genético” se ha descifrado. Se conoce con mucha exactitud el mecanismo por el cual los ácidos nucleicos (núcleos) lo transportan a lugares de síntesis para producirlo en proteínas (citoplasmas). A medida que aumente el conocimiento humano sobre la materia viva, se podrá comprender mejor cómo surgió vida.

2.7.4. Selección natural

Mientras reflexionaba acerca de la selección artificial, Darwin leyó un ensayo del economista Thomas Malthus. El ensayo sugería que de no controlarse la población humana, sobrepasaría con el tiempo a la producción de alimentos, ocasionando una fuerte competencia por la existencia. Darwin se dio cuenta de que las ideas de Malthus podían aplicarse al mundo natural. Razonó que algunos competidores en la lucha por la existencia estarían mejor equipados para sobrevivir que otros. Los menos equipados morirían.

Allí estaba, finalmente, el marco de trabajo para una nueva teoría sobre el origen de las especies. La teoría de Darwin tiene cuatro principios básicos que explican cómo pueden cambiar los rasgos de una población con el tiempo. Primero, los individuos de una población muestran diferencias o variaciones.

Segundo, las variaciones pueden heredarse, lo cual significa que pasan de padre a hijo. Tercero, los organismos tienen más descendientes de los que pueden sobrevivir con los recursos disponibles. El cardenal promedio, por ejemplo, pone nueve huevos cada verano. Si cada pichón de cardenal sobrevive y reproduce una sola vez, tomaría sólo siete años para que produjera un millón de aves.

CAPÍTULO II - MARCO TEÓRICO

Finalmente, las variaciones que aumentan el éxito reproductivo tendrán mayor oportunidad de transmitirse, que aquellas que no lo aumentan. Si tener una cola de abanico ayuda a las palomas a reproducirse exitosamente, futuras generaciones incluirán más palomas con este tipo de cola.

Darwin llamó a su teoría selección natural. Razonó que, dado el tiempo necesario, la selección natural podría modificar una población lo suficiente como para producir nuevas especies. También es necesario conocer la transición de una generación a otra, constan de cuatro elementos básicos:

Selección: mecanismo de selección individual (cadena) para la reproducción acorde con la función de aptitud (valor de la función objetivo). Los algoritmos de selección serán los encargados de escoger qué individuos van a disponer de oportunidades de reproducirse y cuáles no. La idea básica de selección está asociada con la función de aptitud y el sistema original; para su implementación es comúnmente conocida como roulette-wheel (RWS); ésta utiliza una distribución de probabilidad, donde la probabilidad de selección de una cadena es directamente proporcional a su aptitud.

Cruzamiento: método de fusión sobre la información genética de dos individuos; si la codificación es elegida apropiadamente, dos progenitores saludables producirán descendientes sanos. Es el principal operador genético; provee un mecanismo para heredar características a su descendencia; interviene en ambos progenitores.

Mutación: en la evolución real, el material genético puede ser alterado en forma aleatoria debido a un error en la reproducción o la deformación de genes.

En los algoritmos genéticos, la mutación se realiza, con gran probabilidad, como una deformación aleatoria de las cadenas. Produce cambios incrementales al azar en la descendencia, efectuando cambios aleatorios en los valores del alelo en algunos genes. En el caso de cromosomas binarios le corresponde hacer los cambios de posiciones en cada bit. No afecta a toda la población, pero es probable que dañe a algunos. La mutación tiene el efecto de perturbar de manera segura a los cromosomas a fin de introducir nuevas características que no estaban presentes en ningún elemento de los progenitores.

Reemplazo: procedimiento para calcular (crear) una nueva generación de la anterior y sus descendientes.

CAPÍTULO II - MARCO TEÓRICO

Se crea un espacio a la descendencia en la población eliminando de ella a los padres. Los algoritmos genéticos trabajan con un número fijo de cadenas binarias de longitud fija (*Ecuación 2*). Para este fin, se asumirá que las cadenas que se va a considerar son del mismo conjunto.

$$S = \{0,1\}^n;$$

Ecuación 2: Cadena binaria de longitud fija
Fuente: (Cruz, 2010)

Por lo tanto, la generación en el tiempo t es una lista de m cadenas donde se denotará como en la *Ecuación 3*.

$$B_t = (b_{1,t}, b_{2,t}, \dots, b_{m,t}).$$

Ecuación 3: Cadenas de tiempo
Fuente: (Cruz, 2010)

Los AG, en su mayoría, tienen la estructura del *Código 2*.

```
Algoritmo
Para  $t := 0$ 
Se computa la población inicial definida por  $B_0 = (b_{1,0}, b_{2,0}, \dots, b_{m,0})$ ;
Mientras WHILE se alcanza la condición deseada DO
Inicio
FOR  $i := 1$  hasta  $m$  DO
Se selecciona un individuo  $b_{i,t+1}$  de  $B_t$ ;
FOR  $i := 1$  hasta  $m-1$  Paso 2 DO
IF Random  $[0, 1]^{pc}$  THEN
cruza  $b_{i,t+1}$  con  $b_{i+1,t+1}$ ;
FOR  $i := 1$  hasta  $m$  DO
eventualmente mutar
se crea la descendencia cruzando a los individuos;
eventualmente se mutan  $b_{i,t+1}$ ;
se realiza un incremento  $t := t + 1$ 
FIN
```

Código 2: Estructura de un AG
Fuente: (Cruz, 2010)

La selección, el cruzamiento (se realiza sólo con una probabilidad crítica de percolación pc) y la mutación tienen ciertos grados de libertad, mientras que la operación de reemplazo ha sido especificada.

Como es fácil de apreciar, todo individuo seleccionado se reemplaza por su sucesor después del cruzamiento y la mutación; los individuos no seleccionados mueren inmediatamente.

2.7.5. Operaciones genéticas en cadenas binarias

Selección

La selección es la componente que guía el algoritmo para encontrar la solución, prefiriendo dentro de un grupo de baja función de aptitud a los más altos. Puede ocuparse una operación determinista; en la mayoría de las implementaciones tiene componentes aleatorios. La probabilidad de escoger el individuo adecuado es directamente proporcional a su función de aptitud. Se puede observar como un experimento aleatorio con la *Ecuación 4*.

$$P[b_{j,t} \text{ es seleccionada}] = \frac{f(b_{j,t})}{\sum_{k=1} f(b_{k,t})}$$

Ecuación 4; Selección dentro de un algoritmo

Fuente: (Cruz, 2010)

Funciona solo para los valores positivos de las funciones de aptitudes. Si éste no es el caso, se debe aplicar (un cambio en el caso más simple) una transformación de no-decremento $\phi : \mathfrak{R} \rightarrow \mathfrak{R}^+$. Entonces la probabilidad se puede expresar por la *Ecuación 5*.

$$P[b_{j,t} \text{ es seleccionada}] = \frac{\phi(f(b_{j,t}))}{\sum_{k=1} \phi(f(b_{k,t}))}$$

Ecuación 5: Diferentes salidas con probabilidades

Fuente: (Cruz, 2010)

Las diferentes salidas se obtendrán con diferentes probabilidades. La programación del algoritmo se puede proponer como sigue, y la configuración analógica se representa con la expresión matemática anterior.

```
Algoritmo
x: = Random[0,1];
i: = 1
Mientras i < m & x <  $\sum_{j=1}^i f(b_{j\mu}) / \sum_{j=1}^m f(b_{j\mu})$  hacer
i: i+1;
selecciona  $b_{i\mu}$ ;
```

Código 3: Diferentes salidas con probabilidades
Fuente: (Cruz, 2010)

Este método a menudo se denomina selección proporcional (**Código 3**), no es el mejor para la convergencia rápida pero puede funcionar en problemas de optimización básicos.

2.7.6. Cruzamiento

En la reproducción sexual como se desempeña en el mundo real, la materia genética de los progenitores se mezcla cuando los gametos de los progenitores se fusionan. Por lo general los cromosomas son aleatoriamente divididos y fusionados, con la consecuencia de que algunos genes de los descendientes provienen de un progenitor, mientras que otros provienen del otro progenitor. Este mecanismo se llama cruzamiento. Es una herramienta muy potente para introducir nuevos materiales genéticos y mantener la diversidad genética, pero con la notable propiedad de que progenitores saludables también producen buen rendimiento en los descendientes, o incluso mejores. Varias investigaciones han llegado a la conclusión que el cruzamiento es la razón por la que las especies de reproducción sexual se adaptan más rápido que las de reproducción asexual.

Básicamente, el cruzamiento es el intercambio de genes entre los cromosomas de los dos progenitores.

En un caso sencillo, se puede realizar este proceso cortando dos cadenas en una posición elegida al azar e intercambiarlas en sus extremos. Este proceso, que se nombra cruzamiento de único-punto, se aprecia en la **Figura 37**.

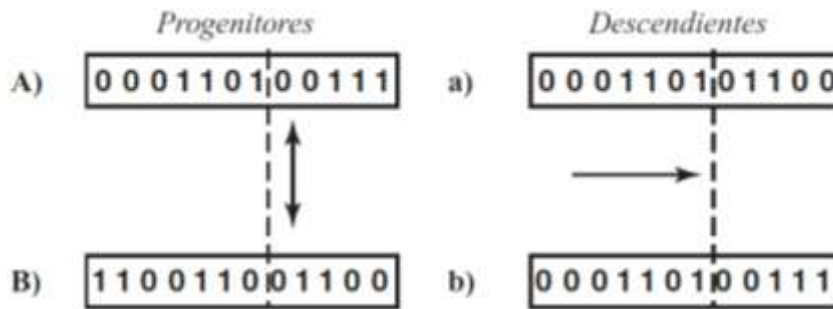


Figura 37: Cruzamiento de unico-punto
Fuente: (Cruz, 2010)

Se muestra la programación para el cruzamiento en el **Código 4**.

```

Algoritmo
  Seleccionar de forma aleatoria
  pos := Random{1, ..., n-1};
  FOR i := 1 hasta pos DO
  Inicio
  Child1[i] := Parent1[i];
  Child2[i] := Parent2[i]
  Fin
  FOR i := pos+1 TO n DO
  Inicio
  Child1[i] := Parent2[i];
  Child2[i] := Parent1[i]
  Fin

```

Código 4: Cruzamiento de único-punto
Fuente: (Cruz, 2010)

Para otros problemas o diferentes codificaciones, otros métodos de cruzamiento pueden ser útiles o incluso necesarios. Se mencionan algunos de ellos a continuación.

Cruzamiento de N-puntos: en lugar de un único punto, se eligen al azar N puntos de ruptura. Cada segunda sección se intercambia. Entre estas clases, la de dos-puntos es particularmente importante.

Cruzamiento segmentado: esta técnica es muy parecida al cruzamiento de N-puntos, con la diferencia de que el número de puntos de ruptura puede variar.

Cruzamiento uniforme: para cada posición, se decide al azar si se intercambian las posiciones.

Cruzamiento aleatorio: primero se escoge una permutación aleatoria que se aplicará a los progenitores, después el cruzamiento de N-puntos se aplica a los progenitores aleatorios, y finalmente los descendientes aleatorios son transformados de nuevo con permutación inversa.

2.7.7. Mutación

El último ingrediente de los AG es la mutación, esto es la deformación aleatoria de la información genética en un individuo como las radiaciones radioactivas u otros medios de influencia. En la reproducción real, la probabilidad de que determinado gen sea mutado es casi igual para todos los genes. Así, está al alcance de la mano usar las siguientes técnicas de mutación para una determinada cadena binaria S , donde P_m es la probabilidad de que un solo gen sea modificado (*Código 5*).

```
Algoritmo
FOR i:=1 TO n DO
  IF Random [0,1] <  $p_M$  THEN
    invert s[i];
```

Código 5: Mutación en un individuo

Fuente: (Cruz, 2010)

Por supuesto que se pueden encontrar muchas alternativas y con más detalles. Algunas de estas técnicas se muestran a continuación.

Inversión de un solo bit: la probabilidad de mutación p_M - bit elegido al azar sea negado.

Inversión por fragmentos: toda la cadena es invertida con probabilidad de mutación P_m .

Selección aleatoria: la probabilidad de mutación p_M de que una cadena elegida al azar sea remplazada.

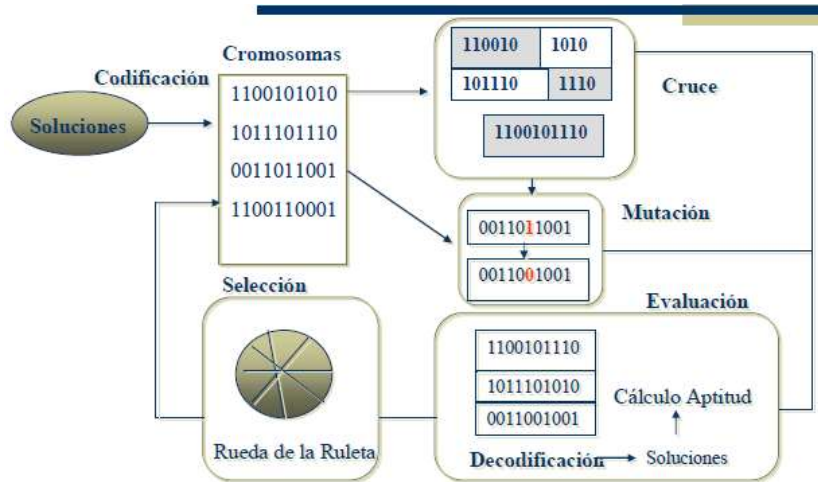


Figura 38: Esquema de un algoritmo genético simple
 Fuente: (Baver_R., 1994)

El algoritmo visto en la **Figura 38** ha servido para que los estudios realizados en torno a él, se hayan aplicado a diversos problemas y modelos en ingeniería, y en la ciencia en general. (Baver_R., 1994). Se destaca por:

- **Optimización:** Se trata de un campo especialmente abonado para el uso de los Algoritmos Genéticos, por las características intrínsecas de estos problemas. Se han utilizado en numerosas tareas de optimización, incluyendo la optimización numérica, y los problemas de optimización combinatoria.
- **Programación automática:** Los Algoritmos Genéticos se han empleado para desarrollar programas para tareas específicas de forma automática y para diseñar otras estructuras computacionales tales como el celular inteligente de la actualidad, y las redes de clasificación.
- **Aprendizaje máquina:** Los algoritmos genéticos utilizados en diversas aplicaciones y para desarrollar determinados aspectos de sistemas particulares de aprendizaje, como pueda ser el de los pesos en una red neuronal, las reglas para sistemas de clasificación de aprendizaje o sistemas de producción simbólica, y los sensores para robots.
- **Economía:** En este caso, se ha hecho uso de estos Algoritmos para modelar procesos de innovación, el desarrollo estrategias de puja, y la aparición de mercados económicos. Las cuales favorecen y orientan en ámbitos financieros y económicos.

CAPÍTULO II - MARCO TEÓRICO

- **Sistemas inmunes:** Los nuevos sistemas son inmunes naturales, incluyendo la mutación somática durante la vida de un individuo y el descubrimiento de familias de genes múltiples en tiempo evolutivo, ha resultado útil el empleo de esta técnica.
- **Ecología:** Coadyuvaron y orientaron en la modelización de fenómenos ecológicos tales como las carreras de armamento biológico, la coevolución de parásito-huesped, la simbiosis, y el flujo de recursos.
- **Genética de poblaciones:** En el estudio de preguntas del tipo “¿Bajo qué condiciones será viable evolutivamente un gene para la recombinación?”. Este estudio descriptivo ayuda y orienta en la preservación de los genes.
- **Evolución y aprendizaje:** Los Algoritmos Genéticos se han utilizado en el estudio de las relaciones entre el aprendizaje individual y la evolución de la especie. Con lo cual le da la capacidad de aprendizaje y sus distintos tipos de evoluciones dependiendo de su entorno.
- **Sistemas sociales:** En el estudio de aspectos evolutivos de los sistemas sociales, tales como la evolución del comportamiento social en colonias de insectos, y la evolución de la cooperación y la comunicación en sistemas multi-agentes.

Los índices descritos anteriormente no son exhaustivas pero sí transmiten la idea de la variedad de aplicaciones que tienen los Algoritmos Genéticos. Gracias al éxito en estas y otras áreas, los Algoritmos Genéticos han llegado a ser un campo puntero en la investigación actual.

EJEMPLO

Encontrar el máximo de la función de la *Ecuación 6*.

$$f_1 : \{0, \dots, 31\} \rightarrow \mathfrak{R}$$
$$x \rightarrow x^2$$

Ecuación 6: Funcion de ejemplo
Fuente: (Cruz, 2010)

Para especificar una cadena con el espacio a lo largo de un sistema de codificación y decodificación. En este ejemplo está al alcance de la mano considerar $S = \{0,1\}^5$, donde el valor de $\{0, \dots, 31\}$ se codifica por su representación binaria. De manera correspondiente, una cadena se decodifica como en la *Ecuación 7*.

$$\tilde{c}(s) = \sum_{i=0}^4 s[4 - i] \cdot 2^i.$$

Ecuación 7: Probabilidad de cruzamiento

Fuente: (Cruz, 2010)

Con una población de $m = 4$, una probabilidad de cruzamiento $pc = 1$ y una probabilidad de mutación $p_M = 0.001$, la primera generación al azar con distribución uniforme de $\{0,1\}^5$, se obtendrá lo siguiente en el primer paso de la **Tabla 7**.

Núm. del individuo	cadena (genotipo)	valor x (fenotipo)	$f(x) \cdot x^2$	P seleccionada, $f_i / \sum f_i$
1	01101	13	169	0.14
2	11000	24	576	0.49
3	01000	8	64	0.06
4	10011	19	361	0.31

Tabla 5: Primera generación del cruzamiento

Fuente: (Cruz, 2010)

La suma de la función de aptitud es 1170, donde el promedio es 293 y el máximo es 576. Se puede apreciar que en las últimas columnas la selección proporcional favorece a los de alta-aptitud (como el núm. 2) sobre los de baja-aptitud (como el núm. 3).

Realizando un experimento aleatorio se ha dejado morir al individuo núm. 3 y seleccionando doble vez al núm. 2, con lo que se obtuvo una segunda generación como sigue en la **Tabla 8**.

Conjunto de individuos seleccionados	Lugar de cruzamiento (aleatorio)	Nueva población	valor x	$f(x) \cdot x^2$
0110 1	(1) 4	01101	12	144
1100 0	(2) 4	11001	25	625
11 000	(2) 2	11011	27	729
10 011	(4) 2	10000	16	256

Tabla 6: Segunda generación del cruzamiento

Fuente: (Cruz, 2010)

2.7.8. Algoritmos Genéticos y TABU con mecanismos de búsqueda local

Algoritmos Genéticos y TABU con mecanismos de búsqueda local, se puede apreciar en la **Figura 55** donde se muestra el flujo que se tiene la cual nos ayuda a obtener resultados óptimos de búsqueda utilizando principalmente el esquema de Algoritmos Genéticos que se describe a continuación:

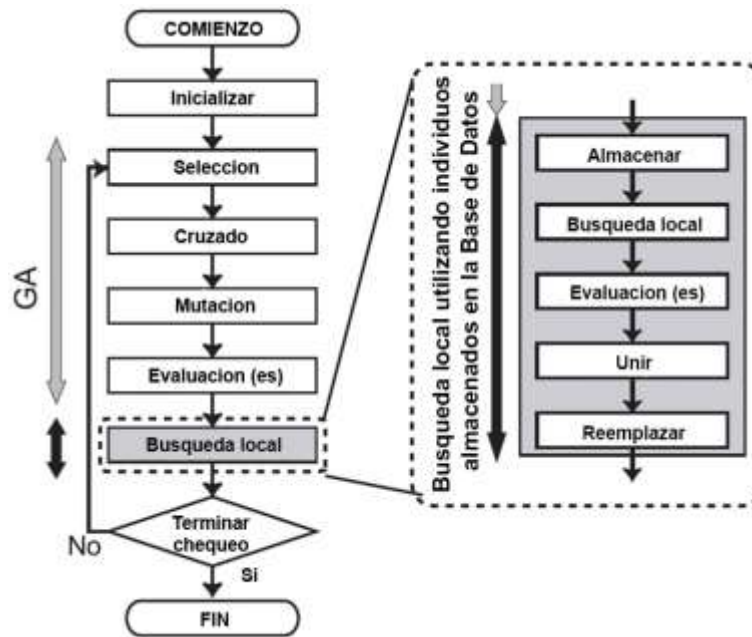


Figura 39: Flujo del método Propuesto
Fuente: (Miki, 2014) - Traducción e interpretación propia

Los pasos son los siguientes:

Paso 1. Generar individuos N_{pop} al azar, donde es N_{pop} el tamaño de la población y se almacenan en la base de datos.

Paso 2. Aplicar los operadores tales como cruces, mutaciones y selecciones a los individuos en una población de Algoritmos Genéticos.

Paso 3. Tienda el y_{mejor} individual, que es el mejor individuo de una población de Algoritmos Genéticos, en la base de datos y se establece en su cadena de bits de la que los valores de todos son '0'; Sin embargo, y_{mejor} no se almacena cuando se incluye en las regiones buscadas, que se indican por individuos que ya han sido almacenados en la base de datos.

Paso 4. Búsqueda local expande una región explorada indicado por un determinado individuo almacenado en la base de datos. Cuando un mejor individuo se encuentra, vuelva a colocar el peor individuo de la población de Algoritmo Genético con él.

Paso 5. Vuelva al paso 2 hasta que algunas condiciones de terminación, por ejemplo costo de la informática llega a una cantidad limitada o la búsqueda exhaustiva se realiza y están satisfechos.

En el paso 3, cuando hay N_{DB} individuos en la base de datos, sustituya el x_{peor} individual que tiene el peor aptitud en la base de datos con y_{mejor} si el estado físico de y_{mejor} es más grande que la de x_{peor} , de lo contrario y_{mejor} no se almacena, donde N_{DB} es el parámetro de capacidad de la base de datos.

Búsqueda local

Se elige un determinado individuo de la base de datos para aplicar la búsqueda local en cada generación. Este individuo satisface la condición de que el índice es el valor mínimo y su condición física es el valor máximo.

Los valores de los cuales se sitúan en '0' de la cadena bits de la persona seleccionada, se convierte en un valor de '1'. El lugar, que ocupa la mayor varianza de genes en conjunto, se selecciona.

Esta operación se realiza debido a que es adecuado para mantener el índice entre los individuos almacenados en la base de datos para el proceso de los individuos de la concentración; Por otra parte, es deseable retener genes que tienen las varianzas inferiores ya que pueden ser consideradas como parte de una solución mejor.

Dada N individuos x_i ($1 \leq i \leq N$) almacenados en la base de datos, el proceso de la búsqueda local se describe como sigue. La **Figura 56** muestra el ejemplo de la búsqueda local.

	Cromosoma	Mascara	Region explorada	Aptitud	Indice
Individual x ₁	1 0 1 1 1 1	1 0 1 0 1 0	* 0 * 1 * 1	5	3
Individual x ₂	1 1 0 1 1 1 ↙ actualizar 1 1 1 1 1 1	1 0 0 0 1 0 ↙ actualizar 1 0 1 0 1 0	* 1 0 1 * 1 ↙ actualizar * 1 * 1 * 1	5 ↙ actualizar 6	2 ↙ actualizar 3
Individual x ₃	0 1 0 1 1 0	0 0 0 1 1 0	0 1 0 * * 0	3	2
Individual x ₄	0 1 1 1 1 0	0 0 1 1 1 0	0 1 * * * 0	4	3

* = 0 or 1

Figura 40 Ejemplo de búsqueda local
 Fuente: (Miki, 2014) - Traducción e interpretación propia

Paso 1. Buscar los resultados de otros utilizando la **Ecuación 8**, la cual es el valor absoluto de la diferencia entre el valor medio de los genes y 0.5 en cada locus.

$$a_l = \left| \frac{1}{N} \sum_{i=1}^N (c_{x_i, l}) - 0.5 \right| \quad (1 \leq l \leq L)$$

Ecuación 8: Valor absoluto
 Fuente: (Miki, 2014)

Paso 2. Seleccione el individuo x que tiene el índice mínimo y la condición física máxima, a partir de \mathcal{N} individuos almacenados en la base de datos.

Paso 3. Seleccione el locus l^* , lo que indica $m_{x, l^*} = 0$. Al mismo tiempo, a_{l^*} es mínimo, del $a_l (1 \leq l \leq L)$.

Paso 4. Preparar la x' individual, que tiene la misma cadena como la de x . En su cromosoma, cada gen es exactamente el mismo que el de x en loci de pie en '1', excepto l^* en su cadena. x es el mejor individuo en virtud de la búsqueda \mathcal{X} , que se debe buscar para expandir la región de búsqueda.

Paso 5. Actualización m_{x', l^*} a '1' y h_x a $h_x + 1$; Por otra parte, sea x ser x' reemplazar el peor individuo de la población de GA por x si $x < x'$. La búsqueda exhaustiva se termina cuando se acerca h_x enfoques \mathcal{L} .

CAPÍTULO II - MARCO TEÓRICO

En el paso 4, se ha descrito anteriormente, \mathcal{X} se compone de individuos producidos por voltear el gen en l^* en el cromosoma de cada individuo incluido en \mathcal{X} . Esto tiene que buscar 2^{lx} individuos donde el índice de x es h_x ; sin embargo se puede aplicar a buscar fácilmente \mathcal{X} .

Este mecanismo utiliza los recursos informáticos de manera efectiva ya que los incrementos espaciales buscados linealmente como el número de aumento de los recursos de una búsqueda exhaustiva se garantiza en los cálculos infinitos.

Operación de Unión Base de Datos

Después de la búsqueda local, una combinación de individuos almacenados en la base de datos se ejecuta en cada generación para evitar la superposición de ping. Se pueden combinar cuando se cumplan las siguientes condiciones:

1. Condición x_a y x_b se dan los individuos. Cuando ellos satisfacen las siguientes condiciones. Tienen las mismas cadenas, $d(x_a, x_b) = 1$, y el locus es l^* , que satisface $M_{x_a} l^* = m_{x_b} l^* = 0$ y $c_{x_a} l^* = l c_{x_b}^*$. En este caso, seleccione el que tiene el mejor estado físico de x_a y x_b , y dejar que su valor de l^* en su cadena de '1'. Al mismo tiempo, el otro se elimina de la base de datos. Por ejemplo, en la **Figura 60**, el individuo y el x_1 x_2 individuo actualizada se pueden combinar con la condición 1.

2. Condición x_a y x_b se dan los individuos. Cuando los individuos satisfacen las condiciones siguientes, que están en condición 2: $d(x_a, x_b) = 0$ y no existe un locus que satisface $M_{x_a} l = 1, m_{x_b} l = 0$ ($1 \leq l \leq L$). En este caso, se elimina x_a de la base de datos debido $X_a \subset X_b$. Por ejemplo, en la **Figura 60**, el individuo y el x_3 x_4 individuales se pueden combinar reunión condición 2. Por lo tanto, se elimina x_3 .

3. Condición x_a y x_b se dan los individuos. Cuando $X_a \cap X_b = \emptyset$ y $|X_a| \geq |X_b|$, que están en la condición 3 se muestra en la Figura 5. En este caso, X_a se expande hasta que se pueda incluir X_b , y luego se fusionan con la condición 2. x indica la región que se requiere para expandir hasta que puede incluir X_b .

$X_a \cap \neg X_b$ debe ser buscado para combinar el uso de la condición 2. Se introduce el parámetro A , que indica la relación entre $|X_a \cap X_b|$ de $|X_a \cap \neg X_b|$, es decir $|X_a \cap X_b| / |X_a \cap \neg X_b|$, porque tenemos una mejor solución bajo la computación limitando la disponibilidad de recursos.

Esta mezcla no se aplica cuando A es mayor que un cierto valor. Para decidir una \mathcal{A} adecuada, el tamaño del espacio de búsqueda y el número de recursos informáticos disponibles o costos deben ser considerados. En el presente trabajo de investigación se creó una a ocho.

A. Representación de la búsqueda en una región

En esta Tesis se introduce una base de datos específica de Algoritmos Genéticos que posee información de la región que ya ha sido objeto de búsqueda. Utilizando individuos codificados en binario. Además de los cromosomas, los individuos almacenados en la base de datos poseen cadenas de bits, lo que llamamos maskstrings.

Maskstrings también son cadenas de bits, longitudes de las cuales son las mismas que las de los cromosomas. Un locus de un cromosoma, que se sitúa en '1' en un maskstring, representa que ya ha buscado todos los genes asignables al mismo. La figura 1 muestra que un conjunto de individuos buscados se comprime a un individuo utilizando un maskstring.

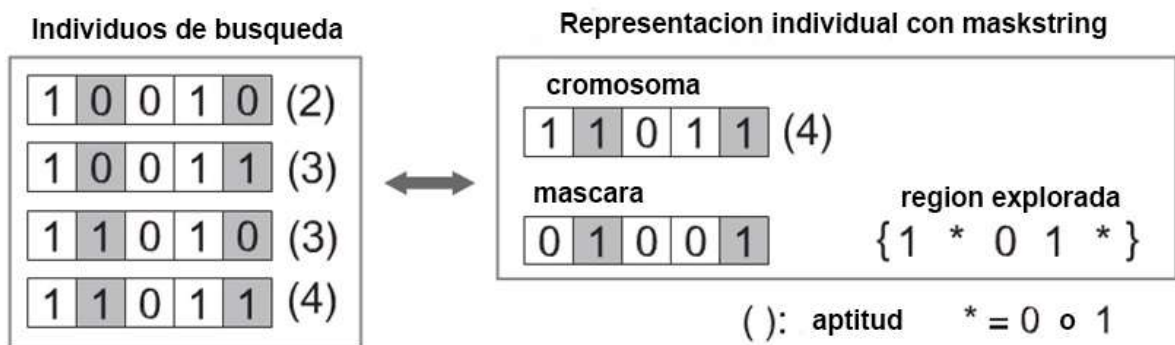


Figura 41: Ejemplo de la base de datos - almacenamiento individual
 Fuente: (Miki, 2014) - Interpretación propia

Nuestra base de datos propuesta almacena estos individuos que tienen maskstrings. La **Figura 57** muestra ejemplo de la base de datos. La x_3 individual, almacenada en la base de datos se muestra en la **Figura 58** implica que "0 1 1 1 1" es la mejor solución en la búsqueda del conjunto de individuos $X_3 = \{0 * * 1 *\}$ ($* = 0$ o 1). Llamamos a una serie '1' en un maskstring la del índice, por ejemplo el índice de x_3 es 3.

	cromosoma	mascara	region explorada	aptitud	indice
Individuo x_1	1 0 1 1 1	1 0 0 0 1	* 0 1 1 *	4	2
Individuo x_2	1 0 0 1 1	0 0 0 1 1	1 0 0 * *	3	2
Individuo x_3	0 1 1 1 1	0 1 1 0 1	0 * * 1 *	4	3
Individuo x_4	1 1 0 0 1	0 0 0 0 0	1 1 0 0 1	3	0

* = 0 o 1

Figura 42: Aspecto de la base de datos del metodo propuesto
 Fuente: (Miki, 2014) - Interpretación propia

Cuando una base de datos almacena toda la información individual, tiene un gran tiempo para comprobar un individuo que ya se ha buscado debido a sus grandes cantidades de datos. Nuestra notación propuesta de las personas buscadas es el método de compresión de alta usando maskstrings y busca regiones que están representadas por varios; Por otra parte, la comprobación individuos almacenados en la base de datos no es mucho tiempo de trabajo.

B. Medidas cuantitativas de búsquedas en una región

Esta notación de los individuos nos permite proporcionar una tasa cuantitativa de una región explorada durante una búsqueda. En un individuo x denotamos su longitud cromosoma por \mathcal{L} , un gen del locus l de un cromosoma por c_{xl} , y un valor del locus l de un maskstring por m_{xl} .

Caso de un individuo. $|X|$ denota una serie de elementos involucrado en un conjunto X . El tamaño de una región de búsqueda se indica por una persona de la que el índice de pie en h es 2^h , por ejemplo $|X_3|$, que es el tamaño de la región de búsqueda se indica por el x_3 individuo almacenada en la base de datos se muestra en la **Figura 62**, como 2^3 .

Caso N individuos. Dada N individuos $x_i(1 \leq i \leq N)$ y sus regiones de búsqueda X_i , la región totales buscado indicado por ellos es la unión de los conjuntos $|\cup_{i \in I} X_i|$, donde el conjunto que consiste en el sufijo i se denota $I = \{1, 2, \dots, N\}$. En la mayoría casos, no puede fácilmente ser derivados como una función de un número de elementos en una unión de conjuntos. Por otra parte, el de una intersección de conjuntos es una expresión cerrada. El

tamaño de la región de búsqueda es un caso tal. Debido a esto, $|\cup_{i \in I} X_i|$ región se busca un caso así. Debido a esto $|\cap_{j \in J} X_j|$ donde J es set de I que se muestra en la **Ecuación 9**.

$$|\cup_{i \in I} X_i| = \sum_{J \subseteq I, J \neq \emptyset} (-1)^{|J|-1} |\cap_{j \in J} X_j|$$

Ecuación 9: Caso N individuos - j es parte de I
Fuente: (Miki, 2014) - *Interpretación propia*

La distancia entre los individuos x_i y x_j la máscara, que se representa como $d(x_i, x_j)$, se define en la **Ecuación 10**.

$$d(x_i, x_j) = \sum_{l=1}^L B |c_{x_i l} - c_{x_j l}|$$

$$B = \begin{cases} 1, & \text{if } m_{x_i l} = m_{x_j l} = 0 \\ 0, & \text{otherwise} \end{cases}$$

Ecuación 10: Distancia entre individuos
Fuente: (Miki, 2014) - *Interpretación propia*

$|\cap_{i \in I} X_i|$ es una expresión cerrada y derivado a continuación como la **Ecuación 11**, donde $\lfloor \mathcal{R} \rfloor = z$ dado número real \mathcal{R} y número entero z .

$$|X_1 \cap X_2 \cap \dots \cap X_N| = \begin{cases} 2^M, & \text{if } K = 0 \\ 0, & \text{otherwise} \end{cases}$$

$$K = \sum_{i=1}^{N-1} \sum_{j=(i+1)}^N d(x_i, x_j)$$

$$M = \sum_{l=1}^L \left[\frac{1}{N} \sum_{i=1}^N m_{x_i l} \right]$$

Ecuación 11: Expresión cerrada
Fuente: (Miki, 2014) - *Interpretación propia*

CAPÍTULO III

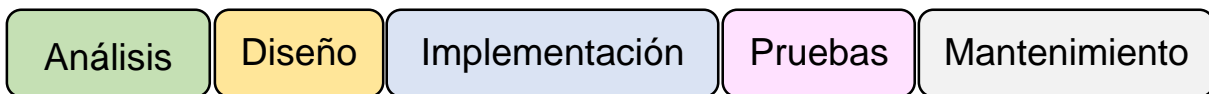
MARCO APLICATIVO

En este capítulo describe la aplicación de las metodologías propuestas para la construcción de la APK para discapacitados de la vista, la construcción de la búsqueda, mutación y posicionamiento de caracteres. Seguidamente se modela y desarrolla un sistema móvil de comunicación asíncrona empleando metodologías híbridas.

3. Introducción

El Sistema Braille se constituye como un sistema de comunicación didáctica el cual permite comunicarse y difundir su conocimiento, sin embargo, es difícil de acceder y comprender la información almacenada.

La presente Tesis plantea la construcción de una aplicación APK a partir de un modelo desarrollado con Algoritmos Genéticos, búsqueda Tabú para efectuar la selección de caracteres multitáctil, diseñando con la metodología Híbrida se utiliza el proceso y método de la ingeniería de software orientado a objetos, basándose en el Lenguaje de Modelado Unificado como herramienta de apoyo en el modelado del Análisis y Diseño partiendo del ciclo de vida clásico en un proceso de desarrollo de software como se puede ver en la **Figura 43**.



*Figura 43: Ciclo de vida clásico para un proceso de desarrollo de software- Traducido
Fuente: (Ramsin, 2008)- Traducción e interpretación propia*

3.1. Diseño de la metodología híbrida

El Diseño de la metodología Híbrida (Hybrid Methodology Design) parte del ciclo de vida tradicional representado en la **Figura 44** de lo cual se aplica el proceso de construcción de la APK desde sus requerimientos combinando con UML se realiza el análisis que se menciona a continuación:

Su primera iteración es la fase de análisis con la intención de mitigar riesgos de desarrollo y diseño también se segmenta para introducir algo de diseño basado en arquitectura.

CAPÍTULO III - MARCO APLICATIVO

Su segunda iteración realiza una integración, añadiendo la generación de ideas en el inicio del ciclo y una prueba de mercado que no se efectuara puesto que la presente tesis como ser una investigación científica se determinó que el costo de desarrollo como de comercialización no se contemplan por ser gratuito.

La tercera iteración integra directamente el “motor de desarrollo” orientado al aseguramiento de la calidad con la idea de disponer de la arquitectura física en una fase temprana del proceso, dentro de la misma es en donde se aplica el modelo de Algoritmos Genéticos con búsqueda Tabú.

En la cuarta iteración se añaden elementos de prototipado; se refina, además, la fase de iniciación y etapas de prueba donde se van corrigiendo los errores encontrados en la anterior iteración.

3.1.1. Primera iteración - Análisis y Diseño

Dentro de la primera iteración solo se diseñan y ejecutan cuatro fases las cuales se basan en el análisis y el diseño, puesto que la implementación y pruebas solo se efectúan cuando el proyecto de investigación se lo efectúa con ámbitos comerciales.

Las fases de la primera iteración se pueden ver en la *Figura 44*.



Figura 44: Primera iteración en el diseño híbrido (análisis y diseño) - traducido
Fuente: (Ramsin, 2008)- Traducción e interpretación propia

En la primera iteración se realizó entrevistas en el centro de discapacitados **MIKY MAIA** al personal administrativo, docentes, discapacitados de la vista y sordos y mudos, recabando sus necesidades primordiales en relación al sistema de escritura en Braille, plasmándolo en los diagramas de UML.

3.1.1.1. Fase - Análisis preliminar

1) Personal involucrado

- **Discapacitado de la vista-** Emisor y/o Receptor
- **Publico en genera-** Receptor

2) Funcionalidad de los actores

- **Discapacitado de la vista:**

Es el usuario de la APK, tiene acceso a todas sus funciones, teniendo como prioridad y capacidad la comunicación asíncrona con los demás con las siguientes funciones: (Teclado transcriptor braille, Teclado alfanumérico con expresión fonológica, Vocabulario abecedario y números, Documentos guardados y Manual de usuario).

- **Público en general:**

Son las personas con las que se hace uso de la APK, las cuales comprenderán de forma directa y sin complicaciones a los discapacitados de la vista

3) Arquitectura general del sistema

El diagrama de la **Figura 45** muestra la arquitectura general del sistema donde se puede apreciar la relación que se tiene con el sistema Charly y público en general.

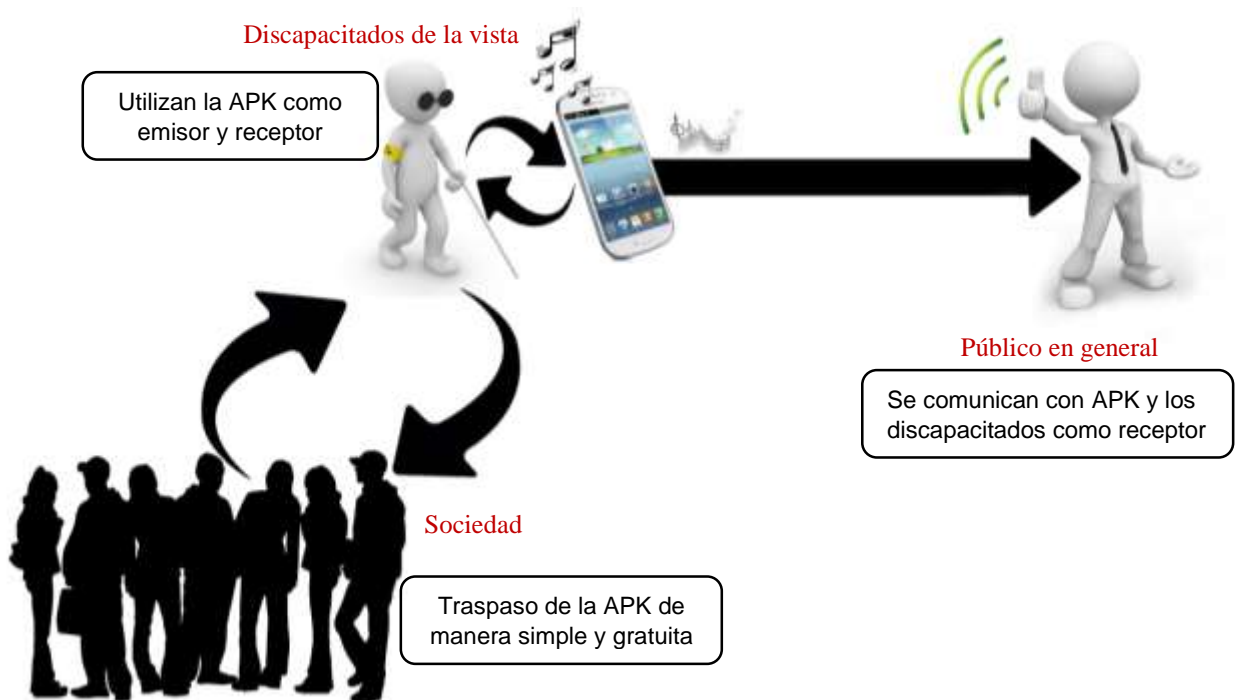


Figura 45: Diagrama general del sistema

Fuente: Elaboración propia

3.1.1.2. Fase - Análisis detallado

Los requerimientos se generaron luego del análisis preliminar y son declaraciones que identifican atributos, capacidades, características y/o cualidades que necesita cumplir la APK para que tenga valor y utilidad para el usuario, mostrando que elementos y funciones son necesarios.

1. **Requisitos funcionales:** Describen el funcionamiento de la APK y sus comportamientos a las diferentes entradas, situaciones y se establecen en la **Tabla 8**.

R1	Acceso y manejo del sistema (APK) de manera amistosa, es decir que la misma se puede manipular sin necesidad de tener privilegios ni ventajas.
R2	Rapidez en el registro de información que se realiza dentro de la misma.
R3	Registro de guardado de datos, modificación y uso de los mismos.
R4	Que el sistema pueda dar indicaciones previas para su uso continuo y su manual respectivo
R5	Indicación continua al usuario por medio de indicaciones asíncronas.

Tabla 7: Requisitos funcionales
FUENTE: Elaboración Propia

2. **Requisitos no Funcionales:** Los requisitos no funcionales describen los atributos y las restricciones que tiene la APK y se establecen en la **Tabla 9**.

R1	El sistema (APK) debe contar con un sistema operativo móvil Android con las siguientes características, las cuales son: Android 4.4 (KitKat), con una versión de API 19.
R2	La APK de no debe ser complejo (usabilidad) para los discapacitados de la vista.
R3	La APK debe ser eficiente y de manejo intuitivo en sus funciones.
R4	La APK debe poseer un tiempo breve de respuesta, por lo cual es utilizado en tiempo real.
R5	La APK podrá guardar la información de manera externa

Tabla 8: Requisitos no funcionales
FUENTE: Elaboración Propia.

CAPÍTULO III - MARCO APLICATIVO

3.1.1.3. Fase –Diseño arquitectónico y detallado

Para la fase del diseño arquitectónico se utilizó el proceso de negocio del UML (*Figura 46*) donde se muestra las actividades diseñadas y su arquitectura para producir una salida específica tanto para los discapacitados de la vista como para el público en general.

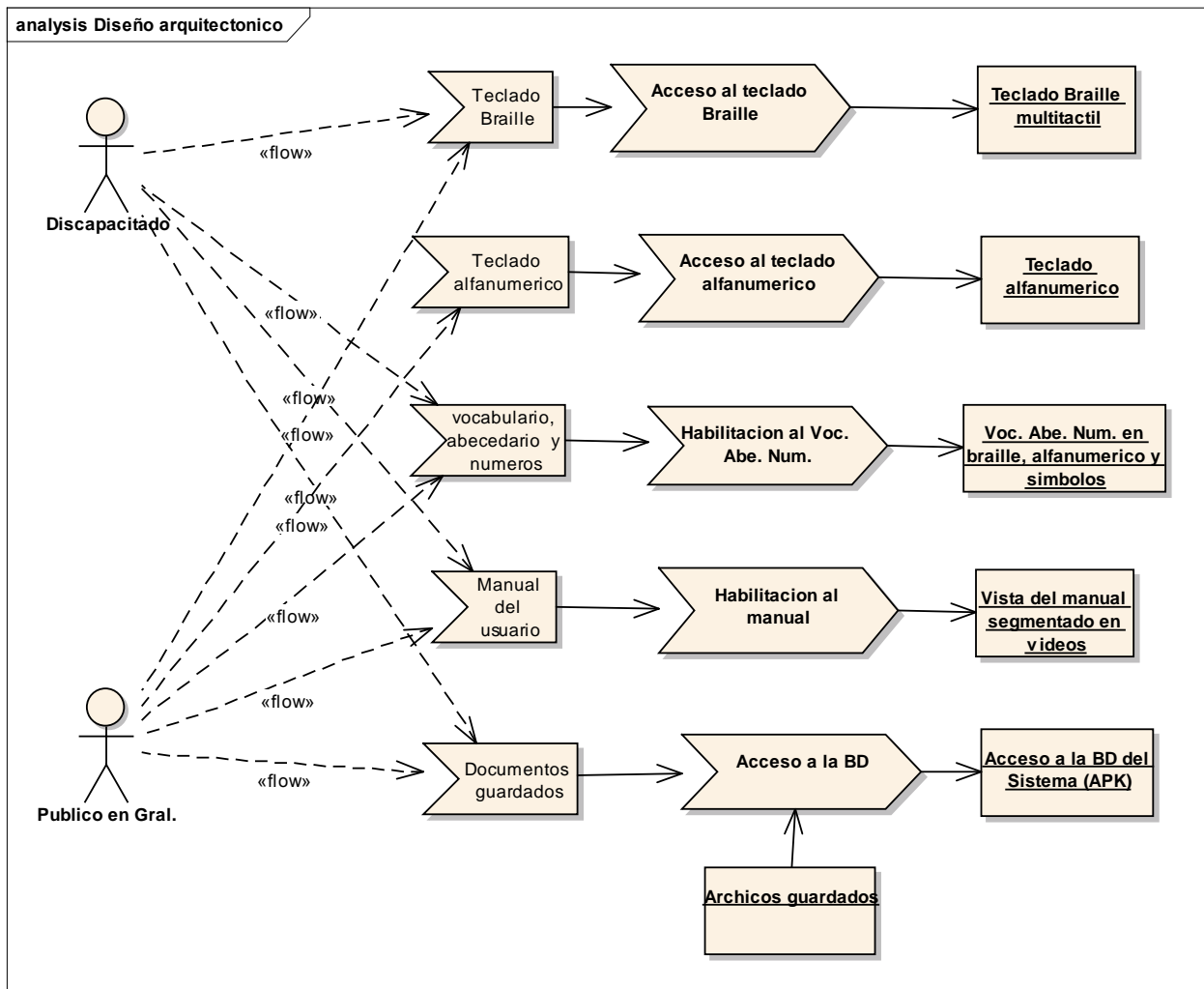


Figura 46 Representación del diagrama de Procesos de la APK
FUENTE: Elaboración Propia

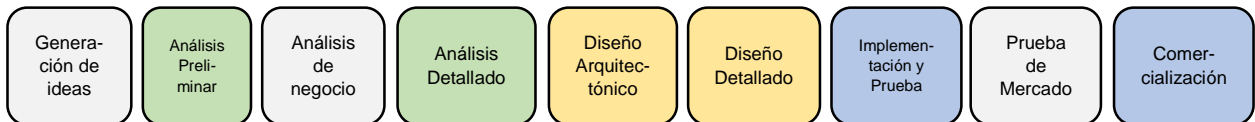
3.1.2. Segunda iteración - Integración

Observando la cantidad de requerimientos y su funcionalidad del mismo se procedió a realizar una integración abarcando la necesidad de sub requerimientos solicitada con pruebas periódicas y necesarias.

Después de observar los requerimientos y el diseño se englobo los problemas mostrando uno en general, subdividiendo los mismos y manteniendo los requerimientos funcionales.

- Teclado transcriptor
- Vocabulario, abecedario y números
- Manual de usuario
- Documentos guardados

Las fases de la segunda iteración son las de la **Figura 47**.



*Figura 47: Segunda iteración en el diseño híbrido (integración) - traducido
Fuente: (Ramsin, 2008)- Traducción e interpretación propia*

La segunda iteración añade la generación de ideas en el inicio del ciclo y una prueba de mercado antes de lanzar la fase de comercialización, la presente tesis no tiene un enfoque de intereses comerciales por lo que no se hizo un análisis de negocio ni una prueba de mercado.

3.1.2.1. Fase –Diseño navegacional – OOHDM

La idea principal es unificar una serie de tareas para obtener el diseño navegacional de la aplicación que permita ejecutar todas las tareas requeridas por los discapacitados de la vista.

Se realiza el diagrama del diseño navegacional de OOHDM que se puede apreciar en la **Figura 44** ilustrando la unión de todas las funciones de contexto de la APK.

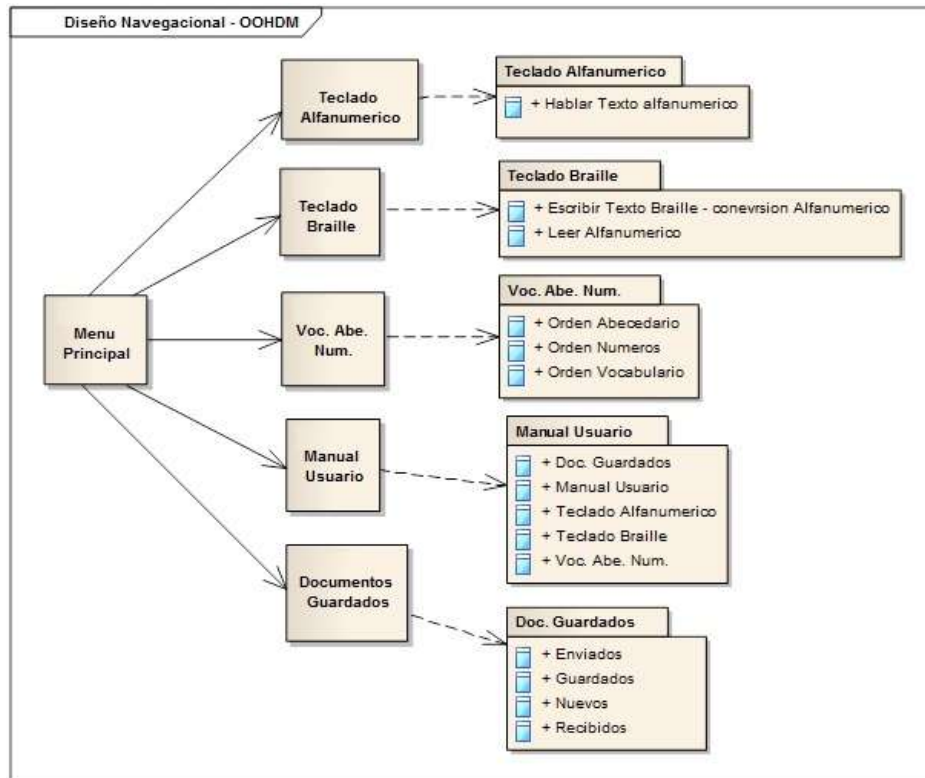


Figura 48: Diseño Navegacional - OOHDM
Fuente: Elaboración propia

CAPÍTULO III - MARCO APLICATIVO

3.1.2.2. Diagrama de clases - Funciones

Se realiza un diagrama de clases de la APK en la cual se observa de manera detallada y explícita la interacción de las clases y su independencia funcional como se puede apreciar en la **Figura 49**.

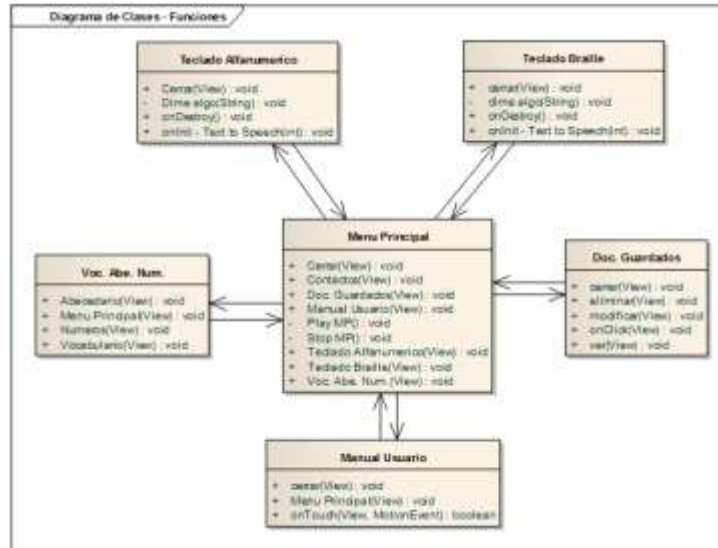


Figura 49 Diagrama de clases - funciones
Fuente: Elaboración propia

3.1.2.3. Base de datos SQLite - Diseño

Se realizó el diseño de la base de datos en SQLite que se enlaza con el programa pasando a ser parte integral del mismo. Con el fin de satisfacer las necesidades del sistema, basado en el resultado de los requerimientos obtenidos con anterioridad.

En el diseño conceptual se obtiene la identificación de las entidades, atributos y sus relaciones, mediante la esquematización de las necesidades reales de una manera fácil de entender. En SQLite todas estas definiciones son guardadas en un solo fichero estándar, se expresa en la **Figura 50**.

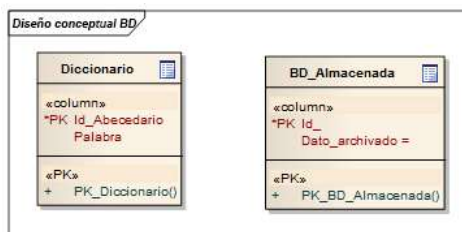


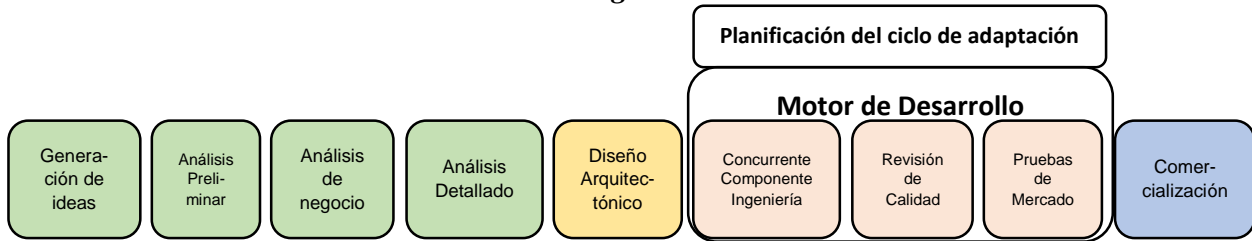
Figura 50: Diseño de la BD
Fuente: Elaboración propia

CAPÍTULO III - MARCO APLICATIVO

3.1.3. Tercera iteración - Motor de Desarrollo

La tercera iteración integra directamente el “motor de desarrollo” de los métodos de desarrollo adaptativo (ASD) orientados al aseguramiento de la calidad en los procesos de desarrollo con la idea de disponer de la arquitectura física en una fase temprana del proceso.

Las fases de la tercera iteración son las de la *Figura 51*.



*Figura 51: Tercera iteración en el diseño híbrido (motor de desarrollo) - traducido
Fuente: (Ramsin, 2008)- Traducción e interpretación propia*

3.1.3.1. Arquitectura de desarrollo utilizada

Los requerimientos de hardware y software para su desarrollo se realizaron con las siguientes características que a continuación se mencionan en las *Tablas 10 – 11*.

Equipo	Descripción	Función
Sony VAIO	Intel Core i5 - 1,6 Ghz 4 Gb RAM Disco Duro 500 Gb	Contendrá el software requerido para el desarrollo de la tesis en general.

*Tabla 9 Hardware de desarrollo
Fuente: Interpretación propia*

Software de desarrollo en la APK, dentro de las herramientas en la fase de construcción, se tiene:

Software	Función
Eclipse Mars 1.0	IDE gratuita para Windows que permite la edición de código Java para la creación de aplicaciones Android.
Java 7	Lenguaje de programación orientado a objetos.
SQLite	SQLite es un gestor de base de datos.
XML	Metalinguaje cuya labor es definir unas reglas que deberá cumplir un lenguaje de marcas (Ventanas).
JDK	Android hace uso de JDK para el desarrollo de aplicaciones.
SDK	SDK proporciona las librerías de las APIs, documentación y ejemplos y todas las herramientas para poder construir, comprobar y depurar aplicaciones
ADT	El ADT (Android Development Tools o Herramientas de Desarrollo de Android) es el plugin de Eclipse.

*Tabla 10: Software de desarrollo
Fuente: Interpretación propia*

3.1.3.2. Requerimientos de la APK

Para realizar el motor de desarrollo en su implementación primeramente se analiza las características del sistema operativo elegido y sus características tanto de hardware como de software.

Sus requerimientos mínimos y ventajas de la Versión Android y de su API son las de la **Tabla 12**.

Requerimientos de la APK	
Versión Android	4.4 (KitKat)
RAM	512 MB
API	19 (adapta el comportamiento de la APK en dispositivos con poca memoria)
Tamaño	El modo de inmersión en la pantalla oculta las interfaces del sistema permitiendo usarla toda en si
Pixel	Sin limitaciones y sin restricciones por que utiliza porcentajes

Tabla 11: Requerimientos de la APK
Fuente: Interpretación propia

Los requerimientos de software y hardware, se analiza de forma detallada los requerimientos de que necesitara la APK, a modo de reducir los mismos se programó las dimensiones en porcentajes con el objetivo de reducir sus limitaciones de las distintas pantallas de los sistemas operativos móviles Android.

3.1.3.3. Modelo táctil del traductor - CHARLY

Implementando las especificaciones del diseño que se ejecutaron con la anterior iteracion se forma el primer prototipo en el cual se observa la interaccion de los mismos dentro del Modelo Navegacional y se dividió por sus funciones principales para entender las subfunciones que tiene cada una de ellas y se muestra en las **Figuras 52 a 56**.

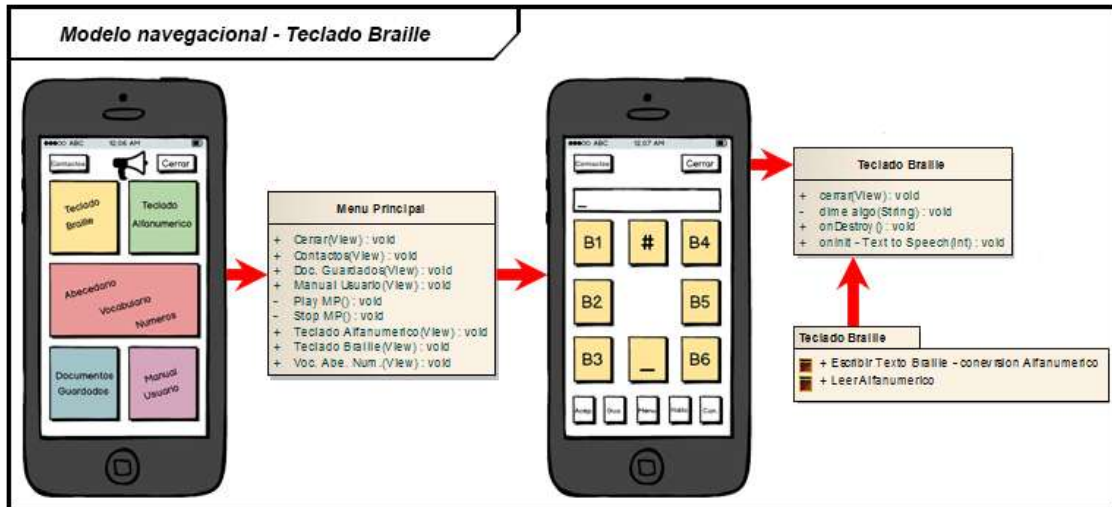


Figura 52: Modelo navegacional - Teclado Braille
 Fuente: Interpretación propia

El modelo táctil del traductor Charly posee características particulares, puesto que después de haber realizado entrevistas con el Director Departamental del I.B.C. (Lic. Gregorio Tellez) y con la especialista en interpretación (Lic. Nancy Portal), se aclaró que de manera puntual al momento de hacer un valor numérico en el sistema de traducción en Braille el mismo debería llevar un antenúmero para diferenciar con las letras.

Posterior a esto, se hizo un análisis exhaustivo se llegó a la conclusión de crear un 7mo. Carácter para los valores numéricos. Esto con el fin de hacer la aplicación más amistosa, puesto que colocar un antenúmero significaría colocar otros 6 caracteres, generando un total de 12 caracteres para un valor numérico, este método facilita de gran forma su escritura y torna la misma más amistosa para el usuario.

Después se generó un 8vo. Carácter para la función de generar espacios entre palabras, esto con el fin de evitar los 12 caracteres normados en el sistema Braille, estas modificaciones en el teclado traductor se lo puede apreciar en la **Figura 52**.

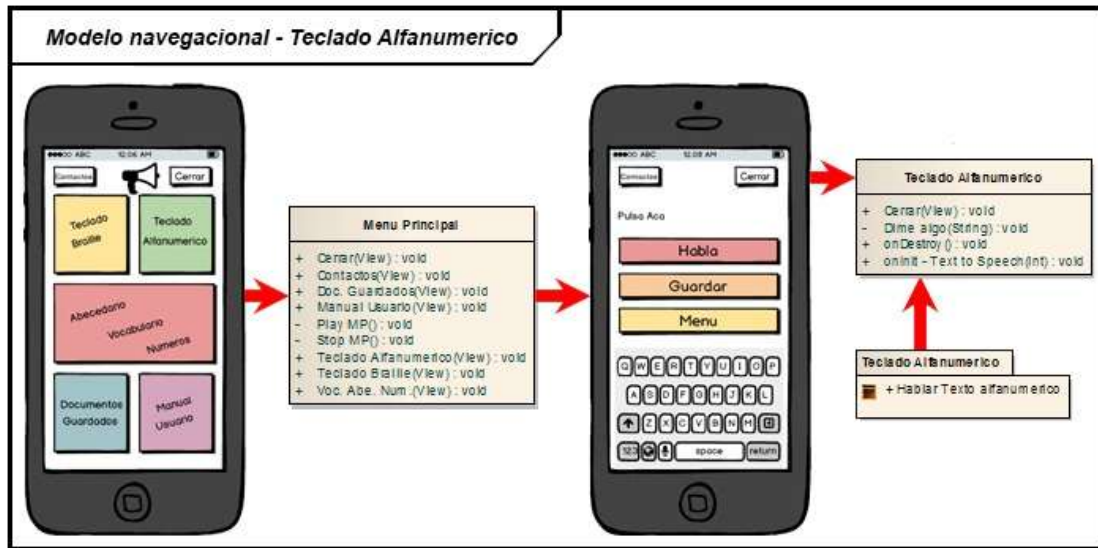


Figura 53 Modelo navegacional - Teclado Alfanumerico
Fuente: Interpretación propia

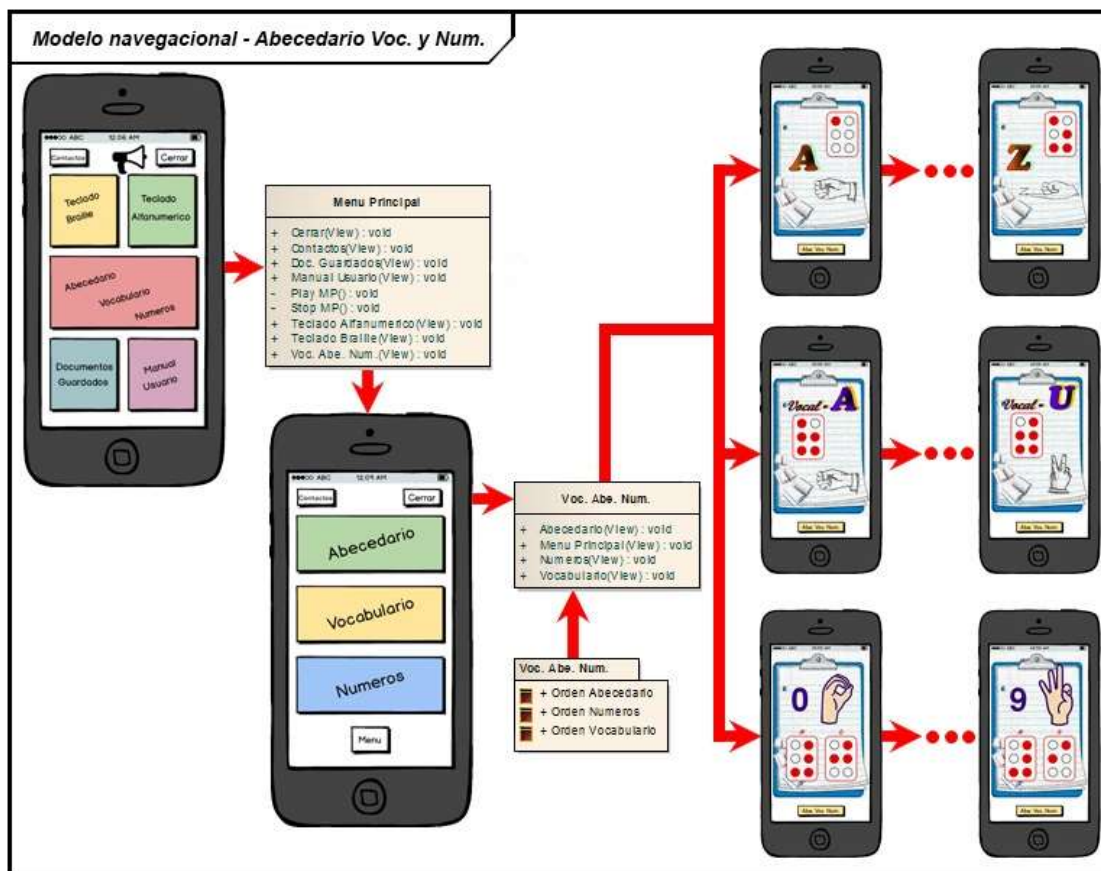


Figura 54 Modelo navegacional - Abecedario Vocabulario y Numeros
Fuente: Interpretación propia

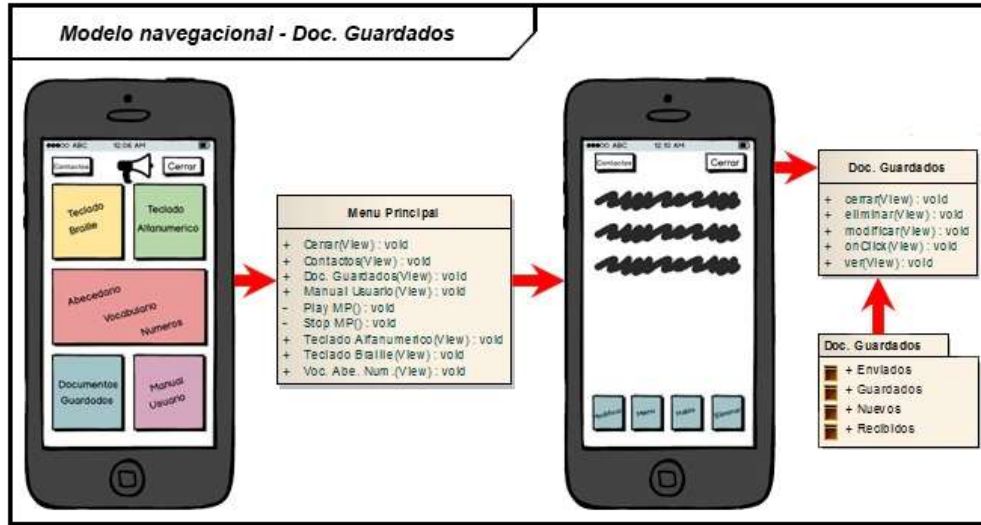


Figura 55 Modelo navegacional - Documentos guardados
Fuente: Interpretación propia

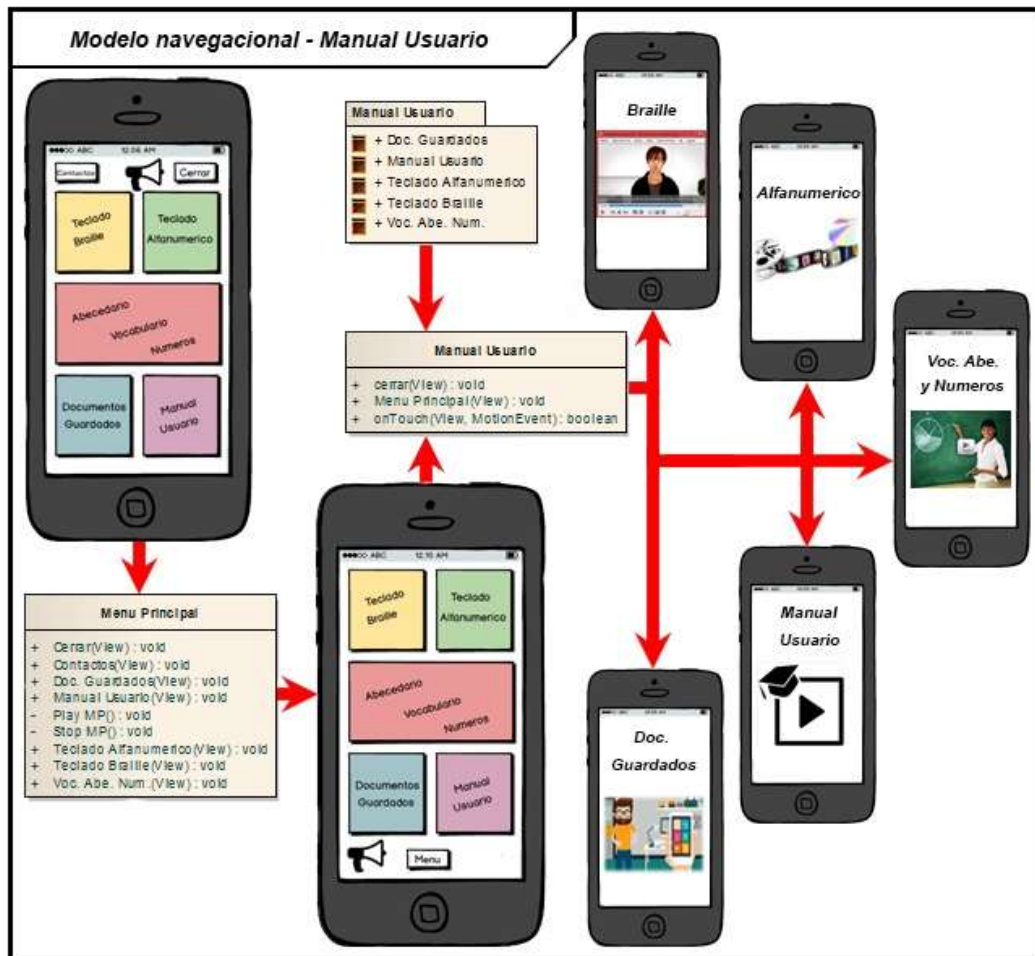


Figura 56 Modelo navegacional - Manual de Usuario
Fuente: Interpretación propia

CAPÍTULO III - MARCO APLICATIVO

Partiendo del diseño navegacional desarrollado, se implementa la estructura del APK representada en la *Figura 57*.

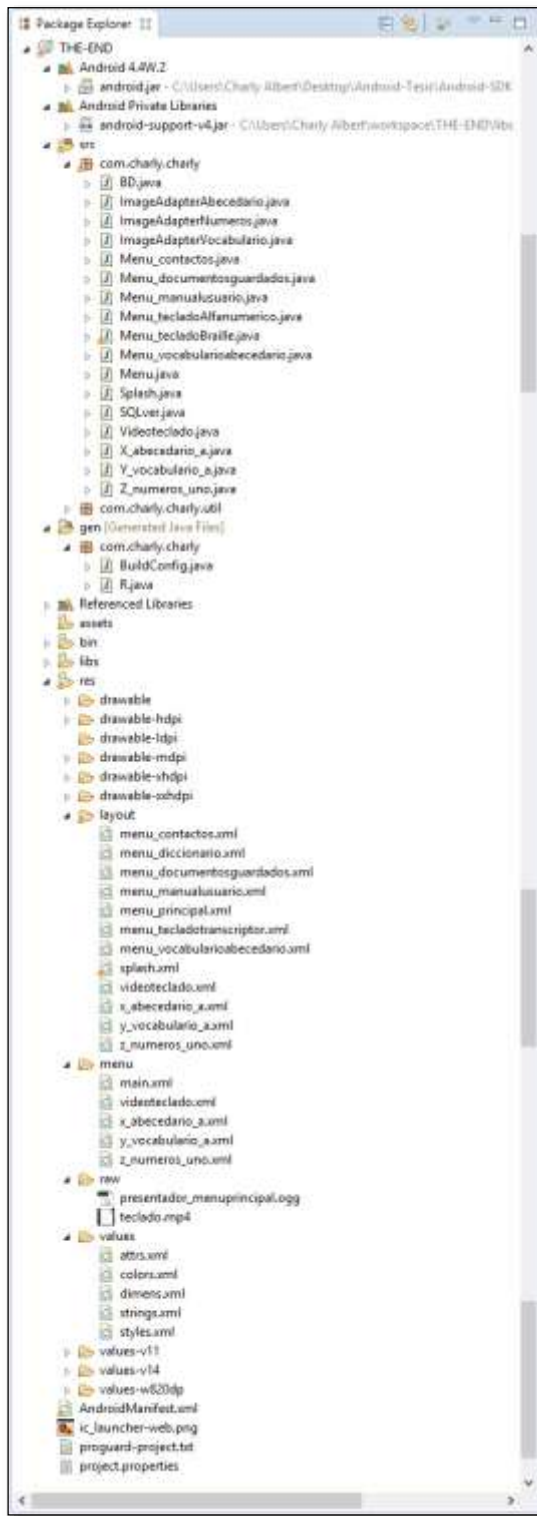


Figura 57: Estructura de la APK
Fuente: Interpretación propia

CAPÍTULO III - MARCO APLICATIVO

Teniendo la estructura se efectúa las líneas de código de las clases hechas en JAVA donde las ventanas de la aplicación son efectuadas en XML, que se muestra a continuación cada una de las clases según su orden secuencial de la APK.

Primero se mostrara la clase de bienvenida con el efecto Splash (*Código 6*).

```
package com.charly.charly;

import android.app.Activity;

public class Splash extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        setContentView(R.layout.splash);

        Thread logouap = new Thread(){
            public void run(){
                try{
                    int tiempo = 0;
                    while(tiempo<5000){
                        sleep(100);
                        tiempo = tiempo + 100;
                    }
                    Intent a = new Intent("com.charly.charly.menulegal");
                    startActivity(a);
                }catch(InterruptedException e){
                    e.printStackTrace();
                }finally{
                    finish();
                }
            }
        };
        logouap.start();
    }
}
```

Código 6: Clase Splash
Fuente: Implementación propia

CAPÍTULO III - MARCO APLICATIVO

Después de que se ejecutó la clase Splash y esperamos el tiempo programado en el Thread, se procederá de forma automática a ejecutar la clase de Menú Principal (*Código 7*).

```
*Menu.java

import android.app.Activity;

public class Menu extends Activity{

    public MediaPlayer mp;

    protected void onCreate(Bundle savedInstanceState) {

        //-----
        private void play_mp() {}
        private void stop_mp() {}
        //-----
        //menus
        public void Contactos_ (View v){
            Intent contactos_ = new Intent(this,Menu_contactos.class);
            startActivity(contactos_);
        }
        //-----
        public void Cerrar_ (View v){}
        //-----
        public void Teclado_ (View v){}
        //-----
        public void Diccionario_ (View v){}
        //-----
        public void Vocabulario_ (View v){}
        //-----
        public void Manual_ (View v){}
        //-----
        public void Guardados_ (View v){}
        //-----
    }
}
```

Código 7 Clase Menu Principal
Fuente: Implementación propia

Se puede observar que en la línea de código mostrada anteriormente solo se muestra las funciones que tiene la clase puesto que la misma es demasiado larga e impediría comprenderla si se la muestra de forma completa, lo mismo se observara en las líneas de código posteriores a modo de que se pueda comprender sus funciones.

CAPÍTULO III - MARCO APLICATIVO

En la línea de *Código 8* mostrada a continuación se puede observar la clase del Teclado Braille

```
*Menu_tecladoBraille.java ✕
package com.charly.charly;
import java.util.Locale;

public class Menu_tecladoBraille extends Activity implements OnInitListener{

    String palabras=""; String letra=""; String usando=""; String compiladoBinario="";

    String Binario1="2"; String Binario2="2"; String Binario3="2";
    String Binario4="2"; String Binario5="2"; String Binario6="2";

    String BinarioNum="2"; String BinarioEsp="2";

    //Motordevoz
    private TextToSpeech x;
    //
    private String leer;
    private Button reproduccion;
    private Button aceptar;
    private Button cancelar;

    protected void onCreate(Bundle savedInstanceState) {}

    public boolean onCreateOptionsMenu(Menu menu) {}
    //Inicia X
    public void onInit(int status) {}
    //metodoconversiondetextoavoz
    private void dime_algo( String texto ) {}
    //Cuando se cierra la aplicacion se destruye el XXX
    public void onDestroy() {}

    public void B (View v){}
}
```

Código 8: Clase Teclado Braille
Fuente: Implementación propia

CAPÍTULO III - MARCO APLICATIVO

En la clase del teclado Alfanumérico observada a continuación se puede destacar la diferencia que posee con el teclado Braille para su comprensión (*Código 9*).

```
*Menu_tecladoAlfanumerico.java
package com.charly.charly;
import java.util.Locale;

public class Menu_tecladoAlfanumerico extends Activity implements OnInitListener{

    //Motordevoz
    private TextToSpeech xxx;

    private EditText textoleido; private Button botonreproduccion;

    protected void onCreate(Bundle savedInstanceState) {}
    public boolean onCreateOptionsMenu(Menu menu) {}
    //Inicia X
    public void onInit(int status) {}
    //metodoconversiondetextoavoz
    private void dime_algo( String texto ) {}
    //Cuando se cierra la aplicacion se destruye el X
    public void onDestroy() {}

    public void E (View v){
}
}
```

Código 9: Clase Teclado Alfanumerico
Fuente: Implementación propia

En la clase Manual Usuario se confirma que el mismo se efectúa en forma de videos asíncronos para su fácil manipulación (*Código 10*).

```
*Menu_manualusuario.java
package com.charly.charly;
import android.app.Activity;

public class Menu_manualusuario extends Activity implements onTouchListener{

    View Rteclado;

    protected void onCreate(Bundle savedInstanceState) {}
    public boolean onTouch(View v, MotionEvent event) {}

    public void D (View v){
}
}
```

Código 10 Clase Manual Usuario
Fuente: Implementación propia

CAPÍTULO III - MARCO APLICATIVO

En la clase Vocabulario Abecedario y Números (*Código 11*) se lo genera de forma independiente a cada uno de ellos para observar y evidenciar sus diferencias y similitudes, puesto que el mismo está diseñado y programado para mostrarlo en Alfanumérico, Braille y símbolos.

```
*Menu_vocabularioabecedario.java ☒
package com.charly.charly;
import android.app.Activity;
public class Menu_vocabularioabecedario extends Activity {
    protected void onCreate(Bundle savedInstanceState) {}
    public void MenuPrincipal_ (View v){}
    public void Abecedario_ (View v){}
    public void Vocabulario_ (View v){}
    public void Numeros_ (View v){}
}
```

Código 11 Clase Vocabulario Abecedario y Numeros
Fuente: Implementación propia

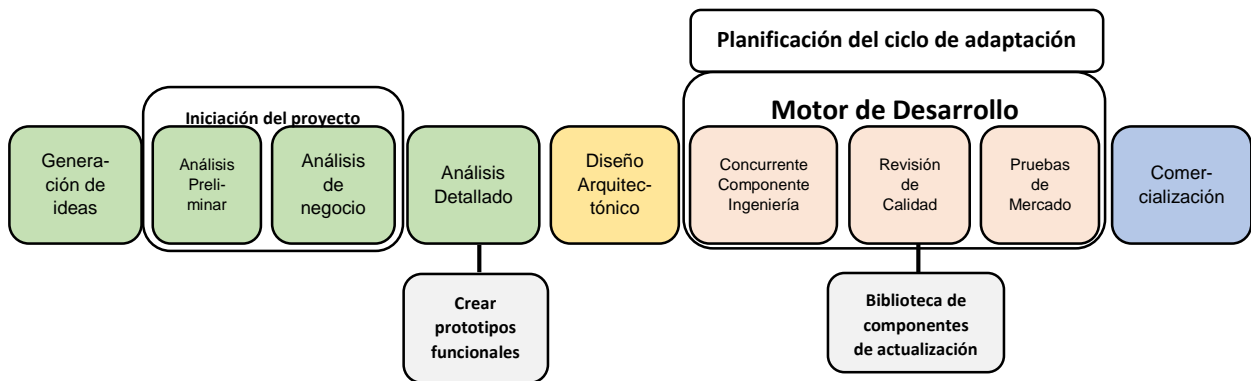
En la clase Documentos Guardados (*Código 12*) se puede ver la interacción con la Base de Datos.

```
*Menu_documentosguardados.java ☒
package com.charly.charly;
import android.app.Activity;
public class Menu_documentosguardados extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.menu_documentosguardados);
    }
    public void C (View v){
        Intent c_ = new Intent(this, com.charly.charly.Menu.class);
        startActivity(c_);
        Menu_documentosguardados.this.finish();
    }
}
```

Código 12: Clase Documentos Guardados
Fuente: Implementación propia

3.1.4. Cuarta iteración – Prototipado

Pensando en aquella propiedad "ideal", en la cuarta iteración se añaden elementos de prototipado final (Harishankar, 2013). Son las etapas de prueba donde se van corrigiendo los errores encontrados en la anterior iteración. Las fases de la cuarta iteración son las de la **Figura 58**.



*Figura 58: Cuarta iteración en el diseño híbrido (prototipado) - traducido
Fuente: (Ramsin, 2008)- Traducción e interpretación propia*

Los layouts respectivos dependen de la funcionalidad de las clases y la interacción con el usuario lo cual se puede observar de manera más detallada en el prototipado final que se encuentra en **Anexo D**

La descripción de la APK depende mucho de su Android Manifest, el cual es una columna vertebral para la aplicación puesto que dentro del mismo se observa sus iteraciones y funciones disponibles dentro de la misma.

El Manifest de Android, denominado columna vertebral se puede observar en línea de **Código 13**.

```
Manifest II
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.charly.charly"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="19"
        android:targetSdkVersion="20" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".Splash"
            android:configChanges="orientation/keyboardHidden/screenSize"
            android:label="@string/app_name"
            android:theme="@style/Theme.Transparent" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".Menu"
            android:configChanges="orientation/keyboardHidden/screenSize"
            android:label="@string/app_name"
            android:theme="@style/FullscreenTheme" >
            <intent-filter>
                <action android:name="com.charly.charly.menulegal" />

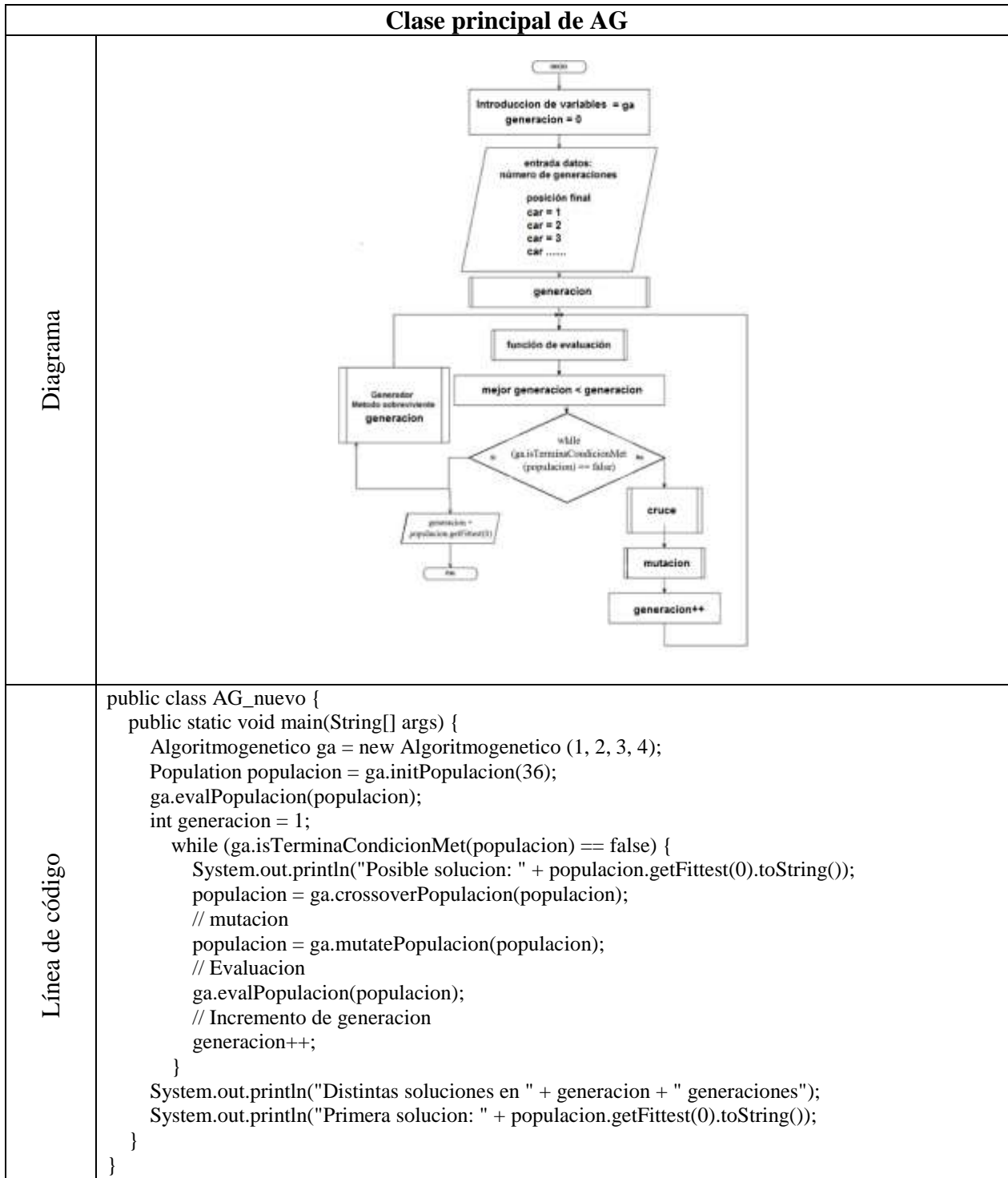
                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
        <activity
            android:name=".Menu_contactos"
            android:label="@string/title_activity_menu_contactos" >
        </activity>
        <activity
            android:name=".Menu_teciadoBraille"
            android:label="@string/title_activity_menu_diccionario" >
        </activity>
        <activity
            android:name=".Menu_documentosguardados"
            android:label="@string/title_activity_menu_documentosguardados" >
        </activity>
        <activity
            android:name=".Menu_manualusuario"
            android:label="@string/title_activity_menu_manualusuario" >
        </activity>
        <activity
            android:name=".Menu_teciadoAlfanumerico"
            android:label="@string/title_activity_menu_teciadotranscriptor" >
        </activity>
        <activity
            android:name=".Menu_vocabularioabecedario"
            android:label="@string/title_activity_menu_vocabularioabecedario" >
        </activity>
        <activity
            android:name=".Z_numeros_uno"
            android:label="@string/title_activity_z_numeros_uno" >
        </activity>
        <activity
            android:name=".X_abecedario_a"
            android:label="@string/title_activity_x_abecedario_a" >
        </activity>
        <activity
            android:name=".Y_vocabulario_a"
            android:label="@string/title_activity_y_vocabulario_a" >
        </activity>
        <activity
            android:name=".Videoteclado"
            android:theme="@style/Theme.Transparent"
            android:label="@string/title_activity_videoteclado" >
        </activity>
    </application>
</manifest>
```

Código 13: Android Manifest
Fuente: Implementación propia

CAPÍTULO III - MARCO APLICATIVO

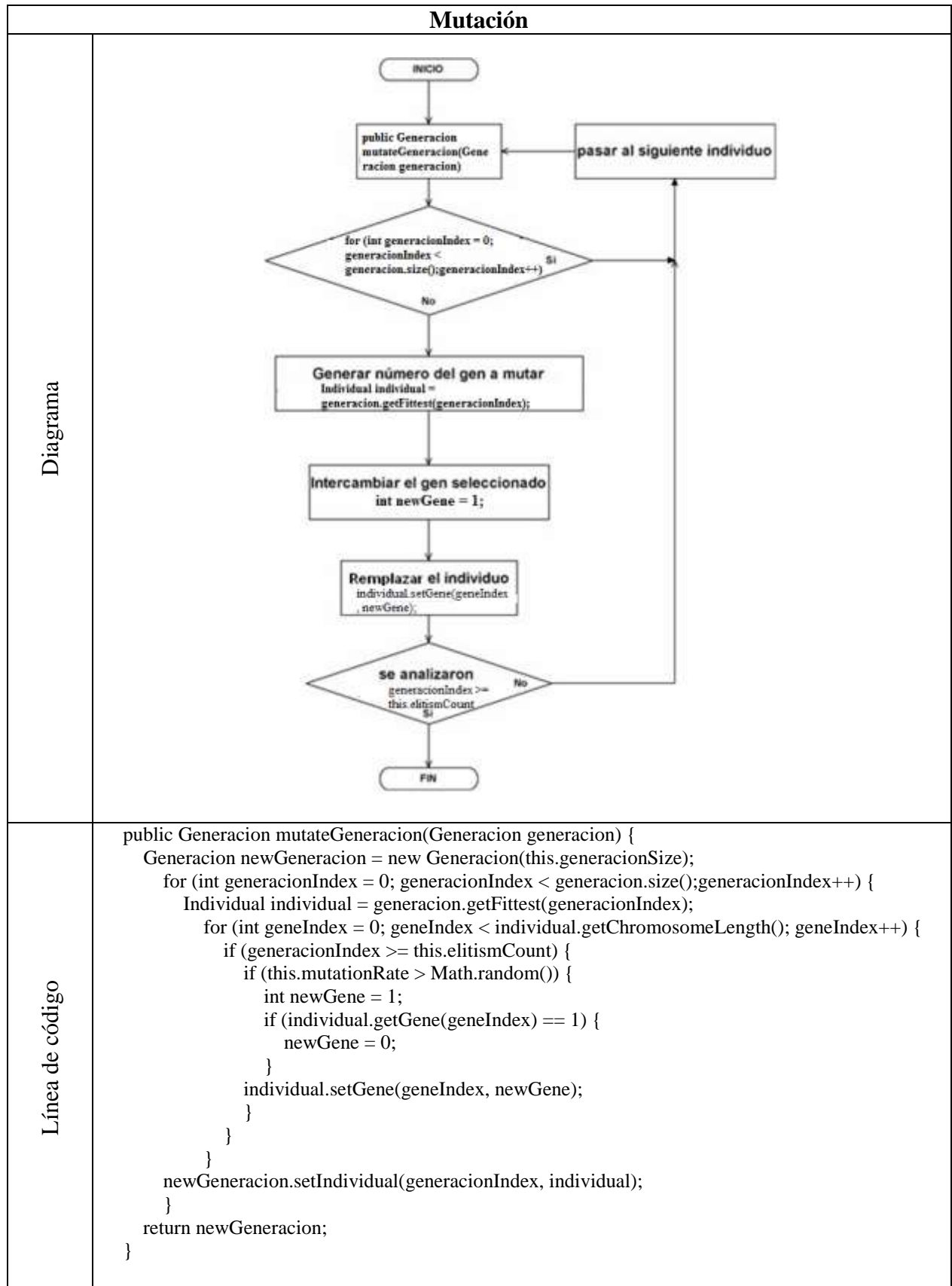
3.1.4.1. Algoritmos Genéticos y Tabú

Los algoritmos genéticos se representan en diagramas, para comprender sus actividades y sus mutaciones. Los que se demuestran en:



Primera generación	
Diagrama	
Línea de código	<pre> public class Algoritmogenetico { private int generacionSize; private double mutacionRate; private double crossoverRate; private int elitismCount; public Algoritmogenetico(int generacionSize, double mutacionRate, double crossoverRate, int elitismCount) { this.generacionSize = generacionSize; this.mutacionRate = mutacionRate; this.crossoverRate = crossoverRate; this.elitismCount = elitismCount; } public Generacion initPoblacion(int chromosomeLength) { Generacion generacion = new Generacion(this.generacionSize, chromosomeLength); return generacion; } public double calcFitness(Individual individual) { int correctGenes = 0; for (int geneIndex = 0; geneIndex < individual.getChromosomeLength(); geneIndex++) { if (individual.getGene(geneIndex) == 1) { correctGenes += 1; } } double fitness = (double) correctGenes / individual.getChromosomeLength(); individual.setFitness(fitness); return fitness; } } </pre>

Cruze	
Diagrama	<pre> graph TD Inicio([Inicio]) --> crossoverGeneraciones[crossoverGeneraciones] crossoverGeneraciones --> generarNewPopulacion[generar newPopulacion] generarNewPopulacion --> forPopulacionIndex[for (int populacionIndex = 0; populacionIndex < generacion.size(); populacionIndex++)] forPopulacionIndex --> selectParent1[Individual parent1 = generacion.getFittest(populacionIndex);] selectParent1 --> crossoverRateCheck{if (this.crossoverRate > Math.random() && populacionIndex > this.elitismCount)} crossoverRateCheck -- No --> returnNewPopulacion[return newPopulacion] crossoverRateCheck -- Si --> selectParent2[Individual parent2 = selectParent(generacion);] selectParent2 --> forGeneIndex[for (int geneIndex = 0; geneIndex < parent1.getChromosomeLength(); geneIndex++)] forGeneIndex --> swapGenes[if (0.5 > Math.random()) { offspring.setGene(geneIndex, parent1.getGene(geneIndex)); } else { offspring.setGene(geneIndex, parent2.getGene(geneIndex)); }] swapGenes --> returnNewPopulacion returnNewPopulacion --> Fin([Fin]) </pre>
Línea de código	<pre> public Generacion crossoverGeneracion(Generacion generacion) { //generacion Generacion newPopulacion = new Generacion(generacion.size()); for (int populacionIndex = 0; populacionIndex < generacion.size(); populacionIndex++) { Individual parent1 = generacion.getFittest(populacionIndex); if (this.crossoverRate > Math.random() && populacionIndex > this.elitismCount) { Individual offspring = new Individual(parent1. getChromosomeLength()); Individual parent2 = selectParent(generacion); for (int geneIndex = 0; geneIndex < parent1.getChromosomeLength(); geneIndex++) { if (0.5 > Math.random()) { offspring.setGene(geneIndex, parent1.getGene(geneIndex)); } else { offspring.setGene(geneIndex, parent2.getGene(geneIndex)); } } newPopulacion.setIndividual(populacionIndex, offspring); } else { newPopulacion.setIndividual (populacionIndex, parent1); } } return newPopulacion; } </pre>



Sobreviviente	
Diagrama	<pre> graph TD INICIO([INICIO]) --> Call[Caracter getFittest
(int offset)
mejor camino ultima generacion] Call --> NI[NI = 0] NI --> N["N = 0;"] N --> Gen[generador aleatorio
valores 0 ó 1] Gen --> Store[Almacena el gen
en la posicion
N del individuo] Store --> Cond1{if (o1.getFitness()
> o2.getFitness())} Cond1 -- Si --> Ret1[return -1;] Cond1 -- No --> Cond2{else if (o1.getFitness()
< o2.getFitness())} Cond2 -- Si --> Ret2[return 1;] Cond2 -- No --> Ret3[return
this.generacion[offset];] Ret3 --> Out[this.generacion] Out --> FIN([FIN]) </pre>
Línea de código	<pre> public Caracter getFittest(int offset) { Arrays.sort(this.generacion, new Comparator<Caracter>() { @Override public int compare(Caracter o1, Caracter o2) { if (o1.getFitness() > o2.getFitness()) { return -1; } else if (o1.getFitness() < o2.getFitness()) { return 1; } return 0; } }); return this.generacion[offset]; } </pre>

CAPÍTULO III - MARCO APLICATIVO

Algoritmos Genéticos y TABU con mecanismos de búsqueda local, que se describe a continuación:

Los pasos son los siguientes:

Paso 1. Generar individuos N_{pop} al azar, donde es N_{pop} el tamaño de la población y se almacenan en la base de datos.

Paso 2. Aplicar los operadores tales como cruces, mutaciones y selecciones a los individuos en una población de Algoritmos Genéticos.

Paso 3. Se tiene el y_{mejor} individual, que es el mejor individuo de una población de Algoritmos Genéticos, en la base de datos y se establece en su cadena de bits de la que los valores de todos son '0'; Sin embargo, y_{mejor} no se almacena cuando se incluye en las regiones buscadas, que se indican por individuos que ya han sido almacenados en la base de datos.

Paso 4. Búsqueda local expande una región explorada indicado por un determinado individuo almacenado en la base de datos. Cuando un mejor individuo se encuentra, vuelva a colocar el peor individuo de la población de Algoritmo Genético con él.

Paso 5. Vuelva al paso 2 hasta que algunas condiciones de terminación, por ejemplo costo de la informática llega a una cantidad limitada o la búsqueda exhaustiva se realiza y están satisfechos.

En el paso 3, cuando hay N_{DB} individuos en la base de datos, sustituya el x_{peor} individual que tiene el peor aptitud en la base de datos con y_{mejor} si el estado físico de y_{mejor} es más grande que la de x_{peor} , de lo contrario y_{mejor} no se almacena, donde N_{DB} es el parámetro de capacidad de la base de datos.

Este mecanismo utiliza los recursos informáticos de manera efectiva ya que los incrementos espaciales buscados linealmente como el número de aumento de los recursos de una búsqueda exhaustiva se garantiza en los cálculos infinitos.

La metodología de Algoritmos Genéticos con Búsqueda local Tabú se integra en el motor de desarrollo (**Código 14**), especificándolo de manera más directa dentro de las clase de Braille y

CAPÍTULO III - MARCO APLICATIVO

Alfanumérico, puesto que dentro de ellas se procede a efectuar la búsqueda de caracteres y guardando a una memoria a corto plazo las posibles soluciones.

```
package com.charly.charly;
import java.util.Locale;

public class Menu_tecladoBucle extends Activity implements OnItemClickListener{

    String palabras=""; String letra=""; String usado=""; String compiladoBinario="";

    String Binario1="2"; String Binario2="2"; String Binario3="2";
    String Binario4="2"; String Binario5="2"; String Binario6="2";

    String Binario7="2"; String Binario8="2";

    //Intento
    private TextToSpeech s;
    //
    private String leer;
    private Button reproduction;
    private Button acceptar;
    private Button cancelar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.menu_diccionario);

        s = new TextToSpeech( this, this );

        final Button B1=(Button)findViewById(R.id.Bot1);          final Button B2=(Button)findViewById(R.id.Bot2);
        final Button B3=(Button)findViewById(R.id.Bot3);          final Button B4=(Button)findViewById(R.id.Bot4);
        final Button B5=(Button)findViewById(R.id.Bot5);          final Button B6=(Button)findViewById(R.id.Bot6);

        final Button B7=(Button)findViewById(R.id.Bot7);          final Button B8=(Button)findViewById(R.id.Bot8);

        final EditText pantalla =(EditText)findViewById(R.id.TextoLeido);

        B1.setOnClickListener(new OnClickListener() {
        B2.setOnClickListener(new OnClickListener() {
        B3.setOnClickListener(new OnClickListener() {
        B4.setOnClickListener(new OnClickListener() {
        B5.setOnClickListener(new OnClickListener() {
        B6.setOnClickListener(new OnClickListener() {
        public void onClick(View v) {}
        });

        //-----
        B7.setOnClickListener(new OnClickListener() {}
        B8.setOnClickListener(new OnClickListener() {}
        //-----
        acceptar = (Button) findViewById(R.id.Acceptar);
        aceptar.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View arg0) {

                compiladoBinario=Binario1;      compiladoBinario=compiladoBinario+Binario2;
                compiladoBinario=compiladoBinario+Binario3;      compiladoBinario=compiladoBinario+Binario4;
                compiladoBinario=compiladoBinario+Binario5;      compiladoBinario=compiladoBinario+Binario6;
                compiladoBinario=compiladoBinario+Binario7;      compiladoBinario=compiladoBinario+Binario8;
                compiladoBinario=compiladoBinario+Binario9;      compiladoBinario=compiladoBinario+Binario10;
                compiladoBinario=compiladoBinario+Binario11;      compiladoBinario=compiladoBinario+Binario12;
                compiladoBinario=compiladoBinario+Binario13;      compiladoBinario=compiladoBinario+Binario14;
                compiladoBinario=compiladoBinario+Binario15;      compiladoBinario=compiladoBinario+Binario16;
                compiladoBinario=compiladoBinario+Binario17;      compiladoBinario=compiladoBinario+Binario18;
                compiladoBinario=compiladoBinario+Binario19;      compiladoBinario=compiladoBinario+Binario20;
                compiladoBinario=compiladoBinario+Binario21;      compiladoBinario=compiladoBinario+Binario22;
                compiladoBinario=compiladoBinario+Binario23;      compiladoBinario=compiladoBinario+Binario24;
                compiladoBinario=compiladoBinario+Binario25;      compiladoBinario=compiladoBinario+Binario26;
                compiladoBinario=compiladoBinario+Binario27;      compiladoBinario=compiladoBinario+Binario28;
                compiladoBinario=compiladoBinario+Binario29;      compiladoBinario=compiladoBinario+Binario30;
                compiladoBinario=compiladoBinario+Binario31;      compiladoBinario=compiladoBinario+Binario32;
                compiladoBinario=compiladoBinario+Binario33;      compiladoBinario=compiladoBinario+Binario34;
                compiladoBinario=compiladoBinario+Binario35;      compiladoBinario=compiladoBinario+Binario36;
                compiladoBinario=compiladoBinario+Binario37;      compiladoBinario=compiladoBinario+Binario38;
                compiladoBinario=compiladoBinario+Binario39;      compiladoBinario=compiladoBinario+Binario40;
                compiladoBinario=compiladoBinario+Binario41;      compiladoBinario=compiladoBinario+Binario42;
                compiladoBinario=compiladoBinario+Binario43;      compiladoBinario=compiladoBinario+Binario44;
                compiladoBinario=compiladoBinario+Binario45;      compiladoBinario=compiladoBinario+Binario46;
                compiladoBinario=compiladoBinario+Binario47;      compiladoBinario=compiladoBinario+Binario48;
                compiladoBinario=compiladoBinario+Binario49;      compiladoBinario=compiladoBinario+Binario50;
                compiladoBinario=compiladoBinario+Binario51;      compiladoBinario=compiladoBinario+Binario52;
                compiladoBinario=compiladoBinario+Binario53;      compiladoBinario=compiladoBinario+Binario54;
                compiladoBinario=compiladoBinario+Binario55;      compiladoBinario=compiladoBinario+Binario56;
                compiladoBinario=compiladoBinario+Binario57;      compiladoBinario=compiladoBinario+Binario58;
                compiladoBinario=compiladoBinario+Binario59;      compiladoBinario=compiladoBinario+Binario60;
                compiladoBinario=compiladoBinario+Binario61;      compiladoBinario=compiladoBinario+Binario62;
                compiladoBinario=compiladoBinario+Binario63;      compiladoBinario=compiladoBinario+Binario64;
                compiladoBinario=compiladoBinario+Binario65;      compiladoBinario=compiladoBinario+Binario66;
                compiladoBinario=compiladoBinario+Binario67;      compiladoBinario=compiladoBinario+Binario68;
                compiladoBinario=compiladoBinario+Binario69;      compiladoBinario=compiladoBinario+Binario70;
                compiladoBinario=compiladoBinario+Binario71;      compiladoBinario=compiladoBinario+Binario72;
                compiladoBinario=compiladoBinario+Binario73;      compiladoBinario=compiladoBinario+Binario74;
                compiladoBinario=compiladoBinario+Binario75;      compiladoBinario=compiladoBinario+Binario76;
                compiladoBinario=compiladoBinario+Binario77;      compiladoBinario=compiladoBinario+Binario78;
                compiladoBinario=compiladoBinario+Binario79;      compiladoBinario=compiladoBinario+Binario80;
                compiladoBinario=compiladoBinario+Binario81;      compiladoBinario=compiladoBinario+Binario82;
                compiladoBinario=compiladoBinario+Binario83;      compiladoBinario=compiladoBinario+Binario84;
                compiladoBinario=compiladoBinario+Binario85;      compiladoBinario=compiladoBinario+Binario86;
                compiladoBinario=compiladoBinario+Binario87;      compiladoBinario=compiladoBinario+Binario88;
                compiladoBinario=compiladoBinario+Binario89;      compiladoBinario=compiladoBinario+Binario90;
                compiladoBinario=compiladoBinario+Binario91;      compiladoBinario=compiladoBinario+Binario92;
                compiladoBinario=compiladoBinario+Binario93;      compiladoBinario=compiladoBinario+Binario94;
                compiladoBinario=compiladoBinario+Binario95;      compiladoBinario=compiladoBinario+Binario96;
                compiladoBinario=compiladoBinario+Binario97;      compiladoBinario=compiladoBinario+Binario98;
                compiladoBinario=compiladoBinario+Binario99;      compiladoBinario=compiladoBinario+Binario100;

                int compiladoBin = Integer.parseInt(compiladoBinario);

                if(compiladoBin==12222222){
                    letra="a";
                    usado=palabras+letra;
                    palabras=usado;
                    leer=palabras;
                    pantalla.setText(""+palabras);
                    Binario1="2"; Binario2="2"; Binario3="2"; Binario4="2"; Binario5="2"; Binario6="2"; Binario7="2";
                }
            }
        });
    }
}
```

Código 14 Modelo Algoritmo Genético con Búsqueda Local Tabu
Fuente: Implementación propia

Cabe resaltar que la base de datos se la hace dentro de la APK, la cual se la efectúa mediante una clase de JAVA la cual se muestra en las líneas de **Código 15** y **16** junto con la clase que lo observara.

```
BD.java
package com.menu.sqlcharly;

import android.content.ContentValues;

public class BD{

    public static final String ID_FILA = "_id";
    public static final String ID_DATOS = "_datos";

    private static final String N_BD = "Datos";
    private static final String N_TABLA = "Tabla_datos";
    private static final int VERSION_BD =1;

    private BDHelper nHelper;
    private final Context nContexto;
    private SQLiteDatabase nBD;

    private static class BDHelper extends SQLiteOpenHelper{
    public BD(Context c){
    public BD abrir ()throws Exception{
    public void cerrar() {}
    public long crearEntrada(String nom) {}
    public String recibir() {}
    //busca la informacion y lo regresa del nombre//
    public String getN(Long lb) {}

    public void editar(long eFilalong, String eNom) throws SQLException{}
    public void borrar(long eliFilalong) throws SQLException{}
    }
}
```

Código 15: Base de datos de la APK
Fuente: Implementación propia

```
SQLver.java
package com.menu.sqlcharly;

import android.app.Activity;

public class SQLver extends Activity{

    TextView texto;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        setContentView(R.layout.ver);

        texto=(TextView) findViewById(R.id.tvTexto);

        BD info = new BD(this);

        try{
            info.abrir();
        }catch(Exception e){
            e.printStackTrace();
        }
        String datos=info.recibir();
        info.cerrar();
        texto.setText(datos);
    }
}
```

Código 16: Clase que observa la Base de Datos
Fuente: Implementación propia

CAPITULO IV

IDEA A DEFENDER

En el capítulo III (Marco aplicativo) se observa la formulación de la propuesta y el desarrollo de un prototipo en base a la propuesta. En este capítulo lo que se ve es la confrontación de la idea a defender, es decir la prueba del prototipo para afirmar o refutar la misma.

4. Introducción

Al probar el prototipo realizado en base a la propuesta, se afirmó la idea a defender (**Figura 59**). Las pruebas fueron desarrolladas en cuatro fases, dentro de cada fase existen procedimientos, tareas.

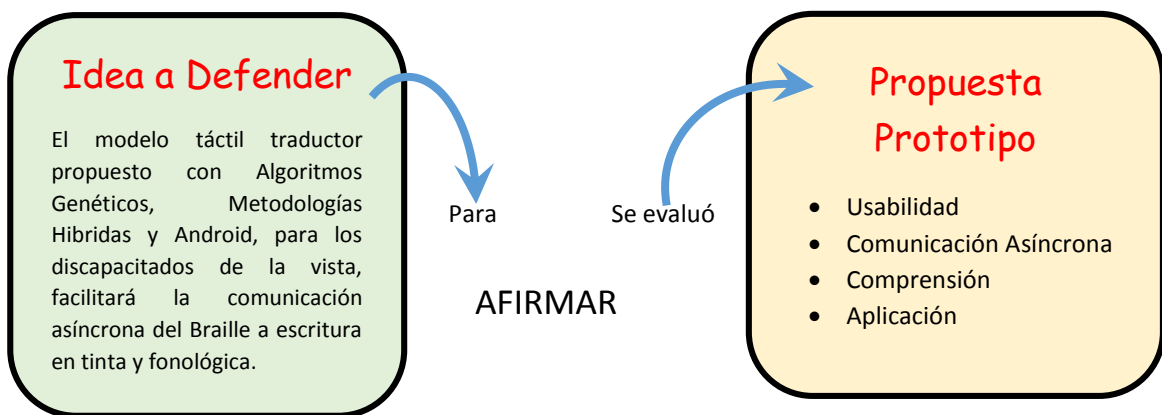


Figura 59: Relación idea a defender propuesta

Fuente: Elaboración propia

4.1. Pruebas al prototipo

Al probar el prototipo realizado en base a la propuesta, se afirmó la idea a defender. Las pruebas fueron desarrolladas en cuatro fases, dentro de cada fase existen procedimientos, tareas.

4.1.1. Fase 1 - Planeación

En esta fase se definió que criterios deberían ser evaluados para comprobar la hipótesis

4.1.1.1. Definición de Pruebas: Usabilidad

En el proceso de desarrollo de un sistema, producto, apk o servicio está siempre involucrado a la palabra usabilidad, lo primero que llega a la mente al escuchar este término es el potencial o las posibilidades de uso de un producto, entonces ¿qué es la usabilidad?, ¿qué la hace tan importante en el proceso de desarrollo?, ¿qué beneficios trae?

CAPÍTULO IV - IDEA A DEFENDER

De acuerdo a la Organización Internacional para la Estandarización (ISO) usabilidad puede definirse de dos formas (ISO, 2015):

- La usabilidad se refiere a la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso (ISO/IEC_9126).
- Usabilidad es la efectividad, eficiencia y satisfacción con la que un producto permite alcanzar objetivos específicos a usuarios específicos en un contexto de uso específico (ISO/IEC 9241).

En ambas definiciones se liga la usabilidad de un sistema a usuarios, necesidades y condiciones específicas, por lo tanto, la usabilidad del sistema no es un atributo del producto. De igual forma el producto sólo tendrá la capacidad de ser usado en un contexto particular y por usuarios particulares. En otras palabras, la usabilidad no puede ser valorada estudiando un producto de manera aislada. (Ferré Grau).

Para evaluar la usabilidad hay que tener un contexto particular (Gabriel Gerónimo Castillo). También es una cualidad demasiado abstracta como para ser medida directamente. Para poder estudiarla se descompone habitualmente en los siguientes atributos básicos (Ferré Grau).

Eficiencia de uso: La eficiencia se refiere al número de tareas que el usuario puede realizar en el sistema en un tiempo determinado. En este atributo de la usabilidad lo que se busca es la máxima velocidad y facilidad de realización de tareas del usuario. Cuanto mayor es la usabilidad de un sistema, más rápido es el usuario al utilizarlo, y el trabajo se realiza con mayor rapidez.

Recuerdo en el tiempo: Los usuarios que no utilizan la APK regularmente deben ser capaces de usar sin tener que aprender cómo funciona partiendo de cero cada vez. Este atributo refleja el recuerdo de los usuarios acerca de cómo funciona.

Tasa de errores: Este atributo contribuye de forma negativa a la usabilidad de una APK, se refiere al número de errores cometidos mientras realiza una determinada tarea. Un buen nivel de usabilidad implica una tasa de errores baja. Los errores reducen la eficiencia y satisfacción del usuario, y pueden verse como un fracaso en la explicación del modo de hacer las cosas en el sistema.

CAPÍTULO IV - IDEA A DEFENDER

Satisfacción: La APK debe ser placentera de usar. Este atributo es subjetivo, ya que es la impresión que se obtiene de la APK.

4.1.1.2. Definición de pruebas: Herramienta didáctica

Las características de una herramienta didáctica son:

- Ser adecuado al tema o asunto de la clase.
- Ser de fácil manejo y comprensión.
- Estar en perfectas condiciones de funcionamiento, sobre todo en el caso de equipos, ya que nada divierte y dispersa más que las fallas en las demostraciones.

En este sentido, los criterios que se evaluaron para saber si una herramienta es o no una herramienta didáctica son:

Contextual.- Esto quiere decir, que antes de proponer una herramienta didáctica hay que velar porque sea accesible.

Adecuación.- Si la herramienta se adecua o no a la comunicación asíncrona.

Fácil manejo.- Si es difícil aprender a utilizar la herramienta y requiere una morosa preparación o no.

Funcionamiento.- Esto se evalúa al evaluar la usabilidad de prototipo.

4.1.1.3. Definición de pruebas: Contenido

Para evaluar el contenido, lo primero que hay que se tomó en cuenta es que “se lo realizo en base al proyecto formativo”. Una vez tomado en cuenta lo anterior, lo que se evaluó es:

- **Conceptos claros.-** Si los conceptos presentados son claros o difíciles de entender.
- **Imágenes Claras.-** Si las imágenes son visibles o difíciles de visualizar.
- **Coherencia de imágenes y texto.-** Si existe coherencia en los contenidos, es decir los textos y las imágenes mostradas están ordenadas de manera coherente o no.

CAPÍTULO IV - IDEA A DEFENDER

4.1.1.4. Definición de pruebas: Comprensión y aplicación

Existen dos maneras obtener resultados sobre esta tipo de pruebas.

Primero.- Una vez realizadas las pruebas anteriores, se obtiene suficiente información para saber si el prototipo ayuda en la comprensión o no.

Segundo.- El prototipo evalúa la comprensión:

- El prototipo le ayuda a manipular la APK.
- El prototipo tiene la opción de interactuar con el discapacitado con el fin de coadyuvar.
- El usuario adquirió la habilidad de plasmar sus pensamientos y expresarlos de forma asíncrona.

Hechas estas pruebas y obteniendo los resultados se afirmara o refutara la hipótesis

4.1.2. Fase 2 - Preparación

Una vez definido los criterios a evaluar, se preparan instrumentos y se coordinan las pruebas.

4.1.2.1. Reuniones de coordinación

En estas reuniones participaron el M. Sc. Lic. Juan Carlos Huanca Guanca, Ing. Juan Gabriel Lazcano Balanza y Unv. Carlos Alberto Osco Flores.

El propósito de estas reuniones fue:

- Mostrar el prototipo al tutor y al asesor
- Escuchar las sugerencias del tutor y el asesor para mejorar la investigación (prototipo, documentos, instrumentos de evaluación).
- Definir las fechas y el lugar de las pruebas.
- Definir la metodología para las pruebas.
- Mostrar los posibles instrumentos de evaluación de pruebas.

En estas reuniones se determinó realizar las pruebas orientado a los discapacitados de la vista. En el mismo se utilizó el prototipo y así se comprobaría la Idea a defender.

4.1.2.2. Definir roles

Se definieron tres roles para las pruebas:

Facilitador.-

Es la persona que va dirigir las pruebas (el postulante). Esta persona tiene un papel muy importante dentro de la prueba, ya que es la encargada de guiar a los usuarios como deben de manipular la APK. El facilitador debe de tener una idea clara de la totalidad del prototipo y saber cómo resolver los errores que puede presentar.

Observador.-

Es la persona que conoce el proyecto de grado y prototipo a evaluar (asesor y/o tutor). Debe tener una idea completa de las funciones que los usuarios van a desarrollar en el transcurso de la prueba. Su función es prestar atención a las acciones, comportamientos, gestos, emociones y otras características que los usuarios presenten en la realización de cada una de las tareas que desarrolla durante las pruebas.

Usuario.-

Este grupo está compuesto por las personas que van a probar el prototipo durante todo el transcurso de la prueba (para luego afirmar o refutar la hipótesis). Para obtener buenos resultados dentro la prueba son suficientes una cantidad reducida de usuarios. Este grupo debe estar compuesto por usuarios reales del prototipo a evaluar para no obtener datos erróneos.

4.1.2.3. Preparación de métodos e instrumentos para evaluar

Los métodos que se decidieron utilizar para evaluar el prototipo fueron:

Tareas para pruebas de uso.- Se lo realizo con celulares con plataforma Android, permite conocer la usabilidad que tiene el prototipo.

Criterios de observación.- Permite captar facialmente las características francas y concretas que los usuarios adoptan con respecto a la satisfacción. Se elaboraron un conjunto de criterios a observar en el transcurso de las pruebas.

Criterios de entrevistas.- Las entrevistas orales establecen aspectos fundamentales para los desarrolladores.

Todos estos instrumentos permitieron comprobar la idea a defender.

4.1.2.4. Selección de muestra

En este segmento lo que hace es definir la muestra con la que se probara el prototipo, entonces, primero se define la población, luego cuantas personas necesitamos para LA EXPERIMENTACIÓN o comprobación de la idea a defender.

Como en las reuniones de coordinación se determinó realizar las pruebas en el centro de Discapacitados MIKI MAYA, se decidió que las personas que realicen la prueba fueron personal capacitado y discapacitados en persona.

4.1.3. Fase 3 – Desarrollo de Pruebas

Finalizada la etapa de construcción se requiere proceder con un conjunto de pruebas necesarias para verificar el correcto funcionamiento de la APK.

4.1.3.1. Pruebas de Caja Blanca

Estas tienen como objetivo probar estatutos condicionales, bucles y ciclos de la aplicación.

A continuación, se analizará a modo de ejemplo el método que realiza la lectura e interpretación del contenido un código dentro de las clases.

Clase Splash

Clase Splash		Grafo
1	public class Splash extends Activity {	
2	protected void onCreate(Bundle savedInstanceState) {	
3	setContentView(R.layout.splash);	
4	Thread logouap = new Thread();	
5	public void run(){	
6	try{	
7	int tiempo = 0;	
8	while(tiempo<5000){ sleep(100);	
9	tiempo = tiempo + 100; }	
10	Intent a = new Intent("com.charly.charly.menulegal");	
11	startActivity(a);	
12	}catch(InterruptedException e){	
13	e.printStackTrace();	
14	}finally{	
15	finish();	
16	}; }	
17	logouap.start();	
18	}	
19	}	

Tabla 12: Prueba de caja blanca, Splash
Fuente: Elaboración propia

Los resultados de la prueba de caja blanca aplicada sobre la clase anteriormente mencionada se muestran en las siguientes tablas:

Nº	Caminos
1	1-2-4-5-6-8-11
2	1-2-4-5-6-12
3	1-2-4
Resultados	
Nº de nodos :8	
Nº de aristas : 10	
Complejidad ciclomatica : 3	

Tabla 13: Resultado de caja blanca, Splash
Fuente: Elaboración propia

Nº	Camino	Caso Prueba	Resultado Esperado	Resultado Real
1	1-2-4-5-6-8-11	No error Sistema Android	Define y ejecuta la aplicación	Define y ejecuta la aplicación
2	1-2-4-5-6-12	Error capacidad Sistema Android	Define y ejecuta la aplicación	Define y en ejecución se cuelga la APK
3	1-2-4	Error versión Sistema Android	Define y ejecuta la aplicación	No ingresa a la APK

Tabla 14: Caso de prueba caja blanca, Splash
Fuente: Elaboración propia

Clase Menú Principal

Clase Menu Principal		Grafo
1	public class Menu extends Activity{	Splash
2	public MediaPlayer mp;	1
3	Intent a = new Intent("com.charly.charly.menulegal");	5
4	startActivity(a);	6
5	setContentView(R.layout.menu_principal); play_mp(); }	19
6	private void play_mp() {	22
7	mp= MediaPlayer.create(this, R.raw.presentador_menuspri mp.start());	25
8	private void stop_mp() { mp.stop(); }	28
9	Intent contactos_ = new Intent(this,Menu_contactos.class);	31
10	startActivity(contactos_); }	34
11	public void Cerrar_(View v){	11
12	AlertDialog.Builder dialog = new AlertDialog.Builder(this);	
13	dialog.setMessage("¿Salir?");	
14	public void onClick(DialogInterface dialog, int which) {	
15	stop_mp(); System.exit(0); } });	
16	dialog.setNegativeButton("No", new DialogInterface.OnClickListener() {	
17	dialog.cancel(); } });	
18	dialog.show(); }	
19	public void TecladoBraille_(View v){	
20	Intent tecladobrilleg_ = new Intent(this,Menu_brilleg.class);	
21	startActivity(tecladobrilleg_); stop_mp(); Menu.this.finish();}	
22	public void Teclado_(View v){	
23	Intent teclado_ = new Intent(this,Menu_tecladotranscriptor.class);	
24	startActivity(teclado_); stop_mp(); Menu.this.finish();}	
25	public void Diccionarios_(View v){	
26	Intent diccionarios_ = new Intent(this,Menu_diccionarios.class);	
27	startActivity(diccionarios_); stop_mp(); Menu.this.finish();}	
28	public void Vocabulario_(View v){	
29	Intent vocabulario_ = new Intent(this,Menu_vocabularioabecedario.class);	
30	startActivity(vocabulario_); stop_mp(); Menu.this.finish();}	
31	public void Manual_(View v){	
32	Intent manual_ = new Intent(this,Menu_manualusuario.class);	
33	startActivity(manual_); stop_mp(); Menu.this.finish();}	
34	public void Guardados_(View v){	
35	Intent guardados_ = new Intent(this,Menu_documentosguardados.class);	
36	startActivity(guardados_); stop_mp(); Menu.this.finish();}	
37	}	

Tabla 15; Prueba de caja blanca, Menú principal
Fuente: Elaboración propia

CAPÍTULO IV - IDEA A DEFENDER

Los resultados de la prueba de caja blanca aplicada sobre la clase del **Menú Principal** se muestran en las siguientes tablas:

Nº	Caminos
1	1-5-6-5-19
2	1-5-6-5-22
3	1-5-6-5-25
4	1-5-6-5-28
5	1-5-6-5-31
6	1-5-6-5-34
7	1-5-6-5-11
Resultados	
Nº de nodos :10	
Nº de aristas : 17	
Complejidad ciclomatica : 7	

Tabla 16: Resultados de caja blanca, Menú principal
Fuente: Elaboración propia

Nº	Camino	Caso Prueba	Resultado Esperado	Resultado Real
1	1-5-6-5-19	No error APK	Define y ejecuta el cambio de Clases	Define y ejecuta el cambio de Clases
2	1-5-6-5-22	No error APK	Define y ejecuta el cambio de Clases	Define y ejecuta el cambio de Clases
3	1-5-6-5-25	No error APK	Define y ejecuta el cambio de Clases	Define y ejecuta el cambio de Clases
4	1-5-6-5-28	No error APK	Define y ejecuta el cambio de Clases	Define y ejecuta el cambio de Clases
5	1-5-6-5-31	No error APK	Define y ejecuta el cambio de Clases	Define y ejecuta el cambio de Clases
6	1-5-6-5-34	No error APK	Define y ejecuta el cambio de Clases	Define y ejecuta el cambio de Clases
7	1-5-6-5-11	No error APK	Define y ejecuta el cambio de Clases	Define y ejecuta el cambio de Clases

Tabla 17: Caso de prueba caja blanca, Menú principal
Fuente: Elaboración propia

Clase Menú Braille

Clase Menu Braille		Grafo
1	public class Menu_braille extends Activity implements OnInitListener{	
2	String Binario1="2"; String Binario2="2"; String Binario3="2";	
3	String Binario4="2"; String Binario5="2"; String Binario6="2";	
4	protected void onCreate(Bundle savedInstanceState) {	
5	final Button B1=(Button)findViewById(R.id.Bot1);	
6	B1.setOnClickListener(new OnClickListener(){ Binario1="1"; })	
7	aceptar = (Button) findViewById(R.id.Aceptar);	
8	aceptar.setOnClickListener(new View.OnClickListener() {	
9	compiladoBinario;	
10	if(compiladoBin==12222222){ letra="A";	
11	pantalla.setText(""+palabras); Binario1="2"; Binario6="2"; }	
12	if(compiladoBin==11121122){ letra="A";	
13	pantalla.setText(""+palabras); Binario1="2"; Binario6="2"; }	
14	if(compiladoBin==12222212){ letra="1";	
15	pantalla.setText(""+palabras); Binario1="2"; Binario6="2"; }	
16	cancelar = (Button) findViewById(R.id.Cancelar);	
17	cancelar.setOnClickListener(new View.OnClickListener() { Binario1="2"; }); }	
18	reproduccion = (Button) findViewById(R.id.Replee);	
19	dime_algo(pantalla.getText().toString());	
20	private void dime_algo(String texto) {	
21	x.speak(texto, TextToSpeech.QUEUE_FLUSH, null);	
22	public void onDestroy() { if (x != null) { x.stop(); x.shutdown(); } }	
23	public void B (View v){ Intent b_ = new Intent(this,com.charly.charly.Menu.class);	
24	startActivity(b_); Menu_braille.this.finish();	

Tabla 18: Prueba de caja blanca, Menú Braille
Fuente: Elaboración propia

Los resultados de la prueba de caja blanca aplicada sobre la clase del **Menú Braille** se muestran en las siguientes tablas:

Nº	Caminos
1	1-4-6-TB-C-17-TB
2	1-4-6-TB-C-10-20-18-23
3	1-4-6-TB-C-12-20-18-23
4	1-4-6-TB-C-14-20-18-23
Resultados	
Nº de nodos :12	
Nº de aristas : 21	
Complejidad ciclomatica : 4	

Tabla 19: Resultados de caja blanca, Menú Braille
Fuente: Elaboración propia

CAPÍTULO IV - IDEA A DEFENDER

Nº	Camino	Caso Prueba	Resultado Esperado	Resultado Real
1	1-4-6-TB-C-17-TB	No error APK	Define y retorna valores erróneos	Define y retorna valores erróneos
2	1-4-6-TB-C-10-20-18-23	No error APK	Define y ejecuta el Abecedario con expresión fonológica	Define y ejecuta el Abecedario con expresión fonológica
3	1-4-6-TB-C-12-20-18-23	No error APK	Define y ejecuta el Vocabulario con expresión fonológica	Define y ejecuta el Vocabulario con expresión fonológica
4	1-4-6-TB-C-14-20-18-23	No error APK	Define y ejecuta los números con expresión fonológica	Define y ejecuta los números con expresión fonológica

Tabla 20: Caso de prueba caja blanca, Menú Braille

Fuente: Elaboración propia

Clase Menú Alfa-Numérico

Clase Menu Teclado Alfa-Numerico	Grafo
1 public class Menu_tecladotranscriptor extends Activity implements OnInitListener{	
2 protected void onCreate(Bundle savedInstanceState) {	
3 setContentView(R.layout.menu_tecladotranscriptor);	
4 botonreproduccion.setOnClickListener(new View.OnClickListener() {	
5 public void onClick(View arg0) { dime_algo(textoleido.getText().tc)	
6 dime_algo(textoleido.getText().toString());	
7 } });	
8 public void onInit(int status) {	
9 if (status == TextToSpeech.SUCESS) {	
10 int result = xxx.setLanguage(Locale.getDefault());	
11 if (result == TextToSpeech.LANG_MISSING_DATA result == TextToSpeech.LANG_NOT_SUPPORTED) {	
12 botonreproduccion.setEnabled(false);	
13 Log.e("TTS", "This Language is not supported");	
14 } else { botonreproduccion.setEnabled(true); }	
15 } else { Log.e("TTS", "Initialization Failed!"); }	
16 private void dime_algo(String texto) {	
17 xxx.speak(texto, TextToSpeech.QUEUE_FLUSH, null);	
18 }	
19 public void onDestroy() { if (xxx != null) { xxx.stop(); xxx.shutdown(); }	
20 super.onDestroy();	
21 }	
22 public void E(View v){ Intent e = new Intent(this, com.charly.charly.Menu.class);	
23 startActivity(e); Menu_tecladotranscriptor.this.finish();	
24 }	
25 }	

Tabla 21: Prueba de caja blanca, Menú Alfa-Numérico

Fuente: Elaboración propia

CAPÍTULO IV - IDEA A DEFENDER

Los resultados de la prueba de caja blanca aplicada sobre la clase del **Menú Alfa-Numérico** se muestran en las siguientes tablas:

Nº	Caminos
1	1-3-8-16-19
2	1-3-8-22
Resultados	
Nº de nodos :6	
Nº de aristas : 7	
Complejidad ciclomatica : 2	

*Tabla 22: Resultados de caja blanca, Menú Alfa-Numérico
Fuente: Elaboración propia*

Nº	Camino	Caso Prueba	Resultado Esperado	Resultado Real
1	1-3-8-16-19	No error APK	Define y ejecuta con expresión fonológica	Define y ejecuta con expresión fonológica
2	1-3-8-22	No error APK	Define y ejecuta retornando al Menú Principal	Define y ejecuta retornando al Menú Principal

*Tabla 23: Caso de prueba caja blanca, Menú Alfa-Numérico
Fuente: Elaboración propia*

Clase Menú VocAbeNum

Clase Menu VocAbeNum		Grafo
1	public class Menu_vocabularioabecedario extends Activity {	
2	protected void onCreate(Bundle savedInstanceState) {	
3	setContentView(R.layout.menu_vocabularioabecedario);	
4	}	
5	public void MenuPrincipal_(View v){	
6	Intent menuprincipal_ = new Intent(this,com.charly.charly.Menu.class);	
7	startActivity(menuprincipal_);	
8	Menu_vocabularioabecedario.this.finish(); }	
9	public void Abecedario_(View v){	
10	Intent abecedario_ = new Intent(this,com.charly.charly.X_abecedario_a.class);	
11	startActivity(abecedario_); }	
12	public void Vocabulario_(View v){	
13	Intent vocabulario_ = new Intent(this,com.charly.charly.Y_vocabulario_a.class);	
14	startActivity(vocabulario_); }	
15	public void Numeros_(View v){	
16	Intent numeros_ = new Intent(this,com.charly.charly.Z_numeros_uno.class);	
17	startActivity(numeros_);	
18	} }	

*Tabla 24: Prueba de caja blanca, Menú VocAbeNum
Fuente: Elaboración propia*

Los resultados de la prueba de caja blanca aplicada sobre la clase del Menú VocAbeNum se muestran en las siguientes tablas:

Nº	Caminos
1	1-3-9-3-5
2	1-3-12-3-5
3	1-3-15-3-5
Resultados	
Nº de nodos :6	
Nº de aristas : 13	
Complejidad ciclomatica : 3	

*Tabla 25: Resultados de caja blanca, Menú VocAbeNum
Fuente: Elaboración propia*

Nº	Camino	Caso Prueba	Resultado Esperado	Resultado Real
1	1-3-9-3-5	No error APK	Define y ejecuta el Abecedario	Define y ejecuta el Abecedario
2	1-3-12-3-5	No error APK	Define y ejecuta el Vocabulario	Define y ejecuta el Vocabulario
3	1-3-15-3-5	No error APK	Define y ejecuta los Números	Define y ejecuta los Números

*Tabla 26: Caso de prueba caja blanca, Menú VocAbeNum
Fuente: Elaboración propia*

Clase Menú Diccionarios

Clase Menu Diccionarios		Grafo
1	public class Menu_diccionarios extends Activity {	
2	protected void onCreate(Bundle savedInstanceState) {	
3	setContentView(R.layout.menu_diccionarios);	
4	}	
5	public void MenuPrincipal_ (View v){	
6	Intent menuprincipal_ = new Intent(this,com.charly.charly.Menu.class);	
7	startActivity(menuprincipal_); Menu_diccionarios.this.finish();	
8	}	
9	public void DiccionariosSimbolos_ (View v){	
10	Intent diccionariossimbolos_ = new Intent(this,Menu_dicsim.class);	
11	startActivity(diccionariossimbolos_); Menu_diccionarios.this.finish();	
12	}	
13	public void DicCharly_ (View v){	
14	Intent diccharly_ = new Intent(this,Menu_diccharly.class);	
15	startActivity(diccharly_); Menu_diccionarios.this.finish();	
16	}	

Tabla 27: Prueba de caja blanca, Menú Diccionarios
Fuente: Elaboración propia

Los resultados de la prueba de caja blanca aplicada sobre la clase del **Menú Diccionarios** se muestran en las siguientes tablas:

Nº	Caminos
1	1-3-9-3-5
2	1-3-13-3-5
Resultados	
Nº de nodos :5	
Nº de aristas : 9	
Complejidad ciclomatica : 2	

Tabla 28: Resultados de caja blanca, Menú Diccionarios
Fuente: Elaboración propia

Nº	Camino	Caso Prueba	Resultado Esperado	Resultado Real
1	1-3-9-3-5	No error APK	Define y ejecuta el Diccionario de Símbolos	Define y ejecuta el Diccionario de Símbolos
2	1-3-13-3-5	No error APK	Define y ejecuta el Diccionario de Charly	Define y ejecuta el Diccionario de Charly

Tabla 29: Caso de prueba caja blanca, Menú Diccionarios
Fuente: Elaboración propia

Clase Menú Diccionario Símbolos

Clase Menu Diccionario Simbolos		Grafo
1	public class Menu_dicsim extends Activity {	
2	protected void onCreate(Bundle savedInstanceState) {	
3	setContentView(R.layout.menu_dicsim);	
4	}	
5	public void MenuPrin_(View v){	
6	Intent menuprincipal_ = new Intent(this,com.charly.charly.Menu.class);	
7	startActivity(menuprincipal_); Menu_dicsim.this.finish();	
8	}	
9	public void intro_(View v){	
10	Intent intro_ = new Intent(this,com.charly.charly.Dic_intro.class);	
11	startActivity(intro_);	
12	public void Dic_(View v){	
13	Intent dic_ = new Intent(this,com.charly.charly.Dic_dic.class);	
14	startActivity(dic_);	
15	}	
16	public void Acti_(View v){	
17	Intent acti_ = new Intent(this,com.charly.charly.Dic_acti.class);	
18	startActivity(acti_);	
19	} }	

Tabla 30: Prueba de caja blanca, Menú Diccionario Símbolos
 Fuente: Elaboración propia

Los resultados de la prueba de caja blanca aplicada sobre la clase del **Menú Diccionario Símbolos** se muestran en las siguientes tablas:

Nº	Caminos
1	1-3-9-3-5
2	1-3-12-3-5
3	1-3-16-3-5
Resultados	
Nº de nodos :6	
Nº de aristas : 13	
Complejidad ciclomatica : 3	

Tabla 31: Resultados de caja blanca, Menú Diccionario Símbolos
 Fuente: Elaboración propia

Nº	Camino	Caso Prueba	Resultado Esperado	Resultado Real
1	1-3-9-3-5	No error APK	Define y ejecuta la introducción al Diccionario	Define y ejecuta la introducción al Diccionario
2	1-3-12-3-5	No error APK	Define y ejecuta el Diccionario en si	Define y ejecuta el Diccionario en si
2	1-3-16-3-5	No error APK	Define y ejecuta las actividades didácticas en si	Define y ejecuta las actividades didácticas en si

Tabla 32: Caso de prueba caja blanca, Menú Diccionario Símbolos
 Fuente: Elaboración propia

Clase ejemplo Image Adapter

Ejemplo de clase - Image Adapter		Grafo
1	public class ImageAdapterAbecedario extends PagerAdapter{	
2	Context context;	
3	private int[] GalAbe = new int[] { R.drawable.a, R.drawable.z };	
4	ImageAdapterAbecedario(Context context){ this.context=context;}	
5	public int getCount() { return GalAbe.length; }	
6	public boolean isViewFromObject(View view, Object object) {	
7	return view == ((ImageView) object);}	
8	public Object instantiateItem(ViewGroup container, int position) {	
9	ImageView imageView = new ImageView(context);	
10	((ViewPager) container).addView(imageView, 0);}	
11	public void destroyItem(ViewGroup container, int position, Object object) {	
12	((ViewPager) container).removeView((ImageView) object);}	

*Tabla 33: Prueba de caja blanca, ejemplo Image Adapter
Fuente: Elaboración propia*

Los resultados de la prueba de caja blanca aplicada sobre el ejemplo de **Image Adapter** se muestran en las siguientes tablas, solo se muestra un ejemplo porque son varias las veces que se utiliza el **Image Adapter** con su vector de caracteres respectivo:

Nº	Caminos
1	1-4-8-3-GAL-11
Resultados	
Nº de nodos : 6	
Nº de aristas : 6	
Complejidad ciclomatica : 1	

*Tabla 34: Resultados de caja blanca, ejemplo Image Adapter
Fuente: Elaboración propia*

Nº	Camino	Caso Prueba	Resultado Esperado	Resultado Real
1	1-4-8-3-GAL-11	No error APK	Define y ejecuta la galería de caracteres seleccionada en orden secuencial	Define y ejecuta la galería de caracteres seleccionada en orden secuencial

*Tabla 35: Caso de prueba caja blanca, ejemplo Image Adapter
Fuente: Elaboración propia*

CAPÍTULO IV - IDEA A DEFENDER

4.1.3.2. Pruebas de caja negra

Las pruebas de caja negra consisten en probar el teclado con datos válidos como no válidos y observar si el comportamiento del módulo es el adecuado. Con el fin de ejemplificar estas pruebas se mostrará el caso correspondiente a la creación de símbolos que no corresponden al alfabeto Braille.

Para realizar estas pruebas es necesario definir la *Tabla 38* de equivalencia.

Condición de Entrada	Tipo	Clase Equivalente Valida	Clase Equivalente Invalida
B1	Carácter	Carácter que cumpla con la condición	Carácter que no cumpla con la condición o en blanco
B2	Carácter	Carácter que cumpla con la condición	Carácter que no cumpla con la condición o en blanco
B3	Carácter	Carácter que cumpla con la condición	Carácter que no cumpla con la condición o en blanco
B4	Carácter	Carácter que cumpla con la condición	Carácter que no cumpla con la condición o en blanco
B5	Carácter	Carácter que cumpla con la condición	Carácter que no cumpla con la condición o en blanco
B6	Carácter	Carácter que cumpla con la condición	Carácter que no cumpla con la condición o en blanco
# (Núm.)	Cadena de caracteres	Cadena de caracteres que si cumplan con su condición	-
_ (Esp.)	Cadena de caracteres	Cadena de caracteres que si cumplan con su condición	-

*Tabla 36: Equivalencias en el teclado Braille
Fuente: Elaboración propia*

Tomando en cuenta que la interfaz gráfica no permite el ingreso de campos fuera de la *APK*, que no cumplan el formato mejorando de esta forma su calidad.

Las pruebas ejecutadas fueron de utilidad para poder depurar algunos errores encontrados durante este proceso de pruebas.

4.1.3.3. Reunión de coordinación con los Usuarios

Luego de haber efectuado las pruebas de caja blanca y negra limitando de esta forma los errores posteriores se procedió a efectuar reuniones con los Usuarios y Personal capacitado del mismo al cual se informó hacer del trabajo de investigación, aclarando dudas, se hizo conocer los beneficios que ellos obtendrían al utilizar esta APK, y el papel que se espera que la misma influya en su vida usándola continuamente, lo cual se puede apreciar en las *Figuras 60 y 61*.



Figura 60: Reunión con Dir. Discapacitados de la vista
Fuente: Elaboración propia



Figura 61: Reunión con la especialista interprete de los Sordos
Fuente: Elaboración propia

4.1.4. Fase 4 - Terminación de las Pruebas

4.1.4.1. Pruebas con prototipo

Se realizaron la pruebas del prototipo, para esto se utilizaron distintos dispositivos móviles, con variedad de sistemas operativos demostrando de esta forma que la APK es independiente del sistema operativo móvil.

Primero se explicó cómo funciona el prototipo para luego ellos poder utilizarlo. Lo anteriormente mencionado de lo puede apreciar en las *Figuras 62 – 63 – 64*.



Figura 62: Aplicando el funcionamiento de la APK 1
Fuente: Elaboración propia



Figura 63: Explicando el funcionamiento de la APK 2
Fuente: Elaboración propia



Figura 64: Explicando el funcionamiento de la APK 3
Fuente: Elaboración propia

Una vez explicado cómo funciona la APK “prototipo”, se les solicitó encarecidamente que manipulen dispositivos móviles con la APK para efectuar la prueba práctica (*Figuras 65 -66 - 67*).



Figura 65: Pruebas prácticas 1
Fuente: Elaboración propia



Figura 66: Pruebas prácticas 2
Fuente: Elaboración propia



Figura 67: Pruebas prácticas 3
Fuente: Elaboración propia

También se les permitió navegar en las otras opciones: literatura, diccionarios, manuales y otros ver Anexo D.

4.1.5. Conclusiones de resultados obtenidos

Terminadas las pruebas de usabilidad se pudo observar que: **La APK es amistosa, no compleja y de fácil manipulación adecuándose al usuario.**

Basados en la norma de calidad ISO/IEC 9126 se elaboró un cuestionario de evaluación estableciendo la forma de medir la calidad de la APK. Se ha asignado un peso (valor numérico) a cada atributo de acuerdo al grado de importancia o nivel de necesidad exigida, va desde el 1 que significa poco importante hasta el 5 que significa extremadamente importante ajustando la misma a cada una de las preguntas a un atributo exigido, con estos resultados se aplicó una fórmula las métricas de cada uno de ellos, el usuario califica de acuerdo a la siguiente escala de la **Tabla 38**.

<i>Calificación</i>	<i>Puntaje</i>
Muy malo	1
Pésimo	2
Regular	3
Bueno	4
Muy Bueno	5

Tabla 37: Escala de calificación para la APK
Fuente: Elaboración propia

CAPÍTULO IV - IDEA A DEFENDER

4.1.5.1. Usabilidad del software

Las pruebas de usabilidad de efectuaron mediante un conjunto de opciones que efectuó el usuario dentro de la APK, después de la manipulación se procedió a evaluar la misma, los resultados se muestran en la **Tabla 39**.

Criterio	ITEM	USUARIOS			TOTAL	PROMEDIO
		1	2	3		
USABILIDAD	Facilidad de comunicación Asíncrona					
	1. Es costoso aprender la APK	4	3	4	11	3,66
	2. Es posible la manipulación automática	4	4	5	13	4,33
	Eficiencia de uso					
	3. No se colgó mientras se manipulaba	5	5	5	15	5
	4. Funciona la APK de forma rápida	4	3	3	10	3,33
	Tasa de errores					
	5. No genera errores mientras se instala	5	5	5	15	5
	6. Salen errores en su manipulación	5	5	5	15	5
	Satisfacción					
	7. Se adecua a su entorno	3	4	3	10	3,33
	8. Es de fácil manejo y comprensión los módulos de la APK	4	4	5	13	4,33
9. La organización de los menús, están en buenas condiciones de funcionamiento y adecuadas a usted	4	5	4	13	4,33	
						4,25

Tabla 38 Resultados de usabilidad
Fuente: Elaboración propia

Terminadas las pruebas de usabilidad de pudo evidenciar que: **La APK cumple con los criterios de usabilidad, es decir que es adecuada para el usuario.**

CAPÍTULO IV - IDEA A DEFENDER

4.1.5.2. Herramienta didáctica

Las pruebas de usabilidad contribuyen con los resultados esperados:

- Se pudo observar que un dispositivo móvil es una herramienta de uso cotidiano y personal que toda persona tiene o planea tener.
- La **APK** no es difícil de manipular.
- No genera errores.

Criterio	ITEM	USUARIOS			TOTAL	PROMEDIO
		1	2	3		
CONTENIDO	El contenido					
	1. Es costoso aprender la APK	4	3	4	11	3,66
	2. Es posible la manipulación automática	4	4	5	13	4,33
	Eficiencia de uso					
	3. No se colgó mientras se manipulaba	5	5	5	15	5
	4. Funciona la APK de forma rápida	4	3	3	10	3,33
	Tasa de errores					
	5. No genera errores mientras se instala	5	5	5	15	5
	6. Salen errores en su manipulación	5	5	5	15	5
	Satisfacción					
	7. Se adecua a su entorno	3	4	3	10	3,33
	8. Es de fácil manejo y comprensión los módulos de la APK	4	4	5	13	4,33
9. La organización de los menús, están en buenas condiciones de funcionamiento y adecuadas a usted	4	5	4	13	4,33	
					4,25	

Tabla 39 Resultados de contenido

Fuente: Elaboración propia

Una vez analizados los cuestionarios, entrevistas y datos de observación se pudo decir: La APK cumple con los criterios de una herramienta que apoya en la comunicación asíncrona (*Tabla 40*).

CAPÍTULO IV - IDEA A DEFENDER

4.1.6. Comprensión y aplicación

Se evaluó mediante las respuestas del cuestionario ¿cómo la APK ayuda en la comunicación asíncrona?, se obtuvo los resultados de la **Tabla 41**.

Criterio	ITEM	USUARIOS			TOTAL	PROMEDIO
		1	2	3		
APLICACIÓN	1. Elaborar frases en Braille desde cero	4	3	4	11	3,66
	2. Recordar el teclado de Braille	4	4	5	13	4,33
	3. Corregir sus errores de texto	4	3	4	11	3,66
	4. Recordar detalles con ayuda de los Diccionarios	4	3	3	10	3,33
	5. Realizar textos y expresarlos fonológicamente	5	5	5	15	5
						3,99

Tabla 40: Resultados de aplicación

Fuente: Elaboración propia

De acuerdo a las pruebas de la idea a defender se pudo ver que los dispositivos móviles ayudan e incentivan a los distintos usuarios, apoyando con un dispositivo de comunicación externo y la evaluación por parte del usuario están en *Anexo G*

4.1.7. Afirmación de la Idea a Defender

El prototipo elaborado en base a la propuesta para probar la idea a defender, se analizó el éxito o fracaso del prototipo para afirmar o refutar la misma.

Terminando con la revisión de los resultados obtenidos con el prototipo de prueba en relación a la Idea a Defender se puede afirmar que:

El modelo táctil de traducción propuesto con Algoritmos Genéticos, Tabú, Metodologías Híbridas y Android para los discapacitados de la vista, facilitará la comunicación asíncrona del Sistema Braille a Escritura en Tinta y su expresión Fonológica.

SE AFIRMO LA IDEA A DEFENDER.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

El presente capítulo describe las conclusiones generales de la Tesis, en lo que se refiere a la descripción de los logros alcanzados, de la misma forma se describe recomendaciones para trabajos e investigaciones posteriores basadas y orientadas por el presente.

5.1. Conclusiones

El alfabeto Braille es el sistema de lecto-escritura específico para discapacitados visuales y su conocimiento es fundamental para su formación, desarrollo y autonomía personal y, en consecuencia, es un instrumento importante para su integración en la sociedad.

Dado que el alfabeto Braille es el único sistema de lectoescritura para ciegos es necesario fomentar y promocionar su aprendizaje y uso y así lo ha explicitado también la Unión Mundial de Discapacitados Visuales.

El desarrollo de las nuevas tecnologías facilitan la producción en Braille al contar con una aplicación APK a la medida permite a la Sociedad de discapacitados visuales a ser más participativos y representativos, ya que es una de las ventajas que adquiere la persona desde el momento de incluir tecnología en el desarrollo de sus actividades, sin importar el carácter de estas.

Ya que el objetivo de esta Tesis fue crear un modelo táctil de traducción modificándolo la forma de escribir en Braille y readecuarlo para la facilidad y comprensión del mismo en una aplicación APK denominado *Charly*, la misma es un software Android que ayuda a la traducir del Sistema Braille al Español de forma escrita y fonológica utilizando para su desarrollo las metodologías Híbridas, que fuera capaz de facilitar el proceso de comunicación asíncrona en la sociedad y discapacitados visuales. La Aplicación está realizada para hacer todas las prácticas que se desee y facilitar la comunicación asíncrona, teniendo en cuenta que están todas las palabras del Sistema Braille.

En lo que respecta al proceso de investigación se puede concluir que la mejor técnica para recopilar información de campo es la entrevista la cual ayuda a recolectar información verídica y eficaz de un grupo de personas.

CAPÍTULO V - CONCLUSIONES Y RECOMENDACIONES

En el desarrollo de la misma se logró vincular los conocimientos adquiridos durante nuestro proceso de formación como estudiante en la U. A. P. junto con los nuevos conocimientos asimilados durante el desarrollo de la presente Tesis gracias a la ayuda del personal docente de la institución MIKI MAIA y a nuestra auto-educación.

5.2. Recomendaciones

Al desarrollar la presente Tesis se ha encontrado diversas recomendaciones al momento de sus pruebas con los usuarios finales, donde se clasifico las más relevantes para su manejo del mismo que a continuación se menciona:

- Cuando el usuario utiliza el sistema no permite digitar un solo botón desde el teclado braille, ya que la persona no vidente presiona varios botones a la vez y no así botón a botón para que el dato vaya cambiando.
- Antes de utilizar las distintas opciones, primero ingrese a la opción del manual para poder comprender de esta forma su funcionamiento y las distintas opciones que posee.
- El usuario inicial primero debe conocer el Sistema Braille en los diccionarios que tiene la APK, antes de poder ingresar a la opción del teclado Braille, para así poder transcribir de forma continua, para poder visualizar su traducción de escritura en tinta y su expresión fonológica.
- Si se desea efectuar investigaciones similares se recomienda implementar una línea Braille reduciendo los caracteres de la opción numérica como se concibió en esta tesis, para facilitar de esta forma su manipulación.

5.2.1. Trabajos futuros

EL campo de los dispositivos móviles con sistema Android es amplio y diverso, más aun cuando se analiza de forma detallada a un grupo selecto de personas y las necesidades que los mismos poseen. El presente prototipo se ha enfocado en los Algoritmos Genéticos, Tabú para apoyar en la traducción y comunicación asíncrona del sistema Braille a escritura en tinta, las mejoras del mismo se proponen las siguientes líneas de trabajo futuras:

- Al realizar proyectos similares se recomienda implementar una línea Braille con mayor número de caracteres y así lograr reproducir en ella oraciones completas.
- investigación propuesta en sistema Android se puede mejorar, aumentando más módulos para poder traducir del Sistema Braille a otros idiomas como ser Cavineño, Quechua, Aymara, etc.
- Tomar en cuenta los Algoritmos genéticos con búsqueda local Tabú para realizar investigaciones para facilitar el tiempo de respuesta cuando se tiene información muy extensa que es muy difícil buscar y el tiempo es vital para obtener información y este modelo ayuda a reducir drásticamente el tiempo.

Bibliografía

- A. Nosseir, D. F. (2012). *Mobile Development Process Spiral*.
- A. Technica, "Open handset alliance members.". (2012). Obtenido de <http://arstechnica.com/microsoft/news/2009/08/windows-mobile-loses-27-of-smartphone-market-in-q2.ars>
- Abrahamsson, P. (2005). Keynote: Mobile software development—. *Proceedings of the International Conference on*.
- Abrahamsson, P. (2007). Agile software development of mobile information systems. *Proceedings of the 19th international conference on Advanced information systems engineering*.
- Alejandro. (2013). "Interfaz de usuario en androide". Obtenido de <http://blogdeprogramacion.blogspot.com/2010/10/interfaz-de-usuario-en-androide.html>
- Android Developers "What is Android?". (2014). Obtenido de <http://developer.android.com/guide/basics/what-is-android.html>
- Arredondo, R. (2001). Estudio exploratorio sobre la escritura de palabras homofonas en español, Braille grado1 y grado 2. *Postgrado de Linguística*.
- Arturo Baz Alonso, I. F. (9 de 12 de 2013). *Dispositivos Móviles*. Obtenido de <http://156.35.151.9/~smi/5tm/09trabajos-sistemas/1/Memoria.pdf>
- Baver_R. (1994). *Genetic Algorithms and Investment Strategies*. Willey: Finance edition.
- Bloom, B. (1956). "Taxonomy of educational objectives: The classification of educational goals: Handbook I, cognitive domain". New York, Toronto: http://www.ops.org/reading/blooms_taxonomy.htm.
- BrailleApp-Free. (2014). Obtenido de http://mobile9apk.com/Download_BrailleApp-Free_APK.html
- Bunge, M. (1989). *La investigación científica*. Paris: Editorial Ariel S. A. Barcelona.

CAPÍTULO V - CONCLUSIONES Y RECOMENDACIONES

- Condesa. (2013). *“Usando layouts basadas en XML con Android”*. Obtenido de <http://androideity.com/2011/07/11/usando-layouts-basadas-en-xml-con-android/>
- Cruz, P. P. (2010). *Inteligencia artificial con aplicaciones a la ingeniería*. México: Alfaomega Grupo Editor, S.A.
- D.E., G. (1989). *Genetic Algorithms in Search Optimization and Machine*. Addison-Wesley.
- Dingsoyr, T. D. (2009). *What Do We Know about Agile Software Development?* IEEE Software vol. 26.
- Dunlop, M. B. (2002). *The Challenge of Mobile Devices for Human Computer*. Volume 6, Issue.
- Ferré Grau, X. (s.f.). Principios Básicos de Usabilidad para Ingenieros Software. *Departamento de Lenguajes y Sistemas Informáticos e Ingeniería del Software*. Madrid, España.
- Forman, G. Z. (1994). *The challenges of mobile computing*. IEEE Computing. Vol. 27.
- Gabriel Gerónimo Castillo, E. H.-T. (s.f.). El Proyecto Edumóvil: INCORPORANDO LA TECNOLOGÍA MÓVIL EN LA EDUCACIÓN PRIMARIA. *Universidad Tecnológica de la Mixteca*. México.
- Gartner, “Gartner says android to become no. 2 worldwide mobile operating system. (2014). Obtenido de <http://www.gartner.com/it/page.jsp?id=1434613>
- Glover, F. (1986). *Future Paths for Integer Programming and Links to Artificial Intelligence*. Computers and Operations Research.
- Glover, F. a. (1997). *Tabu Search*, Kluwer Academic Publishers.
- Glover, F. L. (2006). *Principles of Tabu Search*. To appear in *Approximation Algorithms and Metaheuristics*. Chapman & Hall/CRC.
- Harishankar, S. B. (2013). *Applying Agile Methodology in Mobile Software Engineering: Android Application Development and its Challenges*.

CAPÍTULO V - CONCLUSIONES Y RECOMENDACIONES

- Harishankar_S_B. (2016). *Applying Agile Methodology in Mobile Software Engineering: Android Application Development and its Challenges*.
- Heyes, I. S. (2002). *Just Enough Wireless Computing*, Prentice Hall.
- Huertas. (1988). Peculiaridades de la lectura táctil en Braille "Journal for the Study of Education and Development".
- INE. (1 de Enero de 2014). Cobertura al proceso censal INE.
- InvestorPlace, "New burst of momentum for google android os. (2015). Obtenido de <http://www.investorplace.com/18151/google-android-os-major-corporate-smart-phone-winner>
- ISO. (01 de 02 de 2015). *ISO/IEC TR 9126-4:2004 - Software engineering -- Product quality -- Part 4: Quality in use metrics*. Obtenido de http://www.iso.org/iso/catalogue_detail.htm?csnumber=39752
- Lee, W.-M. (s.f.). *Beggining Android 4: Aplicacion Development*.
- Macworld, "Apple unveils iphone. (2014). Obtenido de <http://www.macworld.com/article/54769/2007/01/iphone.html>
- Marco., B. S. (2013). "Qué es el XML". Obtenido de http://www.mclibre.org/consultar/xml/lecciones/xml_queues.html
- McCracken, S. (s.f.). *Android: Curso de desarrollo de aplicaciones*.
- Mendoza., F. (2014). "Introducción a XML (Extensible Markup Language)". Obtenido de <http://www.ilustrados.com/tema/3783/Introduccion-Extensible-Markup-Language.html>
- Miki, Y. H. (2014). Tabu - Local Search Mechanism for Mega Process Genetic Algorithm. Japon: Doshisha University.
- O. H. Alliance, "Open handset alliance members. (2015). Obtenido de http://www.openhandsetalliance.com/oha_members.html
- O.S.L. (2011). *Documentacion del curso de desarrollo de Android*.

CAPÍTULO V - CONCLUSIONES Y RECOMENDACIONES

OMS. (Agosto de 2014). Organización mundial de la salud.

ONCE_UAB. (2010). *UniversidadAutonomaBarcelona
OrganizacionNacionalCiegosEspañoles.*

openBoxer. (14 de 12 de 2013). “*Lenguaje de programación Java*”. Obtenido de <http://www.openboxer.260mb.com/java.php>

Opera, “*Opera mini app approved for the app store.*”. (2014). Obtenido de <http://www.opera.com/press/releases/2010/04/13/>

P. Abrahamsson, A. H. (2004). Mobile-D: an agile approach for mobile application development. *Companion to the 19th annual ACM SIGPLAN conference on Object-oriented programming.*

Página del desarrollador de Android; (01 de 04 de 2014). Obtenido de <http://developer.android.com/index.html>

Publishers, M. K. (1999). *Foundations of Genetics Algorithms*. Banzhaf W. Reeves.

R. I. M. Limited, “*About compression.* (7 de 2015). Obtenido de http://docs.blackberry.com/en/smartphone_users/deliverables/11499/About_compression_778018_11.jsp

R. I. M. Limited, “*Blackberry service real-time of email messages.* (1 de 05 de 2015). Obtenido de <http://www.blackberry.com/btsc/microsites/search.do?cmd=displayKC&docType=kc&externalId=KB12373>

Ramsin, V. R. (2008). Designing an agile methodology for mobile software development: A hybrid method engineering approach. *Research Challenges in Information Science* (págs. 337-342). Second International Conference.

Rodriguez, T. (2012). *Android en 2012: Desarrollando aplicaciones.*

S. Microsystems, “*Oracle to buy sun.*”. (2012). Obtenido de <http://www.sun.com/third-party/global/oracle/>

Saad, N. (1998). *El sistema Braille*. Eureka.

CAPÍTULO V - CONCLUSIONES Y RECOMENDACIONES

sgoliver. (07 de 02 de 2012). *“Estructura de un proyecto Android”*. Obtenido de <http://www.sgoliver.net/blog/?p=1278>

Software Alfabeto Braille en Linea. (s.f.). Obtenido de <http://www.fbu.edu.uy/alfabeto/alfabeto-online.htm>

Software traductor a Braille Duxbury. (s.f.). Obtenido de http://www.tecnologiaayudas.com.ar/productos_software_traductor_braille_duxbury.html

Soluciones móviles de Safelayer S.A. (01 de 04 de 2014). Obtenido de <http://www.safelayer.com/es/soluciones/mobile-pki>

StatCounter GlobalStats. (14 de 12 de 2013). Obtenido de <http://gs.statcounter.com>

T. A. S. Foundation, “Opera mini approved for app store. (2013). Obtenido de <http://www.apache.org/licenses/LICENSE-2.0>

Tellez, G. (1 de 05 de 2015). Director departamental del instituto de la ceguera. Pando.

Top Braille. (2007). Obtenido de <http://www.elmundo.es/elmundosalud/medicina/1183965035.html>

tutkimuskeskus, O. S. (2006). *Enabling software process improvement in agile software development teams and organisations.* Finland: VTT Technical Research Centre of Finland.

U. T. R. Authority, “Tra announces the suspension of blackberry messenger, blackberry. (2015). Obtenido de http://www.tra.ae/news_TRA_Announces_the_Suspension_of_Blackberry_Messenger,_Blackberry_E_mail_and_Blackberry_Web_Browsing_Services_in_the_UAE_from_October_11,_2010-180-1.php

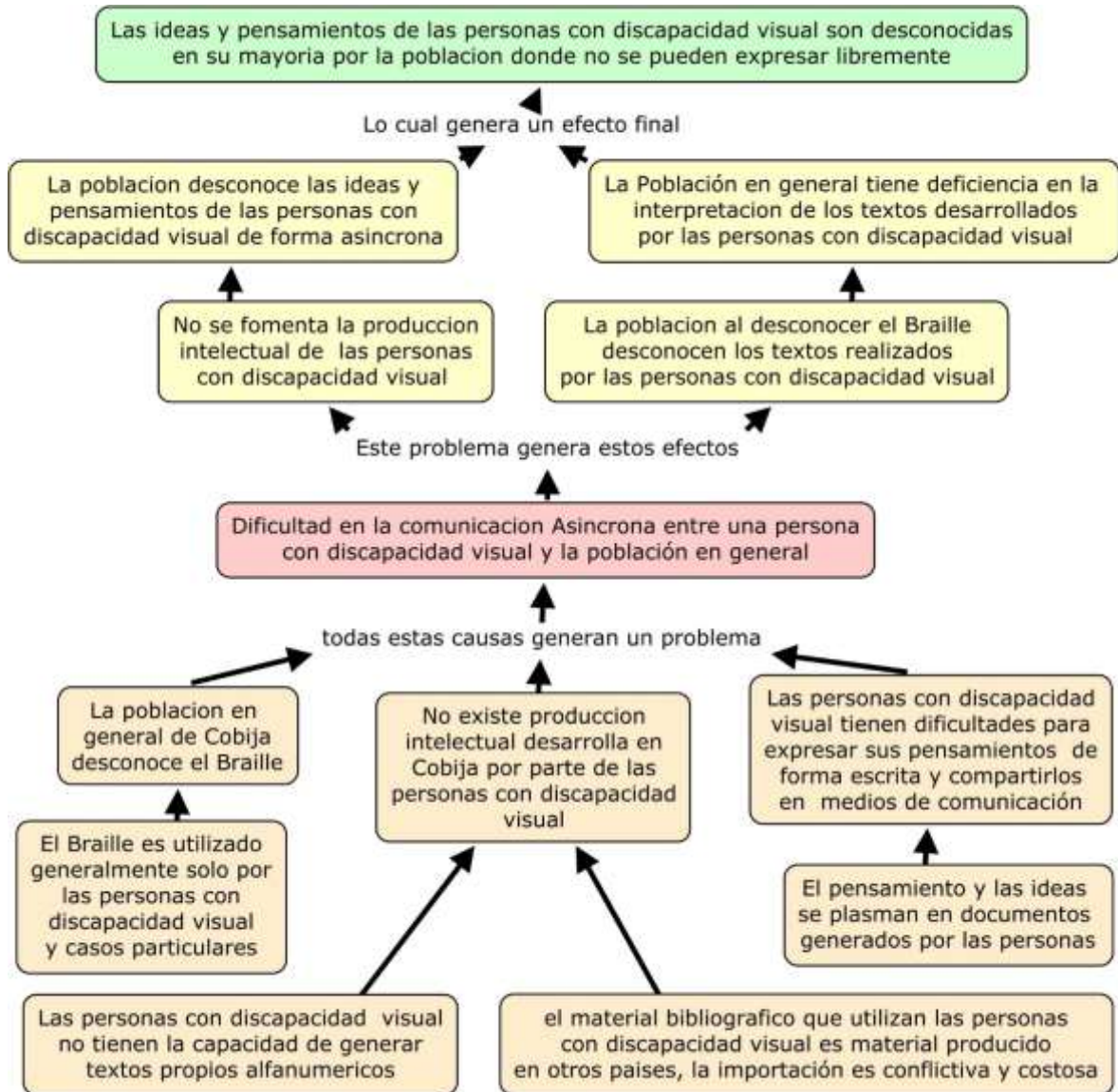
UNESCO, P. d. (1954). *La estructura Braille en el Mundo.* Paris: Mackenzie.

Vilchez, A. (2009). *Que es Android: Características y aplicaciones.*

CAPÍTULO VI

ANEXOS













































A - Arbol de Problemas



B - Alfabeto Braille

1	4
2	5
3	6

Alfabeto Braille

a	b	c	d	e	f	g	h	i	j
									
k	l	m	n	o	p	q	r	s	t
									
u	v	x	y	z	ü	ñ	w	ç	
									
á	é	í	ó	ú					
									
1	2	3	4	5					
									
6	7	8	9	0					
									

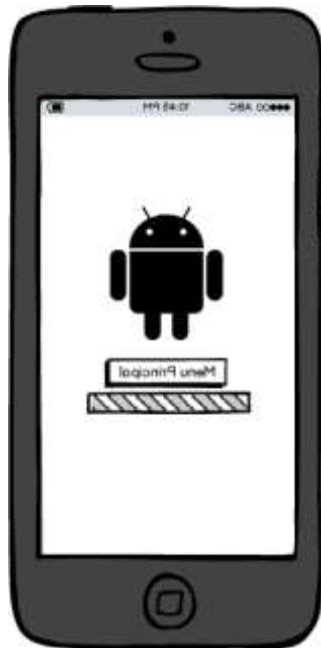
Alfabeto Braille

CAPITULO VI - ANEXOS

C - Conversion Braille – Binario – Decimal – Alfanumerico

B1	B2	B3	B4	B5	B6	#	_	Decimal	Alfa-Núm.
1	0	0	0	0	0	0	0	128	A
1	1	0	0	0	0	0	0	192	B
1	0	0	1	0	0	0	0	144	C
1	0	0	1	1	0	0	0	152	D
1	0	0	0	1	0	0	0	136	E
1	1	0	1	0	0	0	0	208	F
1	1	0	1	1	0	0	0	216	G
1	1	0	0	1	0	0	0	200	H
0	1	0	1	0	0	0	0	80	I
0	1	0	1	1	0	0	0	88	J
1	0	1	0	0	0	0	0	160	K
1	1	1	0	0	0	0	0	224	L
1	0	1	1	0	0	0	0	176	M
1	0	1	1	1	0	0	0	184	N
1	1	0	1	1	1	0	0	220	Ñ
1	0	1	0	1	0	0	0	168	O
1	1	1	1	0	0	0	0	240	P
1	1	1	1	1	0	0	0	248	Q
1	1	1	0	1	0	0	0	232	R
0	1	1	1	0	0	0	0	112	S
0	1	1	1	1	0	0	0	120	T
1	0	1	0	0	1	0	0	164	U
1	1	1	0	0	1	0	0	228	V
0	1	0	1	1	1	0	0	92	W
1	0	1	1	0	1	0	0	180	X
1	0	1	1	1	1	0	0	188	Y
1	0	1	0	1	1	0	0	172	Z
1	1	1	0	1	1	0	0	236	A
0	1	1	1	0	1	0	0	116	E
0	0	1	1	0	0	0	0	48	I
0	0	1	1	0	1	0	0	52	O
0	1	1	1	1	1	0	0	124	U
1	0	0	0	0	0	1	0	130	1
1	1	0	0	0	0	1	0	194	2
1	0	0	1	0	0	1	0	146	3
1	0	0	1	1	0	1	0	154	4
1	0	0	0	1	0	1	0	138	5
1	1	0	1	0	0	1	0	210	6
1	1	0	1	1	0	1	0	218	7
1	1	0	0	1	0	1	0	202	8
0	1	0	1	0	0	1	0	82	9
0	1	0	1	1	0	1	0	90	0

D - Prototipo



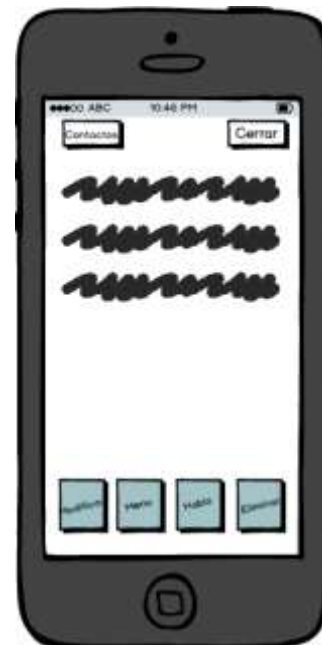
*Figura 68 Splash – Bienvenida
Fuente: Implementación propia*



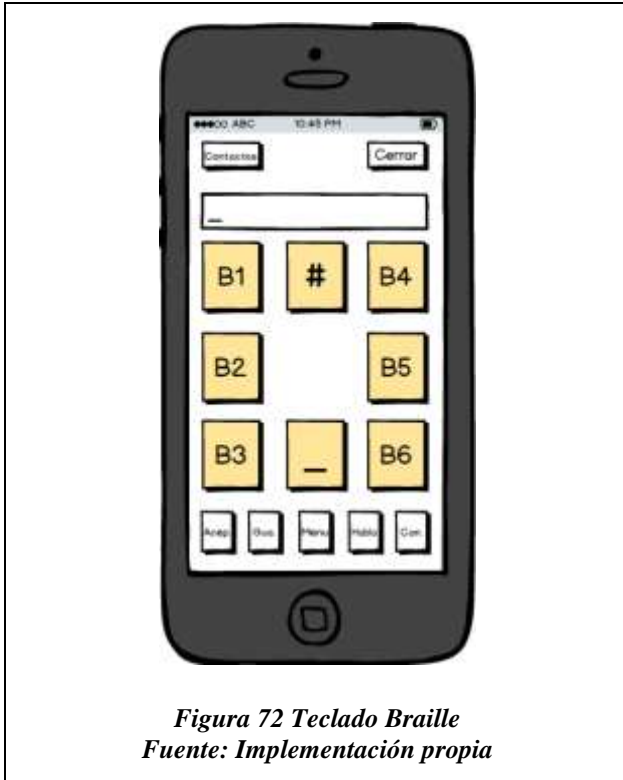
*Figura 69 Menu principal
Fuente: Implementación propia*



*Figura 70 Abecedario Vocabulario y Números
Fuente: Implementación propia*



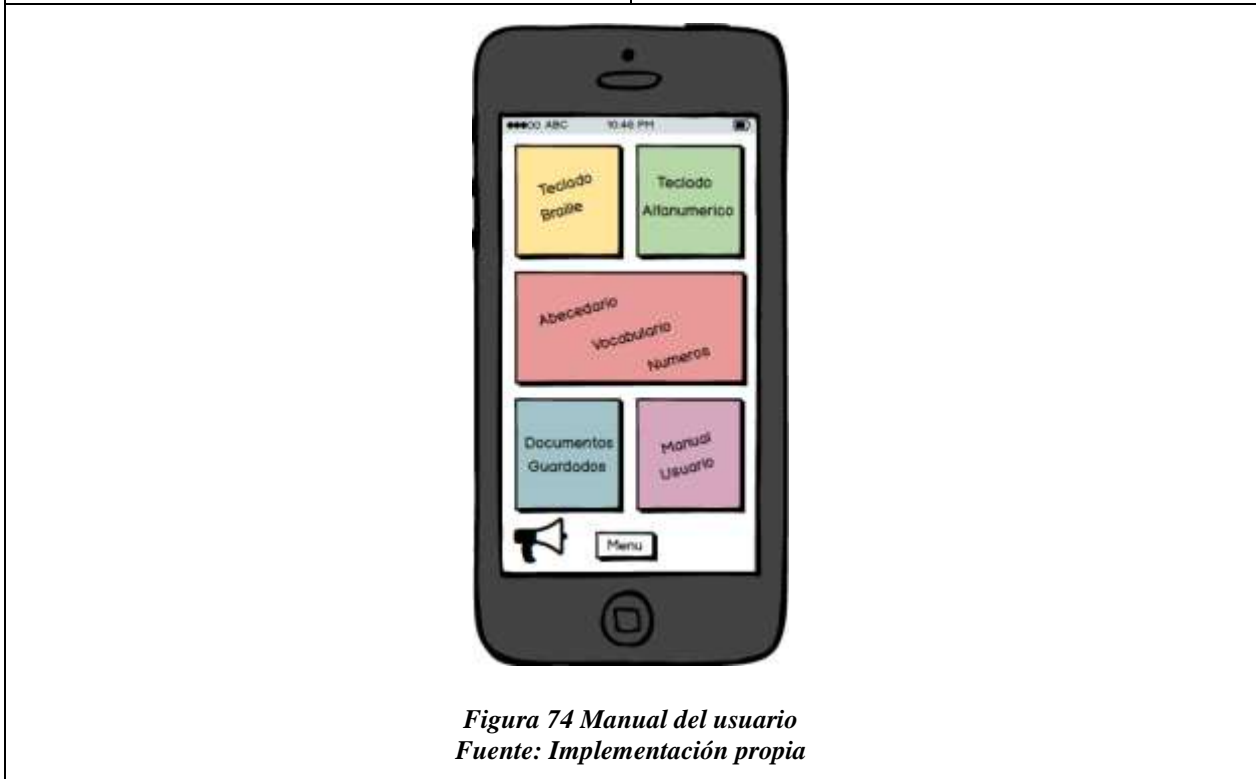
*Figura 71 Documentos guardados
Fuente: Implementación propia*



*Figura 72 Teclado Braille
Fuente: Implementación propia*



*Figura 73 Teclado Alfanumérico
Fuente: Implementación propia*



*Figura 74 Manual del usuario
Fuente: Implementación propia*

E - Entrevista con el Dir. discapacitados de la vista



Lic. Gregorio Téllez
Director Departamental del I.B.C. (Instituto Boliviano de la Ceguera)
Telf.:71116747

Tiene una reunión mensual como Asociación **I.B.C.** (Instituto Boliviano de la Ceguera) con parte del estado, los mismos carecen de fondos, se tiene un convenio con la Gobernación para ser atendidos los mismos pero con poca atención y dándoles un pequeño espacio en el Centro de Educación y rehabilitación de personas con discapacidad (**MIKY MAIA** - Autónomo)

El **Centro Ester Campos** (escuela cobija) fue lastimosamente tergiversado por el **SEDUCA** puesto que el **Lic. Gregorio Téllez** aclaro que el mismo centro es ficticio puesto que no funciona como centro especial debido a que los señores docentes que van a impartir clases son de la normal y no de centros especiales.

Esto provoca que los estudiantes les enseñen a los profesores y después los docentes simplemente repiten lo que los estudiantes les enseñaron, pero los estudiantes pueden cometer errores y no sería de este modo una educación alterna y productiva.

También se tiene la **ACEACRE** (Asociación Ceguera del Acre) el mismo tiene su propia directiva, es una Asociación Privada de Discapacitados de la vista, el entrevistado aclaro que los mismos tienen familias ostentosas y no carecen de muchas necesidades y el Sr. **Luis Antonio** es el encargado al cual se lo puede ubicar en la gobernación porque el mismo trabaja ahí.

Después de la entrevista el Director Departamental del **I.B.C.** para facilitar la investigación facilito el material físico del alfabeto Braille puesto que el mismo es independiente debido a que cada localidad y región se particularizo y adapto para su beneficio.

F - Entrevista con especialista interprete de Sordos



Lic. Nancy Portal
Psicóloga Intérprete (20 años de experiencia)

El Sr. Henry Montero es el presidente de (CODEPEDIS) y la profesora María del Carmen Chávez Guacama es la directora de (CODEPEDIS), ambos están a cargo de los discapacitados sordos, pero ambos simplemente ocupan los cargos.

La especialista en estar en contacto e interpretación es la Lic. Nancy Portal (Psicología) ella es la persona que tiene mejor contacto con los sordos mudos y experiencia por más de 20 años, la misma aclaro que incluso su hermana sufre esta discapacidad por lo que tuvo que crecer aprendiendo su medio y forma de comunicación para poder comprender.

La misma aclaro que la primera lengua de señas que llevo acá fue la lengua Americana, posteriormente se fue modificando y la segunda lengua que aprendieron fue la Lengua de señas de Bolivia.

Después de estos cambios en el año 2001 generaron sus propias señas por la Identidad del Sordo en Bolivia provocando que cada región se independice a lo cual la especialista aclaro que por departamentos crearon 60 nuevas señas particulares, provocando de esta forma una independencia regional.

Esto provoco conflictos con las normales, más aun cuando de las mismas no salen expertos ni especialistas en interpretación generando inconvenientes a gran escala, siendo su problema casi similar al de los discapacitados de la vista por no ser comprendidos.

CAPITULO VI - ANEXOS

También aclaro que a este tipo de discapacitados solo les gusta que les digan sordos y no mudos, esto provoca gran molestia por su parte puesto que los mismos no aprendieron a escuchar es por eso que no hablan, suceden casos particulares en los cuales no se cumple este acontecimiento, pero esos son demasiado raros.

La entrevista lo vio esta tesis de manera productiva pues ella aclaro que es muy importante promover la enseñanza a través de imágenes y dispositivos móviles, esto genera el método.

Ella también aportó con la facilidad que puede otorgar que esta aplicación a su vez ayude en la interpretación del lenguaje de símbolos con el alfa-numérico y el Braille

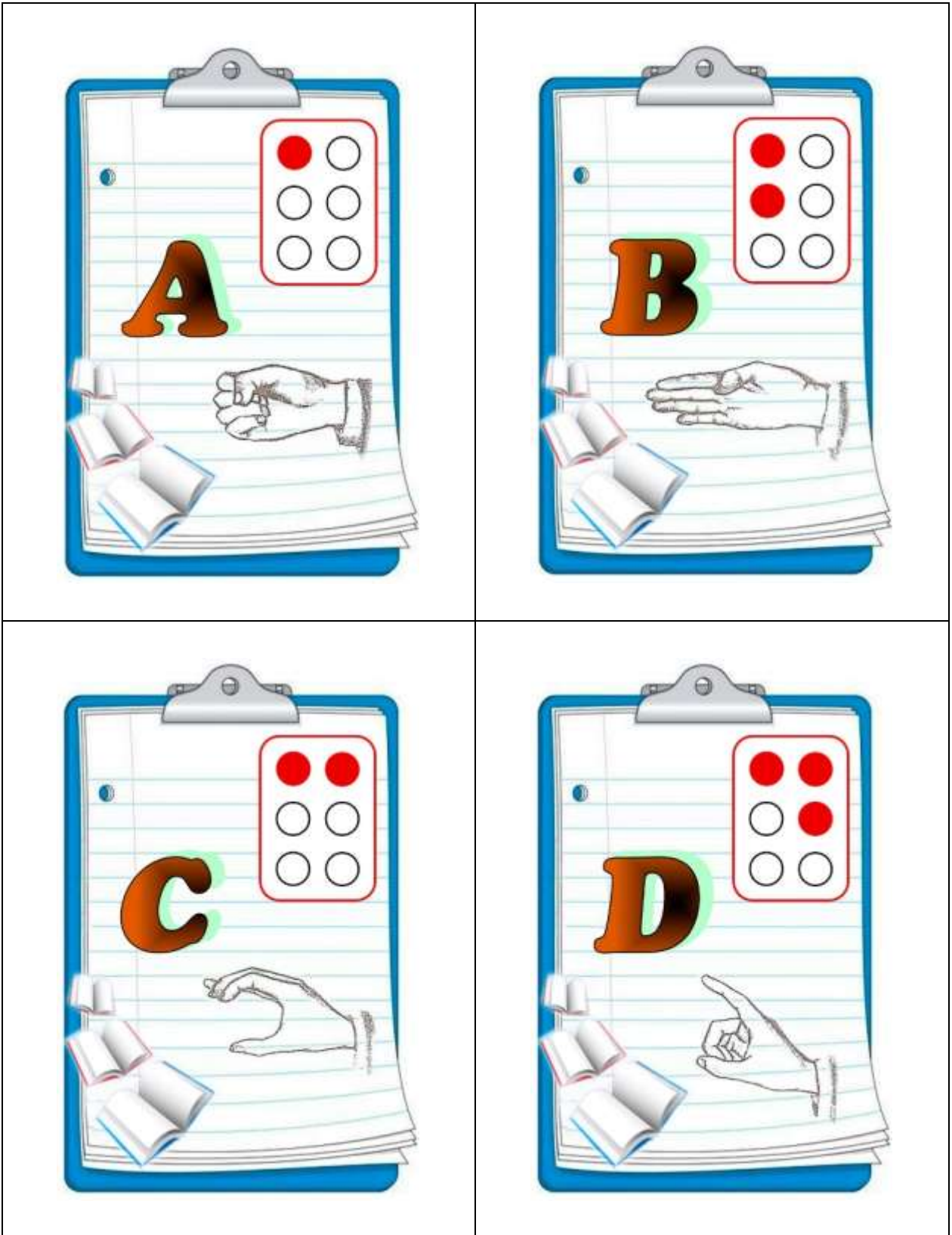
G - Evaluacion de la APK (ISO/IEC 9126)

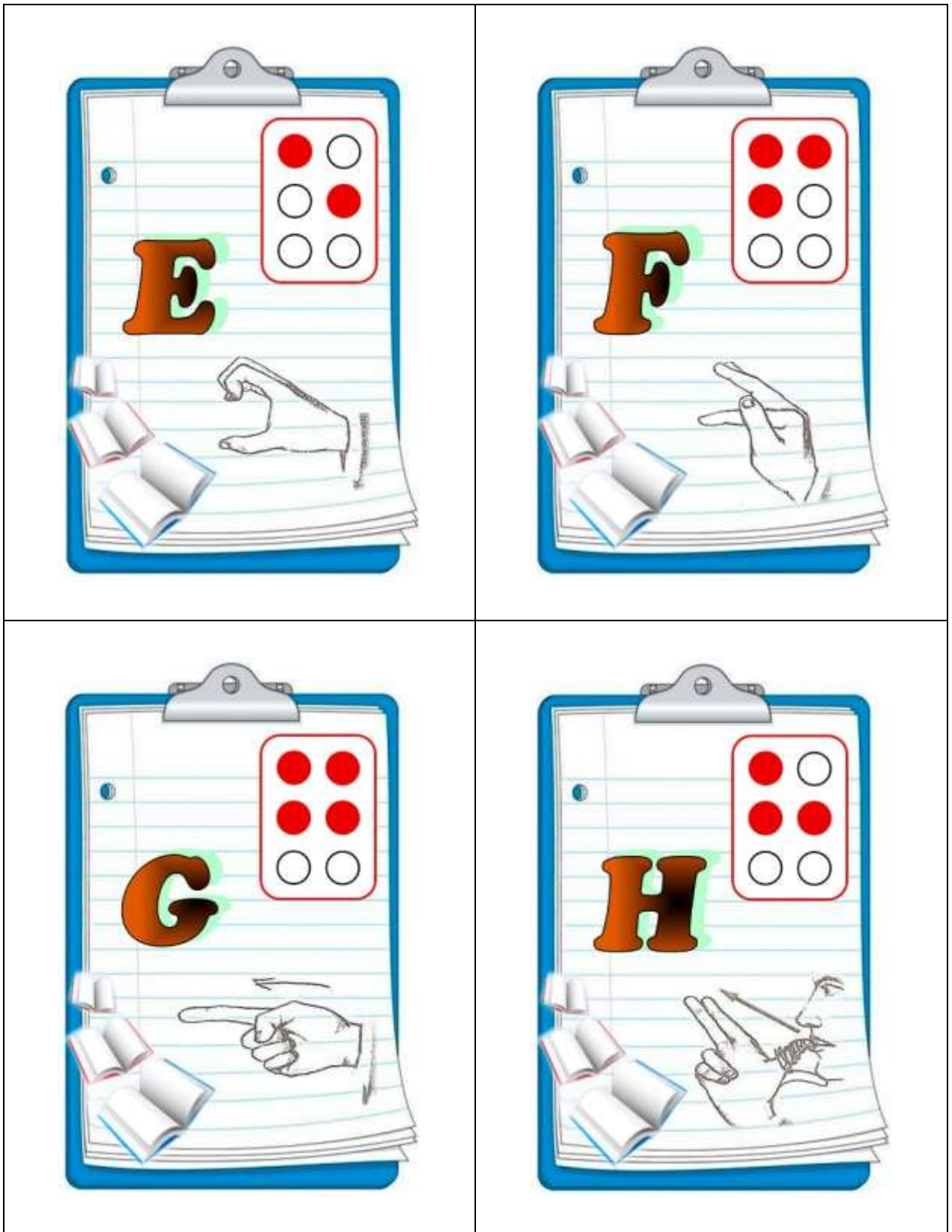
Evaluación de la APK por parte del Usuario con acceso a toda la aplicación de apoyo.

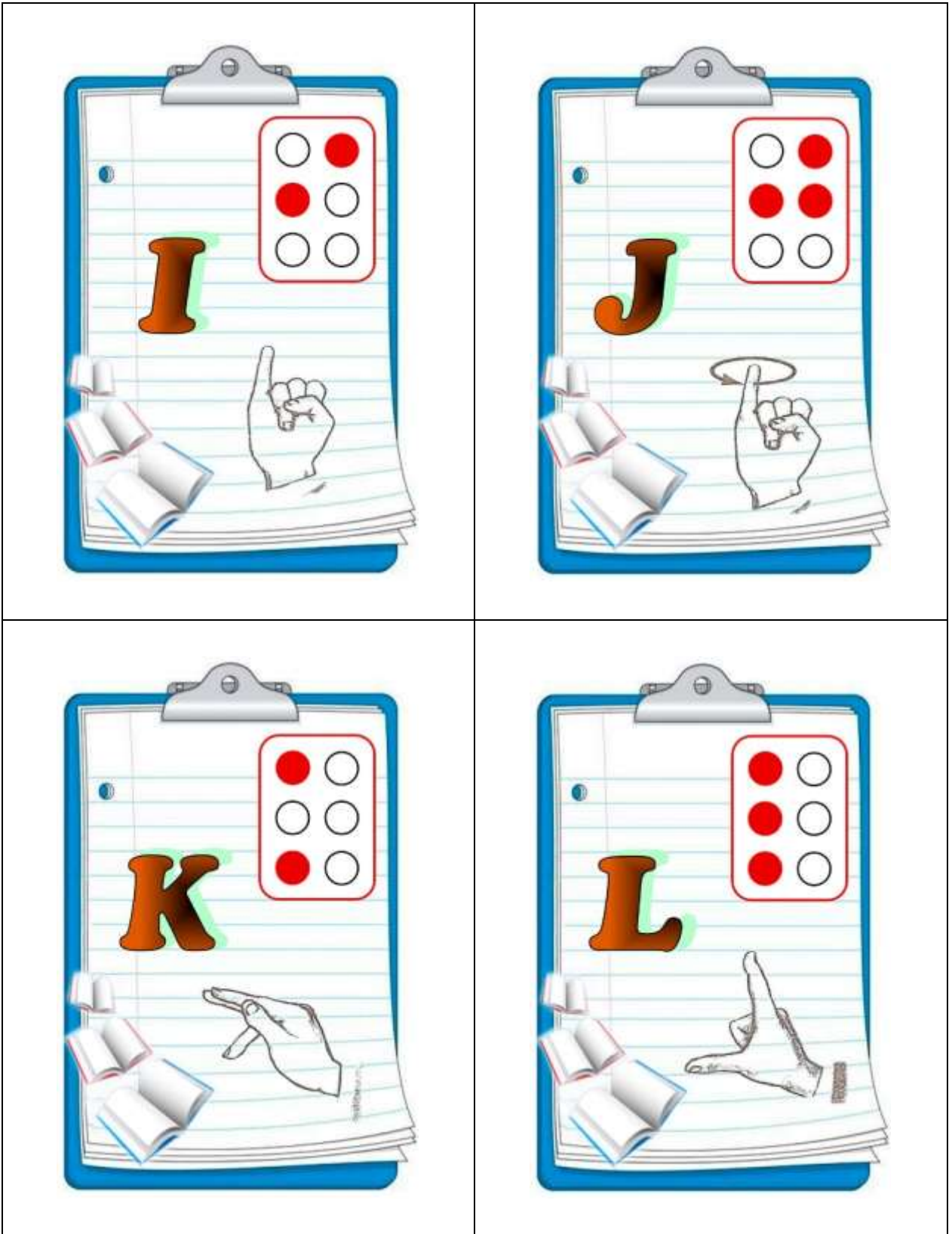
<i>Calificación</i>	<i>Puntaje</i>
Muy malo	1
Pésimo	2
Regular	3
Bueno	4
Muy Bueno	5

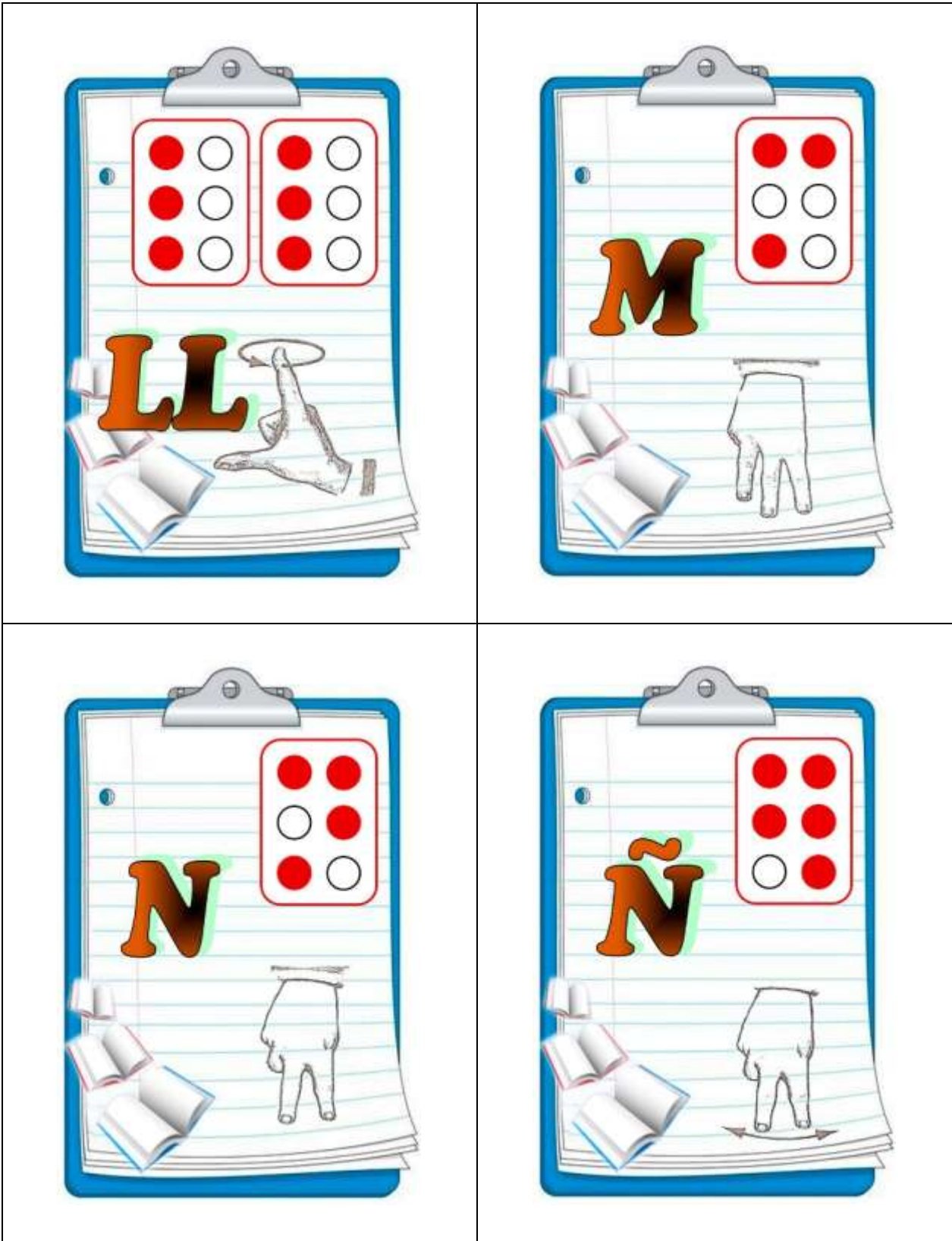
<i>Nro.</i>	<i>Atributos</i>	<i>Puntaje</i>
<i>FUNCIONALIDAD</i>		
<i>1</i>	¿Cumple los requerimientos básicos funcionales?	5
<i>2</i>	¿Contiene un sistema adaptable e independiente a distintos equipos?	5
<i>3</i>	¿Posee niveles de funcionalidad?	4
<i>FIABILIDAD</i>		
<i>4</i>	¿La aplicación es segura y confiable?	5
<i>5</i>	¿El sistema puede aceptar errores involuntarios, sin quedar fuera de servicio?	5
<i>6</i>	¿En caso de contener fallas y no funcione correctamente, su recuperación será inmediata?	5
<i>USABILIDAD</i>		
<i>7</i>	¿Todas las entradas están etiquetadas adecuadamente para su manejo?	5
<i>8</i>	¿El contenido es de fácil manipulación y aprendizaje?	4
<i>9</i>	¿Funciona correctamente sin la necesidad del Internet?	5
<i>10</i>	¿La APK es atractiva y tiene una buena interfaz de usuario?	5
<i>EFICIENCIA</i>		
<i>11</i>	¿El tiempo de generación y ejecución es dinámico y menor a 10 segundos?	5
<i>12</i>	¿El tiempo de instalación de esta aplicación es menor a 5 minutos?	5
<i>MANTENIMIENTO</i>		
<i>13</i>	¿La APK posee la descripción de las fuentes de los archivos y de la base de datos?	3
<i>14</i>	¿Si algún día se desea hacer cambios, se cuenta con el código fuente?	5
<i>15</i>	¿Si algún día se desea hacer cambios dentro de la APK, esta puede seguir funcionando?	5
<i>16</i>	¿Al analizar algún cambio, es posible identificar los errores?	5
<i>PORTABILIDAD</i>		
<i>17</i>	¿La APK es adaptable a cualquier equipo con sistema Android?	5
<i>18</i>	¿Presenta manuales de uso?	5
<i>19</i>	¿La APK puede compartirse manera rápida y sencilla?	5

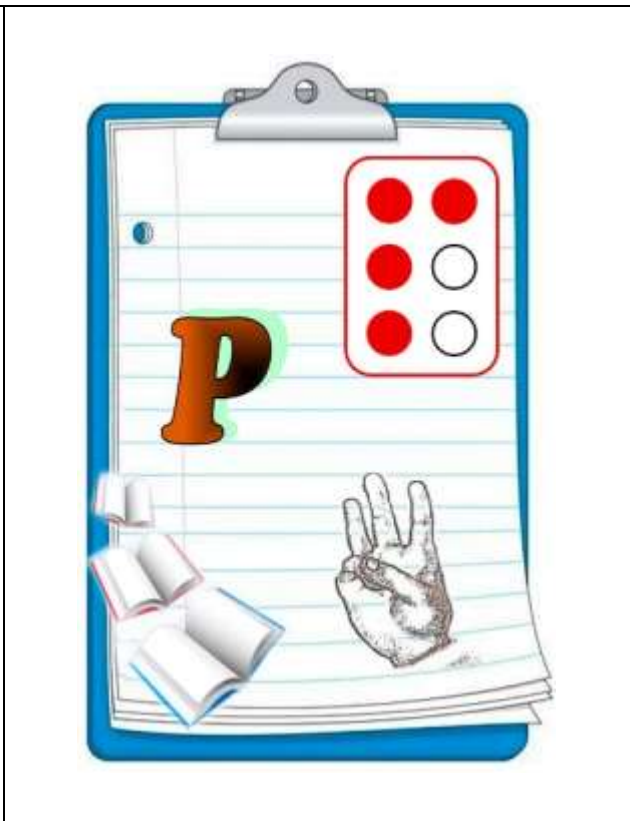
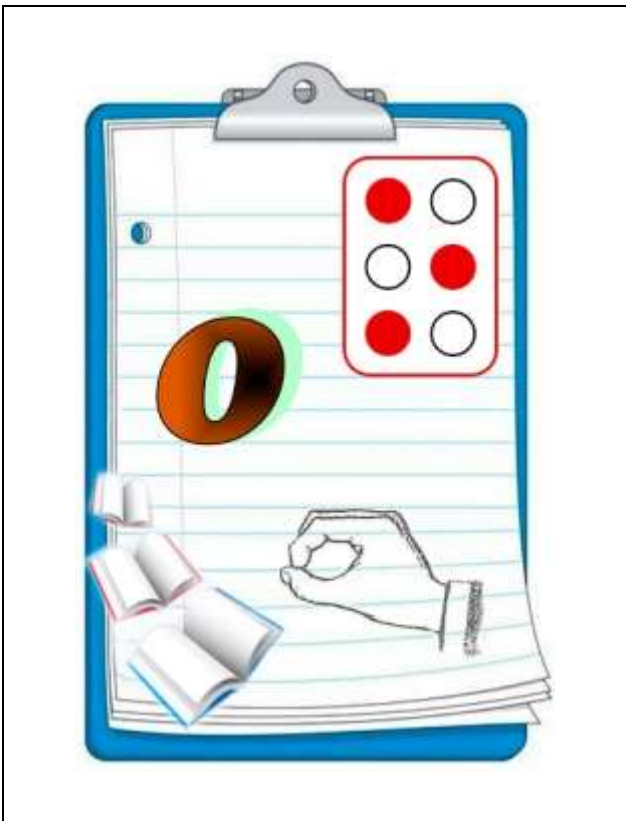
H - Abecedario Sudamericano en Alfa-numerico, Braille y Simbolos

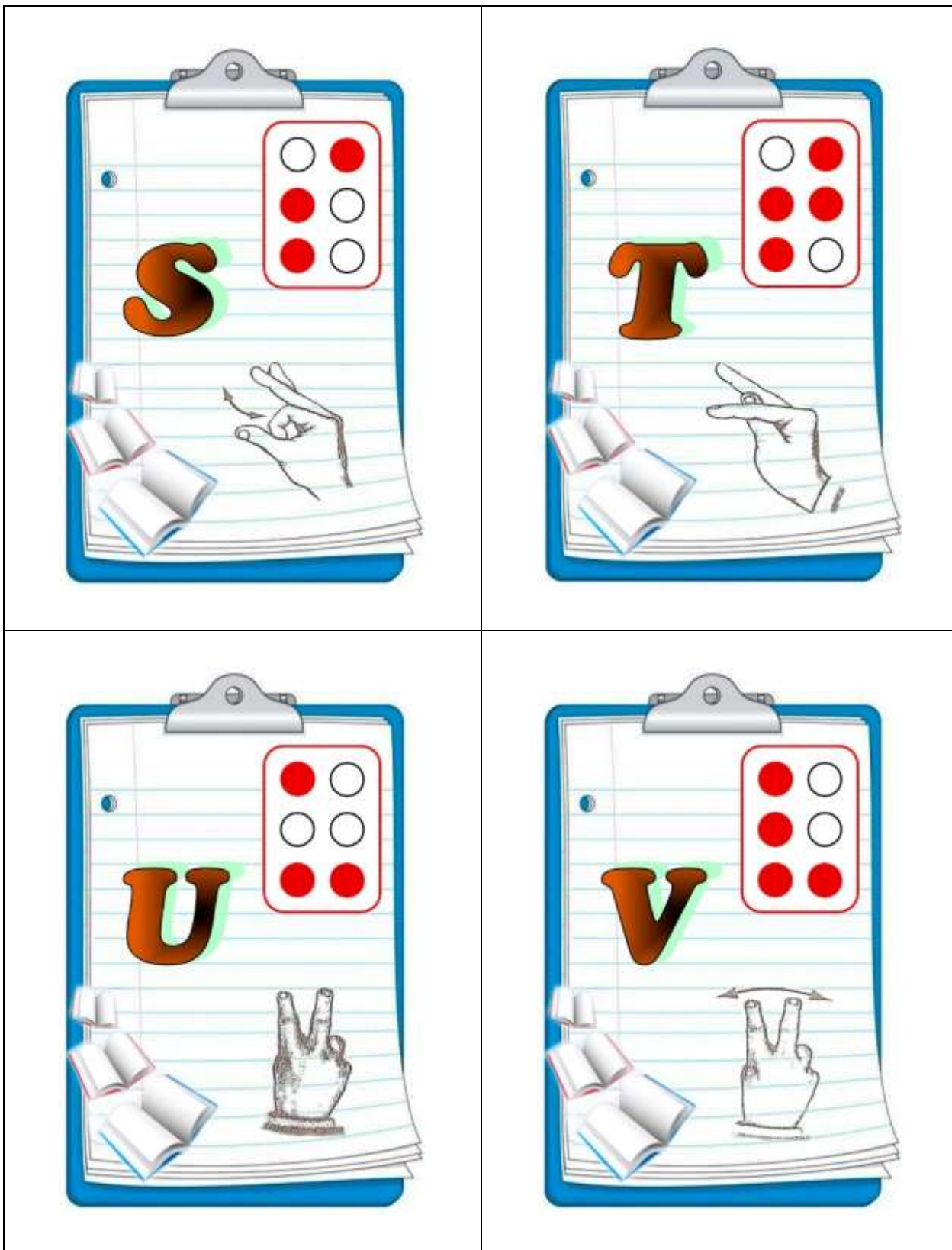


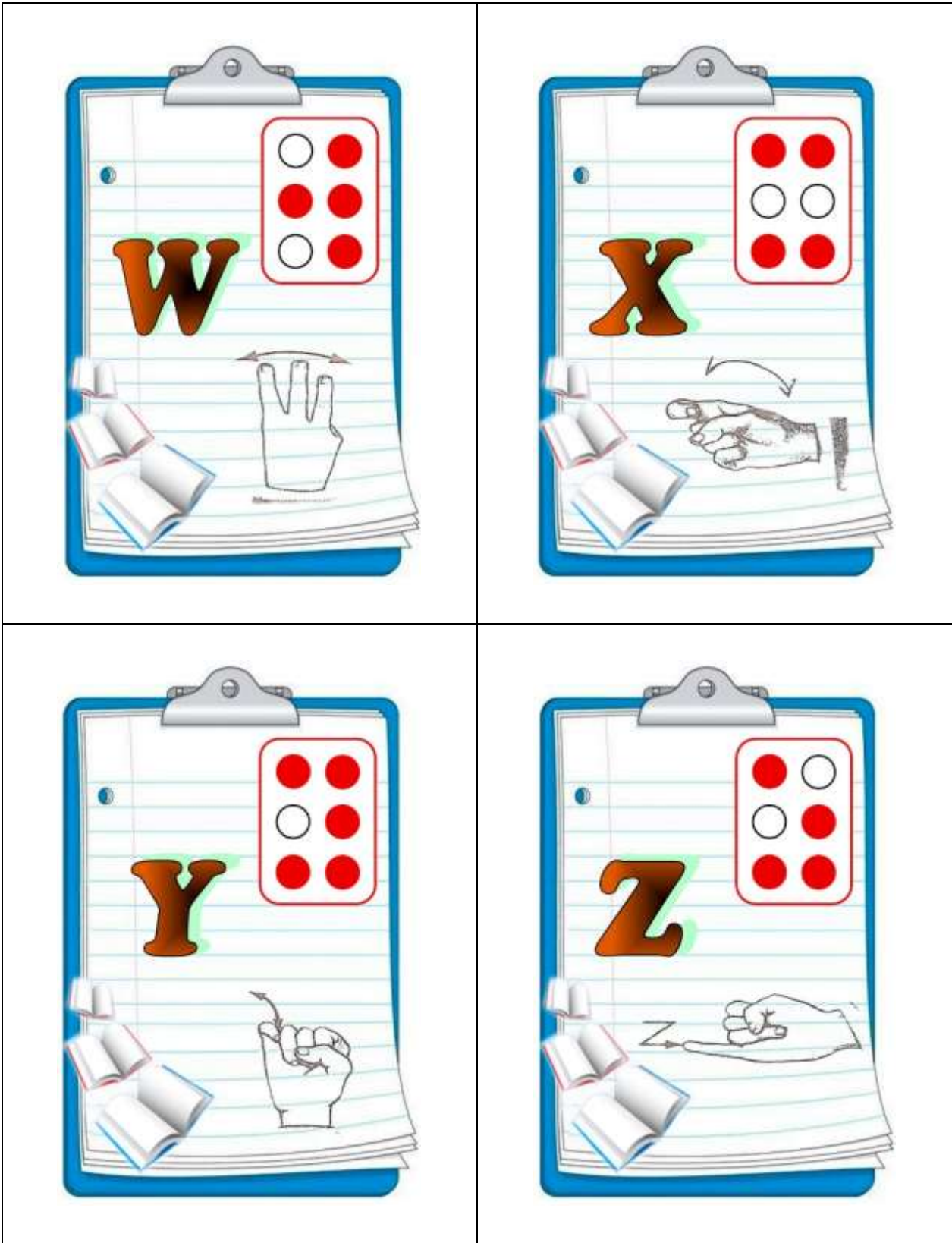




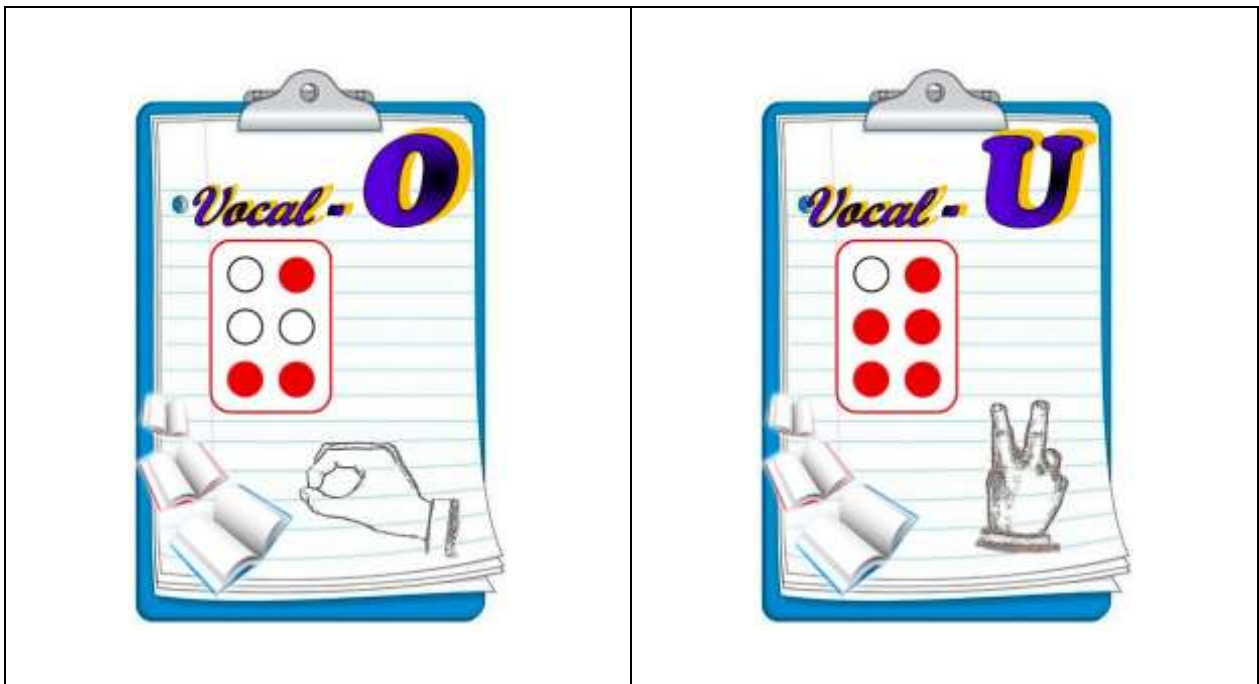
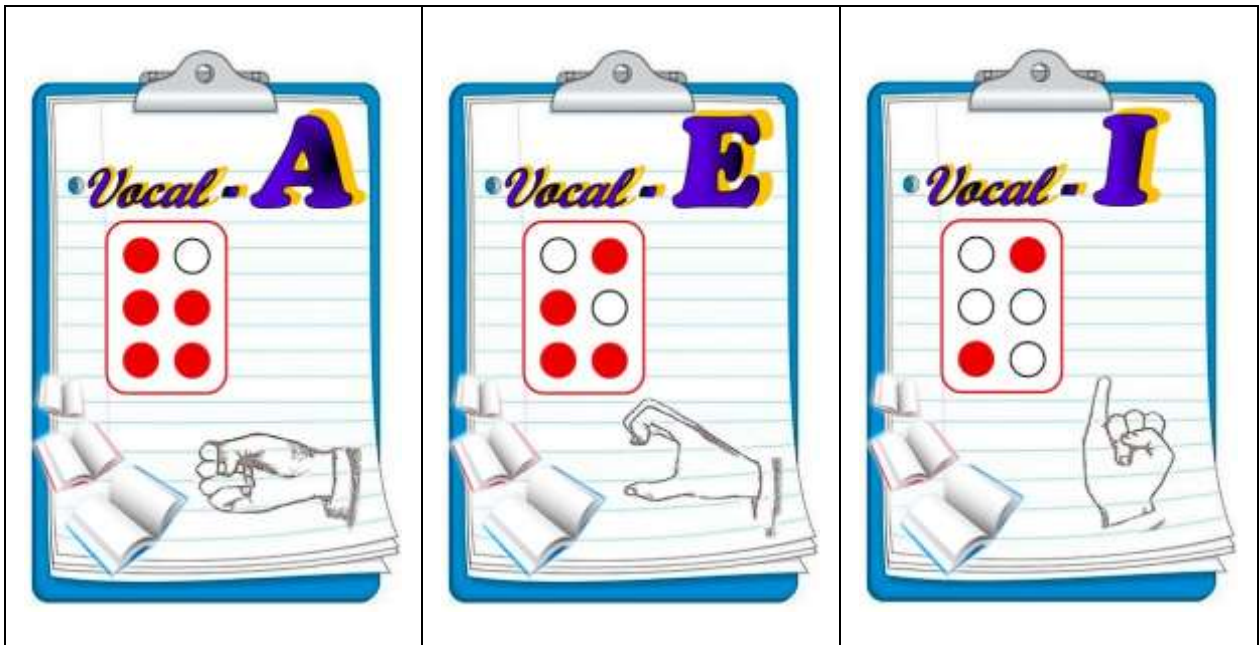








I - Vocabulario en Alfa-numerico, Braille y Simbolos



J - Numeros en Alfa-numerico, Braille y Simbolos

