

**UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN**

SISTEMA DE APOYO A LA ENSEÑANZA DE INGLES

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

GERARDO MIGUEL ONETTO SOTO

**PROFESOR GUÍA:
JOSE A. PINO URTUBIA**

**MIEMBROS DE LA COMISION
JOHAN FABRY
ALEX BORQUEZ GRIMALDI**

**SANTIAGO DE CHILE
ABRIL 2011**

RESUMEN DE LA MEMORIA
PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN COMPUTACIÓN
POR: GERARDO ONETTO S.
FECHA: 25/04/2011
PROF. GUIA: Sr. JOSE A. PINO URTUBIA

SISTEMA DE APOYO A LA ENSEÑANZA DE INGLES

Es de especial interés para Chile, en el marco de la creciente inserción al mundo globalizado, que sus ciudadanos mejoren sus capacidades comunicacionales en inglés. El Ministerio de Educación, en particular, ha ajustado en el 2009 el marco curricular de la reforma educacional iniciada a mediados de la década de los 90, incluyendo un nuevo sector de aprendizaje con objetivos específicos: Inglés.

El Ministerio de Educación por medio de estos ajustes curriculares destaca la importancia de que los alumnos posean las capacidades para utilizar el idioma como una herramienta que les permita acceder a la información, así como resolver situaciones comunicativas simples en forma oral y escrita. Favoreciendo su participación activa en la educación superior o en el mundo laboral.

Uno de los problemas que posee la enseñanza del inglés en Chile es contar con los instrumentos que permitan a los alumnos ejercitar el uso del lenguaje como una herramienta comunicativa más que como un objeto de estudio gramatical.

La solución desarrollada contempló el uso de tecnología y dispositivos móviles para apoyar el proceso de aprendizaje de los alumnos. De los mismos documentos sobre lineamientos, formatos y sugerencias del Ministerio de Educación, se extrajo un conjunto de actividades de aprendizaje que abarcan los contenidos para los primeros niveles de aprendizaje. Estas actividades se centran en el uso del lenguaje para realizar tareas clave, fomentando la cooperación y discusión de los alumnos.

Como resultado se obtuvo un sistema de actividades que puede ser ejecutado en dispositivos móviles que utilicen el sistema operativo Android. Aún no es posible determinar el impacto de un sistema de este tipo en la población de alumnos chilena, pero es de esperarse que siga el mismo patrón positivo que se muestran en estudios de mobile learning, mejorando finalmente las competencias de los alumnos en el uso del lenguaje inglés.

Agradecimientos

A mi familia por comprender y soportarme todo este tiempo. Por darme el apoyo para mi formación profesional, tanto económico como psicológico. Por dejarme perseguir mis metas, y por haberme criado de la manera en que lo hicieron.

A mi novia por estar ahí en las buenas y en las malas, por apoyarme en todo momento, comprenderme y soportarme. Por darme ánimos cuando más lo necesité, por incentivarme y darme soporte para seguir luchando.

A mi profesor guía por apoyarme en este proyecto y creer en mi potencial como futuro ingeniero.

A la profesora de inglés del establecimiento educacional que visité, por darme de su tiempo y apoyo para conseguir acceso al material educativo prudente para niños de enseñanza básica.

ÍNDICE GENERAL

1. Introducción	1
1.1. Antecedentes Generales	1
1.2. Justificación del Tema	2
2. Antecedentes.....	3
2.1. El mineduc y el ajuste curricular	3
2.2. Marco Teórico	5
2.2.1. Clasificación por contexto	5
2.2.2. Clasificación por control.....	6
2.2.3. Clasificación por herramientas de aprendizaje.....	6
2.2.4. Clasificación por comunicación	7
2.2.5. Clasificación por el sujeto y objetivo.....	7
2.3. Proyectos MALL (mobile assisted language learning)	8
2.4. Estado del arte en software comercial.....	9
3. Problema a resolver.....	11
3.1. Objetivos	11
3.2. Requerimientos	11
3.2.1. Unidades 1 y 2: In the classroom & At school:	12
3.2.2. Unidades 3 y 4: In the family & At home:.....	12
3.2.3. Unidades 5 y 6: Food and drinks & Healthy habits:	13
3.2.4. Unidades 7 y 8: Animal world & The world of nature:	13
4. Desarrollo de la solución.....	14
4.1. Arquitectura Física	14
4.2. Arquitectura Lógica	15
4.3. Diseño de clases	16
4.4. Archivos y recursos	26
4.5. Evaluación preliminar	34
5. Conclusiones	36
5.1. Del transcurso del proyecto.....	36
5.2. Objetivos alcanzados	36
5.3. Trabajo Futuro.....	37
6. Anexos.....	38
6.1. Plataforma Android.....	38
6.1.1. Android Software Development Kit	38

6.1.2. Android OS	39
6.1.3. Android Framework.....	41
6.1.3.1. Componentes de aplicación	41
6.1.3.2. Activación de componentes.....	42
6.1.3.2.1. Intents.....	42
6.1.3.2.2. Intent filters	43
6.1.3.3. Ciclo de vida de componentes.....	43
6.1.3.3.1. Activity	43
6.1.3.3.2. Service.....	45
6.2. Capturas de pantalla de la aplicación	47
6.2.1. Pantalla de inicio.....	47
6.2.2. Actividad “Complete the Sketch”	48
6.2.3. Actividad “Describe the Picture”	56
6.2.4. Actividad “Family Members”	61
6.2.5. Actividad “In the House”	65
6.2.6. Actividad “Food groups”	68
6.2.7. Actividad “Animals”	71
6.2.8. Actividad “Chilean Nature”	73
7. Referencias.....	77

ÍNDICE DE FIGURAS

Figura 1: Arquitectura Física	15
Figura 2: Arquitectura Lógica	16
Figura 3: Diagrama de clases del Sistema de Colaboración	17
Figura 4: Diagrama de clases de la ventana principal del Sistema de Actividades.....	18
Figura 5: Diagrama de clases de la actividad “Complete the Sketch”	19
Figura 6: Diagrama de clases de la actividad “Describe the picture”	20
Figura 7: Diagrama de clases de la actividad “Family Members”	21
Figura 8: Diagrama de clases de la actividad “In the house”	22
Figura 9: Diagrama de clases de la actividad “Food Groups”	23
Figura 10: Diagrama de clases de la actividad “Animals”	24
Figura 11: Diagrama de clases de la actividad “Chilean Nature”	25
Figura 12: Estructura de directorios de un proyecto típico Android	27
Figura 13: Recursos claves utilizados en el proyecto.....	28
Figura 14: Schema de Chilean Nature	29
Figura 15: Schema de Describe the Picture.....	30
Figura 16: Schema de Food Groups	31
Figura 17: Schema de Complete the Sketch.....	32
Figura 18: Schema de Vocabulary	33
Figura 19: Diagrama de la arquitectura del sistema operativo Android	39
Figura 20: Ciclo de vida de un objeto android.app.Activity	45
Figura 21: Ciclo de vida de un objeto android.app.Service	46
Figura 22: Pantalla de inicio.....	47
Figura 23: Actividad de vocabulario	48
Figura 24: Ajustes de personalización	48
Figura 25: Inicio de la actividad “Complete the Sketch”	49
Figura 26: Menú de “Complete the Sketch”	50
Figura 27: Instrucciones de los <i>sketchs</i>	51
Figura 28: Pintado de los <i>sketchs</i>	52
Figura 29: Compartir instrucciones – Alumno A	53
Figura 30: Compartir instrucciones – Alumno B	54
Figura 31: Nuevas instrucciones.....	55
Figura 32: <i>Sketch</i> concluido.....	56
Figura 33: Alumno que describe	57
Figura 34: Alumno que adivina la imagen	57
Figura 35: Alumno agrega descripciones	58
Figura 36: Opciones de objetos	58
Figura 37: Descripción enviada.....	59
Figura 38: Descripción recibida.....	59
Figura 39: Alumno elige la imagen.....	60
Figura 40: Imagen correctamente adivinada	61
Figura 41: Inicio de “Family Members”	62
Figura 42: Agregar un miembro	62
Figura 43: Tipo de miembro	63

Figura 44: Cumpleaños.....	63
Figura 45: Imagen.....	64
Figura 46: Edad	64
Figura 47: Miembro añadido a la lista	65
Figura 48: Más miembros	65
Figura 49: Inicio de “In the House”	66
Figura 50: Especificación del lugar	66
Figura 51: Imagen.....	67
Figura 52: Descripción libre	67
Figura 53: Lugar añadido	67
Figura 54: Más lugares	68
Figura 55: Imagen exterior de la casa.....	68
Figura 56: Inicio de “Food groups” - Alumno A.....	69
Figura 57: Inicio de “Food groups” - Alumno B.....	69
Figura 58: Alumno A clasifica un alimento	70
Figura 59: Alumno B clasifica otro alimento	70
Figura 60: Alumno B clasifica correctamente	71
Figura 61: Alumno A clasifica correctamente	71
Figura 62: Inicio de “Animals”	72
Figura 63: Añadir nuevo animal	72
Figura 64: Detalles del nuevo animal	73
Figura 65: Animal añadido	73
Figura 66: Inicio de “Chilean Nature”	74
Figura 67: Alumno selecciona una especie.....	74
Figura 68: Mostrar foto.....	75
Figura 69: Descripción de la especie	75
Figura 70: Descripción de otra especie	76

1. INTRODUCCIÓN

1.1. ANTECEDENTES GENERALES

Resulta particularmente importante para el desarrollo del país que sus alumnos posean habilidades competitivas referentes a la comunicación en inglés [1]. Es por esto que el Ministerio de Educación ha ajustado el marco curricular de la enseñanza de inglés como idioma extranjero. [1]

Este cambio, tiene como objetivo principal preparar a sus alumnos para desenvolverse utilizando el inglés como una herramienta que les permita acceder a la información, así como comunicarse de forma oral y escrita.

Históricamente, la enseñanza del inglés en Chile apuntaba a listas de contenidos donde el énfasis era la gramática y la metodología privilegiaba enseñar estructuras gramaticales aisladas de un contexto real de comunicación. El Mineduc señala al respecto: “Nuestra propia experiencia nos informa que durante décadas nuestros egresados tuvieron un conocimiento básico de la gramática inglesa, pero no comprendían lo que escuchaban o leían en inglés, no escribían y menos aun hablaban en inglés”. [1]

Se define entonces que los contenidos mínimos obligatorios del marco curricular deben contemplar el desarrollo de:

- Comprensión lectora
- Comprensión auditiva
- Expresión escrita
- Expresión oral

Estos objetivos, apuntan a que los alumnos se comuniquen con mensajes reales y no considerar al lenguaje inglés como un ejercicio teórico controlado por la gramática.

1.2. JUSTIFICACIÓN DEL TEMA

Dado el mayor interés en la enseñanza del inglés por parte del Mineduc, es preciso proveer las herramientas necesarias para apoyar el aprendizaje tanto en el aula como fuera de ella. De manera que los alumnos puedan aumentar su conocimiento y experiencia con el uso del lenguaje y aportar a la resolución de los problemas históricos que posee la enseñanza de inglés en Chile. [1]

Una solución tecnológica puede aportar en el aprendizaje, haciendo uso de elementos más intuitivos como los contenidos multimedia [32]. Estos permiten, por ejemplo, visualizar de mejor manera los usos, contextos y significados de las palabras en una comunicación, manteniendo el foco de atención y el interés del estudiante. Este apoyo tecnológico sería preferible a los contenidos estáticos.

Además, una solución tecnológica permite aprovechar la motivación y entusiasmo en el mismo instante en que el estudiante requiere su uso, sin la necesidad de esperar a la siguiente clase o estar en un ambiente relacionado al contexto del estudio. Permite también seguir el ritmo del estudiante, ahondando en los contenidos en que éste está débil y apresurando aquellos que ya maneja con soltura. De esta manera, el uso de la tecnología puede flexibilizar el proceso del aprendizaje, moldeándolo a las preferencias y posibilidades de sus estudiantes. [33]

Uno de los métodos de enseñanza de una lengua extranjera más efectivos corresponde a la inmersión [34]. Este método utiliza la premisa de que un lenguaje extranjero puede aprenderse de igual manera que un lenguaje materno. Esto es, utilizando el lenguaje como una herramienta de comunicación para todas las actividades en vez de un tema de aprendizaje. Existen aplicaciones comerciales que intentan hacer uso de este método y en la literatura se pueden encontrar estudios sobre la inmersión utilizando realidades virtuales [2]. Luego, una solución tecnológica podría utilizar estos elementos para la enseñanza del inglés en Chile.

De cumplirse los objetivos del sistema, se estaría aportando significativamente a la enseñanza del inglés, apoyando los objetivos de la reforma educacional: mejorar la competencia de los egresados de la educación media, permitiendo a los alumnos mejores oportunidades laborales y de educación. [1]

El sistema de apoyo para la enseñanza debe entonces tener un alineamiento con los objetivos, contenidos y actividades propias de la reforma educacional del inglés [35, 36]. Más aún, no sólo se puede apoyar la educación dentro del aula sino además potenciar una conexión fluida y directa con las actividades educativas fuera del aula.

2. ANTECEDENTES

En el siguiente capítulo, se mostrarán las bases y antecedentes recopilados previos al desarrollo de la solución tecnológica propuesta en este tema de memoria. Desde el ajuste curricular del Mineduc hasta una revisión del estado del arte en mobile learning y proyectos relevantes.

2.1. EL MINEDUC Y EL AJUSTE CURRICULAR

El currículum del sector de aprendizaje inglés propuesto por el Mineduc [1] tiene como objetivos:

- Entregar a los estudiantes las habilidades necesarias para utilizar el idioma como una herramienta que les permita acceder a la información, así como resolver situaciones comunicativas simples en forma oral y escrita.
- Promover el desarrollo de habilidades cognitivas de orden superior y desarrollar la capacidad de apreciar otros estilos de vida, tradiciones y maneras de pensar.

Para lograrlo, de acuerdo al ajuste curricular se propone:

- Adoptar una orientación comunicativa más allá de los aspectos formales, con especial atención en los mensajes y usos del lenguaje.
- Definir Objetivos Fundamentales y Contenidos Mínimos Obligatorios (OF-CMO) específicos del inglés para el desarrollo explícito (y con igual relevancia) de las 4 habilidades del lenguaje: comprensión lectora, comprensión auditiva, expresión escrita y expresión oral.
- Alineación de los objetivos de aprendizaje a estándares internacionales. En particular el marco común europeo para el aprendizaje de idiomas CEF.
- Ajustar la progresión de aprendizajes desde 5º básico a 4º medio. De manera que los OF-CMO tengan un crecimiento graduado a lo largo de los 8 años. Se incorpora además un OF para el desarrollo del léxico y otro que conecta el inglés a los objetivos fundamentales transversales de formación y crecimiento personal.
- Actualización del currículum de acuerdo a la investigación de cómo se aprende un idioma extranjero:
 - El rol que cumple el aprendizaje del vocabulario en la adquisición y uso de una lengua extranjera.
 - 2000 palabras de uso frecuente cuyo conocimiento permite comprender el 80% de un texto en inglés sobre un tema general.
- Reconocer el uso de la lengua materna como herramienta en el proceso de aprendizaje.

Es particular, para los primeros 2 años de aprendizaje se definen los siguientes
Objetivos Fundamentales:

1. Escuchar y reconocer los sonidos del nivel, y demostrar comprensión del lenguaje propio de la sala de clases, de palabras y frases de uso frecuente en textos orales muy breves y simples relacionados con los temas del nivel.
2. Leer y comprender palabras y frases de uso frecuente, información explícita destacada del tema en textos muy breves y simples relacionados con los temas del nivel.
3. Expresarse oralmente en situaciones comunicativas de alta frecuencia relacionadas con los temas del nivel, utilizando palabras y frases y oraciones muy breves y simples, y una pronunciación inteligible de los sonidos del nivel.
4. Escribir textos muy breves, con fines instructivos y descriptivos, relacionados con los temas del nivel, utilizando frases y oraciones muy simples.
5. Comprender en textos orales y escritos 300/500 palabras que incluyen el vocabulario temático y el vocabulario de uso frecuente y utilizar parte de este vocabulario, en forma oral y escrita, de acuerdo a sus necesidades y propósitos comunicativos.
6. Descubrir la presencia del inglés en su entorno y acercarse a su aprendizaje con una actitud de respeto que contribuya a su crecimiento personal y valoración de una sociedad pluralista.

Y como Contenidos Mínimos Obligatorios se define: ¹

Temas:

- La escuela y la sala de clases.
- La casa y la familia.
- La comida y la salud.
- El entorno natural inmediato (flora y fauna).

Comprensión auditiva: Interacción con profesores y pares, además de la audición de textos con una serie de características que en términos generales deben ser auténticos, cortos, directos, claros y cumplir con el contenido del aprendizaje asociado al nivel.

Comprensión lectora: Lectura clase a clase de textos tipo instructivo y descriptivo, tales como instrucciones para realizar tareas, menús, recetas, notas, avisos, tarjetas de saludos, postales e invitaciones. En términos generales deben ser auténticos, cortos, directos y cumplir con los contenidos del aprendizaje asociado al nivel.

Expresión oral: Reproducción y producción frecuente de textos orales como diálogos, monólogos, cánticos, canciones, poemas, adivinanzas y trabalenguas muy breves y simples, utilizando los contenidos del aprendizaje asociado al nivel.

¹ La especificación completa detallada aparece en [11]

Expresión escrita: Producción de textos guiados muy breves y simples tales como: notas, tarjetas de saludos e invitaciones. Incluyen elementos ortográficos como el punto y mayúscula, además de utilizar los contenidos del aprendizaje asociado al nivel.

Además de la producción y ejercicio de las actividades por separado, se señala la necesidad de asociar la forma escrita a la oral, así como del apoyo audiovisual.

Las actividades asociadas sugeridas del programa no están actualizadas aún para este ajuste a la reforma [12], la última versión corresponde a la edición del 2004, que utiliza el programa puesto en marcha en 1999 [13] [14]. Pero, dado que no hay un cambio en el enfoque del currículum vigente, se considera apropiado guiarse por este programa en el marco de los OF-CMO del ajuste.

2.2. MARCO TEÓRICO

Varios proyectos de investigación se han realizado referentes al uso de tecnologías móviles para el aprendizaje [6]. Los autores de este análisis crítico, proponen una clasificación de los proyectos de acuerdo a su contexto, herramientas, control, comunicación, sujeto y objetivo, utilizando como *framework* de análisis el “Task model for mobile learning” de Taylor *et. al.* [7].

De acuerdo a los autores, el estado del arte en proyectos de investigación incluye proyectos que aportan valor al aprendizaje de distintas formas: ya sea en motivación, incremento de actividad física/mental, aumento de control de estudiantes, enriquecimiento del entorno, etc. Sin embargo, no hay un valor agregado para el campo de aprendizaje móvil en términos generales. Y, por lo tanto, corresponde a un área aún inmadura que no ha desarrollado todo su potencial.

2.2.1. CLASIFICACIÓN POR CONTEXTO

La clasificación de proyectos por su **contexto** se divide en 4 categorías:

- **Independiente:** Cuando el contexto de estudio no tiene relación con el ambiente actual del estudiante. Típicamente en proyectos del tipo *content-delivery*.
- **Formal:** cuando el ambiente de estudio corresponde a una sala de clases o similar. Donde normalmente se busca incentivar la participación de los estudiantes en el aula.
- **Físico:** El lugar de estudio es importante. En general es estudio en torno a un contexto dado; como un museo. El software es más inteligente y típicamente utiliza herramientas del tipo *context-awareness* para facilitar el estudio indicado.
- **Social:** Lo importante son las relaciones inter-personales y experiencias de los estudiantes, incluyendo el aprendizaje de las situaciones de la vida diaria. Según

los autores no existen proyectos que sean categorizados completamente bajo esta categoría.

Bajo esta clasificación, los autores resaltan el poco desarrollo de la investigación en el contexto social del aprendizaje, así como el nivel relativo de dificultad/ambición en el desarrollo de proyectos desde el nivel Independiente al Social.

2.2.2. CLASIFICACIÓN POR CONTROL

Otra clasificación realizada por los autores corresponde al **control**, donde la escala comprende los extremos del **control completo por parte del profesor** al **control completo del estudiante**. Ambos extremos, explican los autores, tienen ventajas y desventajas; por una parte la falta de motivación, poca participación e insuficiente creatividad cuando hay un control excesivo del profesor y, por otra parte, los peligros de confusión, malinterpretación y frustración cuando hay un control completo de los estudiantes.

De esta forma, consideran prudente un **control intermedio y dinámico**, comenzando con un esquema de aprendizaje guiado en un comienzo y permitiendo una mayor libertad con el tiempo. Se debe prestar, tal vez, especial atención en diferenciar el control individual del grupal.

2.2.3. CLASIFICACIÓN POR HERRAMIENTAS DE APRENDIZAJE

La clasificación de proyectos por las **herramientas de aprendizaje** se dividen en:

- **Distribución de contenido:** El estudiante sólo consume contenido previamente preparado, no hay una mayor interacción o personalización de contenidos.
- **Interacción para el control:** El estudiante aún consume contenido preparado, pero se considera un nivel de interacción de manera que el estudiante puede realizar actividades como responder preguntas de opción múltiple o buscar información. De esta forma el profesor puede hacer un seguimiento de las actividades del alumno.
- **Reflexión guiada:** El estudiante recibe menos contenido, en cambio, recibe tareas relacionadas a su entorno. De esta forma se incentiva la reflexión continua del estudiante y su entorno.
- **Recopilación de información reflexiva:** El estudiante debe explorar el ambiente por sí mismo y llegar a obtener datos genuinamente. La herramienta solamente recopila los datos.
- **Construcción del contenido/conocimiento:** Los estudiantes trabajan activamente con herramientas y producen el conocimiento por sí mismos, logrando un conocimiento aplicado y una comprensión más profunda.

Los autores señalan que la gran mayoría de proyectos residen en la interacción para el control o inferior, claramente con poca ambición pedagógica. Aunque hay una minoría importante que explora estas áreas.

2.2.4. CLASIFICACIÓN POR COMUNICACIÓN

La **comunicación** corresponde a un cuarto aspecto de clasificación. Los autores enfatizan que el proceso de aprendizaje siempre se ve afectado por la interacción y comunicación entre alumnos y el profesor, aumentando la reflexión y un entendimiento más profundo.

Esta clasificación tiene entonces las siguientes categorías:

- **Aislado:** El alumno realiza las actividades de aprendizaje completamente en solitario, sólo interactúa con las herramientas.
- **En parejas:** Los alumnos comparten las herramientas y el material en parejas, pero el contexto de aprendizaje no incentiva la interacción entre ellos.
- **En parejas cohesionadas:** Los alumnos comparten las herramientas y el material en parejas, pero además se incentiva la comunicación e interacción entre ellos. Esto puede incentivar la participación y reflexión permitiendo un mejor entendimiento, además de acelerar el proceso de aprendizaje.
- **Inter-grupos:** Se potencia la comunicación entre varios grupos o parejas para mejorar la reflexión mutua.
- **Colaboración:** Se fuerza la colaboración entre los equipos para completar los objetivos de aprendizaje.

Es pedagógicamente importante, según los autores, promover la comunicación, interacción y cooperación entre los estudiantes para lograr un entendimiento más profundo.

2.2.5. CLASIFICACIÓN POR EL SUJETO Y OBJETIVO

Otros factores a considerar en los proyectos de aprendizaje móvil son el **sujeto** del aprendizaje y el **objetivo** de este. En general, señalan los autores, los proyectos se enfocan en estudiantes novatos en el tema de aprendizaje, probablemente porque es considerablemente más fácil evaluar el impacto del proyecto y este público objetivo es más abundante. Más aún, considerando este público objetivo, los contextos de los proyectos son típicamente del tipo independiente o formal, ya que no serían adecuados los contextos físico o social para estos, ya que requieren que los estudiantes sean capaces de generar conocimiento.

Los autores concluyen en la importancia de cambiar el enfoque del aprendizaje móvil hacia un público objetivo más experimentado, de manera de explotar mejor el potencial de las herramientas móviles.

2.3. PROYECTOS MALL (MOBILE ASSISTED LANGUAGE LEARNING)

Existen múltiples proyectos que se enfocan en el aprendizaje de un idioma extranjero, LOCH (Language learning Outside Classroom with Handhelds) es uno de ellos. [8]

En este proyecto, se utilizaron PDAs para aventurar a un grupo de alumnos extranjeros a aprender japonés en un contexto de situaciones de la vida real en Japón. El proceso de aprendizaje era iniciado por un profesor quien asignaba tareas que requerían que los alumnos interactuasen con personas nativas de la ciudad, tales como:

- Entrevistar a alguien: Los estudiantes debían entrevistar a alguien en japonés, el sistema guardaba el audio de la entrevista y se tomaban fotos de la experiencia.
- Obtener información: Los estudiantes debían ir a un lugar específico y obtener información relevante del lugar, como el estacionamiento: tarifas, capacidad de estacionamientos, horarios, etc.
- Comprar algo local: Los estudiantes iban a un mercado y compraban algún producto típico, consultando como se preparaba.
- Tener una experiencia: Como solicitar un test de presión arterial.

Al realizar las tareas, los alumnos podían registrar texto, audio y fotos en sus PDAs. El profesor a cargo de la actividad monitoreaba constantemente el progreso de los alumnos mediante el posicionamiento GPS, también podía contactar a sus alumnos de manera síncrona o asíncrona para ayudarlos con sugerencias.

Al final de la actividad, los alumnos se reunían con el profesor y compartían sus experiencias junto al material reunido. Evaluaciones preliminares de los autores sugirieron que el sistema era bien aceptado por los estudiantes y profesores, motivándolos positivamente. Se trata entonces, de un ejemplo de inmersión real, no virtual.

Otros proyectos orientados al aprendizaje de un idioma extranjero, utilizan diferentes tecnologías para abordar un aprendizaje específico. El proyecto TANGO (Tag-added Learning Objects) [9] busca enseñar vocabulario basado en lugares y objetos especialmente marcados con etiquetas RFID (Radio-frequency identification), de manera que el sistema puede reconocer el entorno y guiar al estudiante. El proyecto JAPELAS (Japanese Polite Expressions Learning Assisting System) [9], utiliza la misma tecnología pero para enseñar expresiones formales e informales dependiendo del contexto: lugar, personas, edades, rango. El proyecto JAMIOLAS (Japanese Mimicry and Onomatopoeia Learning Assisting System) [9] en cambio, pretende apoyar al estudiante en el aprendizaje de expresiones especiales como onomatopeyas y mímicas japonesas. Para esto emplea una red de sensores inalámbricos dispuestos en diferentes lugares que registran variables ambientales como temperatura, luminosidad, sonido, humedad, etc.

Otro enfoque en el aprendizaje de un idioma se ve ejemplificado en el proyecto PIMS (Personalized Intelligent Mobile Learning System) [10], donde el sistema evalúa el vocabulario y habilidades del estudiante para proponer diferentes lecturas de artículos de noticias en inglés. La teoría de este enfoque se basa en diversos estudios que señalan que la adquisición de un segundo lenguaje depende en gran medida de la comprensión de lectura y el vocabulario manejado, y que la lectura extensiva es efectiva en mejorar la comprensión de lectura. Para ayudar al estudiante en la comprensión y adquisición de vocabulario, el sistema provee traducciones de párrafos a su idioma nativo, argumentando para esto que algunos autores señalan que la utilización de las formas lingüísticas en su forma nativa para comprender aquellas en el idioma extranjero influyen positivamente la comprensión de lectura.

Los estudios estadísticos conducidos por los autores muestran que la comprensión de lectura de sus participantes luego de 5 semanas mejoró significativamente, más aún, el beneficio de aprendizaje y habilidad de lectura fue mayor en el grupo de estudiantes con menor habilidad inicial.

2.4. ESTADO DEL ARTE EN SOFTWARE COMERCIAL

Actualmente los paquetes de software de apoyo al aprendizaje del inglés son principalmente aplicaciones de escritorio comerciales. Los de mayor calidad presentan típicamente lo siguiente:

- Actividades
 - Escritura corta
 - Gramática
 - Pronunciación
 - Vocabulario
- Características del software
 - Reconocimiento del habla
 - Seguimiento del progreso
- Multimedia
 - Imágenes
 - Sonido (Conversaciones y palabras)
 - Video o secuencia de imágenes
 - Textos (diccionario, traducciones, historias)

Algunos ejemplos de dichos software son Rosetta Stone [3] y Tell Me More [4].

Estos paquetes de software tienen muy ricos contenidos y parecen muy completos. Sin embargo, por su carácter comercial, carecen de la posibilidad de administrar el contenido que se enseña o los multimedia que utilizan.

Además, demandan que el estudiante permanezca inmóvil en un entorno aislado. Se desaprovecha así tal vez, el hecho que múltiples alumnos estudien lo mismo y deseen realizar un aprendizaje conjunto.

Otras soluciones comerciales buscan atacar el problema del aprendizaje en un mundo acelerado en que el estudiante típicamente no tiene tiempo para realizar un aprendizaje presencial (como es el caso de los adultos que trabajan). Por lo tanto, utilizan la tecnología móvil como receptor de contenidos para el estudiante o como plataforma de juegos con contenido educativo, tal es el caso del programa “Learn English” del British Council [5].

3. PROBLEMA A RESOLVER

Teniendo en cuenta los antecedentes, la opción de solución contempla el uso de un sistema móvil para apoyar el aprendizaje del inglés, por las siguientes razones:

- Los dispositivos móviles flexibilizan el ambiente de estudio, permitiendo actividades en entornos distintos al de un escritorio o sala de clases.
- Los dispositivos móviles son menos invasivos en el uso del espacio y, por lo tanto, interfieren en menor medida con la comunicación e interacción entre pares o grupos.
- Los costos y prestaciones de los dispositivos móviles los hacen cada vez más convenientes.
- La penetración de mercado de la telefonía móvil en la población chilena es cada vez mayor, llegando a superar el 100% durante el año 2010. [38]

Por otra parte, debido a las particularidades del problema educativo y al no poseer una contraparte cliente clara, los requerimientos de este sistema se proponen en base a los antecedentes y documentos del Mineduc.

3.1. OBJETIVOS

El objetivo principal de este tema de memoria es apoyar la educación del idioma extranjero inglés haciendo uso de la tecnología. El enfoque es para los alumnos de los primeros años de enseñanza básica de acuerdo a la reforma (5to y 6to básico).

Los objetivos específicos abordados contemplan:

- Diseñar e implementar una versión inicial de solución tecnológica móvil que permita apoyar la enseñanza dentro del aula, permitiendo actividades grupales/colaborativas con énfasis en el desarrollo de habilidades de comprensión auditiva y expresión oral.
- Diseñar e implementar algunas actividades siguiendo las recomendaciones del Ministerio de Educación de acuerdo a su programa de estudio y marco curricular.
- Evaluar tentativamente, en un ambiente simulado, el sistema resultante para recoger las observaciones iniciales que se obtengan.

3.2. REQUERIMIENTOS

1. El sistema debe proveer la capacidad de realizar actividades de aprendizaje grupales.

2. Las actividades de aprendizaje deben tener una sintonía con lo propuesto por el Mineduc.
3. Los contenidos de aprendizaje deben ser configurables.
4. El software móvil debe ser lo suficientemente eficiente para responder al usuario, minimizando los retardos en las respuestas.

Las actividades educativas propuestas para el sistema son:

3.2.1. UNIDADES 1 Y 2: IN THE CLASSROOM & AT SCHOOL:

- Completar un dibujo relacionado de forma colaborativa:
 - Un sketch pre-definido contiene objetos comunes de la sala de clases. 2 Alumnos reciben pistas diferentes y complementarias sobre como pintar los objetos, deben escoger entre las alternativas para comunicarse entre sí los colores correctos para cada objeto. La pareja que complete sus dibujos antes gana.
- Describir una imagen de acuerdo a las cantidades de objetos / lugares:
 - De un set de imágenes un alumno recibe 1 para que se la describa a su compañero. De un conjunto de fragmentos de frase, arma las oraciones que describen la cantidad de un objeto determinado en la imagen y se lo transmite a su compañero. El compañero debe buscar entre el conjunto de imágenes aquellas que podrían ser la descrita.

3.2.2. UNIDADES 3 Y 4: IN THE FAMILY & AT HOME:

- Compartir información acerca de los miembros de la familia.
 - Los alumnos toman fotografías con sus dispositivos a los miembros del hogar y de algunos miembros de la familia (a partir de álbumes familiares). Pueden agregar descripciones genéricas guiadas en inglés como: Nombre, edad, parentesco, etc.
 - Con compañeros discuten la información utilizando las frases, realizan preguntas y respuestas, deletrean los nombres en inglés.
- Compartir información acerca de las habitaciones de las casas.
 - Los alumnos toman fotografías con sus dispositivos a las habitaciones de la casa, como: la cocina, el comedor, el living y el dormitorio. Pueden hacer anotaciones sobre los objetos que hay en cada habitación.
 - Con compañeros describen las imágenes de las habitaciones y sus objetos. Realizan preguntas y respuestas acerca de los objetos y sus posiciones.

3.2.3. UNIDADES 5 Y 6: *FOOD AND DRINKS & HEALTHY HABITS:*

- Completar tabla de alimentos con compañeros.
 - En pares o grupos, completar una tabla de alimentos clasificados como Frutas / Verduras / Bebidas / Comida chatarra
 - Comentar los gustos, likes / dislikes.
 - Comentar sobre la comida típica chilena. ¿Cuáles son los platos típicos, que ingredientes llevan?

3.2.4. UNIDADES 7 Y 8: *ANIMAL WORLD & THE WORLD OF NATURE:*

- Traer una lista de imágenes con animales favoritos.
 - Tomar o cargar imágenes de animales favoritos, agregarle descripciones tales como: características distintivas / partes del cuerpo, hábitat, alimentos, comportamiento, que cosas pueden hacer.
 - Discutir en pares o grupos acerca de los animales. What are – like? What are – for? Can – do – ?
- Flora y fauna chilena
 - ¿Qué animales habitan en Chile? ¿Cuáles son sus hábitats?
 - ¿Existen animales en extinción en Chile, cuáles?
 - Comentar sobre los símbolos patrios: Cóndor, Huemul, Copihue.

4. DESARROLLO DE LA SOLUCIÓN

Para el desarrollo de la solución se consideraron las plataformas móviles de Apple iPhone (iOS), Microsoft Windows Mobile, Symbian, JavaME y Android. Pero finalmente se optó por Android, debido a las siguientes razones:

- El costo de las herramientas de desarrollo:
 - Son gratis tanto para Android como para JavaME, el resto son de carácter propietario. Apple iOS SDK incluso requiere Mac OS.
 - Las herramientas para Windows Mobile en tanto, serían gratis debido a convenios con la Universidad de Chile.
- La curva de aprendizaje y facilidad para encontrar información:
 - Android posee un sitio oficial exclusivamente para la documentación y desarrollo en la plataforma. Asimismo hay disponibles muchas conferencias y *podcasts* en la web para diversos niveles de complejidad. Además de varias comunidades activas de desarrollo.
 - JavaME, en tanto, posee amplia documentación y varias comunidades giran en torno a la plataforma. Pero mucha de esta es demasiado antigua y no aplica a las necesidades actuales.
- El lenguaje de programación. Debido a que el tiempo es limitado, es deseable el uso de un lenguaje familiar, en particular Java y C/C++:
 - Android permite el desarrollo de aplicaciones principalmente en Java. También es posible utilizar C/C++ para algunos componentes que necesiten incluir librerías nativas, o para mejorar el rendimiento.
 - Apple iOS utiliza el lenguaje Objective-C.
 - Microsoft Windows Mobile utiliza C#.
 - Symbian utiliza actualmente de forma estándar C++, por medio de Qt. [39] También soporta otros lenguajes de programación como Python o JavaME.
- La penetración de mercado: Según la cuota de mercado y las tendencias del 2010, es esperable que la plataforma Android sea el principal actor en el mercado de los smartphones. [15]

4.1. ARQUITECTURA FÍSICA

La solución planteada comprende el uso de dispositivos móviles (smartphones) para la realización de actividades de aprendizaje, para el caso de actividades colaborativas se pueden utilizar múltiples dispositivos. La arquitectura física entonces involucrada corresponde a la de la figura 1.

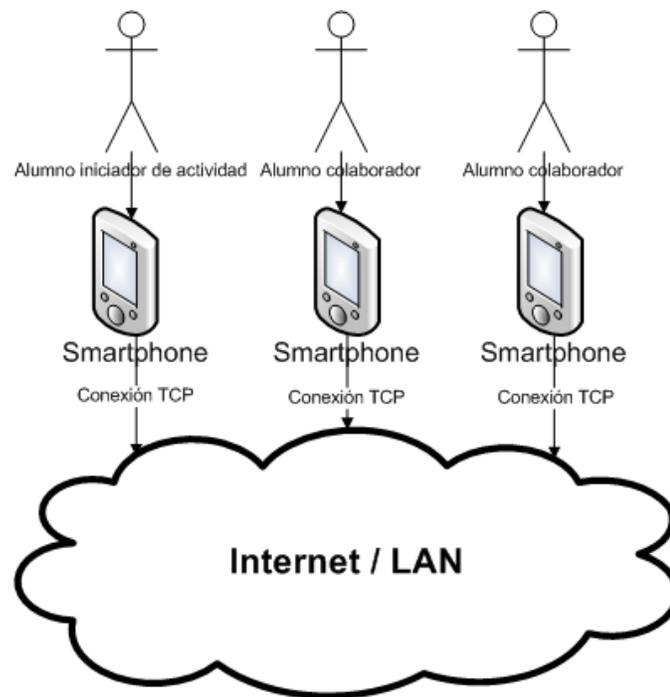


Figura 1: Arquitectura Física

4.2. ARQUITECTURA LÓGICA

Cada dispositivo ejecuta el mismo software, pero con roles distintos. La arquitectura utilizada corresponde a cliente-servidor, donde el servidor es ejecutado en el dispositivo que inicia la actividad, y sus colaboradores (clientes) se conectan a él.

El sistema de actividades contiene las actividades de aprendizaje que los estudiantes realizarán, mientras que el sistema de comunicación se encarga de mantener la conexión y el flujo de datos entre los dispositivos que realizan una actividad colaborativa. El sistema como un todo se encuentra inmerso en la plataforma Android, la cual por medio de su *framework* provee la funcionalidad necesaria para que el sistema ejecute sus requerimientos, tal y como lo muestra el diagrama de la figura 2.

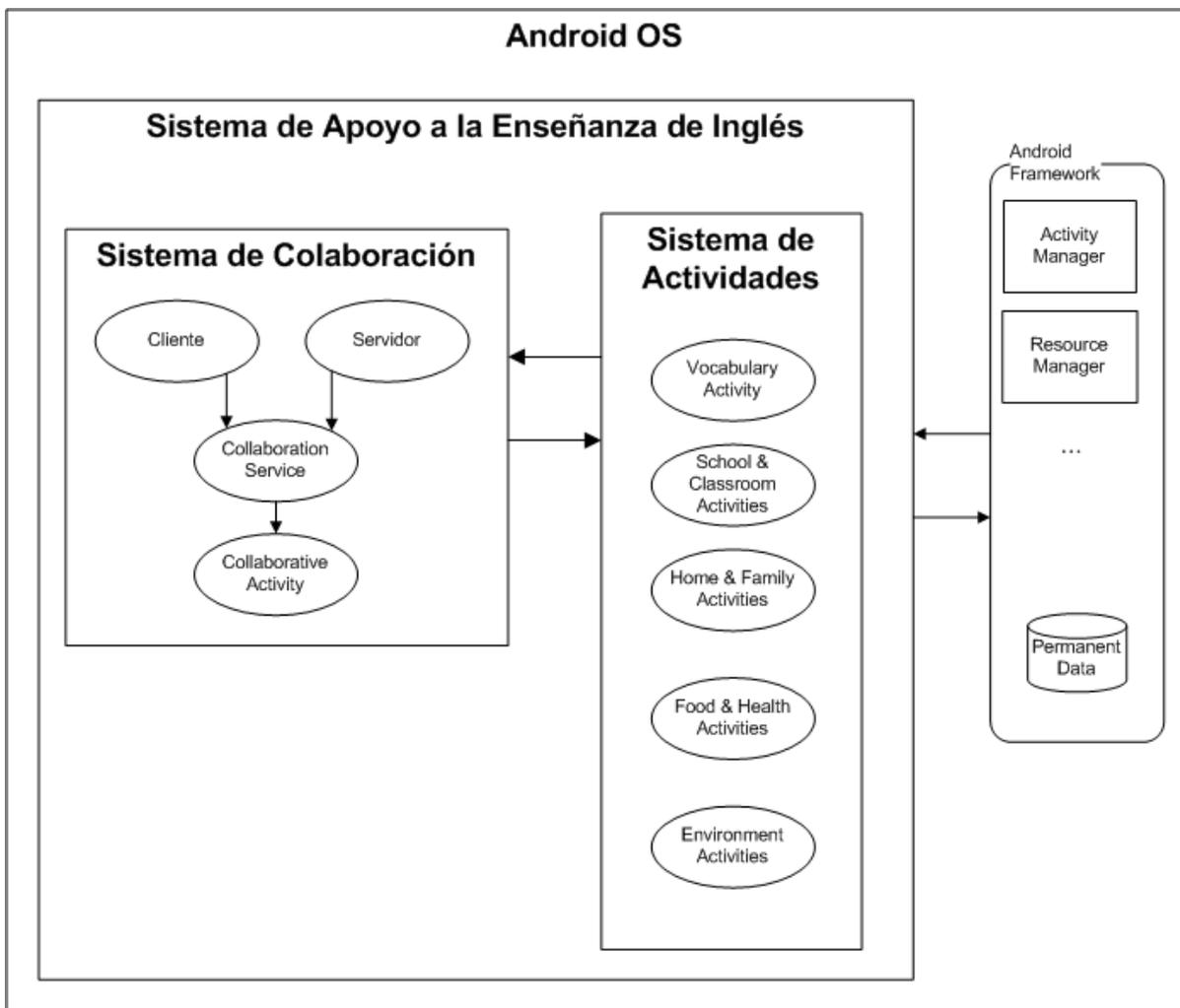


Figura 2: Arquitectura Lógica

4.3. DISEÑO DE CLASES

El diagrama de clases de los distintos componentes del sistema muestra como los componentes interactúan entre ellos para llevar a cabo las actividades de aprendizaje.

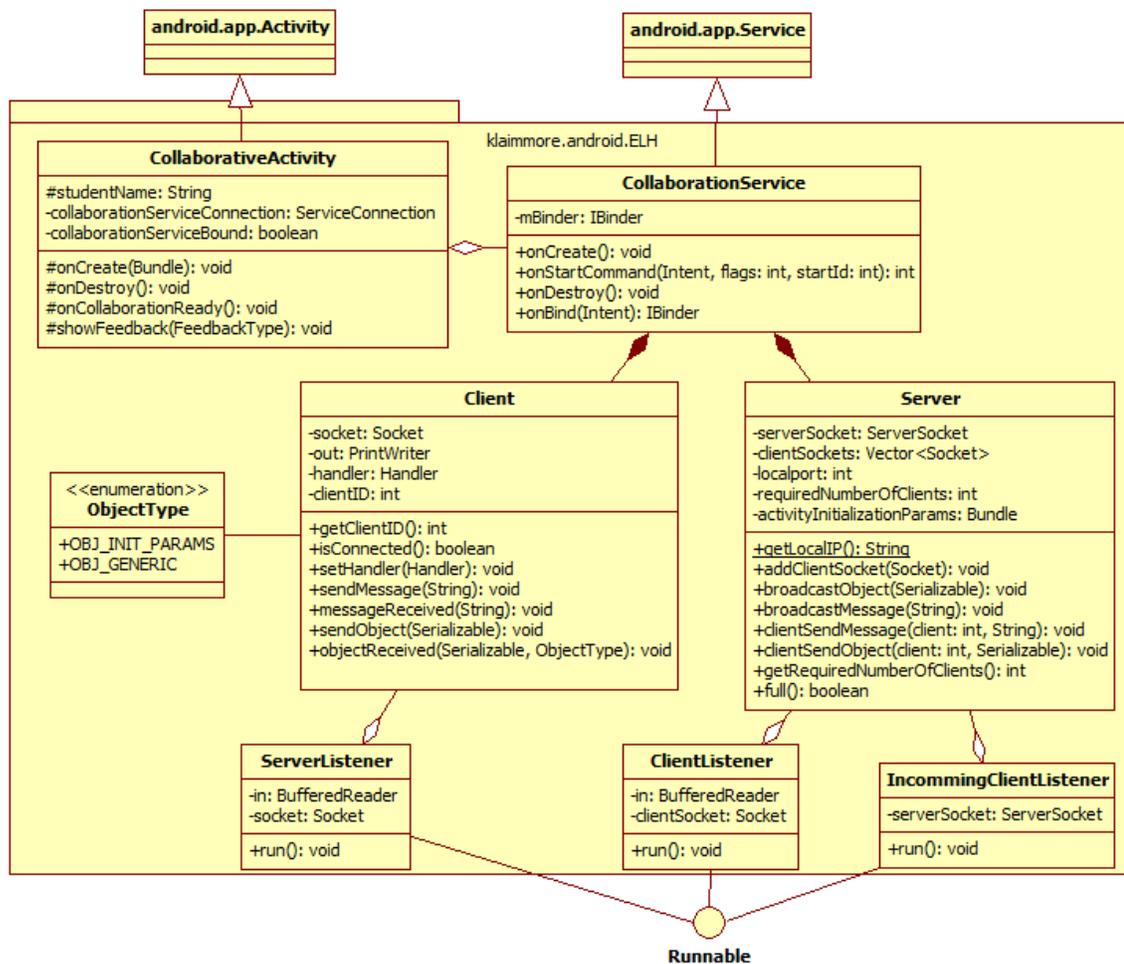


Figura 3: Diagrama de clases del Sistema de Colaboración

El Sistema de Colaboración de la figura 3 implementa los siguientes componentes:

- **Client:** Es la clase encargada de transferir datos u objetos con el `Server` de la actividad. Los datos pertinentes los deriva a la actividad actual por medio de un `android.os.Handler`.
- **Server:** Es la clase encargada de mantener una conexión activa con los clientes de la actividad, contiene los parámetros de inicialización de la actividad actual y los envía a los clientes al comienzo de ésta.
- **CollaborationService:** Es el componente que administra el ciclo de vida del cliente y/o servidor. Hereda de `android.app.Service`, y se mantiene activo durante el ciclo de vida de la actividad hasta que no haya más clases clientes que lo requieran.
- **CollaborationActivity:** Las clases que requieren hacer uso del sistema de colaboración heredan de la clase `CollaborationActivity` la cual realiza las

conexiones pertinentes con el `CollaborationService` y puede informar a las clases clientes el momento en que las conexiones están listas para ser usadas.

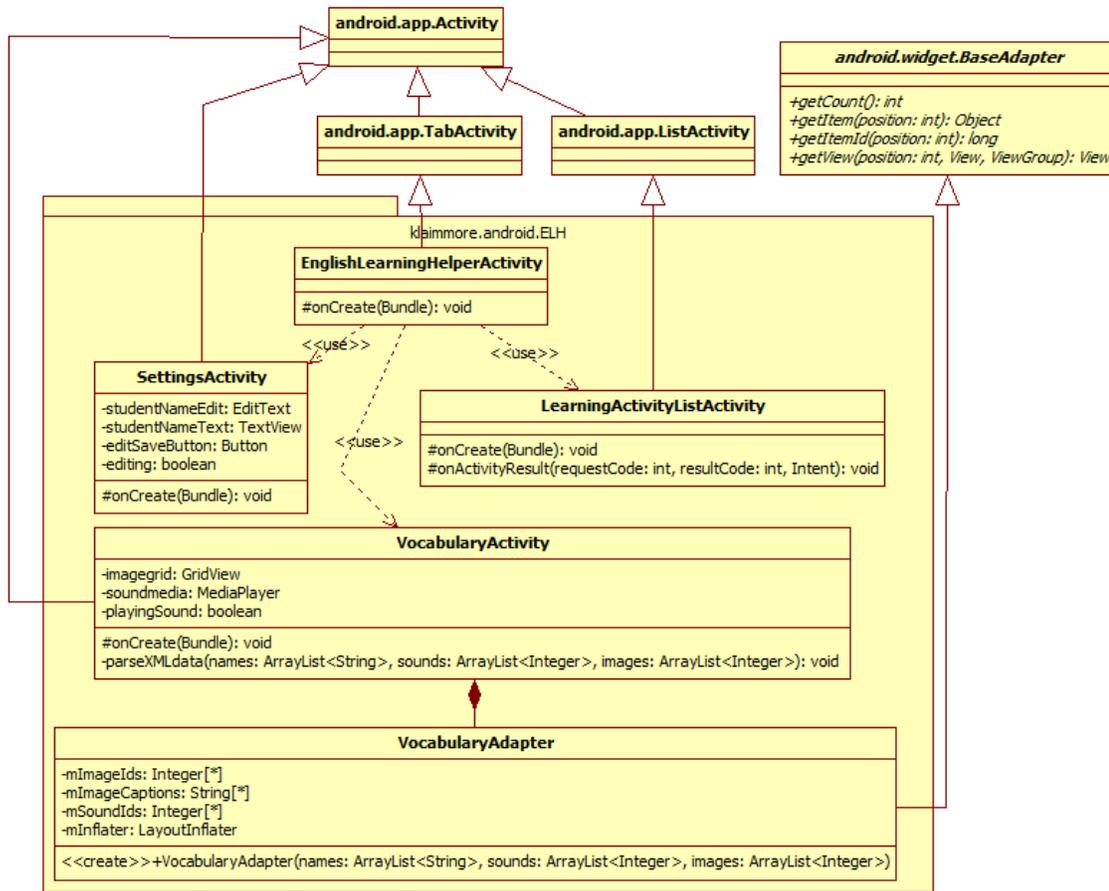


Figura 4: Diagrama de clases de la ventana principal del Sistema de Actividades

El Sistema de Actividades (parcial) de la figura 4 corresponde al resto de la aplicación. Los componentes ilustrados son:

- **EnglishLearningHelperActivity:** Es la ventana principal y la cara visible de la aplicación, desde ella se navega a las demás secciones. Su *layout* corresponde a una interfaz de tres pestañas:
 - **LearningActivityListActivity:** Es la encargada de mostrar e iniciar cada una de las actividades de aprendizaje del sistema.
 - **VocabularyActivity:** Corresponde a una actividad de vocabulario, donde se muestra una grilla de imágenes con su nombre asociado en inglés, al seleccionar un elemento su pronunciación es reproducida.
 - **SettingsActivity:** Es utilizada para configurar o personalizar la aplicación. Por el momento solo se puede personalizar el nombre del estudiante utilizado al realizar actividades colaborativas.

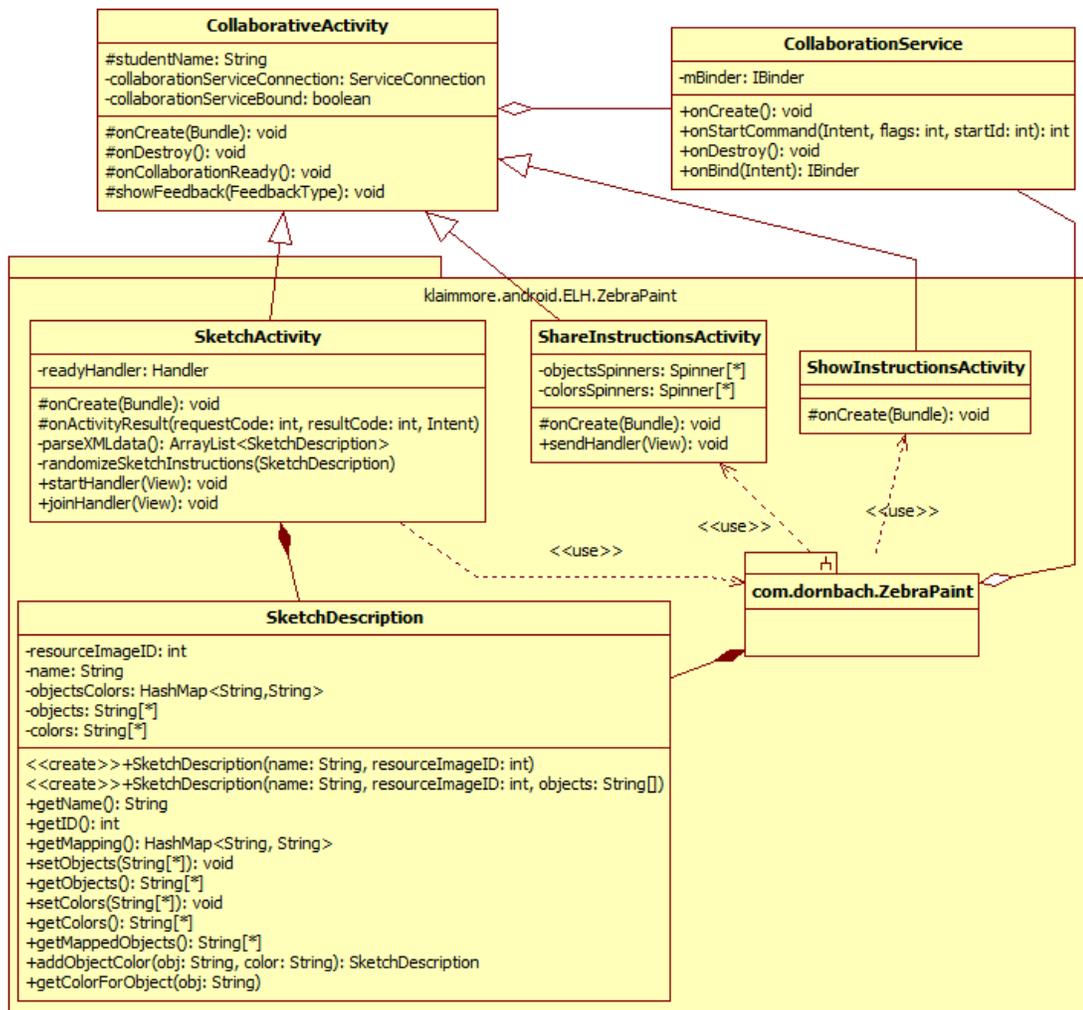


Figura 5: Diagrama de clases de la actividad “Complete the Sketch”

La primera actividad de aprendizaje del sistema corresponde a “Complete the Sketch” de la unidad de aprendizaje “School and Classroom” ilustrada en el diagrama de la figura 5. Sus componentes son:

- **SketchActivity:** Al tratarse de una actividad colaborativa, hereda de la clase CollaborativeActivity. SketchActivity se encarga de definir los parámetros de inicialización de la actividad, iniciando un servidor y cliente en caso de iniciar la actividad en el mismo dispositivo, o iniciando un cliente en caso de unirse a la actividad de otro dispositivo. Cuando la actividad está lista para iniciarse el servidor envía la señal de inicio y se ejecuta el componente externo ZebraPaint.
- **ZebraPaint:** Es una aplicación *open source* que provee la funcionalidad para cargar *sketch* de imágenes para ser pintadas con una paleta de colores [37]. En

el sistema es utilizada para la misma funcionalidad. Para utilizar este software se realizaron modificaciones en la rutina de inicio para cargar una imagen a partir de un parámetro recibido vía un `android.content.Intent`, además de cambios en el menú para lanzar los demás componentes de la actividad. De esta forma se cumple con el objetivo de la actividad de aprendizaje: asociar colores y objetos con sus nombres en inglés.

- **ShowInstructionsActivity:** Es invocado vía el menú de **ZebraPaint**, muestra las instrucciones para completar el *sketch*, tanto las iniciales como las enviadas por los compañeros de actividad.
- **ShareInstructionsActivity:** Es invocado vía el menú de **ZebraPaint**, permite compartir instrucciones para completar el *sketch* actual con los demás compañeros de actividad.

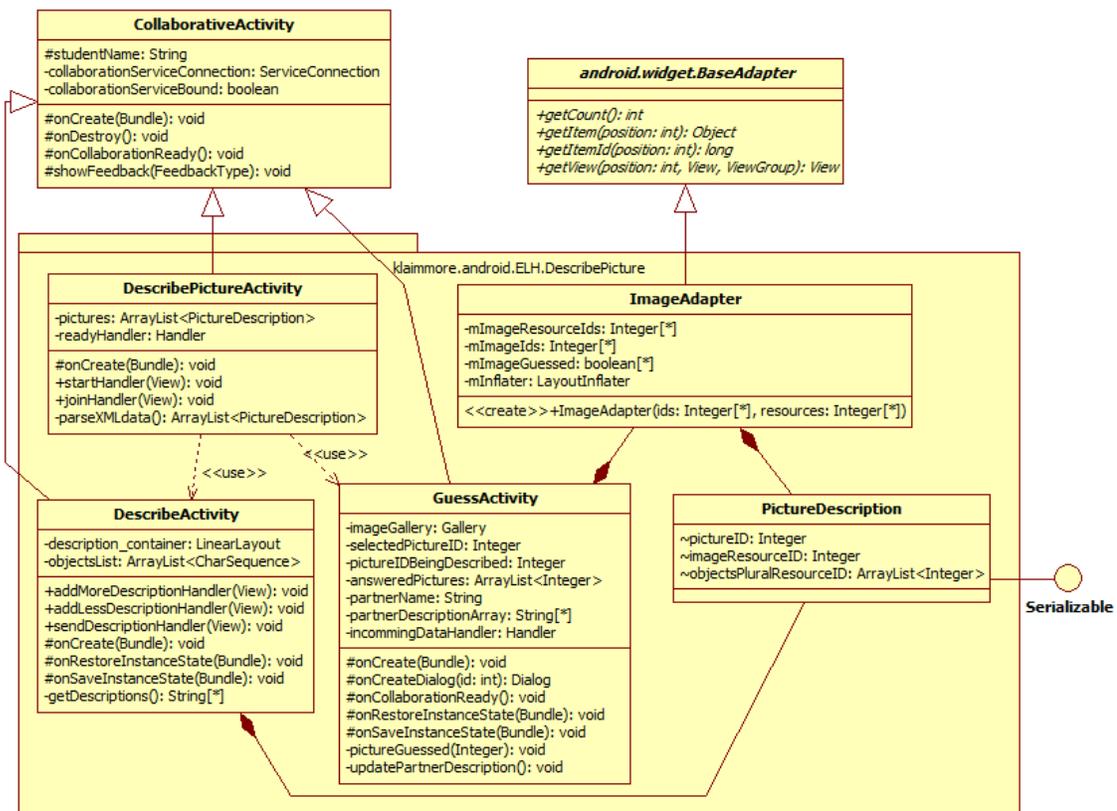


Figura 6: Diagrama de clases de la actividad “Describe the picture”

La actividad “Describe the picture” de la unidad de aprendizaje “School and Classroom” de la figura 6 corresponde a la segunda actividad colaborativa del sistema. Sus componentes son:

- **DescribePictureActivity:** Es la encargada de preparar los parámetros de inicialización, iniciando un servidor y cliente en caso de iniciar la actividad en el mismo dispositivo, o iniciando un cliente en caso de unirse a la actividad de otro dispositivo. Cuando la actividad está lista para ser iniciada, se ejecuta `DescribeActivity` en uno de los dispositivos y `GuessActivity` en el resto.
- **DescribeActivity:** El usuario recibe una imagen para ser descrita de acuerdo a sus objetos y cantidades. Cada descripción realizada por el usuario es enviada a los demás clientes.
- **GuessActivity:** Muestra una serie de imágenes junto a las descripciones realizadas por el compañero que describe. El usuario debe decidir cuál de las imágenes es la descrita.

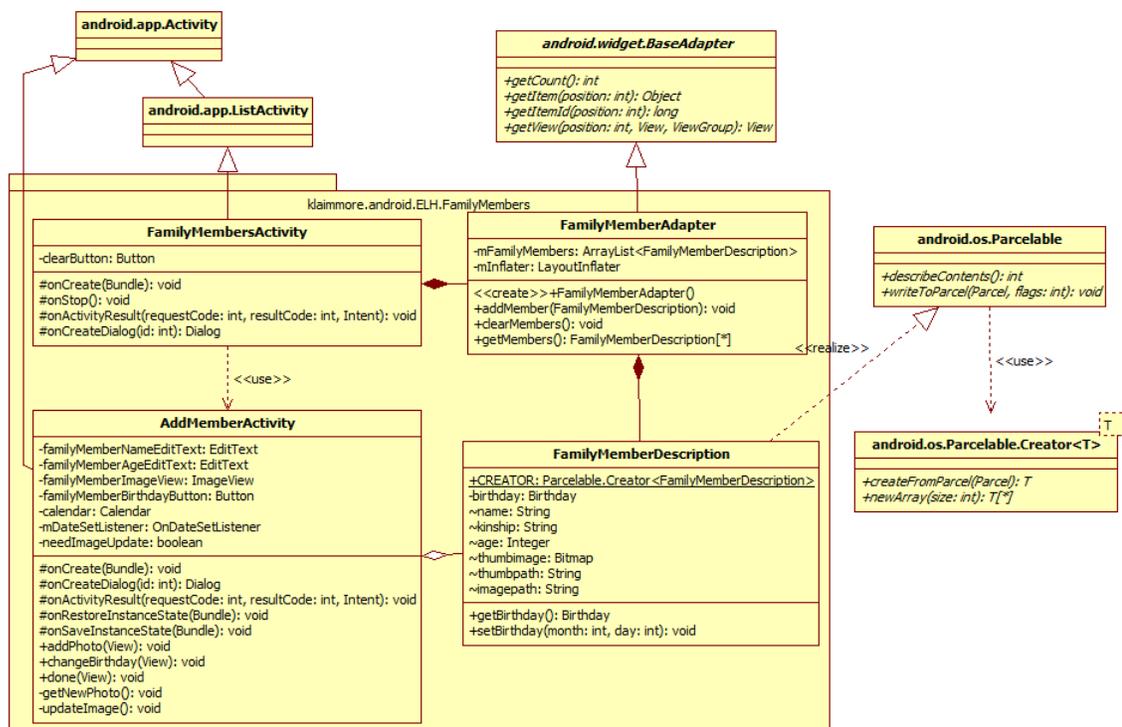


Figura 7: Diagrama de clases de la actividad “Family Members”

La actividad “Family Members” de la unidad de aprendizaje “Home and Family” ilustrada en la figura 7 no es una actividad colaborativa, sino una actividad de discusión e información. Sus componentes son:

- **FamilyMembersActivity:** Muestra la lista de familiares del usuario. En un comienzo esta lista está vacía, el usuario puede agregar familiares invocando a `AddMembersActivity` o limpiar la lista.

- **AddMemberActivity:** El usuario agrega un familiar, utilizando la cámara del dispositivo para tomar una imagen y agregando información relacionada al familiar, como: nombre, edad, parentesco y cumpleaños.

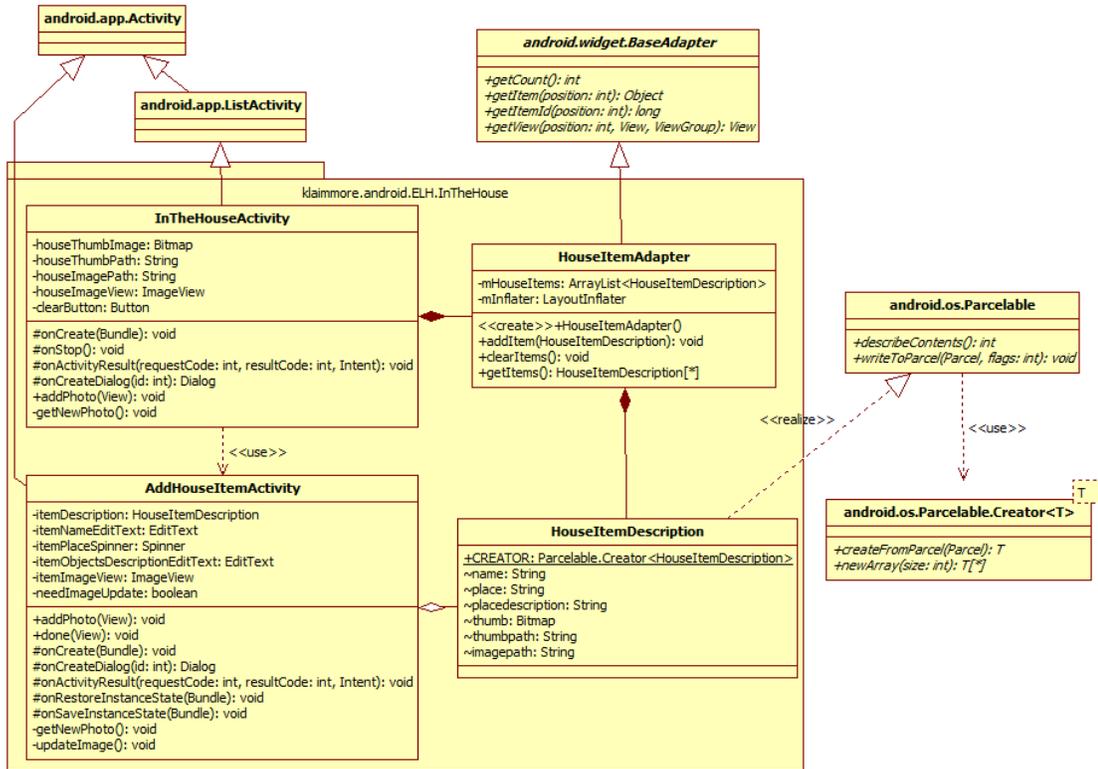


Figura 8: Diagrama de clases de la actividad “In the house”

La actividad “In the house” de la unidad de aprendizaje “Home and Family”, ilustrada en la figura 8 también es una actividad para reunir información y luego discutir con compañeros. Sus componentes son:

- **InTheHouseActivity:** Muestra una lista con los lugares del hogar del usuario, en un comienzo, vacía. El usuario puede agregar lugares invocando a `AddHouseItemActivity` o limpiar la lista.
- **AddHouseItemActivity:** Permite añadir nuevos lugares, agregando una imagen del lugar y descripciones genéricas como nombre, lugar (de una lista de lugares pre-definida) y una descripción en texto libre.

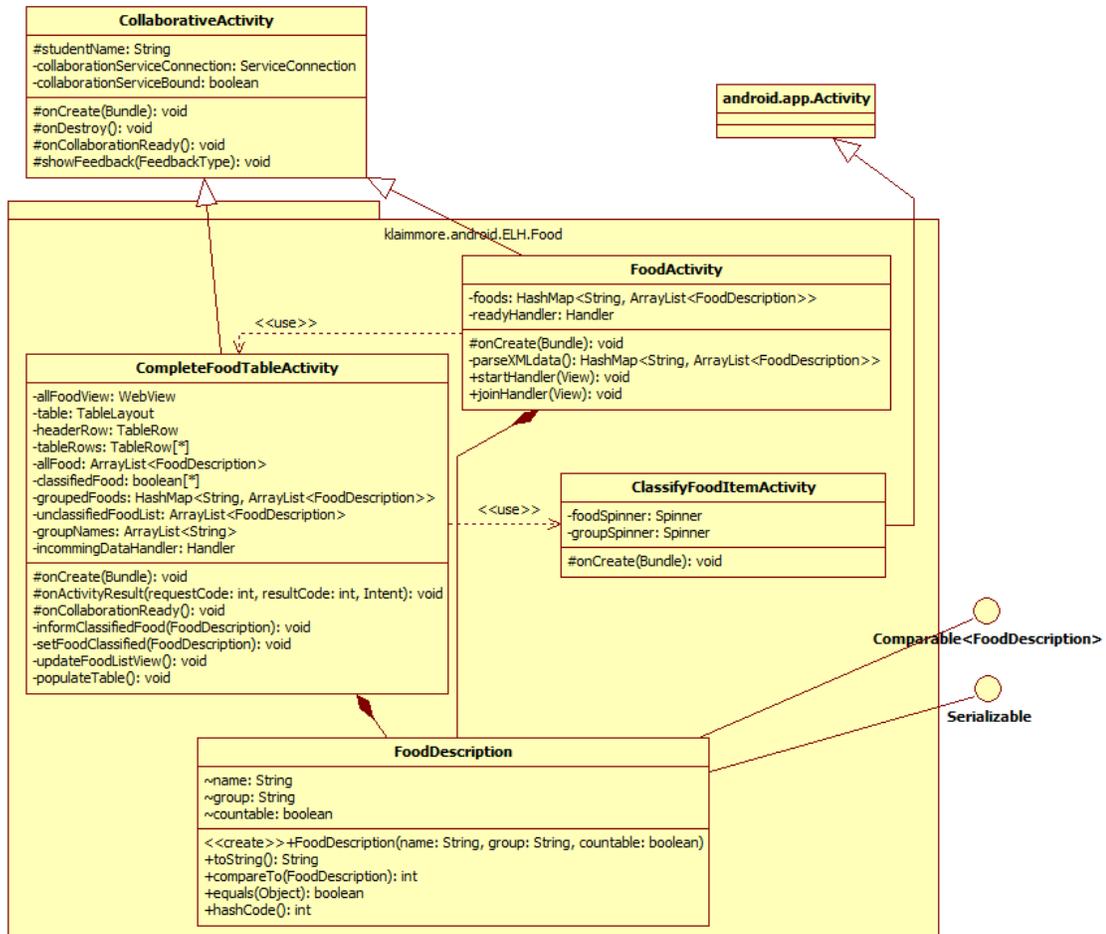


Figura 9: Diagrama de clases de la actividad “Food Groups”

La actividad “Food Groups” de la unidad de aprendizaje “Food and Health” ilustrada en la figura 9 es una actividad colaborativa que permite que los usuarios completen una tabla de alimentos de acuerdo al grupo alimenticio al que pertenecen. Sus componentes principales son:

- **FoodActivity:** Es la encargada de preparar los parámetros de inicialización, iniciando un servidor y cliente en caso de iniciar la actividad en el mismo dispositivo, o iniciando un cliente en caso de unirse a la actividad de otro dispositivo. Cuando la actividad está lista para ser iniciada se invoca a `CompleteFoodTableActivity`.
- **CompleteFoodTableActivity:** Muestra una lista de alimentos a clasificar y una tabla vacía con nombres de grupos alimenticios como encabezado. El usuario puede entonces comenzar a clasificar alimentos, invocando a `ClassifyFoodItemActivity`.

- ClassifyFoodItemActivity:** Cuando el usuario decide clasificar un alimento este componente muestra en pantalla aquellos alimentos aún no clasificados y los grupos alimenticios disponibles. El usuario entonces escoge un alimento y un grupo para agregar a la tabla. Si el alimento es correctamente clasificado el componente retorna satisfactoriamente y se informa a los demás clientes sobre el alimento clasificado. La tabla es entonces completada paso a paso por todos los clientes que participan.

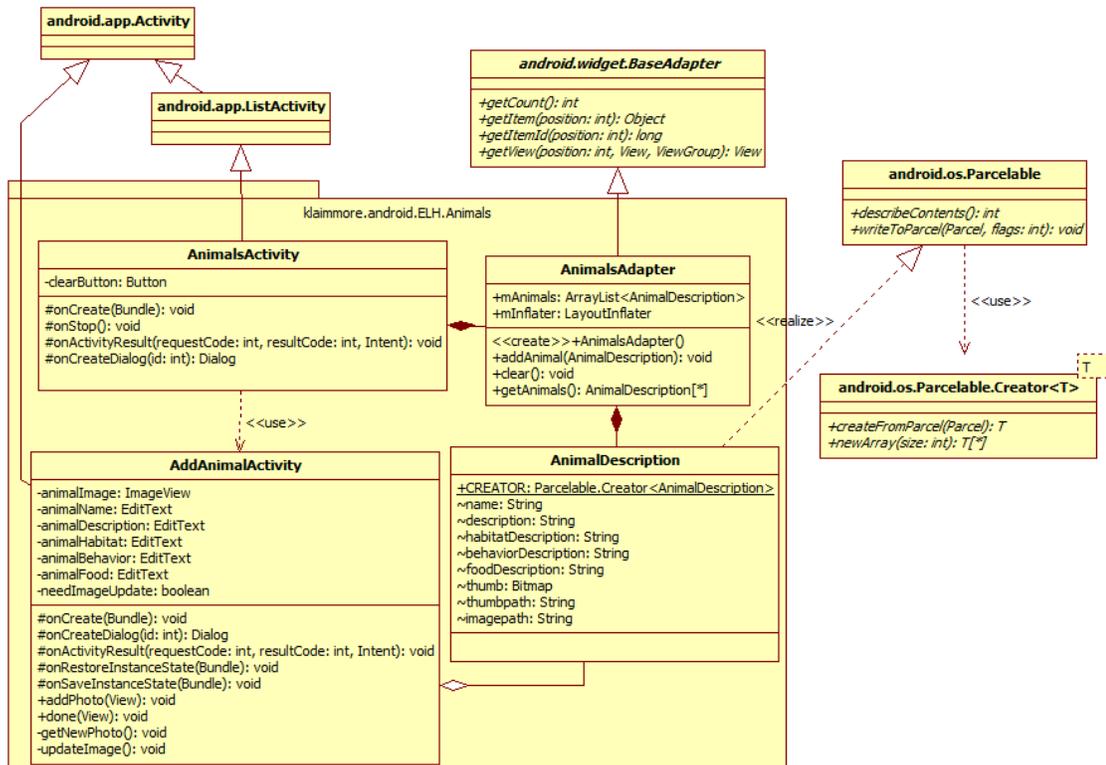


Figura 10: Diagrama de clases de la actividad “Animals”

La actividad “Animals” de la unidad de aprendizaje “The Environment” ilustrada en la figura 10 es una actividad de recolección de información. Como otras actividades de este tipo en el sistema, el usuario es presentado con una lista de animales, en un principio vacía. Sus principales componentes son:

- AnimalsActivity:** Muestra la lista de animales agregados, con su respectiva información asociada. El usuario puede agregar animales invocando a `AddAnimalActivity`.
- AddAnimalActivity:** El usuario puede agregar un animal de interés a la lista. Añadiendo una imagen del animal mediante la cámara del dispositivo, además

de información relacionada como: nombre, descripción física, hábitat, comportamiento y alimentación.

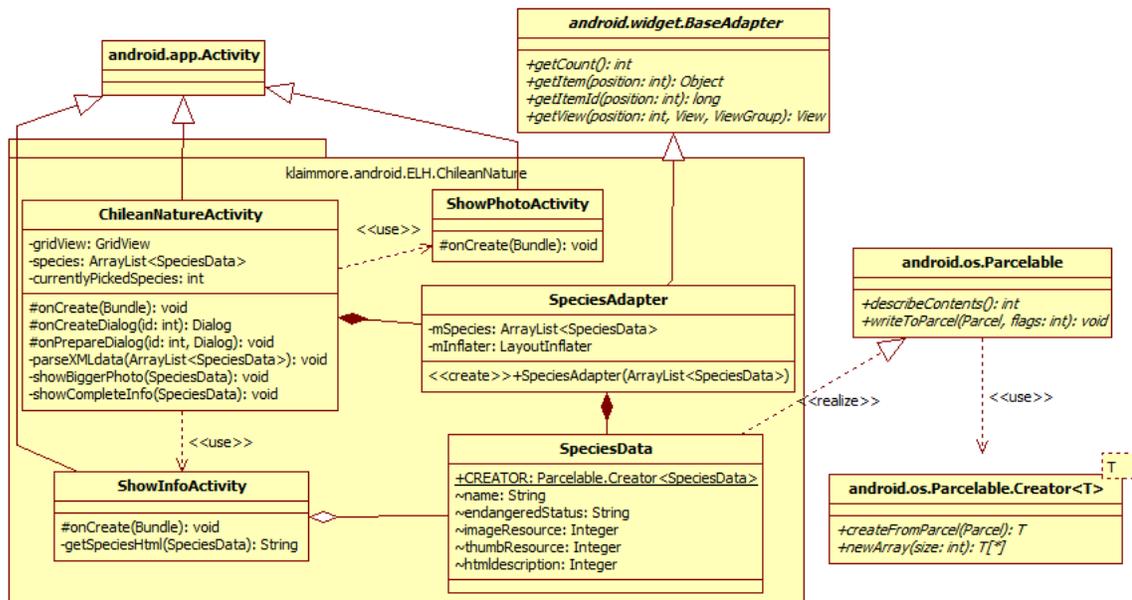


Figura 11: Diagrama de clases de la actividad “Chilean Nature”

La actividad “Chilean Nature” de la unidad de aprendizaje “The Environment” ilustrada en la figura 11 corresponde una actividad de discusión y reflexión. Sus principales componentes son:

- **ChileanNatureActivity:** Muestra una grilla con imágenes de especies chilenas, el usuario puede seleccionar cualquiera de ellas para ver la imagen completa invocando a ShowPhotoActivity o para ver la información de la especie mediante ShowInfoActivity.
- **ShowPhotoActivity:** Muestra la imagen completa de la especie seleccionada.
- **ShowInfoActivity:** Muestra la información de la especie seleccionada. Incluyendo el nombre, el estatus de peligro de extinción y una descripción completa en vista HTML.

4.4. ARCHIVOS Y RECURSOS

El proyecto Android desarrollado, define una estructura de directorios para el código fuente así como para los distintos recursos que utiliza la aplicación, como imágenes, constantes, archivos de *layout*, etc.

La estructura de directorios corresponde a la de la figura 12:

- En el directorio **bin** se generan las clases compiladas del proyecto, además del archivo `.apk` que es el archivo generado por Android para transferir e instalar la aplicación en el dispositivo.
- El directorio **gen** es creado por las herramientas de desarrollo del Android SDK, en él se genera una clase denominada “**R**” que contiene identificadores estáticos para todos los recursos de la aplicación, de manera de poder ser referenciados fácilmente dentro de la aplicación.
- En el directorio **res** se agregan todos los recursos separados por su tipo:
 - **drawable / drawable-Xdpi**: Contiene los recursos gráficos. Pueden ser simples imágenes, archivos *nine-patch* (imágenes PNG con zonas “estirables” para abarcar regiones más grandes), listas de estado (archivos XML que definen gráficos que cambian de estado, por ejemplo botones presionados o no), etc.
El sufijo que acompaña al nombre del directorio (**hdpi**, **ldpi**, **mdpi**) corresponde a la escala de los gráficos, por sus siglas en inglés: *high density per inch*, *low density per inch*, *medium density per inch*. De manera que Android utilice la versión correcta dependiendo de la densidad de la pantalla del dispositivo.
 - **layout**: Contiene archivos XML que definen la forma y los elementos de la interfaz de usuario para una determinada vista de la aplicación.
 - **menu**: Contiene archivos XML que definen los elementos de un determinado menú de la aplicación.
 - **raw**: Contienen todos aquellos recursos que están almacenados en su forma original y no están en el formato de los demás recursos. Por ejemplo: audio, video, archivos de texto plano, HTML, etc.
 - **values**: Contienen archivos XML que definen valores simples, como: Strings, Integers, colores, dimensiones, estilos, etc.
 - **xml**: Archivos XML arbitrarios que siguen algún esquema definido por la aplicación, pueden ser archivos de configuración o meta datos.
- En el directorio **src** se encuentran los archivos de código fuente, con sus respectivos directorios de paquetes Java.
- **AndroidManifest.xml** corresponde a un archivo de configuración requerido por la aplicación y por el sistema operativo Android. Entre otras cosas define:

- El nombre del paquete Java de la aplicación, el cual se convierte en su identificador único.
- Describe los componentes de la aplicación: actividades, servicios, *broadcast receivers*, *content providers*, junto a las clases que implementa cada uno y los filtros que admiten.
- Declara los permisos que son requeridos por la aplicación para utilizar elementos protegidos de la API. Como por ejemplo: Conectarse a internet, utilizar la cámara del dispositivo, etc.
- Declara el nivel mínimo de la API de Android necesario para ejecutar la aplicación.

```

.
|-- bin
|-- gen
|-- res
|   |-- drawable
|   |-- drawable-hdpi
|   |-- drawable-ldpi
|   |-- drawable-mdpi
|   |-- layout
|   |-- menu
|   |-- raw
|   |-- values
|   `-- xml
|-- src
|-- AndroidManifest.xml
`-- default.properties

```

Figura 12: Estructura de directorios de un proyecto típico Android

La figura 13 muestra algunos recursos claves que se utilizaron en las actividades de aprendizaje del proyecto:

- **res/raw:**
 - Para la actividad “Vocabulary” se utilizan archivos de sonido compatibles, que reproducen la pronunciación de las distintas palabras del vocabulario.
 - Para la actividad “Chilean Nature” se utilizan archivos HTML, los cuales proveen una descripción en texto rico de las especies.
- **res/values:**
 - Los distintos archivos XML, contienen la mayor parte de arreglos, Strings y valores utilizados en la interfaz de usuario de toda la aplicación.
- **res/xml:**
 - Los distintos archivos XML, corresponden a información estructurada de las actividades de aprendizaje, el *schema* de cada archivo debe mantenerse, de lo contrario la actividad no podrá cargar esos datos.

```

res
|-- raw
|   |-- araucaria.html
|   |-- condor.html
|   |-- copihue.html
|   |-- darwin_fox.html
|   |-- huemul.html
|   |-- long_tailed_chinchilla.html
|   |-- short_tailed_chinchilla.html
|   |-- vocabulary_copybook_sound.mp3
|   |-- vocabulary_pencil_sound.mp3
|   |-- vocabulary_ruler_sound.mp3
|   `-- vocabulary_whiteboard_sound.mp3
|-- values
|   |-- arrays.xml
|   |-- attrs.xml
|   |-- backgrounds.xml
|   |-- colors.xml
|   |-- dialog_style.xml
|   |-- plural_objects.xml
|   |-- strings.xml
|   |-- styles.xml
|   `-- themes.xml
`-- xml
    |-- chilean_nature_data.xml
    |-- describe_picture_data.xml
    |-- food_data.xml
    |-- sketch_data.xml
    `-- vocabulary_data.xml

```

Figura 13: Recursos claves utilizados en el proyecto

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="species">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="specie" minOccurs="1" maxOccurs="unbounded"
type="Specie"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:complexType name="Specie">
    <xsd:attribute name="name" type="xsd:string"/>
    <xsd:attribute name="endangered" type="StringResource"/>
    <xsd:attribute name="image" type="DrawableResource"/>
    <xsd:attribute name="thumb" type="DrawableResource"/>
    <xsd:attribute name="htmldescription" type="RawResource"/>
  </xsd:complexType>

  <xsd:simpleType name="StringResource">
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="@string/\w* (\w|\d) *"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="DrawableResource">
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="@drawable/\w* (\w|\d) *"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="RawResource">
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="@raw/\w* (\w|\d) *"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:schema>

```

Figura 14: Schema de Chilean Nature

El *schema* utilizado para cargar información en la actividad “Chilean Nature” define como se ingresan las especies y sus atributos.

- El tipo de dato del atributo **name** corresponde a un *String* básico.
- El tipo de dato del atributo **endangered** definido como **StringResource** corresponde a una referencia a los recursos del tipo *String* de Android, definidos en `res/values/strings.xml`. Allí se señalan siete niveles para este atributo: desde 0 - Extinto a 6 - Preocupación Menor.
- El tipo de dato de los atributos **image** y **thumb** definido como **DrawableResource** corresponde a una referencia a un recurso del tipo *drawable* de Android.
- El tipo de dato del atributo **htmldescription** definido como **RawResource** corresponde a una referencia a un recurso *raw* de Android.

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="pictures">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="entry" minOccurs="1" maxOccurs="unbounded"
type="Picture"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:complexType name="Picture">
    <xsd:element name="objects" type="Objects"/>
    <xsd:attribute name="id" type="xsd:positiveInteger"/>
    <xsd:attribute name="image" type="DrawableResource"/>
  </xsd:complexType>

  <xsd:complexType name="Objects">
    <xsd:sequence>
      <xsd:element name="object" minOccurs="1" maxOccurs="unbounded">
        <xsd:complexType>
          <xsd:attribute name="plural" type="PluralsResources"/>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:simpleType name="PluralsResources">
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="@plurals/\w*(\w|\d)*/"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="DrawableResource">
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="@drawable/\w*(\w|\d)*/"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:schema>

```

Figura 15: Schema de Describe the Picture

El *schema* utilizado por la actividad “Describe the Picture” define los atributos de las fotos a ser descritas.

- El tipo de dato utilizado para el atributo **id** corresponde a un entero positivo.
- El tipo de dato utilizado para el atributo **image** definido como **DrawableResource** corresponde a una referencia a un recurso del tipo *drawable* de Android.
- El tipo de dato para el elemento **object** corresponde a una referencia a un recurso del tipo *plural* de Android, definidos en `res/values/plural_objects.xml`.

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="foods">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="group" minOccurs="1" maxOccurs="unbounded"
type="Group">
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>

<xsd:complexType name="Group">
  <xsd:sequence>
    <xsd:element name="food" minOccurs="1" maxOccurs="unbounded" type="Food"/>
  </xsd:sequence>
  <xsd:attribute name="name" type="xsd:string"/>
</xsd:complexType>

<xsd:complexType name="Food">
  <xsd:attribute name="name" type="xsd:string"/>
  <xsd:attribute name="countable" type="Boolean"/>
</xsd:complexType>

<xsd:simpleType name="Boolean">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="(true|false)"/>
  </xsd:restriction>
</xsd:simpleType>
</xsd:schema>

```

Figura 16: Schema de Food Groups

El *schema* utilizado por la actividad “Food Groups” define los atributos de los alimentos utilizados.

- El tipo de dato para el elemento **group** definido como **Group** corresponde a un grupo alimenticio y posee como atributo un *String* básico denominado **name** y como elemento interno una secuencia de **food**.
- El tipo de dato para el elemento **food** definido como **Food** corresponde a un alimento y posee como atributos **name** un *String* básico y **countable** un *String* ‘true’ o ‘false’.

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="sketch">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="entry" minOccurs="1" maxOccurs="unbounded"
type="Sketch"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:complexType name="Sketch">
    <xsd:sequence>
      <xsd:element name="objects" type="Objects"/>
      <xsd:element name="colors" type="Colors"/>
    </xsd:sequence>
    <xsd:attribute name="name" type="xsd:string"/>
    <xsd:attribute name="image" type="DrawableResource"/>
  </xsd:complexType>

  <xsd:complexType name="Objects">
    <xsd:sequence>
      <xsd:element name="object" minOccurs="1" maxOccurs="unbounded"
type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="Colors">
    <xsd:sequence>
      <xsd:element name="color" minOccurs="1" maxOccurs="unbounded"
type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:simpleType name="DrawableResource">
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="@drawable/\w* (\w|\d) *"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:schema>

```

Figura 17: Schema de Complete the Sketch

El *schema* utilizado por la actividad “Complete the Sketch” define los atributos de los distintos *sketchs*.

- El tipo de dato para el elemento **entry** definido como **Sketch**, posee como atributos **name** un *String* básico e **image** una referencia a un recurso *drawable* de Android. Además posee los elementos **objects** y **colors**.
- El tipo de dato para el elemento **objects** corresponde a una lista de objetos, siendo cada **object** un *String* básico.
- El tipo de dato para el elemento **colors** corresponde a una lista de colores, siendo cada **color** un *String* básico.

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="vocabulary">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="entry" minOccurs="1" maxOccurs="unbounded" type="Word"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:complexType name="Word">
    <xsd:attribute name="name" type="xsd:string"/>
    <xsd:attribute name="image" type="DrawableResource"/>
    <xsd:attribute name="sound" type="RawResource"/>
  </xsd:complexType>

  <xsd:simpleType name="DrawableResource">
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="@drawable/\w* (\w|\d) *"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="RawResource">
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="@raw/\w* (\w|\d) *"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:schema>

```

Figura 18: Schema de Vocabulary

El *schema* utilizado por la actividad “Vocabulary” define un conjunto de palabras con atributos adicionales.

- El tipo de dato para **entry** definido como **Word** posee como atributos:
 - **name** un *String* básico.
 - **image** una referencia a un recurso del tipo *drawable* de Android.
 - **sound** una referencia a un recurso del tipo *raw* de Android.

4.5. EVALUACIÓN PRELIMINAR

Se realizó una evaluación preliminar del sistema con dos alumnos de 2º año de enseñanza media. Las respuestas de estos alumnos al hacer uso del sistema son como siguen:

- Complete the Sketch:
 - **Estudiante 1:** Me agrada porque fomenta el trabajo en equipo.
 - **Estudiante 2:** Me gustaría que el compartir instrucciones fuera justo, los 2 comparten al mismo tiempo o ninguno.
Es necesario saber que hacer antes de comenzar, apretar el botón de menú es clave.
- Describe the Picture:
 - **Estudiante 1:** Falta un menú de instrucciones.
 - **Estudiante 2:** Cuando a uno no le toca describir la imagen, se tiene que esperar mucho rato para que el compañero haga la descripción.
- Family Members:
 - **Estudiante 1:** Es bueno porque fomenta el dialogo, se hace mejor hablar de la familia con la información y la foto a la mano.
 - **Estudiante 2:** Es divertido tomar fotos a mi familia y llenar información, podría también incluir apodos.
- In the House:
 - **Estudiante 1:** Sirve mucho para aprenderse los nombres de las habitaciones en inglés, con las imágenes es más fácil aprender.
 - **Estudiante 2:** Tiene más información para llenar y se puede poner lo que uno quiera en la descripción del lugar.
- Food Groups:
 - **Estudiante 1:** No hay menú de instrucciones.
 - **Estudiante 2:** Es bueno que me digan si está correcta o no mi clasificación.
Al final se podría mostrar quien de los compañeros hizo más clasificaciones correctas.
- Animals:
 - **Estudiante 1:** Es bueno porque hay bastante información para llenar en los animales. Al usar tantas palabras para llenar la información se aprende más.
 - **Estudiante 2:** Me parece entretenido poder llenar información sobre los animales que a mí me gustan.
- Chilean Nature:
 - **Estudiante 1:** Está bueno para crear conciencia sobre las especies y valorarlas. Es fácil de usar y bastante directa. Uno aprende inglés leyendo acerca de las especies.

- **Estudiante 2:** Hay muchas cosas que uno no sabe siendo chilena, está muy bueno conocerlas.

Si bien esta evaluación preliminar es en general positiva, muestra que existen muchas deficiencias en cuanto a la interfaz de usuario, como carencia de un menú de instrucciones apropiado.

5. CONCLUSIONES

5.1. DEL TRANSCURSO DEL PROYECTO

Durante el transcurso del proyecto, primeramente fue necesaria la investigación y adquisición de conocimiento respecto a la enseñanza del inglés en Chile, así como de los proyectos sobre el uso de software en contextos de enseñanza, en particular de un idioma extranjero. Las tendencias en la enseñanza enfatizaban el uso de ambientes de aprendizaje donde se utiliza ampliamente la comunicación, cooperación e interacción. La tecnología asociada para apoyar el aprendizaje varía desde ambientes completamente ubicuos, a otros más simples utilizando los dispositivos móviles comunes de los usuarios.

La propuesta de solución considerada fue convergiendo rápidamente a una solución móvil, por la familiaridad de las personas, en particular los jóvenes, a los dispositivos celulares y por las crecientes prestaciones de estos dispositivos culminando en smartphones con amplia capacidad para el desarrollo de aplicaciones.

De las distintas plataformas de desarrollo consideradas para el proyecto, la que resaltó por sus prestaciones, comunidad de desarrollo y penetración de mercado fue la plataforma Android. Siendo necesario dedicar mucho tiempo para la ambientación e investigación en el desarrollo bajo la plataforma.

Durante el tiempo de desarrollo e implementación del software, se probaron los distintos aspectos del *framework* de Android en mini-proyectos orientados a una solución minimal del problema con el fin de disminuir el riesgo y realizar pruebas de concepto. Esto permitió, entre otras cosas, comprender la necesidad de implementar un *Service* de Android para el manejo de la comunicación entre dispositivos.

Ya más avanzado el proyecto, se detectó que ciertos componentes de la aplicación se hacían más lentos a medida que el usuario ingresaba más datos, esto demandó que se buscaran las causas y posibles soluciones. Se investigó sobre cómo hacer la interfaz de usuario más eficiente y con menos retardos, culminando en la implementación de las prácticas recomendadas por Google para el manejo de *Views* y *Adapters*, así como la utilización de *thumbnails* y *Bitmaps* pre-cargados para los gráficos que no se necesitaban en tamaño real.

5.2. OBJETIVOS ALCANZADOS

De los objetivos alcanzados en el desarrollo del proyecto, se encuentra el haber desarrollado las actividades propuestas por el sistema. Haciendo uso de actividades que podrían promover la interacción y cooperación de los alumnos.

Queda aún por determinar el impacto de la implementación de este sistema en un ambiente real de aprendizaje, como los establecimientos de educación básica chilenos.

5.3. TRABAJO FUTURO

Aún hay mucho por mejorar en sistemas de este tipo. En particular, sería bueno explorar las capacidades de reconocimiento de voz que ofrece Android e idear actividades que pudieran sacar provecho de esto (Por lo pronto, Google ofrece el servicio de *Voice Search* en EEUU, Francia, Alemania, Italia y España).

6. ANEXOS

6.1. PLATAFORMA ANDROID

La plataforma de desarrollo utilizada en esta memoria corresponde a la plataforma de Android. En el presente capítulo se explica en qué consiste, sus principales componentes y los fundamentos para el desarrollo de un proyecto Android.

6.1.1. ANDROID SOFTWARE DEVELOPMENT KIT

El Android SDK provee las herramientas de desarrollo, el `application framework` y API para desarrollar aplicaciones móviles bajo la plataforma Android. Entre sus componentes se encuentran:

- **SDK tools:** un conjunto de herramientas de desarrollo como el emulador de Android, el `Android Development Tools plugin para Eclipse`[19], el `Android Virtual Devices` que permite administrar los dispositivos del emulador, el `Android Debug Bridge` que permite acceder al emulador por línea de comandos y provee mecanismos de depuración, etc.
- **Android Platforms:** el conjunto completo de librerías e imágenes de sistema necesarios para compilar y ejecutar las aplicaciones Android.
- **USB driver para Microsoft Windows:** usado para ejecutar y depurar aplicaciones en un dispositivo real de Android (en vez del emulador)
- **Ejemplos:** algunos ejemplos que demuestran el uso de la plataforma.
- **Documentación:** copia local de la documentación del *framework*.

6.1.2. ANDROID OS

La arquitectura del sistema Android cuenta con los siguientes componentes principales:

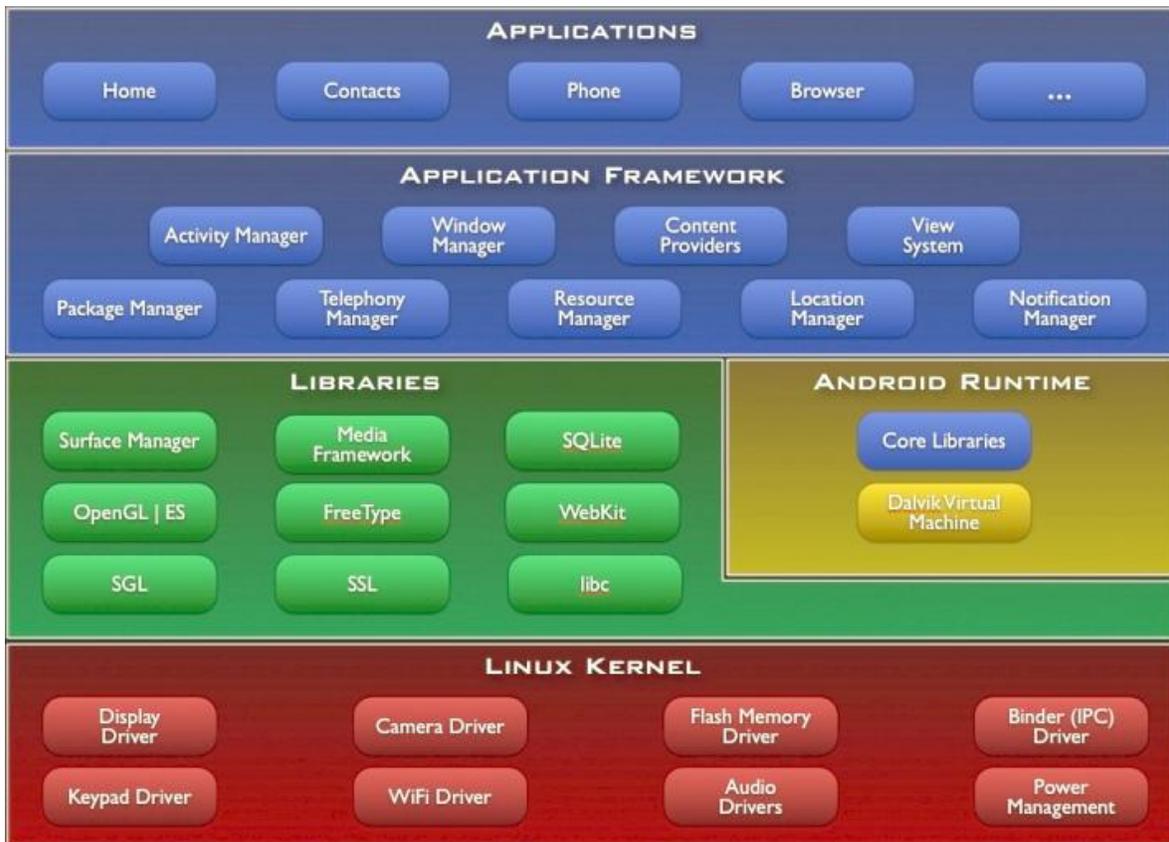


Figura 19: Diagrama de la arquitectura del sistema operativo Android

- **Aplicaciones**

El conjunto de aplicaciones por defecto que provee el sistema puede ser usado por las demás aplicaciones desarrolladas bajo el *framework*.

- **Application Framework**

Mediante el *framework* se exponen distintos servicios y componentes del sistema o el hardware, incluyendo acceso a la información de geolocalización, la ejecución de servicios en segundo plano, alarmas y notificaciones de la barra de estado.

La arquitectura de aplicaciones simplifica la reutilización de componentes, cualquier aplicación desarrollada puede publicar sus capacidades para que otras aplicaciones las utilicen.

Algunos servicios y sistemas expuestos para las aplicaciones son:

- **Sistema de vistas:** un conjunto de *widgets* para la interfaz de usuario.
- **Content Providers:** permiten el acceso de datos desde otras aplicaciones (como la lista de contactos), así como compartir datos propios.

- **Resource Manager:** provee acceso a los diferentes recursos que componen las aplicaciones, como gráficos, especificaciones de *layout*, constantes, etc.
- **Notification Manager:** permite a las aplicaciones el uso de notificaciones en la barra de estado del sistema.
- **Activity Manager:** maneja el ciclo de vida de las aplicaciones y provee la funcionalidad de la pila de navegación.
- **Bibliotecas**
 Conjunto de bibliotecas utilizadas por los componentes del sistema y expuestas a través del *framework* para los desarrolladores. Tales como:
 - **System C library:** Una implementación derivada de BSD[20] de la biblioteca de sistema C estándar (*libc*), especialmente modificada para sistemas embebidos.
 - **Media Libraries:** Soportan reproducción y grabación de diferentes formatos de audio, video e imágenes. Como MPEG4[21], H.264[22], MP3[23], AAC[21], AMR[24], JPG[25] y PNG[26].
 - **Surface Manager:** Gestiona el acceso al subsistema de visualización y las capas de gráficos 2D y 3D.
 - **LibWebCore:** Motor web utilizado en el browser de Android así como para visualización web embebida en vistas.
 - **SGL:** Motor de gráficos 2D.[27]
 - **3D Libraries:** Implementación basada en OpenGL ES 1.0[28]. Utiliza aceleración 3D por hardware si está disponible o un software incluido de rasterización 3D altamente optimizado.
 - **FreeType:** Renderizado de fuentes de mapas de bits o vectoriales.
 - **SQLite:** Motor de base de datos relacional de bajo peso y alto rendimiento.[29]
- **Android Runtime**
 Incluye un set de bibliotecas que proveen la mayoría de la funcionalidad del núcleo de bibliotecas disponibles en Java.
 Cada aplicación en Android se ejecuta en su propio proceso, en su propia instancia de la máquina virtual Dalvik[30]. Dalvik fue diseñada para que el dispositivo pueda correr varias máquinas virtuales eficientemente. La máquina virtual Dalvik a diferencia de la estándar de Java, está basada en registros y está diseñada para sistemas con recursos reducidos de memoria y procesamiento.
- **Linux Kernel**
 Android se basa en la versión 2.6 de Linux[31] para los servicios de sistema como seguridad, gestión de memoria, gestión de procesos, pila de red y modelo de *drivers*. El *kernel* también es una capa de abstracción entre el hardware y el resto de la pila de software.

6.1.3. ANDROID FRAMEWORK

6.1.3.1. Componentes de aplicación

Una de las características de Android es que una aplicación puede usar elementos de otras aplicaciones que lo permiten. Por ejemplo, si una aplicación requiere mostrar una lista de imágenes y otra aplicación posee dicha funcionalidad disponible para otras aplicaciones, entonces la primera puede invocar la funcionalidad de la segunda sin necesidad de importar librerías o conocer los detalles.

Para esto, el sistema puede iniciar el proceso de la aplicación cualquiera sea la parte que se necesita e instanciar los objetos requeridos para esa parte. Una aplicación Android entonces no tiene un solo punto de entrada y posee en cambio **componentes** esenciales que el sistema puede ejecutar cuando los necesite:

- **Activities**

Una actividad deriva de la clase `android.app.Activity` y representa el componente visible que interactúa con el usuario para una labor específica. Por ejemplo, una aplicación de mensajería podría tener una actividad para mostrar los remitentes, otra para escribir un mensaje al remitente seleccionado y otra para ver el listado de mensajes.

Una aplicación podría entonces componerse de un grupo de actividades que se invocan unas a otras. Aunque típicamente una de ellas corresponde a la actividad inicial.

El contenido visual de las actividades está dado por una jerarquía de vistas, las cuales derivan de la clase `android.view.View` y típicamente los nodos finales de la jerarquía corresponden a *widgets* mientras que los superiores son contenedores y *layouts*.

- **Services**

Un servicio deriva de la clase `android.app.Service`, no posee una interfaz de usuario y se ejecuta en segundo plano por un tiempo indefinido. Un ejemplo corresponde a un servicio de reproducción de música, el usuario puede seguir escuchando la música aún cuando la interfaz del reproductor no está en primer plano.

Los servicios se ejecutan en el hilo principal de la aplicación, para no interrumpir otros componentes usualmente invocan nuevos hilos cuando ejecutan tareas que consumen mucho tiempo.

- **Broadcast receivers**

Es un componente cuya única función es recibir anuncios globales. Algunos de ellos se originan por el sistema, como por ejemplo, cambios en la zona horaria o batería baja. Las aplicaciones también pueden iniciar *broadcasts* para dar aviso

a otras de cambios ocurridos.

Los *broadcast receivers* derivan de la clase

`android.content.BroadcastReceiver` y no poseen una interfaz de usuario, aunque pueden iniciar actividades en respuesta a los cambios que reciben.

- **Content providers**

Un *content provider* deriva de la clase `android.content.ContentProvider` y su rol es hacer pública a otras aplicaciones una porción específica de datos.

Las aplicaciones utilizan un `android.content.ContentResolver` para invocar los métodos expuestos por el *content provider*. Los *content providers* corresponden a la única forma de compartir datos entre aplicaciones.

6.1.3.2. Activación de componentes

Los *content providers* son activados cuando reciben un requerimiento por el *content resolver*, mientras que el resto de los componentes son activados por medio de *intents*.

6.1.3.2.1. Intents

Un *intent* es un objeto de la clase `android.content.Intent`, este actúa como un mensaje asíncrono y contiene los datos necesarios para activar un componente.

Principalmente contiene:

- **Component Name:** El nombre del componente que debería responder al *intent* (Clase y *package*). Si este campo no está establecido entonces Android utiliza la demás información del *intent* para encontrar un componente apropiado para manejarlo.
- **Action:** Un `String` que nombra la acción que debe realizarse o, en el caso de un *broadcast*, la acción que ocurrió y está siendo reportada.
- **Data:** El URI de los datos y el tipo MIME de ellos.
- **Category:** Un `String` representando información adicional acerca del tipo de componente que debería responder al *intent*.
- **Extras:** Mapeo de objetos adicionales que utilizará el componente para responder al *intent*.
- **Flags:** Diferentes *flags* para controlar como es manejado el *intent*. En particular pueden cambiar como se maneja la pila de la historia de actividades.

Si el *intent* posee un *component name* se le llama “explícito”, y el sistema instanciará la clase específica para responder al *intent*.

Por el contrario, si el *intent* es “implícito” el sistema busca un componente que pueda responder al *intent* mediante el listado de *intent filters* de los componentes disponibles en el sistema.

6.1.3.2.2. *Intent filters*

Los *intent filters* describen las capacidades de un componente, definiendo que tipo de *intents* puede recibir. Corresponden a anotaciones XML en el archivo `manifest.xml` de la aplicación que contiene al componente.

Cada *intent filter* puede filtrar simultáneamente tres aspectos de un *intent*:

- **Action:** El campo *action* especificado en el *intent* debe ser igual a uno de los elementos *action* especificados en el filtro. Debe haber al menos un elemento *action* especificado en el filtro, de lo contrario sólo se aceptarán *intents* “explícitos”.
Si el *intent* no posee el campo *action* entonces pasará el filtro.
- **Category:** Cada *category* definida en el *intent* debe estar listada en el filtro. Los *intents* “implícitos” siempre poseen la *category* `android.intent.category.DEFAULT` por lo que el filtro debería incluir este elemento para aceptar *intent* “implícitos”.
- **Data:** El elemento *data* puede especificar una URI y un tipo MIME. Una URI en el filtro puede estar compuesta de *scheme*, *authority* (host y puerto) y *path*. Un *intent* pasará el filtro si:
 - El *intent* no posee URI ni tipo MIME y el filtro tampoco.
 - El *intent* contiene URI pero no tipo MIME y el filtro contiene la misma URI y tampoco posee tipo MIME.
 - El *intent* contiene tipo MIME pero no URI y el filtro contiene el mismo tipo MIME y tampoco posee URI.
 - El *intent* contiene tanto URI como tipo MIME, el filtro contiene el mismo tipo MIME y además la URI especificada en el filtro es la misma o el *intent* posee URI del tipo `content:` o `file:` y el filtro no especifica ninguna.

Un *intent* que pase los tres tipos de filtros especificados podrá entonces ser atendido por el componente. Si hay más de un componente que puede atender el *intent* el usuario podría ser consultado sobre que componente ejecutar.

6.1.3.3. Ciclo de vida de componentes

Los distintos componentes poseen un ciclo de vida que comienza al ser instanciados por el sistema para responder a *intents* y termina cuando son destruidos.

6.1.3.3.1. *Activity*

Una actividad puede tener tres estados:

- **Activa** o en ejecución si está en el frente de la pantalla (en el tope de la pila de actividades). Esta será la actividad que tiene el foco de la interacción con el usuario.
- **Pausada** si no posee el foco pero aún esta visible para el usuario. Esto es cuando otra actividad está en frente de la pantalla pero es semi-transparente o no llena todo el espacio de la actividad pausada. El sistema puede terminar la actividad en condiciones extremas de memoria.
- **Detenida** si está completamente oculta al usuario. El sistema usualmente la terminará si necesita reutilizar la memoria.

Cuando el sistema desea reutilizar la memoria podrá invocar el método `finish()` de la actividad o simplemente terminar el proceso. Cuando el usuario la solicita nuevamente ésta debe ser restaurada a su estado previo.

Los cambios de estado de la actividad son notificados mediante funciones protegidas:

- El ciclo de vida completo ocurre entre la primera llamada a la función `onCreate()` y la única llamada a `onDestroy()`.
- El ciclo de vida visible ocurre entre una llamada a `onStart()` y la siguiente llamada a `onStop()`.
- El ciclo de vida frente a la pantalla ocurre entre la llamada a `onResume()` y la llamada a `onPause()`.

El diagrama con el ciclo de vida completo se encuentra en la figura 20.

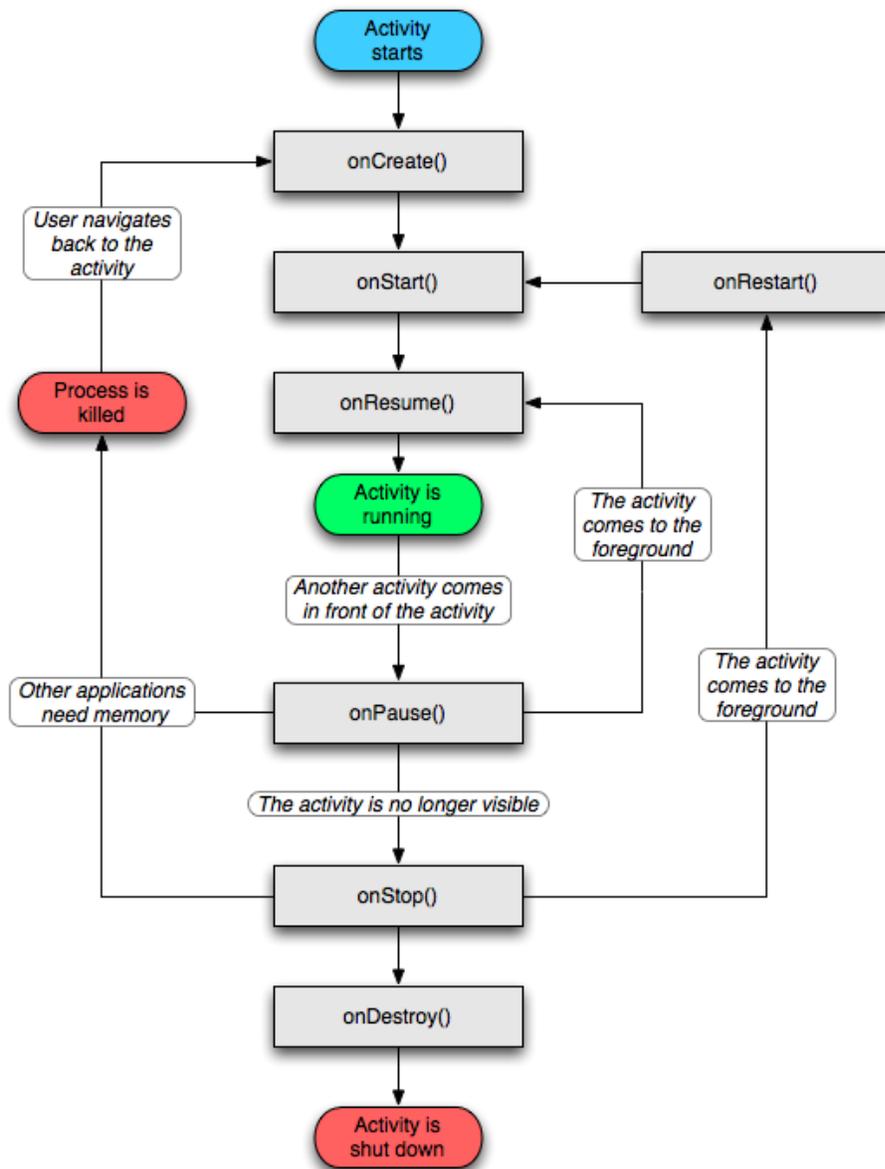


Figura 20: Ciclo de vida de un objeto `android.app.Activity`

6.1.3.3.2. Service

Un servicio puede ser usado utilizando la clase `android.content.Context` de dos formas:

- Puede ser iniciado y continuar activo hasta que alguien lo detenga. Mediante `Context.startService()` y `Context.stopService()`
- Puede ser operado de manera programática mediante una interfaz definida por el servicio en particular. Los clientes establecen una conexión mediante `Context.bindService()` y la cierran con `Context.unbindService()`

Como una actividad, un servicio posee funciones para notificar cambios de estado:

- El ciclo completo comienza con `onCreate()` y termina en `onDestroy()`
- Si el servicio se inició con `Context.startService(Intent)`, entonces el ciclo activo comienza con `onStartCommand(Intent, ...)` donde recibe el *intent* como parámetro. No existe un método para notificar que terminó el ciclo activo.
- Si el servicio se inició con `Context.bindService(Intent, ...)` entonces `onBind(Intent)` es ejecutado y retorna el canal de comunicación. Cuando todos los clientes se han desconectado del canal de comunicación el servicio llama a `onUnbind(Intent)`.

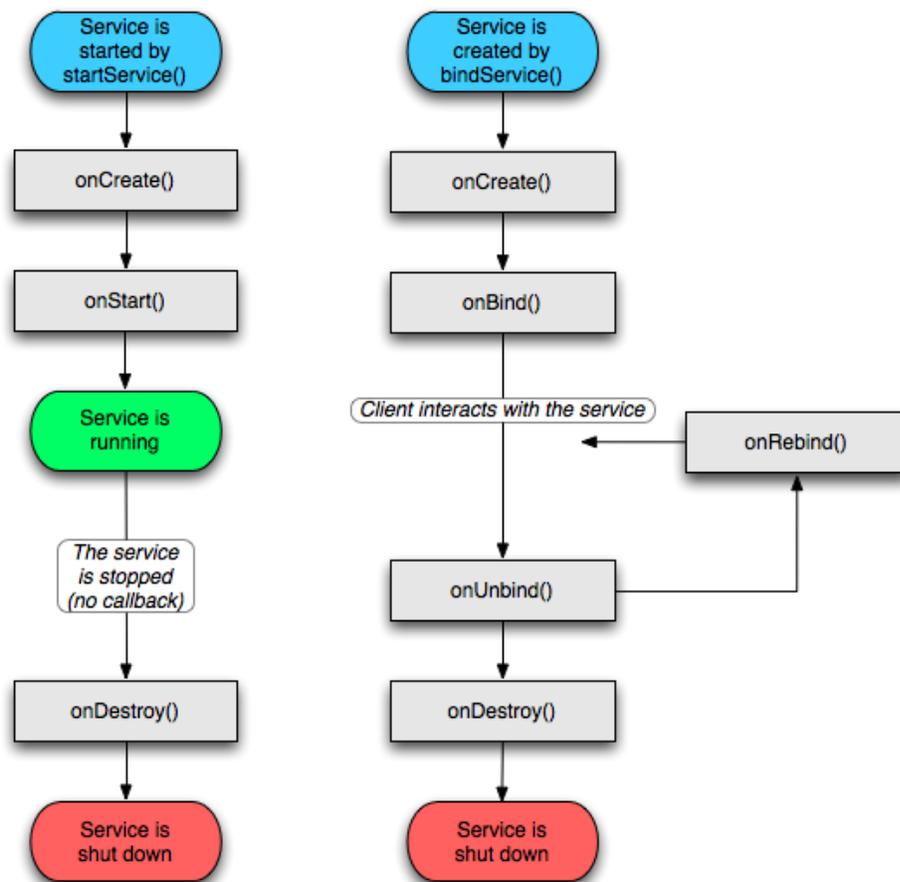


Figura 21: Ciclo de vida de un objeto `android.app.Service`

6.2. CAPTURAS DE PANTALLA DE LA APLICACIÓN

A continuación se muestran algunas capturas de pantalla haciendo uso de las distintas actividades de la aplicación.

6.2.1. PANTALLA DE INICIO

- En la figura 22 se muestra la lista de actividades disponibles.
- La figura 23 muestra la actividad de vocabulario que permite asociar palabra, audio e imagen.
- La figura 24 muestra la pantalla de ajustes.



Figura 22: Pantalla de inicio



Figura 23: Actividad de vocabulario

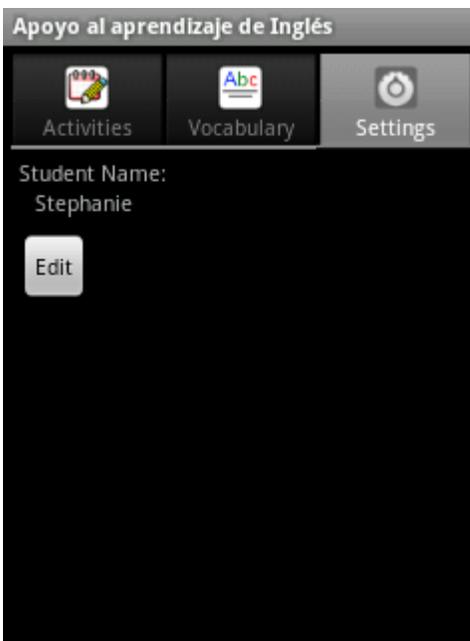
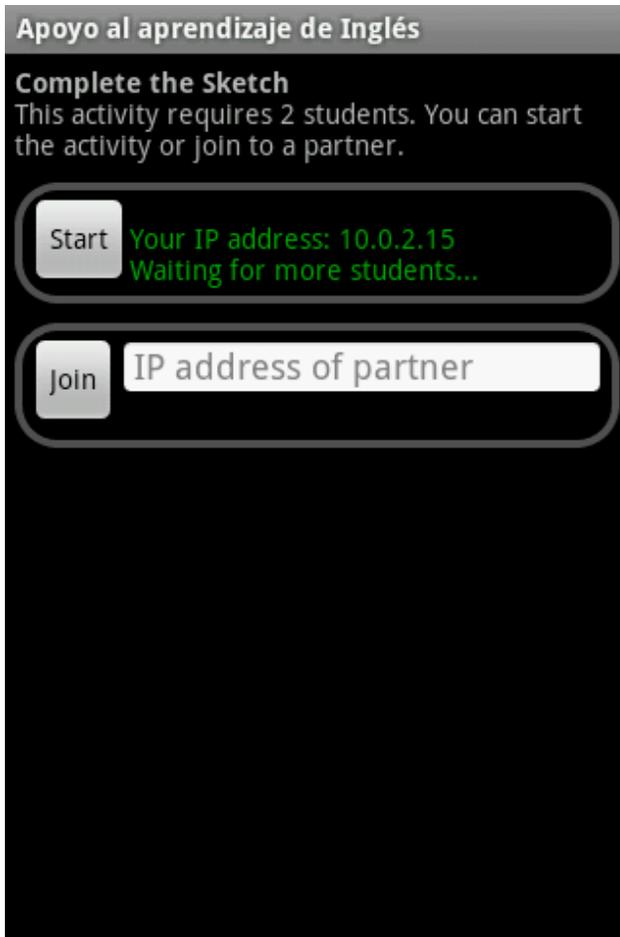


Figura 24: Ajustes de personalización

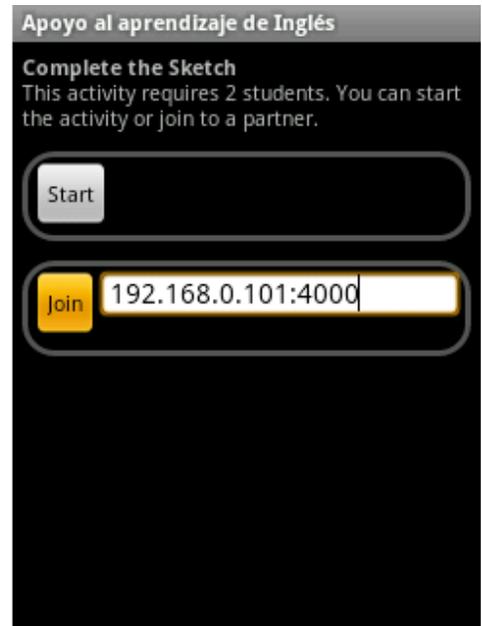
6.2.2. ACTIVIDAD "COMPLETE THE SKETCH"

- La figura 25 muestra la pantalla de conexión que se muestra antes de una actividad colaborativa. En la foto se utiliza una dirección IP y puerto distinto al sugerido por la aplicación debido al uso del emulador y algunos proxies en el ambiente de desarrollo.
- La figura 26 muestra el menú de opciones disponible.
- La figura 27 muestra las instrucciones de los *sketchs* de cada alumno.

- La figura 28 muestra el proceso de pintado de los *sketchs*.
- La figura 29 y 30 muestra como se comparten instrucciones.
- La figura 31 muestra la visualización de las nuevas instrucciones.
- La figura 32 muestra el *sketch* terminado.



A



B

Figura 25: Inicio de la actividad "Complete the Sketch"

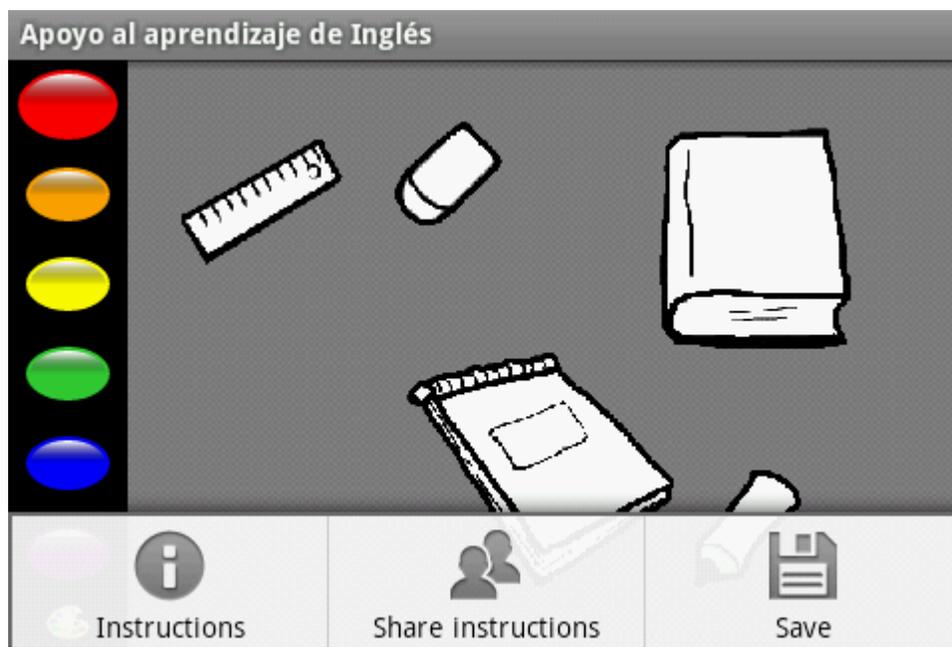
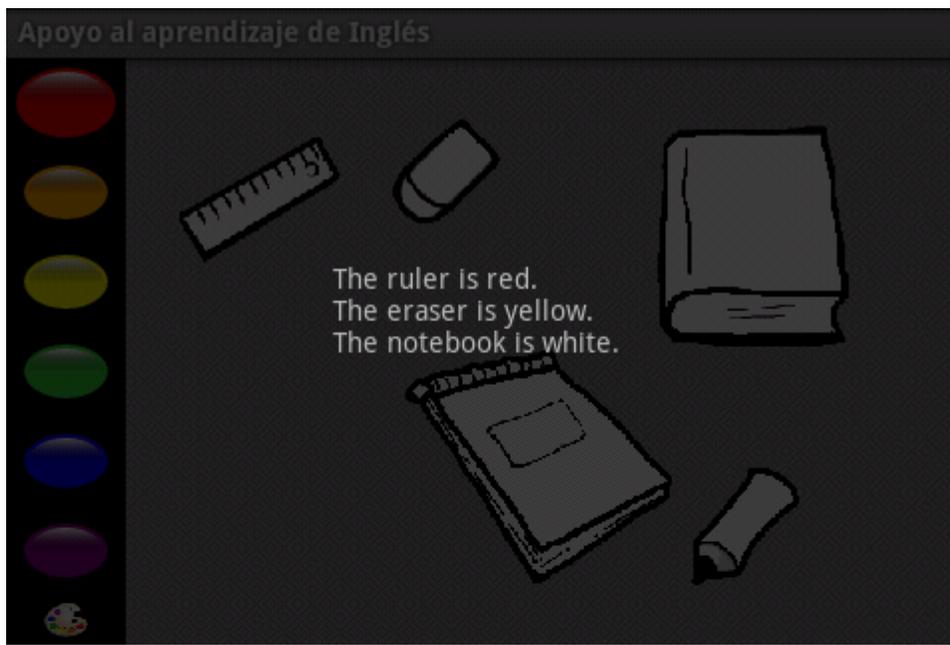


Figura 26: Menú de “Complete the Sketch”



A



B

Figura 27: Instrucciones de los *sketchs*

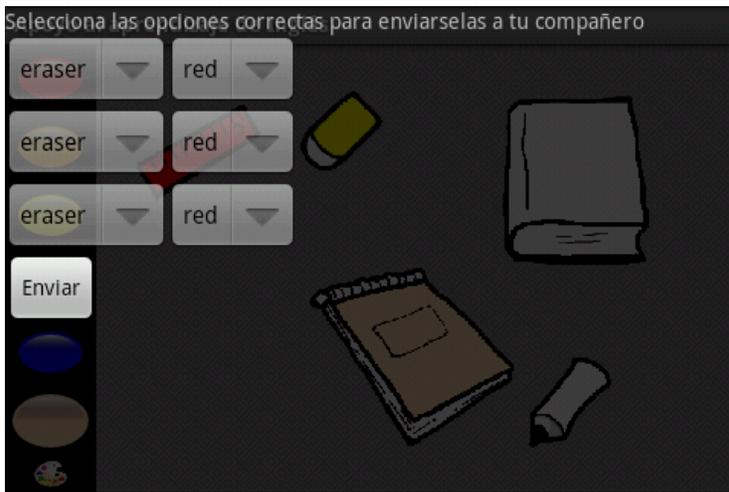


A

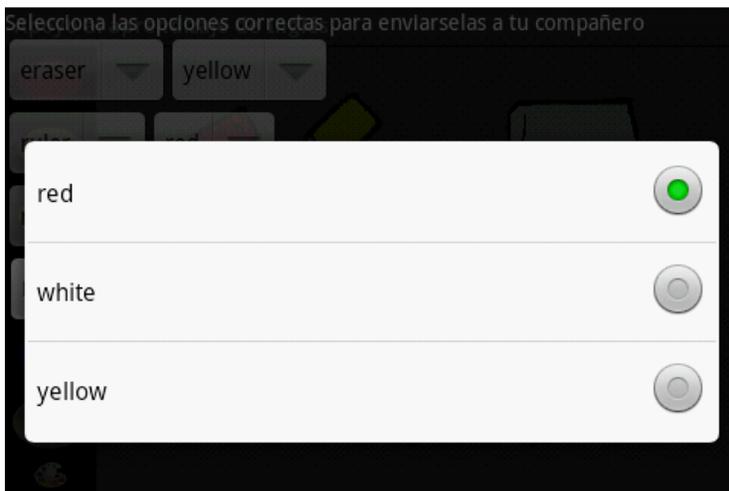


B

Figura 28: Pintado de los *sketchs*

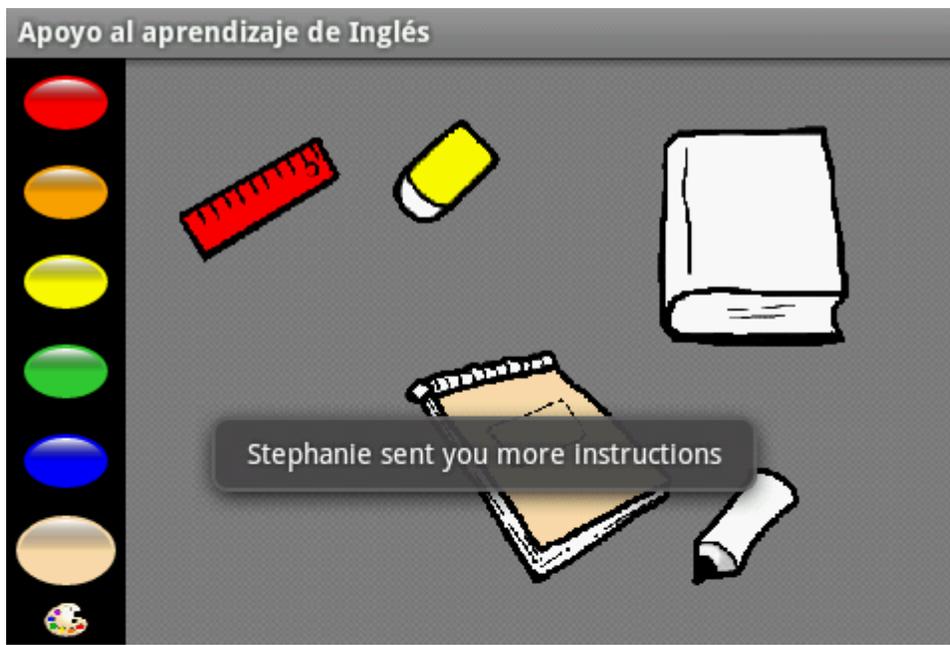


A



B

Figura 29: Compartir instrucciones – Alumno A

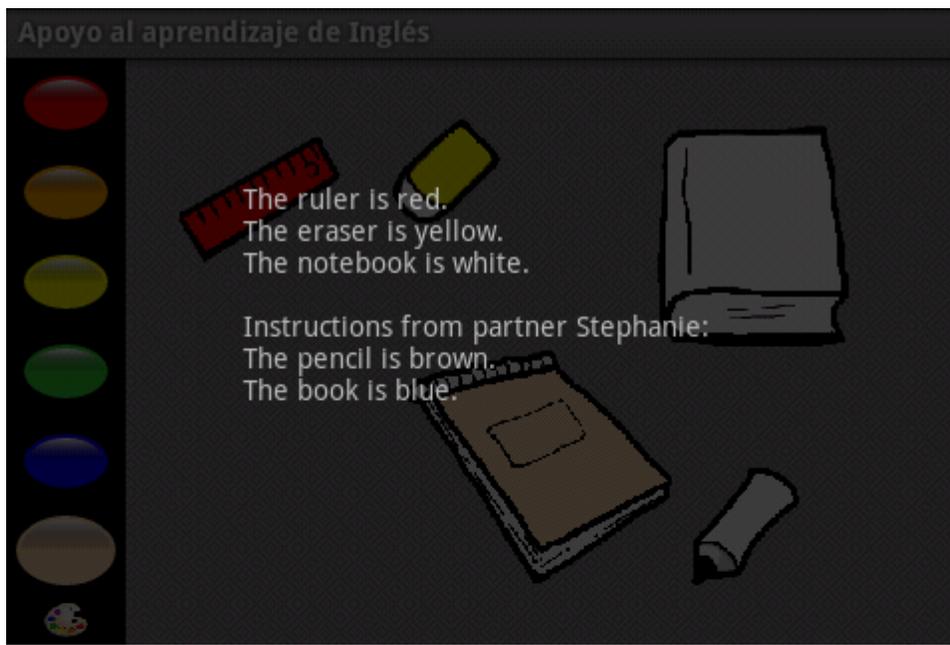


A

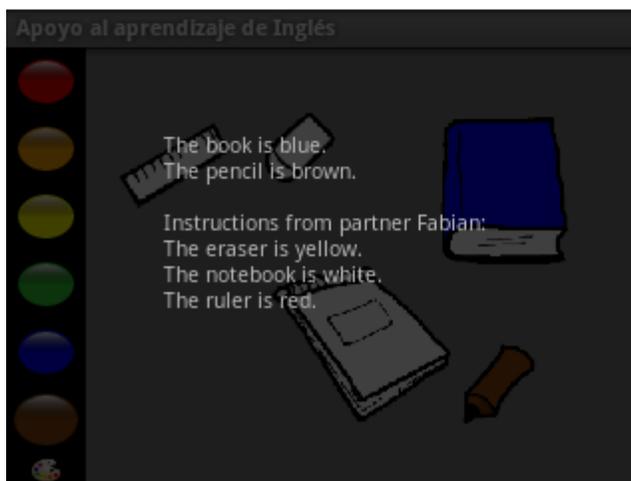


B

Figura 30: Compartir instrucciones – Alumno B



A



B

Figura 31: Nuevas instrucciones



A



B

Figura 32: Sketch concluido

6.2.3. ACTIVIDAD "DESCRIBE THE PICTURE"

- La figura 33 muestra la vista del alumno que describe.
- La figura 34 muestra la vista del alumno que debe adivinar la imagen.
- Las figuras 35, 36 y 37 muestran el proceso de descripción que realiza el alumno.
- Las figuras 38, 39 y 40 muestran lo que recibe el otro alumno y como adivina la imagen.



Figura 33: Alumno que describe

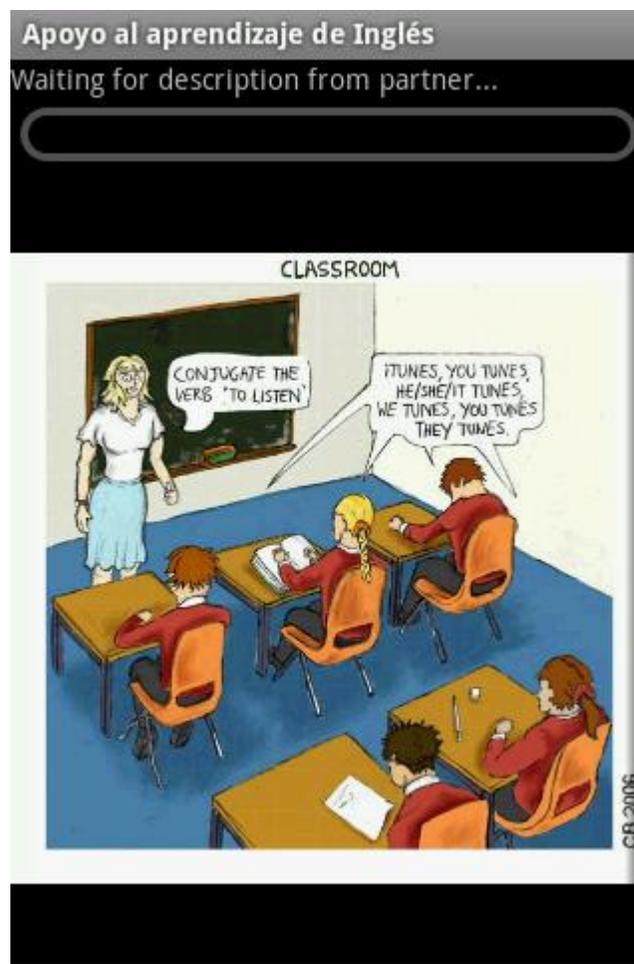


Figura 34: Alumno que adivina la imagen



Figura 35: Alumno agrega descripciones

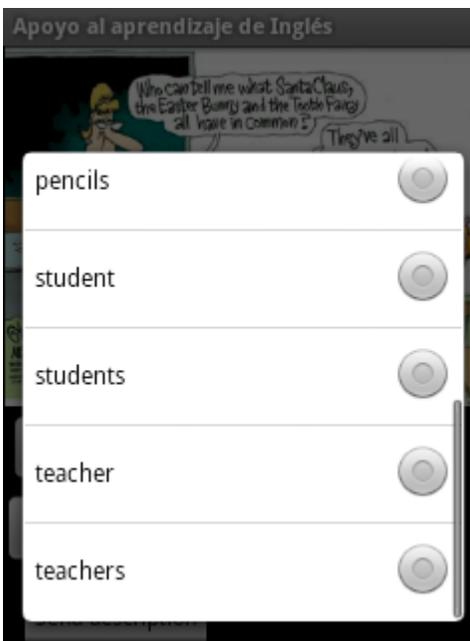


Figura 36: Opciones de objetos



Figura 37: Descripción enviada

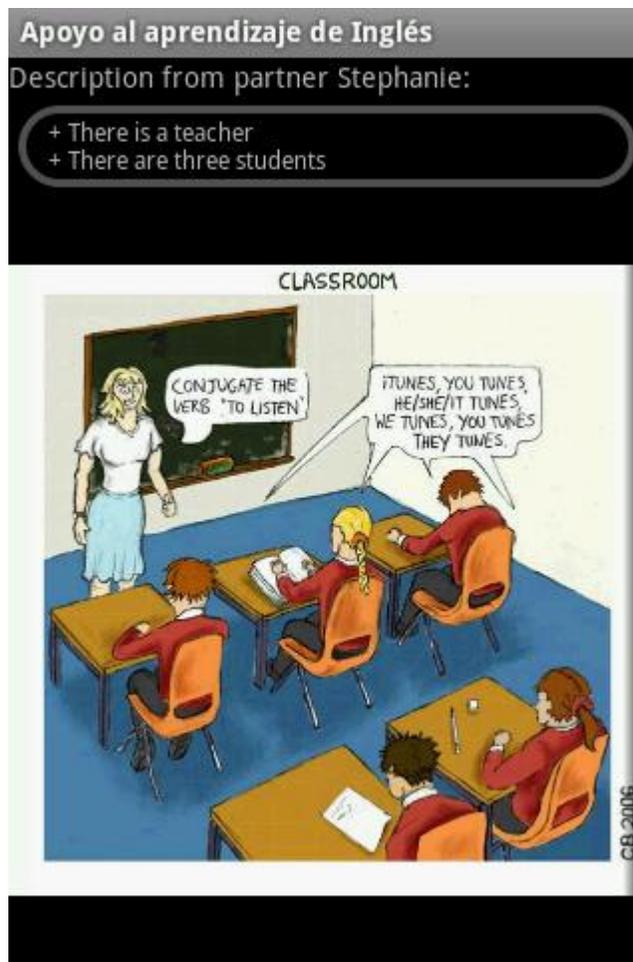


Figura 38: Descripción recibida

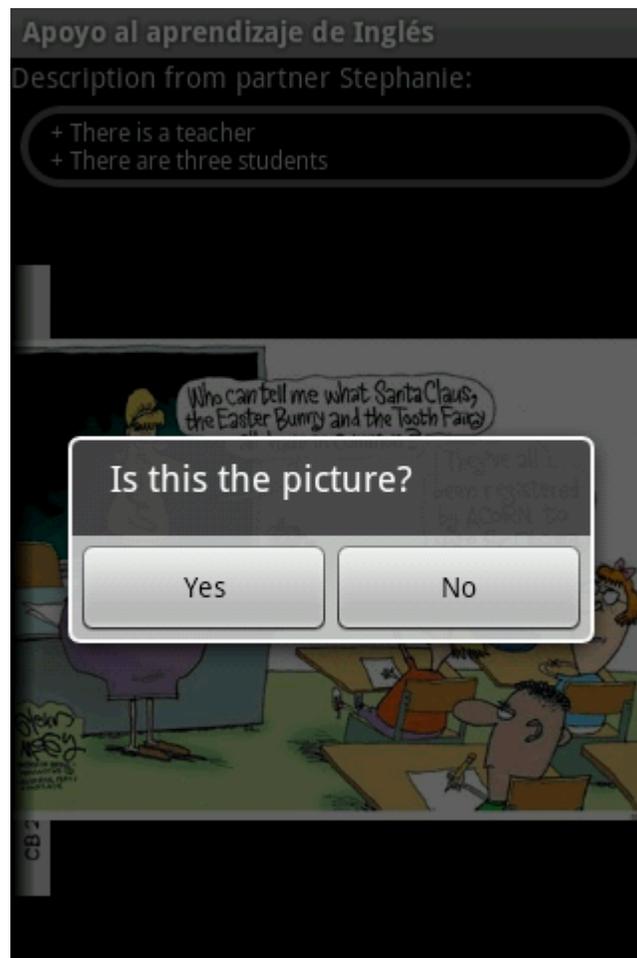


Figura 39: Alumno elige la imagen

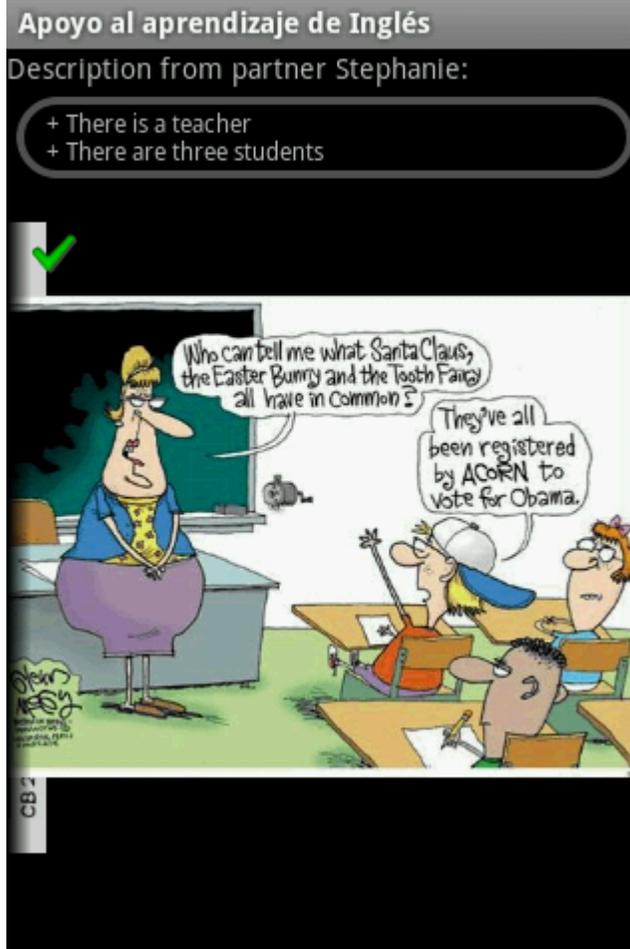


Figura 40: Imagen correctamente adivinada

6.2.4. ACTIVIDAD “FAMILY MEMBERS”

- La figura 41 muestra el inicio de la actividad, una lista de miembros vacía.
- La figura 42, 43, 44, 45, 46 muestra como se agregan los datos de un nuevo miembro: Nombre, parentesco, edad, cumpleaños e imagen asociada.
- La figura 47 y 48 muestran la lista de miembros actualizada.

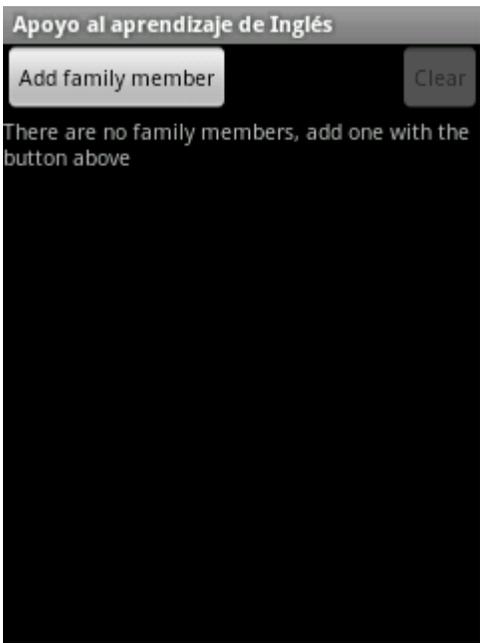


Figura 41: Inicio de “Family Members”

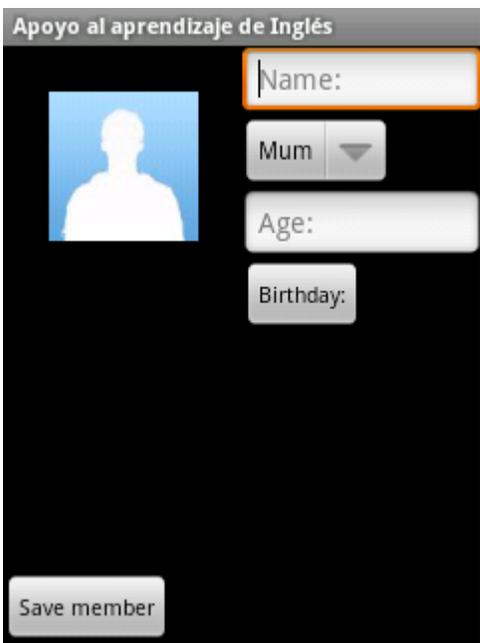


Figura 42: Agregar un miembro

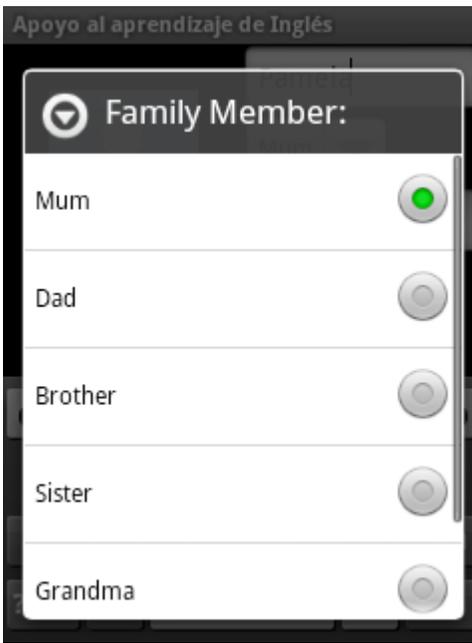


Figura 43: Tipo de miembro



Figura 44: Cumpleaños



Figura 45: Imagen

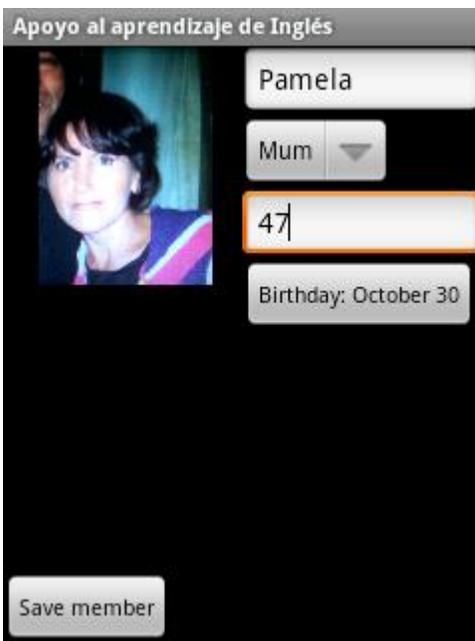


Figura 46: Edad

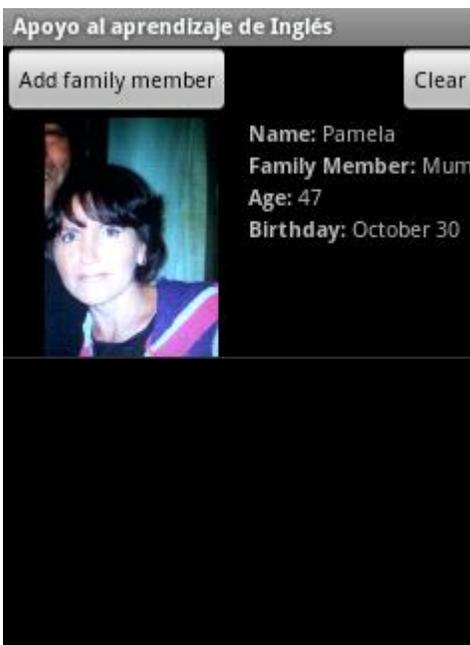


Figura 47: Miembro añadido a la lista



Figura 48: Más miembros

6.2.5. ACTIVIDAD "IN THE HOUSE"

- La figura 49 muestra el inicio de la actividad con la lista de lugares vacía.
- Las figuras 50, 51 y 52 muestra el proceso para añadir un nuevo lugar.
- Las figuras 53 y 54 muestran la lista de lugares actualizada.
- La figura 55 muestra el icono de la casa reemplazado por una foto del usuario.

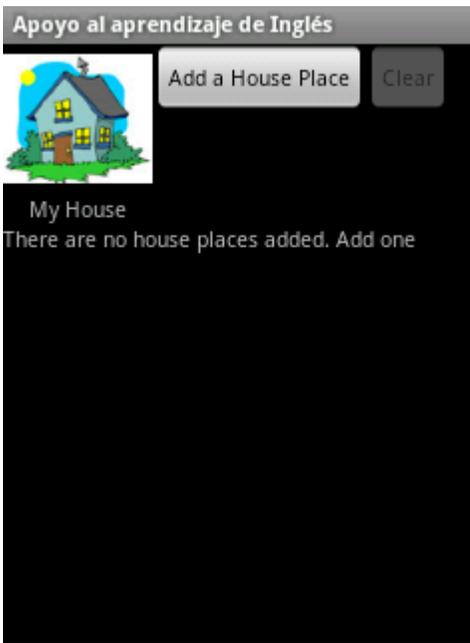


Figura 49: Inicio de “In the House”

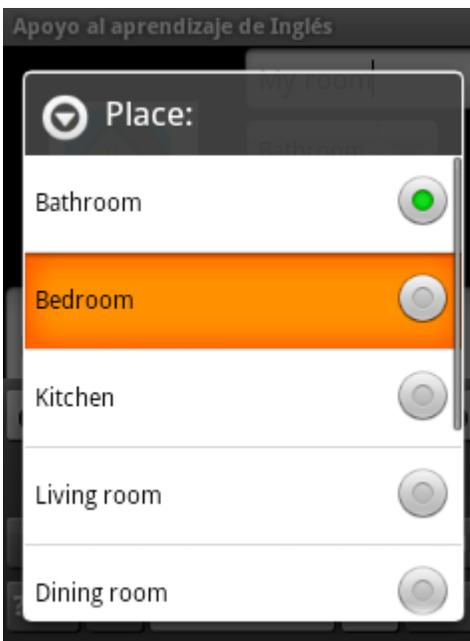


Figura 50: Especificación del lugar

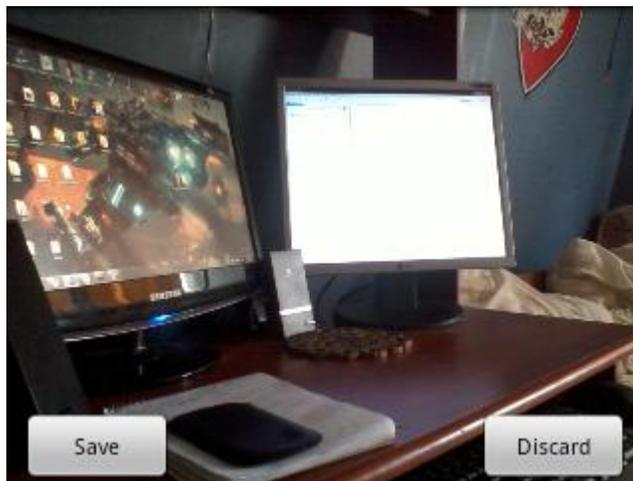


Figura 51: Imagen



Figura 52: Descripción libre

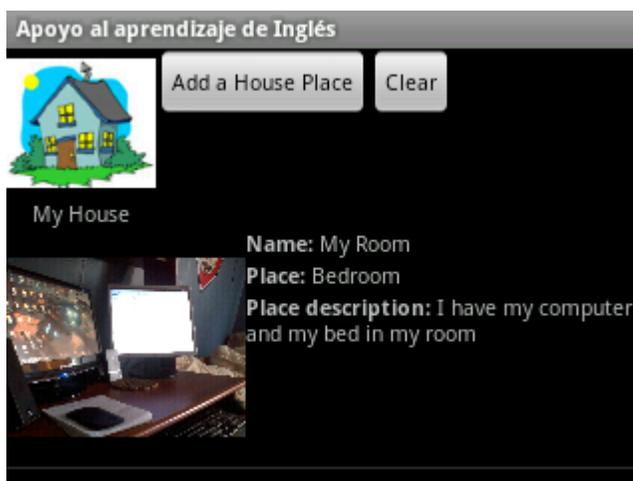


Figura 53: Lugar añadido

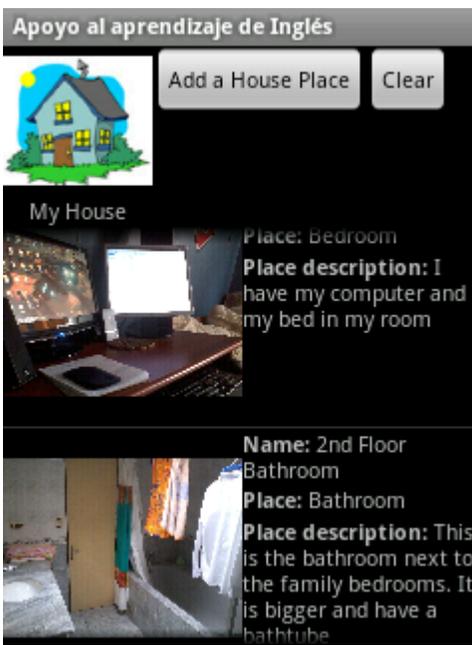


Figura 54: Más lugares

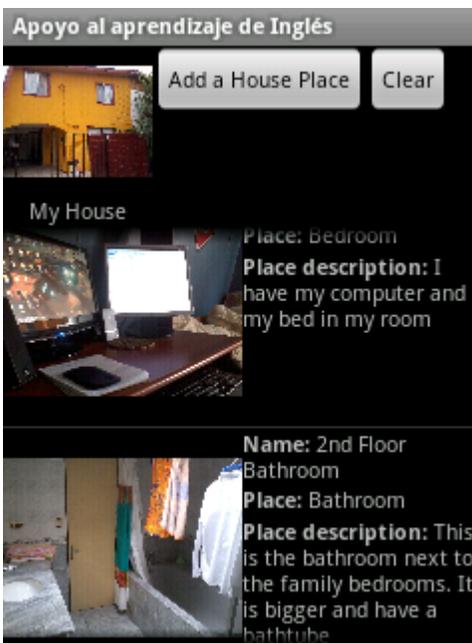


Figura 55: Imagen exterior de la casa

6.2.6. ACTIVIDAD “FOOD GROUPS”

- Las figuras 56 y 57 muestran el inicio de la actividad para 2 alumnos.
- Las figuras 58 y 59 muestran como los alumnos clasifican los alimentos.
- Las figuras 60 y 61 muestran el resultado de la clasificación.

Apoyo al aprendizaje de Inglés

Food: **apple, biscuit, bread, butter, carrot, cheese, coffee, crisps, egg, ice cream, juice, lettuce, milk, orange, pear, rice, sausage, soft drinks, spaghetti, steak, tea, tomatoe.**

Classify					
vegetables	dairy	drinks	snacks and desserts	meat	fruit

Figura 56: Inicio de “Food groups” - Alumno A

Apoyo al aprendizaje de Inglés

Food: **apple, biscuit, bread, butter, carrot, cheese, coffee, crisps, egg, ice cream, juice, lettuce, milk, orange, pear, rice, sausage, soft drinks, spaghetti, steak, tea, tomatoe.**

Classify						
vegetables	dairy	drinks	snacks and desserts	meat	fruits	bread, ce

Figura 57: Inicio de “Food groups” - Alumno B



Figura 58: Alumno A clasifica un alimento

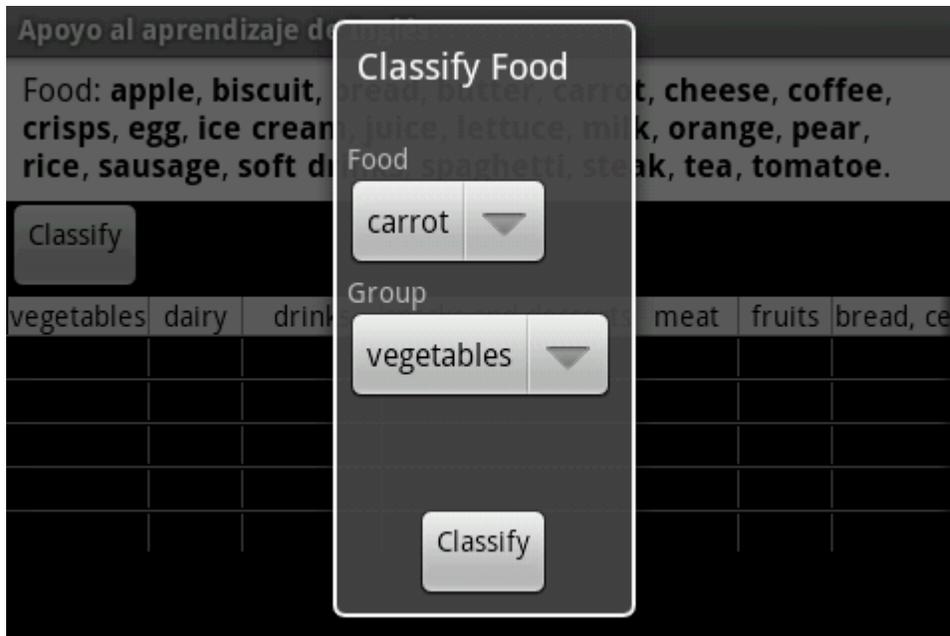


Figura 59: Alumno B clasifica otro alimento

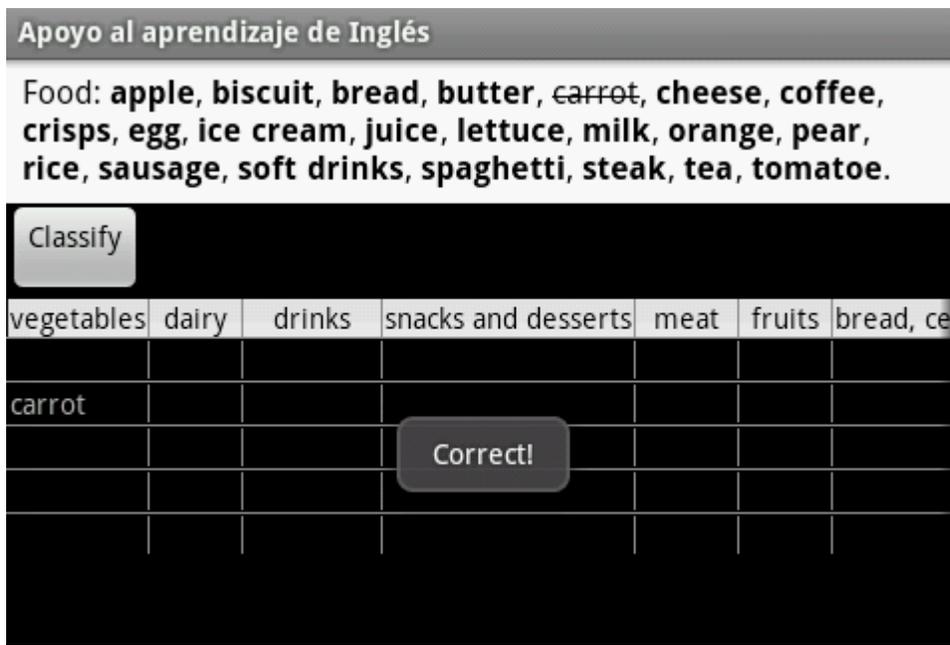


Figura 60: Alumno B clasifica correctamente

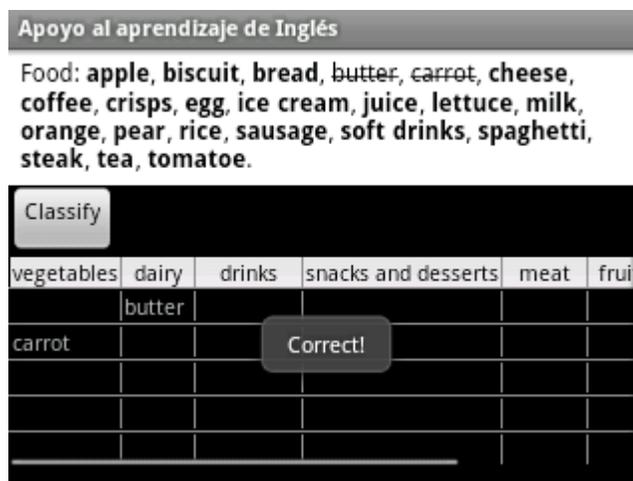


Figura 61: Alumno A clasifica correctamente

6.2.7. ACTIVIDAD "ANIMALS"

- La figura 62 muestra el inicio de la actividad, con la lista de animales vacía.
- Las figuras 63 y 64 muestran como se agregan los detalles de los animales.
- La figura 65 muestra la lista actualizada.

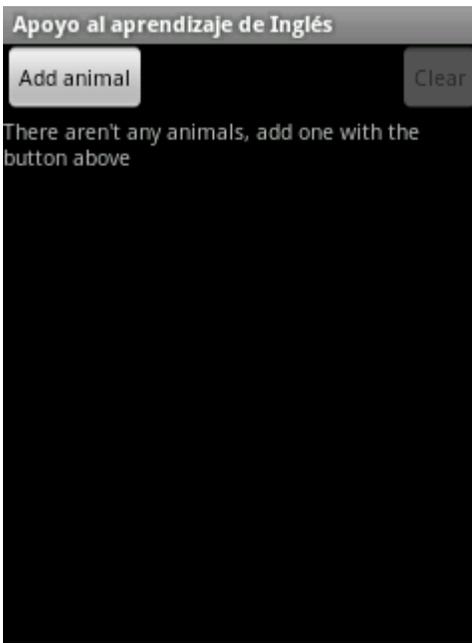


Figura 62: Inicio de "Animals"

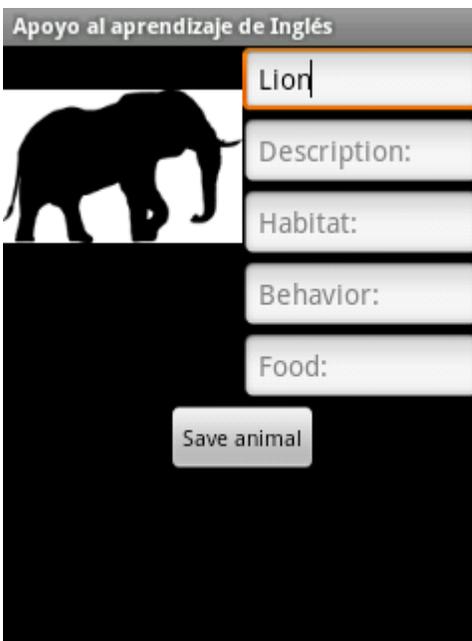


Figura 63: Añadir nuevo animal

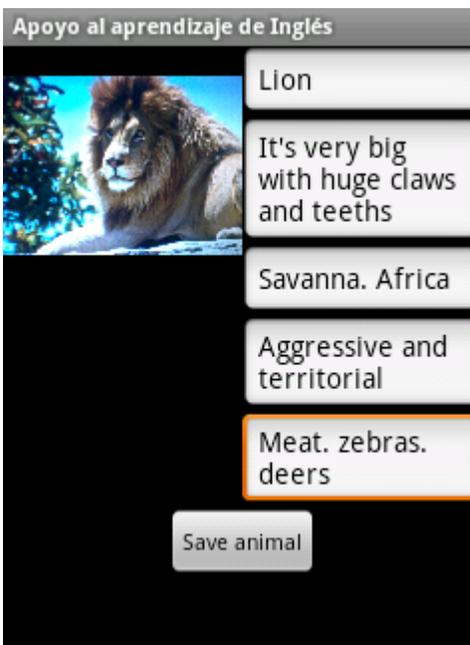


Figura 64: Detalles del nuevo animal

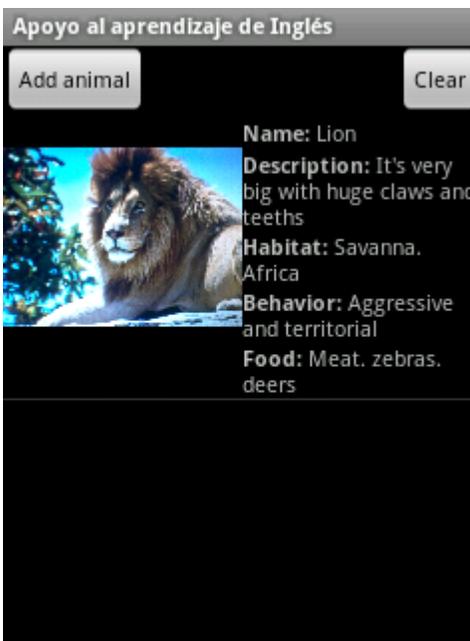


Figura 65: Animal añadido

6.2.8. ACTIVIDAD “CHILEAN NATURE”

- La figura 66 muestra el inicio de la actividad, una grilla con distintas especies.
- La figura 67 muestra las opciones que se proveen al alumno una vez selecciona una especie.
- La figura 68 muestra la imagen ampliada de la especie.
- Las figuras 69 y 70 muestran los detalles de las especies.



Figura 66: Inicio de "Chilean Nature"



Figura 67: Alumno selecciona una especie



Figura 68: Mostrar foto



Species Name: Condor

Endangered Status: Near Threatened

Description:

The **Andean Condor** (*Vultur gryphus*) is a species of South American bird in the New World vulture family Cathartidae and is the only member of the genus **Vultur**. Found in the Andes mountains and adjacent Pacific coasts of western South America, it has the largest wing span (at 3.2 m) of any land bird.

It is a large black vulture with a ruff of white feathers surrounding the base of the neck and, especially in the male.

Figura 69: Descripción de la especie



Species Name: Araucaria

Endangered Status: Vulnerable

Description:

Araucaria araucana (popularly called **The Monkey-puzzle Tree**) is an evergreen tree growing to 40 metres (130 ft) tall with a 2 metres (7 ft) trunk diameter. The tree is native to south-central Chile and west central Argentina. *Araucaria araucana* is the hardiest species in the conifer genus *Araucaria*. Because of the species's great age it is

Figura 70: Descripción de otra especie

7. REFERENCIAS

1. **MINEDUC.** Fundamentos del Ajuste Curricular en el sector del Idioma Extranjero, Inglés. *Curriculum Nacional*. [En línea] 2009. [Citado el: 12 de Julio de 2010.] http://www.curriculum-mineduc.cl/docs/apoyo/articulo_fundamentos_ajuste_ingles_300309.pdf
2. **Shih, Y.-C., & Yang, M.-T.** (2008). A Collaborative Virtual Environment for Situated Language Learning Using VEC3D. *Educational Technology & Society*, 11 (1), 56-68.
3. **Rosetta Stone Ltd.** Language Learning. [Citado el: 12 de Agosto de 2010] <http://www.rosettastone.com/>
4. **Auralog.** Language Learning Software. *Tell Me More*. [Citado el: 12 de Agosto de 2010] <http://www.tellmemore.com/>
5. **British Council.** Mobile Learning. *Learn English*. [Citado el: 12 de Agosto de 2010] <http://learnenglish.britishcouncil.org/en/mobile-learning>
6. **Frohberg, Dirk, Göth, Christoph y Schwabe, Gerhard.** (2009). Mobile Learning projects - a critical analysis of the state of the art. *Journal of Computer Assisted Learning*, 25 (4), 307-331.
7. **Taylor J., Sharples M., Malley C. O', Vavoula G. & Waycott J.** (2006). Towards a task model for mobile learning: a dialectical approach. *International Journal of Learning Technology*, 2 (2-3), 138–158.
8. **Ogata, H.; Chengjiu Yin; Paredes, R.G.; Nobuji, J.; Saito, A.; Yano, Y.; Oishi, Y.; Ueda, T.** (2006) Supporting Mobile Language Learning outside Classrooms. *Proceedings of the Sixth International Conference on Advanced Learning Technologies*, 928 - 932.
9. **Ogata, H.; Gotoda, N.; Miyata, M.; & Yano, Y.:** Development of Mobile and Ubiquitous Learning Environments. In Nelson Baloian, Wolfram Luther, Dirk Soffker, Yoshiyori Urano (Eds.) *Interface and Interaction Design for Learning and Simulation Environments*, (2009) pp. 164-196.
10. **Chen, C.-M., & Hsu, S.-H.** (2008). Personalized Intelligent Mobile Learning System for Supporting Effective English Learning. *Educational Technology & Society*, 11 (3), 153-180
11. **MINEDUC.** Marco Curricular Educación Básica y Media Actualización 2009. *Curriculum Mineduc*. [En línea] 2009. [Citado el: 12 de Julio de 2010.] http://curriculum-mineduc.cl/docs/Marco_Curricular_Ed_Basica_y_Media_Actualizacion_2009.pdf
12. **MINEDUC.** Vigencia Ajuste Curricular. *Curriculum Mineduc*. [En línea] 2010. [Citado el: 1 de Agosto de 2010.] http://www.curriculum-mineduc.cl/docs/Carta_Ajuste_Curricular.pdf.
13. **MINEDUC.** Programa de estudio 5º Básico Inglés. *Curriculum Nacional*. [En línea] 2008. [Citado el: 12 de Julio de 2010.] http://www.curriculum-mineduc.cl/docs/fichas/5b08_ingles.pdf.
14. **MINEDUC.** Programa de Estudio 6º Básico Inglés. *Curriculum Nacional*. [En línea] 2008. [Citado el: 12 de Julio de 2010.] http://www.curriculum-mineduc.cl/docs/fichas/6b08_ingles.pdf

15. **Gartner Inc.** Gartner Says Android to Command Nearly Half of Worldwide Smartphone Operating System Market by Year-End 2012. *Gartner Newsroom*. [En línea] [Citado el: 18 de Abril de 2011] <http://www.gartner.com/it/page.jsp?id=1622614>
16. **Google Inc.** What is Android?. *Android Developers*. [En línea] [Citado el: 12 de Julio de 2010.] <http://developer.android.com/guide/basics/what-is-android.html>.
17. **Ogata, H., Saito, N. A., Paredes J. R. G., San Martin, G. A., & Yano, Y.** (2008). Supporting Classroom Activities with the BSUL System. *Educational Technology & Society*, 11 (1), 1–16.
18. **Yang, J. C., & Lin, Y. L.** (2010). Development and Evaluation of an Interactive Mobile Learning Environment with Shared Display Groupware. *Educational Technology & Society*, 13 (1), 195–207.
19. **The Eclipse Foundation.** Eclipse. *The Eclipse Foundation open source community website*. [En línea] [Citado el: 18 de Abril de 2011.] <http://www.eclipse.org/>
20. **The FreeBSD Foundation.** About FreeBSD. *The FreeBSD Project*. [En línea] [Citado el: 18 de Abril de 2011.] <http://www.freebsd.org/about.html>
21. **Moving Picture Experts Group.** Overview of the MPEG-4 Standard. *MPEG standards*. [En línea] [Citado el: 18 de Abril de 2011.] <http://mpeg.chiariglione.org/standards/mpeg-4/mpeg-4.htm>
22. **International Telecommunication Union.** H.264 : Advanced video coding for generic audiovisual services. *ITU-T Recommendations*. [En línea] [Citado el: 18 de Abril de 2011.] <http://www.itu.int/rec/T-REC-H.264>
23. **Moving Picture Experts Group.** Short MPEG-1 description. *MPEG standards*. [En línea] [Citado el: 18 de Abril de 2011.] <http://mpeg.chiariglione.org/standards/mpeg-1/mpeg-1.htm>
24. **The 3rd Generation Partnership Project.** Mandatory speech CODEC speech processing functions; AMR speech Codec; General description. *3GPP Specification series*. [En línea] [Citado el: 18 de Abril de 2011.] <http://www.3gpp.org/ftp/Specs/html-info/26071.htm>
25. **Joint Photographic Experts Group.** JPEG Homepage. *The JPEG Committee home page*. [En línea] [Citado el: 18 de Abril de 2011.] <http://jpeg.org/jpeg/index.html>
26. **Joint ISO/IEC International Standard, W3C.** Portable Network Graphics (PNG) Specification, W3C/ISO/IEC version. *PNG Home Page*. [En línea] [Citado el: 18 de Abril de 2011.] <http://www.libpng.org/pub/png/spec/iso/>
27. **Skia Project.** Skia – 2D Graphics Library. *Skia Project Home*. [En línea] [Citado el: 18 de Abril de 2011.] <http://code.google.com/p/skia/>
28. **Khronos Group.** OpenGL ES - The Standard for Embedded Accelerated 3D Graphics. *OpenGL ES 1.X*. [En línea] [Citado el: 18 de Abril de 2011.] http://www.khronos.org/opengles/1_X/

29. **SQLite.** About SQLite. *SQLite Home Page*. [En línea] [Citado el: 18 de Abril de 2011.] <http://www.sqlite.org/about.html>
30. **Dalvik Project.** Dalvik – Code and documentation from Android's VM team. *Dalvik Project Home*. [En línea] [Citado el: 18 de Abril de 2011.] <http://code.google.com/p/dalvik/>
31. **Linux Kernel Organization, Inc.** The Linux Kernel Archives. [En línea] [Citado el: 18 de Abril de 2011.] <http://www.kernel.org/>
32. **Chun, D. M., & Plass, J.L.** (1996). Effects of multimedia annotations on vocabulary acquisition. *The Modern Language Journal*, 80, 183-197.
33. **Ogata, H., and Yano, Y.**, "Context-Aware Support for Computer Supported Ubiquitous Learning", *Proc. of IEEE WMTE 2004*, 2004, pp.27-34.
34. **Baker, C.** (1993). *Foundations of Bilingual Education and Bilingualism*. Clevedon: Multilingual Matters.
35. **MINEDUC.** Decreto 254. *Gobierno Transparente*. [En línea] 2009. [Citado el: 18 de Abril de 2010.] <http://www.diarioficial.cl/GobiernoTransparente/mineduc/NG/DCTO/2009/08/48661.html>
36. **MINEDUC.** Decreto 256. *Gobierno Transparente*. [En línea] 2009. [Citado el: 18 de Abril de 2010.] <http://www.diarioficial.cl/GobiernoTransparente/mineduc/NG/DCTO/2009/08/48662.html>
37. **Zebra Paint.** Open source version of Zebra Paint. *Zebra-paint project home*. [En línea] [Citado el: 18 de Abril de 2010.] <http://code.google.com/p/zebra-paint/>
38. **SUBTEL.** Información Estadística. Subsecretaría de Telecomunicaciones. [En línea] [Citado el: 18 de Abril de 2010.] http://www.subtel.cl/prontus_subtel/site/artic/20070212/pags/20070212182348.html
39. **Qt.** Qt - Cross-platform application and UI framework. *Qt home page*. [En línea] [Citado el: 18 de Abril de 2010.] <http://qt.nokia.com/>