

**Universidad Autónoma Metropolitana Unidad Azcapotzalco**

División de Ciencias Básicas e Ingeniería

Manual de usuario

Licenciatura en Ingeniería en Computación

Aplicación Android para practicar la pronunciación de los idioms del idioma inglés

Modalidad: Proyecto Tecnológico

Trimestre 2015 Primavera

José Salvador López Balleza

Matricula: 210304254

jslb\_cafcb10@hotmail.com



Asesor:

Dra. María Lizbeth Gallardo López

Profesor Investigador

Departamento de Sistemas

glizbeth@correo.azc.uam.mx



Fecha de entrega: 22 de julio de 2015

## Índice

1	Introducción .....	3
2	Pantalla inicial .....	5
3	Sección Memory Game .....	6
4	Sección Multiple Choice .....	10
5	Sección Voice and Image .....	14
6	Sección Scores.....	18

## Índice de figuras

Figura 1 Pantalla inicial de la aplicación.....	5
Figura 2 Pantalla inicial de la sección <i>Memory Game</i> .....	6
Figura 3 Pantalla al dar click a una imagen.....	6
Figura 4 Pantalla al dar click a una segunda imagen.....	7
Figura 5 Se ocultan las imágenes anteriores y se muestra una nueva .....	7
Figura 6 Pantalla mostrada al encontrar dos imágenes iguales .....	8
Figura 7 Pantalla final de <i>Memory Game</i> .....	8
Figura 8 Pantalla <i>Scores</i> de <i>Memory Game</i> .....	9
Figura 9 Pantalla inicial de <i>Multiple Choice</i> .....	10
Figura 10 Pantalla al seleccionar la opción "Comenzar" .....	10
Figura 11 Pantalla con cuadro de diálogo que activa micrófono del dispositivo.....	11
Figura 12 Pantalla con lista de posibles frases semejantes a lo pronunciado.....	11
Figura 13 Pantalla al obtener un acierto .....	12
Figura 14 Pantalla al obtener una respuesta incorrecta .....	12
Figura 15 Pantalla <i>Scores</i> de <i>Multiple Choice</i> .....	13
Figura 16 Pantalla inicial de <i>Voice and Image</i> .....	14
Figura 17 Pantalla después de seleccionar "Comenzar" .....	14
Figura 18 Pantalla después de seleccionar el botón con la imagen del micrófono .....	15
Figura 19 Pantalla con la lista de frases semejantes a lo pronunciado.....	15
Figura 20 Pantalla mostrada cuando se obtiene un acierto.....	16
Figura 21 Pantalla mostrada cuando se obtiene un error.....	16
Figura 22 Pantalla <i>Scores</i> de <i>Voice and Image</i> .....	17
Figura 23 Pantalla <i>Scores</i> .....	18

# 1 Introducción

El inglés es considerado una lengua universal, principalmente porque es uno de los idiomas más hablados en el mundo y el más empleado en casi todas las áreas de conocimiento.

Aprender el idioma inglés requiere de práctica en cuanto a la gramática y la pronunciación. Para ello, se están proponiendo gran variedad de aplicaciones móviles, lo cual abre el espectro de posibilidades para practicar este idioma. Es conocido que la memoria humana asocia una imagen a la frase de un idioma, lo cual le ayuda a recordar la frase y situarla en el contexto adecuado.

El presente proyecto tiene la finalidad de entregar al usuario una aplicación para su dispositivo móvil con Sistema Operativo Android que pueda utilizar en cualquier momento y en cualquier lugar para practicar la pronunciación de frases, específicamente los conocidos como *idioms*; esta aplicación recibe el nombre de *Idioms Game*. Los *idioms* se definen como, “Grupo de palabras que poseen un significado diferente al de cada una de las que lo componen” [1], “Grupo de palabras cuyo significado se deduce del contexto de la frase y no del significado individual de cada palabra que lo componen”[2]; por ejemplo, *A bed of roses* literalmente se traduce al español como: una cama de rosas, pero su significado tiene que ver con el contexto cultural del país de habla inglesa; por lo tanto, la traducción al español es: fácil, agradable, placentero. El hecho de que estas frases no tengan una traducción literal al español pueden convertirse en un obstáculo para estudiantes del idioma inglés, por consiguiente, *Idioms Game* servirá para practicar la pronunciación de los *idioms*, y eventualmente el aprendizaje de los mismos, auxiliándose de tres módulos principales. Uno de ellos tiene la tarea de familiarizar al jugador con las relaciones entre las imágenes e *idioms* y los dos módulos restantes tienen el objetivo de reconocer la voz del usuario, transformando la voz a texto escrito.

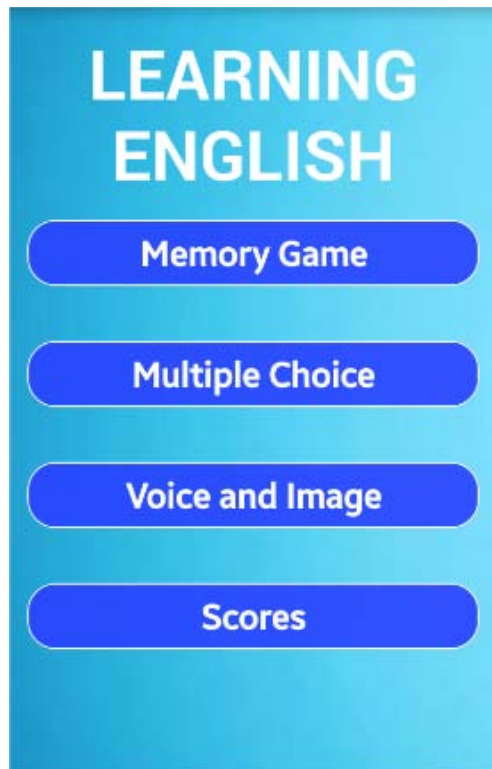
Este manual sirve como guía para el usuario de la aplicación *Idioms Game*; se explica el funcionamiento de sus principales secciones:

- Jugar *Memory Game*
- Jugar *Multiple Choice*
- Jugar *Voice and Image*
- Scores

A continuación, se va a mostrar cómo funcionan algunos módulos de la aplicación *Idioms Game*, partiendo de la pantalla principal; luego se mostrarán las funcionalidades disponibles de la sección jugar *Memory Game*, *Multiple Choice*, *Voice and Image*.

## 2 Pantalla inicial

En la Figura 1 se muestra el menú principal de la aplicación, el cual consta de cuatro opciones: *Memory Game*, *Multiple Choice* y *Voice and Image*; además de ver los puntajes más altos que se encuentra en la opción “Scores”. Al elegir una opción se desplegará la ventana del mini-juego correspondiente.



**Figura 1** Pantalla inicial de la aplicación

### 3 Sección Memory Game

Al iniciar la sección *Memory Game* se visualizan doce imágenes de un signo de interrogación además de un cronómetro que comienza a contar desde cero (Ver Figura 2).



Figura 2 Pantalla inicial de la sección *Memory Game*

Al dar click en alguna de las imágenes, la aplicación muestra aleatoriamente una imagen que trae consigo un rótulo con el respectivo *idiom* que le corresponde (Ver Figura 3).



Figura 3 Pantalla al dar click a una imagen

Al dar click en otra imagen, nuevamente la aplicación muestra una imagen aleatoria.



**Figura 4 Pantalla al dar click a una segunda imagen**

En este caso (Ver Figura 4) las imágenes mostradas no son iguales, por lo que la aplicación ocultará estas imágenes y mostrará nuevamente la imagen con el signo de interrogación hasta volver a dar click a otra ellas.



**Figura 5 Se ocultan las imágenes anteriores y se muestra una nueva**

Al encontrar dos imágenes iguales, la aplicación las dejará visibles y el usuario repetirá el mismo procedimiento para intentar encontrar los seis pares posibles para ganar *Memory Game*.



**Figura 6** Pantalla mostrada al encontrar dos imágenes iguales

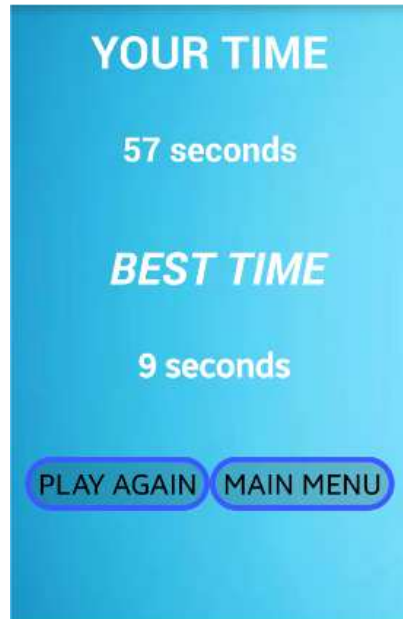
A continuación, la Figura 7 muestra la pantalla que despliega la aplicación al encontrar los seis pares de imágenes.



**Figura 7** Pantalla final de Memory Game



En la Figura 7 se observa que la aplicación detiene el cronómetro, y además, habilita el botón “Continuar”, al dar click en esta opción, la aplicación dirige a la pantalla “Scores” (Figura 8), la cual muestra al usuario la leyenda YOUR TIME que es el tiempo obtenido al encontrar los seis pares posibles del mini-juego, también se muestra la leyenda BEST TIME que es el menor tiempo obtenido en todas las partidas jugadas de *Memory Game*.

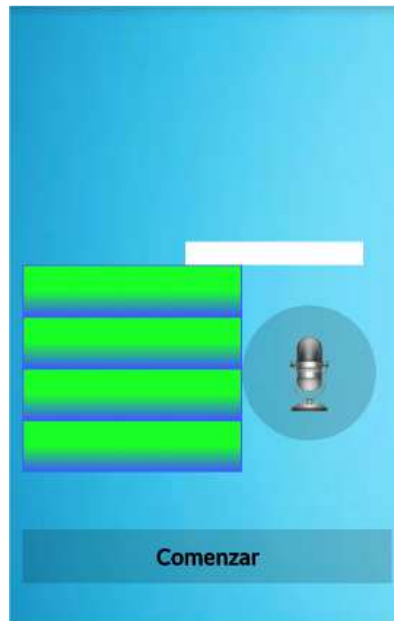


**Figura 8 Pantalla Scores de *Memory Game***

Dentro de la pantalla (Figura 8) aparecen dos opciones. La opción PLAY AGAIN, permite reiniciar el mini-juego y la opción MAIN MENU que dirige a la pantalla inicial donde se encuentra el menú principal de la aplicación.

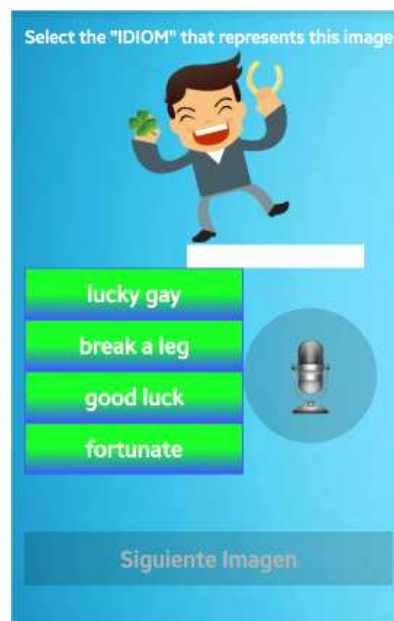
## 4 Sección Multiple Choice

La sección Multiple Choice, muestra la opción “Comenzar”, la cual permite iniciar con la dinámica del mini-juego *Multiple Choice* (Ver Figura 9).



**Figura 9** Pantalla inicial de *Multiple Choice*

Al dar click en la opción “Comenzar”, la aplicación despliega en pantalla una imagen junto con cuatro cuadros de texto en donde cada uno contiene una frase (Ver Figura 10).



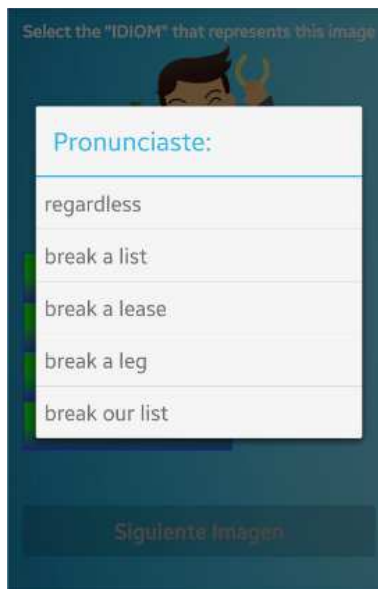
**Figura 10** Pantalla al seleccionar la opción "Comenzar"

Como se observa en la Figura 10, se visualiza un botón con la imagen de un micrófono, al seleccionarlo, la aplicación desplegará un cuadro de diálogo con la leyenda “Habla ahora”, lo cual indica que la aplicación se encuentra lista para escuchar una frase de las cuatro posibles opciones que aparecen en pantalla (Ver Figura 11).



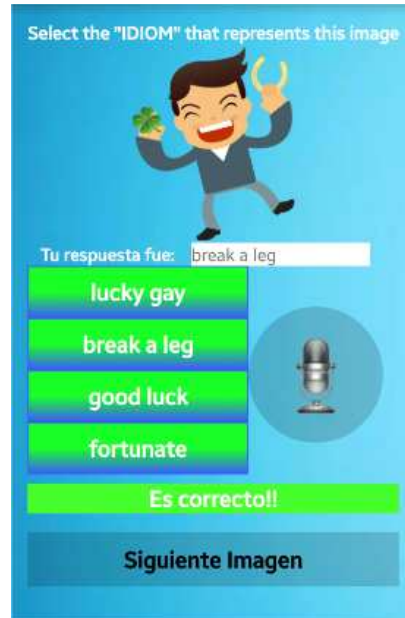
**Figura 11 Pantalla con cuadro de diálogo que activa micrófono del dispositivo**

Al pronunciar alguna frase, la aplicación muestra una lista con las frases que se asemejan a lo pronunciado con el micrófono del dispositivo (Ver Figura 12). El usuario tiene que dar click a la opción de la lista que sea exactamente igual a lo que pronunció.



**Figura 12 Pantalla con lista de posibles frases semejantes a lo pronunciado**

Si el usuario pronunció la respuesta correcta, la aplicación mostrará un mensaje en color verde con el texto: “Es correcto!!”, aumentará en uno el puntaje (Ver Figura 13 ).



**Figura 13 Pantalla al obtener un acierto**

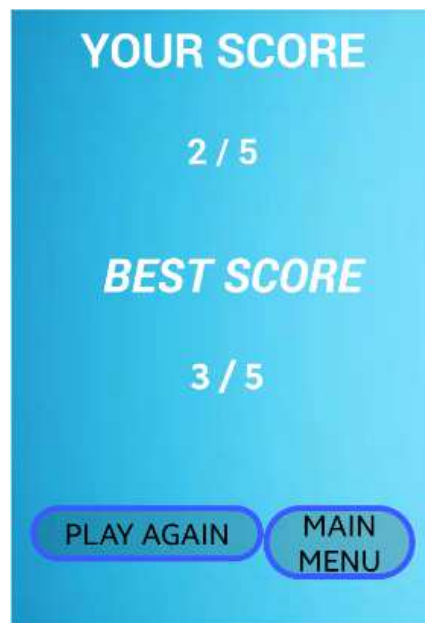
En caso contrario, si el usuario pronuncia una respuesta incorrecta, la aplicación muestra un mensaje en color rojo con el texto: “Incorrecto!, la respuesta correcta es: respuesta\_correcta (Ver Figura 14).



**Figura 14 Pantalla al obtener una respuesta incorrecta**

Como se observa tanto en la Figura 13 como en la Figura 14 al momento de obtener el resultado sobre lo que se pronunció, la aplicación habilita la opción “Siguiete Imagen”, la cual permite mostrar una nueva imagen tomada aleatoriamente con sus cuatro posibles respuestas.

El procedimiento de *Multiple Choice* termina cuando el usuario haya respondido a 10 imágenes, pues a partir de aquí se muestra la pantalla *Scores* (Ver Figura 15 ) que muestra la leyenda YOUR SCORE que contiene el número de aciertos obtenidos en la partida actual y la leyenda BEST SCORE que contiene el máximo número de aciertos que se ha obtenido en todas las partidas jugadas.



**Figura 15 Pantalla *Scores* de *Multiple Choice***

En la pantalla de la Figura 15 se tienen las opciones PLAY AGAIN y MAIN MENU, la primera permite regresar a la pantalla inicial del mini-juego (Figura 9) para reiniciar y comenzar de nuevo con la dinámica del mini-juego y la segunda permite regresar a la pantalla inicial de la aplicación

## 5 Sección Voice and Image

En esta sección se muestran dos botones, uno de ellos contiene el texto: “Comenzar” y el otro contiene la imagen de un micrófono, al seleccionar el primero se inicia la dinámica del mini-juego *Voice and Image* mostrando una imagen.



Figura 16 Pantalla inicial de *Voice and Image*



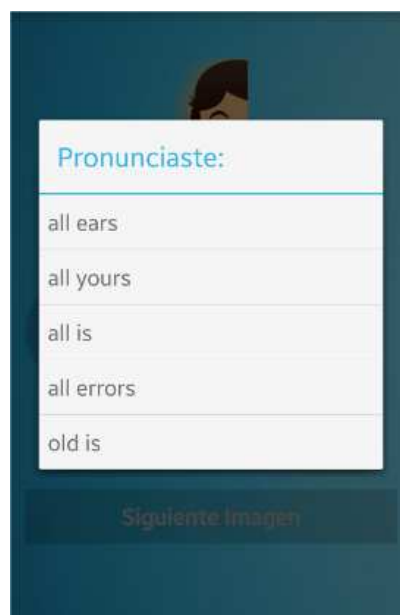
Figura 17 Pantalla después de seleccionar "Comenzar"

Al dar click en el botón que contiene la imagen del micrófono, la aplicación despliega en pantalla un cuadro de dialogo con el texto: “Habla ahora”, lo cual indica que el micrófono del dispositivo se encuentra listo para escuchar una frase (Ver Figura 18).



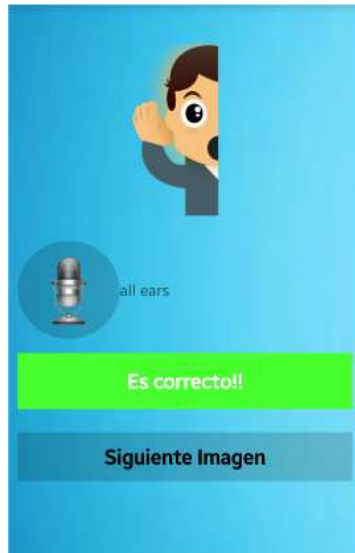
**Figura 18 Pantalla después de seleccionar el botón con la imagen del micrófono**

Al pronunciar alguna frase, la aplicación muestra una lista con las frases que se asemejan a lo pronunciado por el usuario, éste tiene que seleccionar una opción de la lista, la cual debe de ser exactamente la misma frase que pronunció (Ver Figura 19).



**Figura 19 Pantalla con la lista de frases semejantes a lo pronunciado**

Si el usuario pronunció y seleccionó de la lista (Figura 19) el *idiom* que le corresponde a la imagen mostrada, entonces la aplicación desplegará un mensaje en color verde con el texto: “Es correcto!!”, lo que indica que se obtuvo un acierto y por lo tanto el puntaje aumenta en uno (Ver Figura 20).



**Figura 20 Pantalla mostrada cuando se obtiene un acierto**

En el otro caso, cuando el usuario pronunció un *idiom* distinto o seleccionó una opción de la lista (Figura 19) diferente a la que se tiene como correcta, la aplicación mostrará un mensaje en color rojo con el texto: “Incorrecto!, la respuesta correcta es: respuesta\_correcta”, lo que indica que se obtuvo un error y por lo tanto el puntaje aumenta en cero unidades.

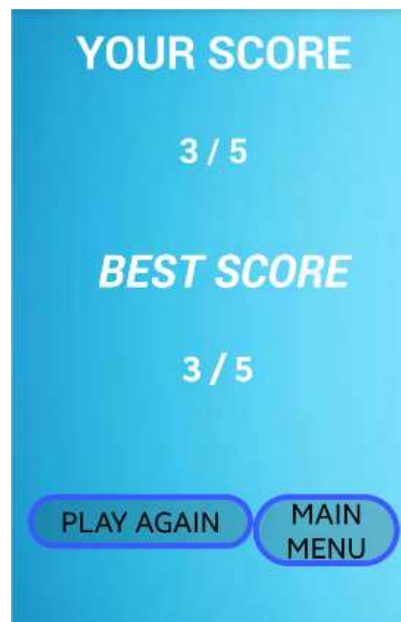


**Figura 21 Pantalla mostrada cuando se obtiene un error**



Como se observa en las imágenes anteriores (Figura 20 y Figura 21) al momento de obtener el texto sobre lo pronunciado por el usuario, la aplicación habilita la opción “Siguiete Imagen”, al seleccionarla, permite mostrar una nueva imagen tomada aleatoriamente.

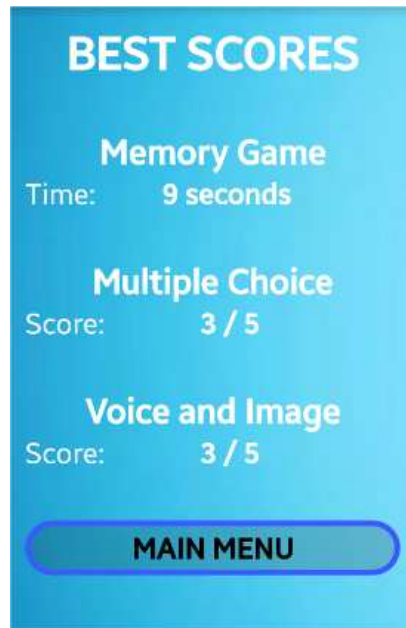
El proceso de *Voice and Image* termina cuando el usuario haya respondido a 10 imágenes. A partir de aquí se muestra la pantalla *Scores* (Ver Figura 22) que muestra la leyenda YOUR SCORE que contiene el número de aciertos obtenidos en la partida actual y la leyenda BEST SCORE que contiene el máximo número de aciertos que se ha obtenido en todas las partidas jugadas.



**Figura 22** Pantalla *Scores* de *Voice and Image*

## 6 Sección Scores

En esta sección se encuentran los mejores puntajes obtenidos en cada mini-juego. El mejor puntaje para el mini-juego *Memory Game* está definido por el menor tiempo en encontrar los 6 pares de imágenes por parte del jugador. El mejor puntaje para *Multiple Choice* y para *Voice and Image* está definido por el mayor número de aciertos por parte del jugador (Ver Figura 23).



**Figura 23 Pantalla Scores**

En la sección *Scores* se observa también la opción MAIN MENU, al seleccionarla, permite regresar al menú principal de la aplicación.

**Universidad Autónoma Metropolitana Unidad Azcapotzalco**

División de Ciencias Básicas e Ingeniería

Manual Técnico del Proyecto de Integración

Licenciatura en Ingeniería en Computación

Aplicación Android para practicar la pronunciación de los idioms del idioma inglés

Modalidad: Proyecto Tecnológico

Trimestre 2015 Primavera

José Salvador López Balleza

Matricula: 210304254

jslb\_cafcb10@hotmail.com



Asesor:

Dra. María Lizbeth Gallardo López

Profesor Investigador

Departamento de Sistemas

glizbeth@correo.azc.uam.mx



Fecha de entrega: 22 de julio de 2015

## Índice

1	Introducción.....	4
2	Ambiente de desarrollo del sistema.....	5
2.1	Hardware.....	5
2.2	Software utilizado .....	5
2.2.1	Android Studio IDE.....	5
2.2.2	Lenguaje de programación .....	6
2.2.3	Google Voice Search.....	6
2.2.4	SharedPreferences .....	6
2.2.5	Sistema Operativo .....	7
2.3	Estructura del directorio del código de la aplicación.....	7
3	Ambiente de producción del sistema.....	11
3.1	Hardware.....	11
3.2	Software .....	11
3.3	Proceso de instalación de la aplicación.....	12
3.4	Pruebas.....	14
3.5	Problemas durante el desarrollo.....	16
4	Bibliografía.....	17

## Índice de figuras

Figura 1 Estructura del proyecto Android .....	7
Figura 2 Estructura de la carpeta app .....	7
Figura 3 Contenido de la carpeta main .....	8
Figura 4 Estructura de la carpeta assets .....	8
Figura 5 Estructura de la carpeta res .....	9
Figura 6 Estructura de la carpeta layout .....	9
Figura 7 Estructura de la carpeta values .....	10
Figura 8 Fichero AndroidManifest.xml .....	10
Figura 9 Pantalla para buscar la aplicación Google .....	12
Figura 10 Pantalla que muestra la aplicación que se debe de elegir .....	13
Figura 11 Pantalla que se muestra al finalizar la descarga .....	13
Figura 12 Funcionamiento de reconocimiento de voz .....	14
Figura 13 Puntajes al ejecutar por primera vez la aplicación .....	15
Figura 14 Puntajes en la segunda ejecución de la aplicación .....	15

# 1 Introducción

El inglés es considerado una lengua universal, principalmente porque es uno de los idiomas más hablados en el mundo y el más empleado en casi todas las áreas de conocimiento.

Aprender el idioma inglés requiere de práctica en cuanto a la gramática y la pronunciación. Para ello, se están proponiendo gran variedad de aplicaciones móviles, lo cual abre el espectro de posibilidades para practicar este idioma. Es conocido que la memoria humana asocia una imagen a la frase de un idioma, lo cual le ayuda a recordar la frase y situarla en el contexto adecuado.

El presente proyecto tiene la finalidad de entregar al usuario una aplicación para su dispositivo móvil con Sistema Operativo Android que pueda utilizar en cualquier momento y en cualquier lugar para practicar la pronunciación de frases, específicamente los conocidos como *idioms*; esta aplicación recibe el nombre de *Idioms Game*. Los *idioms* se definen como, “Grupo de palabras que poseen un significado diferente al de cada una de las que lo componen” [1], “Grupo de palabras cuyo significado se deduce del contexto de la frase y no del significado individual de cada palabra que lo componen”[2]; por ejemplo, *A bed of roses* literalmente se traduce al español como: una cama de rosas, pero su significado tiene que ver con el contexto cultural del país de habla inglesa; por lo tanto, la traducción al español es: fácil, agradable, placentero. El hecho de que estas frases no tengan una traducción literal al español pueden convertirse en un obstáculo para estudiantes del idioma inglés, por consiguiente, *Idioms Game* servirá para practicar la pronunciación de los *idioms*, y eventualmente el aprendizaje de los mismos, auxiliándose de tres módulos principales. Uno de ellos tiene la tarea de familiarizar al jugador con las relaciones entre las imágenes e *idioms* y los dos módulos restantes tienen el objetivo de reconocer la voz del usuario, transformando la voz a texto escrito.

El presente documento está conformado por tres secciones. La sección 2 presenta el ambiente de desarrollo, donde se describe tanto el hardware como el software empleados en el desarrollo de *Idioms Game*; también, se describe cada uno de los componentes de software utilizados durante su implementación; además se describe detalladamente la estructura de directorios del proyecto de la aplicación. En la sección 3 se describe el ambiente de producción, donde se describe el hardware y software necesarios para poner en producción a *Idioms Game*; además, se describe el procedimiento para realizar su correcta instalación; también se describe detalladamente la serie de pruebas que fueron realizadas durante su desarrollo, y finalmente, se describen los problemas que se presentaron durante la implementación.

## 2 Ambiente de desarrollo del sistema

### 2.1 Hardware

La aplicación fue desarrollada e instalada en una Laptop Dell Inspiron15R N5110 con los siguientes recursos:

- Memoria RAM 6 GB
- Disco duro 500 GB
- Procesador Intel(R) Core(TM) i5-2410M a 2.30 GHz
- Sistema Operativo Windows 7 Professional

Dispositivos de experimentación y pruebas

- *Smartphone Samsung Galaxy S5 SM-G900M*
  - Pantalla de 5.1 pulgadas
  - Procesador Quad-Core de 2.5 GHz
  - Sistema Operativo Android 4.4.2
- *Tablet Samsung Galaxy Tab 3*
  - Pantalla de 7 pulgadas
  - Procesador Dual-Core a 1.2 GHZ
  - Sistema Operativo Android 4.2.2

### 2.2 Software utilizado

El software utilizado para el desarrollo del proyecto es de licencia libre. A continuación se describe cada uno de ellos.

#### 2.2.1 Android Studio IDE

Android Studio[1] versión 1.0.1 es un entorno de desarrollo integrado (IDE) para la plataforma Android, está programado en el lenguaje de programación Java y además se encuentra disponible para probarlo gratuitamente. Basado en IntelliJ IDEA <sup>1</sup>de JetBrains, está diseñado específicamente para desarrollar para Android.

---

<sup>1</sup>IntelliJ IDEA es un entorno de desarrollo integrado construido por JetBrains utilizado para el desarrollo de programas de software.

## 2.2.2 Lenguaje de programación

El código de la aplicación fue implementado con el lenguaje de programación orientado a objetos Java, versión 8 [2] y el conjunto de herramientas de desarrollo SDK<sup>2</sup> de Android.

## 2.2.3 Google Voice Search

Es una implementación propuesta por Google añadida en el API<sup>3</sup> nivel 3 de Android, se utilizó la versión 4.7.13.19 [3]. Ésta se utilizó para desarrollar el reconocimiento por voz, responde a un Recognizer Intent mostrando un cuadro de diálogo con “Habla ahora” y transmite la información a los servidores de Google para que estos procesen la información y transformen lo pronunciado por el usuario a texto. Para el correcto funcionamiento de la aplicación es necesario, desde Google Voice Search, ajustar el idioma por defecto a inglés, para que los servidores de Google arrojen sólo respuestas en este idioma y de esta manera se logre el cometido de intentar pronunciar correctamente las frases del idioma inglés.

## 2.2.4 SharedPreferences

Para la implementación de la aplicación se utilizó *Shared Preferences*, la cual fue añadida en el API nivel 1[4]. Las Preferencias Compartidas (traducción en español) no son más que datos que una aplicación debe guardar para personalizar la experiencia del usuario. Cada preferencia se almacenará en forma de clave-valor, es decir, cada una de ellas estará compuesta por un identificador único y un valor asociado a dicho identificador. Además, los datos se guardan en un archivo XML que se almacena en una ruta que sigue el siguiente patrón:

```
/data/data/paquete.java/shared_prefs/nombre_coleccion.xml
```

Si descargamos este fichero y lo abrimos con cualquier editor de texto veremos un contenido como el siguiente:

```
<?xmlversion='1.0'encoding='utf-8'standalone='yes'?>
<map>
  <intname="aciertos">numeroAciertos</int>
</map>
```

Para la aplicación, este método fue utilizado para guardar los mejores puntajes obtenidos por el jugador en los mini-juegos y de esta manera se puedan visualizar cada vez que se ejecute la aplicación.

---

<sup>2</sup> SDK por sus siglas en inglés Software Development Kit, es un conjunto de herramientas de desarrollo de software que permite al programador crear aplicaciones.

<sup>3</sup> Interfaz de programación de aplicaciones (API – Application Programming Interface). Conjunto de funciones o métodos que exponen las clases de alguna biblioteca para ser utilizado por otro software.

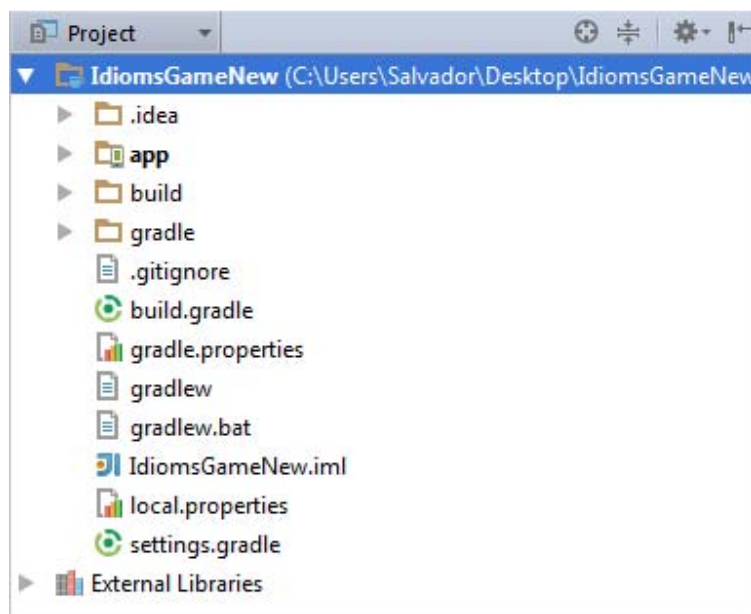


### 2.2.5 Sistema Operativo

El Sistema Operativo utilizado durante todo el desarrollo de la aplicación, junto con todas las tecnologías mencionadas anteriormente fue Windows 7 Professional.

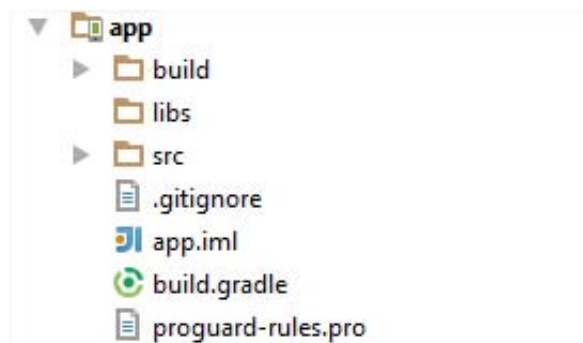
### 2.3 Estructura del directorio del código de la aplicación

A continuación se describe cada uno de los directorios y archivos con los que cuenta el proyecto de la aplicación *Idioms Game*.



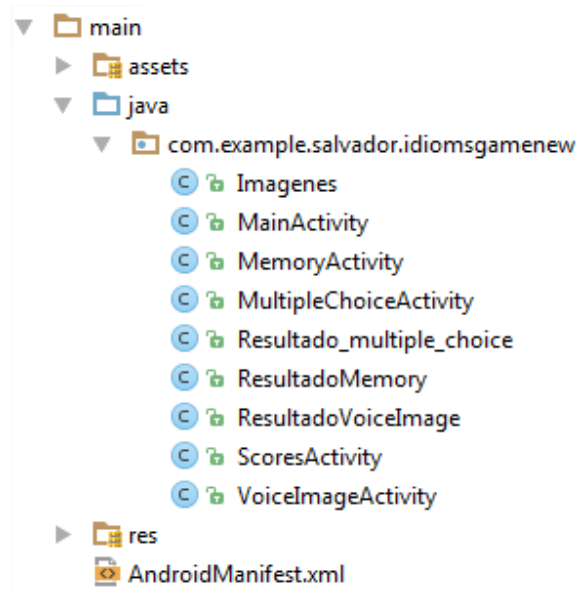
**Figura 1 Estructura del proyecto Android**

La carpeta app es la que contiene todo lo relacionado con el proyecto, es donde se incluyen los archivos necesarios para que la aplicación sea empaquetada. Dentro de ésta se encuentran tres carpetas: build, libs y src (Ver Figura 2).



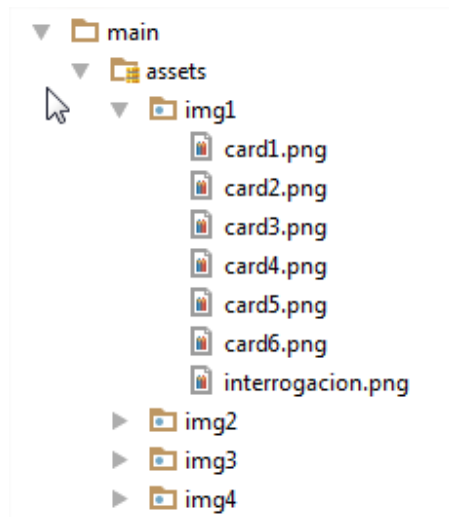
**Figura 2 Estructura de la carpeta app**

Dentro de la carpeta src (Source) se ubica la carpeta main, la cual contiene todos los archivos fuente Java, es decir, las clases necesarias para darle funcionamiento a la aplicación. Ver Figura 3.



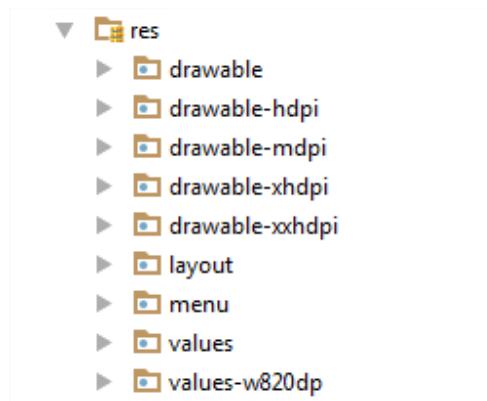
**Figura 3 Contenido de la carpeta main**

Además dentro de la carpeta main, también se encuentra la carpeta assets, ésta contiene subcarpetas con los nombres: img1, img2, img3, img4, las cuales contienen, a su vez, las imágenes que serán mostradas al jugador en *Memory Game*. Ver Figura 4.



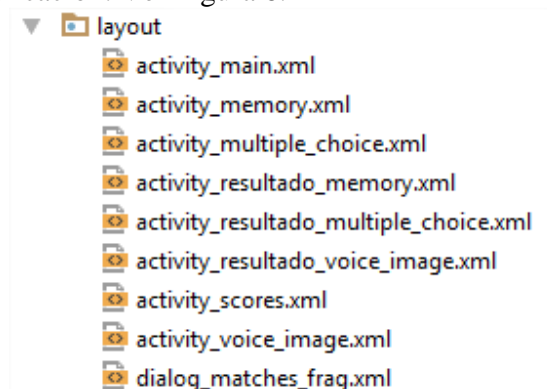
**Figura 4 Estructura de la carpeta assets**

Otra subcarpeta perteneciente a la carpeta main tiene el nombre de `res` (Resources), la cual contiene los recursos del proyecto, en este caso, las imágenes y los estilos para darle presentación a la interfaz gráfica de la aplicación (Ver Figura 5).



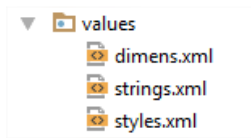
**Figura 5 Estructura de la carpeta res**

Como se observa en la Figura 5, existen también las carpetas `drawable`, `layout`, `menu` y `values`. La carpeta `drawable` contiene las imágenes de la aplicación así como otros elementos gráficos. Se dividen en subcarpetas `/drawable-ldpi`, `/drawable-mdpi` y `/drawable-hdpi` para definir los recursos dependiendo de la resolución de la pantalla del dispositivo. La carpeta `layout` contiene los ficheros XML que representan el diseño de la interfaz gráfica de la aplicación. Ver Figura 6.



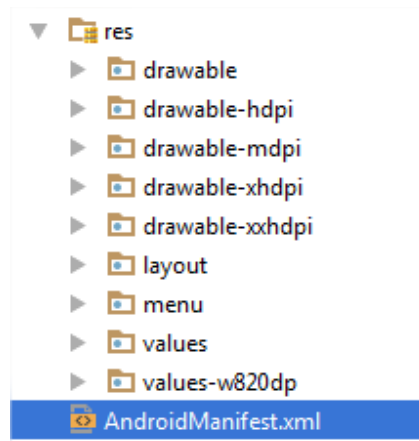
**Figura 6 Estructura de la carpeta layout**

La carpeta `values` contiene otros ficheros XML de recursos de la aplicación, como por ejemplo, cadenas de texto (`strings.xml`), estilos (`styles.xml`), dimensiones (`dimens.xml`), etc. Ver Figura 7.



**Figura 7 Estructura de la carpeta values**

Por último, dentro de la carpeta `res` se encuentra el archivo `AndroidManifest.xml` (Figura 8). Éste es un archivo XML que contiene nodos descriptivos sobre las características de la aplicación. Características como: la versión de SDK usada, los permisos necesarios para ejecutar servicios (en este caso se utiliza el permiso para poder ver el estado de la conexión a internet del dispositivo) y muchos más. En otras palabras el `AndroidManifest` es un panorama de toda la aplicación.



**Figura 8 Fichero AndroidManifest.xml**

### **3 Ambiente de producción del sistema**

#### **3.1 Hardware**

Los aspectos de hardware a tomar en cuenta para la instalación de la aplicación en el dispositivo móvil son:

- Tener pantalla táctil (no es necesario que sea multi-toque)
- Procesador a 1.2 GHZ (mínimo)

#### **3.2 Software**

El software necesario para que la aplicación se instale correctamente en el dispositivo móvil es:

- Sistema operativo Android con la versión 4.0 IceCreamSandwich (mínimo).

### 3.3 Proceso de instalación de la aplicación

Para poder instalar la aplicación correctamente debemos tener en cuenta dos aspectos principales: El primero, asegurarnos de que el dispositivo en el que estará alojada la aplicación sea un Smartphone con sistema operativo Android con versión mínima 4.0 IceCreamSandwich. El segundo, es corroborar que se tiene instalada la aplicación *Google Voice Search* en el dispositivo (gran número de dispositivos la tienen instalada por defecto) para que se puedan responder a las peticiones del usuario cuando éste se encuentre intentando utilizar el reconocimiento de voz; es decir, conectar con los servidores de Google para transformar la voz en texto. En caso de que no se tenga instalada la aplicación *Google Voice Search*, se deben de seguir los siguientes pasos:

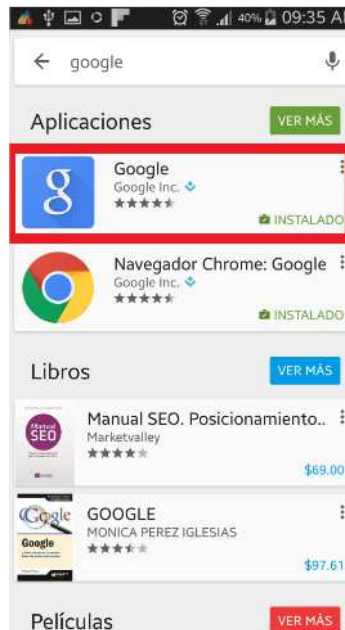
1.- Abrir la *Play Store*<sup>4</sup> y escribir en el buscador: *Google* (Ver Figura 9)



**Figura 9** Pantalla para buscar la aplicación Google

<sup>4</sup> Google Play es una tienda de software en línea desarrollada por Google para los dispositivos con sistema operativo Android.

2.- Seleccionar de la pestaña aplicaciones la primera opción (Ver Figura 10).



**Figura 10** Pantalla que muestra la aplicación que se debe de elegir

3.- Dar click en INSTALAR, e inmediatamente se comenzará la descarga.

4.- Al finalizar, dar click en Abrir (Ver Figura 11).



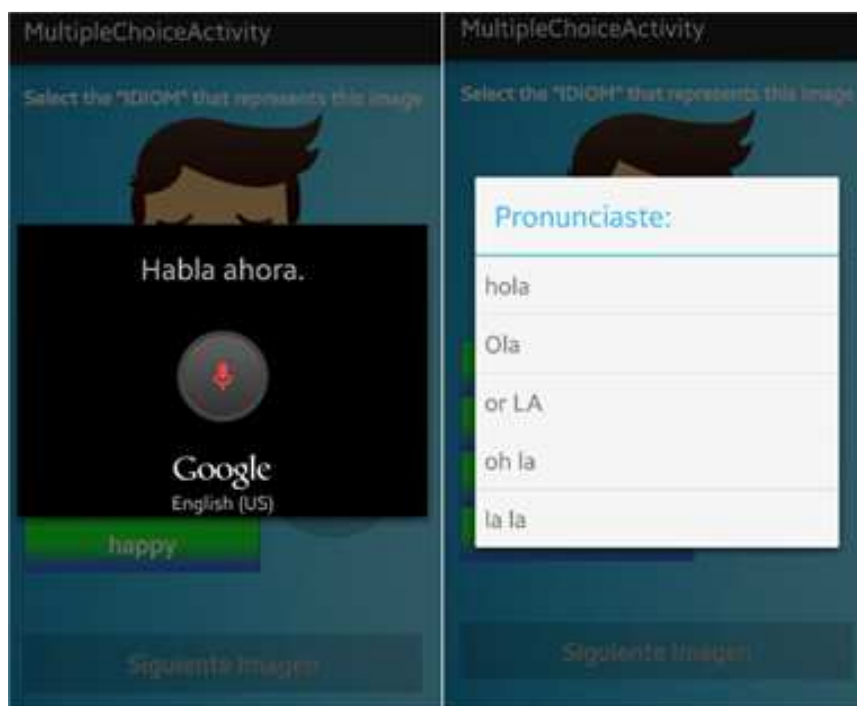
**Figura 11** Pantalla que se muestra al finalizar la descarga

Finalmente, si la aplicación *Google Voice Search* se abrió correctamente quiere decir que la instalación fue exitosa.

Por último, otro de los aspectos a tomar en cuenta antes de la instalación de la aplicación *Idioms Game* es, además de tener acceso a internet, se deberá contar con pantalla táctil (no se requiere que la pantalla del dispositivo sea multitoque).

### 3.4 Pruebas

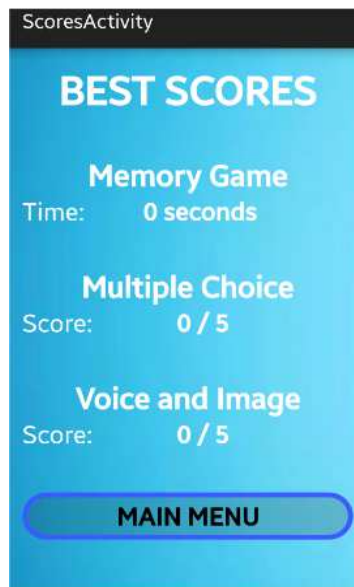
Dentro de la funcionalidad de *Multiple Choice* y *Voice and Image* se encuentra el reconocimiento de voz, por lo tanto, las pruebas se centraron principalmente en estos dos mini-juegos, pues se requería saber si la transformación a texto se realiza correctamente. En la Figura 12 se muestra una prueba de la funcionalidad del reconocimiento de voz, que devuelve al jugador una lista de palabras que se asemejan a lo pronunciado. En la imagen, aparece un cuadro de diálogo con la leyenda: “Habla ahora”, la cual indica que la aplicación se encuentra lista para escuchar y transformar la voz a texto.



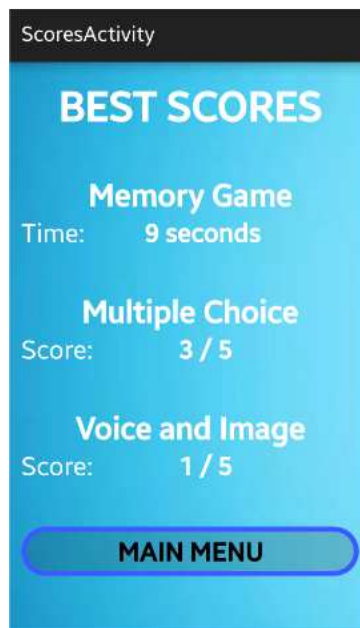
**Figura 12 Funcionamiento de reconocimiento de voz**

No solo se comprobó el funcionamiento de los mini-juegos que hacen uso del reconocimiento de voz. También se probó el funcionamiento de las preferencias compartidas, es decir, se verificó que los puntajes obtenidos por el usuario se guardaban correctamente y no se borraban cada vez que se ejecutaba la aplicación.





**Figura 13** Puntajes al ejecutar por primera vez la aplicación



**Figura 14** Puntajes en la segunda ejecución de la aplicación

Como se observa en la Figura 13, en la pantalla de *Scores*, la primera vez que se ejecuta la aplicación (no se ha jugado ningún mini-juego), el puntaje en cada mini-juego es cero. En la Figura 14 se muestra la pantalla *Scores* después de la segunda ejecución de la aplicación (por lo menos se ha jugado 1 vez cada mini-juego) en donde se observa que ahora el valor del puntaje obtenido en cada mini-juego ha cambiado. Por lo tanto se puede verificar que los puntajes obtenidos por el usuario se guardan correctamente cada vez que el usuario impone una nueva marca.

### 3.5 Problemas durante el desarrollo

Se registraron algunos problemas durante la implementación de cada uno de los mini-juegos. Por ejemplo, en el juego *Memory Game*, tanto la imagen como el texto que se mostraba en ésta eran muy pequeños, por lo que, no se podía percibir correctamente la frase con la que se asociaba la imagen, por lo tanto se concluyó que el problema era meramente de interfaz, entonces lo que se hizo fue cambiar el tamaño tanto de las imágenes como el texto asociado a éstas, para que se pudieran visualizar correctamente en la pantalla del dispositivo del usuario.

Otro de los problemas presentados durante el desarrollo de la aplicación *Idioms Game* fue que tanto en *Multiple Choice* como en *Voice and Image* se quería mostrar al jugador imágenes escogidas de manera aleatoria, cada vez que se oprimía "Siguiete Imagen", inicialmente se utilizó un número aleatorio para mostrar dicha imagen, el inconveniente fue que ese número aleatorio algunas veces se repetía y como consecuencia mostraba la misma imagen, lo cual no era el objetivo que se requería para el correcto funcionamiento del mini-juego; pero finalmente esto se solucionó utilizando API Collections de Java, que permitió ordenar aleatoriamente cada uno de las imágenes y con ayuda de un índice que aumentaba en cada click al botón de "Siguiete Imagen" las imágenes que se mostradas ya no se repetían.

Finalmente, se presentó un problema en la implementación de las Preferencias Compartidas (*SharedPreferences*), pues al obtener un puntaje en alguno de los mini-juegos, éste se guardaba correctamente dentro de la aplicación pero al jugar el mismo mini-juego nuevamente y obtener un nuevo puntaje, éste sustituía al anterior no importando si era una mejor marca o no, lo cual no era correcto ya que lo que se pretende es guardar solamente el puntaje en caso de que éste sea mejor que el anterior. Al analizar la implementación se encontró que el puntaje obtenido en la partida, siempre se guardaba en la aplicación (no se tomaba en cuenta valores obtenidos anteriormente). Para solucionar el problema se tuvo que analizar el código para ubicar donde conviene realizar la sustitución del puntaje anterior con el nuevo puntaje (en caso de que éste fuera mejor), de manera que solo se guarde el mejor puntaje.

## 4 Bibliografía

- [1] JetBrains, «Android Developers,» JetBrains. [En línea]. Available: <https://developer.android.com/sdk/index.html>.
- [2] Oracle, «Java,» Oracle, [En línea]. Available: <http://www.java.com>. [Último acceso: 21 06 2015].
- [3] G. Inc., «Google play,» Google, [En línea]. Available: <https://play.google.com/store/apps/details?id=com.google.android.googlequicksearchbox>. [Último acceso: 21 06 2015].
- [4] A. Developers, «Android Developers,» Google, [En línea]. Available: <http://developer.android.com/reference/android/content/SharedPreferences.html>. [Último acceso: 21 06 2015].
- [5] C. U. Press, «Cambridge Dictionaries Online,» 10 Abril 2015. [En línea]. Available: <http://dictionary.cambridge.org>. [Último acceso: 10 Abril 2015 Abril 2015].
- [6] O. U. Press, «Oxford Dictionaries Language matters,» 10 Abril 2015. [En línea]. Available: <http://www.oxforddictionaries.com/>. [Último acceso: 10 Abril 2015].

**Universidad Autónoma Metropolitana Unidad Azcapotzalco**

División de Ciencias Básicas e Ingeniería

Memoria de Diseño del Proyecto de Integración

Licenciatura en Ingeniería en Computación

Aplicación Android para practicar la pronunciación de los idioms del idioma inglés

Modalidad: Proyecto Tecnológico

Trimestre 2015 Primavera

José Salvador López Balleza

Matrícula: 210304254

jslb\_cafcb10@hotmail.com



Asesor:

Dra. María Lizbeth Gallardo López

Profesor Investigador

Departamento de Sistemas

glizbeth@correo.azc.uam.mx



Fecha de entrega: 22 de julio de 2015

## Índice

1	Introducción .....	3
2	Especificación del sistema .....	4
2.1	Requerimientos funcionales.....	4
2.2	Requerimientos no funcionales.....	5
2.3	Lista de usuarios .....	6
3	Diagrama de casos de uso del sistema .....	7
4	Casos de uso de texto .....	8
4.1	Caso de uso: Jugar <i>Memory Game</i> .....	8
4.2	Caso de uso: Jugar <i>Multiple Choice</i> .....	9
4.3	Caso de uso: Jugar <i>Voice and Image</i> .....	10
4.4	Caso de uso: Ver Puntajes más altos .....	11
5	Diagrama de clases.....	12
5.1	Clase Activity .....	12
5.2	Clase MainActivity .....	13
5.3	Clase MemoryActivity.....	13
5.4	Clase Imágenes .....	13
5.5	Clase MultipleChoiceActivity .....	14
5.6	Clase VoiceImageActivity.....	14
5.7	Clase ScoresActivity.....	14
5.8	Clase ResultadoMemory.....	14
5.9	Clase Resultado_multiple_choice.....	15
5.10	Clase ResultadoVoiceImage .....	15
6	Diagrama de estados.....	16
7	Diagramas de secuencia .....	17
7.1	Diagrama de secuencia: Jugar <i>Memory Game</i> .....	18
7.2	Diagrama de secuencia: Jugar <i>Multiple Choice</i> .....	19
7.3	Diagrama de secuencia: Jugar <i>Voice and Image</i> .....	20
7.4	Diagrama de secuencia: Ver Puntajes más altos.....	21
8	Arquitectura del sistema.....	22

# 1 Introducción

El inglés es considerado una lengua universal, principalmente porque es uno de los idiomas más hablados en el mundo y el más empleado en casi todas las áreas de conocimiento.

Aprender el idioma inglés requiere de práctica en cuanto a la gramática y la pronunciación. Para ello, se están proponiendo gran variedad de aplicaciones móviles, lo cual abre el espectro de posibilidades para practicar este idioma. Es conocido que la memoria humana asocia una imagen a la frase de un idioma, lo cual le ayuda a recordar la frase y situarla en el contexto adecuado.

El presente proyecto tiene la finalidad de entregar al usuario una aplicación para su dispositivo móvil con Sistema Operativo Android que pueda utilizar en cualquier momento y en cualquier lugar para practicar la pronunciación de frases, específicamente los conocidos como *idioms*; esta aplicación recibe el nombre de *Idioms Game*. Los *idioms* se definen como, “Grupo de palabras que poseen un significado diferente al de cada una de las que lo componen” [1], “Grupo de palabras cuyo significado se deduce del contexto de la frase y no del significado individual de cada palabra que lo componen” [2]; por ejemplo, *A bed of roses* literalmente se traduce al español como: una cama de rosas, pero su significado tiene que ver con el contexto cultural del país de habla inglesa; por lo tanto, la traducción al español es: fácil, agradable, placentero. El hecho de que estas frases no tengan una traducción literal al español pueden convertirse en un obstáculo para estudiantes del idioma inglés, por consiguiente, *Idioms Game* servirá para practicar la pronunciación de los *idioms*, y eventualmente el aprendizaje de los mismos, auxiliándose de tres módulos principales. Uno de ellos tiene la tarea de familiarizar al jugador con las relaciones entre las imágenes e *idioms* y los dos módulos restantes tienen el objetivo de reconocer la voz del usuario, transformando la voz a texto escrito.

El presente documento está conformado por ocho secciones. La sección 2 presenta la especificación del sistema, incluye los requerimientos funcionales, requerimientos no funcionales y la lista de usuarios que interactuarán con *Idioms Game*; la sección 3 presenta el diagrama de casos de uso; la sección 4 describe detalladamente los casos de uso de texto para las funcionalidades del sistema; la secciones 5, 6 y 7 muestran el Diagrama de clases, Diagrama de Estados y los Diagramas de secuencia respectivamente. Por último la sección 8 presenta la arquitectura de la aplicación, la cual está basada en el Modelo-Vista-Controlador (MVC).

## 2 Especificación del sistema

Los requerimientos funcionales, no funcionales y la lista de usuarios del sistema se describen en las secciones 2.1, 2.2 y 2.3 respectivamente.

### 2.1 Requerimientos funcionales

A continuación se listan los requerimientos funcionales con los que contará el sistema, así como una breve descripción de cada uno de ellos.

**Iniciar Menú Principal.** Para iniciar la ejecución de *Idioms Game*, la aplicación debe permitir:

1. Elegir el inicio de *Memory Game*.
2. Elegir el inicio de *Multiple Choice*.
3. Elegir el inicio de *Voice and Image*.
4. Elegir la opción para ver los mejores puntajes.

**Jugar *Memory Game*.** Para jugar el mini-juego *Memory Game*, la aplicación debe:

1. Ejecutar un cronómetro, iniciado desde cero que incrementará en uno cada segundo.
2. Permitir elegir una tarjeta<sup>1</sup> con la imagen del signo de interrogación.
3. Mostrar una imagen con un texto que identifica el *idiom* con la que se relaciona (cada que se elija una tarjeta con la imagen del signo de interrogación).
4. En caso de encontrar dos imágenes iguales, dejarlas visibles.
5. En caso de no encontrar dos imágenes iguales, voltear ambas tarjetas hacia su estado inicial, es decir, se debe mostrar nuevamente la tarjeta con la imagen con el signo de interrogación.
6. Detener el cronómetro al encontrar seis pares de imágenes iguales.
7. Guardar el tiempo transcurrido en el cronómetro.
8. Habilitar un botón “Continuar”.
9. Direccionar a una nueva pantalla “Scores” para observar el tiempo transcurrido en la ejecución del mini- juego, así como el mejor tiempo obtenido en la totalidad de todas las partidas jugadas.
10. Habilitar dos botones: “Play Again” y “Main Menu”.

**Jugar *Multiple Choice*.** Para jugar el mini-juego *Multiple Choice*, la aplicación debe:

1. Mostrar una imagen representativa a un *idiom*.
2. Activar el micrófono del dispositivo.
3. Convertir la voz del jugador a texto escrito.

---

<sup>1</sup> Se conocerá como tarjeta a las imágenes representadas con un signo de interrogación en el mini-juego *Memory Game*

4. Mostrar un mensaje para indicar si se obtuvo respuesta correcta o incorrecta.
5. Después de contestar a diez imágenes mostradas, direccionar a una pantalla “Scores” para observar el número de aciertos obtenidos en la ejecución del mini-juego.
6. Habilitar dos botones: “Play Again” y “Main Menu”.

**Jugar *Voice and Image*.** Para jugar el mini-juego *Voice and Image*, la aplicación debe:

1. Mostrar una imagen representativa a un idioma.
2. Activar el micrófono del dispositivo.
3. Convertir la voz del jugador a texto escrito.
4. Mostrar un mensaje para indicar si se obtuvo respuesta correcta o incorrecta.
5. Después de contestar a diez imágenes mostradas, direccionar a una pantalla “Scores” para observar el número de aciertos obtenidos en la ejecución del mini-juego.
6. Habilitar dos botones: “Play Again” y “Main Menu”.

**Ver puntajes más altos.** Para poder observar los récords impuestos en cada uno de los mini-juegos, la aplicación debe:

1. Mostrar el mejor tiempo obtenido en *Memory Game*.
2. Mostrar el mayor número de aciertos obtenidos en *Multiple Choice*.
3. Mostrar el mayor número de aciertos obtenidos en *Voice and Image*.
4. Permitir elegir la opción “Main Menu” para regresar al menú principal.

## 2.2 Requerimientos no funcionales

A continuación se describirán los requerimientos no funcionales para la aplicación a construir.

### **Usabilidad.**

- La aplicación sólo funciona a partir de la versión 4.0 de Android.
- La aplicación sólo funciona con conexión a internet, ya sea por vía WIFI o por plan de datos.
- El dispositivo debe de tener preinstalada la aplicación *Google App* para *Android*.

### **Rendimiento.**

- El sistema debe responder a las operaciones solicitadas en un máximo de 10 segundos.

### **Interfaz.**

- El dispositivo debe de disponer de una interfaz con pantalla multi táctil.



## 2.3 Lista de usuarios

- **Jugador:** Es el encargado de manipular completamente la aplicación *Idioms Game*.  
El jugador realizará acciones como:
  - 1) Jugar *Memory Game*
  - 2) Jugar *Multiple Choice*
  - 3) Jugar *Voice and Image*
  - 4) Ver *Scores*

### 3 Diagrama de casos de uso del sistema

Los casos de uso relacionados con las funcionalidades que proporcionará la aplicación se muestra en la Figura 1.

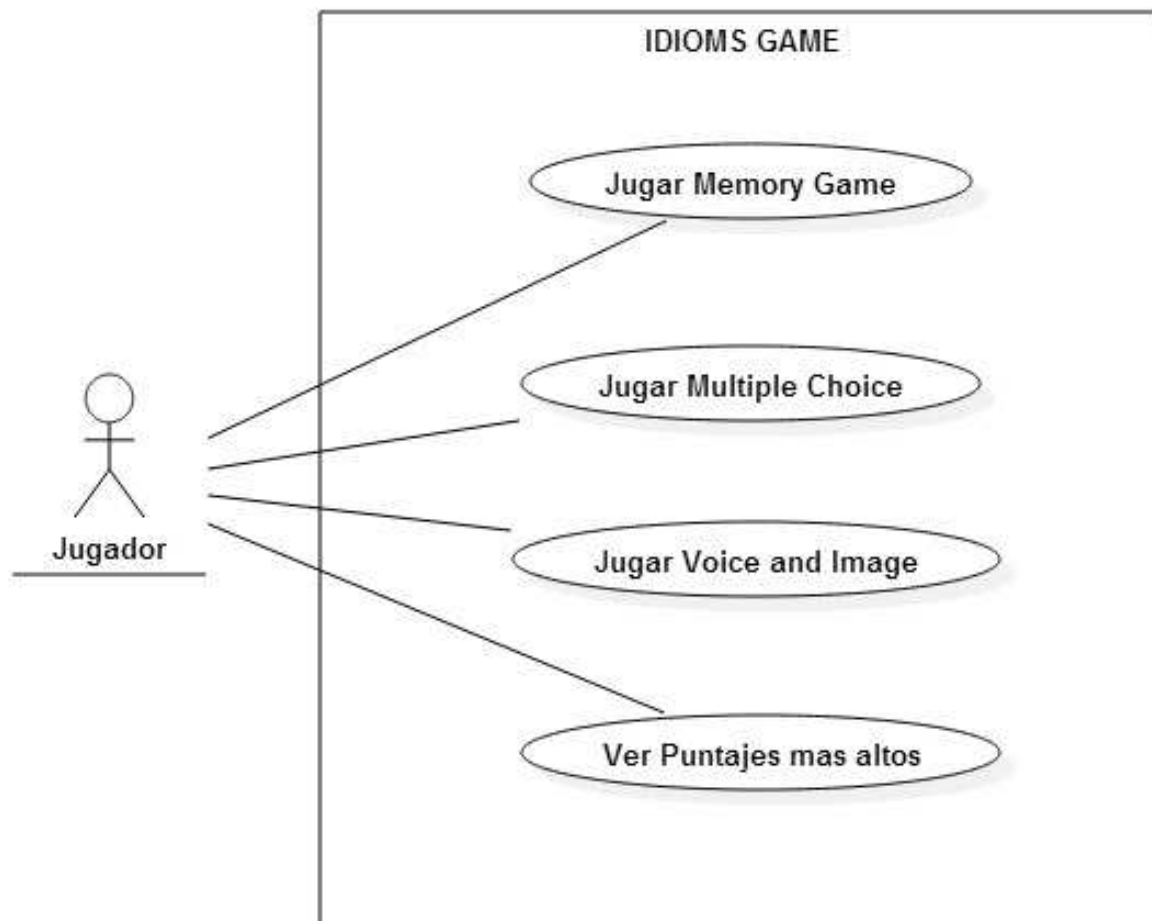


Figura 1 Diagrama de casos de uso

## 4 Casos de uso de texto

A continuación se presentan los casos de uso de texto, con los cuales se tendrá una mayor comprensión sobre el funcionamiento de la aplicación.

### 4.1 Caso de uso: Jugar *Memory Game*

**Descripción:**

Esta operación permitirá conocer la dinámica de un juego para ejercitar la memoria. El jugador va a conocer la relación entre las imágenes mostradas y el *idiom* que le corresponde, pues la imagen trae consigo un rótulo con el *idiom* al que se asocia.

**Actores y sus intereses:**

El actor principal (Jugador) está interesado en comenzar con la ejecución del juego *Memory Game* para conocer las imágenes con las que cuenta la aplicación, así como el *idiom* con el que se asocia.

**Disparador:**

La operación inicia cuando el Jugador, desde el menú principal, elige la opción *Memory Game*.

**Pre-condición:**

El jugador seleccionó el ícono que lanza la aplicación.

**Post-condición:**

Al terminar el caso de uso, el tiempo que le tomó al jugador encontrar los seis pares quedará registrado en un archivo dentro de la aplicación.

**Flujo principal:**

- 1) El jugador selecciona, desde el menú principal, la opción *Memory Game*.
- 2) La aplicación coloca aleatoriamente las posiciones en que aparecerán las 12 imágenes.
- 3) La aplicación muestra un cronómetro, el cual inicia desde cero e incrementa en 1 cada segundo transcurrido. También presenta 12 imágenes con un signo de interrogación (representan la cara de abajo de una tarjeta) acomodadas en 3 columnas con 4 imágenes cada una.
- 4) El jugador da click a una imagen cara abajo.
- 5) La aplicación muestra una imagen que representa la cara de arriba que el actor principal oprimió.
- 6) El actor principal da click a una nueva imagen cara abajo.
- 7) La aplicación muestra una nueva imagen que representa la cara de arriba.
- 8) La aplicación verifica si las 2 imágenes mostradas son las mismas, en caso positivo, las imágenes se quedan cara arriba, en caso contrario, las imágenes regresan a su estado original con la cara hacia abajo.
- 9) Se repite el proceso desde el paso 2 hasta el paso 6.
- 10) El jugador encuentra los seis pares posibles.
- 11) La aplicación detiene el cronómetro y habilita un botón "Continuar".
- 12) El jugador selecciona "Continuar".
- 13) La aplicación muestra una pantalla con las leyendas: *YOUR TIME* que representa el tiempo que tardó el jugador en encontrar los 6 pares, *BEST TIME* que representa el mejor tiempo de entre todas las veces que se ha jugado y además muestra dos botones: "Play Again" y

“Main Menu”.

14) El jugador selecciona “Play Again”, el proceso se repite desde el paso 2.

**Flujo alternativo:**

1) El jugador selecciona “Main Menu”, la aplicación dirige a la pantalla con el menú principal.

## 4.2 Caso de uso: Jugar *Multiple Choice*

**Descripción:**

En este caso de uso se presenta el funcionamiento del mini juego “Multiple Choice”.

**Actores y sus intereses:**

→ El actor principal (Jugador) está interesado en comenzar con la dinámica del juego “Multiple Choice”.

**Disparador:**

La operación inicia cuando el Jugador, desde el menú principal, elige la opción “Multiple Choice”.

**Pre-condición:**

El jugador jugó *Memory Game* para conocer la correspondencia entre la imagen y el *idiom* con el que se asocia.

**Post-condición:**

Al terminar el caso de uso, el número de aciertos obtenido por el jugador quedará registrado en un archivo dentro de la aplicación.

**Flujo Principal:**

- 1) El jugador selecciona, desde el menú principal, la opción “Multiple Choice”.
- 2) La aplicación muestra una pantalla con dos botones: “Comenzar” y otro con una imagen de un micrófono.
- 3) El jugador da click en el botón “Comenzar”.
- 4) La aplicación muestra una imagen y 4 cuadros de texto con frases que representan las posibles respuestas que representan a dicha imagen en donde sólo una es la correcta.
- 5) El jugador observa las 4 posibles respuestas, elige una de ellas y da click en el botón con la imagen del micrófono.
- 6) La aplicación muestra una ventana con la leyenda “Habla ahora”, que indica que el micrófono del dispositivo está en espera de escuchar alguna frase.
- 7) El jugador pronuncia la frase que eligió anteriormente.
- 8) La aplicación despliega una lista con las frases que coinciden con lo pronunciado.
- 9) El jugador selecciona la frase que pronunció.
- 10) La aplicación verifica si la frase pronunciada coincide con la que se tiene como correcta, en caso positivo, muestra un mensaje indicando que fue correcto y el número de aciertos aumenta en uno; en caso contrario, muestra un mensaje indicando que lo pronunciado es incorrecto.
- 11) El jugador da click en el botón “Siguiente Imagen”.
- 12) Se repite el proceso desde el paso 4 al paso 11 hasta que pasen 10 imágenes.
- 13) La aplicación muestra una pantalla con las leyendas: *YOUR SCORE* que representa el número de aciertos obtenidos en la partida actual, *BEST SCORE* que representa el mayor

número de aciertos obtenidos de entre todas las veces que se ha jugado y además muestra dos botones: “Play Again” y “Main Menu”.

14) El jugador selecciona “Play Again”, el proceso se repite desde el paso 2.

**Flujo alternativo:**

15) El jugador selecciona “Main Menu”, la aplicación dirige a la pantalla con el menú principal.

### 4.3 Caso de uso: Jugar *Voice and Image*

**Descripción:**

En este caso de uso se presenta el funcionamiento del mini juego “Voice and Image”.

**Actores y sus intereses:**

El actor principal (Jugador) está interesado en comenzar con la dinámica del juego “Voice and Image”.

**Disparador:**

La operación inicia cuando el Jugador, desde el menú principal, elige la opción “Voice and Image”.

**Pre-condición:**

El jugador jugó *Memory Game* para conocer la correspondencia entre la imagen y el *idiom* con el que se asocia.

**Post-condición:**

Al terminar el caso de uso, el número de aciertos obtenido por el jugador quedará registrado en un archivo dentro de la aplicación.

**Flujo Principal:**

- 1) El jugador selecciona, desde el menú principal, la opción “Multiple Choice”.
- 2) La aplicación muestra una pantalla con dos botones: “Comenzar” y otro con una imagen de un micrófono.
- 3) El jugador da click en el botón “Comenzar”.
- 4) La aplicación muestra una imagen y habilita el botón con la imagen del micrófono.
- 5) El jugador da click en el botón que contiene la imagen de un micrófono.
- 6) La aplicación muestra una ventana con la leyenda “Habla ahora”, que indica que el micrófono del dispositivo está en espera de escuchar alguna frase.
- 7) El jugador pronuncia la frase que cree que es la que corresponde con la imagen mostrada.
- 8) La aplicación despliega una lista con las frases que coinciden con lo pronunciado.
- 9) El jugador selecciona la frase que pronunció.
- 10) La aplicación verifica si la frase pronunciada coincide con la que se tiene como correcta, en caso positivo, muestra un mensaje indicando que fue correcto y el número de aciertos aumenta en uno; en caso contrario, muestra un mensaje indicando que lo pronunciado es incorrecto.
- 11) El jugador da click en el botón “Siguiente Imagen”.
- 12) Se repite el proceso desde el paso 4 al paso 11 hasta que pasen 10 imágenes.
- 13) La aplicación muestra una pantalla con las leyendas: *YOUR SCORE* que representa el número de aciertos obtenidos en la partida actual, *BEST SCORE* que representa el mayor

número de aciertos obtenidos de entre todas las veces que se ha jugado y además muestra dos botones: “Play Again” y “Main Menu”.

14) El jugador selecciona “Play Again”, el proceso se repite desde el paso 2.

**Flujo alternativo:**

15) El jugador selecciona “Main Menu”, la aplicación dirige a la pantalla con el menú principal.

#### 4.4 Caso de uso: Ver Puntajes más altos

**Descripción:**

Esta operación permitirá al jugador observar cuáles han sido sus mejores puntajes en los tres mini juegos: *Memory Game*, *Multiple Choice* y *Voice and Image*.

**Actores y sus intereses:**

El actor principal (Jugador) está interesado en observar cuáles son sus mejores marcas establecidas a lo largo de todas las partidas jugadas en cada uno de los mini juegos.

**Disparador:**

La operación inicia cuando el Jugador, desde el menú principal, elige la opción “Scores”.

**Pre-condición:**

Para que se cumpla esta operación, es necesario que el jugador haya jugado por lo menos una vez cada uno de los mini-juegos.

**Post-condición:**

Al terminar el caso de uso, el jugador observará los mejores puntajes que ha obtenido en cada uno de los 3 mini juegos, para que de esta manera tenga la pauta para establecer nuevas marcas.

**Flujo principal:**

- 1) El jugador selecciona, desde el menú principal, la opción “Scores”.
- 2) La aplicación muestra la pantalla “*BEST SCORES*”, en donde se observa el menor tiempo que se ha obtenido en *Memory Game*, el mayor número de aciertos obtenidos tanto en “Multiple Choice” como en “Voice and Image” y el botón “Main Menu”.
- 3) El jugador observa sus mejores puntajes y da click en el botón “Main Menu”.
- 4) La aplicación dirige a la pantalla con el menú principal.

## 5 Diagrama de clases

Los casos de uso de texto de la aplicación *Idioms Game* permitieron identificar las clases que representan el funcionamiento de cada uno de los mini juegos. A continuación, en la Figura 2 se observa el diagrama de clases de software que conforman la aplicación.

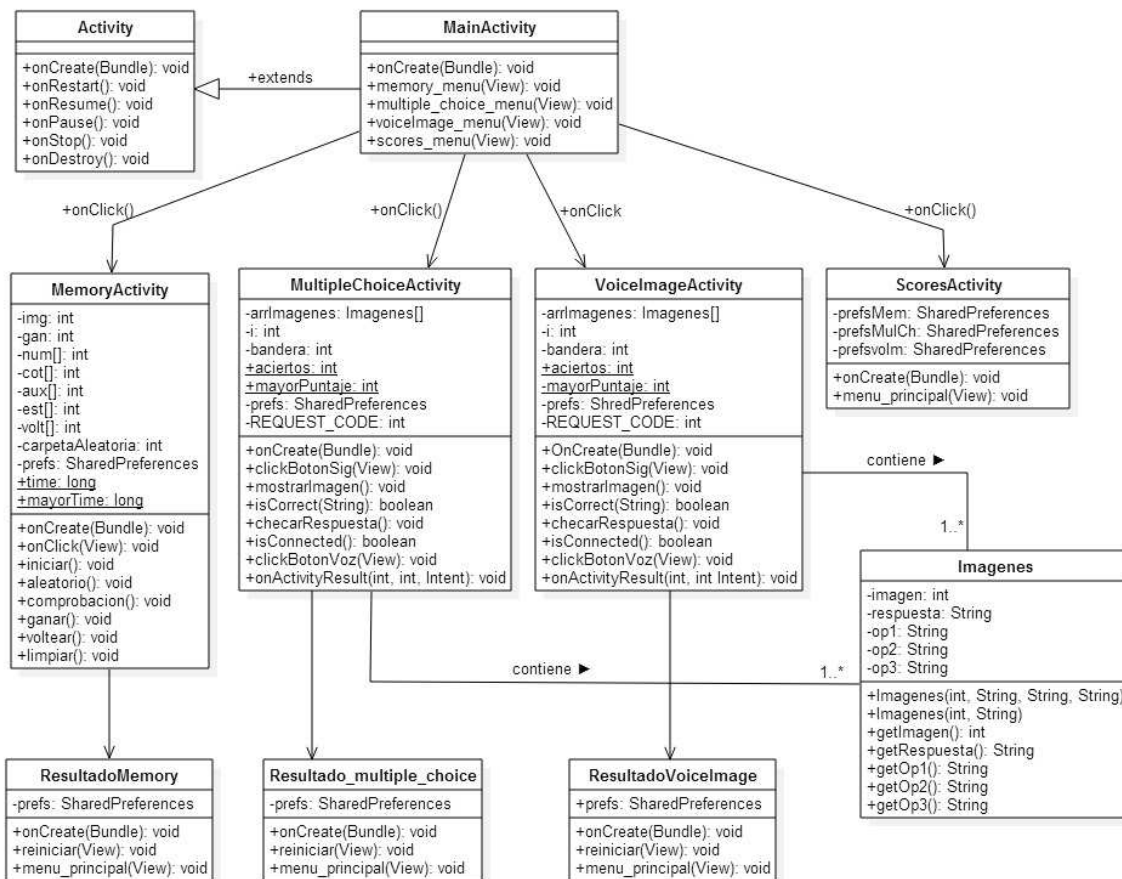


Figura 2 Diagrama de clases de la aplicación

A continuación, se listan cada una de las clases con las que cuenta la aplicación, así como una descripción sobre las tareas que realiza y sus atributos.

### 5.1 Clase Activity

Una *Activity* es una ventana que contiene la interface de usuario de la aplicación. El objetivo principal de una *activity* es interactuar con el usuario. Desde el momento que una *activity* aparece en la pantalla hasta el momento que se oculta, pasa por varias etapas, conocido como el ciclo de vida de una *activity* [3].

Esta clase define los siguientes eventos:

- `onCreate()` – Llamado cuando la activity es creada.
- `onStart()` – Llamado cuando la activity empieza a ser visible al usuario.
- `onPause()` – Llamado cuando la activity actual va a ser pausada y la activity previa será reanudada.
- `onStop()` – Llamado cuando la activity ya no es visible para el usuario.
- `onDestroy()` – Llamado antes de que la activity sea destruida por el sistema (también manualmente o por el sistema para conservar memoria).
- `onRestart()` – Llamado cuando la activity ha sido detenido y es reiniciada la activity de nuevo.

## 5.2 Clase MainActivity

La clase `MainActivity` es la encargada de administrar la pantalla que contiene el menú principal de la aplicación, esta clase hereda de `Activity` el método `onCreate()` y además define los métodos necesarios para poder navegar hacia las pantallas tanto de los 3 mini-juegos como de los *Scores*.

## 5.3 Clase MemoryActivity

La clase `MemoryActivity` hereda de `Activity` el método `onCreate()`, tiene la tarea de desplegar la interfaz gráfica, además aquí se implementa la lógica del juego *Memory Game*, para esto se definieron los atributos: `img`, `cant`, `gan`, además de los arreglos `num`, `cot`, `aux` y `est`, los cuales en conjunto mantendrán la consistencia de la lógica del juego pues con estos atributos se controla que las imágenes no se repitan más de dos veces, también se mantiene una variable para tener el control de cuantos pares se han conseguido y de esta manera saber hasta cuando se termina el juego. Por último, se tienen también como atributos a: `carpetaAleatoria`, que tomará aleatoriamente una carpeta que contiene un banco con 6 imágenes para que cada vez que se ejecute el juego se muestren nuevas imágenes; `time` y `mayorTime`, los cuales son atributos que sirven como auxiliares para llevar el control del tiempo empleado en terminar el juego, así como validar si este tiempo es un nuevo récord.

## 5.4 Clase Imagenes

En la clase `Imagenes` se definen dos constructores para manejar las imágenes que forman parte de las posibles respuestas (`op1`, `op2` y `op3`); donde solo una de ellas es la correcta (respuesta). Asimismo, esta clase cuenta con métodos `set` y `get`, con los cuales, se pueden definir y recuperar los atributos: `imagen`, `respuesta`, `op1`, `op2` y `op3` para cada una de las instancias que sean creadas de esta clase, a partir de los juegos: `MultipleChoiceActivity` y `VoiceImageActivity`.



## 5.5 Clase `MultipleChoiceActivity`

La clase `MultipleChoiceActivity` hereda de `Activity` e implementa toda la lógica del juego *Multiple Choice*, se relaciona directamente con la clase `Imagenes` ya que es necesario obtener tanto la imagen a desplegar como su respuesta con ayuda de un arreglo de imágenes, el cual es definido como un atributo llamado `arrImagenes` que será inicializado con ayuda de uno de los dos constructores de la clase `Imagenes`, además de éste, se definieron los siguientes atributos: `i`, `bandera`, `aciertos`, `mayorPuntaje` y `REQUEST_CODE`, en donde el atributo `aciertos` mantiene el puntaje obtenido en el mini-juego.

## 5.6 Clase `VoiceImageActivity`

Al igual que las clases `MemoryActivity` y `MultipleChoiceActivity`, `VoiceImageActivity` hereda de `Activity`. Esta clase tiene relación directa con la clase `Imagenes`, pues al igual que `MultipleChoiceActivity`, tiene que obtener tanto la imagen a desplegar como la respuesta con que se asocia; por lo tanto, es aquí donde se define la lógica que seguirá el juego *Voice and Image*: Desplegará la imagen, activará el micrófono para transformar la voz del jugador en texto, validará el texto obtenido con la respuesta que se tiene como correcta, aumentará y guardará en un archivo, dentro de la aplicación, el número de aciertos en caso de ser necesario.

## 5.7 Clase `ScoresActivity`

La clase `ScoresActivity` hereda de `Activity`, tiene relación directa con `MainActivity`, pues de no ser por la clase principal, el contenido de ésta clase no podría ser visualizado. La tarea principal de `ScoresActivity` es recuperar los mejores puntajes obtenidos por el jugador en cada uno de los 3 mini-juegos con ayuda de los atributos `prefsMem`, `prefsMulCh` y `prefsvoIm`.

## 5.8 Clase `ResultadoMemory`

La clase `ResultadoMemory` tiene relación directa con `MemoryActivity`. `ResultadoMemory` recupera el puntaje obtenido por el jugador en su atributo `prefs`; el puntaje debió ser registrado previamente por la clase donde se implementa la lógica del juego *Memory Game*.

## **5.9 Clase Resultado\_multiple\_choice**

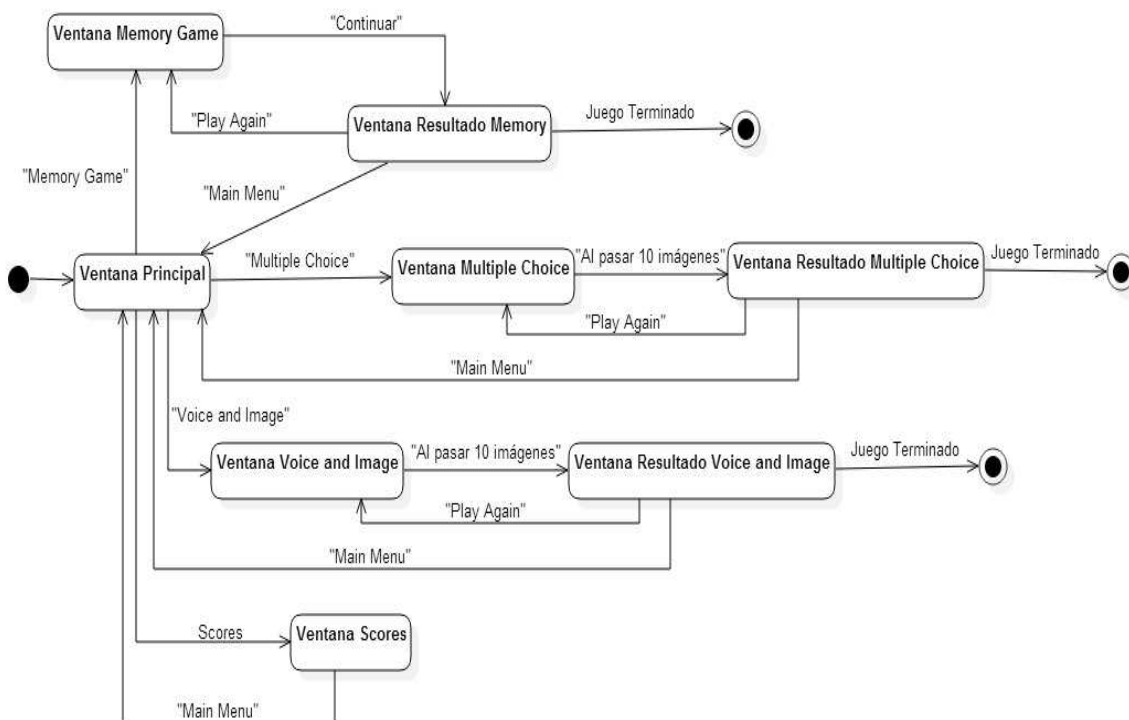
La clase `Resultado_multiple_choice` se relaciona directamente con `MultipleChoiceActivity`. `Resultado_multiple_choice` tiene la tarea de recuperar el puntaje obtenido por el jugador en el atributo `prefs`; el puntaje debió ser registrado previamente por la clase `MultipleChoiceActiviy`, que es la clase en donde se implementa la lógica del juego *Multiple Choice*.

## **5.10 Clase ResultadoVoiceImage**

La clase `ResultadoVoiceImage` tiene relación directa con la clase `VoiceImageActivity`. `ResultadoVoiceImage` tiene la finalidad de recuperar el puntaje obtenido por el jugador en su atributo `prefs`, el cual debió ser almacenado previamente por la clase `VoiceImageActivity`, en la cual se desarrolla la lógica del juego *Voice and Image*.

## 6 Diagrama de estados

En el diagrama de la Figura 3 se observa la secuencia entre las diferentes ventanas de la aplicación, desde el punto inicial hasta el punto final, indicando los diferentes caminos de decisión que se pueden seguir durante la ejecución de la aplicación. Gracias a este diagrama es posible comprender mejor las transiciones de los mini-juegos.



**Figura 3 Diagrama de estados de la aplicación**

Al arrancar la aplicación se muestra la pantalla del menú principal, donde el jugador puede tomar una de cuatro decisiones: Jugar *Memory Game*, jugar *Multiple Choice*, jugar *Voice and Image* o visualizar los mejores puntajes. Cuando el jugador concluye un juego, la aplicación muestra una pantalla donde se visualiza tanto el puntaje obtenido en esa oportunidad, como el mejor puntaje obtenido en todas las veces que se ha jugado; en esta pantalla se pueden tomar dos decisiones más: “Play Again” y “Main Menu”. La primera reinicia el juego y la segunda regresa al menú principal de la aplicación.

Mientras la aplicación está en ejecución, es necesario detectar el evento de pulsación de pantalla para iniciar el mini juego que desee el jugador. Al detectarse la pulsación, se visualiza la pantalla correspondiente a la opción del jugador. Dicho evento se considera como externo, pues es una acción provocada por el usuario. Por ejemplo, en el mini juego *Memory Game* una pulsación de pantalla sobre la imagen de una carta provoca voltearla;

otro ejemplo, en *Multiple Choice*, una pulsación sobre la imagen del micrófono activa el micrófono del dispositivo.

Por otro lado, los eventos internos de la aplicación dependerán del estado de la partida. Por ejemplo, en el juego *Memory Game*, al desplegarse 6 pares de imágenes la partida finaliza, pues al conseguir todos los pares de imágenes posibles en el juego, la aplicación dirige inmediatamente a la pantalla en donde se observa el tiempo obtenido para consumir el objetivo del juego.

## 7 Diagramas de secuencia

A continuación se presenta los diagramas de secuencia de los casos de uso con los que cuenta *Idioms Game*. Cada diagrama de secuencia representa a un caso de uso, pero además se añadió un diagrama de secuencia general.

En la Figura 4, se muestra el diagrama de secuencia general, el cual muestra el flujo que sigue la aplicación desde que se selecciona tanto el inicio de cada uno de los mini juegos como cuando se selecciona la pantalla para observar los mejores puntajes.

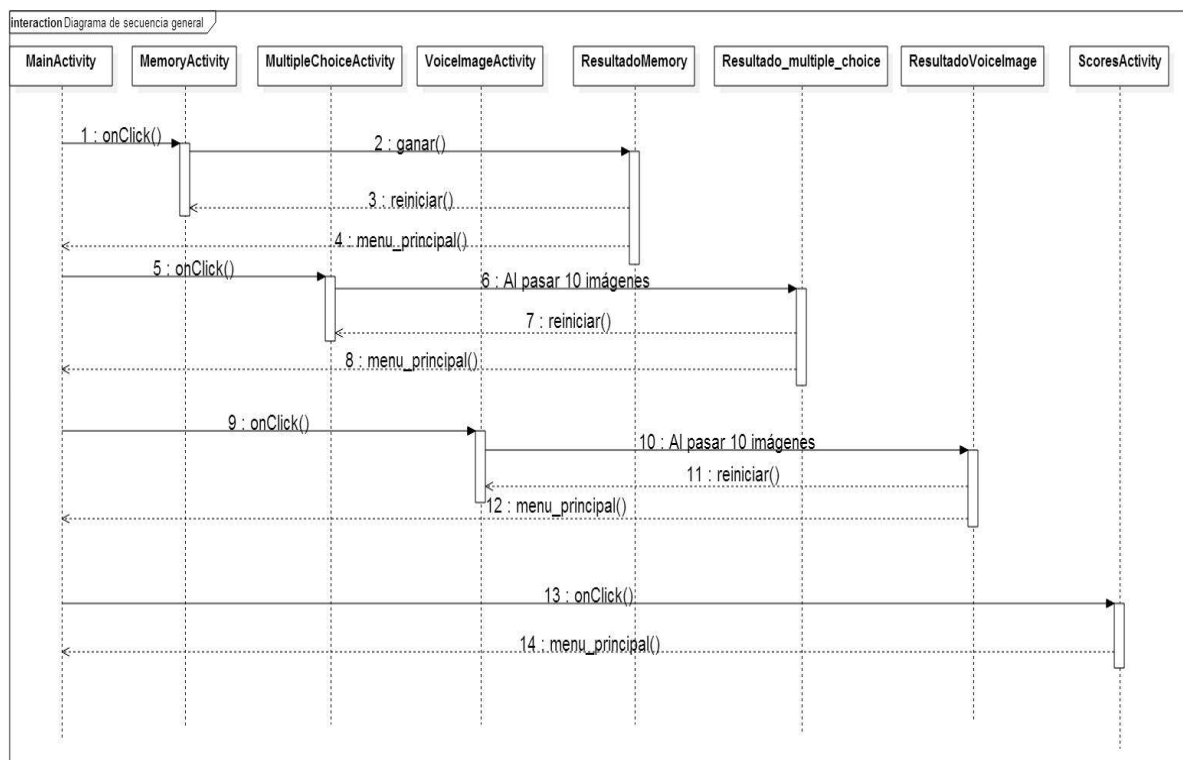


Figura 4 Diagrama de secuencia general

### 7.1 Diagrama de secuencia: Jugar *Memory Game*

El jugador elige la opción *Memory Game*, abre la ventana *Memory Game*, el cronómetro empieza con el conteo, el jugador elige 2 cartas, la clase *MemoryActivity* valida si las cartas elegidas son iguales, si es así, deja las imágenes visibles, en otro caso, las oculta. Se sigue con la secuencia hasta encontrar 6 pares de imágenes iguales, en este caso, se llama al método *ganar()*. El jugador elige la opción *Continuar*, de esta forma se muestra la ventana *Resultado* con el puntaje obtenido recuperado desde la clase *ResultadoMemory*.

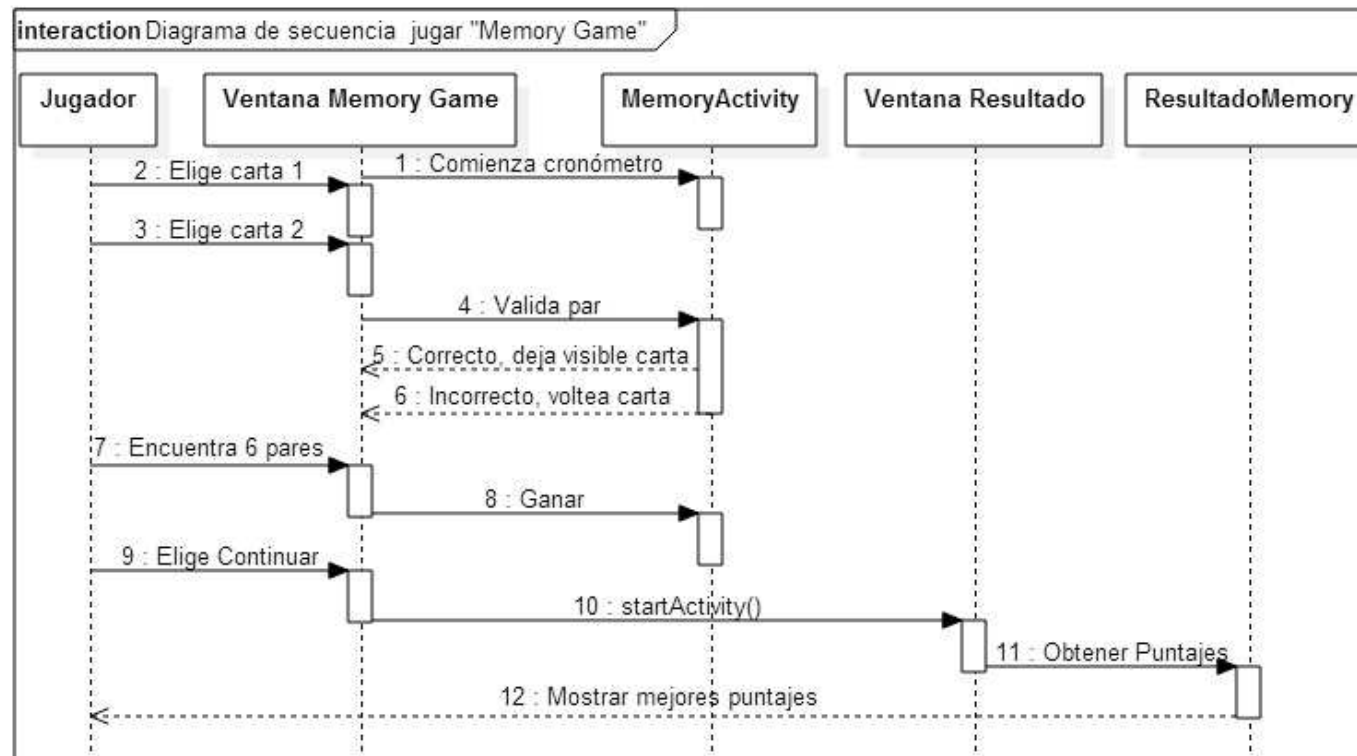


Figura 5 Diagrama de secuencia. Jugar *Memory Game*

## 7.2 Diagrama de secuencia: Jugar *Multiple Choice*

El jugador elige la opción *Multiple Choice*, abre la ventana *Multiple Choice*, presiona la opción Iniciar, la clase *MultipleChoiceActivity*, de esta manera se obtienen las imágenes y se inicializa el arreglo de imágenes que se van a mostrar. El jugador selecciona el botón *Voz* y pronuncia la frase, la clase *MultipleChoiceActivity* procesa la voz y la convierte en texto, se valida si lo pronunciado es correcto. Sigue con la secuencia hasta pasar 10 imágenes; como consecuencia, se invoca el método `startActivity()` que abre la ventana *Resultado* con el puntaje obtenido, recuperado desde la clase *Resultado\_multiple\_choice*.

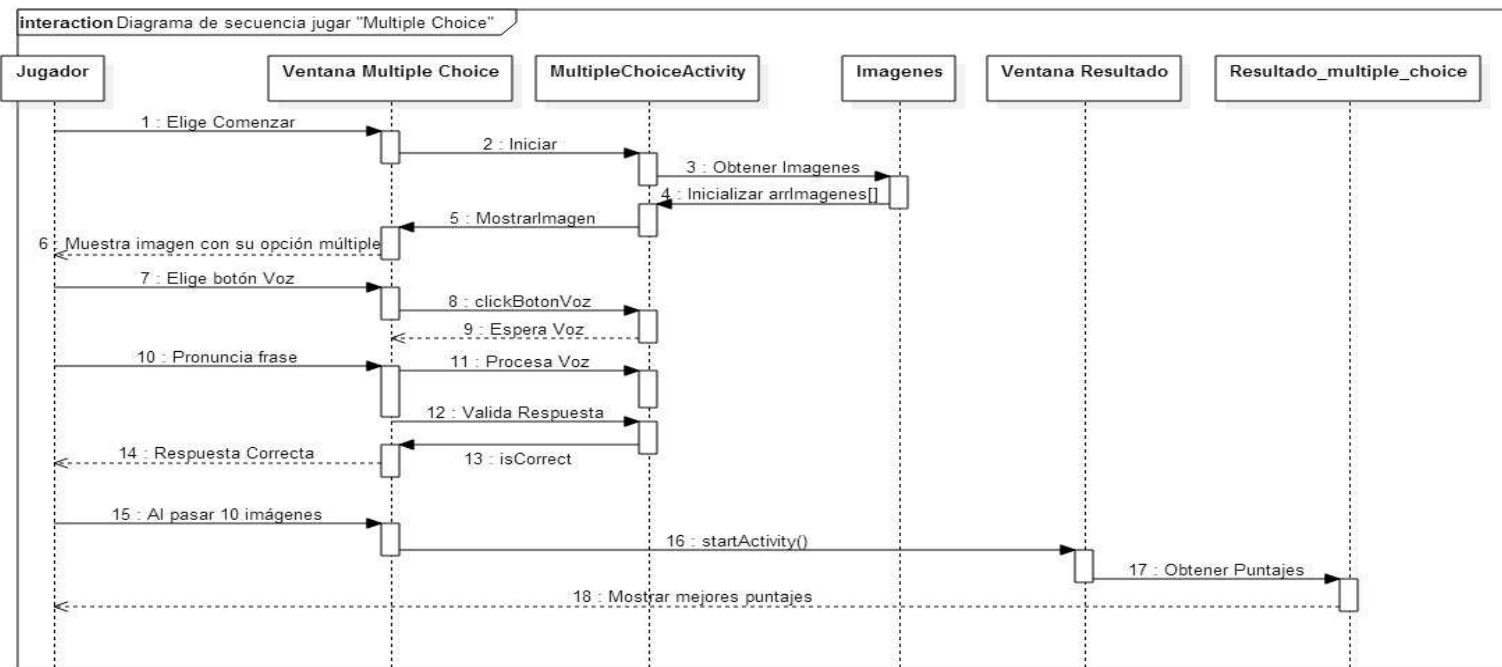


Figura 6 Diagrama de secuencia. Jugar *Multiple Choice*

### 7.3 Diagrama de secuencia: Jugar *Voice and Image*

El jugador elige la opción *Voice and Image*, abre la ventana *Voice and Image*, presiona la opción *Iniciar*, la clase *VoiceImageActivity*, de esta manera se obtienen las imágenes y se inicializa el arreglo de imágenes que se van a mostrar. El jugador selecciona el botón *Voz* y pronuncia la frase, la clase *VoiceImageActivity* procesa la voz y la convierte en texto, se valida si lo pronunciado es correcto. Sigue con la secuencia hasta pasar 10 imágenes, como consecuencia, se invoca el método `startActivity()` que abre la ventana *Resultado* con el puntaje obtenido, recuperado desde la clase *ResultadoVoiceImage*.

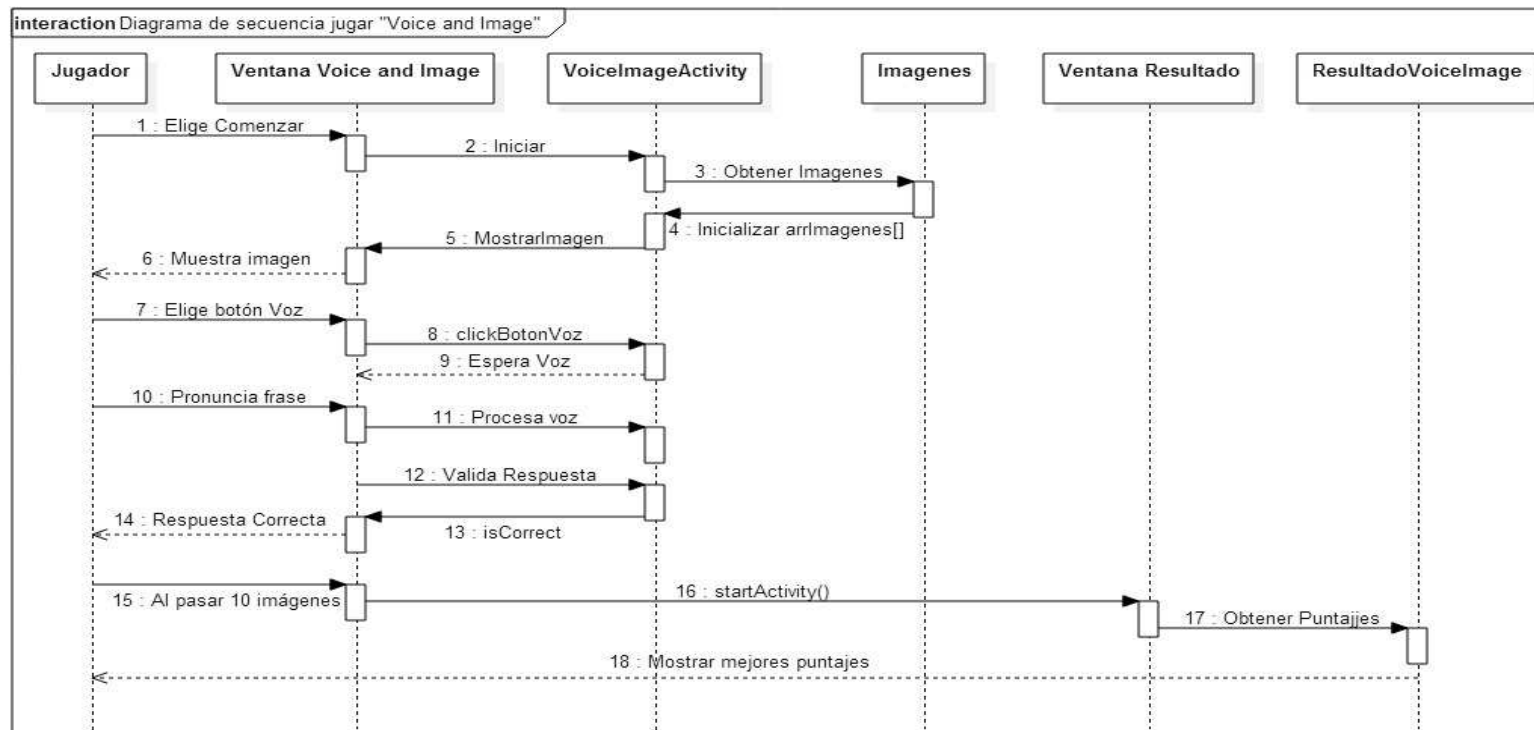


Figura 7 Diagrama de secuencia. Jugar *Voice and Image*

#### 7.4 Diagrama de secuencia: Ver Puntajes más altos

El jugador selecciona la opción *Scores*, se muestra la ventana *Scores* donde se muestran los mejores puntajes de cada uno de los mini-juegos (*Memory Game*, *Multiple Choice*, *Voice and Image*) que fueron previamente almacenados y se recuperan desde la clase *ScoresActivity*.

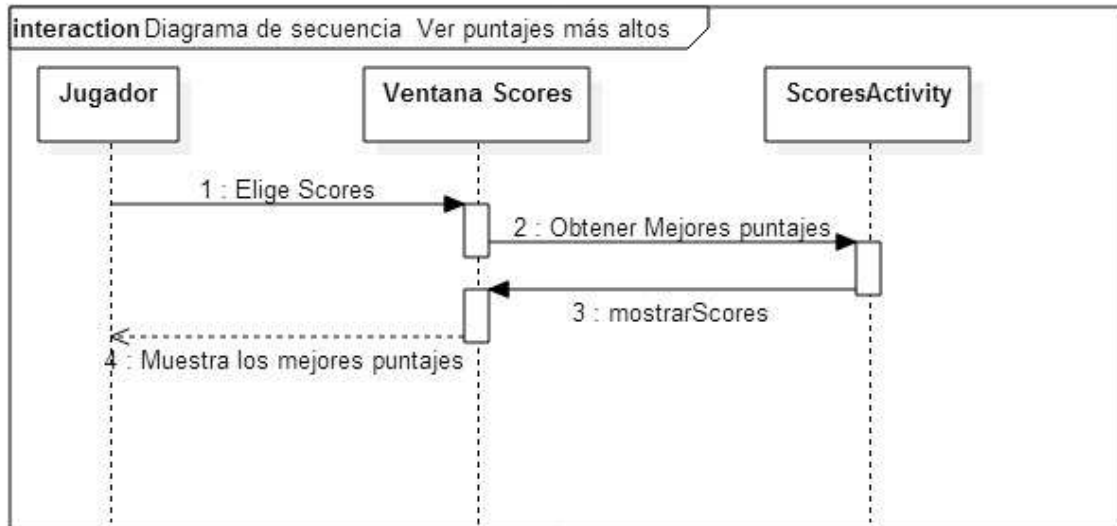


Figura 8 Diagrama de secuencia. Ver Puntajes más altos



## 8 Arquitectura del sistema

La arquitectura del sistema (Figura 9) que se va a utilizar será la de Modelo Vista Controlador (MVC) cuya principal bondad es separar los datos de una aplicación, la interfaz de usuario y la lógica en tres componentes en donde cada una de éstas se comunicará para lograr la correcta funcionalidad del sistema.

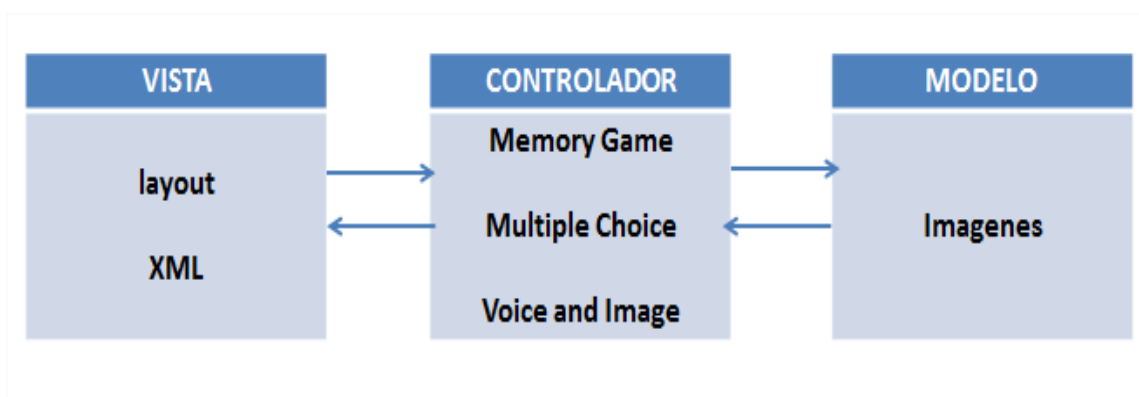


Figura 9 Arquitectura del sistema

En la Figura 10 se observa el árbol de directorios de la aplicación *Idioms Game* con las distintas partes de la arquitectura, las cuales se indican en diferentes colores para diferenciar una de la otra.

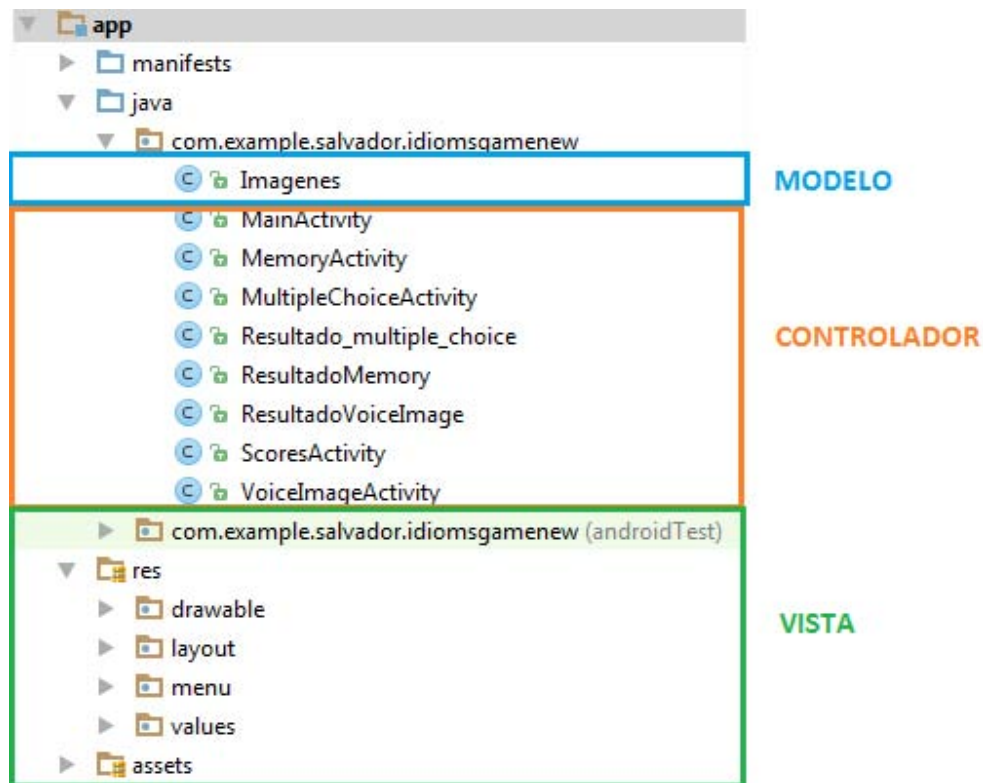


Figura 10 Árbol de directorios de la aplicación *Idioms Game*

### 8.1 Aplicación *Idioms Game*

El modelo está representado por la clase `Imagenes`, la cual tiene la función de inicializar y obtener las imágenes que se mostrarán en cada uno de los mini-juegos. Esta clase encapsula la lógica de negocio de la aplicación *Idioms Game*.

La vista está conformada por el directorio `res`, la cual contiene las subcarpetas: `drawable`, `layout`, `menu` y `values`; éstas contienen todos los recursos para el diseño de la vista de *Idioms Game*, a saber: íconos, botones, imágenes y mensajes escritos, todo esto en conjunto genera la interfaz gráfica que el jugador observará.

El controlador está formado por las clases contenidas en el paquete `com.example.salvador.idiomsgamenew`. Estas clases son las encargadas de dar funcionalidad a las interfaces, es el intermediario entre la vista y el modelo, se define la respuesta del sistema a los eventos generados por el jugador.

Cabe mencionar que los datos recuperados desde el controlador para observar los puntajes obtenidos por el controlador son almacenados en archivos XML<sup>2</sup>.

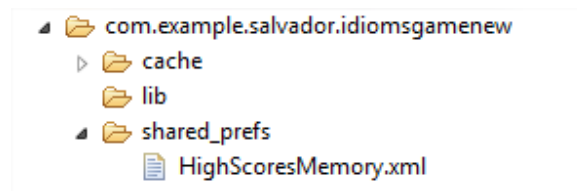
Estos archivos XML se almacenan en una ruta que sigue el siguiente patrón:

/data/data/paquete.java/shared\_prefs/nombre\_coleccion.xml

Así, por ejemplo, al jugar cada uno de los mini juegos, se crean los archivos HighScoresMemory.xml, HighScoresMultipleChoice.xml y HighScoresVoiceImage.xml, los cuales son almacenados en el dispositivo en el que se instala la aplicación, en este caso en la ruta:

/data/data/com.example.salvador.idiomsgamenew/shared\_prefs/

Ejemplo, la Figura 11 muestra la ubicación del archivo creado al jugar *Memory Game*, en esta misma ubicación se crean los archivos HighScoresMultipleChoice.xml y HighScoresVoiceImage.xml una vez que se haya jugado tanto *Multiple Choice* como *Voice and Image*



**Figura 11 Ubicación de los archivos para recuperar los puntajes más altos**

---

<sup>2</sup> XML, siglas en inglés de eXtensible Markup Language ('lenguaje de marcas extensible'), es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible.

**Universidad Autónoma Metropolitana Unidad Azcapotzalco**

División de Ciencias Básicas e Ingeniería

Reporte Final del Proyecto de Integración

Licenciatura en Ingeniería en Computación

Aplicación Android para practicar la pronunciación de los idioms del idioma inglés

Modalidad: Proyecto Tecnológico

Trimestre 2015 Primavera

José Salvador López Balleza

Matricula: 210304254

jslb\_cafcb10@hotmail.com



Asesor:

Dra. María Lizbeth Gallardo López

Profesor Investigador

Departamento de Sistemas

glizbeth@correo.azc.uam.mx



Fecha de entrega: 22 de julio de 2015

Yo, María Lizbeth Gallardo López, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.

Asesor:



---

Dra. María Lizbeth Gallardo López

Yo, José Salvador López Balleza, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.

Alumno:



---

José Salvador López Balleza

## Resumen

Uno de los problemas para las personas de habla hispana cuando deciden aprender el idioma inglés es la pronunciación, porque generalmente las personas tienden a pronunciar las palabras tal y como se escriben. Actualmente existen aplicaciones tanto para computadoras como para dispositivos móviles que ayudan en el aprendizaje del idioma inglés; sin embargo, la mayoría de ellas se centran en la práctica del vocabulario en su forma escrita, y muy pocas se preocupan en la práctica de la correcta pronunciación.

Este proyecto de integración tuvo como objetivo entregar al usuario una aplicación para dispositivos móviles, con sistema operativo Android, que ayude en la práctica de la pronunciación de los conocidos como *idioms* del idioma inglés. La metodología empleada en el desarrollo del proyecto es el Proceso Unificado, el cual está dirigido a través de casos de uso. La construcción del software se realizó a través de iteraciones e incrementos, hasta obtener las funcionalidades de la aplicación, a la cual llamamos *Idioms Game*.

La aplicación *Idioms Game* está formada por tres mini-juegos: *Memory Game*, *Multiple Choice* y *Voice and Image*, además de contar con un marcador de aciertos llamado *Best Scores*. *Memory Game* es un juego de memoria para familiarizar al jugador con los *idioms* y sus imágenes. *Multiple Choice* es un juego de trivia donde el jugador debe pronunciar el *idiom* correspondiente, el cual selecciona de las oraciones dadas como opción. *Voice and Image* es un juego de trivia donde el jugador debe pronunciar el *idiom* correspondiente, pero sin contar con apoyo de oraciones. Es importante mencionar que se empleó una herramienta de reconocimiento de voz llamada *Google Voice Search*, la cual opera en red.

## Índice

1	Introducción.....	7
2	Antecedentes.....	8
2.1	Referencias externas .....	8
2.2	Referencias internas .....	9
3	Justificación.....	10
4	Objetivos.....	10
4.1	Objetivo general.....	10
4.2	Objetivos particulares .....	10
5	Marco teórico.....	11
5.1	Android .....	11
5.1.1	¿Qué es Android? .....	11
5.1.2	Conceptos básicos .....	11
5.1.3	Versión Android .....	12
5.2	Reconocimiento de voz.....	12
5.2.1	Concepto.....	12
5.2.2	Clasificación .....	13
5.2.3	Usos y aplicaciones .....	13
5.2.4	Búsqueda por voz de Google.....	14
6	Desarrollo del proyecto .....	15
6.1	Metodología empleada en el desarrollo del proyecto .....	15
6.2	Diseño del sistema .....	16
6.2.1	Diagrama de casos de uso.....	16
6.2.2	Casos de uso de texto .....	17
6.2.3	Diagrama de clases .....	19
6.2.4	Diagrama de estados.....	22
6.2.5	Arquitectura del sistema.....	23
6.3	Uso del sistema .....	26
6.3.1	Jugar <i>Memory Game</i> .....	26
6.3.2	Ver Scores .....	31
6.4	Hardware y Software necesario .....	32

6.4.1	Tecnología para el desarrollo de la aplicación .....	32
6.4.2	Tecnología para la instalación y puesta en marcha de la aplicación .....	34
7	Resultados.....	36
8	Análisis y discusión de resultados .....	37
9	Conclusiones.....	39
10	Perspectivas del proyecto .....	40
11	Referencias Bibliográficas.....	41



## Índice de figuras

FIGURA 1 DIAGRAMA DE CASOS DE USO .....	16
FIGURA 2 DIAGRAMA DE CLASES DE LA APLICACIÓN.....	19
FIGURA 3 DIAGRAMA DE ESTADOS DE LA APLICACIÓN .....	22
FIGURA 4 ARQUITECTURA DEL SISTEMA .....	23
FIGURA 5 ÁRBOL DE DIRECTORIOS DE LA APLICACIÓN IDIOMS GAME.....	24
FIGURA 6 UBICACIÓN DE LOS ARCHIVOS PARA RECUPERAR LOS PUNTAJES MÁS ALTOS .....	25
FIGURA 7 PANTALLA INICIAL DE LA SECCIÓN MEMORY GAME.....	26
FIGURA 8 PANTALLA AL DAR CLICK A UNA IMAGEN .....	27
FIGURA 9 PANTALLA AL DAR CLICK A UNA SEGUNDA IMAGEN .....	27
FIGURA 10 SE OCULTAN LAS IMÁGENES ANTERIORES Y SE MUESTRA UNA NUEVA.....	28
FIGURA 11 PANTALLA MOSTRADA AL ENCONTRAR DOS IMÁGENES IGUALES .....	28
FIGURA 12 PANTALLA FINAL DE MEMORY GAME.....	29
FIGURA 13 PANTALLA SCORES DE <i>MEMORY GAME</i> .....	30
FIGURA 14 PANTALLA SCORES.....	31
FIGURA 15 PANTALLA PARA BUSCAR LA APLICACIÓN GOOGLE .....	34
FIGURA 16 PANTALLA QUE MUESTRA LA APLICACIÓN QUE SE DEBE DE ELEGIR.....	35
FIGURA 17 PANTALLA QUE SE MUESTRA AL FINALIZAR LA DESCARGA .....	35
FIGURA 18 FUNCIONAMIENTO DE RECONOCIMIENTO DE VOZ.....	37
FIGURA 19 PUNTAJES AL EJECUTAR POR PRIMERA VEZ LA APLICACIÓN .....	38
FIGURA 20 PUNTAJES EN LA SEGUNDA EJECUCIÓN DE LA APLICACIÓN .....	38

## 1 Introducción

El inglés es considerado una lengua universal, principalmente porque es uno de los idiomas más hablados en el mundo y el más empleado en casi todas las áreas de conocimiento.

Aprender el idioma inglés requiere de práctica en cuanto a la gramática y la pronunciación. Para ello, se están proponiendo gran variedad de aplicaciones móviles, lo cual abre el espectro de posibilidades para practicar este idioma. Es conocido que la memoria humana asocia una imagen a la frase de un idioma, lo cual le ayuda a recordar la frase y situarla en el contexto adecuado.

El presente proyecto tiene la finalidad de entregar al usuario una aplicación para su dispositivo móvil que pueda utilizar en cualquier momento y en cualquier lugar para practicar la pronunciación de frases, específicamente los conocidos como *idioms*. Los *idioms* se definen como, “Grupo de palabras que poseen un significado diferente al de cada una de las que lo componen”[1], “Grupo de palabras cuyo significado se deduce del contexto de la frase y no del significado individual de cada palabra que lo componen”[2]; por ejemplo, *A bed of roses* literalmente se traduce al español como: una cama de rosas, pero su significado tiene que ver con el contexto cultural del país de habla inglesa; por lo tanto, la traducción al español es: fácil, agradable, placentero. El hecho de que estas frases no tengan una traducción literal al español pueden convertirse en un obstáculo para estudiantes del idioma inglés, por consiguiente, este proyecto servirá para practicar la pronunciación de los *idioms*, y eventualmente el aprendizaje de los mismos.

## **2 Antecedentes**

### **2.1 Referencias externas**

- LA ENSEÑANZA DEL INGLÉS COMO LENGUA EXTRANJERA EN LA TITULACIÓN DE FILOLOGÍA INGLESA: EL USO DE CANCIONES DE MÚSICA POPULAR NO SEXISTAS COMO RECURSO DIDÁCTICO [3]

Esta es una tesis doctoral que pone énfasis en el aprendizaje del idioma inglés, añade como principal recurso las canciones, argumentando que las canciones transmiten información lingüística, social y cultural. La similitud entre esta tesis y la aplicación que se pretende desarrollar se encuentra en que ambas tienen como objetivo el aprendizaje del idioma inglés como lengua extranjera en los estudiantes de nivel universitario, sin embargo utilizan herramientas distintas, en la tesis el uso de canciones y en la propuesta, el uso de imágenes de texto y audio.

#### **Artículos:**

- RECONOCIMIENTO DE VOZ [4]

En este artículo se mencionan algunos antecedentes del reconocimiento por voz, explican cómo es que comenzó el desarrollo de esta herramienta, además menciona algunos inconvenientes para lograr una interpretación aceptable del mensaje acústico recibido, dado que a veces se presenta información plagada de ambigüedades y errores debido al ruido del entorno. Este artículo sirvió como marco teórico sobre el reconocimiento de voz.

#### **Software:**

- DUOLINGO[5]

Es una aplicación que se presenta tanto para dispositivos móviles como para plataforma web, su objetivo es que el usuario poco a poco vaya adquiriendo conocimientos mediante preguntas o grabaciones para ir subiendo de nivel en el idioma que desee, entre ellos el inglés. La similitud entre Duolingo y la propuesta de la aplicación recae en que se tiene el objetivo de estimular la práctica del idioma inglés mediante imágenes y comandos de voz en forma de juego; sin embargo, en la propuesta sólo se pretende desarrollar el aprendizaje de 1 idioma (el inglés), en cambio Duolingo presenta una gran variedad de idiomas para aprender como: Portugués, Alemán, Francés e Italiano.

## 2.2 Referencias internas

- SRCV: SISTEMA COMPUTACIONAL INTERACTIVO BASADO EN RECONOCIMIENTO DE COMANDOS DE VOZ PARA APLICACIONES EDUCATIVAS [6]

El sistema SRCV es una herramienta que fomenta el aprendizaje y estimula las habilidades de niños en etapa preescolar, pues les presenta una serie de preguntas y mensajes de evaluación (audio e imagen) con las cuales el niño se va a guiar y dará una respuesta por medio de su voz.

En el proyecto propuesto, al igual que el sistema SRCV, pretende fomentar el aprendizaje por medio de recursos audiovisuales; además, hacen uso de la herramienta de reconocimiento de comandos de voz; sin embargo, nuestra propuesta, a diferencia del sistema SRCV, va dirigida a un público que tenga previos conocimientos del idioma inglés y no a niños en etapa preescolar.

- APLICACIÓN COLABORATIVA PARA DISPOSITIVOS MÓVILES CON SISTEMA OPERATIVO ANDROID [7]

La aplicación colaborativa es una herramienta que facilita la interacción entre miembros de un grupo de trabajo en la edición de figuras geométricas, así como en la edición texto. Además, para que los miembros del grupo de trabajo puedan compartir la información (figuras o texto) es necesario que estén conectados a una misma red. Nuestra propuesta no pretende introducir el trabajo colaborativo, sino que pretende desarrollar un conocimiento individual del idioma inglés, cabe mencionar además que ambos operan en la plataforma Android para dispositivos móviles.

- APLICACIÓN ANDROID PARA LA PRÁCTICA DE VERBOS COMPUESTOS DEL IDIOMA INGLÉS [8]

Este proyecto presenta una aplicación Android para dispositivos móviles que apoya en el aprendizaje de los *phrasal verbs* del idioma inglés. Consta de tres juegos donde se involucran imágenes y oraciones, además de proponer tres niveles principiantes, intermedios y avanzados. Nuestro proyecto también contará con tres niveles y tres juegos, pero se enfocará en los *idioms* y no en los *phrasal verbs* del idioma inglés; además, se añadirá la herramienta del reconocimiento de voz, cabe mencionar que ambos operan en dispositivos móviles con sistema operativo Android.

### 3 Justificación

Uno de los problemas para las personas de habla hispana cuando deciden aprender el idioma inglés es la pronunciación, pues generalmente las personas tienden a pronunciar las palabras tal y como se escriben. Aunque actualmente existen aplicaciones tanto para computadoras como para dispositivos móviles que ayudan en el aprendizaje del idioma inglés, la mayoría de ellas se centran en sólo aprender vocabulario y muy pocas se preocupan en el aprendizaje de la correcta pronunciación.

El presente proyecto considera que en un ambiente de juego se hará mucho más atractiva la práctica de los *idioms*. La base de los juegos será la asociación de una frase con la imagen y audio correspondientes. La aplicación estará formada por tres mini juegos, los cuales emplearán una herramienta de reconocimiento de voz.

La aplicación para la práctica de los *idioms* será desarrollada para dispositivos móviles con sistema operativo *Android*, principalmente porque el consumo de estos dispositivos está aumentado; por lo tanto, permitirá que un mayor número de personas acceda a los mini juegos que se están proponiendo en este proyecto.

### 4 Objetivos

#### 4.1 Objetivo general

Desarrollar una aplicación para dispositivos móviles con sistema operativo Android para practicar la pronunciación de las frases denominadas *idioms* del idioma inglés.

#### 4.2 Objetivos particulares

- Diseñar e implementar un módulo para el juego *Memory Game*
- Diseñar e implementar un módulo para el juego *Multiple Choice*
- Diseñar e implementar un módulo para el juego *Voice and Image*
- Integrar como parte de los juegos una herramienta de reconocimiento de voz que permita evaluar la pronunciación del usuario

## 5 Marco teórico

### 5.1 Android

#### 5.1.1 ¿Qué es Android?

Android es un sistema operativo inicialmente pensado para teléfonos móviles, al igual que iOS, Symbian y Blackberry OS. Lo que lo hace diferente es que está basado en Linux, un núcleo de sistema operativo libre, gratuito y multiplataforma [9].

El sistema permite programar aplicaciones en una variación de Java llamada Dalvik. El sistema operativo proporciona todas las interfaces necesarias para desarrollar aplicaciones que accedan a las funciones del teléfono (como el GPS, las llamadas, la agenda, etc.) de una forma muy sencilla en un lenguaje de programación muy conocido como es Java.

#### 5.1.2 Conceptos básicos

A continuación se enumerarán los conceptos básicos que hay que tener en cuenta en el desarrollo de toda aplicación Android y que se harán referencia de ahora en adelante:

##### Activity

“Las actividades (*activities*) representan el componente principal de la interfaz gráfica de una aplicación Android”[10], una *activity* contiene la interfaz que con la que el usuario interactúa. Es decir, se puede pensar una actividad como el elemento análogo a una ventana o pantalla en cualquier otro lenguaje visual. Además es importante mencionar un concepto importante al hablar de *activities*, y es conocido como el ciclo de vida de una *activity*, el cual se puede definir como las etapas por las que pasa una *activity* desde el momento en que se muestra en pantalla hasta el momento que se destruye.

##### View

“Los objetos *view* son los componentes básicos con lo que se crea la interfaz gráfica de la aplicación (*activity*)”[10]. Inicialmente Android pone a disposición gran cantidad de estos componentes básicos (*views*): cuadros de texto, botones, listas desplegadas, imágenes, etc. Aunque también existe la posibilidad de extender la funcionalidad de estos controles básicos o crear nuestros propios controles personalizados.

##### Widget

“Los widgets son componentes visuales, normalmente interactivos, que se muestran la pantalla principal (*home screen*) del dispositivo Android”[10]. Permiten mostrar información de la aplicación al usuario directamente sobre la pantalla principal.

## Intent

“Un intent es el elemento básico de comunicación entre los distintos componentes Android que han sido descritos antes”[10]. Estos elementos se pueden entender como el medio de comunicación entre los distintos elementos de una aplicación, ya que es posible el envío de mensajes o peticiones entre sí. “Mediante un intent se puede mostrar una actividad desde cualquier otra, iniciar un servicio, enviar un mensaje *broadcast*, iniciar otra aplicación, etc.” [10].

### **5.1.3 Versión Android**

La plataforma Android proporciona un framework API<sup>1</sup> que las aplicaciones pueden utilizar para interactuar con el sistema Android [11]. El framework API consta de:

- Un conjunto básico de paquetes y clases
- Un conjunto de elementos y atributos XML declarados en un archivo *manifest*
- Un conjunto de elementos y atributos XML para la declaración y el acceso a los recursos
- Un conjunto de *Intents*
- Un conjunto de permisos que las aplicaciones pueden solicitar, así como permisos de aplicación incluidos en el sistema.

“Cada versión de la plataforma Android puede incluir actualizaciones en el framework API de aplicación que proporciona. El API declarado en la aplicación es muy importante por razones de compatibilidad del dispositivo y el tiempo de vida de desarrollo y mantenimiento del software” [11].

La versión utilizada durante el desarrollo de este proyecto fue la 4.0 Ice Cream Sandwich añadida en el API nivel 14.

## **5.2 Reconocimiento de voz**

### **5.2.1 Concepto**

El reconocimiento de voz es una disciplina de la inteligencia artificial, es una herramienta computacional con la capacidad de procesar la señal de voz emitida por el usuario y reconocer la información contenida en ésta, para después convertirla en texto o emitir órdenes que actúan sobre un proceso [12].

---

<sup>1</sup> Interfaz de programación de aplicaciones (API – ApplicationProgramming Interface). Conjunto de funciones o métodos que exponen las clases de alguna biblioteca para ser utilizado por otro software.

En síntesis, el principal objetivo de la herramienta de reconocimiento de voz, es permitir la comunicación hablada entre seres humanos y computadoras.

### 5.2.2 Clasificación

“Los sistemas de reconocimiento de voz pueden clasificarse según los siguientes criterios” [12]:

- *Entrenabilidad*: determina si el sistema necesita un entrenamiento previo antes de empezar a usarse.
- *Dependencia del hablante*: determina si el sistema debe entrenarse para cada usuario o es independiente del hablante.
- *Continuidad*: determina si el sistema puede reconocer habla continua o el usuario debe hacer pausas entre palabra y palabra.
- *Robustez*: determina si el sistema está diseñado para usarse con señales poco ruidosas o, por el contrario, puede funcionar aceptablemente en condiciones ruidosas, ya sea ruido de fondo, ruido procedente del canal o la presencia de voces de otras personas.
- *Tamaño del dominio*: determina si el sistema está diseñado para reconocer lenguaje de un dominio reducido (unos cientos de palabras, por ejemplo las reservaciones de vuelos o peticiones de información meteorológica) o extenso (miles de palabras).

### 5.2.3 Usos y aplicaciones

“Aunque en teoría cualquier tarea en la que se interactúe con un ordenador puede utilizar el reconocimiento de voz, actualmente las siguientes aplicaciones son las más comunes”[12]:

*Dictado automático*: El dictado automático es, hasta hoy, el uso más común de las tecnologías de reconocimiento de voz. En algunos casos, como en el dictado de recetas médicas y diagnósticos o el dictado de textos legales, se usan corpus especiales para incrementar la precisión del sistema.

*Control por comandos*: Los sistemas de reconocimiento de habla diseñados para dar órdenes a un computador (por ejemplo "Abrir Firefox", "cerrar ventana") se llaman Control por comandos. Estos sistemas reconocen un vocabulario muy reducido, lo que incrementa su rendimiento.

*Telefonía*: Algunos sistemas PBX permiten a los usuarios ejecutar comandos mediante el habla, en lugar de pulsar tonos. En muchos casos se pide al usuario que diga un número para navegar un menú.

*Sistemas portátiles*: Los sistemas portátiles de tamaño reducido, como los relojes o los teléfonos móviles, tienen unas restricciones muy concretas de tamaño y forma, así que el habla es una solución natural para introducir datos en estos dispositivos.



*Sistemas diseñados para discapacitados:* Los sistemas de reconocimiento de voz pueden ser útiles para personas con discapacidades que les impidan teclear con fluidez, así como para personas con problemas auditivos, que pueden usarlos para obtener texto escrito a partir de habla. Esto permitiría, por ejemplo, que los aquejados de sordera pudieran recibir llamadas telefónicas.

#### **5.2.4 Búsqueda por voz de Google**

La búsqueda por voz de Google (Google Voice Search en inglés) es un producto que permite a los usuarios utilizar la búsqueda a partir de la voz, en un teléfono móvil o en una computadora; es decir, se realiza la búsqueda de datos en internet solamente ingresando lo que se desea buscar con ayuda de la voz [13].

“En el reconocimiento de voz se busca encontrar una relación directa entre una onda de sonido y un texto, y no entre dos palabras o frases en distintos idiomas – aunque en la traducción en modo conversación, se requieren ambas cosas funcionando a la par” [14].

##### **5.2.4.1 Funcionamiento**

Son tres partes las que conforman el modelo de reconocimiento de voz de Google. Primero, está el modelo acústico que mapea todas las maneras posibles en que las ondas sonoras pueden formar fonemas, tales como “ah”, “mm” o “bah”. Después los fonemas se reúnen en el modelo léxico, que básicamente es un diccionario sobre cómo pronunciar todas las palabras en un idioma. “Finalmente, las palabras se entretajan como parte de un modelo de lenguaje que indica qué palabras suelen venir después o antes de otras [14].

##### **5.2.4.2 Búsqueda por voz de Google en Android**

En Android, además de realizar búsquedas, se pueden redactar correos a partir de la voz, o realizar un dictado como parte de una aplicación, en lugar de emplear el teclado; de hecho la opción para emplear la voz como método de entrada se ha agregado por defecto al teclado de Android como alternativa a escribir [14].

## **6 Desarrollo del proyecto**

### **6.1 Metodología empleada en el desarrollo del proyecto**

Para la creación del presente proyecto se utilizó el Proceso Unificado como metodología de base para el desarrollo de la aplicación móvil *Idioms Game*. “El proceso unificado es un marco de desarrollo de software, el cual se caracteriza por estar centrado en la arquitectura del sistema; por estar dirigido a través de casos de uso, y por llevar el desarrollo de forma iterativa e incremental” [15]. Este proceso está compuesto por fases, las cuales se dividen, a su vez, en una serie de iteraciones. Por cada iteración, se realizaron una serie de actividades, a saber: Modelo de negocio, Análisis de requerimientos, Diseño, Implementación y Pruebas.

Cada iteración realizada durante el proyecto, significó un incremento del producto desarrollado, lo cual añadió mejoras a las funcionalidades de la aplicación. En cada iteración, se tomó una parte de los casos de uso, con los cuales se desarrolló todo el camino a través de las distintas actividades mencionadas anteriormente.

En la iteración de inicio de este proyecto las actividades que tuvieron mayor relevancia fueron: el análisis de requerimientos y el diseño.

En el análisis de requerimientos se determinó que funciones debía hacer la aplicación y con qué fin, para ello se elaboró el documento de casos de uso de texto, en el que se pudieron comprender los requisitos del sistema; así como, la interacción con el usuario. En el diseño se propuso una solución para satisfacer los requerimientos del sistema; para ello, se elaboraron los diagramas de clases, diagramas de estados y diagramas de secuencia.

Con la experiencia obtenida en la primera iteración fue posible implementar el resto de los casos de uso en iteraciones subsecuentes, con lo cual se fue incrementando y mejorando la aplicación *Idioms Game*.

## 6.2 Diseño del sistema

En esta sección se presentarán los artefactos de diseño que se utilizaron para modelar el comportamiento de la aplicación *Idioms Game*. Dichos artefactos son:

- Diagrama de casos de uso
- Casos de uso de texto
- Diagrama de clases
- Diagrama de estados
- Arquitectura del sistema

### 6.2.1 Diagrama de casos de uso

Este diagrama forma parte de la Memoria de Diseño de *Idioms Game*, en el cual se observa tanto al actor principal (el cual será el jugador) como los casos de uso, con los cuales el actor busca producir un resultado de valor agregado a su tarea. El diagrama de casos de uso se muestra en la Figura 1.

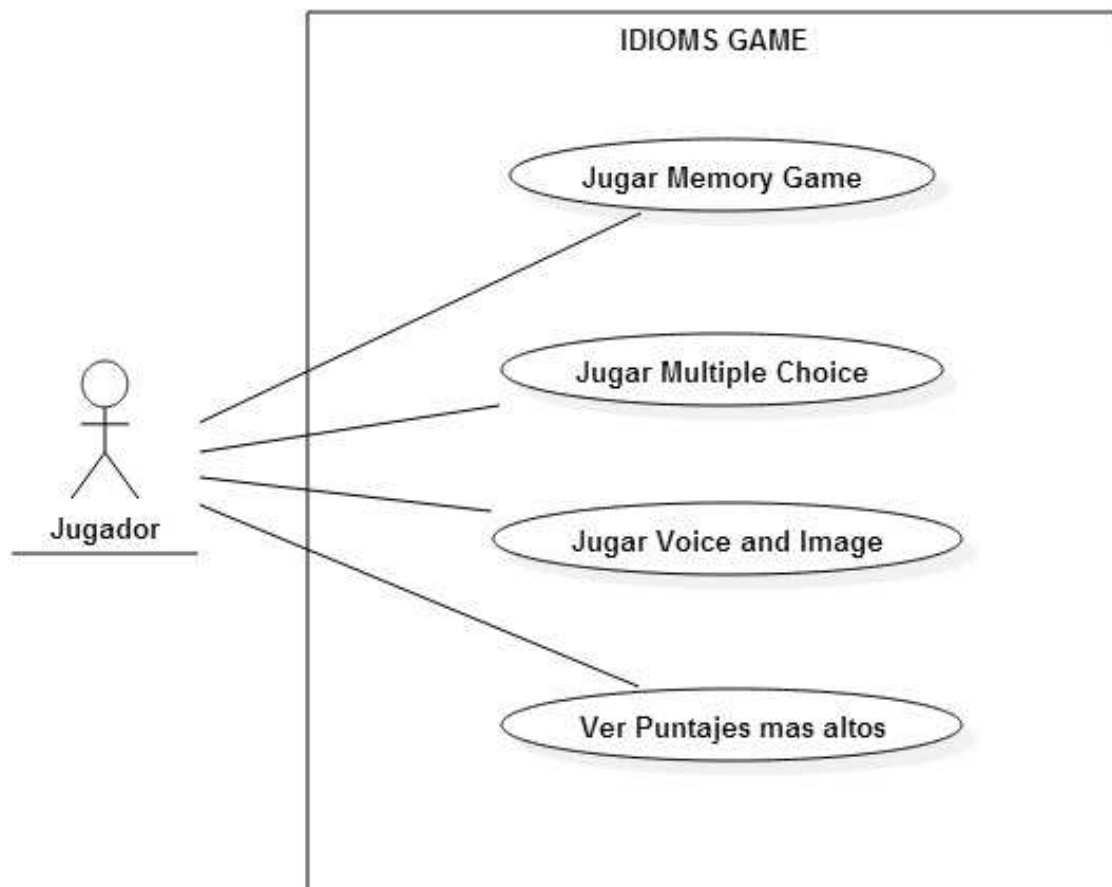


Figura 1 Diagrama de casos de uso

## 6.2.2 Casos de uso de texto

A continuación se presentan los casos de uso de texto: Jugar *Memory Game* y Ver puntajes más altos, para ejemplificar el diseño del sistema. El detalle de todos los casos de uso los casos de uso se encuentran en la Memoria de Diseño del sistema.

### 6.2.2.1 Caso de uso: Jugar *Memory Game*

**Descripción:**

Esta operación permitirá conocer la dinámica de un juego para ejercitar la memoria. El jugador va a conocer la relación entre las imágenes mostradas y el *idiom* que le corresponde, pues la imagen trae consigo un rótulo con el *idiom* al que se asocia.

**Actores y sus intereses:**

El actor principal (Jugador) está interesado en comenzar con la ejecución del juego *Memory Game* para conocer las imágenes con las que cuenta la aplicación, así como el *idiom* con el que se asocia.

**Disparador:**

La operación inicia cuando el Jugador, desde el menú principal, elige la opción *Memory Game*.

**Pre-condición:**

El jugador seleccionó el ícono que lanza la aplicación.

**Post-condición:**

Al terminar el caso de uso, el tiempo que le tomó al jugador encontrar los seis pares quedará registrado en un archivo dentro de la aplicación.

**Flujo principal:**

- 1) El jugador selecciona, desde el menú principal, la opción *Memory Game*.
- 2) La aplicación coloca aleatoriamente las posiciones en que aparecerán las 12 imágenes.
- 3) La aplicación muestra un cronómetro, el cual inicia desde cero e incrementa en 1 cada segundo transcurrido. También presenta 12 imágenes con un signo de interrogación (representan la cara de abajo de una tarjeta) acomodadas en 3 columnas con 4 imágenes cada una.
- 4) El jugador da click a una imagen cara abajo.
- 5) La aplicación muestra una imagen que representa la cara de arriba que el actor principal oprimió.
- 6) El actor principal da click a una nueva imagen cara abajo.
- 7) La aplicación muestra una nueva imagen que representa la cara de arriba.
- 8) La aplicación verifica si las 2 imágenes mostradas son las mismas, en caso positivo, las imágenes se quedan cara arriba, en caso contrario, las imágenes regresan a su estado original con la cara hacia abajo.
- 9) Se repite el proceso desde el paso 2 hasta el paso 6.

- 10) El jugador encuentra los seis pares posibles.
- 11) La aplicación detiene el cronómetro y habilita un botón “Continuar”.
- 12) El jugador selecciona “Continuar”.
- 13) La aplicación muestra una pantalla con las leyendas: *YOUR TIME* que representa el tiempo que tardó el jugador en encontrar los 6 pares, *BEST TIME* que representa el mejor tiempo de entre todas las veces que se ha jugado y además muestra dos botones: “Play Again” y “Main Menu”.
- 14) El jugador selecciona “Play Again”, el proceso se repite desde el paso 2.

**Flujo alternativo:**

- 1) El jugador selecciona “Main Menu”, la aplicación dirige a la pantalla con el menú principal.

### 6.2.2.2 Caso de uso: Ver Puntajes más altos

**Descripción:**

Esta operación permitirá al jugador observar cuáles han sido sus mejores puntajes en los tres mini juegos: *Memory Game*, *Multiple Choice* y *Voice and Image*.

**Actores y sus intereses:**

El actor principal (Jugador) está interesado en observar cuáles son sus mejores marcas establecidas a lo largo de todas las partidas jugadas en cada uno de los mini juegos.

**Disparador:**

La operación inicia cuando el Jugador, desde el menú principal, elige la opción “Scores”.

**Pre-condición:**

Para que se cumpla esta operación, es necesario que el jugador haya jugado por lo menos una vez cada uno de los mini-juegos.

**Post-condición:**

Al terminar el caso de uso, el jugador observará los mejores puntajes que ha obtenido en cada uno de los 3 mini juegos, para que de esta manera tenga la pauta para establecer nuevas marcas.

**Flujo principal:**

- 1) El jugador selecciona, desde el menú principal, la opción “Scores”.
- 2) La aplicación muestra la pantalla “*BEST SCORES*”, en donde se observa el menor tiempo que se ha obtenido en *Memory Game*, el mayor número de aciertos obtenidos tanto en “Multiple Choice” como en “Voice and Image” y el botón “Main Menu”.
- 3) El jugador observa sus mejores puntajes y da click en el botón “Main Menu”.
- 4) La aplicación dirige a la pantalla con el menú principal.

### 6.2.3 Diagrama de clases

Los casos de uso de texto de la aplicación *Idioms Game* permitieron identificar las clases que representan el funcionamiento de cada uno de los mini juegos. A continuación, en la Figura 2 se observa el diagrama de clases de software que conforman la aplicación.

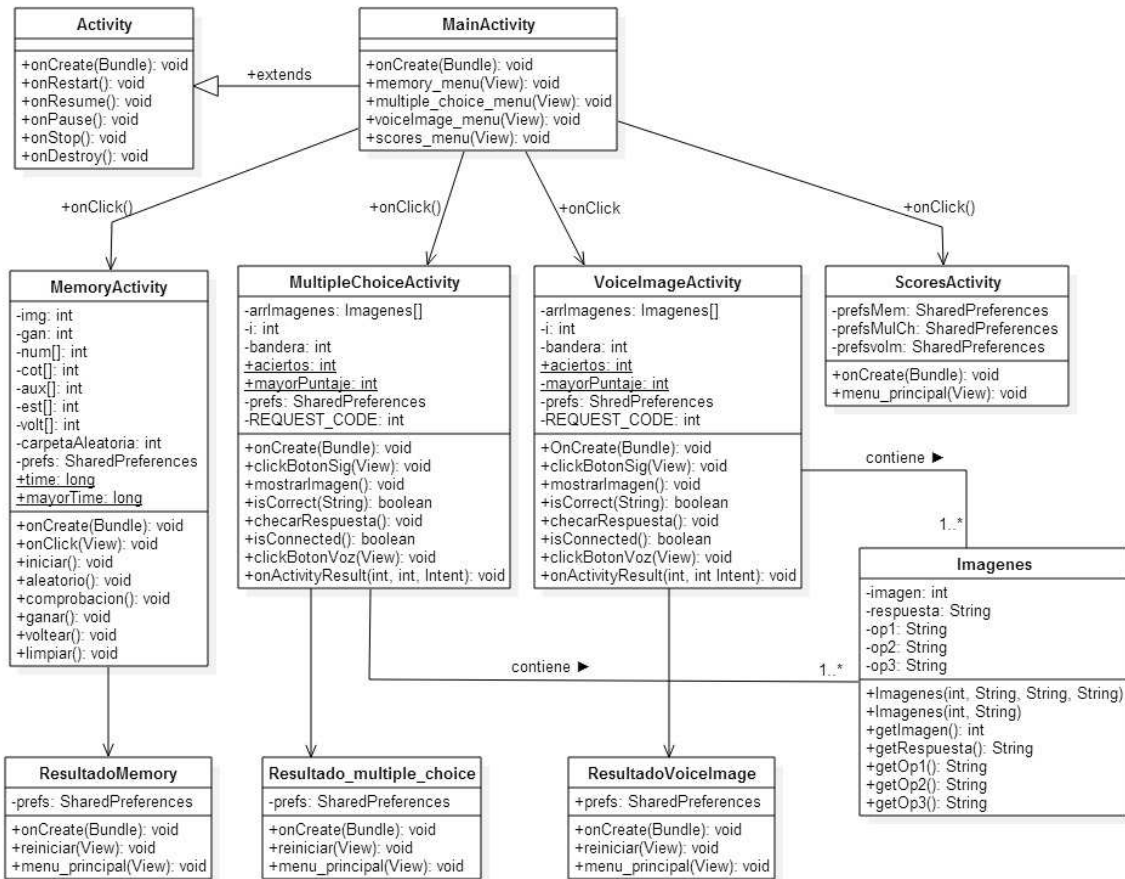


Figura 2 Diagrama de clases de la aplicación

A continuación, se listan cada una de las clases con las que cuenta la aplicación, así como una descripción sobre las tareas que realiza y sus atributos.

#### 6.2.3.1 Clase Activity

Una Activity es una ventana que contiene la interface de usuario de la aplicación. El objetivo principal de una activity es interactuar con el usuario. Desde el momento que una activity aparece en la pantalla hasta el momento que se oculta, pasa por varias etapas, conocido como el ciclo de vida de una activity.

Esta clase define los siguientes eventos:

- onCreate() – Llamado cuando la activity es creada.
- onStart() – Llamado cuando la activity empieza a ser visible al usuario.
- onPause() – Llamado cuando la activity actual va a ser pausada y la activity previa será reanudada.
- onStop() – Llamado cuando la activity ya no es visible para el usuario.
- onDestroy() – Llamado antes de que la activity sea destruida por el sistema (también manualmente o por el sistema para conservar memoria).
- onRestart() – Llamado cuando la activity ha sido detenido y es reiniciada la activity de nuevo.

### 6.2.3.2 Clase MainActivity

La clase MainActivity es la encargada de administrar la pantalla que contiene el menú principal de la aplicación, esta clase hereda de Activity el método onCreate() y además define los métodos necesarios para poder navegar hacia las pantallas tanto de los 3 mini-juegos como de los Scores.

### 6.2.3.3 Clase MemoryActivity

La clase MemoryActivity hereda de Activity el método onCreate(), tiene la tarea de desplegar la interfaz gráfica, además aquí se implementa la lógica del juego *Memory Game*, para esto se definieron los atributos: img, cant, gan, además de los arreglos num, cot, aux y est, los cuales en conjunto mantendrán la consistencia de la lógica del juego pues con estos atributos se controla que las imágenes no se repitan más de dos veces, también se mantiene una variable para tener el control de cuantos pares se han conseguido y de esta manera saber hasta cuando se termina el juego. Por último, se tienen también como atributos a: carpetaAleatoria, que tomará aleatoriamente una carpeta que contiene un banco con 6 imágenes para que cada vez que se ejecute el juego se muestren nuevas imágenes; time y mayorTime, los cuales son atributos que sirven como auxiliares para llevar el control del tiempo empleado en terminar el juego, así como validar si este tiempo es un nuevo récord.

### 6.2.3.4 Clase Imagenes

En la clase Imagenes se definen dos constructores para manejar las imágenes que forman parte de las posibles respuestas (op1, op2 y op3); donde solo una de ellas es la correcta (respuesta). Asimismo, esta clase cuenta con métodos set y get, con los cuales, se pueden definir y recuperar los atributos: imagen, respuesta, op1, op2 y op3 para cada una de las instancias que sean creadas de esta clase, a partir de los juegos: MultipleChoiceActivity y VoiceImageActivity.

### **6.2.3.5 Clase MultipleChoiceActivity**

La clase `MultipleChoiceActivity` hereda de `Activity` e implementa toda la lógica del juego *Multiple Choice*, se relaciona directamente con la clase `Imagenes` ya que es necesario obtener tanto la imagen a desplegar como su respuesta con ayuda de un arreglo de imágenes, el cual es definido como un atributo llamado `arrImagenes` que será inicializado con ayuda de uno de los dos constructores de la clase `Imagenes`, además de éste, se definieron los siguientes atributos: `i`, `bandera`, `aciertos`, `mayorPuntaje` y `REQUEST_CODE`, en donde el atributo `aciertos` mantiene el puntaje obtenido en el mini-juego.

### **6.2.3.6 Clase VoiceImageActivity**

Al igual que las clases `MemoryActivity` y `MultipleChoiceActivity`, `VoiceImageActivity` hereda de `Activity`. Esta clase tiene relación directa con la clase `Imagenes`, pues al igual que `MultipleChoiceActivity`, tiene que obtener tanto la imagen a desplegar como la respuesta con que se asocia; por lo tanto, es aquí donde se define la lógica que seguirá el juego *Voice and Image*: Desplegará la imagen, activará el micrófono para transformar la voz del jugador en texto, validará el texto obtenido con la respuesta que se tiene como correcta, aumentará y guardará en un archivo, dentro de la aplicación, el número de aciertos en caso de ser necesario.

### **6.2.3.7 Clase ScoresActivity**

La clase `ScoresActivity` hereda de `Activity`, tiene relación directa con `MainActivity`, pues de no ser por la clase principal, el contenido de ésta clase no podría ser visualizado. La tarea principal de `ScoresActivity` es recuperar los mejores puntajes obtenidos por el jugador en cada uno de los 3 mini-juegos con ayuda de los atributos `prefsMem`, `prefsMulCh` y `prefsvoIm`.

### **6.2.3.8 Clase ResultadoMemory**

La clase `ResultadoMemory` tiene relación directa con `MemoryActivity`. `ResultadoMemory` recupera el puntaje obtenido por el jugador en su atributo `prefs`; el puntaje debió ser registrado previamente por la clase donde se implementa la lógica del juego *Memory Game*.

### **6.2.3.9 Clase Resultado\_multiple\_choice**

La clase `Resultado_multiple_choice` se relaciona directamente con `MultipleChoiceActivity`. `Resultado_multiple_choice` tiene la tarea de recuperar el puntaje obtenido por el jugador en el atributo `prefs`; el puntaje debió ser registrado previamente por la clase `MultipleChoiceActivity`, que es la clase en donde se implementa la lógica del juego *Multiple Choice*.

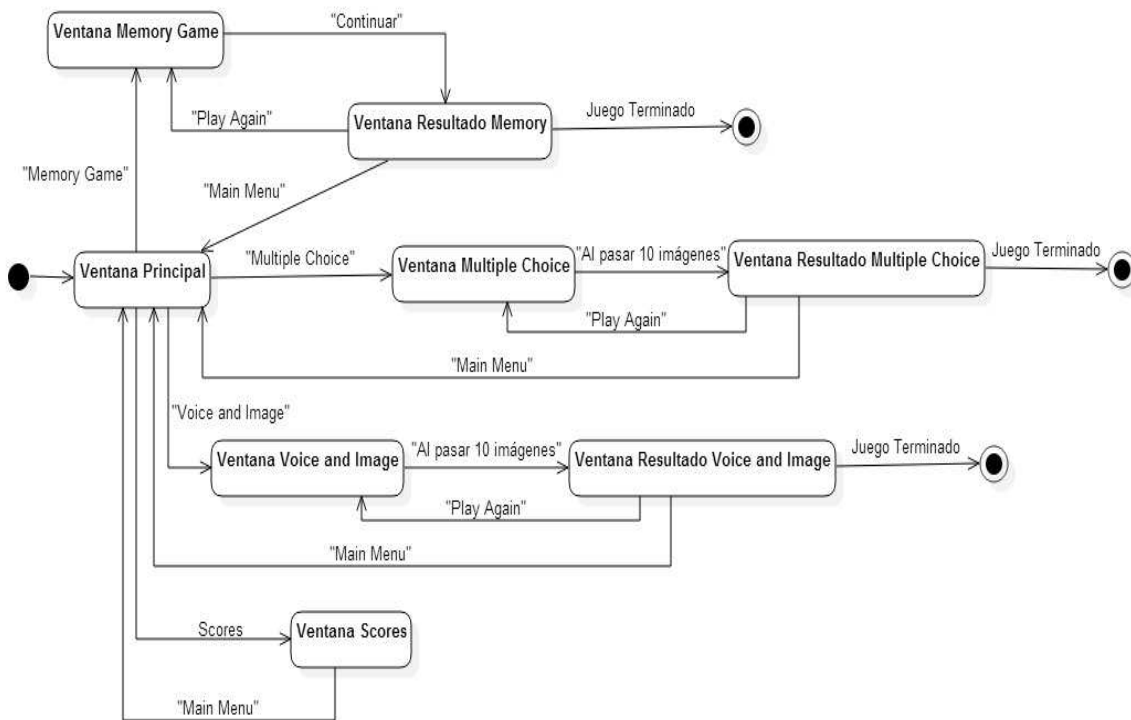


### 6.2.3.10 Clase ResultadoVoiceImage

La clase `ResultadoVoiceImage` tiene relación directa con la clase `VoiceImageActivity`. `ResultadoVoiceImage` tiene la finalidad de recuperar el puntaje obtenido por el jugador en su atributo `prefs`, el cual debió ser almacenado previamente por la clase `VoiceImageActivity`, en la cual se desarrolla la lógica del juego *Voice and Image*.

### 6.2.4 Diagrama de estados

En el diagrama de la Figura 3 se observa la secuencia entre las diferentes ventanas de la aplicación, desde el punto inicial hasta el punto final, indicando los diferentes caminos de decisión que se pueden seguir durante la ejecución de la aplicación. Gracias a este diagrama es posible comprender mejor las transiciones de los mini-juegos.



**Figura 3 Diagrama de estados de la aplicación**

Al arrancar la aplicación se muestra la pantalla del menú principal, donde el jugador puede tomar una de cuatro decisiones: Jugar *Memory Game*, jugar *Multiple Choice*, jugar *Voice and Image* o visualizar los mejores puntajes. Cuando el jugador concluye un juego, la aplicación muestra una pantalla donde se visualiza tanto el puntaje obtenido en esa oportunidad, como el mejor puntaje obtenido en todas las veces que se ha jugado; en esta

pantalla se pueden tomar dos decisiones más: “Play Again” y “Main Menu”. La primera reinicia el juego y la segunda regresa al menú principal de la aplicación.

Mientras la aplicación está en ejecución, es necesario detectar el evento de pulsación de pantalla para iniciar el mini juego que desee el jugador. Al detectarse la pulsación, se visualiza la pantalla correspondiente a la opción del jugador. Dicho evento se considera como externo, pues es una acción provocada por el usuario. Por ejemplo, en el mini juego *Memory Game* una pulsación de pantalla sobre la imagen de una carta provoca voltearla; otro ejemplo, en *Multiple Choice*, una pulsación sobre la imagen del micrófono activa el micrófono del dispositivo.

Por otro lado, los eventos internos de la aplicación dependerán del estado de la partida. Por ejemplo, en el juego *Memory Game*, al desplegarse 6 pares de imágenes la partida finaliza, pues al conseguir todos los pares de imágenes posibles en el juego, la aplicación dirige inmediatamente a la pantalla en donde se observa el tiempo obtenido para consumir el objetivo del juego.

#### 6.2.5 Arquitectura del sistema

La arquitectura del sistema (Figura 4) que se va a utilizar será la de Modelo Vista Controlador (MVC) cuya principal bondad es separar los datos de una aplicación, la interfaz de usuario y la lógica en tres componentes en donde cada una de éstas se comunicará para lograr la correcta funcionalidad del sistema.

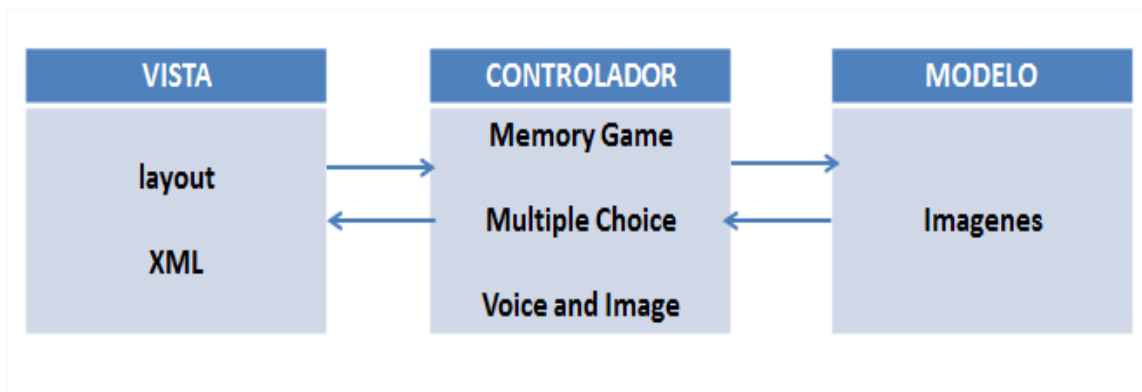
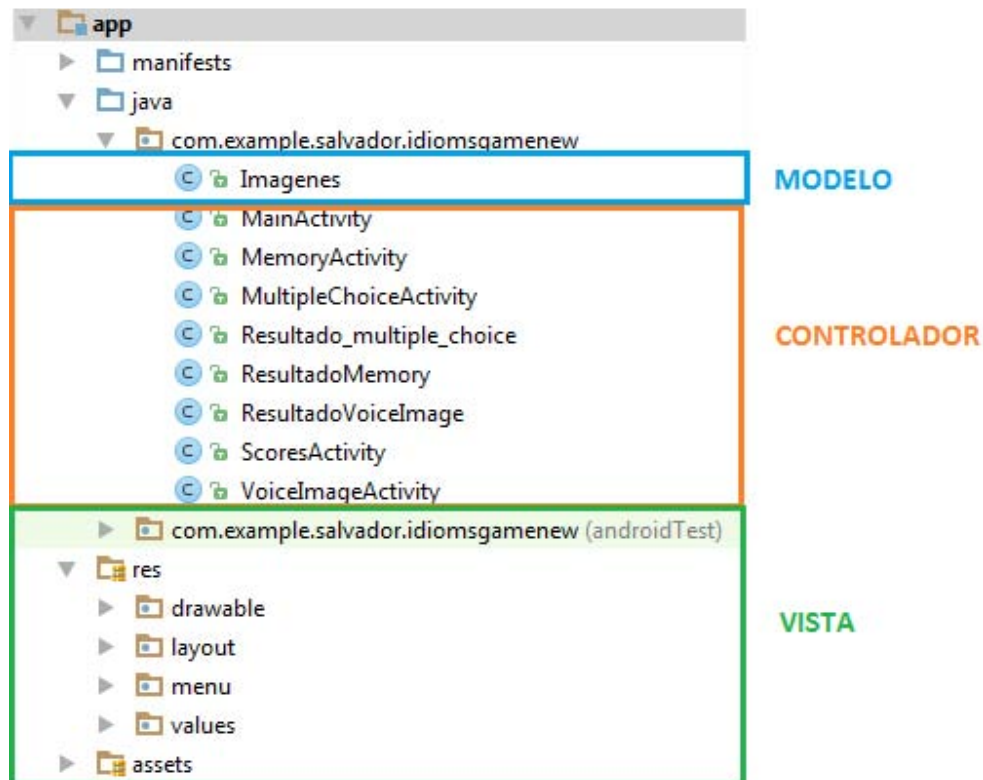


Figura 4 Arquitectura del sistema

En la Figura 5 se observa el árbol de directorios de la aplicación *Idioms Game* con las distintas partes de la arquitectura, las cuales se indican en diferentes colores para diferenciar una de la otra.



**Figura 5** Árbol de directorios de la aplicación *Idioms Game*

El modelo está representado por la clase *Imagenes*, la cual tiene la función de inicializar y obtener las imágenes que se mostrarán en cada uno de los mini-juegos. Esta clase encapsula la lógica de negocio de la aplicación *Idioms Game*.

La vista está conformada por el directorio *res*, la cual contiene las subcarpetas: *drawable*, *layout*, *menu* y *values*; éstas contienen todos los recursos para el diseño de la vista de *Idioms Game*, a saber: íconos, botones, imágenes y mensajes escritos, todo esto en conjunto genera la interfaz gráfica que el jugador observará.

El controlador está formado por las clases contenidas en el paquete *com.example.salvador.idiomsgamenew*. Estas clases son las encargadas de dar funcionalidad a las interfaces, es el intermediario entre la vista y el modelo, se define la respuesta del sistema a los eventos generados por el jugador.

Cabe mencionar que los datos recuperados desde el controlador para observar los puntajes obtenidos por el controlador son almacenados en archivos XML<sup>2</sup>.

<sup>2</sup>XML, siglas en inglés de eXtensibleMarkupLanguage ('lenguaje de marcas extensible'), es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible.

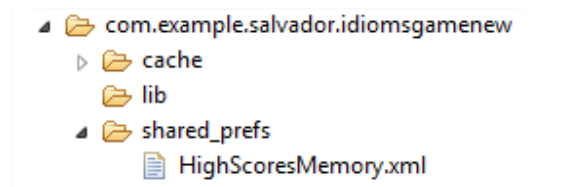
Estos archivos XML se almacenan en una ruta que sigue el siguiente patrón:

```
/data/data/paquete.java/shared_prefs/nombre_coleccion.xml
```

Así, por ejemplo, al jugar cada uno de los mini juegos, se crean los archivos HighScoresMemory.xml, HighScoresMultipleChoice.xml y HighScoresVoiceImage.xml, los cuales son almacenados en el dispositivo en el que se instala la aplicación, en este caso en la ruta:

```
/data/data/com.example.salvador.idiomsgamenew/shared_prefs/
```

Ejemplo, la Figura 6 muestra la ubicación del archivo creado al jugar *Memory Game*, en esta misma ubicación se crean los archivos HighScoresMultipleChoice.xml y HighScoresVoiceImage.xml una vez que se haya jugado tanto *Multiple Choice* como *Voice and Image*



**Figura 6 Ubicación de los archivos para recuperar los puntajes más altos**

## 6.3 Uso del sistema

En este apartado se presentan uso de las funcionalidades: *Memory Game* y *Ver Scores* de la aplicación *Idioms Game*, desde la perspectiva del jugador. Para observar el detalle la interacción que se tendrá con los juegos *Multiple Choice* y *Voice and Image*, remitirse al documento de Manual de usuario.

### 6.3.1 Jugar *Memory Game*

Al iniciar la sección *Memory Game* se visualizan doce imágenes de un signo de interrogación además de un cronómetro que comienza a contar desde cero (Ver Figura 7 ).



**Figura 7** Pantalla inicial de la sección *Memory Game*

Al dar click en alguna de las imágenes, la aplicación muestra aleatoriamente una imagen que trae consigo un rótulo con el respectivo *idiom* que le corresponde (Ver Figura 8).



**Figura 8 Pantalla al dar click a una imagen**

Al dar click en otra imagen, nuevamente la aplicación muestra una imagen aleatoria.



**Figura 9 Pantalla al dar click a una segunda imagen**

En este caso (Ver Figura 10) las imágenes mostradas no son iguales, por lo que la aplicación ocultará estas imágenes y mostrará nuevamente la imagen con el signo de interrogación hasta volver a dar click a otra ellas.



**Figura 10** Se ocultan las imágenes anteriores y se muestra una nueva

Al encontrar dos imágenes iguales, la aplicación las dejará visibles y el usuario repetirá el mismo procedimiento para intentar encontrar los seis pares posibles para ganar *Memory Game*.



**Figura 11** Pantalla mostrada al encontrar dos imágenes iguales

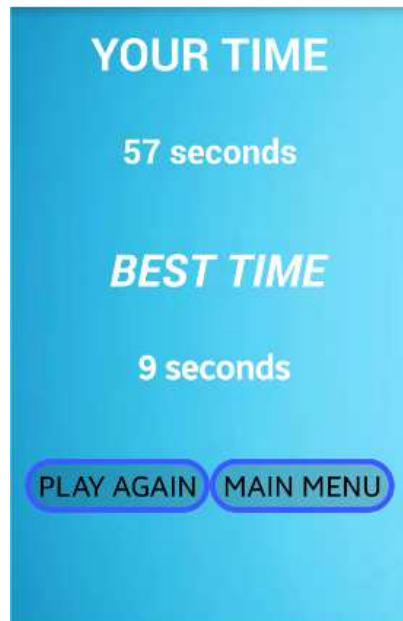
A continuación, la Figura 12 muestra la pantalla que despliega la aplicación al encontrar los seis pares de imágenes.



**Figura 12 Pantalla final de Memory Game**

En la Figura 12 se observa que la aplicación detiene el cronómetro, y además, habilita el botón “Continuar”, al dar click en esta opción, la aplicación dirige a la pantalla “Scores” (Figura 13), la cual muestra al usuario la leyenda YOUR TIME que es el tiempo obtenido al encontrar los seis pares posibles del mini-juego, también se muestra la leyenda BEST TIME que es el menor tiempo obtenido en todas las partidas jugadas de *Memory Game*.



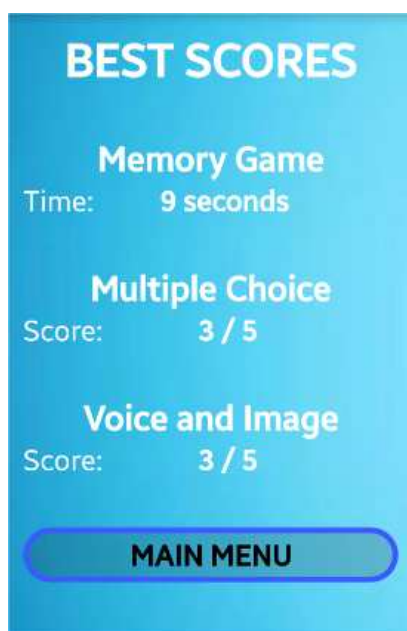


**Figura 13** Pantalla *Scores* de *Memory Game*

Dentro de la pantalla (Figura 13) aparecen dos opciones. La opción **PLAY AGAIN**, permite reiniciar el mini-juego y la opción **MAIN MENU** que dirige a la pantalla inicial donde se encuentra el menú principal de la aplicación.

### 6.3.2 Ver puntajes más altos (Best Scores)

En esta sección se encuentran los mejores puntajes obtenidos en cada mini-juego, es decir, el menor tiempo en *Memory Game* y el mayor número de aciertos tanto en *Multiple Choice* como en *Voice and Image* (Ver Figura 14).



**Figura 14** Pantalla *Scores*

En la sección *Scores* se observa también la opción MAIN MENU, al seleccionarla, permite regresar al menú principal de la aplicación.

## 6.4 Hardware y Software necesario

### 6.4.1 Tecnología para el desarrollo de la aplicación

#### 6.4.1.1 Software

El software utilizado para el desarrollo del proyecto es de licencia libre. A continuación se describe cada uno de ellos.

##### 6.4.1.1.1 Lenguaje de programación

El código de la aplicación fue implementado con el lenguaje de programación orientado a objetos Java, versión 8 [16] y el conjunto de herramientas de desarrollo SDK<sup>3</sup> de Android.

##### 6.4.1.1.2 Google Voice Search

Es una implementación propuesta por Google añadida en el API<sup>4</sup> nivel 3 de Android, se utilizó la versión 4.7.13.19 [17]. Ésta se utilizó para desarrollar el reconocimiento por voz, responde a un RecognizerIntent mostrando un cuadro de diálogo con “Habla ahora” y transmite la información a los servidores de Google para que estos procesen la información y transformen lo pronunciado por el usuario a texto. Para el correcto funcionamiento de la aplicación es necesario, desde Google Voice Search, ajustar el idioma por defecto a inglés, para que los servidores de Google arrojen sólo respuestas en este idioma y de esta manera se logre el cometido de intentar pronunciar correctamente las frases del idioma inglés.

##### 6.4.1.1.3 SharedPreferences

Para la implementación de la aplicación se utilizó SharedPreferences, la cual fue añadida en el API nivel 1 [18]. Las Preferencias Compartidas (traducción en español) no son más que datos que una aplicación debe guardar para personalizar la experiencia del usuario. Cada preferencia se almacenará en forma de clave-valor; es decir, cada una de ellas estará compuesta por un identificador único y un valor asociado a dicho identificador. Además, los datos se guardan en un archivo XML que se almacena en una ruta con el siguiente patrón:

```
/data/data/paquete.java/shared_prefs/nombre_coleccion.xml
```

Al abrir este archivo con cualquier editor de texto veremos un contenido como el siguiente:

```
<?xmlversion='1.0'encoding='utf-8'standalone='yes'?>
<map>
  <intname="aciertos">numeroAciertos</int>
</map>
```

---

<sup>3</sup> SDK por sus siglas en inglés Software Development Kit, es un conjunto de herramientas de desarrollo de software que permite al programador crear aplicaciones.

<sup>4</sup> Interfaz de programación de aplicaciones (API – Application Programming Interface). Conjunto de funciones o métodos que exponen las clases de alguna biblioteca para ser utilizado por otro software.

Para la aplicación, este método fue utilizado para guardar los mejores puntajes obtenidos por el jugador en los mini-juegos, y de esta manera visualizarlos cada vez que se ejecute la aplicación.

#### **6.4.1.1.4 Sistema Operativo**

El Sistema Operativo utilizado durante todo el desarrollo de la aplicación, junto con todas las tecnologías mencionadas anteriormente, fue Windows 7 Professional.

#### **6.4.1.2 Hardware**

La aplicación fue desarrollada e instalada en una Laptop Dell Inspiron 15R N5110 con los siguientes recursos:

- Memoria RAM 6 GB
- Disco duro 500 GB
- Procesador Intel(R) Core(TM) i5-2410M a 2.30 GHz
- Sistema Operativo Windows 7 Professional

Dispositivos de experimentación y pruebas

- *Smartphone Samsung Galaxy S5 SM-G900M*
  - Pantalla de 5.1 pulgadas
  - Procesador Quad-Core de 2.5 GHz
  - Sistema Operativo Android 4.4.2
  
- *Tablet Samsung Galaxy Tab 3*
  - Pantalla de 7 pulgadas
  - Procesador Dual-Core a 1.2 GHZ
  - Sistema Operativo Android 4.2.2

## 6.4.2 Tecnología para la instalación y puesta en marcha de la aplicación

Para instalar la aplicación *Idioms Game* correctamente debemos tener en cuenta dos aspectos principales: primero, asegurarnos de que el dispositivo en el que estará alojada la aplicación sea un Smartphone con sistema operativo Android, con una versión mínima 4.0 *IceCreamSandwich*; segundo, corroborar que se tiene instalada la aplicación *Google Voice Search* en el dispositivo (gran número de dispositivos la tienen instalada por defecto) para responder a las peticiones del jugador cuando éste intente utilizar el reconocimiento de voz. Es importante mencionar que el empleo del reconocimiento de voz implica una conexión con los servidores de Google para que éstos transformen la voz en texto. En caso de que no se tenga instalada la aplicación *Google Voice Search*, se deben de seguir los siguientes pasos:

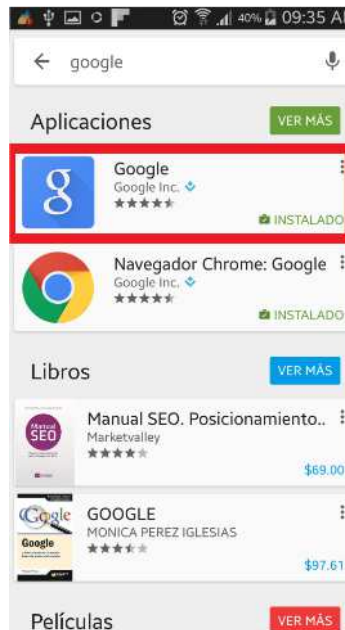
1.- Abrir la *Play Store*<sup>5</sup> y escribir en el buscador: *Google* (Ver Figura 15)



Figura 15 Pantalla para buscar la aplicación Google

<sup>5</sup> Google Play es una tienda de software en línea desarrollada por Google para los dispositivos con sistema operativo Android.

2.- Seleccionar de la pestaña aplicaciones la primera opción. (Ver Figura 16)



**Figura 16 Pantalla que muestra la aplicación que se debe de elegir**

3.- Dar click en INSTALAR, e inmediatamente comenzará la descarga.

4.- Al finalizar, dar click en Abrir. (Ver Figura 17)



**Figura 17 Pantalla que se muestra al finalizar la descarga**

Por último, además de contar con acceso a internet, el dispositivo móvil debe contar con pantalla táctil, pero no se requiere que la pantalla sea multitoque.

## 7 Resultados

La finalidad de este proyecto fue implementar un juego para dispositivo móvil con sistema operativo Android, para practicar la pronunciación de los conocidos como *idioms* del idioma inglés. Como resultado de este proyecto de integración se obtuvo la aplicación *Idioms Game*, la cual consta de las siguientes funcionalidades:

El juego *Memory Game*. Aquí el jugador descubre una imagen y el *idiom* con el que se asocia, pues cada una de las imágenes con las que cuenta la aplicación viene acompañada de un texto que corresponde al *idiom* con el que se relaciona. Además se añadió un cronómetro a la pantalla para que el jugador pueda observar el tiempo que le está tomando consumir el objetivo del juego correctamente.

El juego *Multiple Choice*. Aquí el jugador pone en práctica lo aprendido en el juego *Memory Game*, ya que se le presentan en pantalla las mismas imágenes que en el juego anteriormente mencionado, con una diferencia, que en este juego la imagen no trae consigo el texto que identifica su *idiom* correspondiente, sino que se presentan 4 opciones en donde sólo una es la correcta, la cual debe ser pronunciada con ayuda del micrófono, cuya imagen aparece en pantalla. En este juego, se mantiene un puntaje que aumenta en una unidad cada que el jugador obtiene una respuesta correcta.

El juego *Voice and Image*. Este juego presenta al jugador las mismas imágenes que en el juego *Memory Game*, con una diferencia, que en este juego, no se le presentan opciones que sirvan de ayuda para obtener la respuesta correcta, sino que únicamente se observa la imagen y el botón del micrófono para reconocer la voz. En este juego, se mantiene un puntaje que aumenta en una unidad cada que el jugador obtiene una respuesta correcta.

Herramienta de reconocimiento de voz que permita evaluar la pronunciación del usuario. Se añadió la herramienta de reconocimiento de voz en los juegos *Multiple Choice* y *Voice and Image* para evaluar si el jugador pronuncia correctamente el *idiom*. Esta herramienta tiene la funcionalidad de obtener la pronunciación del jugador y transformarla a texto, para que de esta manera se pueda realizar la evaluación y otorgar al jugador un acierto o un error.

## 8 Análisis y discusión de resultados

Dado que este proyecto pretende fomentar la práctica de la pronunciación del inglés de sus usuarios, se utilizaron los *idioms* que se emplean en conversaciones cotidianas. En consecuencia, al realizar la práctica de estas frases y además asociarlas con imágenes, los usuarios podrán situarlas en el contexto adecuado y eventualmente los aprenderán.

Para comprobar el correcto funcionamiento *Idioms Game* con la integración de la herramienta de reconocimiento de voz (*Google Voice Search*), fue necesario realizar algunas pruebas, las cuales se mencionan a continuación:

Como parte de los juegos *Multiple Choice* y *Voice and Image* se emplea la aplicación *Google Voice Search*, por lo tanto, las pruebas se centraron principalmente en estos dos mini-juegos, pues se requería saber si la transformación a texto se realiza correctamente. En la Figura 18 se muestra una prueba de la funcionalidad del reconocimiento de voz, que devuelve al jugador una lista de palabras que se asemejan a lo pronunciado. En la imagen, aparece un cuadro de diálogo con la leyenda: “Habla ahora”, la cual indica que la aplicación se encuentra lista para escuchar y transformar la voz a texto.

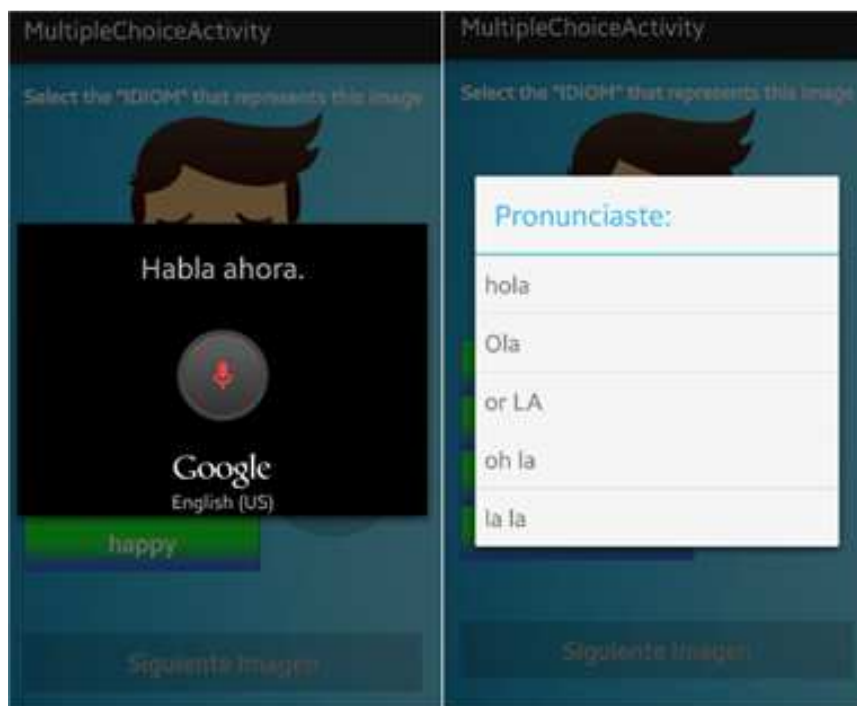
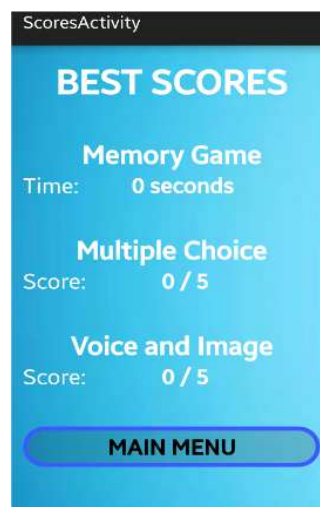


Figura 18 Funcionamiento Idioms Game con la integración de *Google Voice Search*

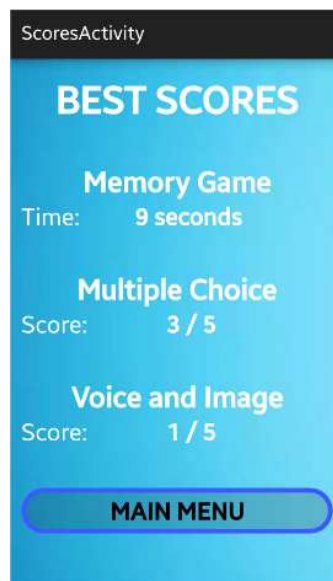


Al realizar las pruebas anteriores, se comprobó que *Idioms Game* funciona correctamente contando con la integración de *Google Voice Search*; para conseguir con éxito el funcionamiento sólo basta con configurar en esta última el idioma al que se quiere convertir la voz del jugador, en este caso, al idioma inglés.

No solo se comprobó el funcionamiento de los mini-juegos que hacen uso del reconocimiento de voz. También se probó el funcionamiento de las preferencias compartidas; es decir, se verificó que los puntajes obtenidos por el jugador se guardan correctamente y que no son borrados cada vez que se ejecuta *Idioms Game*.



**Figura 19** Puntajes al ejecutar por primera vez la aplicación



**Figura 20** Puntajes en la segunda ejecución de la aplicación

Como se observa en la Figura 19, en la pantalla de *Scores*, la primera vez que se ejecuta la aplicación el puntaje en cada mini-juego es cero (no se ha jugado ningún mini-juego). En la Figura 20 se muestra la pantalla *Scores* después de la segunda ejecución de la aplicación (por lo menos se ha jugado 1 vez cada mini-juego) en donde se observa que ahora el puntaje obtenido en cada mini-juego ha cambiado. Por lo tanto, se puede verificar que los puntajes obtenidos por el usuario se guardan correctamente cada vez que el usuario impone una nueva marca.

## 9 Conclusiones

**Objetivo: Desarrollar una aplicación para dispositivos móviles con sistema operativo Android para practicar la pronunciación de las frases denominadas *idioms* del idioma inglés**

Este objetivo se cumplió completamente. La principal funcionalidad de la aplicación es la práctica de la pronunciación, la cual con ayuda de la herramienta de reconocimiento de voz transforma la voz del jugador a texto, la oración en texto es validada con un conjunto de frases existentes dentro de la aplicación, a partir de esta validación, se determina si el jugador ha pronunciado correctamente o incorrectamente.

**Objetivo particular: Diseñar e implementar un módulo para el juego *Memory Game***

Este objetivo se cumplió completamente. En este juego se aplicó la dinámica de un juego de Memorama, el cual pretende agilizar la mente del jugador pues tiene que encontrar dos imágenes iguales recordando el lugar en el que éstas se encuentran. En *Memory Game* se tienen que encontrar 6 pares de imágenes y además sirve como entrenamiento para los siguientes dos juegos; pues en éste, aparece cada una de las imágenes junto con el *idiom* al que se asocia.

**Objetivo particular: Diseñar e implementar un módulo para el juego *Multiple Choice***

Este objetivo se cumplió completamente. En este módulo se implementó el juego *Multiple Choice* en el cual se hizo uso de todo el banco de imágenes del juego *Memory Game*. En *Multiple Choice* la imagen aparece sin ningún texto de ayuda para el jugador. Aquí la imagen aparece junto con 4 opciones, donde sólo una es la correcta. El jugador se auxilia de las opciones para obtener la respuesta correcta. El jugador pronuncia el *idiom* ,y si la pronunciación es correcta se incrementa el puntaje en una unidad.

Objetivo particular: **Diseñar e implementar un módulo para el juego *Voice and Image***

Este objetivo se cumplió completamente. En este módulo se implementó el juego *Voice and Image* en el cual se hizo uso de todo el banco de imágenes del juego *Memory Game*. En *Voice and Image* la imagen aparece sin ningún texto de ayuda para el jugador. Aquí la imagen está acompañada con el botón para accionar el micrófono. El jugador pronuncia el *idiom*, y si la pronunciación es correcta se incrementa el puntaje en una unidad.

Objetivo particular: **Integrar como parte de los juegos una herramienta de reconocimiento de voz que permita evaluar la pronunciación del usuario**

Este objetivo se cumplió completamente. En los juegos *Multiple Choice* y *Voice and Image* se integró la herramienta de reconocimiento de voz *Google Voice Search*, la cual obtiene la voz del jugador y transmite esa información a los servidores de Google para que estos la procesen y transformen lo pronunciado por el usuario a una oración en texto. De esta manera, la oración en texto obtenida, se valida con el conjunto de frases que se tiene dentro de la aplicación *Idioms Game* para discernir entre si el jugador pronunció correcta o incorrectamente.

## 10 Perspectivas del proyecto

Finalmente, como trabajo futuro de este proyecto; proponemos algunas funcionalidades por desarrollar:

- Integrar un módulo que retroalimente al jugador, cuando su pronunciación es errónea, emitiendo la pronunciación correcta de la frase.
- Integrar un módulo para soportar otro tipo de frases del inglés, además de los *idioms*, frases más elaboradas, de uso coloquial en conversaciones nativas del idioma inglés. Se podría integrar la aplicación de *phrasal verbs*, la cual se mencionó en los antecedentes de este documento.

## 11 Referencias Bibliográficas

- [1] C. U. Press, «Cambridge Dictionaries Online,» 10 Abril 2015. [En línea]. Available: <http://dictionary.cambridge.org>. [Último acceso: 10 Abril 2015].
- [2] O. U. Press, «Oxford Dictionaries Language matters,» 10 Abril 2015. [En línea]. Available: <http://www.oxforddictionaries.com/>. [Último acceso: 10 Abril 2015].
- [3] M. S. Ros, «La enseñanza del inglés como lengua extranjera en la titulación de filología inglesa: el uso de canciones de música popular no sexistas como recurso didáctico,» Universidad de Málaga, Málaga, España, 2006.
- [4] M. E. A. y J. Á. Briseño, «Reconocimiento de voz,» Instituto Tecnológico de Aguascalientes, Aguascalientes, México, 2003.
- [5] Duolingo Inc., «Duolingo,» Abril 2013. [En línea]. Available: <https://es.duolingo.com>. [Último acceso: 14 Noviembre 2014].
- [6] A. D. Mendoza, «Srcv: Sistema computacional interactivo basado en reconocimiento de comandos de voz para aplicaciones educativas,» Universidad Autónoma Metropolitana Azcapotzalco, México, 2011.
- [7] J. P. A. O. y J. D. L. Jaimes, «Aplicación colaborativa para dispositivos móviles con sistema operativo Android,» Universidad Autónoma Metropolitana Azcapotzalco, México, 2012.
- [8] J. C. C. García, «Aplicación Android para la práctica de verbos compuestos del idioma inglés,» Universidad Autónoma Metropolitana Azcapotzalco, México, 2013.
- [9] A. N. González, «Xataka Android,» 08 02 2011. [En línea]. Available: <http://www.xatakandroid.com/sistema-operativo/que-es-android>. [Último acceso: 15 05 2015].
- [10] E. M. Hernández, «academia.edu,» Academia.edu, 2011. [En línea]. Available: <http://www.academia.edu/>. [Último acceso: 17 07 2015].
- [11] G. Inc., «Android Developers,» 2015. [En línea]. Available: <https://developer.android.com/guide/components/fundamentals.html>. [Último acceso: 18 07 2015].

- [12] Wikipedia, «Reconocimiento del habla,» 17 02 2015. [En línea]. Available: [https://es.wikipedia.org/wiki/Reconocimiento\\_del\\_habla](https://es.wikipedia.org/wiki/Reconocimiento_del_habla). [Último acceso: 18 07 2015].
- [13] G. Inc, «Google Voice,» Google Inc, 11 03 2009. [En línea]. Available: <https://www.google.com/voice>. [Último acceso: 20 07 2015].
- [14] P. Norvig, «Tecnología y anuncios de Google,» Google, 23 11 2011. [En línea]. Available: <http://tecnologiayproductosgoogle.blogspot.mx/>. [Último acceso: 18 07 2015].
- [15] P. Norvig, «Tecnología y Anuncios de Google,» Google, 11 11 2011. [En línea]. Available: [http://tecnologiayproductosgoogle.blogspot.mx/2011/11/una-mirada-dentro-de-la-tecnologia-de\\_23.html](http://tecnologiayproductosgoogle.blogspot.mx/2011/11/una-mirada-dentro-de-la-tecnologia-de_23.html). [Último acceso: 18 05 2015].
- [16] J. R. Ivar Jacobson, El Proceso Unificado de Desarrollo de Software, Pearson Addison-Wesley, 2000.
- [17] Oracle, «Java,» Oracle, [En línea]. Available: <http://www.java.com>. [Último acceso: 21 06 2015].
- [18] G. Inc., «Google play,» Google, [En línea]. Available: <https://play.google.com/store/apps/details?id=com.google.android.googlequicksearchbox>. [Último acceso: 21 06 2015].
- [19] A. Developers, «Android Developers,» Google, [En línea]. Available: <http://developer.android.com/reference/android/content/SharedPreferences.html>. [Último acceso: 21 06 2015].
- [20] D. Salas, «El androide libre,» 11 08 2011. [En línea]. Available: <http://www.elandroidelibre.com/2011/08/la-historia-y-los-comienzos-de-android-el-sistema-operativo-de-google.html>. [Último acceso: 15 05 2015].
- [21] O. H. Alliance, «open handset alliance,» 2011. [En línea]. Available: <http://www.openhandsetalliance.com/>. [Último acceso: 15 05 2015].
- [22] Awe, «ANDROID.SCENEBETA.COM,» 2014. [En línea]. Available: <http://android.scenebeta.com/tutorial/historia-de-la-plataforma-android>. [Último acceso: 15 05 2015].