



---

**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

**FACULTAD DE INGENIERÍA**

**Desarrollo de una aplicación  
móvil para la enseñanza del  
idioma Coreano**

**TESIS**

Que para obtener el título de  
**Ingeniera en Computación**

**P R E S E N T A**

Guadalupe Cruz Mendoza

**DIRECTOR DE TESIS**

Ing. Marco Antonio Martínez Quintana



Ciudad Universitaria, Cd. Mx., Octubre 2018



# Índice

<b>Introducción</b> .....	1
<b>Capítulo 1</b>	
Planteamiento del problema .....	3
Objetivos .....	3
<b>Capítulo 2</b>	
Marco Teórico .....	5
Dispositivos Móviles .....	6
Aplicaciones Móviles .....	6
Tecnologías de Acceso y Redes .....	7
Historia del Cómputo Móvil .....	7
Hardware Independiente .....	8
Software Independiente.....	9
Sistema Operativo Móvil .....	9
Navegadores en los dispositivos móviles .....	11
Mundo móvil e inalámbrico .....	12
Proveedor de servicio.....	13
Arquitecturas para aplicaciones móviles y envío de mensajes .....	14
Ingeniería de Software .....	16
Etapas del proceso de software .....	16
Modelos de procesos de software .....	17
UML (Leguaje Unificado de Modelado).....	18
Diagramas UML .....	18
Diagramas de Caso de uso.....	19
Relaciones en diagramas de casos de uso.....	20
Análisis y Diseño Orientado a Objetos (ADOO) .....	22
Modelado .....	22
Programación Orientada a Objetos.....	23
Abstracción.....	24
Encapsulamiento .....	24
Herencia .....	25

Dirección de la jerarquía de clases.....	25
Polimorfismo .....	25
M-Learning .....	26
El idioma coreano.....	27
<b>Capítulo 3</b>	
Desarrollo .....	31
Fase de análisis de requerimientos .....	31
Fase de diseño y construcción.....	36
Aplicación del ADOO .....	36
Generación del prototipo de la aplicación .....	36
Diagramas de casos de uso .....	46
Aplicación de la POO a una aplicación móvil.....	51
Fase de implementación .....	53
Fase de pruebas.....	55
Pruebas unitarias.....	55
Pruebas de integración.....	56
Fase de liberación.....	58
<b>Capítulo 4</b>	
<b>Resultados</b> .....	61
<b>Conclusiones</b> .....	63
<b>Anexos</b> .....	65
Diseño de construcción de los trazos del alfabeto coreano.....	65
Código Fuente .....	69
<b>Referencias</b> .....	89

## Introducción

En el presente documento se describe el proceso que se llevó a cabo para la realización de la aplicación Hangul para el aprendizaje del idioma coreano.

Esta aplicación surgió de la necesidad de practicar lo aprendido en clase de una forma sencilla y amena, en la cual se cubran los principales aspectos al iniciarse en este idioma, como lo son la escritura de los trazos, la diferenciación entre vocales y consonantes, así como verbos, vocabulario y frases cotidianas. Todo esto inmerso en una aplicación móvil.

En el capítulo 1 se muestra la problemática inicial y el por qué el desarrollo de esta aplicación, así como los objetivos que se cumplieron en este trabajo.

En el capítulo 2 se muestra el marco teórico con todos los conceptos necesarios para el análisis, diseño, desarrollo e implementación de una aplicación móvil, pasando desde cómo se creó el cómputo móvil, las generaciones que existieron a lo largo de la historia, su clasificación, así como los tipos de arquitecturas que existen para poner en marcha una aplicación móvil. También en este mismo capítulo se habla acerca de una de las tendencias más importantes de la quinta generación del cómputo móvil, el aprendizaje móvil o mejor conocido como M-Learning, el cual es la línea base del desarrollo de la aplicación.

En el Capítulo 3 se describe a detalle las etapas que conformaron el desarrollo de la aplicación móvil, pasando desde el análisis y obtención de requerimientos, hasta la realización de pruebas para cada parte de la aplicación.

Después, en el Capítulo 4 constato que los objetivos planteados al inicio de este documento se cumplieron, ya que la aplicación fue hecha desde cero, para cubrir con todas y cada una de las necesidades identificadas en la etapa de análisis de requerimientos.

Por último, comparto mi conclusión acerca del desarrollo de la tesis con la cual me presento como Ingeniera en Computación de la Facultad de Ingeniería de la UNAM ante la sociedad y el mundo entero como una ingeniera capaz de resolver problemas aplicando todos y cada uno de los conocimientos adquiridos en las aulas y a través de la experiencia compartida por mis profesores de esta Universidad.



# Capítulo 1

## Planteamiento del problema

Debido al avance tecnológico y a la nueva tendencia del aprendizaje en movimiento o M-Learning se decidió desarrollar una aplicación móvil para la enseñanza del idioma coreano para los hispanohablantes, dado que las herramientas que actualmente existen no cuentan con las características necesarias para llevar un progreso paulatino del aprendizaje y/o solo cubren algunos aspectos básicos del idioma.

Así también uno de los principales retos que enfrenta el desarrollo de apps es que no siempre están disponibles para ciertas regiones o incluso para todos los dispositivos, es por eso que a lo largo de este documento se detallan todos y cada uno de los pasos que se siguieron para esta aplicación que nos ayuda a iniciarnos en el mundo del idioma coreano.

## Objetivos

- Recopilar información acerca de la enseñanza del idioma coreano
- Encontrar los principales tópicos necesarios a cubrir por la aplicación móvil
- Diseñar una aplicación móvil que cubra los principales aspectos necesarios para el aprendizaje del idioma coreano
- Desarrollar la aplicación móvil en el sistema operativo Android
- Realizar pruebas a la aplicación móvil



## Capítulo 2

### Marco Teórico

El desarrollo de la tecnología en los últimos años ha tenido un crecimiento acelerado debido al descubrimiento y evolución de los materiales eléctricos, teniendo un gran impacto en el campo de la microelectrónica, permitiendo diseñar y fabricar dispositivos de dimensiones cada vez más pequeñas.

Los primeros pasos de la comunicación móvil comenzaron con la comunicación oral y hoy en día destaca por su capacidad de organizar grupos de trabajo a distancia, mensajería instantánea, compartir información, acceder a datos e información de manera remota, entre otros.

El cómputo móvil es aquel que no necesita de cables y para su funcionamiento hace uso de redes inalámbricas o satélites. También se entiende como cómputo móvil a los equipos o dispositivos portables, que a través de su hardware y software hacen uso de un gran poder de cómputo.

El cómputo móvil está compuesto por tres elementos básicamente:

- Dispositivos móviles
- Aplicaciones móviles
- Tecnologías de acceso y redes

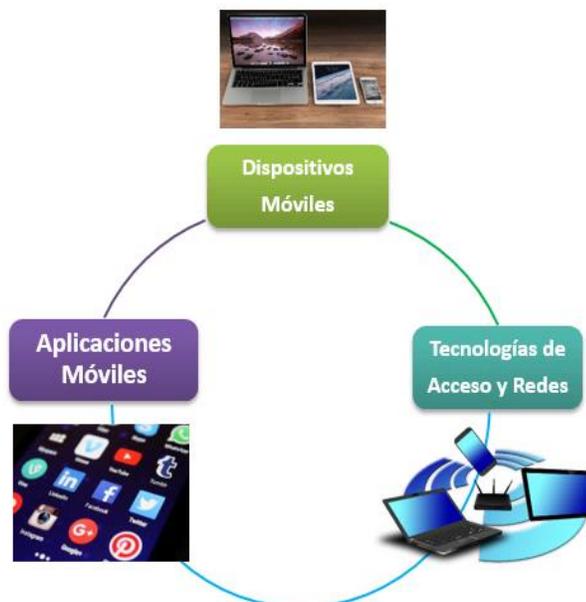


Figura 1.- Elementos del cómputo móvil

## **Dispositivos Móviles**

Un dispositivo móvil se define como un aparato tecnológico de dimensiones generalmente pequeñas, con capacidad de procesamiento, conexión a alguna red (generalmente inalámbrica) y memoria, que está diseñado para realizar algún propósito en específico.

Algunos ejemplos de dispositivos móviles son:

- PDAs (Personal Digital Assistant)
- Teléfonos móviles
- Computadoras portátiles
- Tablets

## **Aplicaciones Móviles**

Una aplicación móvil es un programa que se ejecuta en un dispositivo móvil y se desarrolla específicamente para adecuarse visual y funcionalmente a las características físicas del dispositivo sobre el cual tiene que ejecutarse.

Algunas categorías de aplicaciones móviles son:

- Entretenimiento y ocio
- Clima
- Música
- Noticias
- Redes sociales
- Vídeos y películas
- Educación
- Salud

## **Tecnologías de Acceso y Redes**

Es el canal a través del cual se realiza la comunicación entre dos o más dispositivos y permiten el acceso a la información de forma remota.

Algunos ejemplos son:

- Red celular
- WiFi
- Bluetooth
- RFID

## **Historia del Cómputo Móvil**

Hace muchos años las personas tenían dificultades para comunicarse hacia otros lugares mientras se desplazaban de un lugar a otro, ya que para realizar una llamada o dejar un mensaje mientras se trasladaban tenían que ubicar un teléfono público para poder comunicarse. Hoy en día las personas pueden comunicarse fácilmente desde casi cualquier parte del mundo.

El cómputo móvil inicio a principios de la década de los 50's con la compañía Motorola quien creó el Handie Talkie H12-16, el cual permitía el contacto vía ondas de radio cuya banda de frecuencia no superaba los 60 MHz, esto marca la revolución en la historia de los componentes móviles y por ello nace la primera generación (1G).

En los años 90 nace la segunda generación (2G) con lo cual inicia la era de la digitalización que permite una mejor calidad en las llamadas. También se caracteriza por dispositivos de menor tamaño, se impone el estándar GSM y se llega a la posibilidad de enviar mensajes de texto SMS. con la cual llega la digitalización, esto permite una mejor calidad en las llamadas.

A finales de la década de los 90's se inicia la primera versión del protocolo de aplicaciones móviles inalámbricas (WAP) que permite acceso a servicios de internet desde el teléfono móvil.

Con la generación 2.5G se destaca por un sistema de mensajes mejorados (EMS) y la opción de mensajes multimedia (MMS) y se experimenta un incremento en la velocidad de transmisión de datos, gracias a las tecnologías como GPRS y EDGE.

La tecnología de tercera generación (3G) presenta una evolución imprescindible al integrar las opciones de telefonía y los servicios de internet de manera cómoda y fluida, desde cualquier lugar.

La generación 4G se comienza con el uso de datos y voz que dan base a la transmisión del video HD y la televisión móvil de alta definición.

La base para las actuales tecnologías táctiles fueron los dispositivos PDA (Asistente Digital Personal) que poseían un calendario, directorio de contactos, software para almacenar notas y conexión a la red telefonía mediante un accesorio llamado “módem”.

La evolución de los dispositivos móviles ha sido progresiva ya que en un principio eran de gran tamaño y peso, tenían poca capacidad de almacenamiento y procesamiento. Hoy en día existe una gran cantidad de modelos con mejores características, por ejemplo: son pequeños, poseen mejor capacidad de almacenamiento y procesamiento, tienen conexión inalámbrica, capacidad de interacción mediante pantalla o teclado, son personalizables, baterías de mayor duración entre otros.

Con base a lo anterior surgió el teléfono inteligente mejor conocido como smartphone que es un teléfono móvil, pero con mayor capacidad de computación, casi semejante a la de una computadora.

Una de las características destacables de los smartphones es la posibilidad de instalar aplicaciones más avanzadas desarrolladas por algún fabricante o por terceros, para mejorar funcionalidades o adicionar alguna.

Hoy en día estamos entrando a la quinta generación del cómputo móvil que involucra el manejo de inteligencia artificial, IoT, Big Data y M-Learning.

## **Hardware Independiente**

El hardware que poseen los dispositivos móviles tiene características especiales definidas por los fabricantes, y cada uno de ellos desarrolla nuevas tecnologías que implementa en los diferentes modelos que producen.

Para el funcionamiento del dispositivo móvil generalmente se tienen las siguientes partes:

- Placa base
- Antena
- Antena WiFi
- Antena NFC
- Pantalla

- Micrófono
- Bocina
- Batería
- Puerto de carga de energía

## Software Independiente

El software se basa de acuerdo con el sistema operativo del dispositivo móvil, ya que ese define varias características importantes como la compatibilidad y acceso a los recursos del hardware del dispositivo. Por ejemplo, el acelerómetro, el giroscopio, las tarjetas de proximidad, entre otros.

## Sistema Operativo Móvil

Es el software principal encargado de gestionar todos los recursos para que se usen de manera eficiente y sin presentar alguna interrupción.

Algunos sistemas operativos móviles son:

- **Android:** Fue adquirido por Google y la Open Handset Alliance con la finalidad de satisfacer las necesidades de los dispositivos móviles con pantalla táctil como smartphones, tablets, notebooks entre otros.
- **iOS:** Es el sistema que Apple creó para sus dispositivos móviles y es el segundo más utilizado.
- **Symbian:** Es un sistema con una larga tradición en el mundo de los móviles las empresas Nokia, Sony Ericsson y Motorola desarrollaron por muchos años dispositivos apoyándose en la confiabilidad de Symbian.
- **Windows Phone:** Desarrollado por Microsoft para dispositivos móviles. Ha sido adoptado por diversas empresas de móviles de todo el mundo. Previo a Windows Phone se desarrollaron Windows CE y Windows Mobile.
- **BlackBerry OS:** Se desarrollo a finales de los 90's, es la denominación del sistema operativo instalado en la línea de teléfonos móviles de BlackBerry.
- **webOS:** Es un sistema creado por Palm en el año 2009, posteriormente paso a manos de HP bajo el nombre de hp webOS. En el año 2011 se anunció que no se continuaría con la fabricación de dispositivos que lo incluyan, sin embargo, se mantiene el soporte y se ha modificado para que sea software libre.

- **Firefox OS:** Es el nombre con el que se ha lanzado el SO de Mozilla Corporation. Es desarrollado bajo el modelo código abierto y se basa en el núcleo de Linux, apoyándose en las virtudes del motor Gecko. Se destaca por ser una opción liviana para dispositivos de hardware limitado. Se basa en los estándares Web y corre aplicaciones creadas en HTML5.
- **Tizen:** Es el sistema operativo desarrollado por Samsung. Las interfaces de desarrollo de Tizen están basadas en HTML5 y otros estándares web.

Según StatCounter que es un servicio de análisis web, menciona que el sistema operativo móvil más usado nacional e internacionalmente es Android.

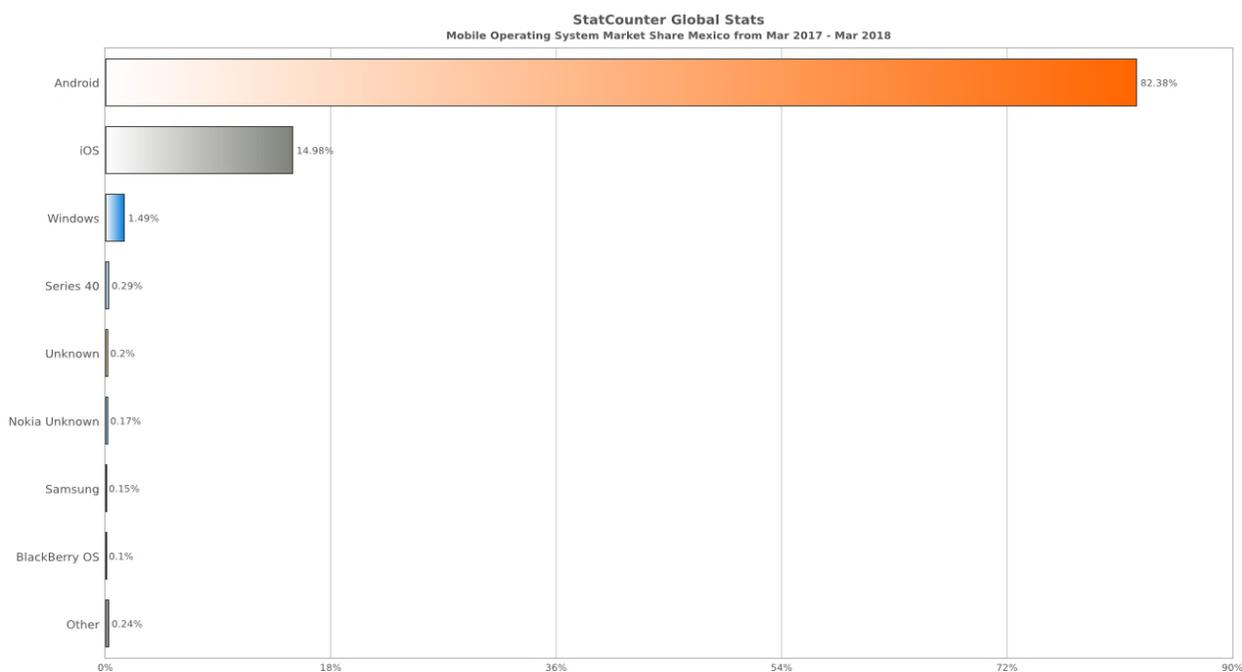
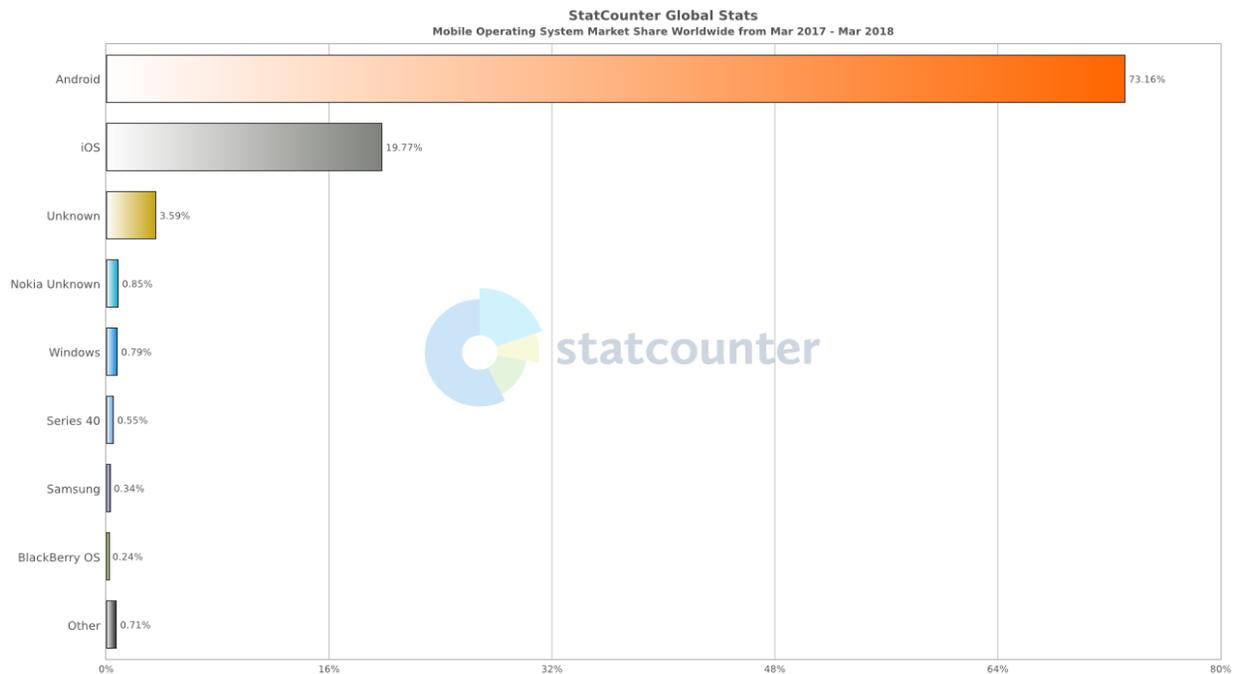


Figura 2.- Sistemas operativos móviles más usados en México



*Figura 3.- Sistemas operativos móviles más usados en el mundo*

## Navegadores en los dispositivos móviles

El uso de navegadores en los dispositivos móviles no es exclusivo del tipo de sistema operativo, existen varias opciones como:

- Google Chrome
- Safari
- Internet Explorer Mobile
- Opera Mini
- Firefox

## Mundo móvil e inalámbrico

Una red inalámbrica es una red en la que dos o más terminales se pueden comunicar sin la necesidad de una conexión por cable.

Con las redes inalámbricas, un usuario puede mantenerse conectado cuando se desplaza dentro de una determinada área geográfica.

Las redes inalámbricas se clasifican en varias categorías, de acuerdo con el área geográfica desde la que el usuario se conecta a la red.

- Redes de corto alcance: Son aquellas que operan sobre espectro sin licencia (2.4, 5 o 40 GHz). Conecta escuelas, edificios, casas, etc.
  - WPAN (Wireless Personal Area Network):
    - Área de cobertura: Espacio personal de 10 metros.
    - Función: Tecnología de reemplazo de cables
    - Costo asociado: Muy bajo
    - Velocidad típica: 0.1 a 4 Mbps
    - Estándares que acepta: IrDA, Bluetooth, 802.15, Zigbee
  - WLAN (Wireless Local Area Network):
    - Área de cobertura: Edificios, campus de 100 metros
    - Función: Extensión o alternativa de las LAN's con cable
    - Costo asociado: Bajo-medio
    - Velocidad típica: 1-100 Mbps
    - Estándares que acepta: 802.11 a/b/g/n, HIPERLAND/2
  - WMAN WiMax (Worldwide Interoperability for Microwave Access):
    - Es conocido como la familia 802.16
    - Rango de más de 100 Km
    - No hay espectro uniforme globalmente aceptado
    - Su uso en las comunicaciones móviles de alta velocidad (4G) y para conectar Hotspots WiFi entre ellos.
- Redes de largo alcance: Operan sobre espectro con licencia. Utilizan proveedores de servicio (carriers u operadores) y conectan áreas metropolitanas, estados y países.
  - WWAN (Wireless Wide Area Networks):
    - Área de cobertura: Nacional, proporcionada por los operadores.
    - Función: Extensión de LAN's

- Velocidad típica: 8 Kbps a 10 Mbps
  - Estándares que acepta: GSM, CDMA, EV-DO, UMTS
  - Costo asociado: Medio-alto
- Redes Satelitales:
    - Área de cobertura: Continental o global.
    - Función: Extensión de LAN's
    - Velocidad típica: 2 Kbps a 2 Mbps
    - Estándares que acepta: TDMA, CDMA, FDMA, Iridium

## **Proveedor de servicio**

Es una empresa que presta servicios a otras empresas. El negocio más habitual de los proveedores de servicios es la oferta de suscripciones o contratos. Por ejemplo, la telefonía móvil, el acceso a Internet y el alojamiento de sitios web son algunos de los negocios.

Un proveedor de telefonía móvil es una compañía de teléfono que proporciona servicios para los usuarios de teléfonos móviles. El operador le da una tarjeta SIM para el cliente, que se inserta en el teléfono móvil para acceder al servicio.

Existen dos tipos de operadores de telefonía móvil:

- Un operador de red móvil MNO que es propietario de los activos de la red y del espectro subyacente necesario para ejecutar el servicio.
- Un operador de red móvil virtual (MVNO) que compra al por mayor el servicio de un MNO y vende a sus propios clientes.

Un proveedor de servicios de Internet (ISP) son las empresas y organizaciones que proporcionan a los usuarios el acceso a Internet y servicios relacionados.

Estos proveedores conectan a los clientes de otros proveedores de servicio por medio de redes. A menudo, son empresas que proporcionan servicios de telecomunicaciones de datos y de conexión telefónica.

## Arquitecturas para aplicaciones móviles y envío de mensajes

Las arquitecturas implementadas en el desarrollo de aplicaciones móviles, se basan de acuerdo al tipo de información que se maneja en ellas, dentro de las cuales podemos encontrar las siguientes:

- Autocontenidas
- Internet inalámbrico
- Clientes inteligentes (SMART)
- Arquitectura WAP

Una arquitectura autocontenida tiene como principal característica que su contenido es estático, es decir, sus imágenes, su información, sus menús entre otros, nunca cambian. Esto debido a que todo su contenido se encuentra autocontenido dentro de la misma aplicación, como es el caso de la calculadora, la cual está presente en todos los sistemas operativos y no necesita actualizarse.

Una arquitectura de internet inalámbrico cuenta con tres elementos:

- Un micronavegador o microbrowser
- Un servidor web
- Una base de datos

El micronavegador lo ocupa el dispositivo para conectarse al servidor web a través de una dirección URL. Esta URL contiene una petición que el servidor web procesa y el contenido de la respuesta es enviada al dispositivo móvil para ser adaptada y poderla visualizar correctamente.

El servidor web que generalmente está en la nube es aquel que escucha e interpreta las peticiones hechas por el cliente, los procesa y envía la respuesta apropiada.

Y por último la base de datos es la que contiene de manera ordenada e integra los datos usados por la aplicación.

Una arquitectura SMART o de cliente inteligente es aquella en la que las aplicaciones móviles pueden ser ejecutadas en cualquier momento, e incluso aun cuando la conexión inalámbrica o el servicio de telefonía no está disponible.

Cuando existe conexión, ellas se comunican con sus servidores, y se establece un proceso de sincronización entre cliente y servidor. Se conforma de tres componentes:

- La aplicación móvil específica
- Un servidor de sincronización
- Una base de datos

Entre algunos ejemplos de dicha arquitectura podemos encontrar aplicaciones como WhatsApp y Gmail, las cuales no necesitan conexión a internet o datos para ver su contenido.

Por último, una arquitectura WAP (Wireless Application Protocol) es un protocolo basado en los estándares de Internet que ha sido desarrollado para permitir a teléfonos celulares navegar a través de Internet.

Con la tecnología WAP se pretende que desde cualquier teléfono celular WAP se pueda acceder a la información que hay en Internet, así como realizar operaciones de e-commerce mejor conocido como comercio electrónico.

## **Ingeniería de Software**

La ingeniería de software es una disciplina cuyo objetivo es el desarrollo y mantenimiento de software a través de técnicas y métodos especializados, algunos de ellos se mostrarán de manera general en el presente documento.

El software se define como un conjunto intangible de programas y datos de la computadora. Este se clasifica en tres grandes rubros:

- Aplicaciones standalone
- Aplicaciones web
- Aplicaciones móviles

La primera hace referencia al desarrollo de aplicaciones propias de un sistema operativo para PC, como lo son, la paquetería de Office, el calendario, la calculadora, entre otros.

La segunda hace referencia a la arquitectura cliente servidor en la cual tenemos uno o varios servidores que contienen la aplicación y que puede ser accedida a través de un cliente haciendo uso de los navegadores web.

Y por último las aplicaciones móviles son aquellas que están diseñadas para dispositivos móviles (smartphones, tablets, sistemas embebidos, wearables, entre otros) y su desarrollo es específico para cada dispositivo, ya que dependen del sistema operativo que se esté manejando.

## **Etapas del proceso de software**

Son los pasos que permiten desarrollar y obtener un software de alta calidad iniciando con el análisis de requerimientos hasta la liberación de este. A continuación, se muestra todas y cada una de las etapas del desarrollo de software:

- Análisis de requerimientos
- Diseño
- Desarrollo
- Pruebas
- Liberación

En la etapa de análisis de requerimientos se recopilan todos los requisitos que debe de llevar un software, una de las técnicas más utilizadas en esta etapa es la entrevista.

Para el desarrollo específico de aplicaciones móviles la entrevista es un elemento crucial al analizar los requerimientos ya que va de la mano con la experiencia del usuario.

En la etapa de diseño tenemos una de las principales actividades para el desarrollo de software, ya que de este depende el éxito del mismo. Esta consiste en el desarrollo de prototipos para dar una visión general de cómo quedará el software en su versión final y para acordar con el cliente si concuerda con los requisitos recabados. En una aplicación móvil existen varios términos para esta etapa como los mockups, los prototipos y el maquetado, los cuales consisten en hacer un boceto de cada pantalla de la aplicación móvil.

En la etapa de desarrollo se eligen las tecnologías necesarias para llevar a cabo el desarrollo del software acorde al diseño anteriormente descrito. En las aplicaciones móviles se utilizan los IDE's para el desarrollo de dichas aplicaciones, ya que la librería necesaria para el desarrollo de estas es muy amplia.

En la etapa de pruebas, se realizan verificaciones del software a diferentes niveles de precisión. En el caso de las aplicaciones móviles hay dos pruebas necesarias antes de liberar una aplicación, las cuales son:

- Pruebas unitarias
- Pruebas de integración

Las primeras consisten en verificar el funcionamiento adecuado de cada pantalla de la aplicación de manera aislada, mientras que las segundas verifican el árbol de flujo que debe seguir la aplicación al interactuar con ella.

Por último, la etapa de liberación consiste en poner en marcha el software (la aplicación) en él o los dispositivos destino, para su propósito final.

## **Modelos de procesos de software**

Un modelo de procesos de software define cómo se dará solución a la problemática del desarrollo del software.

Existen diversos modelos de procesos de software como el de cascada, prototipos, espiral, incremental, RAD (Rapid Application Development), entre otros.

Para el desarrollo de la aplicación móvil se utilizó el modelo incremental, específicamente el relacionado con el análisis y diseño orientado a objetos (ADOO), el cual se explica a detalle en otra sección.

## UML (Leguaje Unificado de Modelado)

Es una herramienta que permite el desarrollo de sistemas. Esto se debe a que permite a los creadores de sistemas generar diseños que expresan de manera fácil sus ideas para comunicarlas a otras personas. Es decir, es una técnica para la especificación de sistemas en todas sus fases.

El UML es necesario porque permite mostrar de manera sencilla y estandarizada el modelado de un sistema, para este caso, una aplicación móvil.

El UML es una notación gráfica y textual, rica y expresiva, que permite textos si las gráficas fueran difíciles de entender. No tiene un proceso, no se aplica como una receta de cocina y cada diagrama de UML brinda una vista distinta del sistema en cuestión. El UML no tiene un ciclo de vida, se utiliza en un ciclo iterativo e incremental.

## Diagramas UML

Los diagramas son las gráficas que describen el contenido de una vista.

UML tiene nueve tipos de diagramas que son utilizados en combinación para proveer todas las vistas de un sistema:

- Diagrama de Casos de Uso
- Diagrama de Clases
- Diagrama de Objetos
- Diagramas de Comportamiento
  - Diagrama de Estados
  - Diagrama de Actividad
- Diagramas de Interacción
  - Diagrama de Secuencia
  - Diagrama de Colaboración
- Diagramas de implementación
  - Diagrama de Componentes
  - Diagrama de Despliegue

Para el desarrollo de la aplicación móvil se usaron los diagramas de casos de uso.

## Diagramas de Caso de uso

Un diagrama de caso de uso representa la funcionalidad completa tal y como la percibe un actor.

Los elementos de un diagrama de caso de uso son:

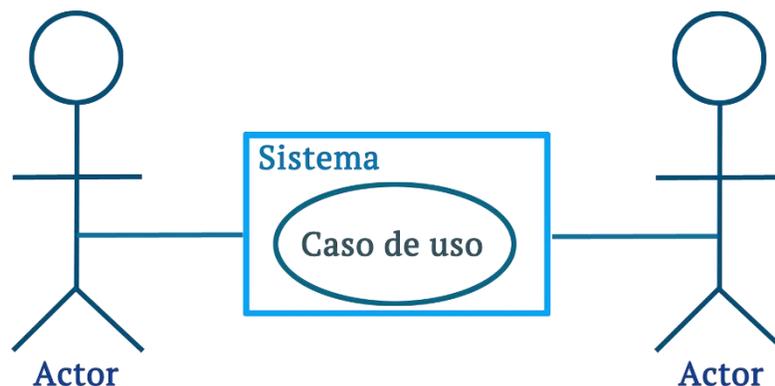
- Actor
- Sistema
- Caso de uso

El actor es alguien o algo que interactúa con el sistema; es quien utiliza el sistema.

Por la frase "interactúa con el sistema" se debe entender que el actor envía a o recibe del sistema unos mensajes o intercambia información con el sistema.

Un sistema es descrito como una caja; el nombre del sistema aparece arriba o dentro de la caja.

Un caso de uso es definido como un conjunto de acciones que un sistema ejecuta y que permite un resultado observable por un actor en particular.



*Figura 4.- Representación gráfica de un diagrama de caso de uso*

## Relaciones en diagramas de casos de uso

Los casos de uso que se definen a lo largo del proceso de desarrollo del software no son independientes, sino que es posible establecer relaciones de dependencia entre ellos.

Las principales relaciones consideradas por UML son:

- **Include (inclusión):** Se presenta cuando relacionamos dos casos de uso, en el cual estamos diciendo que el primero (el caso de uso base) incluye al segundo (el caso de uso incluido). El segundo es parte esencial del primero. Sin el segundo, el primero no podría funcionar bien; pues no podría cumplir su objetivo.

Su representación gráfica es:

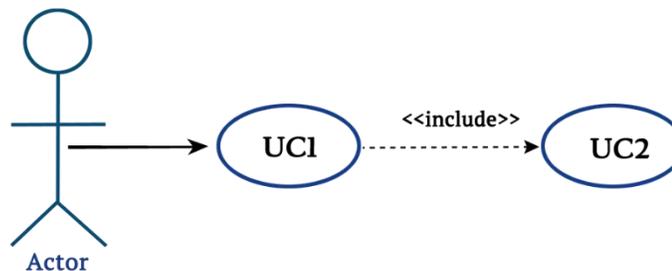


Figura 5.- Representación gráfica de una relación include

- **Extend (extensión):** Es una relación que amplía la funcionalidad de un caso de uso, hay situaciones en que el caso de uso de extensión no es indispensable que ocurra, y cuando lo hace ofrece un valor extra (extiende) al objetivo original del caso de uso base.

Su representación gráfica es:

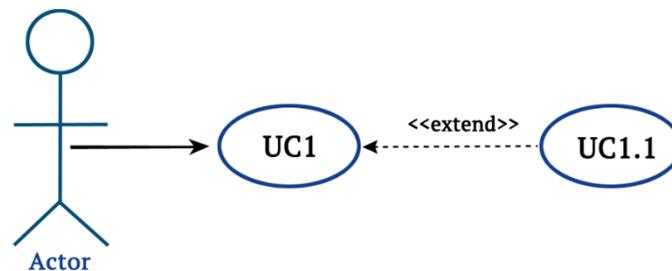


Figura 6.- Representación gráfica de una relación extend

Debemos de recordar que al modelar el diagrama de casos de uso no buscamos analizar el detalle, y mucho menos los flujos.

Todo ese detalle lo podremos plasmar en otro tipo de diagramas, como los diagramas de interacción, de actividad, de estados, o simplemente un texto en una especificación.

Los diagramas de casos de uso se utilizan principalmente para mostrar en un resumen ejecutivo la estructura y funcionalidad de un sistema, ya sea al líder de proyecto o a los desarrolladores, estos últimos sólo para dar conocer un panorama general del mismo.

## **Análisis y Diseño Orientado a Objetos (ADOO)**

El análisis y diseño orientado a objetos es una técnica para modelar sistemas que se encarga de describir y modelar el sistema como si fuera un conjunto de objetos relacionados que interactúan entre sí.

Los métodos orientados a objetos centran su atención en objetos y clases:

- Un objeto es “una cosa”.
- Una clase es “una plantilla o definición de cosas”.

Los métodos orientados a objetos crean modelos con los cuales:

- Se crean clases y objetos
- Definen su estructura, comportamiento y propósito
- Definen la relación entre clases
- Definen la relación entre objetos

## **Modelado**

A finales de los años 90, el modelado empezó a ser una parte fundamental dentro de las actividades del desarrollo de software. En la cual se construyen modelos para comunicar la estructura y el comportamiento deseado del sistema.

El objetivo del software es centrar y optimizar al máximo el código, en otras palabras, es escribir menos líneas y conseguir el mejor rendimiento sin que disminuya la eficacia.

Los principios del modelado son:

1. El primer principio se basa principalmente en cómo abordar el problema y cómo se da la forma de solución.
2. El segundo principio nos dice que todo modelo puede ser expresado a diferentes niveles de precisión. Un analista o un usuario final se centrarán en el “qué”, mientras que un desarrollador se centrará en el “cómo”.
3. El tercer principio establece que los mejores modelos están ligados a la realidad.
4. El cuarto principio establece que un único modelo no es suficiente, de manera que cualquier sistema se aborda mejor a través de un pequeño conjunto de modelos casi independientes.

Así mismo cumple cuatro objetivos:

1. Ayudar a visualizar cómo es o debería ser un sistema.
2. Especificar la estructura o el comportamiento de un sistema.
3. Proporcionar plantillas que guíen la construcción de un sistema.
4. Documentar las decisiones adoptadas.

## Programación Orientada a Objetos

La Programación Orientada a Objetos (POO) se compone de cuatro pilares:

- Abstracción
- Encapsulamiento
- Herencia
- Polimorfismo



*Figura 7.- Pilares de la POO*

A continuación, se describen los pilares de la programación orientada a objetos:

## **Abstracción**

Es el primer pilar de la POO, se caracteriza por describir una entidad en forma simple, a través de sus propiedades principales, omitiendo los detalles.

Existen dos tipos de abstracción:

- Abstracción de datos
- Abstracción funcional

La primera se refiere a las características y las propiedades de un objeto.

La segunda hace referencia a las acciones y/o eventos que hace el objeto.

## **Encapsulamiento**

Es el segundo pilar de la POO y se refiere a ocultar información de los detalles de la implementación, porque mantienen ocultos los procesos internos evitando acceder a datos por otro medio distinto a los determinados, lo que permite dar integridad y calidad al software.

Se compone de tres elementos:

- Interfaz pública
- Implementación
- Información interna

La primera se refiere a las operaciones que podemos hacer con un objeto para interactuar con él.

La segunda se refiere a las operaciones internas que realiza un objeto para lograr su propósito.

Y la última se refiere a los datos que ocupa el objeto de manera interna para completar su función.

En una aplicación móvil el encapsulamiento es importante porque nos ayuda a definir qué elementos estarán disponibles en la interfaz gráfica de usuario (GUI).

## **Herencia**

Es el tercer pilar de la POO la cual puede crear una nueva clase u objeto heredando los atributos y métodos de una clase padre.

Las nuevas clases que se van creando por herencia van estableciendo una jerarquía de clases.

En este contexto se usa el termino superclase para describir cualquiera de las clases de orden superior en una misma jerarquía.

La herencia es adecuada para la reutilización de código además de ser muy utilizada en las aplicaciones móviles para usar elementos principales como botones, textview, recyclerview, entre otros.

## **Dirección de la jerarquía de clases**

Expresa si se generaliza o se especializa una clase.

La generalización identifica y define atributos y operaciones comunes en una colección de objetos, de esta manera se reduce la redundancia en el proceso de desarrollo, lo que ayuda a la reutilización.

La especialización crea nuevas clases más específicas que heredan de una clase mayor.

## **Polimorfismo**

Es el último pilar de la POO, la palabra polimorfismo se deriva del griego polys (muchos), morfo (formas) e -ismo (relativo a). En el enfoque de la POO al implementar una operación de herencia en una subclase se puede realizar acciones de diferentes formas.

## M-Learning

El aprendizaje móvil mejor conocido por sus siglas en inglés como M-Learning, es un procedimiento que facilita la enseñanza y aprendizaje en casi cualquier momento y lugar a través de pequeños dispositivos móviles como smartphones, tablet's, entre otros.

Algunas ventajas que ofrece el M-Learning son:

- Multifuncionalidad
- Conectividad
- Portabilidad
- Uso personal
- Flexibilidad
- Ubicuidad

M-Learning no sustituye a la enseñanza tradicional, por el contrario, tiene como objetivo apoyar en el proceso de enseñanza-aprendizaje, mediante la interacción del usuario con herramientas tecnológicas, en este caso, las aplicaciones móviles.

Un de las subramas del M-Learning es el aprendizaje de un idioma asistido a través de un dispositivo móvil llamado MALL por sus siglas en inglés (Mobile-Assisted Language Learning) que usa la tecnología móvil en el aprendizaje de idiomas. En este tipo de aprendizaje no es necesario sentarse en un aula o en una computadora para acceder a los materiales de aprendizaje.

Otra de las subramas es el aprendizaje de lenguas asistido por computadora llamado CALL por sus siglas en inglés (Computer-Assisted Language Learning). En este tipo de aprendizaje se debe usar la computadora para hacer uso de los materiales de aprendizaje.



*Figura 8.- M-Learning*

## El idioma coreano

La creación del alfabeto en un país es esencial para su desarrollo cultural, social e identidad nacional. Existen diferentes tipos de acontecimientos que dejan huella en el país, para Corea del Sur su gran motivo de celebración cada año es la invención de un sistema de escritura llamado 한글 (*hangul* o, en ocasiones, *hangeul*).

El nacimiento del hangul es conocido como el milagro del siglo XV por que en Corea eran usados los caracteres chinos para expresar por escrito la lengua coreana.

Entre la cultura china y coreana existía una estrecha relación entre sus lenguas y sus sistemas de escritura. China fue la primera civilización que surgió en Asia Oriental por lo cual tuvo una gran influencia en países vecinos. Corea fue integrando elementos de la cultura y organización social proveniente de China. Y por consecuencia fue la introducción de la lengua escrita China en Corea, lo cual al paso de varios siglos empezó a preocupar al Rey Sejong, ya que la gente común era incapaz de aprender los complicados caracteres chinos, dichos caracteres se usaban para transcribir la lengua coreana y fueron modificados a lo largo de la historia para facilitar la transcripción de los sonidos del lenguaje oral. Pero ninguna de estas modificaciones facilitó el acceso de la mayoría del pueblo a la cultura. Lo que provoca al pueblo expresar sus sentimientos y pensamientos por escrito, ya que la escritura china era usada por los eruditos del reino, pero no era fácil expresar el significado del pensamiento coreano por medio de una escritura adoptada de otra lengua cuyos sonidos, escritura y gramática eran completamente distintas.

Para solucionar este problema, el rey pensó en la necesidad de crear un alfabeto originalmente coreano, adaptado a la realidad cultural de su pueblo y que fuera fácil de aprender por todos sus súbditos.

En el año de 1444 el rey Sejong publicó algunos libros escritos con las nuevas letras del alfabeto, en el que destaca un libro de cantos titulado “Los dragones voladores”.

La lengua coreana tomó vocabulario de la lengua china en un gran número, adaptando las raíces chinas a los hábitos coreanos de pronunciación. Esto hace que en el vocabulario coreano existan alrededor del 50% de palabras provenientes del idioma chino.

El primer alfabeto coreano fue finalmente promulgado en 1446 bajo el nombre de 'Hoon Min Jung Um (훈민정음)' por el Rey Sejong. Significa "El verdadero sonido (letras) que se utilizará para enseñar a las personas".

Sin embargo, los nobles de la época no acogieron la llegada de Hangeul, ya que tenían inculcados la idea de que China es el centro del mundo, por ese motivo mantuvieron una actitud de desprecio al Hangeul en comparación con los caracteres y escrituras chinas.

Por lo tanto, los caracteres y las escrituras chinas se llamaban 'Jinseo (진서)', que significan 'letras verdaderas', mientras que Hangeul a menudo se llamaba 'Unmun (언문)' y hay algunas opiniones de que el Hangeul se llamaba 'Amkul (암글)' porque fue utilizado principalmente por mujeres, o 'Ahatkul (아햇글)' porque era solo para el uso de niños que no aprendían caracteres chinos. Sin embargo, no podemos confirmar si los nombres fueron realmente utilizados porque no hay un registro claro del uso de los nombres 'Amkul' o 'Ahatkul'.

Durante el período de la civilización, nombres como 'Jung Um (정음)' y 'Gook Mun (국문)' se usaban a menudo debido a la alta valoración de Hangul (letras nativas de Corea), ya que la gente comenzó a apreciar el espíritu nacional.

Joo Shi Kyung parece ser el primero que llamó a las letras coreanas 'Hangul'. Él se dedicó a educar y estudiar el idioma y las letras coreanas con gran afecto por ellos. Empezó a utilizar términos como 'Han Nara Gul (한나라 글)', Han Nara Mal (한나라 말), Han Mal (한말)' y cambió los nombres de 'Bae Dal Mal Gul Mot Um (배달말 글)' o 'Chosuno Kangsup

Won (조선어 강습 원) a 'Hangeul Mo (한글 모)', 'Hangeul Bae Got (한글 배곧)', y también escribió 'Hangeul Pul lee (한글 풀이) ' para la revista infantil Aidle Boy (아이들 보이). El nombre Hangeul se generalizó desde entonces.

Cada año, el día 9 de octubre los coreanos celebran la creación y divulgación de su sistema de escritura. Éste les permitió por primera vez sentirse libres de la influencia de potencias extranjeras y les dio la herramienta necesaria para que todo el pueblo pudiera al fin comunicarse por escrito en su propio idioma.



Figura 9.- Diferentes nombres del alfabeto coreano



## Capítulo 3

### Desarrollo

Para el desarrollo de la aplicación de la enseñanza del idioma coreano “Hangul” se efectuaron las siguientes actividades tomando en cuenta las etapas de la ingeniería de software, como se menciona a continuación:

- Fase de análisis de requerimientos
- Fase de diseño
- Fase de desarrollo e implementación
- Fase de pruebas
- Fase de liberación

En cada uno de los siguientes apartados se detalla cada una de estas etapas y se menciona cómo se llevaron a cabo y porqué.

### Fase de análisis de requerimientos

Para poder llevar a cabo esta fase del desarrollo de la aplicación móvil se realizaron las siguientes actividades:

- Consulta de aplicaciones existentes en la tienda de aplicaciones de Google
- Entrevistas a compañeras del curso de Coreano
- Consulta de libros de aprendizaje del idioma coreano
- Identificación de necesidades

Con respecto a las aplicaciones que podemos encontrar en la tienda de aplicaciones de Google, también llamada Play Store, podemos observar un gran listado como se muestra en la siguiente figura.

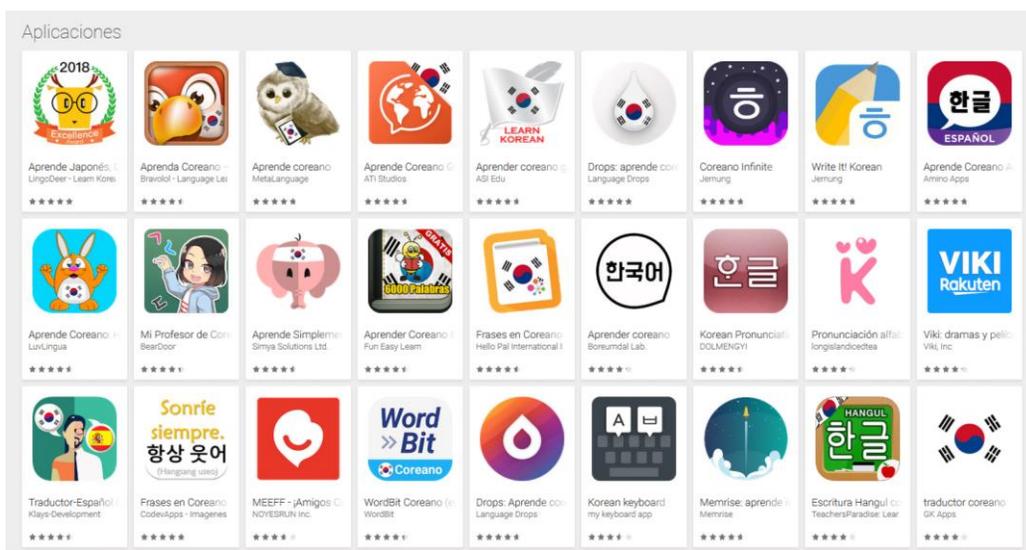


Figura 10.- Diferentes aplicaciones en la Play Store

El inconveniente de estas aplicaciones es que algunas sólo se especializan en traducción, otras en frases comunes, otras en vocabulario, etc., pero ninguna de ellas nos habla acerca de la organización del alfabeto, el orden que se debe seguir para la escritura de los trazos, o mezclan todas estas habilidades en una sola aplicación que nos ayude de manera integral a iniciarnos en el mundo del idioma coreano.

Lo anterior coincide con la siguiente etapa del análisis de requerimientos que es la entrevista, la cual realicé a mis compañeras de la clase de coreano, y las preguntas que hice fueron las siguientes:

- ¿Cuántos años llevas aprendiendo coreano?
- ¿Has probado alguna aplicación móvil para el aprendizaje de dicho idioma? Si tu respuesta fue afirmativa, ¿Cuáles aplicaciones has utilizado?
- ¿Qué mejoras le harías a esas aplicaciones?
- ¿Si descargas una aplicación para aprender coreano, qué te gustaría que tuviera?
- ¿Qué sistema operativo móvil tiene tu smartphone?

Obteniendo como respuestas lo siguiente:

Para la primera pregunta, la mayoría comenta que lleva entre 2 y 2 años y medio con el aprendizaje del idioma, lo cual nos da una idea general de que ya están en nivel intermedio y pueden identificar necesidades básicas para el aprendizaje de dicho idioma.

Para la segunda pregunta nos indican que sí han utilizado aplicaciones para el aprendizaje del coreano y la más utilizada (de hecho por todas) ha sido Duolingo, pero comentan que es una aplicación muy buena, sólo que ya debes tener nociones del idioma para poder utilizarla, si no, es muy complicado entender por qué están organizadas así las lecciones.

Con respecto a las mejoras de la aplicación todas coincidieron en que estaría genial que tuviera una lección, sección o parte que nos introduzca al idioma, para conocer desde el inicio el panorama general del aprendizaje del coreano. Otra de las recomendaciones que estuvo presente en la mayoría de las encuestadas fue que les gustaría practicar los trazos por separado del alfabeto coreano pintando con el dedo en la pantalla como lo hace el traductor de Google para buscar algún significado. Y por último, otra de las cosas que nos comentaron es que la mayoría de las aplicaciones funcionan sólo con acceso a internet y eso no les gusta, por lo que estaría genial que también tuvieran la funcionalidad de modo offline.

Por otra parte, con respecto al sistema operativo móvil que manejan en sus dispositivos móviles el 95% de las encuestadas contestó que manejan Android, mientras que el 5% maneja iOS.

Una vez realizadas las entrevistas y revisadas algunas de las aplicaciones móviles de la PlayStore me basé en los siguientes libros para estructurar la aplicación, así como tomar algunos ejemplos para la práctica del idioma:

- KOREAN LEVEL 1
- Coreano para principiantes
- Idioma coreano nivel inicial

Las referencias completas de estos libros se encuentran en la bibliografía al final del presente documento.

Por último, derivado de las entrevistas a mis compañeras de clase, del análisis de las aplicaciones existentes en la Play Store y de la consulta de bibliografía especializada, decidí hacer (con el visto bueno de mi director de tesis) una aplicación móvil para el sistema operativo Android que contenga las siguientes características:

- ✓ Una pantalla de inicio de la aplicación, en la cual se muestre algo representativo de Corea.
- ✓ Una pantalla principal en la que se muestre qué se puede encontrar en la aplicación.
- ✓ Una pantalla de “Primeros pasos” para que te introduzcas al mundo del aprendizaje del idioma coreano.
- ✓ Una sección que sólo aborde el alfabeto coreano dividido por vocales, consonantes y consonantes dobles.

- ✓ Una sección que tenga el vocabulario más utilizado como por ejemplo, colores, frutas, países, actividades, entre otros.
- ✓ Una sección que tenga las frases más usuales en el idioma coreano.
- ✓ Una sección de verbos en infinitivo y en presente.
- ✓ Para cada sección que se pueda ver la teoría, que te permita escribir con el dedo los trazos principales y/o con el teclado coreano las frases o palabras y una para poder practicar, tipo examen de opción múltiple.
- ✓ La opción de práctica tipo examen de opción múltiple que tenga varios niveles de dificultad (básico, intermedio y avanzado).
- ✓ Que funcione sin conexión a internet.
- ✓ Que sólo funcione en modo vertical o portrait.
- ✓ Que no te haga registrarte en algún sitio o ingresar datos personales.

Una vez obtenidos los requerimientos para el desarrollo de la aplicación móvil para la enseñanza del idioma coreano, actualicé el plan de trabajo para el desarrollo de la tesis, para buscar la forma óptima de cumplir con los requerimientos recabados en tiempo y forma.

Cronograma de actividades																										
Actividades	Octubre					Noviembre				Diciembre				Enero				Febrero			Marzo					
	2	9	16	23	30	6	13	20	27	4	11	18	25	1	8	15	22	29	5	12	19	26	5	12	19	26
<b>INICIO</b>																										
Delimitación	■																									
Hipótesis		■																								
Objetivos		■	■																							
Revisión Bibliográfica			■	■																						
Esquema de Investigación				■																						
<b>DESARROLLO</b>																										
Elaboación del Marco Teórico			■	■	■	■																				
Elección de herramientas de desarrollo						■	■																			
Diseño de la Interfaz							■	■	■	■																
Desarrollo de Contenidos									■	■	■	■														
Programación										■	■	■	■	■	■	■	■	■								
Pruebas del sistema													■	■	■	■	■	■								
Ajustes del sistema																	■	■	■							
Implementación del sistema																		■	■							
<b>CIERRE</b>																										
Organización de Información																			■	■						
Redacción del borrador																				■	■	■				
Revisión y corrección del borrador																						■	■	■		
Entrega Final																								■	■	

Figura 11.- Cronograma de actividades anterior

<b>Cronograma de actividades</b>																								
Actividades	Oct-Nov				Dic-Ene				Mar-Abr				May-Jun				Jul-Ago				Sep-Oct			
	1	15	1	15	1	15	1	15	1	15	1	15	1	15	1	15	1	15	1	15	1	15	1	15
<b>INICIO</b>																								
Delimitación	■	■																						
Hipótesis		■	■																					
Objetivos			■	■																				
Revisión Bibliográfica			■	■																				
Esquema de Investigación				■	■	■																		
<b>DESARROLLO</b>																								
Elaboación del Marco Teórico						■	■	■																
Elección de herramientas de desarrollo								■	■	■														
Diseño de la Interfaz								■	■	■	■													
Desarrollo de Contenidos										■	■	■	■											
Programación											■	■	■	■	■									
Pruebas del sistema												■	■	■	■									
Ajustes del sistema															■	■	■	■	■					
Implementación del sistema															■	■	■	■	■	■				
<b>CIERRE</b>																								
Organización de Información																■	■	■	■					
Redacción del borrador																	■	■	■	■				
Revisión y corrección del borrador																		■	■	■	■			
Entrega Final																				■	■			

Figura 12.- Cronograma de actividades actualizado

## **Fase de diseño y construcción**

### **Aplicación del ADOO**

Aplicando el Análisis y Diseño Orientado a Objetos logré los cuatro objetivos del modelado de la siguiente manera.

Para el primer objetivo, que es visualizar cómo es o debería ser un sistema, me ayudé de las respuestas de las entrevistas a mis compañeras de clase, y al análisis de aplicaciones existentes.

Después, para el segundo y tercer objetivo me basé en el maquetado o generación de prototipos para poder conocer la estructura de la aplicación móvil (en una sección posterior se mostrará a detalle cómo se aplicó para cada una de las pantallas de la aplicación). Asimismo, dentro del segundo objetivo del modelado, para conocer el comportamiento de la aplicación, me apoyé en el uso de los Diagramas de Casos de Uso, los cuales son idóneos cuando se quiere saber qué podemos hacer con una pantalla de una aplicación, y ya conjuntando el maquetado y los diagramas de casos de uso cumplimos con el tercer objetivo que es la generación de plantillas para guiar la construcción de la app.

Y, por último, para cumplir con el cuarto objetivo del modelado que es documentar las decisiones adoptadas, se describieron cada una de las pantallas, así como su diagrama de casos de uso, para que no quedara duda de su diseño e implementación (En una sección posterior se muestra el detalle de esta actividad).

### **Generación del prototipo de la aplicación**

Para este apartado se desarrolló el maquetado, prototipo o mockup (en inglés) de la aplicación móvil, el cual se define como una maqueta o un boceto navegable de la aplicación móvil que permite representar de forma aproximada el producto final que se desea obtener.

Esto es una parte muy importante al realizar una aplicación móvil, ya que de ello depende el avance de esta, la necesidad de elementos dentro de la aplicación, los componentes necesarios, entre otros aspectos, como la conexión a alguna red inalámbrica (por ejemplo, internet), la generación del flujo de la aplicación, los colores a utilizar, el nombre de los botones, las dimensiones necesarias, entre otros.

Además, nos sirve como guía para las pruebas de la aplicación una vez programada, es decir, se puede verificar que todos los elementos coincidan como se propusieron, el flujo de avance sea el correcto y en caso de error, se identifique fácilmente la causa, o en su defecto, la pantalla en la que se origina tal evento.

A continuación, se explica el funcionamiento y propósito de cada diseño de pantalla como se mencionó en la sección anterior, esto con el fin de que el desarrollo e implementación de la aplicación móvil sea una tarea más sencilla y se conozcan desde el inicio los elementos que va a tener cada parte de la aplicación:



Pantalla inicial también conocida como screen splash, es una imagen que se muestra brevemente al iniciar la app.

***Pantalla 1.- Pantalla Inicial***



*Pantalla 2.- Menú Principal*

La aplicación se divide en 4 secciones:

- Alfabeto Coreano
- Vocabulario
- Verbos
- Expresiones



*Pantalla 3.- Sección Alfabeto Coreano*

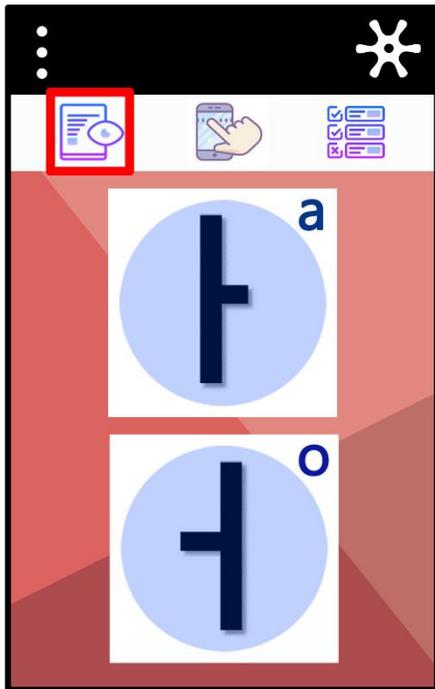
La sección de alfabeto coreano tiene un submenú el cual se divide en 4 secciones:

- Primeros pasos: Se da a conocer los aspectos generales de la escritura coreana.
- Vocales
- Consonantes
- Consonantes Dobles

Las últimas tres secciones cuentan con módulos que permiten practicar el tópico seleccionado.

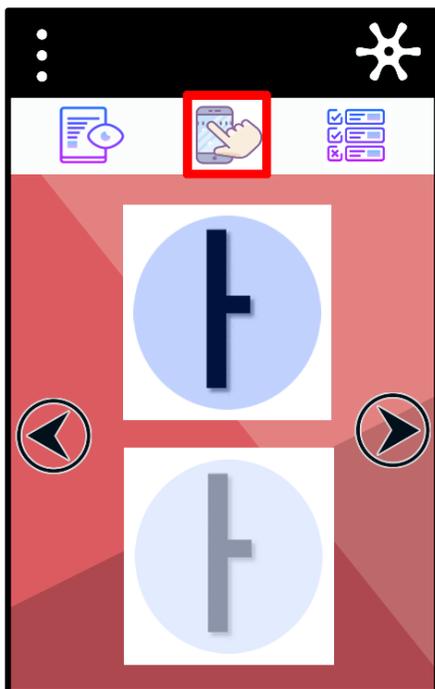
Los módulos son:

1. Vista general
2. Escritura
3. Test



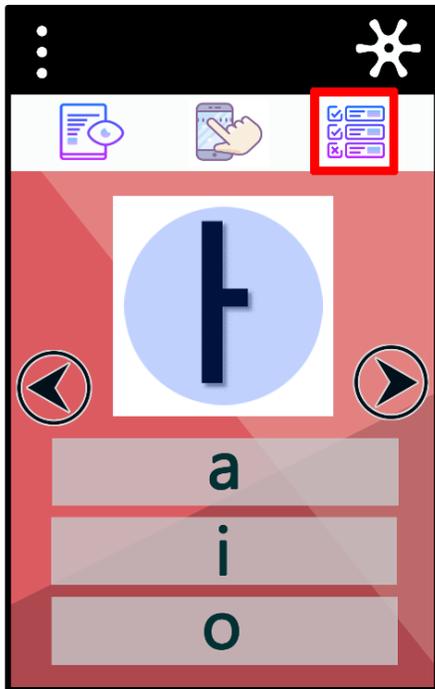
**Pantalla 3.1.- Vista General, sección vocales**

Ejemplo módulo vista general, en sección vocales.



**Pantalla 3.2.- Escritura, sección vocales**

Ejemplo módulo escritura, en sección vocales.



*Pantalla 3.3.- Test, sección vocales*

Ejemplo módulo test, en sección vocales.



*Pantalla 4.- Sección Vocabulario*

La sección vocabulario contiene categorías que permiten al usuario escoger la categoría de su preferencia.

Cada categoría cuenta con módulos que permiten practicar la categoría seleccionada.

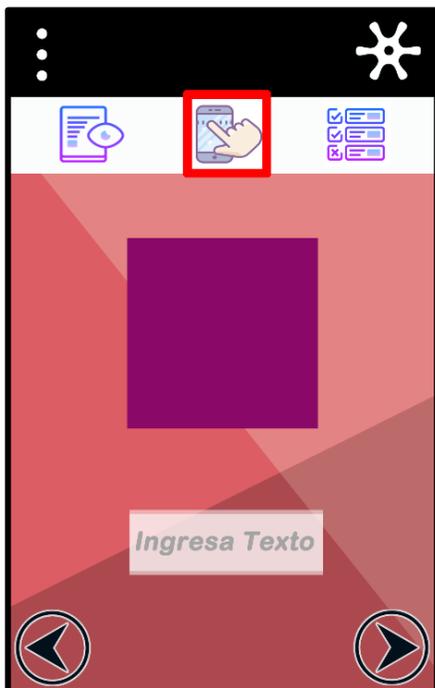
Los módulos son:

1. Vista general
2. Escritura
3. Test



Ejemplo módulo vista general, en categoría colores.

**Pantalla 4.1.- Vista General, sección vocabulario**



Ejemplo módulo escritura, en categoría colores.

**Pantalla 4.2.- Escritura, sección vocabulario**



*Pantalla 4.2.- Test, sección vocabulario*

Ejemplo módulo test, en categoría colores.



*Pantalla 5.- Sección Verbos*

La sección verbos permite al usuario escoger entre el tiempo verbal infinitivo y presente.

Cada tiempo verbal cuenta con módulos que permiten practicar el tiempo verbal seleccionado.

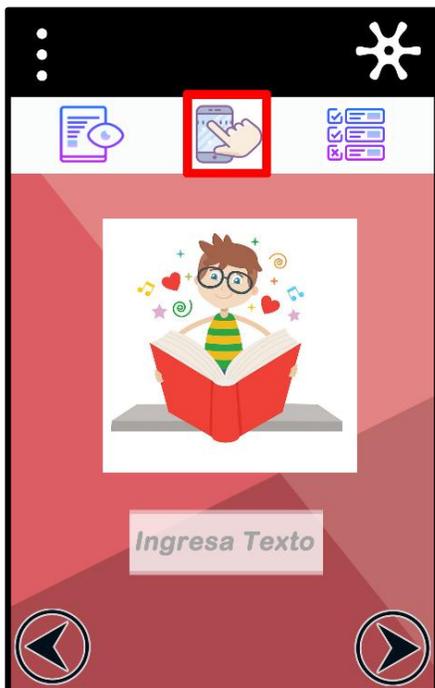
Los módulos son:

1. Vista general
2. Escritura
3. Test



*Pantalla 5.1.-Vista General, sección verbos*

Ejemplo módulo vista general, en tiempo verbal infinitivo.



*Pantalla 5.1.-Escritura, sección verbos*

Ejemplo módulo escritura, en tiempo verbal infinitivo.



**Pantalla 5.3.-Test, sección verbos**

Ejemplo módulo test, en tiempo verbal infinitivo.



**Pantalla 6.- Sección Expresiones**

La sección expresiones contiene frases básicas en el idioma coreano, las cuales al pulsar sobre ellas nos muestran su significado.



*Pantalla 6.1.- Sección Expresiones, significado*

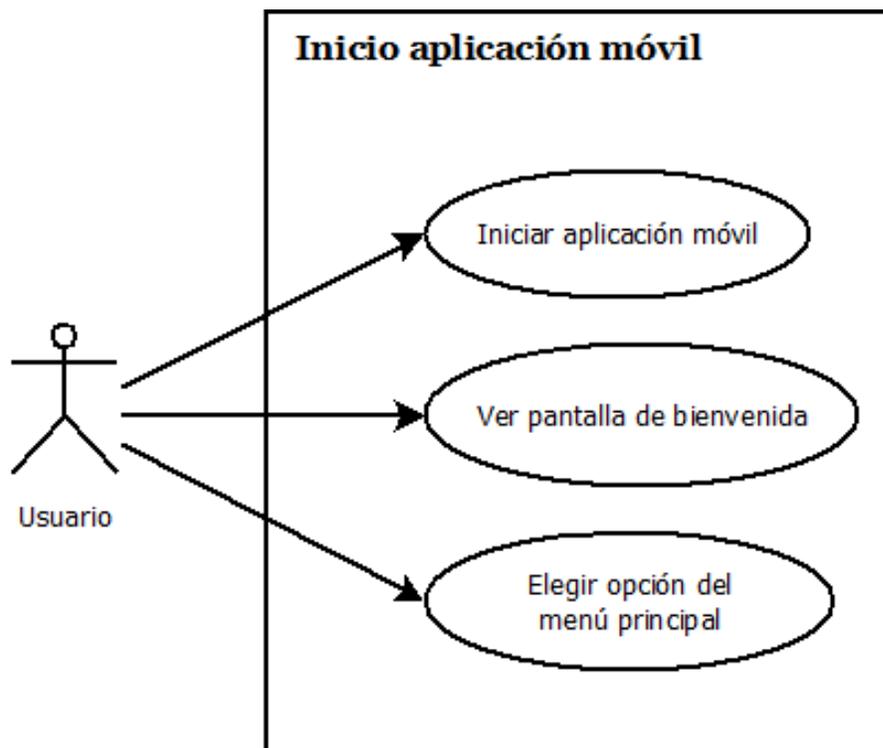
Ejemplo del significado de la expresión  
안녕하세요.

## Diagramas de casos de uso

Una vez diseñadas cada una de las pantallas de la aplicación prototipo, continúe con la creación de los diagramas UML, específicamente los diagramas de casos de uso, ya que en las aplicaciones móviles este tipo de diagramas es de los que más se utilizan para saber las acciones que se pueden realizar en cada pantalla de la aplicación.

A continuación, se muestran cada uno de los diagramas de casos de uso de la aplicación “Hangul”:

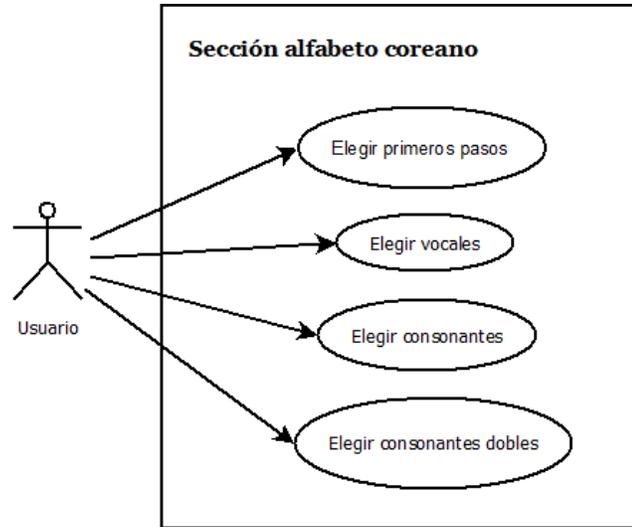
- Iniciar aplicación móvil



*Caso de uso 1.- Inicio aplicación móvil*

El usuario inicia la app, después se muestra la pantalla de bienvenida. Y por último se muestra el menú principal de la app.

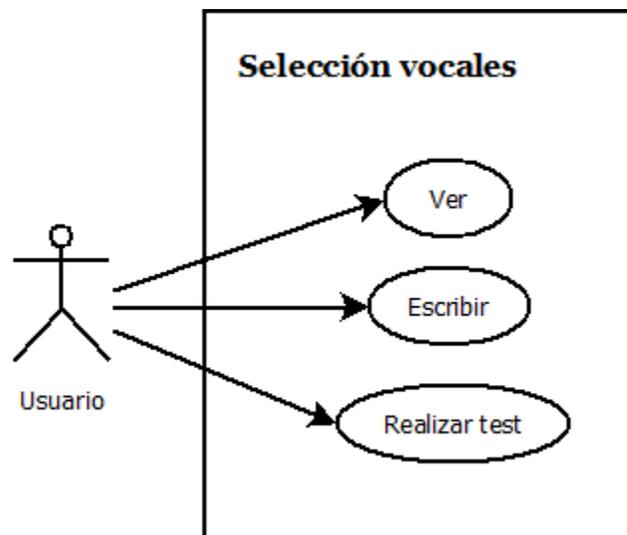
- Seleccionar sección alfabeto coreano



*Caso de uso 2.- Sección alfabeto coreano*

El usuario debe escoger alguna opción de la sección del alfabeto coreano.

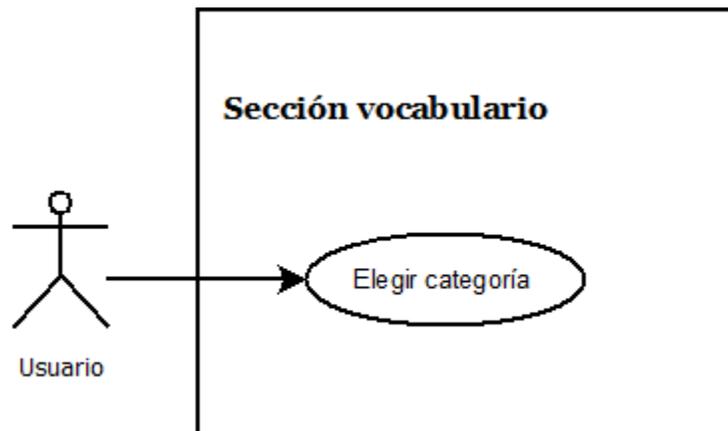
- Elegir vocales



*Caso de uso 3.- Selección vocales*

El usuario al elegir la sección de vocales puede escoger alguno de los tres módulos (vista general, escribir y test). Por defecto inicia en el módulo vista general.

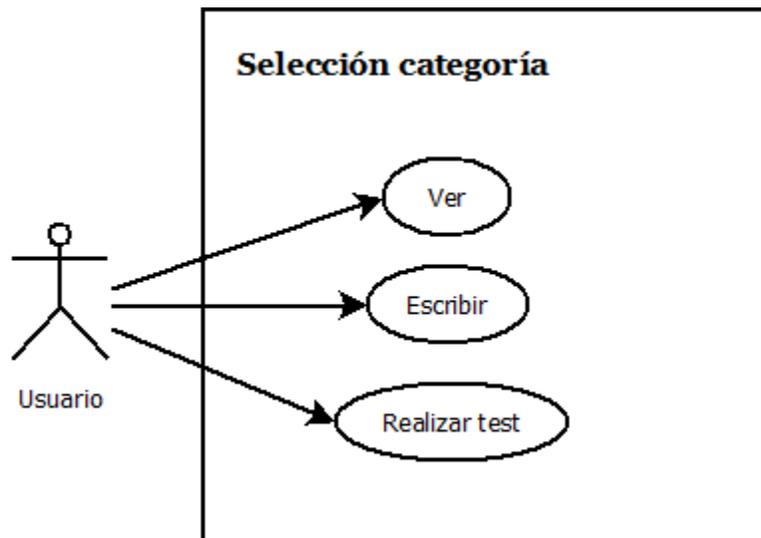
- Seleccionar categoría de la sección vocabulario



*Caso de uso 4.- Sección vocabulario*

El usuario debe escoger alguna categoría de la sección vocabulario.

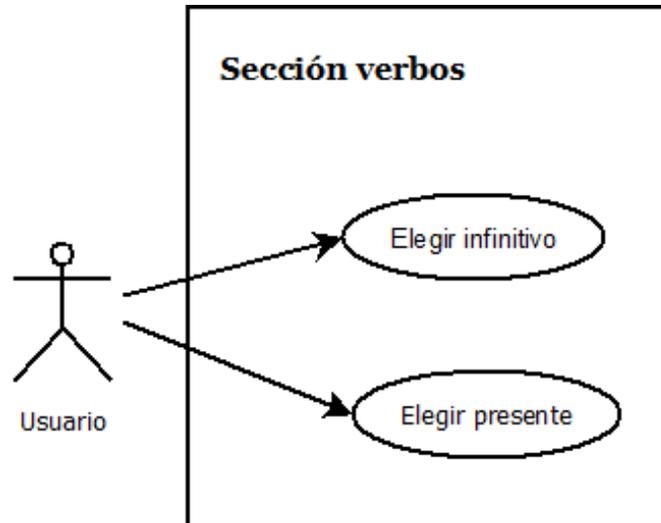
- Elegir categoría



*Caso de uso 5.- Selección categoría*

El usuario al elegir la categoría deseada puede escoger alguno de los tres módulos (vista general, escribir y test). Por defecto inicia en el módulo vista general.

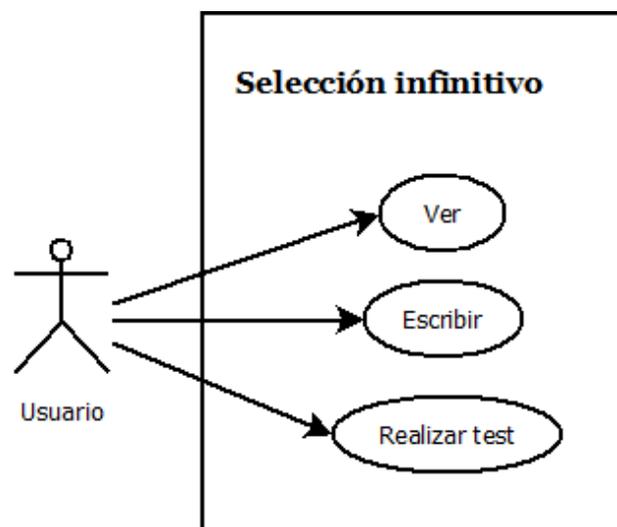
- Seleccionar sección verbos



*Caso de uso 6.- Sección verbos*

El usuario debe elegir una opción de la sección verbos

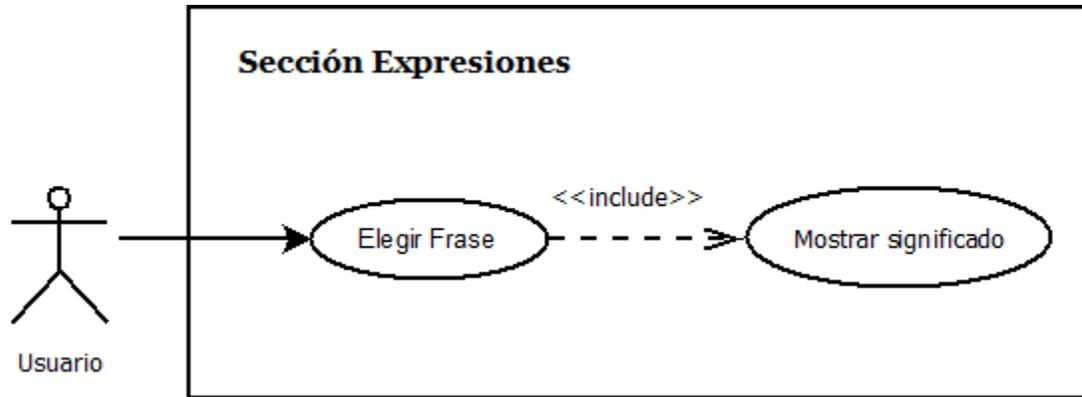
- Seleccionar Infinitivo



*Caso de uso 7.- Selección infinitivo*

El usuario al elegir infinitivo puede escoger alguno de los tres módulos (vista general, escribir y test). Por defecto inicia en el módulo vista general.

- Seleccionar sección expresiones



*Caso de uso 8.- Selección opciones*

El usuario al pulsar sobre la frase se le mostrará el significado.

## Aplicación de la POO a una aplicación móvil

Antes de comenzar con la parte de programación, realicé una de las actividades que aprendí en la materia de cómputo móvil cuando se desarrollan aplicaciones móviles y es la aplicación de la Programación Orientada a Objetos, esto es, ver cada pantalla como si fuera un objeto y aplicarle los cuatro pilares de la Programación Orientada a Objetos como se muestra a continuación:



Figura 13.- Sección alfabeto coreano

La pantalla “Sección alfabeto coreano” se utilizó como ejemplo, pero se describe de manera general en esta sección la aplicación de los cuatro pilares de la programación orientada a objetos para la aplicación en general.

Para empezar, al obtener la abstracción de una pantalla encontramos las características y funcionalidades fundamentales de esta, en este caso para la pantalla “alfabeto coreano” tenemos:

- Abstracción de datos:
  - Nombre: pantalla sección alfabeto coreano
  - Fondo: azul
- Abstracción Funcional
  - Ver primeros pasos
  - Practicar vocales
  - Practicar consonantes

- Practicar consonantes dobles

Ahora para la parte de encapsulamiento, dividimos la pantalla en 3 elementos

- Interfaz pública
- Implementación
- Información interna

La interfaz pública es lo que el usuario va a ver y tendrá acceso, en este caso el diseño del prototipo nos ayuda con esto.

Para la parte de implementación se escribió el código necesario en XML y java para el funcionamiento de cada una de las pantallas (en el anexo se encuentra el código de las pantallas principales de la aplicación).

Y por último como parte de la información interna encontramos las cadenas de texto que van a ir dentro de nuestra aplicación, los colores utilizados, así como las dimensiones de la misma.

Pasando a la herencia, esto se puede ver como un conjunto de vistas, que tienen un comportamiento similar, que se reutilizaron en algunas partes de la aplicación, como es el caso de los botones, las vistas recicladas o RecyclerView, entre otros.

Así también una de las aplicaciones más importantes de aplicar el concepto de herencia a una aplicación móvil es la reutilización de código, ya que esta característica hace que no reescribamos cada una de las librerías necesarias para hacer funcionar la aplicación, sino que solamente heredemos de la clase que la contiene y comencemos a utilizarla.

Y por último se aplicó polimorfismo, la capacidad de usar un mismo objeto de diferentes formas, y esto se ve reflejado al reutilizar el evento click de la función OnClick para todos los botones (“Primeros pasos”, “vocales”, “consonantes” y “consonantes dobles”) para abrir otra pantalla, pero cada uno de los botones nos abre pantallas diferentes como se ve en el diagrama de casos de uso de la figura 14.

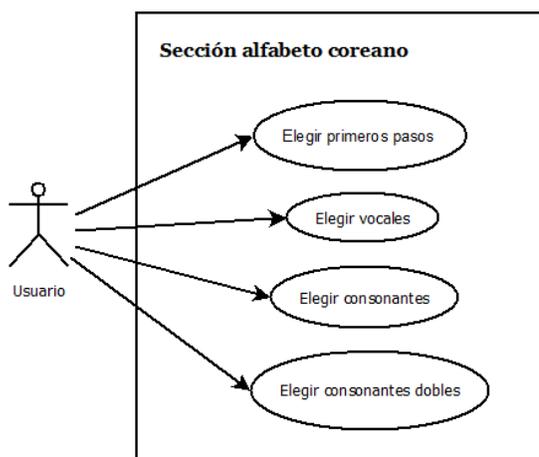


Figura 14.- Ejemplo de caso de uso

## Fase de implementación

Una vez teniendo los requerimientos de la aplicación, el desarrollo del prototipo, los diagramas de casos de uso y la aplicación del análisis y diseño orientado a objetos, así como la aplicación de dicho paradigma, podemos pasar a la parte de programación de la aplicación móvil.

Para este desarrollo utilicé Android Studio, un IDE (Integrated Development Environment o Entorno de Desarrollo Integrado), el cual nos ayudó a codificar, visualizar, editar, actualizar, simular y generar nuestra aplicación final.

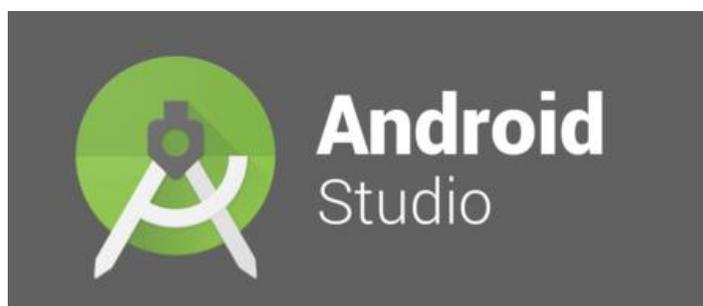


Figura 15.- Logotipo Android Studio

La estructura del proyecto quedó como sigue:

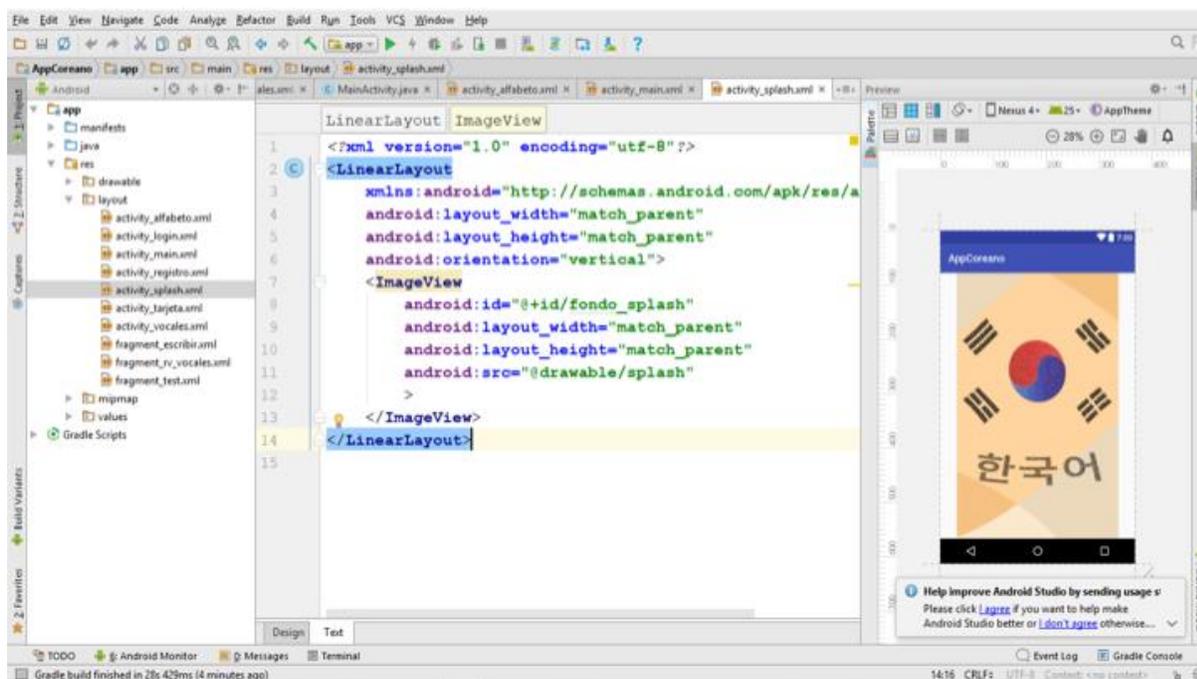
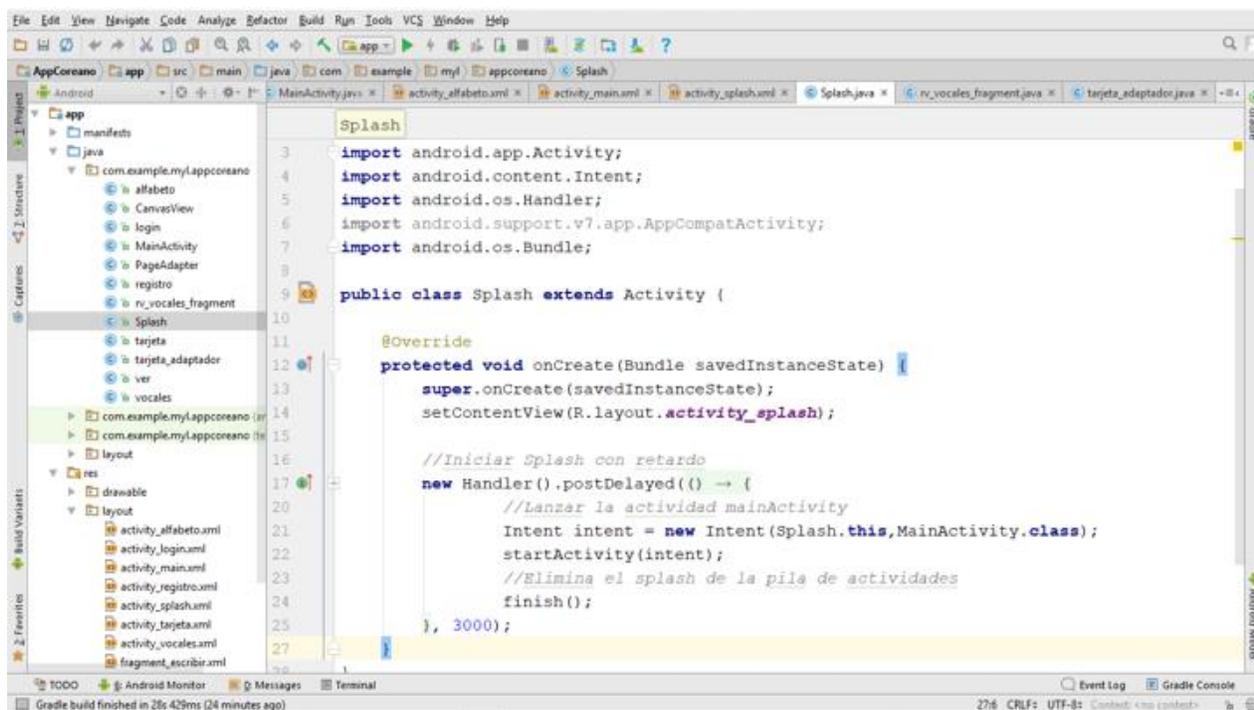


Figura 16.- Ejemplo código XML

Como podemos observar en la figura 16, la organización del código de la aplicación quedó de la siguiente manera:

- App: Contiene todas las subcarpetas de nuestro proyecto
- Manifest: Contiene la configuración y los permisos necesarios para nuestra aplicación.
- Java. Contiene todos los códigos que hacen la funcionalidad lógica de la aplicación.
- Res. Contiene todos los diseños de las pantallas, así como los recursos de la aplicación como son:
  - Cadenas de texto
  - Imágenes
  - Colores
  - Dimensiones
- Gradle: Contiene los Scripts necesarios para la sincronización de la aplicación con todas las librerías necesarias para su funcionamiento.

El código fuente de cada pantalla de la aplicación (Java y XML) se encuentra en el Anexo correspondiente. A continuación, se muestra un ejemplo del código que hace funcionar el screen-splash de forma correcta.



```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
AppCoreano app src main java com example myl appcoreano Splash
Android MainActivity.java activity_alfabeto.xml activity_main.xml activity_splash.xml Splash.java rv_vocales_fragment.java tarjeta_adaptador.java
Project
  app
    manifests
    java
      com.example.mylappcoreano
        alfabeto
        CanvasView
        login
        MainActivity
        PageAdapter
        registro
        rv_vocales_fragment
        Splash
        tarjeta
        tarjeta_adaptador
        ver
        vocales
      com.example.mylappcoreano (ar
      com.example.mylappcoreano (it
    layout
    res
      drawable
      layout
        activity_alfabeto.xml
        activity_login.xml
        activity_main.xml
        activity_registro.xml
        activity_splash.xml
        activity_tarjeta.xml
        activity_vocales.xml
        fragment_escribir.xml
Build Variants
  2. Features
  Android Monitor
  Messages
  Terminal
  Event Log
  Gradle Console
  276 CRLF: UTF-8: Coded: 450 (166%)
Gradle build finished in 28s 429ms (24 minutes ago)
```

```
3 import android.app.Activity;
4 import android.content.Intent;
5 import android.os.Handler;
6 import android.support.v7.app.AppCompatActivity;
7 import android.os.Bundle;
8
9 public class Splash extends Activity {
10
11     @Override
12     protected void onCreate(Bundle savedInstanceState) {
13         super.onCreate(savedInstanceState);
14         setContentView(R.layout.activity_splash);
15
16         //Iniciar Splash con retardo
17         new Handler().postDelayed(() -> {
18             //Lanzar la actividad mainActivity
19             Intent intent = new Intent(Splash.this, MainActivity.class);
20             startActivity(intent);
21             //Elimina el splash de la pila de actividades
22             finish();
23         }, 3000);
24     }
25 }
26
27
```

Figura 17.- Ejemplo código Java

## Fase de pruebas

De las buenas prácticas de ingeniería de software tenemos dos pruebas principales al desarrollar aplicaciones móviles, las pruebas unitarias y las pruebas de integración.

## Pruebas unitarias

Para esta parte opté por hacer pruebas pantalla a pantalla, es decir, cada que terminaba de diseñar y codificar una pantalla la probé como pantalla única. Lo anterior quiere decir que esa pantalla era la principal del proyecto, esto es, al iniciar la aplicación, es la primera que nos lanza y a través del Log de Android Studio pude observar que todo funcionaba conforme a los requerimientos. En la figura 18 se muestra un ejemplo del Log que Android Studio nos provee.

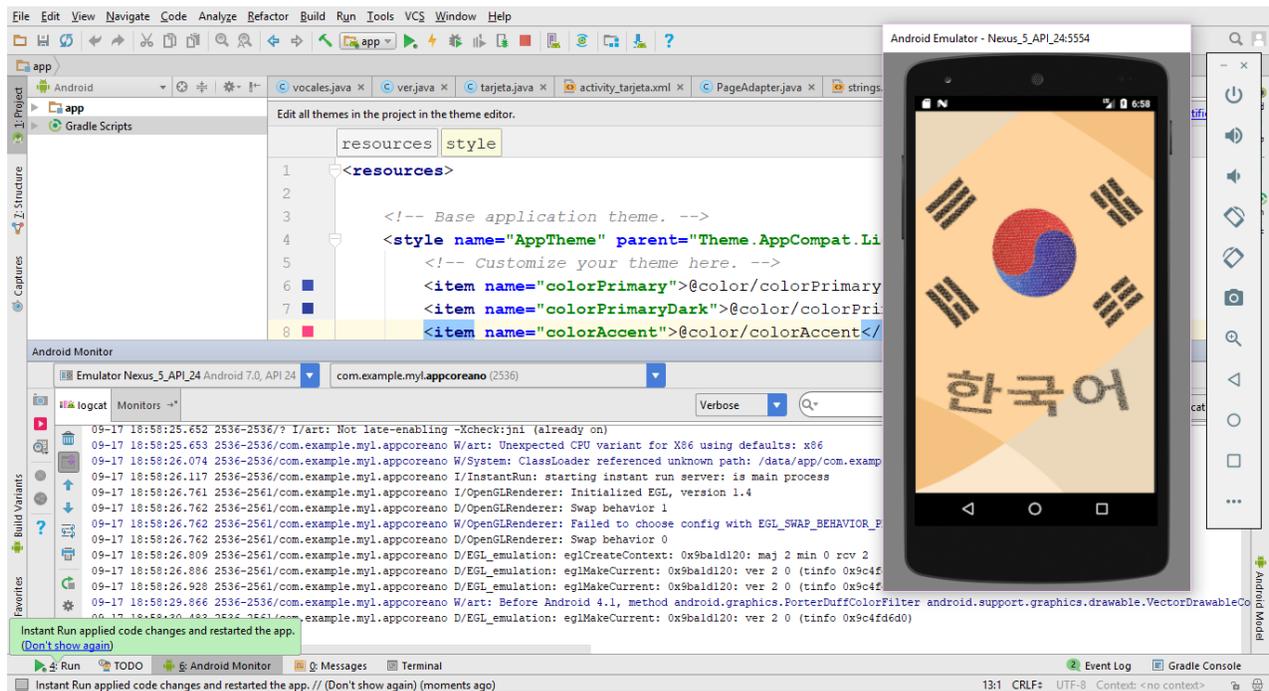


Figura 18.- Logcat de Android Studio

## Pruebas de integración

Una vez verificado el funcionamiento de las pantallas a través de las pruebas unitarias, construí el árbol de depuración que se muestra a continuación para la realización de las pruebas de integración.

Esto es:

- Seguir el flujo de pantallas de la aplicación.
- Ejecutar correctamente la pantalla de Splash (es decir, que una vez terminada, se borre de la pila de actividades como se muestra en la imagen del ciclo de vida de un activity).
- Verificar la bitácora o Log del sistema para corroborar los eventos lanzados al trasladarse de una pantalla a otra.

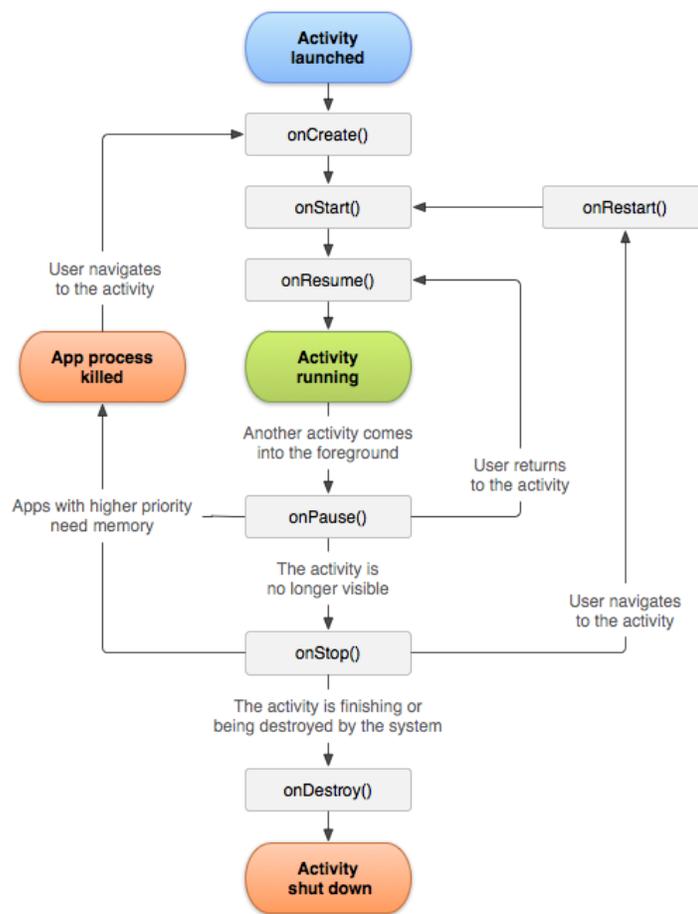


Figura 19.- Ciclo de vida de un activity tomado de Android Developers

A continuación, se muestra el árbol de depuración de la aplicación móvil para el aprendizaje del idioma coreano “Hangul”:

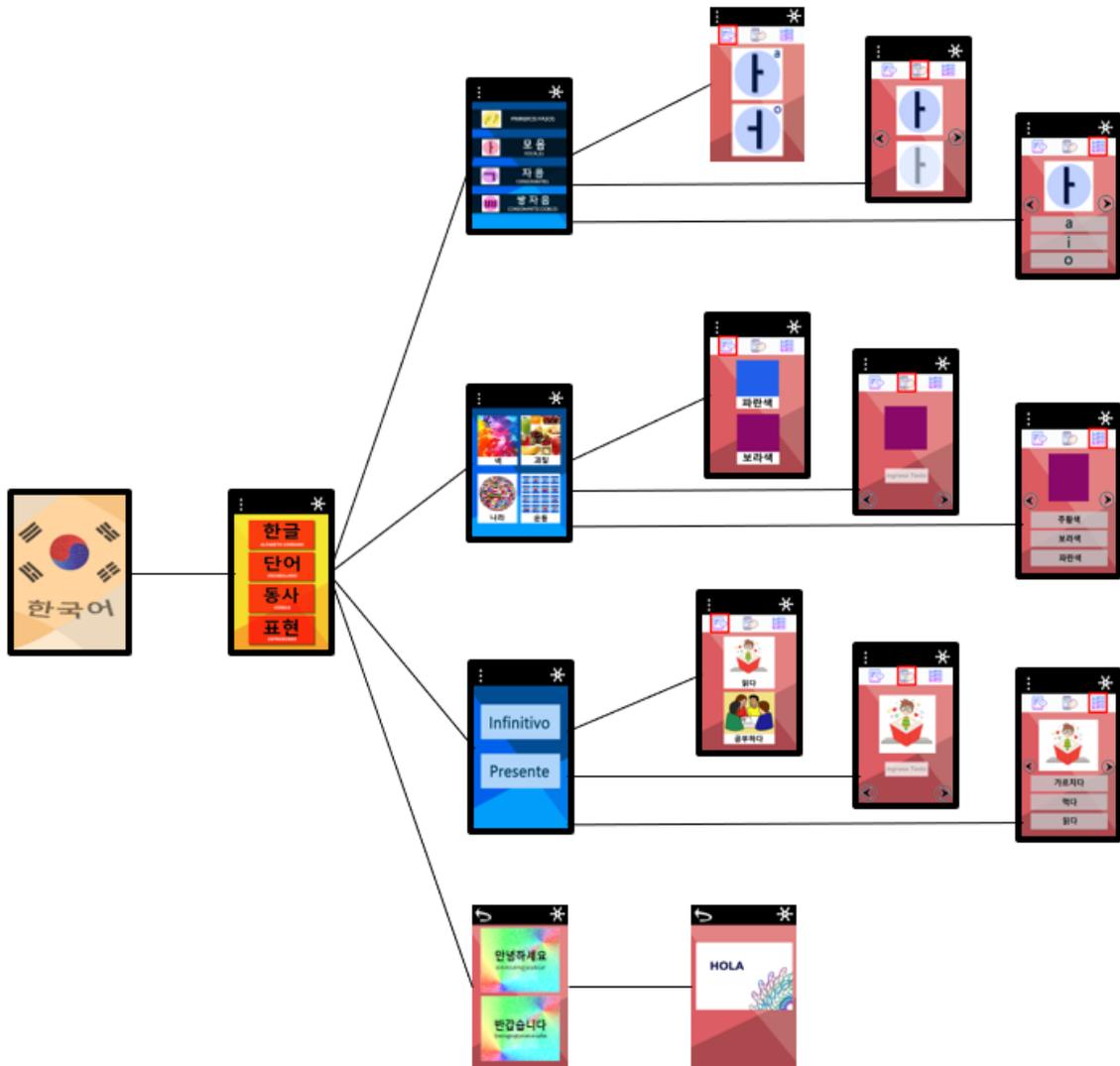


Figura 20.- Árbol de depuración de la aplicación

## Fase de liberación

Para esta última etapa del proyecto, se generó el apk a través de Android Studio correspondiente a la aplicación y se instaló en los smartphones de mi grupo de la clase de coreano, para que la probaran e hicieran comentarios al respecto.

Así también se instaló en los siguientes dispositivos para probar que se ejecutara bien en diferentes sistemas operativos, configuraciones y tipos de acceso a la red.

Los dispositivos en los que se probó fueron los siguientes:

- HTC Desire 510
- Samsung Galaxy Fame
- Samsung Galaxy Young
- Samsung mini
- Motorola C plus
- Emulador Nexus
- Samsung Galaxy Tab A6 with Spen

Y en todos cumplió tanto con las pruebas unitarias como las de integración.



Figura 21.- Prueba de la aplicación en el Emulador



Figura 22.- Prueba de la aplicación en HTC Desire 526G



Figura23.- Prueba de la aplicación en Samsung Galaxy A6 with S pen



## Capítulo 4

### Resultados

Como resultado principal de la tesis se tiene el desarrollo de la aplicación "Hangul" para la enseñanza del idioma coreano, la cual basa su filosofía en el aprendizaje en movimiento o mejor conocido como M-Learning. Su principal aporte es que la aplicación desarrollada cubre con todos y cada uno de los requerimientos obtenidos en el análisis y obtención de requerimientos mostrados al inicio del presente documento en la sección de objetivos.

Como primer objetivo se tenía que recopilar información acerca de la enseñanza del idioma coreano, este objetivo se logró con base en mis clases de Coreano en ese entonces CELE (Centro de Enseñanza de Lenguas Extranjeras), actualmente ENALLT (Escuela Nacional de Lenguas, Lingüística y Traducción), las cuales fueron clave para el desarrollo de dicha aplicación, ya que sin ellas, a pesar de que me gusta muchísimo el idioma coreano, siempre se necesita una guía para poder lograr el mayor aprendizaje.

Otro de los objetivos logrados gracias al desarrollo de esta tesis, fue encontrar los principales tópicos a cubrir por la aplicación móvil. Lo anterior lo cumplí a través de diversas actividades, principalmente la entrevista a mis compañeras de clase en base a su experiencia con otras aplicaciones para dicho propósito, así también, a través de mi propia experiencia con el uso de otras aplicaciones.

Lo anterior implicó un gran reto, ya que es la primera vez que desarrollo una aplicación móvil de este alcance, debido a que la mayoría de las aplicaciones que había realizado no tenían tantos requerimientos técnicos como los que necesitó esta aplicación. Uno de ellos fue el uso del RecyclerView, un elemento de android capaz de optimizar la memoria del dispositivo a través del reciclaje de vistas plantilla, en las cuales se van colocando sólo los elementos que se muestran en pantalla y no todos los elementos de una vista general, es decir, si en una pantalla estamos viendo la imagen del alfabeto coreano, se muestran y cargan en memoria sólo las que caben en la pantalla, sin embargo, si hubiera utilizado un ListView, se hubieran tenido que cargar todos los trazos del alfabeto coreano y eso haría menos eficaz la aplicación. Otro de los elementos y de los que estoy orgullosa de haber podido implementar, fue el canvas, ya que este me permitió realizar uno de los principales requerimientos de la aplicación móvil, que es la práctica de los trazos en la pantalla del dispositivo utilizando ya sea una pluma stylus o bien el dedo.

Por último me gustaría recalcar que uno de los resultados más importantes en mi desarrollo personal y profesional ha sido darme cuenta de que puedo ser capaz de desarrollar aplicaciones móviles desde su inicio (análisis y obtención de requerimientos)

hasta su conclusión (etapa de liberación del producto final) y sobre todo puedo ir perfeccionando las características de la aplicación con nuevos elementos que vayan saliendo para hacerla óptima y no dejarla como muchas aplicaciones que sólo se quedan en la versión 1.0 y no son actualizadas para extender su vida útil.

## Conclusiones

Gracias al desarrollo de esta aplicación me permitió llevar a la práctica todas y cada una de las etapas que conlleva la ingeniería de software, comenzando desde el análisis y obtención de requerimientos hasta la liberación del producto final, en este caso la generación del archivo apk para la instalación de este en diferentes dispositivos móviles como tablets y/o smartphones.

En la fase de análisis de requerimientos pude ver el reto que conlleva entender lo que el cliente quiere (en este caso yo incluida) y poder plasmarlo como requerimientos para el diseño y desarrollo de la aplicación.

En la etapa de diseño me fueron muy útiles los conocimientos de la paquetería de Adobe Creative Suite, en específico Adobe Photoshop, el cual me permitió generar los mockups o prototipos de la aplicación y armar el árbol de flujo de pantallas con lo cual pude descubrir los elementos que reutilicé a lo largo de la aplicación como el RecyclerView, las interfaces, entre otros.

También en esta etapa comprendí el trabajo que el diseño de una aplicación móvil involucra, ya que en la sección de práctica del Hangul diseñé la forma y orden de construir cada trazo que representa el alfabeto coreano.

Otro reto que me gustó fue la búsqueda y/o edición de imágenes representativas para cada palabra del vocabulario.

Hoy en día estamos viviendo la quinta generación del cómputo móvil la cual involucra inteligencia artificial, IoT (internet de las cosas), Big Data y M-Learning y dado que la mayoría de las personas ya cuentan con un smartphone a su alcance, quiero continuar con la siguiente versión de esta aplicación, la cual involucre también fonética, comprensión de lectura y más vocabulario existente.

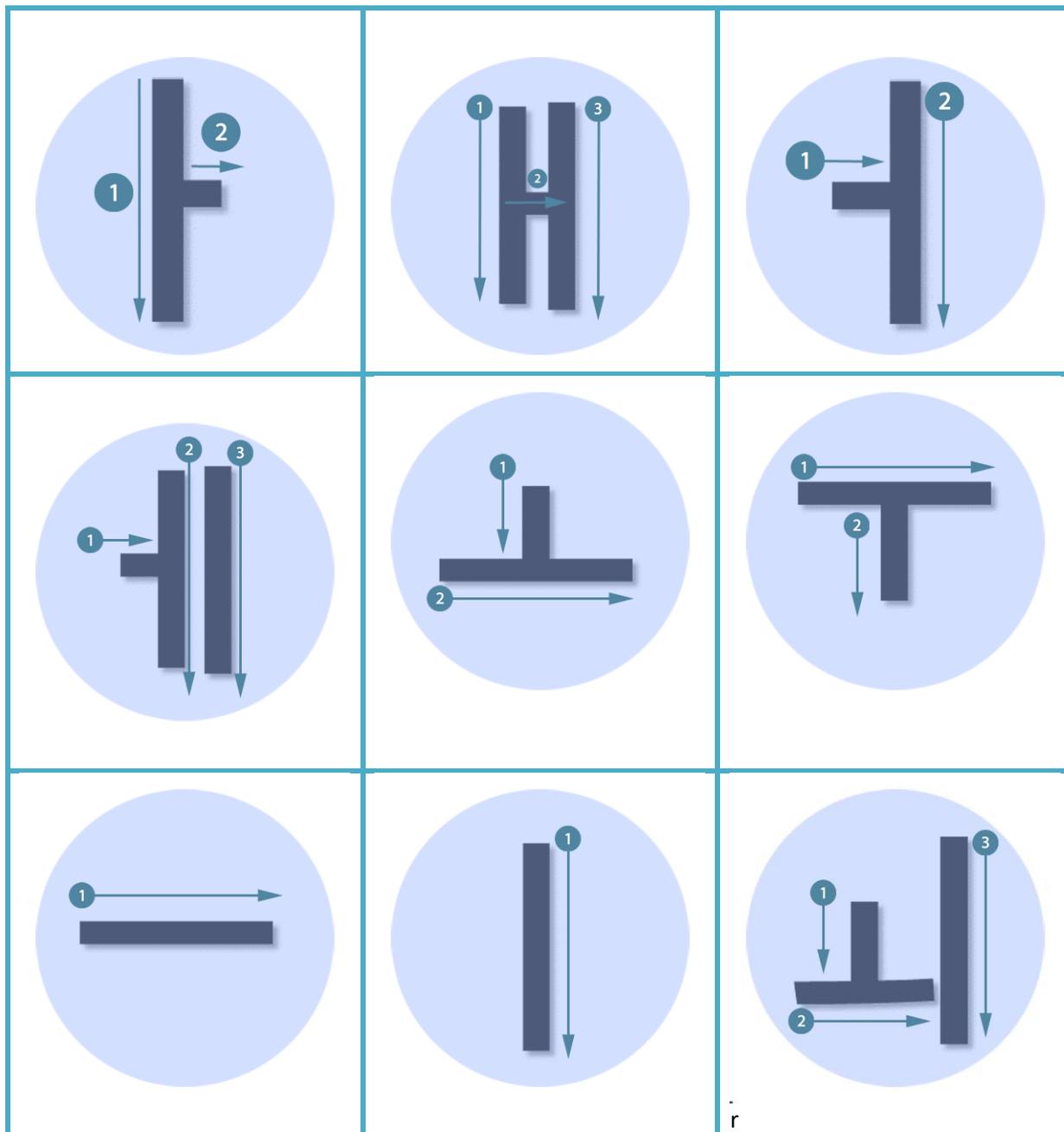
Así también quiero desarrollar esta misma aplicación para dispositivos con sistema operativo iOS y Windows 10, ya que platicando acerca de la aplicación me encontré con personas que no cuentan con un smartphone con sistema operativo Android.

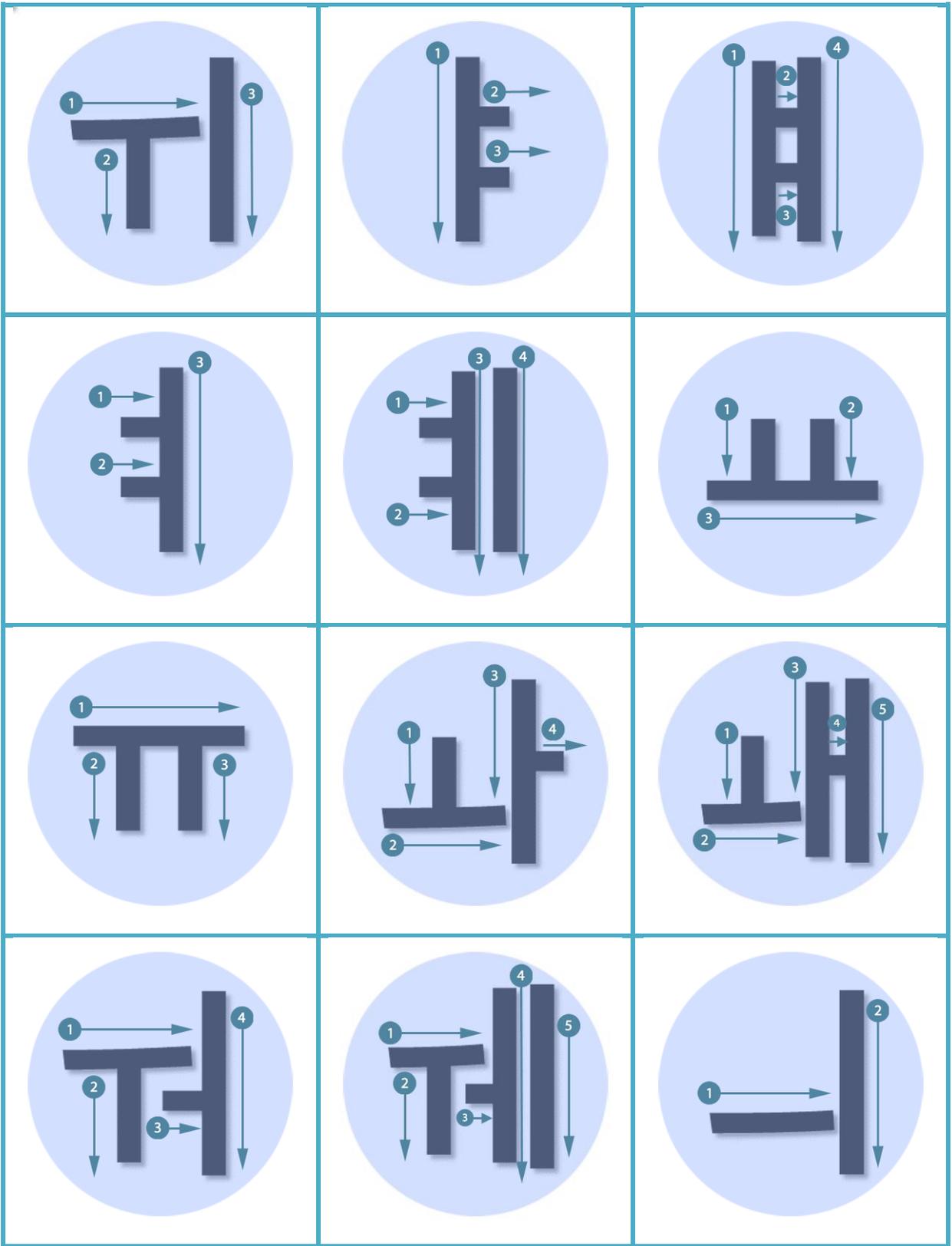


## Anexos

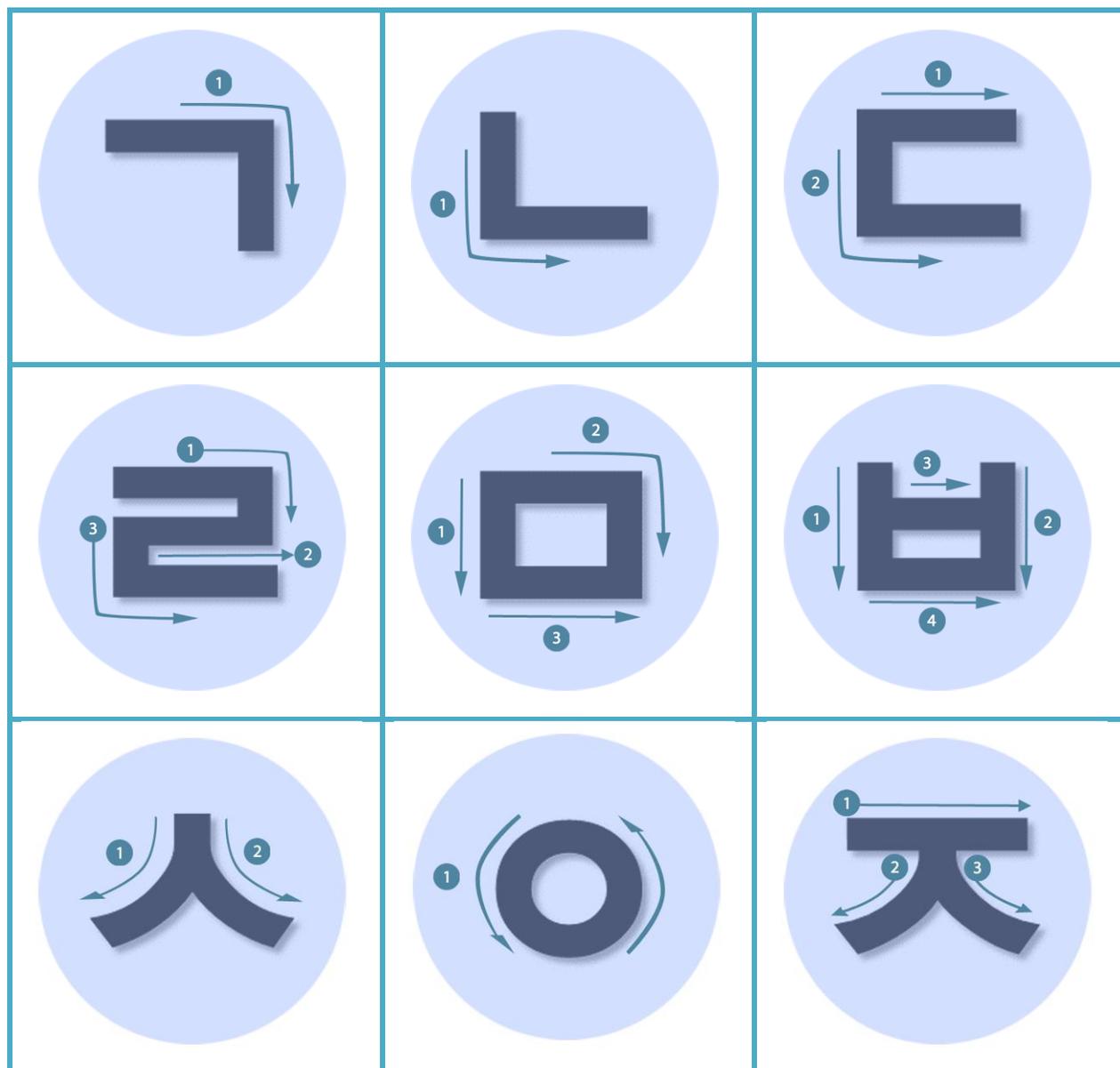
### Diseño de construcción de los trazos del alfabeto coreano

Vocales





# Consonantes





## Código Fuente



Splash

## Código XML

```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <ImageView
        android:id="@+id/fondo_splash"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:src="@drawable/splash"
    >
    </ImageView>
</LinearLayout>
```

## Código Java

```
package com.example.myl.appcoreano;

import android.app.Activity;
import android.content.Intent;
import android.os.Handler;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class Splash extends Activity {

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

        //Iniciar Splash con retardo
        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                //Lanzar la actividad MainActivity
                Intent intent = new
Intent(Splash.this, MainActivity.class);
                startActivity(intent);
                //Elimina el splash de la pila de actividades
                finish();
            }
        }, 3000);
    }
}
```



## Código XML

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/base_amarillo"
    android:orientation="vertical">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@color/negro">
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/titulo"
            android:textColor="@color/blanco"
            android:textSize="20dp"
            android:gravity="center_horizontal"
            android:textStyle="bold"
            android:fontFamily="sans-serif"
            android:layout_marginTop="10dp"
            android:layout_marginBottom="10dp"/>
    </LinearLayout>
</LinearLayout>

```

```

android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
android:gravity="center_vertical">
<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="@color/naranja"
    android:src="@drawable/alfabeto_coreano"
    android:layout_gravity="center_horizontal"
    android:layout_marginTop="20dp"
    android:layout_marginBottom="10dp"
    android:paddingRight="40dp"
    android:paddingLeft="30dp"
    android:onClick="alfabeto"
/>
<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="@color/naranja"
    android:src="@drawable/vocabulario"
    android:layout_gravity="center_horizontal"
    android:layout_marginTop="10dp"
    android:layout_marginBottom="10dp"
    android:paddingRight="30dp"
    android:paddingLeft="30dp"
    android:onClick="vocabulario"
/>
<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="@color/naranja"
    android:src="@drawable/verbos"
    android:layout_gravity="center_horizontal"
    android:layout_marginTop="10dp"
    android:layout_marginBottom="10dp"
    android:paddingRight="30dp"
    android:paddingLeft="30dp"
    android:onClick="verbos"
/>
<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="@color/naranja"
    android:src="@drawable/expresion"
    android:layout_gravity="center_horizontal"

```

```

        android:layout_marginTop="10dp"
        android:layout_marginBottom="10dp"
        android:paddingRight="30dp"
        android:paddingLeft="30dp"
        android:onClick="expresiones"
    />

</LinearLayout>
</LinearLayout>

```

## Código Java

```

package com.example.myl.appcoreano;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;

public class MainActivity extends AppCompatActivity {

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

    //Ir a alfabeto coreano
    public void alfabeto(View v)
    {
        Intent intent=new
Intent(MainActivity.this,alfabeto.class);
        startActivity(intent);
    }

    //Ir a vocabulario
    public void vocabulario(View v)
    {
        Intent intent=new Intent(MainActivity.this,login.class);
        startActivity(intent);
    }

    //Ir a alfabeto verbos
    public void verbos(View v)
    {

```

```

        Intent intent=new
Intent(MainActivity.this,registro.class);
        startActivity(intent);
    }

    //Ir a expresiones
    public void expresiones(View v)
    {
        Intent intent=new Intent(MainActivity.this,login.class);
        startActivity(intent);
    }
}

```



### Código XML

```

<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/base_azul"
    android:orientation="vertical">
    <LinearLayout

```

```

    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@color/negro">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/titulo"
        android:textColor="@color/blanco"
        android:textSize="20dp"
        android:gravity="center_horizontal"
        android:textStyle="bold"
        android:fontFamily="sans-serif"
        android:layout_marginTop="10dp"
        android:layout_marginBottom="10dp"/>
</LinearLayout>
<!--Primeros pasos-->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:background="@color/azul"
    android:layout_marginTop="20dp"
    android:onClick="primeros_pasos">
    <ImageView
        android:layout_width="0dp"
        android:layout_height="75dp"
        android:layout_weight="0.3"
        android:src="@drawable/pasos"
        android:layout_marginTop="5dp"
        android:layout_marginBottom="5dp"/>
    <TextView
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="0.7"
        android:text="@string/primeros_pasos"
        android:textColor="@color/blanco"
        android:textSize="25dp"
        android:gravity="center_horizontal"
        android:layout_gravity="center_vertical"/>
</LinearLayout>
<!--Vocales-->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:background="@color/azul"

```

```

android:layout_marginTop="20dp"
android:onClick="vocales">
<ImageView
    android:layout_width="0dp"
    android:layout_height="75dp"
    android:layout_weight="0.3"
    android:src="@drawable/a"
    android:layout_marginTop="5dp"
    android:layout_marginBottom="5dp"
    android:layout_gravity="center_vertical"/>
<TextView
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="0.7"
    android:text="@string/vocales"
    android:textColor="@color/blanco"
    android:textSize="25dp"
    android:gravity="center_horizontal"
    android:layout_gravity="center_vertical"/>
</LinearLayout>
<!--Consonantes-->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:background="@color/azul"
    android:layout_marginTop="20dp"
    android:onClick="consonantes">
<ImageView
    android:layout_width="0dp"
    android:layout_height="75dp"
    android:layout_weight="0.3"
    android:src="@drawable/g"
    android:layout_marginTop="5dp"
    android:layout_marginBottom="5dp"/>
<TextView
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="0.7"
    android:text="@string/consonantes"
    android:textColor="@color/blanco"
    android:textSize="25dp"
    android:gravity="center_horizontal"
    android:layout_gravity="center_vertical"/>
</LinearLayout>
<!--Consonantes dobles-->

```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:background="@color/azul"
    android:layout_marginTop="20dp"
    android:onClick="consonantes_dobles">
    <ImageView
        android:layout_width="0dp"
        android:layout_height="75dp"
        android:layout_weight="0.3"
        android:src="@drawable/p"
        android:layout_marginTop="5dp"
        android:layout_marginBottom="5dp"/>
    <TextView
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="0.7"
        android:text="@string/consonantes_dobles"
        android:textColor="@color/blanco"
        android:textSize="25dp"
        android:gravity="center_horizontal"
        android:layout_gravity="center_vertical"/>
</LinearLayout>
</LinearLayout>

```

## Código Java

```

package com.example.myl.appcoreano;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;

public class alfabeto extends AppCompatActivity {

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

    //Ir a vocales
    public void vocales(View v)

```

```

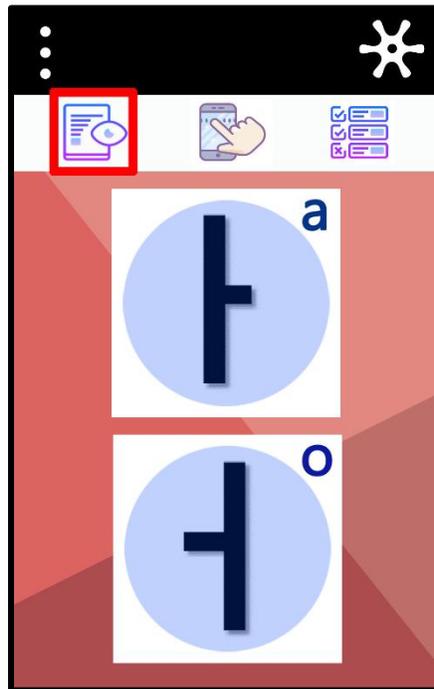
    {
        Intent intent=new Intent(alfabeto.this,vocales.class);
        startActivity(intent);
    }

    //Ir a consonantes dobles
    public void consonantes_dobles(View v)
    {
        Intent intent=new Intent(alfabeto.this,vocales.class);
        startActivity(intent);
    }

    //Ir a consonantes
    public void consonantes(View v)
    {
        Intent intent=new Intent(alfabeto.this,vocales.class);
        startActivity(intent);
    }

    //Ir a primeros_pasos
    public void primeros_pasos(View v)
    {
        Intent intent=new Intent(alfabeto.this,vocales.class);
        startActivity(intent);
    }
}

```



Vocales

Código XML

Vocales.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_height="match_parent"
    android:layout_width="wrap_content"
    android:orientation="vertical">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@color/negro">
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/titulo"
            android:textColor="@color/blanco"
            android:textSize="20dp"
            android:gravity="center_horizontal"
            android:textStyle="bold"
            android:fontFamily="sans-serif"
            android:layout_marginTop="10dp"
```

```

        android:layout_marginBottom="10dp"/>
    </LinearLayout>
<android.support.design.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/base_rosa"
    android:orientation="vertical">
    <android.support.v7.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_scrollFlags="scroll|enterAlways">
    </android.support.v7.widget.Toolbar>
    <android.support.design.widget.TabLayout
        android:id="@+id/tabLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
    </android.support.design.widget.TabLayout>
    <android.support.v4.view.ViewPager
        android:id="@+id/viewPager"
        android:layout_width="match_parent"
        android:layout_height="match_parent"

app:layout_behavior="@string/appbar_scrolling_view_behavior">
    </android.support.v4.view.ViewPager>
</android.support.design.widget.CoordinatorLayout>
</LinearLayout>

```

Fragment\_rv\_vocales.xml

```

<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <android.support.v7.widget.RecyclerView
        android:id="@+id/rvVocales"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginTop="50dp">
    </android.support.v7.widget.RecyclerView>
</LinearLayout>

```

## Activity\_tarjeta.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:gravity="center_horizontal"
    android:layout_marginTop="10dp"
    android:layout_marginBottom="10dp">
    <ImageView
        android:id="@+id/img_tarjeta"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/a"/>
</LinearLayout>
```

## Código Java

### Vocales.java

```
package com.example.myl.appcoreano;

import android.support.design.widget.TabLayout;
import android.support.v4.app.Fragment;
import android.support.v4.view.ViewPager;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.GridLayoutManager;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.support.v7.widget.Toolbar;

import java.util.ArrayList;

import layout.escribirFragment;
import layout.testFragment;

public class vocales extends AppCompatActivity {

    //Manejo del ViewPager
    private Toolbar toolbar;
    private TabLayout tabLayout;
    private ViewPager viewpager;
```

```

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

    toolbar = (Toolbar) findViewById(R.id.toolbar);
    tabLayout = (TabLayout) findViewById(R.id.tabLayout);
    viewPager = (ViewPager) findViewById(R.id.viewPager);
    configurarViewPager();

    if(toolbar != null){
        setSupportActionBar(toolbar);
    }
    getSupportActionBar().setTitle("");
}

private ArrayList<Fragment> agregarFragment(){
    ArrayList<Fragment> fragments = new ArrayList<>();
    fragments.add(new rv_vocales_fragment());
    fragments.add(new escribirFragment());
    fragments.add(new testFragment());
    return fragments;
}

private void configurarViewPager(){
    viewPager.setAdapter(new
PageAdapter(getSupportFragmentManager(), agregarFragment()));
    tabLayout.setupWithViewPager(viewPager);
    tabLayout.getTabAt(0).setIcon(R.drawable.ver);
    tabLayout.getTabAt(1).setIcon(R.drawable.pantalla_t_ctil);
    tabLayout.getTabAt(2).setIcon(R.drawable.para_hacer);
}
}

```

ver.java

```
package com.example.myl.appcoreano;

public class ver {

    //Abstracción de datos
    private int imagen;

    //Constructor
    public ver(int imagen){
        this.imagen = imagen;
    }

    //Getters y Setters
    public int getImagen() {
        return imagen;
    }

    public void setImagen(int imagen) {
        this.imagen = imagen;
    }
}
```

tarjeta\_adaptador.java

```
package com.example.myl.appcoreano;

import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;

import java.util.ArrayList;

public class tarjeta_adaptador extends
RecyclerView.Adapter<tarjeta_adaptador.tarjetaViewHolder>{

    //Lista de ver
    ArrayList<ver> tarjetas;
    //constructor
    public tarjeta_adaptador(ArrayList<ver> tarjetas){
```

```

        this.tarjetas=tarjetas;
    }

    //Conecta con cada tarjeta del recyclerview, es decir infla el
    layout
    //y lo pasa al viewHolder para obtener los view
    @Override
    public tarjetaViewHolder onCreateViewHolder(ViewGroup parent,
int viewType) {
        View v =
        LayoutInflater.from(parent.getContext()).inflate(R.layout.activity
        _tarjeta,parent,false);
        return new tarjetaViewHolder(v);
    }

    //Asocia cada elemento de la lista a cada view
    @Override
    public void onBindViewHolder(tarjetaViewHolder holder, int
    position) {
        ver elemento_ver=tarjetas.get(position);

        holder.img_tarjeta.setImageResource(elemento_ver.getImagen());
    }

    //Cantidad de la lista de tarjetas ver
    @Override
    public int getItemCount() {
        return tarjetas.size();
    }

    //Clase anidada estática (utilizarla dentro)
    public static class tarjetaViewHolder extends
    RecyclerView.ViewHolder{

        private ImageView img_tarjeta;

        public tarjetaViewHolder(View itemView) {
            super(itemView);
            //Asociar parte gráfica y parte lógica
            img_tarjeta = (ImageView)
            itemView.findViewById(R.id.img_tarjeta);
        }
    }
}

```

rv\_vocales\_fragment.java

```
package com.example.myl.appcoreano;

import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.v4.app.Fragment;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import java.util.ArrayList;

public class rv_vocales_fragment extends Fragment{
    //Tarjetas para vocales
    ArrayList<ver> tarjetas;
    private RecyclerView tarjeta_vocales;

    @Nullable
    @Override
    public View onCreateView(LayoutInflater inflater, @Nullable
ViewGroup container, @Nullable Bundle savedInstanceState) {
        //return super.onCreateView(inflater, container,
savedInstanceState);
        //Equivalente a setContentView para asociar xml y java
View v =
inflater.inflate(R.layout.fragment_rv_vocales, container, false);
        //Para el recyclerView
        tarjeta_vocales = (RecyclerView)
v.findViewById(R.id.rvVocales);
        LinearLayoutManager llm= new
LinearLayoutManager(getActivity());
        llm.setOrientation(LinearLayoutManager.VERTICAL);
        tarjeta_vocales.setLayoutManager(llm);
        //GridLayoutManager glm = new
GridLayoutManager(getActivity(),2);
        //tarjeta_vocales.setLayoutManager(glm);
        inicializarListaImagenes();
        inicializarAdaptador();

        return v;
    }
}
```

```

    public void inicializarListaImagenes() {
        tarjetas = new ArrayList<ver>();
        tarjetas.add(new ver(R.drawable.a));
        tarjetas.add(new ver(R.drawable.i));
        tarjetas.add(new ver(R.drawable.u));
    }

    public void inicializarAdaptador() {
        tarjeta_adaptador adaptador= new
tarjeta_adaptador(tarjetas);
        tarjeta_vocales.setAdapter(adaptador);
    }
}

```

page\_adapter.java

```

package com.example.myl.appcoreano;

import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentPagerAdapter;

import java.util.ArrayList;

//Para incluir cada fragment en el viewpager
public class PageAdapter extends FragmentPagerAdapter{

    private ArrayList<Fragment> fragments;

    public PageAdapter(FragmentManager fm, ArrayList<Fragment>
fragments) {
        super(fm);
        this.fragments = fragments;
    }

    @Override
    public Fragment getItem(int position) {
        return fragments.get(position);
    }

    @Override
    public int getCount() {
        return fragments.size();
    }
}

```

## Colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#3F51B5</color>
    <color name="colorPrimaryDark">#303F9F</color>
    <color name="colorAccent">#FF4081</color>
    <color name="negro">#000000</color>
    <color name="blanco">#FFFFFF</color>
    <color name="boton_azul">#4081de</color>
    <color name="naranja"> #f1531c </color>
    <color name="azul"> #110e6f </color>
</resources>
```

## Strings.xml

```
<resources>
    <string name="app_name">AppCoreano</string>
    <string name="titulo">HANGUL</string>
    <string name="primeros_pasos">PRIMEROS PASOS</string>
    <string name="vocales">VOCALES</string>
    <string name="consonantes">CONSONANTES</string>
    <string name="consonantes_dobles">CONSONANTES DOBLES</string>
</resources>
```



## Referencias

- Want to know about Hangeul?, Consultado en diciembre del 2017 en: [http://www.korean.go.kr/eng\\_hangeul/](http://www.korean.go.kr/eng_hangeul/)
- Hangul: El alfabeto coreano, Consultado en diciembre del 2017 en: <http://han-association.com/2017/03/22/hangul-alfabeto-coreano/>
- Apuntes de Coreano. Damian Castilleja Gerardo S. 2017
- Apuntes de Cómputo. Martínez Quintana Marco Antonio, 2017
- La lengua escrita en corea y la creación del alfabeto coreano: Hanguel. Consultado en diciembre del 2018 en: <http://revistas.uma.es/index.php/baetica/article/view/281/232><http://kokoro-no-saku.blogspot.mx/search/label/Aprender%20Coreano>
- Marco conceptual de la computación móvil, Consultado en enero del 2018 en: <https://repository.eafit.edu.co/bitstream/handle/10784/2789/5.%20capitulos.pdf?sequence=5&isAllowed=y>
- Computación Móvil. Consultado en enero del 2018 en: [https://scielo.conicyt.cl/scielo.php?pid=S0718-33052012000300001&script=sci\\_arttext&tlng=pt](https://scielo.conicyt.cl/scielo.php?pid=S0718-33052012000300001&script=sci_arttext&tlng=pt)
- The Changes To Mobile Phones Over The Last 30 Years. Consultado el 22 marzo 2018 en: <http://www.thepeoplehistory.com/mobilephones.html>
- Infographic: A History of Mobile Productivity. Consultado el 22 de marzo 2018 en: <https://www.companionlink.com/blog/2011/08/a-history-of-mobile-productivity/>
- A Brief History of Mobility. Consultado el 22 de marzo 2018 en: <http://www.baselinemag.com/c/a/Mobile-and-Wireless/A-Brief-History-of-Mobility-122428>
- The History of Laptop Computers. Consultado el 22 de marzo 2018 en: <https://www.thoughtco.com/history-of-laptop-computers-4066247>

- History of Mobile Applications. Consultado el 22 de marzo 2018 en: <http://www.uky.edu/~jclark/mas490apps/History%20of%20Mobile%20Apps.pdf>
- Estadísticas de uso de sistemas operativos móviles a nivel nacional e internacional. Consultado en marzo del 2018: <http://gs.statcounter.com/>
- Sistemas Operativos Móviles, Consultado en abril del 2018 en: <http://www.areatecnologia.com/informatica/sistemas-operativos-moviles.html>
- Aprendizaje móvil y desarrollo de habilidades en foros asincrónicos de comunicación. Consultado en mayo del 2018 en: <http://www.redalyc.org/html/158/15812486011/>
- Herramienta de autor para la identificación de estilos de aprendizaje utilizando mapas auto-organizados en dispositivos móviles. Consultado en junio del 2018 en: [http://www.scielo.org.mx/scielo.php?pid=S1607-40412011000100003&script=sci\\_arttext&tlng=pt](http://www.scielo.org.mx/scielo.php?pid=S1607-40412011000100003&script=sci_arttext&tlng=pt)
- Mitos y posibilidades del mLearning. Consultado en agosto del 2018 en: <https://boletin1020.bunam.unam.mx/pdf/36.pdf>
- Inducción en TIC. Consultado en agosto del 2018 <https://inducccion.educatic.unam.mx/mod/glossary/view.php?id=1085>
- Android Developers. Consultado en julio del 2018 en: <https://developer.android.com/guide/components/activities?hl=es-419>
- Moonjinmedia.(2000).Korean Level 1.
- Kim, H., (2011). Coreano para principiantes. Ediciones Salamanca, Salamanca.
- Lee, Eun Kyung. (ED), Idioma Coreano nivel inicial.
- Sommerville, Ian (2005), Ingeniería de Software. Pearson Education, Madrid
- Pressman, Roger S. (2010), Ingeniería de Software. Un enfoque práctico. McGraw Hill. México