

UNIVERSIDAD POLITÉCNICA DE PACHUCA



## Proyecto Final-Codeland-oop

ING. KEVIN EMMANUEL JARAMILLO LIEVANO

DESARROLLO DE APLICACIONES MÓVILES

INTEGRANTES:

JUAN ANTONIO AYOLA CORTES

JUAN DANIEL SOTO DIMAS

MIGUEL ÁNGEL VITE HERNÁNDEZ

INGENIERÍA EN SOFTWARE

# Índice general

<b>1. Introducción</b>	<b>2</b>
1.1. Planteamiento del problema . . . . .	2
1.2. Objetivo general . . . . .	2
1.3. Objetivos específicos . . . . .	2
<b>2. Justificación</b>	<b>3</b>
2.1. ¿Porqué una aplicación móvil? . . . . .	3
2.1.1. Las TIC en la educación . . . . .	3
2.1.2. M-Learning en la educación . . . . .	3
2.1.3. La importancia de la programación orientada a objetos . . . . .	4
2.2. ¿Dónde se aplicará? . . . . .	4
<b>3. Marco teórico</b>	<b>5</b>
3.1. Herramientas utilizadas . . . . .	5
<b>4. Metodología de desarrollo y calidad</b>	<b>11</b>
4.1. Análisis . . . . .	11
4.1.1. Requerimientos funcionales . . . . .	11
4.1.2. Requerimientos no funcionales . . . . .	12
4.2. Diseño . . . . .	12
4.2.1. Casos de uso . . . . .	12
4.2.2. Mockup . . . . .	13
4.2.3. Diagrama de navegabilidad . . . . .	15
4.2.4. Diagrama entidad-relación . . . . .	15
4.3. Programación . . . . .	15
4.4. Implementación . . . . .	15
4.5. Cronograma . . . . .	15
4.6. Pruebas . . . . .	16
<b>5. Resultados</b>	<b>17</b>
5.1. Pantallas . . . . .	17
<b>6. Conclusiones</b>	<b>18</b>
6.1. Juan Daniel Soto Dimas . . . . .	18
6.2. Miguel Ángel Vite Hernández . . . . .	18
6.3. Juan Antonio Ayola Cortes . . . . .	18

# Índice de figuras

2.1. Evolución de las TIC en la educación. (Basado en Leinonen 2005) . . . . .	3
4.1. Diagrama de Casos de uso. . . . .	12
4.2. Pantallas de inicio de sesión y sección de ayuda . . . . .	13
4.3. Pantallas del menú principal, temas, subtemas y evaluación. . . . .	13
4.4. Pantallas de la sección de juegos . . . . .	14
4.5. Diagrama de navegabilidad . . . . .	15
4.6. Diagrama de Gantt . . . . .	15

# Índice de cuadros

3.1. Marco teórico . . . . .	5
4.1. Pruebas . . . . .	16

# Capítulo 1

## Introducción

### 1.1. Planteamiento del problema

La programación orientada a objetos es uno paradigma muy importante dentro de la programación por lo tanto resulta muy útil que los estudiantes que cursan carreras técnicas o universitarias enfocadas al ámbito de la programación tengan un alto conocimiento de esto, más sin embargo a pesar de llevarla como materia, les es difícil aprenderla por los amplios conceptos que maneja, por esto nuestro equipo aspira desarrollar una aplicación que sirva para dar apoyo a los alumnos a fin de que refuercen lo que han aprendido en sus clases, así como también puedan aprender cosas nuevas. Por lo tanto, los alumnos se encuentran ante una cantidad abrumadora de conceptos en un periodo corto de tiempo, lo que dificulta su asimilación y el desarrollo de las habilidades para generar líneas de código como lo explica Spigariol and Passerini (2013):

“Los docentes veían en los estudiantes que el uso del lenguaje representaba una curva de aprendizaje abrupta en los primeros momentos de la materia ya que requieren el manejo de una cantidad amplia de conceptos antes de poder realizar algo relativamente sencillo (...). La disociación entre teoría y práctica que se generaba era ciertamente contraproducente y dificultaba el proceso de aprendizaje.”

### 1.2. Objetivo general

Desarrollar una aplicación móvil que sirva como una herramienta para el aprendizaje del paradigma de programación orientada a objetos con el fin de que cualquier persona pueda acceder a la información, aprender y/o reforzar su conocimiento sobre este tema, desarrollada a través de Android Studio

### 1.3. Objetivos específicos

- Desarrollar una aplicación.
- Proporcionar una herramienta a los estudiantes y/o personas interesadas en la programación.

## Capítulo 2

# Justificación

### 2.1. ¿Porqué una aplicación móvil?

#### 2.1.1. Las TIC en la educación

De acuerdo con Teemu Leinonen (2005) en la educación el uso de las nuevas tecnologías de la información ha pasado por las siguientes etapas:

Esta evolución muestra como cada vez el aprendizaje va utilizando más tecnología; pasando de ser ésta simplemente una herramienta de apoyo, hasta ser la plataforma a través de la cual se presentan los contenidos y evalúan los conocimientos.

Mix-Learning. La etapa posterior al e-learning es la aplicación de una mezcla de sus herramientas con sistemas educativos tradicionales. La finalidad es dirigir más específicamente los contenidos a los estudiantes. Es así que el Blend Learning, Mix Learning o Hybrid Learning se presenta como “la combinación efectiva de los diferentes modelos de reparto, modelos de enseñanza y modelos de aprendizaje”(Heinze, A.y C. Procter, 2004, p 1.).

#### 2.1.2. M-Learning en la educación

Según la Organización de las Naciones Unidas para la Educación la Ciencia y la Cultura (UNESCO, 2012) existen 5.9 billones de suscripciones de teléfonos móviles en el mundo, contra los 7.04 billones de habitantes. Además, en el año 2020 los dispositivos móviles serán la principal herramienta de conexión a internet para la mayoría de la población; en Japón, actualmente el 75 % de su población tiene un dispositivo móvil como primer medio de acceso a internet, (SCOPEO, 2011).

Asimismo, en años recientes el uso de la tecnología móvil para fines educativos, conocido como m-learning, ha tenido un grán desarrollo en la educación superior, ya que existen universidades de Europa y América que cuentan con sistemas de educación móvil, (Traxler, 2007).

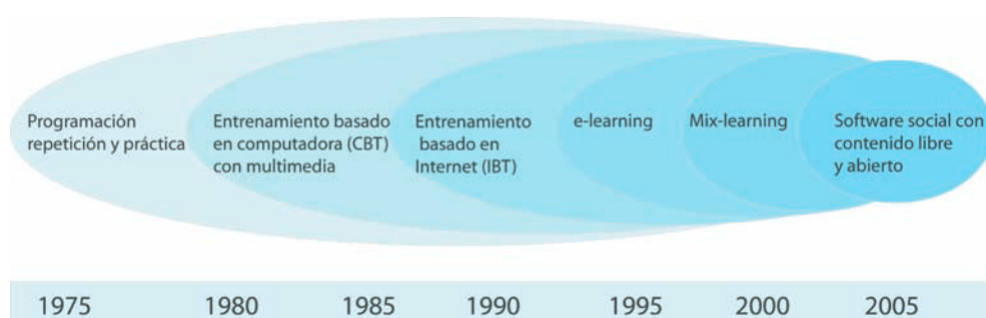


Figura 2.1: Evolución de las TIC en la educación. (Basado en Leinonen 2005)

### **2.1.3. La importancia de la programación orientada a objetos**

La programación Orientada a Objetos surge como el paradigma que permite manejar ampliamente las nuevas plataformas que garanticen desarrollar aplicaciones robustas, portables y reutilizables que puedan ofrecer una solución a largo plazo en un mundo donde los cambios se dan a cada momento.

El desarrollo de programas orientados a objetos es un enfoque diferente del mundo informático. Implica la creación de modelos del mundo real y la construcción de programas informáticos basados en esos modelos.

La importancia de esta programación radica en que, favorece la creación de programas de calidad, fuerza en mantenimiento, en extensión y reutilización de programas. Está basada en el modo de pensar del hombre y en el modo de trabajar de la máquina.

Es muy importante que los estudiantes sean capaces, no sólo de manejar los conceptos de orientación a objetos, sino también de aplicarlos de manera efectiva en el desarrollo de programas.

## **2.2. ¿Dónde se aplicará?**

La aplicación se tiene pensado aplicar en la Universidad Politécnica de Pachuca, en la carrera de Ingeniería en Software, ya que en esta misma es básico aprender este paradigma, ya que constituye las bases para la programación.

## Capítulo 3

# Marco teórico

### 3.1. Herramientas utilizadas

Cuadro 3.1: Marco teórico

Marco Teórico			
Lista de Temas	Subtemas	Referencia 1	Referencia 2
Programación orientada a objetos	Concepto	Varios. (2003). Programación, Orientada a Objetos. marzo 9, 2018, de openlibra Sitio web;: <a href="https://openlibra.com/es/book/programacion-orientada-a-objetos">https://openlibra.com/es/book/programacion-orientada-a-objetos</a>	Velarde de Barraza, Olinda, Murillo de Velásquez Mitzi, Gómez de Meléndez Ludia, Castillo de Krol, Felícita. (2006). INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA A OBJETOS. México;: PEARSON EDUCACIÓN.
	Programa	Joyanes Aguilar Luis (1996). Programación, Orientada a Objetos. 28023 Aravaca (Madrid): McGraw-Hill	Velarde de Barraza, Olinda, Murillo de Velásquez Mitzi, Gómez de Meléndez Ludia, Castillo de Krol, Felícita. (2006). INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA A OBJETOS. México;: PEARSON EDUCACIÓN.
	Tipo de dato	Joyanes Aguilar Luis (1996). Programación, Orientada a Objetos. 28023 Aravaca (Madrid): McGraw-Hill	Velarde de Barraza, Olinda, Murillo de Velásquez Mitzi, Gómez de Meléndez Ludia, Castillo de Krol, Felícita. (2006). INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA A OBJETOS. México;: PEARSON EDUCACIÓN.



**Table 3.1 continued from previous page**

Objeto	Varios. (2003). Programación, Orientada a Objetos. marzo 9, 2018, de openlibra Sitio web:; <a href="https://openlibra.com/es/book/programacion-orientada-a-objetos">https://openlibra.com/es/ book/programacion -orientada-a-objetos</a>	Velarde de Barraza, Olinda, Murillo de Velásquez Mitzi, Gómez de Meléndez Ludia, Castillo de Krol, Felícita. (2006). INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA A OBJETOS. México:; PEARSON EDUCACIÓN.
Clase	Varios. (2003). Programación, Orientada a Objetos. marzo 9, 2018, de openlibra Sitio web:; <a href="https://openlibra.com/es/book/programacion-orientada-a-objetos">https://openlibra.com/es/ book/programacion -orientada-a-objetos</a>	Velarde de Barraza, Olinda, Murillo de Velásquez Mitzi, Gómez de Meléndez Ludia, Castillo de Krol, Felícita. (2006). INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA A OBJETOS. México:; PEARSON EDUCACIÓN.
Herencia	Varios. (2003). Programación, Orientada a Objetos. marzo 9, 2018, de openlibra Sitio web:; <a href="https://openlibra.com/es/book/programacion-orientada-a-objetos">https://openlibra.com/es/ book/programacion -orientada-a-objetos</a>	Velarde de Barraza, Olinda, Murillo de Velásquez Mitzi, Gómez de Meléndez Ludia, Castillo de Krol, Felícita. (2006). INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA A OBJETOS. México:; PEARSON EDUCACIÓN.
Polimorfismo	Varios. (2003). Programación, Orientada a Objetos. marzo 9, 2018, de openlibra Sitio web:; <a href="https://openlibra.com/es/book/programacion-orientada-a-objetos">https://openlibra.com/es/ book/programacion -orientada-a-objetos</a>	Velarde de Barraza, Olinda, Murillo de Velásquez Mitzi, Gómez de Meléndez Ludia, Castillo de Krol, Felícita. (2006). INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA A OBJETOS. México:; PEARSON EDUCACIÓN.

**Table 3.1 continued from previous page**

Encapsulación	Varios. (2003). Programación, Orientada a Objetos. marzo 9, 2018, de openlibra Sitio web:; <a href="https://openlibra.com/es/book/programacion-orientada-a-objetos">https://openlibra.com/es/book/programacion-orientada-a-objetos</a>	Velarde de Barraza, Olinda, Murillo de Velásquez Mitzi, Gómez de Meléndez Ludia, Castillo de Krol, Felícita. (2006). INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA A OBJETOS. México; PEARSON EDUCACIÓN.
Abstracción	Varios. (2003). Programación, Orientada a Objetos. marzo 9, 2018, de openlibra Sitio web:; <a href="https://openlibra.com/es/book/programacion-orientada-a-objetos">https://openlibra.com/es/book/programacion-orientada-a-objetos</a>	Velarde de Barraza, Olinda, Murillo de Velásquez Mitzi, Gómez de Meléndez Ludia, Castillo de Krol, Felícita. (2006). INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA A OBJETOS. México; PEARSON EDUCACIÓN.
Atributos	Joyanes Aguilar Luis (1996). Programación, Orientada a Objetos. 28023 Aravaca (Madrid): McGraw-Hill	Velarde de Barraza, Olinda, Murillo de Velásquez Mitzi, Gómez de Meléndez Ludia, Castillo de Krol, Felícita. (2006). INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA A OBJETOS. México; PEARSON EDUCACIÓN.
Métodos	Joyanes Aguilar Luis (1996). Programación, Orientada a Objetos. 28023 Aravaca (Madrid): McGraw-Hill	Velarde de Barraza, Olinda, Murillo de Velásquez Mitzi, Gómez de Meléndez Ludia, Castillo de Krol, Felícita. (2006). INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA A OBJETOS. México; PEARSON EDUCACIÓN.

**Table 3.1 continued from previous page**

Parámetros	Joyanes Aguilar Luis (1996). Programación, Orientada a Objetos. 28023 Aravaca (Madrid): McGraw-Hill	Velarde de Barraza, Olinda, Murillo de Velásquez Mitzi, Gómez de Meléndez Ludia, Castillo de Krol, Felicita. (2006). INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA A OBJETOS. México;, PEARSON EDUCACIÓN.
Algoritmo	Joyanes Aguilar Luis (1996). Programación, Orientada a Objetos. 28023 Aravaca (Madrid): McGraw-Hill	Velarde de Barraza, Olinda, Murillo de Velásquez Mitzi, Gómez de Meléndez Ludia, Castillo de Krol, Felicita. (2006). INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA A OBJETOS. México;, PEARSON EDUCACIÓN.
Variable	Joyanes Aguilar Luis (1996). Programación, Orientada a Objetos. 28023 Aravaca (Madrid): McGraw-Hill	Velarde de Barraza, Olinda, Murillo de Velásquez Mitzi, Gómez de Meléndez Ludia, Castillo de Krol, Felicita. (2006). INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA A OBJETOS. México;, PEARSON EDUCACIÓN.
Constante	Joyanes Aguilar Luis (1996). Programación, Orientada a Objetos. 28023 Aravaca (Madrid): McGraw-Hill	Velarde de Barraza, Olinda, Murillo de Velásquez Mitzi, Gómez de Meléndez Ludia, Castillo de Krol, Felicita. (2006). INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA A OBJETOS. México;, PEARSON EDUCACIÓN.

**Programación Orientada a Objetos:** La programación orientada a objetos, ha tomado las mejores ideas de la programación estructurada y los ha combinado con varios conceptos nuevos y potentes que incitan a contemplar las tareas de programación desde un nuevo punto de vista. La programación orientada a objetos, permite descomponer más fácilmente un problema en subgrupos de partes relacionadas del problema. Entonces, utilizando el lenguaje se pueden traducir estos subgrupos a unidades auto contenidas llamadas objetos..

**Programa:** Un programa es una secuencia lógica de instrucciones escritas en un determinado lenguaje de programación que dicta a la computadora las acciones que debe llevar a cabo.

**Tipo de dato:** es la representación simbólica de un atributo de una entidad; en programación, los datos expresan características de las entidades sobre las que opera un algoritmo. Los datos representan hechos, observaciones, cantidades o sucesos y pueden tomar la forma de números, letras o caracteres especiales.

**Objeto:** Una estructura de datos y conjunto de procedimientos que operan sobre dicha estructura. Una definición más completa de objeto es: una entidad de programa que consiste en datos y todos aquellos procedimientos que pueden manipular aquellos datos; el acceso a los datos de un objeto es solamente a través de estos procedimientos, únicamente estos procedimientos pueden manipular, referenciar y/o modificar estos datos. Para poder describir todos los objetos de un programa, conviene agrupar éstos en clases.

**Clase:** Podemos considerar una clase como una colección de objetos que poseen características y operaciones comunes. Una clase contiene toda la información necesaria para crear nuevos objetos.

**Encapsulación:** Es una técnica que permite localizar y ocultar los detalles de un objeto. La encapsulación previene que un objeto sea manipulado por operaciones distintas de las definidas. La encapsulación es como una caja negra que esconde los datos y solamente permite acceder a ellos de forma controlada.

**Abstracción:** En el sentido más general, una abstracción es una representación concisa de una idea o de un objeto complicado. En un sentido más específico, la abstracción localiza y oculta los detalles de un modelo o diseño para generar y manipular objetos.

**Objetos:** Un objeto es una entidad lógica que contiene datos y un código que manipula estos datos; el enlazado de código y de datos, de esta manera suele denominarse encapsulación. Cuando se define un objeto, se está creando implícitamente un nuevo tipo de datos.

**Polimorfismo:** Significa que un nombre se puede utilizar para especificar una clase genérica de acciones.

**Herencia:** La herencia es un proceso mediante el cual un objeto puede adquirir las propiedades de otro objeto. (Varios , 2003)

**Atributos:** Describen las características de los objetos: tipo de acceso (privado, protegido, público) y tipo de dato (entero, real, booleano, etcétera).

**Métodos:** Describen lo que puede hacer la clase; es decir, el método define las instrucciones necesarias para realizar un proceso o tarea específicos. La definición del método se compone de tipo de acceso, tipo de retorno, nombre del método, parámetros, si los requiere, y el cuerpo del método.

**Algoritmo:** Se define como una técnica de solución de problemas que consiste en una serie de instrucciones paso por paso y que produce resultados específicos para un problema determinado.

**Variable:** Es un área de almacenamiento temporal a la que se ha asignado un nombre simbólico y cuyo valor puede ser modificado a lo largo de la ejecución de un programa.

**Constante:** Es un valor definido que no cambia durante la ejecución de un programa.

**Parámetros formales:** Son variables que reciben valores desde el punto de llamada que activa al método.

**Java:** Java es uno de los lenguajes de programación más populares del mundo. Es un lenguaje orientado a objetos, potente, versátil y multiplataforma (corre en cualquier sistema operativo moderno). Java fue elegido como el lenguaje para el entorno de desarrollo de Android, el sistema operativo móvil líder en smartphones y tablets.

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

**SQLite:** SQLite es un sistema de gestión de bases de datos relacional compatible con ACID, contenida en una relativamente pequeña ( 275 kiB) biblioteca escrita en C.

**Material Design:** Material Design es un concepto, una filosofía, unas pautas enfocadas al diseño utilizado en Android.

**Git:** Git es un sistema de control de versiones distribuido cuyo objetivo es el de permitir mantener una gran cantidad de código a una gran cantidad de programadores eficientemente.

**UML:** UML son las siglas de “Unified Modeling Language” o “Lenguaje Unificado de Modelado”. Se trata de un estándar que se ha adoptado a nivel internacional por numerosos organismos y empresas para crear esquemas, diagramas y documentación relativa a los desarrollos de software.

**Android Studio:** Es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones para Android y se basa en IntelliJ IDEA.

## Capítulo 4

# Metodología de desarrollo y calidad

### 4.1. Análisis

Metodología propuesta: Xtreme programming Justificación: la metodología usada es Xtreme programming ya que es el más destacado de los procesos ágiles de desarrollo de software, esta metodología nos ofrece una gran cantidad de ventajas como:

- Tener una programación totalmente organizada.
- Fomenta la comunicación entre los clientes y los desarrolladores, este punto es muy importante ya que si no hay una buena comunicación entre los desarrolladores y el cliente va a ser muy difícil que el proyecto cumpla con los requisitos establecidos por el cliente.
- Se hacen pruebas continuas durante el proyecto. Así se obtendrá una menor tasa de errores en el proyecto.
- Es muy recomendable en utilizarlo en proyectos cortos.
- Permite ahorrar mucho tiempo y dinero.

La metodología Xtreme programming tiene muchas ventajas establece las mejores prácticas de Ingeniería de Software en los desarrollos de proyectos, mejora la productividad de los proyectos y garantiza la calidad del software, haciendo que supere las expectativas del cliente.

#### 4.1.1. Requerimientos funcionales

- El sistema contara con un login de Usuario
- El sistema permite el registro de usuarios nuevos  
Permite registrar usuarios mediante Gmail
- El menu principal tendra acceso a los Temas y a los juegos de la aplicaión
- La informción se dividira en Temas de estudio
- Cada tema contara con una evaluación.

- El menu de juegos tendra 3 formas de juego.  
Cuestionario.  
Encuentra el Error.  
Completa la sintaxis.
- Dentro del menu de juego podras entrar al modulo de puntuaciones.
- El menu de Temas mostrara la puntuacion mas alta que tengas de cada Tema.

#### 4.1.2. Requerimientos no funcionales

- Toda la funcionalidad del sistema debe responder al usuario en menos de 5 segundos.
- El sistema debe mostrar mensajes de error que sean informativos y orientados a usuario final.  
El sistema debe poseer interfaces gráficas bien formadas
- La aplicación no podra ocupar más de 2 GB de espacio en disco.

## 4.2. Diseño

### 4.2.1. Casos de uso

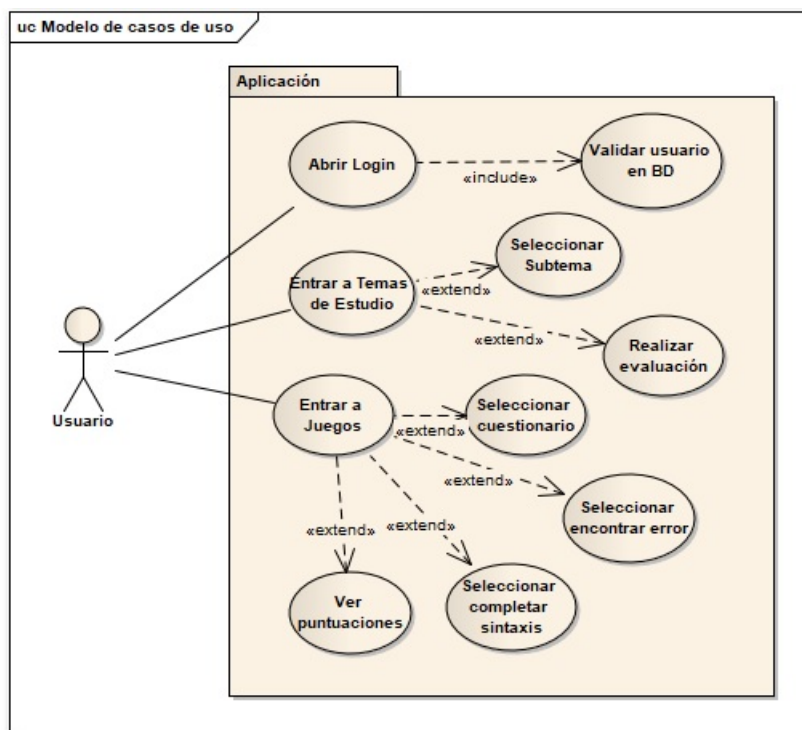


Figura 4.1: Diagrama de Casos de uso.

#### 4.2.2. Mockup

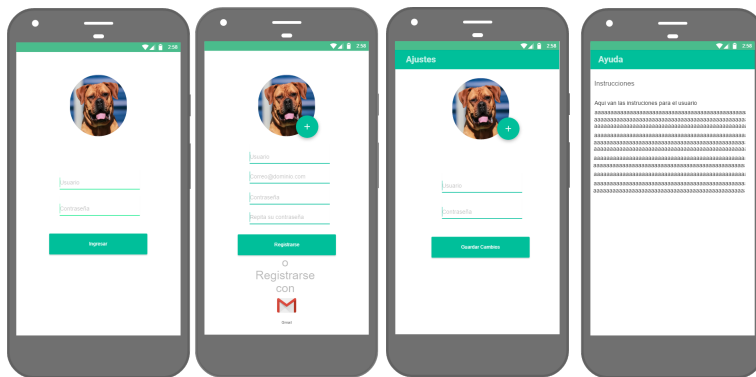


Figura 4.2: Pantallas de inicio de sesión y sección de ayuda

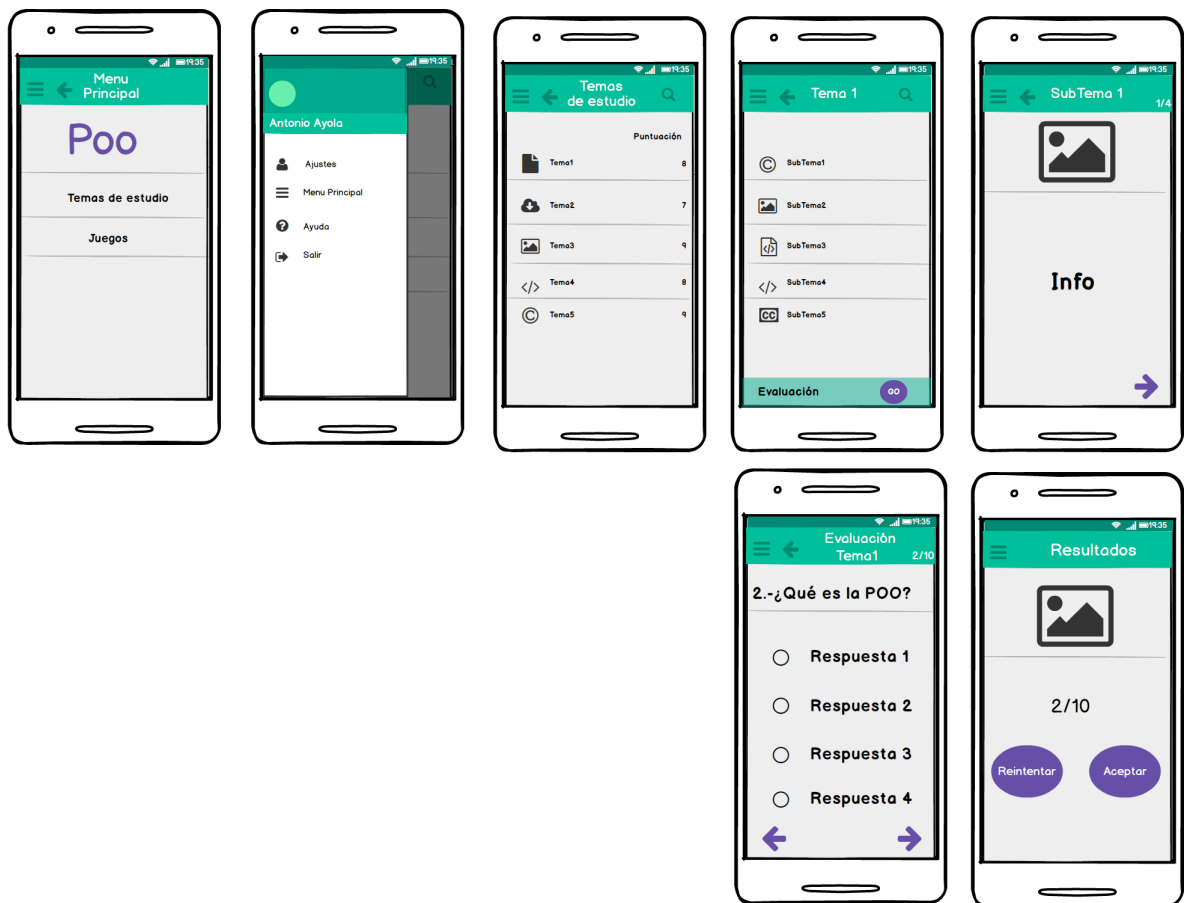


Figura 4.3: Pantallas del menú principal, temas, subtemas y evaluación.



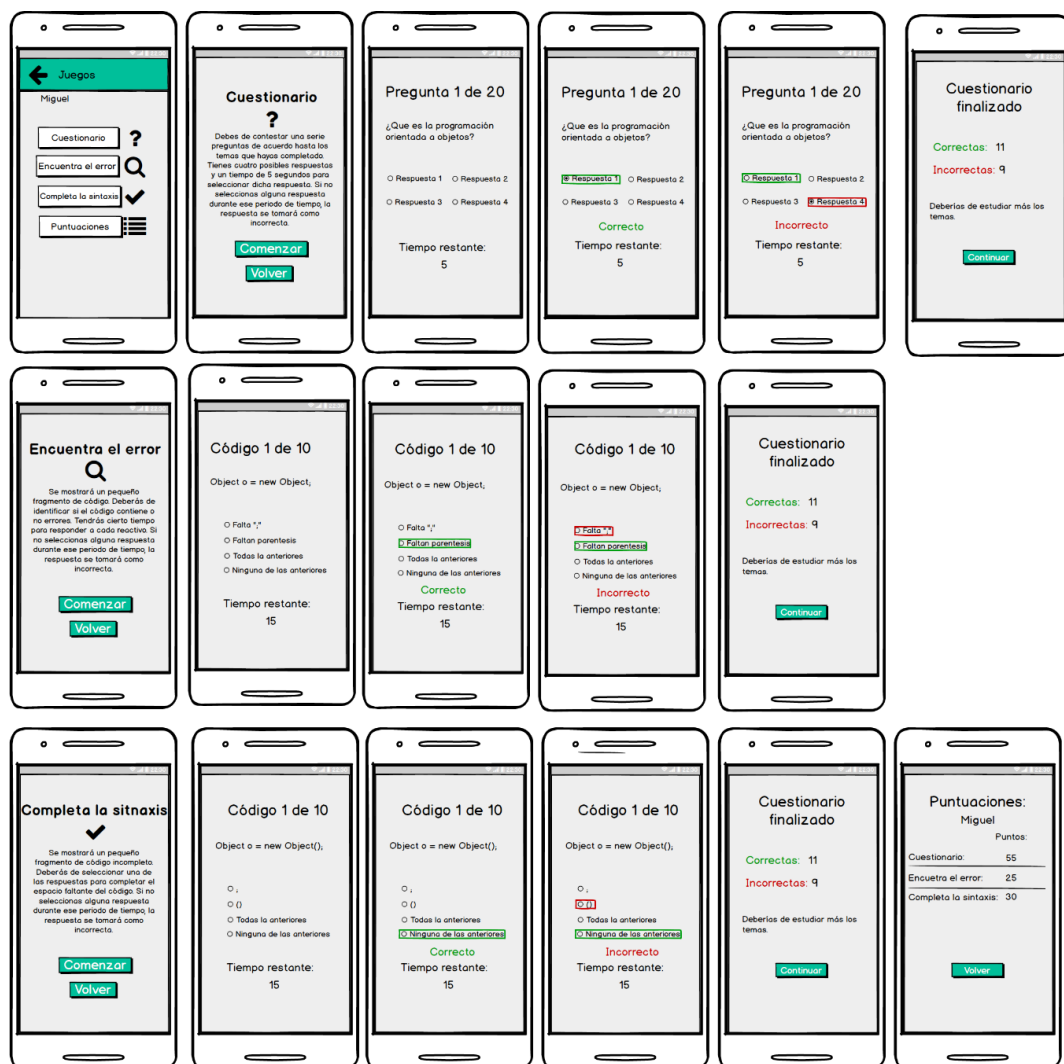


Figura 4.4: Pantallas de la sección de juegos

### 4.2.3. Diagrama de navegabilidad

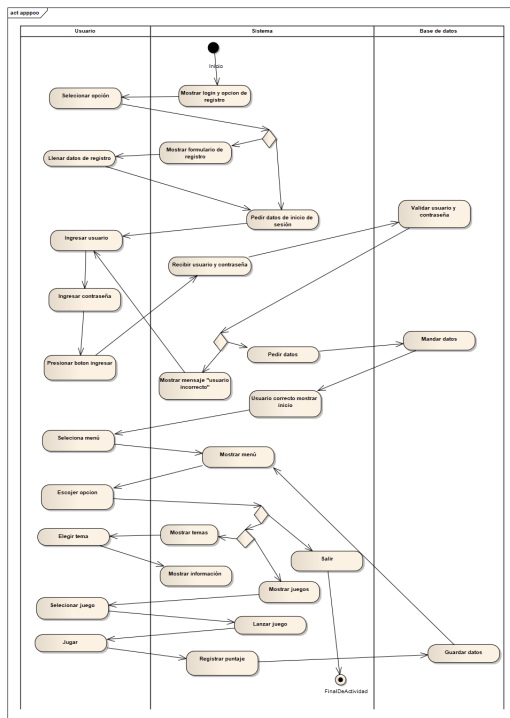


Figura 4.5: Diagrama de navegabilidad

### 4.2.4. Diagrama entidad-relación

## 4.3. Programación

## 4.4. Implementación

## 4.5. Cronograma

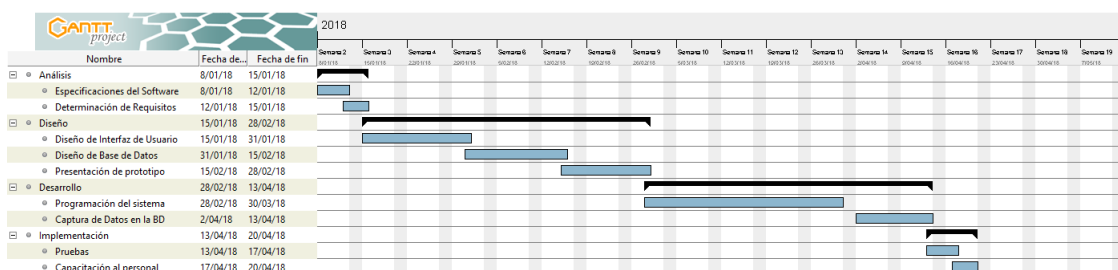


Figura 4.6: Diagrama de Gantt

## 4.6. Pruebas

Cuadro 4.1: Pruebas

Caso de Prueba#	Caso de Prueba	Pasos
1	Login permite acceso a usuario correcto	-Ingresar Usuario correcto -Ingresar contraseña correcta -Ingresar
2	Login no permite acceso a usuario incorrecto	-Ingresar Usuario correcto -Ingresar contraseña incorrecta -Ingresar
3	El menu de temas muestra correctamente todos los temas	-Menu Principal -Seleccionar Temas de estudio

## **Capítulo 5**

# **Resultados**

### **5.1. Pantallas**

## Capítulo 6

# Conclusiones

### 6.1. Juan Antonio Ayola Cortes

En conclusion debo decir que este proyecto me ayudo a ver de lo que se puede lograr con android studio. Android Studio me ha encantado. Facil Intuitivo y me ayudo a entender mejor el mundo de la programación en android.

### 6.2. Juan Daniel Soto Dimas

### 6.3. Miguel Ángel Vite Hernández

Con este proyecto he aprendido a hacer uso de Android Studio, de como funciona, y como desarrollar aplicaciones para Android. Al principio se me dificultó un poco, ya que no comprendía del todo como se desarrolla para Android, pero fui resolviendo mis dudas investigando acerca de la sintaxis y de como organizar los fragments y activities.

# Referencias

Spigariol, L. and Passerini, N. (2013). Enseñando a programar en la orientación a objetos.  
*UTN FRC, Córdoba, Argentina.*