



Facultad de Ingeniería  
Escuela de Ingeniería en Bioinformática  
Ingeniería de Software

---

# Proyecto Juego Espacios Turísticos en 360°

---

***Integrantes:***

Felipe Durán  
Ignacio Gajardo  
Alex Molina

***Profesor:***

Felipe Besoain

***Ayudante:***

José Francisco Riffo

# Índice

<b>1. Introducción</b>	<b>7</b>
1.1. Propósito . . . . .	7
1.2. Descripción breve del problema . . . . .	8
<b>2. Planificación del Trabajo</b>	<b>10</b>
2.1. Descripción del grupo de trabajo . . . . .	10
2.2. Estimación de esfuerzo . . . . .	12
2.3. Asignación de recursos . . . . .	14
2.4. Planificación temporal de actividades . . . . .	14
<b>3. Análisis</b>	<b>16</b>
3.1. Contexto . . . . .	16
3.1.1. Descripción General . . . . .	16
3.1.2. Descripción de Clientes y Usuarios: . . . . .	17
3.2. Especificación de Requerimientos . . . . .	18
3.2.1. Funciones del Sistema . . . . .	18
3.2.2. Atributos del Sistema . . . . .	19

3.2.3. Atributos por Función . . . . .	20
3.3. Actores . . . . .	21
3.4. Casos de Uso . . . . .	22
3.4.1. Caso de Uso Esencial . . . . .	22
3.4.2. Diagrama de Caso de Uso . . . . .	25
3.4.3. Contrato . . . . .	27
3.4.4. Modelo Conceptual . . . . .	30
3.4.5. Diagrama de Secuencia o Colaboración . . . . .	30
3.4.6. Priorización . . . . .	31
3.5. Modelo de Dominio . . . . .	32
3.5.1. Entidades Reconocidas . . . . .	32
3.5.2. Modelo de Dominio . . . . .	32
3.5.3. Matriz de Rastreabilidad . . . . .	33
<b>4. Validación</b>	<b>34</b>
4.1. Prototipo de validación funcional . . . . .	34
<b>5. Diseño</b>	<b>35</b>

5.1. Derivación del Modelo de Software . . . . .	35
5.1.1. Modelo de software inicial . . . . .	35
5.1.2. Diagramas de Clases . . . . .	35
5.1.3. Diagramas de Interacción . . . . .	35
5.1.4. Diagramas de Estados . . . . .	35
5.2. Refinamientos . . . . .	35
5.2.1. Lugar de Refinamiento . . . . .	35
5.2.2. Para cada Lugar . . . . .	35
5.2.2.1. Refinamientos considerados . . . . .	35
5.2.2.2. Selección y descripción de una opción . . . . .	35
<b>6. Implantación</b>	<b>36</b>
6.1. Código fuente completo (parcial) . . . . .	36
6.2. Modelo de implantación . . . . .	36
6.3. Dependencias . . . . .	36
<b>7. Anexos</b>	<b>37</b>
7.1. Glosario . . . . .	37

## Índice de figuras

# Índice de cuadros

1.	Descripción Personal FD . . . . .	10
2.	Descripción Personal IG . . . . .	10
3.	Descripción Personal AM . . . . .	11
4.	Recurso FD_Notebook . . . . .	11
5.	Recurso IG_Notebook . . . . .	11
6.	Recurso AM_Notebook . . . . .	12
7.	Asignacion del personal a sus distintos cargos . . . . .	14
8.	Funciones agrupadas . . . . .	20
9.	Primer caso de uso . . . . .	22
10.	Segundo caso de uso . . . . .	23
11.	Tercer caso de uso . . . . .	23
12.	Cuarto caso de uso . . . . .	24
13.	Quinto caso de uso . . . . .	24
14.	Sexto caso de uso . . . . .	25

# 1. Introducción

## 1.1. Propósito

Este documento se muestra el análisis, modelo de trabajo, tecnologías y diseño utilizado para la producción de la aplicación “Proyecto Juego Espacios Turísticos en 360°” desarrollada para la compañía MauleTec.

El enfoque de este documento se encuentra en el área de programación de el proyecto desarrollado, por lo que se espera de los lectores tener un interés en la creación de aplicaciones móviles para un grupo de usuarios casuales. Si bien la información encontrada requiere conocimiento en terminología relacionada a ingeniería de software, es recomendada para lectores que cuenten con un mínimo de programación básica en el lenguaje de Kotlin, incluyendo conocimiento básico en el área de base de datos ya que los desarrolladores han determinado que el nivel de entrada de este documento es bajo.

La estructura de este informe comienza con la introducción del proyecto describiendo de forma concisa. Luego se presenta el equipo de trabajo y su metodología de producción. El tercer punto es un análisis técnico de la aplicación pasando por su funcionalidad y utilizando una amplia cantidad de material visual para representar el trabajo realizado. Cuarto es la presentación del prototipo para validar su funcionalidad y demostrar su uso de forma práctica. El quinto punto es el diseño trabajado durante el desarrollo, mostrando la evolución del proyecto y cómo se llegaron a ciertas decisiones. En el punto seis refiere a la implantación usando el código de fuente para ser representado. El informe finaliza con un glosario con terminología relacionada al área presentada.

## 1.2. Descripción breve del problema

El proyecto fue presentado por la compañía MauleTec, que entregó un documento base explicando la problemática y los requerimientos de la aplicación. En base a la información entregada en el documento base se concluyó la siguiente problemática. El equipo de mau-leTec cuenta con la habilidad para generar imágenes en 360 grados, por lo que presenta la oportunidad de crear una aplicación utilizando esta tecnología. Al buscar un enfoque lúdico se llega a la conclusión de desarrollar una aplicación estilo juego de mesa, donde una cantidad finita de jugadores exploran un tablero virtual en el que pueden acceder a estas imágenes en 360 grados. Luego de que los usuarios vean estas imágenes se les entregará un grupo de palabras al azar y tendrán que competir para crear la mejor historia utilizando el paisaje presentado en la imagen y las palabras. Luego pasarán a una fase donde los mismos usuarios votarán por la mejor frase y el jugador con la mejor valoración recibirá una victoria, para luego pasar a otra iteración de la misma forma, el juego finaliza cuando uno de los usuarios llega al máximo de victorias. Del análisis se destacan 3 puntos que el grupo de desarrollo concluyó como puntos críticos.

El primero es la realización de una base de datos, que cuenta como la parte central para la creación de esa aplicación, funcionando como conexión entre los usuarios. Para esta área se contará con el conocimiento del equipo de programación para llegar a una conclusión de cómo implementarla, ya sea con el uso de aplicaciones externas o no.

El segundo siendo el diseño de la aplicación ya que solo se entregó una simple descripción de actividades básicas que requiere el software, lo que, aunque entrega una libertad al equipo desarrollador también le pide más trabajo en los aspectos más detallados de este. Para solucionar esta situación se le dará un enfoque en la preproducción del proyecto solo para llegar a una idea más desarrollada del producto final.

Finalmente, el tercero es el material audiovisual necesario para la creación del software



con la necesidad de usar imágenes en 360. Tomando en cuenta que el equipo de desarrollo se encuentra en ciudades distintas y la situación mundial se tendrá que recurrir a la búsqueda de este material por internet, asegurándose de que este se encuentre disponible para su uso público.

## 2. Planificación del Trabajo

### 2.1. Descripción del grupo de trabajo

A continuación se especificará el grupo de trabajo, el cual estará encargado del desarrollo de la aplicación de conquista de espacios turísticos en 360°. Se especificará su ID, nombre, conocimientos, rol y contacto de cada uno de los integrantes del grupo de trabajo.

<b>ID</b>	FD
<b>Nombre</b>	Felipe Durán
<b>Conocimientos</b>	Experiencia en lenguaje de programación como Python, C, C++, C# , Java, JavaScript, Kotlin y Conocimientos con base de datos MySQL.
<b>Rol</b>	Planificador y Programador de la aplicación móvil.
<b>Contacto</b>	fduran16@alumnos.otalca.cl

Cuadro 1: Descripción Personal FD

<b>ID</b>	IG
<b>Nombre</b>	Ignacio Gajardo
<b>Conocimientos</b>	Experiencia en lenguaje de programación como Python, C, C++, C# , Java, JavaScript, Kotlin y Conocimientos con base de datos MySQL.
<b>Rol</b>	Planificador y Programador de la aplicación móvil.
<b>Contacto</b>	igajardo16@alumnos.otalca.cl

Cuadro 2: Descripción Personal IG

<b>ID</b>	AM
<b>Nombre</b>	Alex Molina
<b>Conocimientos</b>	Experiencia en lenguaje de programación como Python, C, C++, C# , Java y Conocimientos con base de datos MySQL.
<b>Rol</b>	Planificador y Programador de la aplicación móvil.
<b>Contacto</b>	amolina16@alumnos.utalca.cl

Cuadro 3: Descripción Personal AM

Los recursos que se utilizarán en el desarrollo del proyecto del software de conquista de espacios turísticos en 360° son:

<b>ID</b>	FD_Notebook
<b>Tipo de dispositivo</b>	Notebook
<b>Sistema operativo</b>	Window 10 Home
<b>Modelo</b>	Asus
<b>Procesador</b>	AMD FX-9830P RADEON R7

Cuadro 4: Recurso FD\_Notebook

<b>ID</b>	IG_Notebook
<b>Tipo de dispositivo</b>	Notebook
<b>Sistema operativo</b>	Window 10 Home
<b>Modelo</b>	MSI
<b>Procesador</b>	Intel Core i7-6700

Cuadro 5: Recurso IG\_Notebook

<b>ID</b>	AM_Notebook
<b>Tipo de dispositivo</b>	Notebook
<b>Sistema operativo</b>	Window 10 Home
<b>Modelo</b>	HP
<b>Procesador</b>	Intel Core i5-7300

Cuadro 6: Recurso AM\_Notebook

## 2.2. Estimación de esfuerzo

Se a realizado un análisis de todos los aspectos posibles que serán parte del desarrollo del software y que competen a la estimación de esfuerzo. Sin embargo, todo lo analizado queda sujeto a modificaciones, debido principalmente a que el proyecto está aún en desarrollo y no se encuentra una versión base o una visión profesionalmente detallada de las iteraciones para desarrollar el producto final. Tanto a nivel de programación como de diseño a de ser necesaria una frecuente revisión y actualización con cada iteración y avance en este proyecto.

Según lo conversado, pactado y analizado con el equipo de desarrollo en la primera iteración, el análisis del proyecto se puede apreciar en las siguientes graficas de estimación de puntos de esfuerzo.

### Software Development (Elaboration and Construction)

Effort = 18.3 Person-months

Schedule = 12.4 Months

Cost = \$14600

Total Equivalent Size = 6912 SLOC

#### Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	1.1	1.6	0.7	\$876
Elaboration	4.4	4.7	0.9	\$3504
Construction	13.9	7.8	1.8	\$11097
Transition	2.2	1.6	1.4	\$1752

Figura 1: Precio de mese hombre

#### Software Effort Distribution for RUP/MBASE (Person-Months)

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.2	0.5	1.4	0.3
Environment/CM	0.1	0.4	0.7	0.1
Requirements	0.4	0.8	1.1	0.1
Design	0.2	1.6	2.2	0.1
Implementation	0.1	0.6	4.7	0.4
Assessment	0.1	0.4	3.3	0.5
Deployment	0.0	0.1	0.4	0.7

Figura 2: Tabla Mes-hombre

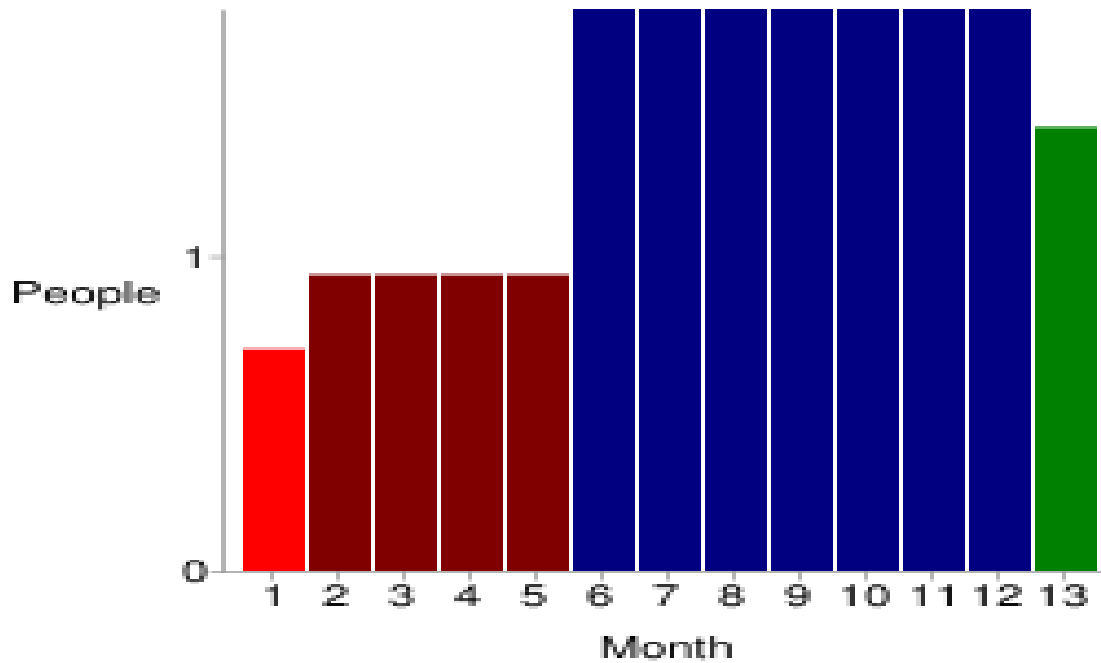


Figura 3: Grafica Mes-hombre

Evaluación Inventario de Transacciones y Ficheros						
	Bajo		Medio		Alto	
	Número	Puntuación	Número	Puntuación	Número	Puntuación
Ficheros Lógicos Internos a la Aplicación	1	7	0	10	0	15
Ficheros Lógicos Externos a la Aplicación	0	5	0	7	0	10
Inputs Externos	5	3	4	4	0	6
Outputs Externos	4	4	0	5	0	7
Queries	0	3	0	4	0	6
		38		16		0
Puntos de Función sin ajustar	<div>54</div>					
VAF	<div>0,79</div>					
Puntos de Función Ajustados	<div>42</div>					
Productividad : Puntos de Función por Mes	<div>16</div>					
Duración en meses	<div>2,63</div>					
Horas por mes	180					
Duración en horas	<div>473</div>					

Figura 4: Estimacion de puntos de esfuerzo

## 2.3. Asignación de recursos

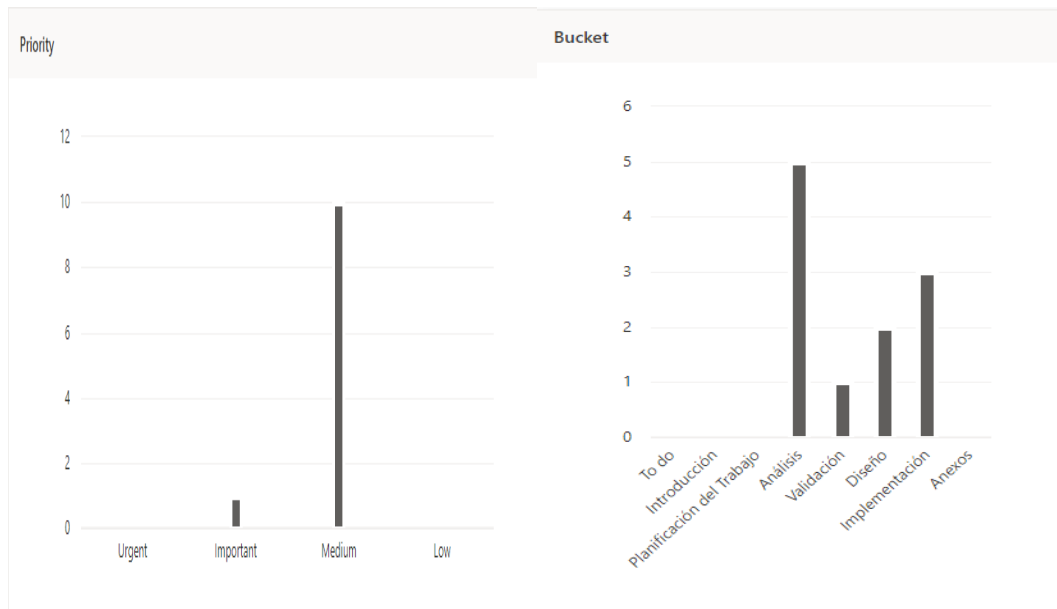
En esta parte va la carta gantt .“Estamos arreglando esta parte”

Tarea	Categoría	Prioridad	Asignado a
Dependencias	Implementación	Medio	Alex Molina
Modelo de Implantación	Implementación	Medio	Ignacio Gajardo
Codigo Fuente	Implementación	Medio	Felipe Duran

Cuadro 7: Asignacion del personal a sus distintos cargos

## 2.4. Planificación temporal de actividades

Gráficos de la planificación actual:



(a) prioridad

(b) Gráfico de barra

Figura 5: Estado de la planificación

## **3. Análisis**

### **3.1. Contexto**

#### **3.1.1. Descripción General**

Como todo proyecto el paso inicial es un análisis del objetivo a realizar, y este caso no es la excepción. La información utilizada para este análisis consiste en una explicación concisa de los requerimientos necesarios de la aplicación, principalmente mecánicas y características que debe contener la versión final.

El resultado del análisis llevó a una descripción propia del proyecto, el resultado final es una aplicación que funcione como herramienta interactiva basada en una enseñanza de exploración y el contenido que esta acción entrega. La parte lúdica del aprendizaje se ve con la implementación de un sistema inspirado en juegos de mesa donde un numero plural de usuarios toman turnos para realizar acciones y avanzar en un mapa para llegar a una meta final. Cada turno se les entrega a los jugadores una imagen en 360 grados de un área o paisaje en particular y una cantidad de conceptos escogidos al azar (estas palabras son encontradas en una base de datos) con esto los usuarios tendrán que concebir una historia usando los datos mencionados. Al final todos tendrán que votar por otro usuario que consideren haber creado la mejor historia, el usuario ganador avanza un espacio y se sigue la misma idea cada turno hasta que uno llegue al final.

En cuanto a los problemas mencionados en el punto 1.1.2 se tomó la esquematización de estos y se analizaron con profundidad. La base de datos a utilizar debe centrarse en una cantidad reducida de usuarios (cuantos jugadores simultáneos se encuentran), también se llegó a la conclusión de que cada usuario cuenta con su propio dispositivo (un smartphone) por lo que se requerirá uso de internet para facilitar los datos a los usuarios. El análisis del proyecto y de la situación del equipo de desarrollo encuentra que el uso de Android studio (con el lenguaje de kotlin) y la implementación de firebase a este es la opción mas viable.



El diseño de la aplicación lleva a la conclusión de comenzar la estructura principal de este (encontrado en el caso de estudio entregado al equipo) dándole una mayor importancia la implementación las mecánicas claves recibidas, asegurándose de que estas funcionen sin problemas antes de ampliar o expandir el desarrollo del proyecto.

Lo que respecta a material audiovisual el equipo encarga en las ultimas fases de desarrollo un tiempo en particular para la búsqueda e implementación de sonidos o imágenes necesarias, y al igual que los otros casos, dándole mayor importancia a las direcciones entregadas al equipo, en este caso siendo el uso de imágenes en 360 grados en la aplicación.

### **3.1.2. Descripción de Clientes y Usuarios:**

Luego de un análisis realizado en base a la información entregada del resultado esperado por la aplicación se a encontrado un perfil definido para tanto clientes como usuarios.

Los clientes son parte del laboratorio mauletec y fueron concretados como un grupo centrado en el turismo, una de las principales mecánicas presentadas. En específico, la vista de paisajes y lugares geográficos en 360 grados le interesa a un grupo enfocado en turismo; como forma de enseñar los paisajes de una zona o región en particular. El otro enfoque encontrado en la aplicación que puede llamar la atención para un cliente es la mecánica de crear historias en base a palabras encontradas en una base de datos, tomando con mayor atención la parte literaria de juego centrada en la creatividad del usuario, pero siendo usada como forma de enseñanza. Para resumir, la motivacion principal de nuestro cliente con este proyecto es dar a conocer la cultura de la region del maule de manera lúdica y entretenida al publico general. Por otro lado, durante el desarrollo del software los usuarios tomaron un enfoque más genérico que el de los clientes, se encontró que en cualquier caso estos tenían requerimientos simples ya que lo más importante para ellos es que la aplicación sea de uso sencillo. El usuario objetivo se identificó como un individuo o individua con un conocimiento básico de tecnología, tomando en cuenta de que la plataforma utilizada es de smartphones. Ademas de contar con un dispositivo que posea un sistema operativo que permita instalar y usar la aplicación, se

requiere que el usuario entienda el concepto de identificarse de forma virtual con su correo electronico ademas de entender las reglas básicas del juego. Además, este debe tener un conocimiento literario de enseñanza básica como mínimo, tomando en cuenta de que la interacción principal de parte del usuario es crear y contar una historia en base a un grupo de palabras entregadas, En definitiva, nuestro usuario es cualquier persona natural can lal capacidades necesarias para utilizar la tecnologia basica y con capacidades de comprension y habla al nivel de un niño en enseñanza basica.

## 3.2. Especificación de Requerimientos

### 3.2.1. Funciones del Sistema

Tablas de funciones evidentes del sistema

Identificador	Nombre	Descripción
F.1.01	Registrar nuevo usuario	Permitir al usuario ingresar con una cuenta nueva
F.1.02	Realizar autentificacion con correo	Autorizar el ingreso del ususario por medio de reconocimiento con cuenta de correo
F.1.03	Realizar autentificacion con contraseña	Autorizar el ingreso del ususario por medio de reconocimiento con contraseña
F.1.04	Realizar autentificacion con Gmail	Autorizar el ingreso del ususario por medio de reconocimiento con cuenta de gmail
F.1.05	Mostrar imagenes en 360°	Principal característica del sistema que permite mostrar por pantalla una imagen es 360°
F.1.06	Sistema de turnos	Mecánica principal del juego para que todos los usuarios participen
F.1.07	Generar lista de palabras	Se genera una lista de palabras sacadas de la base de datos
F.1.08	Mostrar lista de palabras	Le muestra al usuario la lista de palabras generadas para darle sentido al juego
F.1.09	Multijugador global	Función del sistema que permite a varios usuarios jugar una misma partida

Cuadro 8: Funciones evidentes

Tablas de funciones ocultas del sistema

Identificador	Nombre	Descripción
F.2.01	Cargar imagenes en 360°	Sube las imagenes para ser usadas luego por la aplicación
F.2.02	Crear base de datos para palabras	Se crea y guarda una lista de palabras previamente analizadas para ser usadas en el juego
F.2.03	Crear base de datos para puntajes	Se almacenan los puntajes en la base de datos para ser analizados luego

Cuadro 9: Funciones ocultas

Tablas de funciones supérfluas del sistema

Identificador	Nombre	Descripción
F.3.01	Generar una tabla de puntajes	Se muestran al jugador los puntajes ordenados de manera ascendente

Cuadro 10: Funciones supérfluas

### 3.2.2. Atributos del Sistema

Los atributos del sistema son las cualidades con las que cuenta un software, y responden a las características con 5 factores de calidad que son: Funcionalidad, usabilidad, confiabilidad, prestación y soporte; mas comunmente llamado como "modelo FURPS+" por sus iniciales en inglés.

Las cualidades ya mencionadas cubren los factores de calidad y los atributos, restricciones de diseño y requerimientos de implementación, físicos y de interfaz. El unico contratiempo de este modelo, es que no abarca la portabilidad de los productos software que se consideran, esto es una limitacion debido a que no se tienen en cuenta las exigencia del dispositivo.

Tabla de atributos del sistema

Identificador	Nombre	Descripción
A.01	Intuitiva	De fácil entendimiento. El usuario final debe saber que hacer o como empezar a jugar sin mucha complejidad
A.02	Optimizado	La aplicación debe usar recursos del celular solo de ser sumamente necesario
A.03	Baja frecuencia de fallos	El Programa no debe presentar mas de cuatro fallos al mes
A.04	Facilidad de configuración	El usuario debe poder tener la aplicación como a el más le acomode y anexada a su email personal de preferencia
A.05	Facilidad de mantenimiento	A los programadores y gente encargada de mantener la aplicación funcionando se les debe hacer sencillo ingresar al código, entenderlo y aplicar las reparaciones

Cuadro 11: Atributos del sistema

### 3.2.3. Atributos por Función

Ref#	Función	Atributo asociado
R1.1	F.1.01	Facilidad de configuración
R1.2	F.1.02, F.1.03, F.1.04	Facilidad de configuración
R1.3	F.1.05	Intuitiva
R1.4	F.1.06	Intuitiva
R1.5	F.1.07, F.1.08	Intuitiva
R1.6	F.1.09	Facilidad de configuración
R1.7	F.2.01	Facilidad de configuración
R1.8	F.2.02, F.2.03	Facilidad de mantenimiento

Cuadro 12: Atributos por función

### 3.3. Actores

1. Jugador : El fin último de la aplicación es entretener al público, por lo que todos los jugadores, es decir, aquellos que frecuentan la aplicación, deben ser siempre nuestro punto de ancla a la hora de generar ideas para nuestro software.
2. Google FireBase: Google FireBase es uno de los actores que permite la comunicación entre distintas herramientas, y así facilitar la funcionalidad del software. Algunas de las herramientas que utilizamos a través de FireBase son:
  - a) Base de Datos : La aplicación debe mantener contacto con la base de datos de forma directa, para poder acceder a las palabras, imágenes, usuarios, etc.
  - b) Gmail : Herramienta principal para identificar al usuario por medio de correo electrónico.

### 3.4. Casos de Uso

En las siguientes subsecciones identificaremos y analizaremos algunos de los casos de uso detectados y como estos tienen relación con los actores, requisitos, y funciones del sistema vistos con anterioridad en este informe.

### 3.4.1. Caso de Uso Esencial

Identificador	1
Caso de Uso	Jugar un turno de Conquista Turística 360°
Actores	Jugador
Propósito	Jugar un turno
Tipo	Primario y esencial
Descripción	El jugador recibe por pantalla palabras que debe usar para crear una historia. El sistema recibe el feedback de los demás jugador y según lo que puntue este, avanza o no.
Referencias Cruzadas	R1.4 ,R1.5 ,R1.6

Cuadro 13: Primer caso de uso

Acción de los actores	Respuesta del sistema
1.-El jugador inicia un turno de juego	2.-La GUI le presenta al usuario un listado de palabras
3.-El usuario crea una historia usando las palabras presentadas	
4.-Los otros participantes califican la historia	5.-El sistema recibe las puntuaciones y procesa la información
	6.-El jugador recibe una calificación positiva y avanza al siguiente punto
<b>Flujo alternativo</b>	
Línea 6.-El jugador recibe una calificación negativa y se mantiene en su posición hasta la siguiente ronda	

Cuadro 14: Descripción de procesos caso de uso N°1



Identificador	2
Caso de Uso	Iniciar sesión
Actores	Jugador y FireBase
Propósito	Entrar en su cuenta personal
Tipo	Primario y Real
Descripción	El jugador abre la aplicación he ingresa sus datos, la aplicación los recibe y los compara en la base de datos a traves del FireBase, si la informacion es correcta el jugador ingresa a la pantalla principal
Referencias Cruzadas	R1.1 ,R1.2

Cuadro 15: Segundo caso de uso

<b>Acción de los actores</b>	<b>Respuesta del sistema</b>
1.-El jugador abre la aplicación	2.-La GUI le presenta al usuario un formulario de usuario y contraseña para ser rellenado
3.-El usuario rellena los campos señalados	4.-El sistema verifica la información a traves del firebase
	5.-La aplicación recibe los datos del firebase y accede a la pantalla principal
<b>Flujo alternativo</b>	
Línea 2.-Se identifica que el usuario ya habia ingresado con anterioridad y la aplicacion accede directamente a la pantalla principal	
Línea 5.-Se identifica que los datos ingresados son incorrectos y la aplicacion le envia feedback al usuario	

Cuadro 16: Descripción de procesos caso de uso N°2

Identificador	3
Caso de Uso	Generar lista de palabras
Actores	Jugador y FireBase
Propósito	Entregar palabras para que el jugador pueda intervenir
Tipo	Primario y esencial
Descripción	El jugador toca la imagen 360° y la aplicación accede a la base de datos a través del FireBase, busca 4 palabras aleatorias y se las muestra al jugador por pantalla
Referencias Cruzadas	R1.3 ,R1.5

Cuadro 17: Tercer caso de uso

<b>Acción de los actores</b>	<b>Respuesta del sistema</b>
1.-El jugador interactúa con la imagen 360°	2.-El software accede a la base de datos a través del firebase
	3.-Se buscan y seleccionan palabras al azar
	4.-Se le presenta esta lista al usuario

Cuadro 18: Descripción de procesos caso de uso N°3

Identificador	4
Caso de Uso	Mostrar imagen 360°
Actores	Jugador
Propósito	Retroalimentar al jugador
Tipo	Primario y esencial
Descripción	Cuando el jugador conquista una posición, avanza a la siguiente y el programa le presenta a través de una imagen 360° su nuevo objetivo
Referencias Cruzadas	R1.3

Cuadro 19: Cuarto caso de uso

<b>Acción de los actores</b>	<b>Respuesta del sistema</b>
1.-El jugador conquista una posición	
2.-El jugador avanza al siguiente punto en el mapa	3.-La GUI le presenta al usuario la nueva imagen 360°

Cuadro 20: Descripción de procesos caso de uso N°4

Identificador	5
Caso de Uso	Mostrar tabla de puntajes
Actores	Jugador y FireBase
Propósito	Retroalimentar al jugador
Tipo	Secundario y real
Descripción	El jugador accede a la tabla de puntajes y el programa a su vez ingresa a la base de datos a través del FireBase donde están almacenados todos los puntajes y se los presenta al jugador de manera ordenada
Referencias Cruzadas	R1.8

Cuadro 21: Quinto caso de uso

<b>Acción de los actores</b>	<b>Respuesta del sistema</b>
1.-El jugador accede a la tabla de puntajes	2.-El sistema recibe la orden y ingresa a la base de datos
	3.-El sistema adquiere los puntajes y los ordena de mayor a menor
	4.-La GUI le muestra al usuario la tabla ordenada con los puntajes

Cuadro 22: Descripción de procesos caso de uso N°5

Identificador	6
Caso de Uso	Puntuar Jugador
Actores	Jugador y Firebase
Propósito	Retroalimentar al jugador
Tipo	Secundario y concreto
Descripción	los usuarios califican la historia del jugador de turno a través de la GUI
Referencias Cruzadas	F.2.03, F.3.01

Cuadro 23: Sexto caso de uso

<b>Acción de los actores</b>	<b>Respuesta del sistema</b>
1.-El jugador cuenta su historia	2.-La GUI le muestra a los demás jugadores un sistema de calificación
3.-Los demás usuarios en la partida votan la historia narrada	4.-El sistema almacena y analiza las votaciones
	5.-La GUI devuelve el feedback al jugador de turno

Cuadro 24: Descripción de procesos caso de uso N°6

### 3.4.2. Diagrama de Caso de Uso

Esta sección es para representar de manera gráfica como interactúan los autores con el sistema en los distintos casos de uso, para así tener una visión clara de como se lleva a cabo el proceso para realizar las acciones especificadas en los casos de uso que representan.

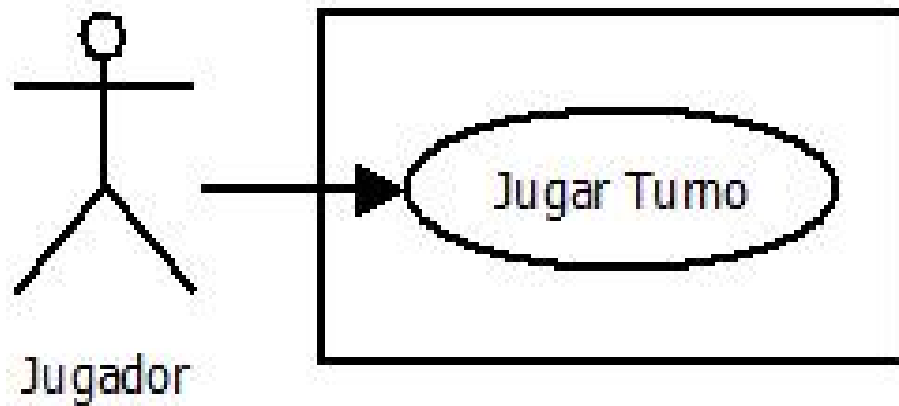


Figura 6: Digrana de casos de uso 1

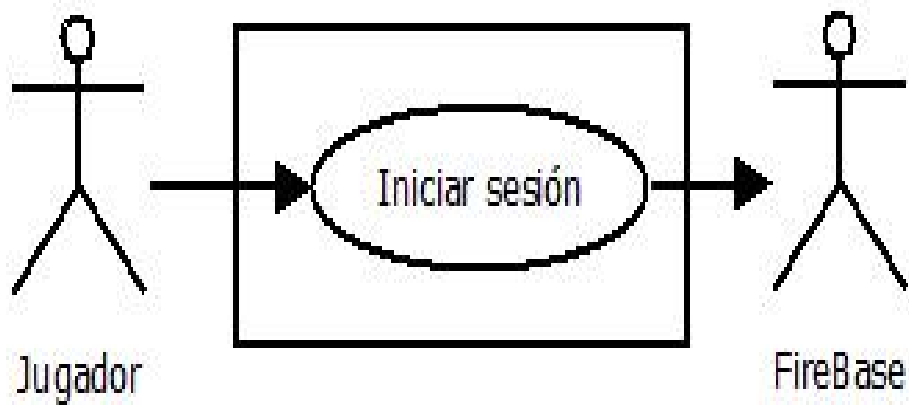


Figura 7: Digrana de casos de uso 2



Figura 8: Digramas de casos de uso 3

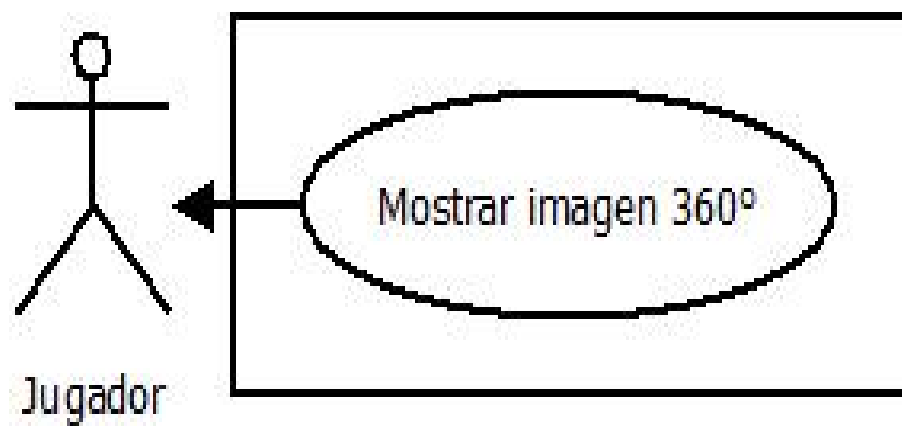


Figura 9: Digramas de casos de uso 4



Figura 10: Digramas de casos de uso 5

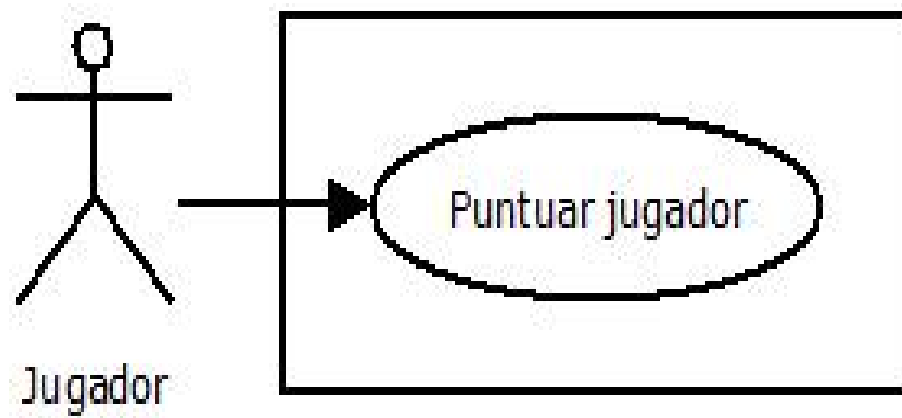


Figura 11: Digramas de casos de uso 6



Este diagrama muestra como serian todos los diagramas vistos anteriormente en un formato generalizado y en convivencia con todos los demás y como interactúan entre si.

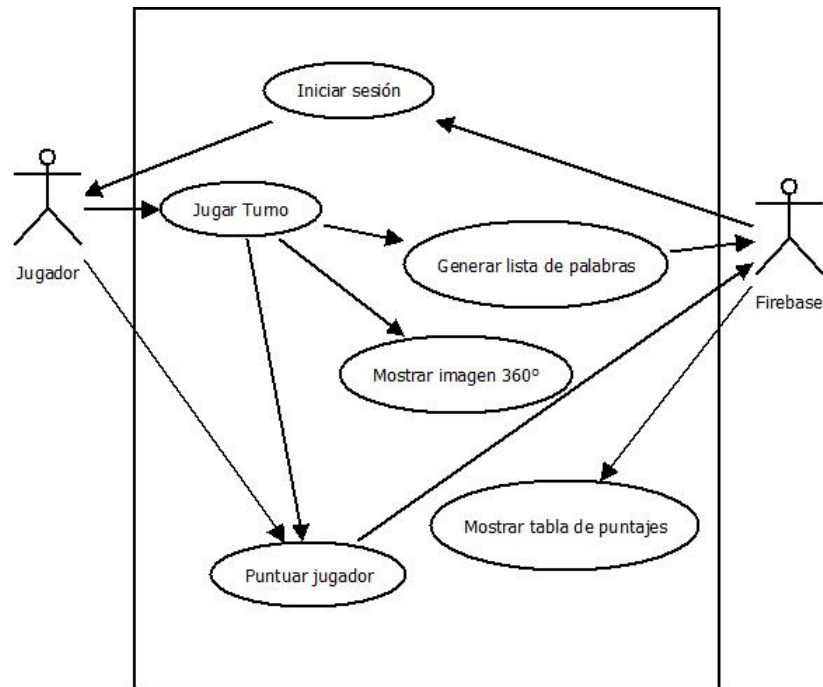


Figura 12: Diagrama general de casos de uso

### 3.4.3. Contrato

Los contratos describen que es lo que se espera de una operación, enfatizando más en el análisis que en el diseño, es decir, responde principalmente al "¿Que debe hacer el sistema?". Para esto, es común utilizar pre y postcondiciones que giran en torno a los cambios de estado. Estos contratos son, en resumidas cuentas, una forma de presentar funciones o acciones concretas y un tanto específicas de lo que hace nuestro software.

A continuación presentamos algunos de ellos.

<b>Nombre:</b> Inicio del juego
<b>Responsabilidades:</b> Comunicarle al usuario de que la aplicación se a iniciado
<b>Tipo:</b> Pantalla inicial
<b>Referencias cruzadas:</b> Nulo
<b>Caso de uso:</b> Nulo
<b>Notas:</b>
<b>Excepciones:</b>
<b>Salida:</b>
<b>Precondiciones:</b> La aplicación se inicia con ningún tipo de error
<b>Poscondiciones:</b> Se entrega acceso a el uso de la aplicación luego de verificar y autentificar los datos. Referencia a tabla ??

Cuadro 25: Contrato: inicio de juego

<b>Nombre:</b> Inicio de sesion
<b>Responsabilidades:</b> Reconocer al usuario o entregarle la posibilidad de crear un perfil para ingresar al juego
<b>Tipo:</b> Sistema
<b>Referencias cruzadas:</b> R1.1 R1.2
<b>Caso de uso:</b> 2. Iniciar sesión
<b>Notas:</b> Existe un mensaje de error al ingresar datos incorrectos
<b>Excepciones:</b> Nulo
<b>Salida:</b> Se avisa a la base de datos el ingreso de un usuario al sistema
<b>Precondiciones:</b> La aplicación debe tener una conexión con la base de datos para reconocer al usuario
<b>Poscondiciones:</b> Se obtuvo los datos del usuario y se avisa de su ingreso a la aplicación

Cuadro 26: Contrato: inicio de sesión

<b>Nombre:</b> Lista de palabras
<b>Responsabilidades:</b> Generar una lista de palabras al azar y entregarselas a los usuarios
<b>Tipo:</b> Sistema
<b>Referencias cruzadas:</b> R1.5 R1.8
<b>Caso de uso:</b> 3. Generar lista de palabras
<b>Notas:</b>
<b>Excepciones:</b> Nulo
<b>Salida:</b> Nulo
<b>Precondiciones:</b> La aplicación contiene una cantidad minima de palabras a elegir y una conexión con los usuarios
<b>Poscondiciones:</b> Se generan las palabras y se entregan a los usuarios

Cuadro 27: Contrato: Lista de palabras

<b>Nombre:</b> Pantalla imagen 360°
<b>Responsabilidades:</b> Se ve la imagen en 360 en cada partida
<b>Tipo:</b> Visual
<b>Referencias cruzadas:</b> R1.3
<b>Caso de uso:</b> 4. Mostrar imagen 360°
<b>Notas:</b> Se debe mover el dispositivo físico para apreciar la imagen completa
<b>Excepciones:</b> Nulo
<b>Salida:</b> Nulo
<b>Precondiciones:</b> Debe existir una imagen cargada previamente y los usuarios que verán esta
<b>Poscondiciones:</b> Se entrefa la imagen con la opción de finalizar su visualización

Cuadro 28: Contrato: Pantalla imagen 360°

<b>Nombre:</b> Presentación de puntajes
<b>Responsabilidades:</b> Entregar feedback luego de la realización de un turno
<b>Tipo:</b> Sistema
<b>Referencias cruzadas:</b> R1.8
<b>Caso de uso:</b> 5. Mostrar tabla de puntajes
<b>Notas:</b>
<b>Excepciones:</b> Nulo
<b>Salida:</b> Puntaje en base a la valoración de los usuarios
<b>Precondiciones:</b> accede a la tabla de puntajes y el programa a su vez se ingresae este valor a la base de datos
<b>Poscondiciones:</b> Los puntajes se analizan y se entregan a los usuarios

Cuadro 29: Contrato: Presentación de puntajes

<b>Nombre:</b> Cargar imagen 360°
<b>Responsabilidades:</b> Carga de una imagen en 360 para ser visualizada por los usuarios
<b>Tipo:</b> Sistema
<b>Referencias cruzadas:</b> R1.7
<b>Caso de uso:</b> 6. Cargar imagen 360°
<b>Notas:</b> Nulo
<b>Excepciones:</b> La ruta de la imagen debe exisitir de lo contrario no funciona
<b>Salida:</b> Nulo
<b>Precondiciones:</b> Una imagen de 360
<b>Poscondiciones:</b> La imagen esta lista para ser presentada al momento de que los usuarios ingresen

Cuadro 30: Contrato: Cargar imagen 360°

#### 3.4.4. Modelo Conceptual

El modelo conceptual que se presenta a continuacion muestra las relaciones que hay entre los diversos organismos, clases y conceptos dentro del software. Presenta como es el flujo entre cada uno de los miembros, es decir, que relacion tiene cada uno de los conceptos. La forma escrita de representarlo seria: El mapa de juego(lo que vendria siendo el tablero), contiene imagenes 360° y a su vez requiere poder usar el fire base para poder usar la base de datos, quien a su vez almacena palabras, jugadores y una tabla de puntajes.

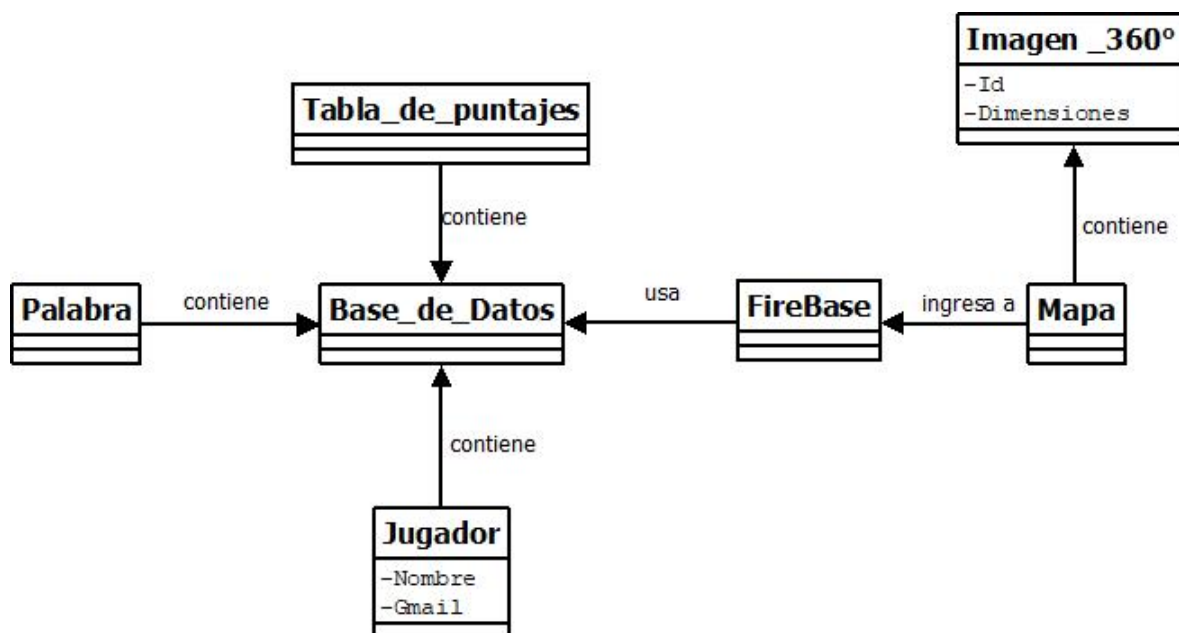


Figura 13: Modelo conceptual

### 3.4.5. Diagrama de Secuencia o Colaboración

Los función y proposito de los siguientes diagramas de secuencia o colaboración sirven para ejemplificar de manera grafica y como apoyo visual la comunicación entre los autores y el sistema de cada caso de uso.

Los siguientes diagramas hacen referencia a los casos de uso 1 y 2 respectivamente  
(Tablas ?? y ??)

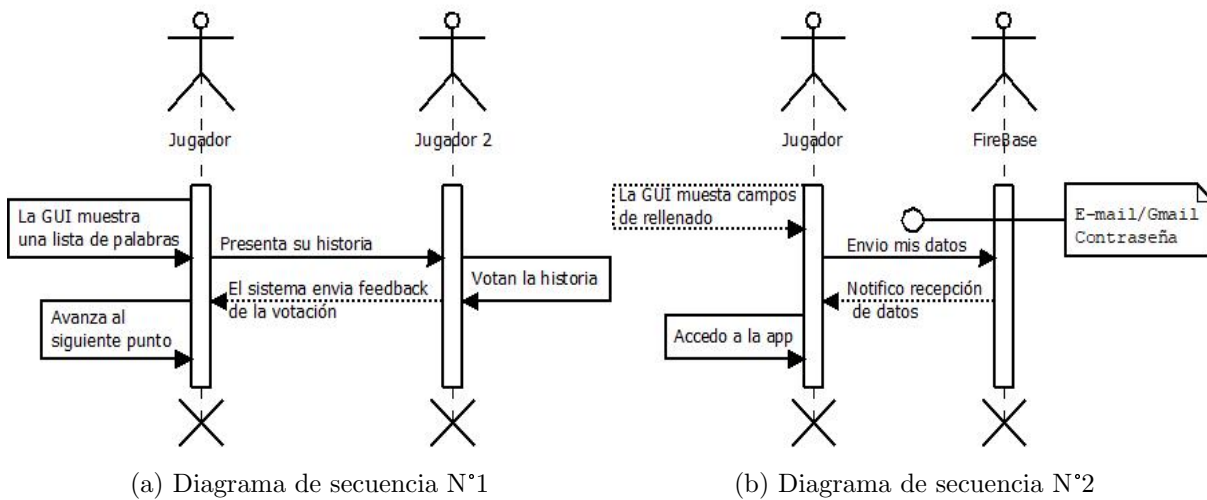


Figura 14: Digramas de secuencia 1 y 2

Los siguientes diagramas hacen referencia a los casos de uso 3 y 4 respectivamente  
(Tablas ?? y ??)

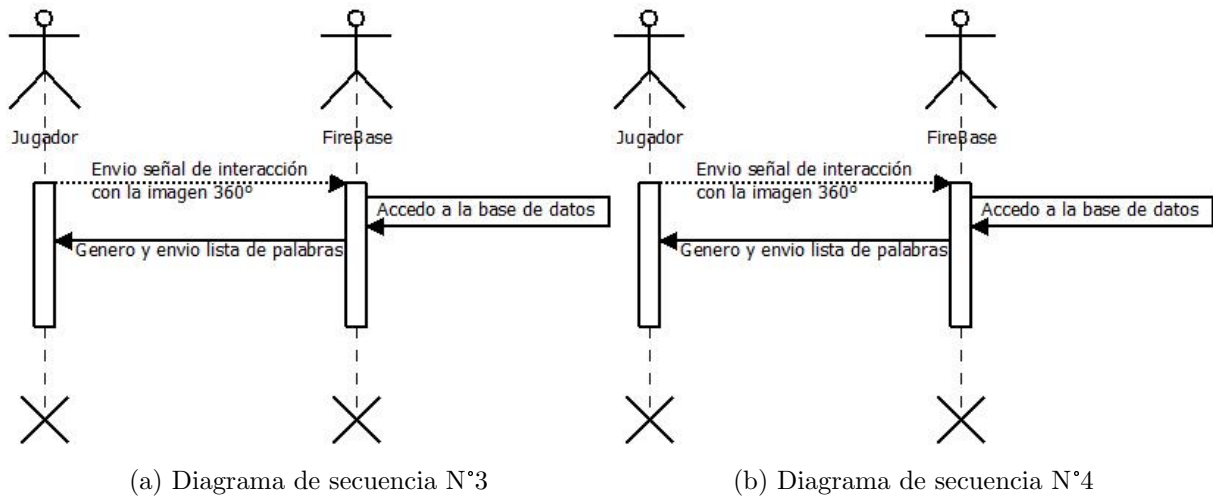


Figura 15: Digramas de secuencia 3 y 4

Los siguientes diagramas hacen referencia a los casos de uso 5 y 6 respectivamente  
(Tablas ?? y ??)

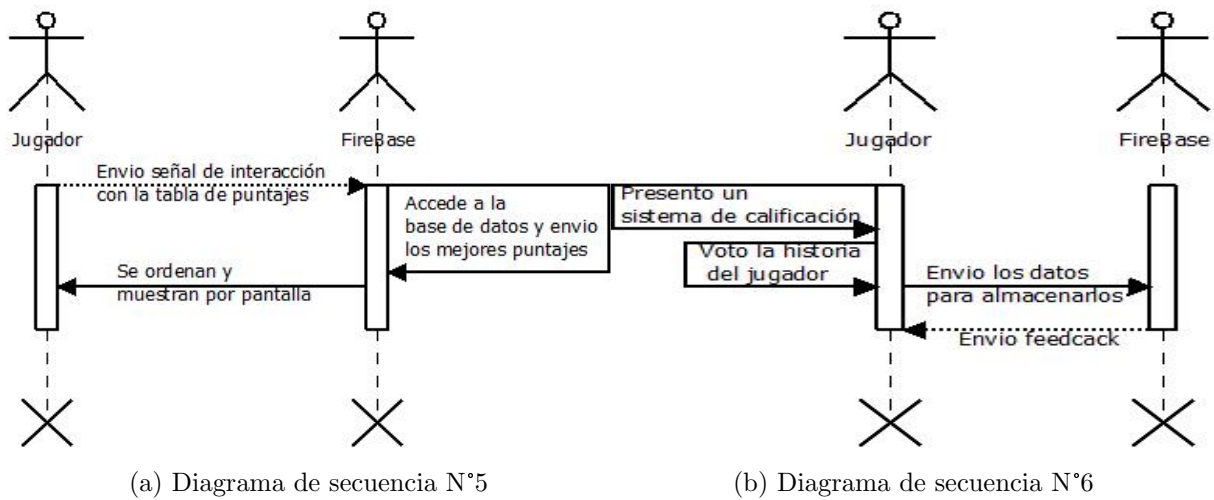


Figura 16: Digramas de secuencia 5 y 6



### 3.4.6. Priorización

Para tener claridad de los que nos resulta mas apremiante o necesario para el software necesitamos tener muy presente el motivo de su creación, que queremos lograr con su desarrollo. Es por esto que debemos evaluar las funciones de nuestro juego y hacernos la gran pregunta "Si elimino esto, ¿Seguira funcionando la aplicación", si la respuesta es **NO**, entonces nos encontramos con una parte **Esencial** para nuestro software, ahora, en caso de que la respuesta sea si, "¿Sigue siendo jugable? o ¿Entiendo como proseguir si lo borro?", en caso de que la respuesta sea una negativa entonces tambien cuenta como un atributo **Esencial** para el desarrollo. Ahora bien, todos los elementos esteticos cuya función es simplemente la de hacer mas agradable y comprensible todo el resto de funciones, forman parte de atributos o funciones **Importantes**.

Los siguientes son los principales funciones y características con las que nuestro software debe contar, utilizando el metodo de priorizacion antes mencionado.

1. Jugar una partida de Conquista Turística 360° **Esencial**.
2. Iniciar sesión **Esencial**
3. Generar lista de palabras **Esencial**
4. Mostrar imagen 360 **Esencial**
5. Mostrar tabla de puntajes **Importante**
6. Cargar imagen 360° **Esencial**

### 3.5. Modelo de Dominio

El modelo de dominio presentado cumple con la presentacion de manera conceptual de todas las clases dentro de nuestro sistema, con el fin de que se comprenda la relacion ente los objetos del mundo real que se tienen y como estos afectan a nuestro desarrolllo y visión de programa, o, en otras palabras, como tomamos importancia de hacer que nuestra software sea lógico y cumpla a su vez con un modelo de programación facil de comprender, aproximandose lo mas posible a lo que significa programación orientada a objetos.

#### 3.5.1. Entidades Reconocidas

Dentro de este software solo pudimos identificar a la base de datos como una entidad, tomando en consideración lo que entendemos por entidad. Base de Datos: Sistema de almacenamiento de recursos necesarios para el funcionamiento de la aplicación.

#### 3.5.2. Modelo de Dominio

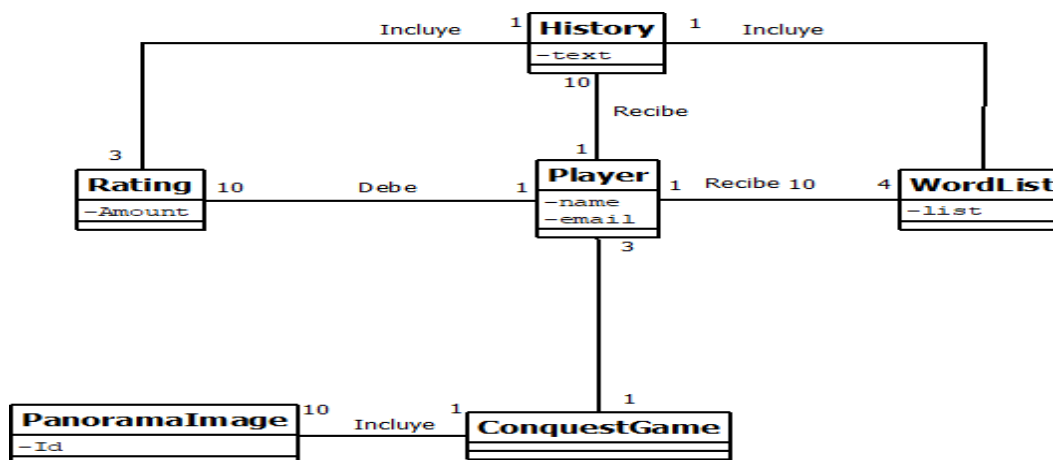


Figura 17: Diagrama de dominio: UML

### 3.5.3. Matriz de Rastreabilidad

	History	Rating	Player	WordList	PanoramaImage	ConquestGame
1	X	X	X	X	X	X
2			X			
3				X		
4					X	
5		X				
6					X	

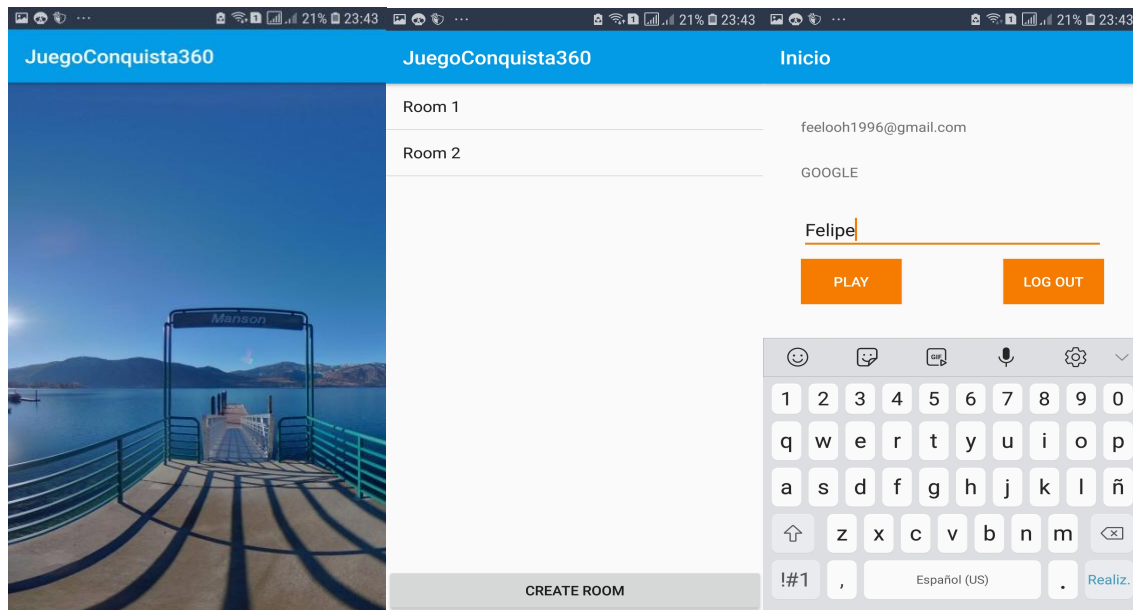
Cuadro 31: Matriz de rastreabilidad

## 4. Validación

Las motivaciones del empleador han sido comprendidas y se han tomado las consideraciones requeridas para responder a su motivación, para esto se han aplicado funciones como iniciar sesión con alguna cuenta, poder visualizar imágenes en 360° de lugares turísticos, característicos o icónicos de la región del maule, presentar un diseño que sea relajante, entre otras funciones mencionadas con anterioridad en este documento.

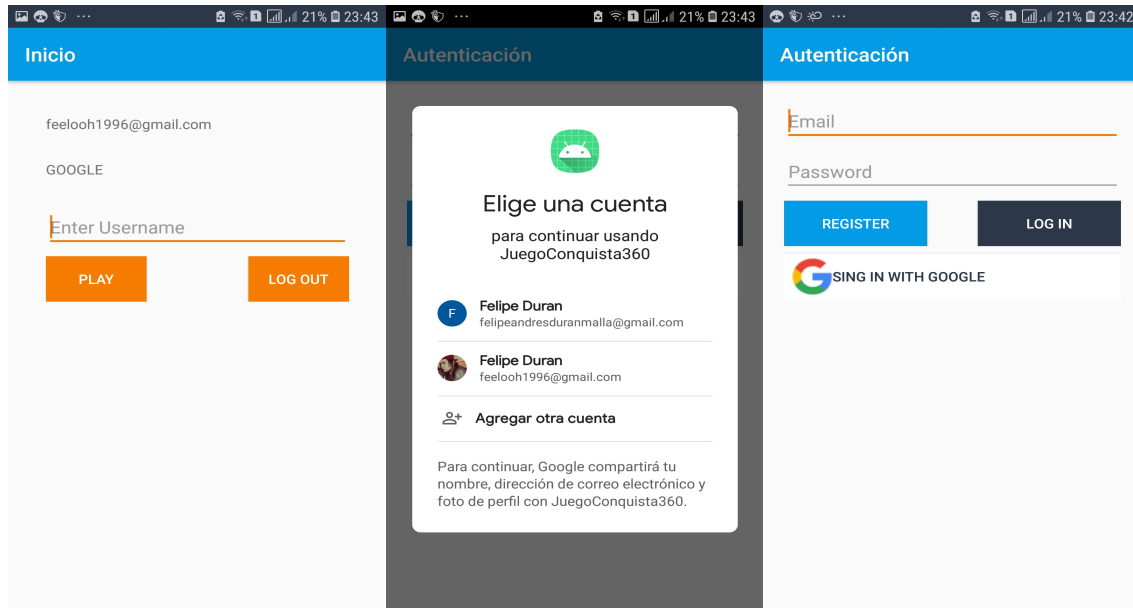
### 4.1. Prototipo de validación funcional

A continuación se presentan algunas capturas de pantalla obtenidas de la aplicación para mostrar el progreso y el diseño realizado por los desarrolladores.



(a) Screenshot de la app N°01 (b) Screenshot de la app N°02 (c) Screenshot de la app N°03

Figura 18: Screenshot 1, 2 y 3



(a) Screenshot de la app N°04    (b) Screenshot de la app N°5    (c) Screenshot de la app N°6

Figura 19: Screenshot 4, 5 y 6

## 5. Diseño

### 5.1. Derivación del Modelo de Software

#### 5.1.1. Modelo de software inicial

Tomando como base el modelo de dominio, el modelo de diseño inicial modifica mayormente el orden de las clases para definir de forma más lineal su funcionamiento, sin perder la función cíclica que la aplicación ya contiene. La aplicación comienza con la identificación del usuario que además entrega una forma de reconocerse ante la base de datos. Luego de ingresar el jugador debe seleccionar una de las salas encontradas en la clase Room, donde además funciona como una forma de esperar a otros posibles jugadores. Al entrar al juego se mantiene el orden cíclico del modelo de dominio donde se puede ver la imagen en 360° y las palabras entregadas para luego crear la historia. Finalmente se entrega una valoración para los otros jugadores de sus historias que culmina con la presentación del ganador de la ronda, es decir, el jugador con la mejor valoración, para lo cual se pase a otra ronda.

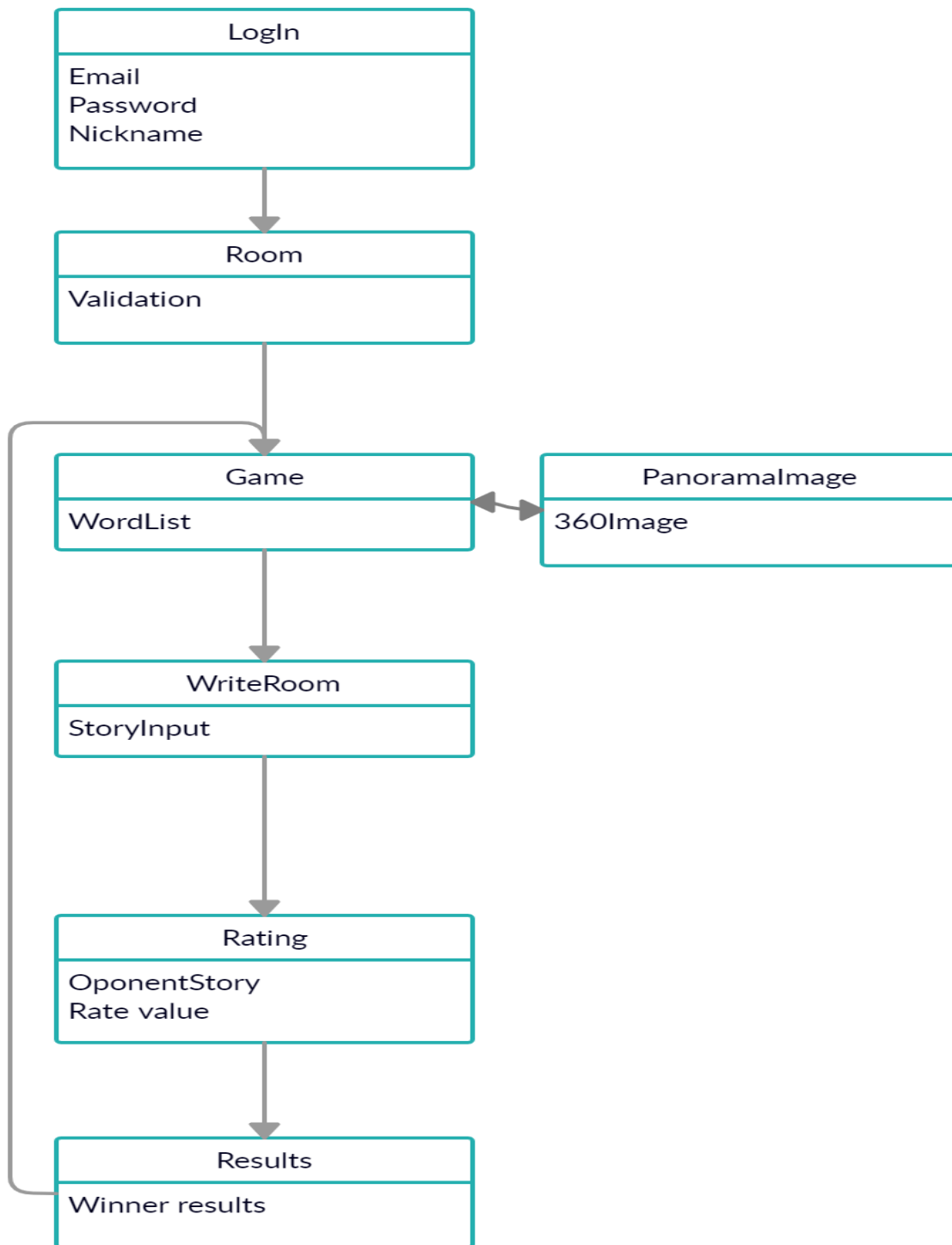


Figura 20: Modelo de software inicial

### 5.1.2. Diagramas de Clases

A continuación se presenta el diagrama de clases utilizado en el desarrollo de nuestra aplicación, con el proposito de poder estudiar y analizar su estructura, funciones, atributos y composicion de manera mas gráfica y sencilla.

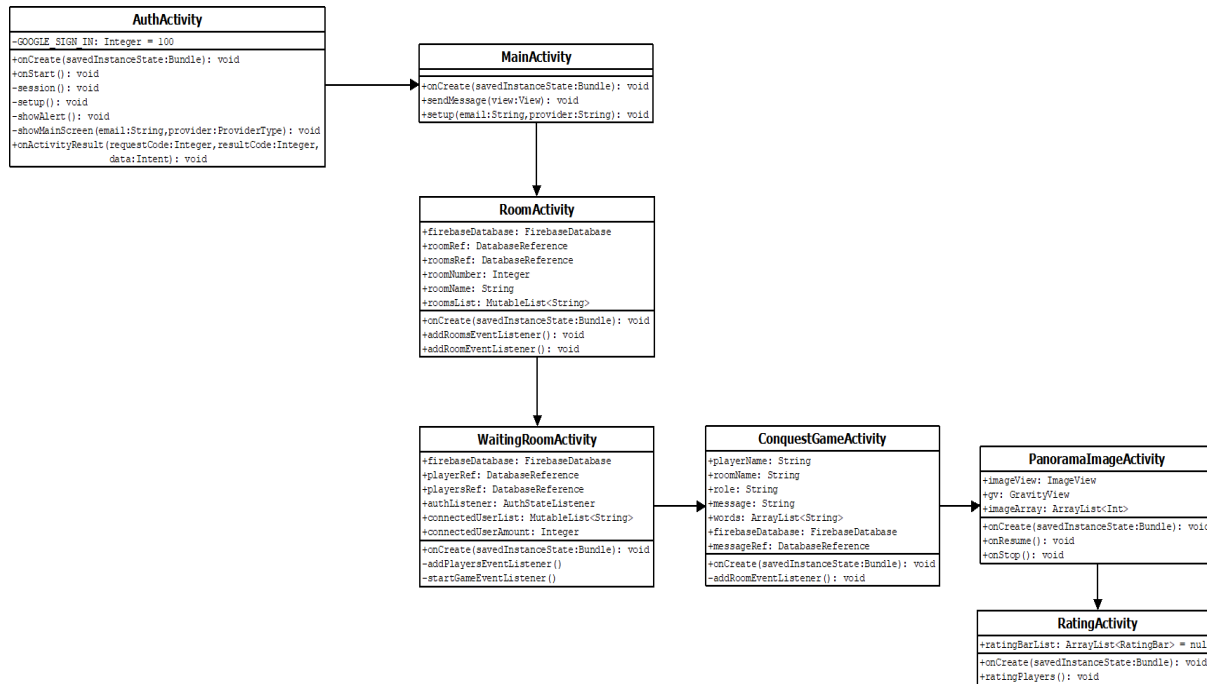


Figura 21: Diagrama de clases de diseño (UML)



### 5.1.3. Diagramas de Interacción

Los diagramas de interaccion que se presentan a continuación grafican el recorrido de accion de los contratos presentados anteriormente en las tablas ??, ??, ??, ??, ?? y ??.

1. **Cargar Imágenes:** El juego carga la imágenes una vez dentro de la partida, para así optimizar lo mejor posible el funcionamiento de la app, es por eso que, luego de ingresar a una partida en “RoomActivity”, el juego carga las imagenes en “ConquestGameActivity”.



Figura 22: Diegrama de interacción: Cargar Imagenes

2. **Inicio de juego:** El inicio de juego y todas sus verificaciones para el correcto funciona-  
miento ocurren dentro de la clase “MainActivity”, en esta clase, cuya función y tal como  
su nombre lo indica, hace de sala principal para redireccionar a las distintas instancias y  
demás clases.



Figura 23: Diegrama de interacción: Inicio de juego

3. **Inicio de Sesión:** El inicio de sesión ocurre de dos formas dependiendo si la aplicación reconoce un usuario ingresado anteriormente. Si se entra por primera vez, es decir, no hay recolección de una sesión iniciada el usuario pasa a la clase de identificación donde debe ingresar sus datos o su cuenta de google para pasar a la clase “MainActivity”. En el caso de que ya se encuentre un usuario se pasa directamente a “MainActivity”.



Figura 24: Diegrama de interacción: Inicio de Sesión

4. **Mostrar Palabras:** Luego de que todos los usuarios se encuentren en la sala, se da inicio a la partida, donde el primer paso es mostrarles las palabras que deben utilizar para crear la historia, esto se hace por medio de la clase “ConquestGameActivity” presentando el pool de palabras a los usuarios.



Figura 25: Diegrama de interacción: Mostrar Palabras

5. **Presentacion de puntaje:** Luego de ingresar y ver las historias entregadas por todos los usuarios se pasa a la clase “RatingActivity”, donde todos ponen una nota a las historias creadas. Luego de que la aplicación se asegure de que todos los votos han sido ingresados como feedback, se vuelve a la clase de “ConquestGameActivity” donde se da inicio a la siguiente partida.

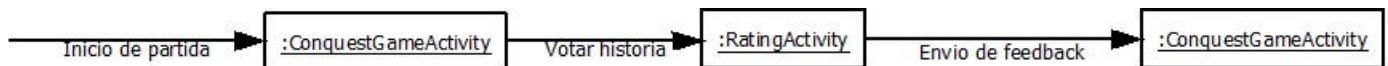


Figura 26: Diegrama de interacción: Presentacion de puntaje

6. **Ver Imagen 360°:** Una vez dentro de la partida la imagen del punto de conquista siempre estará visible para el jugador, y para mostrarla el juego debe acceder a la base de datos desde la clase “PanoramaImageActivity”, donde busca la imagen y la muestra al usuario.



Figura 27: Diegrama de interacción: Ver Imagen 360°

#### 5.1.4. Diagramas de Estados

Los diagramas de estado que se presentan en esta subseccion muestran el proceso de estados por los que pasa el programa segun sea el caso de uso en funcionamiento, mismos que se presentan en las tablas ??, ??, ??, ??, ?? y ??.



Figura 28: Diegrama de estado: Jugar un turno

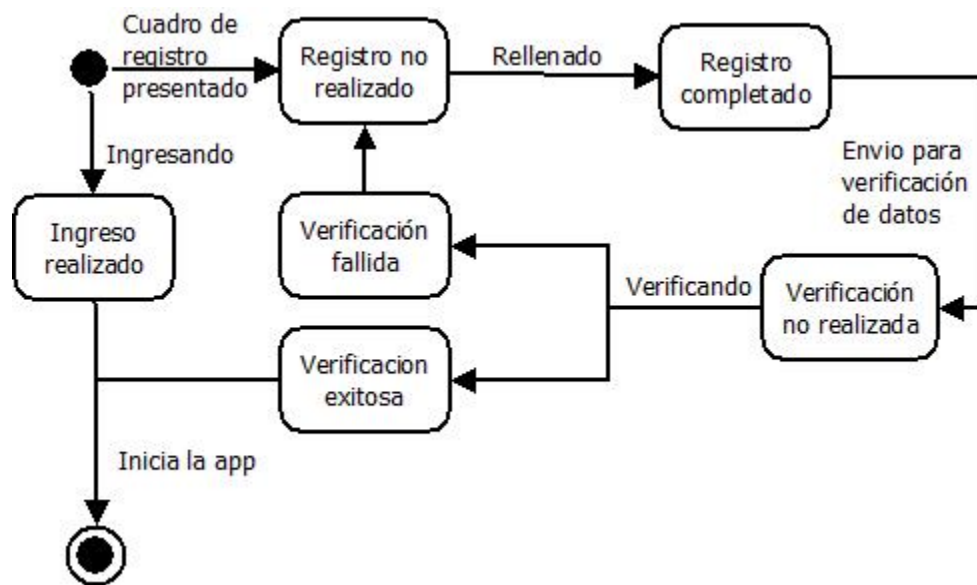


Figura 29: Diegrama de estado: Iniciar sesión

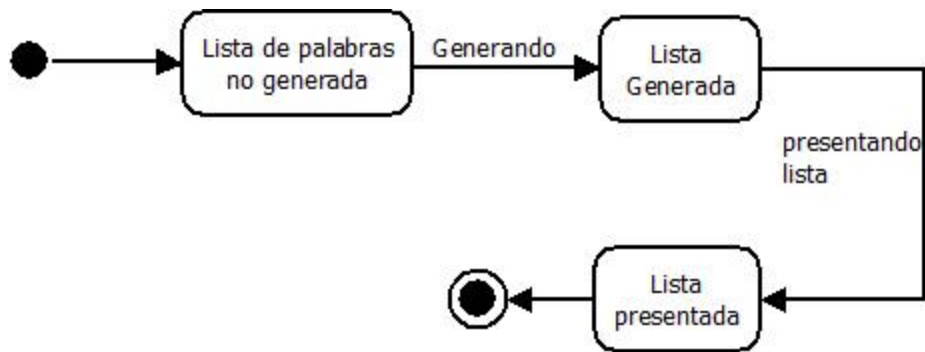


Figura 30: Diegrama de estado: Gnerar lista de palabras

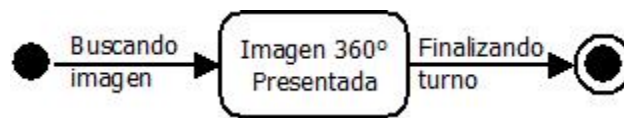


Figura 31: Diegrama de estado: Mostrar Imagen 360°

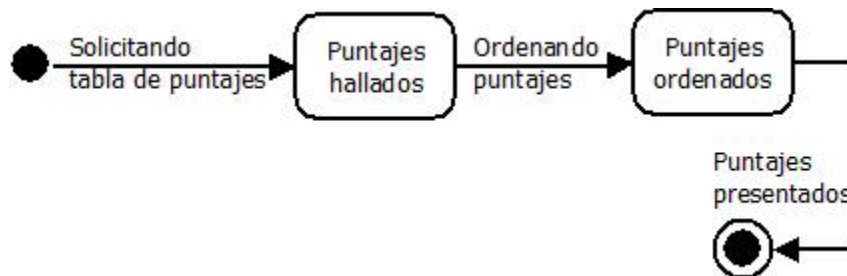


Figura 32: Diegrama de estado: Mostrar tabla de puntaje



Figura 33: Diegrama de estado: Puntuar jugador

## 5.2. Refinamientos

### 5.2.1. Lugar de Refinamiento

A pesar de no haberle dedicado mucho tiempo a esta sección en particular, nos hemos ido dando cuenta de ciertos puntos que requieren nuestra atención, mejorar la manera en que dividimos las tareas entre las distintas clases y mejorar la estructuración del código, son dos puntos que creemos merecen y pueden mejorarse aún, esto a su vez permitiría que el programa funcione con mayor fluidez.

### 5.2.2. Para cada Lugar

#### 5.2.2.1. Refinamientos considerados

Los refinamientos considerados fueron obtenidos gracias a las previas experiencias en programación y los conocimientos obtenidos a lo largo de nuestros estudios, todo eso en conjunto con el análisis realizado al proyecto y los atributos que se planean respetar he implementar.

Los refinamientos son los siguientes:

1. **Mejorar la estructuración del código:** Se planea ordenar y comentar más específicamente las clases, atributos y funciones en el código, con el fin de hacer mas sencillo manipular el codigo en caso de realizarle modificaciones o mantenimiento.
2. **Optimizar las clases:** La idea es que cada clase contenga solo lo que necesita poseer por si misma, pero tambien se debe alivianar la carga de cada clase, para esto se piensa en implementar mas clases, con el fin de hacer clases más específicas para cada cosa.

### 5.2.2.2. Selección y descripción de una opción

**Optimizar las clases:** La finalidad de esto, que fue explicada con anterioridad, sirve en gran medida para alivianar la carga de cada clase y hacer el código mas legible y menos pesado de entender.

## 6. Implantación

### 6.1. Código fuente completo (parcial)

```
package com.example.juegoconquista360

import ...

class ConquestGameActivity : AppCompatActivity() {

    //Variables de room y autenticación
    var playerName: String? = null
    var roleName: String? = null
    var role: String? = null
    var message: String? = null
    //arreglo de palabras
    var words = arrayOf("Mareado", "Lago", "Compañía", "Paseo", "Bus", "Limpio", "Hambre", "Atardecer", "Comida", "Turista", "Carta", "Música", "Cafetería", "Miedo", "Fauna",
        "Amigos", "Fotos", "Exótico", "Noche", "Soleado", "Edificio", "Cansado", "Internet", "Viaje", "Tren", "Vegetación", "Nublado", "Lluvia", "Fiesta", "Conversación", "Silueta", "Alegria")
    var used = arrayOf<String>()
    //Variable databases
    var firebaseDatabase: FirebaseDatabase? = null
    var messageRef: DatabaseReference? = null
    var psInt: Int? = null

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_conquest_game)

        //Referencias de widgets encontrados en xml
        val answer = findViewById<EditText>(R.id.et_history)
        val button = findViewById<Button>(R.id.bt_open_photo)
        val button2 = findViewById<Button>(R.id.bt_upload_history)

        //Acciones de botones
        button.setOnClickListener { itView:
            val intent = Intent(applicationContext, PanoramaImageActivity::class.java)
            startActivity(intent)
        }
    }
}
```

Figura 34: Código Clase ConquestGameActivity parte 1

```

val str = answer.text.toString()
button2.setOnClickListener { //View
    val intent = Intent(applicationContext, RateActivity::class.java)
    intent.putExtra( name: "Story",str)
    startActivity(intent)
}

//Acceso a la database para identificar host
firebaseDatabase = FirebaseDatabase.getInstance()
val prefs = getSharedPreferences("com.example.juegoconquista360.PREFERENCE_FILE_KEY", Context.MODE_PRIVATE).edit()
playerName = prefs.putString("playerName", "").toString()

val bundle = Bundle()
if(bundle!=null){
    roomName = bundle.getString( key: "roomName")
    role = if(roomName.equals(playerName)){ "host" }else{ "guest" }
}

// Boton de acceso a actividad ver imagen 360
bt_open_photo.setOnClickListener { //View
    button.isEnabled = false
    messageRef!!.setValue(message)
    val intent = Intent(applicationContext, PanoramaImageActivity::class.java)
    startActivity(intent)
}

//Recepcion de mensajes
messageRef = firebaseDatabase!!.getReference( path: "rooms/$roomName")

messageRef!!.child( pathString: "message").setValue(message)

// Creacion de lista de palabras random
for(x in 0 until 5)
{
    c = Random.nextInt( range: 0 until words.size)
    used.add(words[c!!!])
    words.removeAt(c!!!)
}

```

Figura 35: Codigo Clase ConquestGameActivity parte 2

```

word1.text = used[0]
word2.text = used[1]
word3.text = used[2]
word4.text = used[3]
word5.text = used[4]

messageRef!!.child( pathString: "word1").setValue(used[0])
messageRef!!.child( pathString: "word2").setValue(used[1])
messageRef!!.child( pathString: "word3").setValue(used[2])
messageRef!!.child( pathString: "word4").setValue(used[3])
messageRef!!.child( pathString: "word5").setValue(used[4])

addRoomEventListener()

}

//Envio de datos a firebase
private fun addRoomEventListener() {
    messageRef!!.addValueEventListener(object : ValueEventListener {
        override fun onCancelled(error: DatabaseError) {
            //Error
            Toast.makeText(applicationContext, text: "Error!", Toast.LENGTH_SHORT).show()
        }

        override fun onDataChange(dataSnapshot: DataSnapshot) {
            //message received
            dataSnapshot.child( path: "word1").value
            dataSnapshot.child( path: "word2").value
            dataSnapshot.child( path: "word3").value
            dataSnapshot.child( path: "word4").value
            dataSnapshot.child( path: "word5").value
        }
    })
}

```

Figura 36: Codigo Clase ConquestGameActivity parte 3

## 6.2. Dependencias



## 7. Anexos

### 7.1. Glosario

Base de datos : Conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

Código fuente : Conjunto de líneas de texto con los pasos que debe seguir la computadora para ejecutar un programa.

Diagrama de clases : Es un tipo de diagrama de estructura estática que describe la estructura de un sistema mostrando las clases del sistema, sus atributos, operaciones (o métodos), y las relaciones entre los objetos.

ID : Identificador de seguridad.

Iteración : Repetición, reiteración.

Procesador : Unidad central de procesamiento (CPU) interpreta las instrucciones y procesa los datos de los programas de computadora.

Prototipo : Primer ejemplar que se fabrica de una figura, un invento u otra cosa, y que sirve de modelo para fabricar otras iguales, o molde original con el que se fabrica.

Sistema operativo : Conjunto de órdenes y programas que controlan los procesos básicos de una computadora y permiten el funcionamiento de otros programas.

Software : Conjunto de programas y rutinas que permiten a la computadora realizar determinadas tareas.